Fraunhofer
SCAI

Marco Hülsmann

Effiziente und neuartige Verfahren zur Optimierung von Kraftfeldparametern bei atomistischen Molekularen Simulationen kondensierter Materie

FRAUNHOFER VERLAG

Fraunhofer-Institut
für Algorithmen und Wissenschaftliches Rechnen SCAI

Effiziente und neuartige Verfahren zur Optimierung von Kraftfeldparametern bei atomistischen Molekularen Simulationen kondensierter Materie

von Marco Hülsmann

FRAUNHOFER VERLAG

Kontaktadresse:
Fraunhofer-Institut für Algorithmen
und Wissenschaftliches Rechnen SCAI
Schloss Birlinghoven
53754 Sankt Augustin
Telefon: 02241 14-2500
Telefax: 02241 14-2460
info@scai.fraunhofer.de
www.scai.fraunhofer.de

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.
ISBN: 978-3-8396-0426-7

D 38

Zugl.: Köln, Univ., Diss., 2012

Druck: Mediendienstleistungen des
Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart

Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 80 04 69, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-25 00
Telefax 07 11 9 70-25 08
E-Mail verlag@fraunhofer.de
URL http://verlag.fraunhofer.de

Effiziente und neuartige Verfahren zur Optimierung von Kraftfeldparametern bei atomistischen Molekularen Simulationen kondensierter Materie

Inaugural-Dissertation

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Marco Hülsmann

aus Düren

Erstgutachter:	Prof. Dr. Ulrich Trottenberg, Uni Köln
Zweitgutachterin:	Prof. Dr. Caren Tischendorf, Uni Köln
Externer Drittgutachter:	Prof. Dr. Florian Müller-Plathe, TU Darmstadt

Tag der mündlichen Prüfung: 16. Mai 2012

Vorwort des Institutsleiters

In der Computerchemie werden statt arbeits- und kostenaufwendiger Laborexperimente numerische Simulationen im Computer durchgeführt. Die vorliegende Arbeit behandelt ein zentrales Problem der Computerchemie im Bereich der molekularen Simulationen: die Bestimmung geeigneter Parameter für das Kraftfeld, das die Wechselwirkungen zwischen Atomen innerhalb eines Moleküls (intramolekuar) und zwischen verschiedenen Molekülen (intermolekular) beschreibt.

Die Parameterbestimmung wird in dieser Arbeit als mathematisches Optimierungsproblem interpretiert und behandelt. Schwierigkeiten bestehen vor allem in der außerordentlichen Komplexität der Fragestellung, aber auch in der Notwendigkeit, das statistische Rauschen der Zielgrößen zu beherrschen. Bei den hier vorgestellten Optimierungsalgorithmen werden unter anderem ableitungsfreie Verfahren auf der Basis des aktuellen „Dünne-Gitter-Ansatzes“ benutzt.

Die Besonderheit dieser Arbeit besteht in ihrem interdisziplinären Charakter zwischen zentralen industrierelevanten Fragestellungen der Chemie und modernen Methoden der numerischen Mathematik. Die Arbeit entstand in enger Kooperation zwischen dem Mathematischen Institut der Universität zu Köln und der Abteilung „Simulationsanwendungen“ des Fraunhofer-Instituts für Algorithmen und Wissenschaftliches Rechnen (SCAI). Sie wurde von der Mathematisch-Naturwissenschaftlichen Fakultät der Universität zu Köln im Sommersemester 2012 als Dissertation angenommen.

Ulrich Trottenberg

Für Maurice

Kurzzusammenfassung

Molekulare Simulationen ermöglichen es, den Einfluß mikroskopischer Prozesse auf makroskopische Phänomene zu studieren. Um Simulationen in den Naturwissenschaften bis hin zur Verfahrenstechnik erfolgreich anwenden zu können, müssen geeignete molekulare Modelle vorliegen. Das Fundament einer Simulation zur quantitativen Vorhersage von physikalischen Eigenschaften ist das sogenannte Kraftfeld. Letzteres beschreibt sowohl intra- als auch intermolekulare Wechselwirkungen. Die Hauptschwierigkeit liegt in der Parametrisierung eines Kraftfelds, insbesondere des intermolekularen Teils. Dies per Hand durchzuführen, ist äußerst zeitaufwendig, da für jeden Satz von Kraftfeldparametern eine rechenaufwendige Molekulare Simulation durchzuführen ist. Um ein sehr kleines System, bestehend aus beispielsweise 1000 kleinen Molekülen, eine Nanosekunde lang auf einem modernen Parallelrechner zu simulieren, sind bereits zwei bis vier Stunden notwendig. In dieser Arbeit wird ein neues automatisiertes Parametrisierungsschema vorgeschlagen, welches auf der Formulierung und Lösung eines mathematischen Optimierungsproblems basiert. Während intramolekulare Freiheitsgrade und atomare Partialladungen mithilfe der Quantenmechanik berechnet werden können, ist die Einstellung von Parametern zur Beschreibung intermolekulare Wechselwirkungen keineswegs trivial. Innerhalb des Optimierungsprozesses werden Eigenschaften, die aus einer Molekularen Simulation resultieren, wie Dichte, Verdampfungsenthalpie, Selbstdiffusionskoeffizient und Dampfdruck, an ihre entsprechenden experimentellen Referenzdaten angepaßt. Da Molekulare Simulationen eine hohe Komplexität aufweisen und die resultierenden Eigenschaften mit statistischem Rauschen behaftet sind, wird das Optimierungsproblem sowohl mit bereits vorhandenen als auch mit neuartigen, effizienten und robusten numerischen Algorithmen gelöst. Ein weiteres Ziel dieser Arbeit besteht darin, die optimierten Kraftfelder bezüglich ihrer physikalischen und chemischen Anwendbarkeit zu evaluieren.

Abstract

Molecular simulations enable studies of the impact of microscopic processes on macroscopic phenomena. In order to be able to apply simulations successfully within natural science and process engineering, appropriate molecular models have to be present. The foundation of a simulation to predict physical properties quantitatively is the so-called force field. The latter describes both intramolecular and intermolecular interactions. The main difficulty lies in the parameterization of a force field, especially of the intermolecular part. To do this manually is extremely time consuming, as for each set of force field parameters, a numerically complex molecular simulation has to be performed. Already two to four hours are required for the simulation of a very small system, e.g. consisting of 1000 small molecules, during one nanosecond on a modern parallel high-performance computer. In this thesis, a new automated parameterization scheme based on the formulation and solution of a mathematical optimization problem is proposed. While intramolecular degrees of freedom and partial atomic charges can be calculated by quantum mechanics, the calibration of parameters describing intermolecular interactions, is not trivial at all. Within the optimization procedure, properties resulting from a molecular simulation like density, enthalpy of vaporization, self-diffusion coefficient, and vapor pressure, are fitted to their respective experimental reference data. As molecular simulations exhibit a high complexity and the resulting properties are affected by statistical noise, the optimization problem is solved by both already existing and noval, efficient, and robust numerical algorithms. Another goal of this thesis is to evaluate the optimized force fields with regard to their physical and chemical applicability.

Inhaltsverzeichnis

1 Einleitung

Computersimulationen haben sowohl aus wissenschaftlicher als auch aus industrieller Sicht vor allem mit dem Aufkommen von Hochleistungsrechenclustern und der damit verbundenen Möglichkeit zur Parallelisierung unabhängiger Rechnungen in den letzten Jahrzehnten enorm an Bedeutung gewonnen. Ein wichtiger Teilbereich der Computersimulationen sind Molekulare Simulationen, welche es ermöglichen, Auswirkungen von Änderungen innerhalb von mikroskopischen Zuständen auf makroskopische Systemeigenschaften zu studieren. Im Gegensatz zu Simulationen in der Verfahrenstechnik wird nicht mehr die Kontinuumsebene betrachtet, sondern es werden Systeme auf molekularer Ebene modelliert. Das Ziel besteht darin, interatomare und -molekulare Wechselwirkungen so zu beschreiben, daß einerseits bestimmte Genauigkeitsanforderungen erfüllt werden und andererseits der damit verbundene Rechenaufwand möglichst gering gehalten wird, denn auch auf hochmodernen Rechenclustern sind Molekulare Simulationen stets numerisch gesehen äußerst aufwendig. Molekulare Simulationen sollen außerdem prädiktiv sein, das heißt, ein bereits bestehendes akkurates molekulares Modell soll dazu in der Lage sein, eine Vielzahl an thermodynamischen Stoffdaten eines Systems, bestehend aus einem oder mehreren Stoffen, vorherzusagen. Die industrielle Relevanz Molekularer Simulationen zeigt sich dadurch, daß arbeits- und kostenaufwendige Experimente im Labor eingespart werden können. So können Systeme bei nur schwer realisierbaren Drücken und Temperaturen simuliert werden, die Eigenschaften von sehr toxischen Stoffen können ohne jedwede Risiken und ohne Berücksichtigung wichtiger Vorbeugungsmaßnahmen zur Durchführung von Experimenten ermittelt werden, und Vorgänge an Grenzflächen oder innerhalb von Membranen sind auf mikroskopischer Ebene im Detail beobachtbar. Ein weiterer Vorteil von Molekularen Simulationen gegenüber Experimenten besteht darin, daß Ort und Geschwindigkeit sämtlicher Teilchen regelmäßig nach sehr kleinen Zeitintervallen gespeichert werden. Daher lassen sich mikroskopische Veränderungen innerhalb eines Systems äußerst detailliert beobachten. All dies hat dazu geführt, daß sich Molekulare Simulationen neben Theorie und Experiment zu einem eigenständigen Wissenszweig entwickelt und etabliert haben. Aufgrund der stetig ansteigenden Rechenkapazität moderner Computer werden darüber hinaus die Anwendungsbereiche Molekularer Simulationen in allen Naturwissenschaften in naher und ferner Zukunft höchstwahrscheinlich weiterhin kontinuierlich wachsen (Allen u. Tildesley, 1987; Frenkel u. Smit, 2006).

1.1 Kraftfelder – Kernelement Molekularer Simulationen

Mithilfe der Quantenmechanik ist es prinzipiell möglich, Wechselwirkungen innerhalb eines Systems exakt zu beschreiben. Dabei ist eine Differentialgleichung, die so genannte *Schrödinger-Gleichung*, zu lösen. Da die Lösung dieser Gleichung gerade bei Vielteilchensystemen zu schwierig ist, geht man über zur Klassischen Mechanik, wo das Problem vereinfacht wird. Dies hat den zusätzlichen Vorteil, daß man nicht mehr zwischen Kernen und Elektronen unterscheiden

muß, sondern ein vollständiges Atom als kleinste Einheit betrachtet wird. In der Statistischen Mechanik, einem Teilgebiet der Klassischen Mechanik, gibt es Molekulare Simulationstechniken, mithilfe derer eine Vielzahl an makroskopischen physikalischen Eigenschaften ermittelt werden kann. Die beiden wichtigsten Simulationsmethoden sind die Molekulardynamik und Monte-Carlo-Simulationen, welche auch in dieser Arbeit eine wesentliche Rolle spielen. Wechselwirkungen zwischen Atomen innerhalb desselben und verschiedener Moleküle werden bei Molekularen Simulationen durch sogenannte *Kraftfelder* beschrieben. Ein Kraftfeld besteht aus einer analytischen Form und aus bestimmten einzustellenden Parametern. Diese sind in zwei Kategorien einteilbar: Einige Parameter beschreiben intramolekulare und andere intermolekulare Wechselwirkungen. Die Realität durch Kraftfelder zu beschreiben, ist zwar physikalisch motiviert, stellt jedoch lediglich eine Approximation dar, das heißt, nicht alle Kraftfeldparameter sind physikalisch ableitbar. Durch quantenmechanische Methoden sind insbesondere intermolekulare Kraftfeldparameter nicht bestimmbar. In diesen Fällen bedient man sich sogenannter *empirischer* Methoden, das heißt, es sind experimentelle Stoffdaten erforderlich, und die entsprechenden Kraftfeldparameter sind so zu justieren, daß die auf dem zugehörigen Kraftfeld basierende Molekulare Simulation die experimentellen Daten möglichst genau wiedergibt. Dies kann beispielsweise manuell erfolgen oder durch einen Transfer von Kraftfeldparametern. Letzteres bedeutet, daß publizierte und somit erprobte Kraftfeldparameter für gewisse Atomtypen einer bestimmten Substanz auf die Atome desselben Typs innerhalb einer anderen Substanz übertragen werden. Ob das Kraftfeld der neuen Substanz tatsächlich verwendbar ist, muß in einem nachfolgenden Schritt in Bezug auf gewisse physikalische Eigenschaften mittels Molekularer Simulationen überprüft werden. Oftmals sind transferierte Kraftfeldparameter nachzujustieren. Dies manuell zu versuchen, ist in vielen Fällen zu aufwendig und teilweise unsystematisch, da dies stets durch Probieren geschehen muß.

In dieser Dissertation wird daher ein systematischerer Ansatz verfolgt: Für jedes Kraftfeld, welches innerhalb einer Molekularen Simulation verwendet wird, werden die aus der Simulation ermittelten physikalischen Stoffdaten mit experimentell bestimmten Referenzwerten verglichen. Dadurch läßt sich ein mathematisches Optimierungsproblem formulieren und lösen, bei dem eine Zielfunktion mittels numerischer Optimierungsmethoden minimiert wird. Bei letzterer handelt es sich um eine Fehlerfunktion zwischen experimentellen und per Simulation berechneten physikalischen Stoffdaten. Ein derartiges Optimierungsproblem zu lösen, ist keineswegs trivial: Zum einen sind Molekulare Simulationen äußerst rechenaufwendig, und für jede Auswertung der Zielfunktion innerhalb des numerischen Optimierungsalgorithmus sind eine oder mehrere Simulationen durchzuführen. Weiterhin sind die resultierenden simulierten Zielgrößen mit statistischem Rauschen behaftet, da sie mithilfe von Durchschnitten über eine endliche Anzahl an Stichproben approximiert werden. Ein weiteres Problem ist die Tatsache, daß stets geeignete Startparameter zur Lösung des Optimierungsproblems notwendig sind. Zwar gibt es in der Literatur bereits Standardkraftfelder für eine Vielzahl an Atomtypen, allerdings sind diese bei weitem nicht auf alle chemischen Substanzen anwendbar. Aufgrund dieser drei Problematiken müssen die gesuchten numerischen Verfahren zumindest die folgenden Eigenschaften erfüllen: Sie müssen effizient sein, das heißt mit möglichst wenig Funktionsevaluationen zum Ziel führen, robust in Bezug auf statistisches Rauschen und möglichst startwertunabhängig.

Im letzten Jahrzehnt ist die Minimierung einer derartigen Zielfunktion, aus der optimale Kraftfelder erhalten werden können, auf ein sehr großes Interesse im Bereich Molekularer Simulationen gestoßen, denn ohne optimal eingestellte Kraftfeldparameter kann nicht erwartet werden, daß Simulationen quantitativ korrekte Ergebnisse liefern oder gar prädiktiv sind. Die Parametrisierung von Kraftfeldern ist somit unentbehrlich, und aufgrund der hohen Schwierigkeit und Komplexität ist sie zu einem eigenen Forschungsfeld geworden. Ein Physiker oder Chemiker, der Molekulare Simulationen als Werkzeug für seine Studien verwendet, ist daran interessiert, ohne großen Arbeitsaufwand an optimale Kraftfeldparameter zu gelangen, welche jedoch für ihn unbedingt erforderlich sind. Ein Kraftfeld manuell zu justieren, würde beispielsweise zu viel Zeit kosten, und die Lösung eines mathematischen Optimierungsproblems liegt zu weit enfernt von seiner Expertise. Daher sollte die Kraftfeldoptimierung die zusätzliche Eigenschaft besitzen, daß sie automatisiert erfolgen kann und leicht handhabbar ist.

1.2 Ziel der Dissertation und Vorgehensweise

Der in Abschnitt 1.1 motivierte Optimierungsprozeß sollte eine Vielzahl an physikalischen Zielgrößen simultan reproduzieren, insbesondere auch zu verschiedenen Temperaturen und Drücken. Weiterhin sollte er substanzunabhängig sein. Es wurden bereits einige wenige mathematische Verfahren eingesetzt, um derartige Optimierungsprobleme zu lösen. Es handelt sich dabei um das Simplex-Verfahren nach Nelder und Mead und eine etwas abgewandelte Variante eines Gauß-Newton-Verfahrens. Diese Verfahren weisen jedoch gewisse, teilweise schwerwiegende Nachteile auf. Sie sind allesamt nicht dazu in der Lage, alle hier an die Optimierung gestellten Anforderungen zu erfüllen. Ziel der Dissertation ist es daher, bestehende beziehungsweise neuartige effiziente numerische Optimierungsverfahren automatisiert einzusetzen beziehungsweise zu entwickeln, so daß die vom Anwender gewünschten physikalischen Eigenschaften für die jeweils gewünschten Moleküle mit möglichst wenig Rechenaufwand an die zugehörigen experimentellen Referenzdaten angepaßt werden können. Dabei steht neben den mathematischen Ansätzen vor allem die Anwendbarkeit der resultierenden Kraftfeldoptimierung im Bereich Molekularer Simulationen im Vordergrund. Sämtliche hier erhaltenen Kraftfelder sind daher sowohl in Bezug auf Effizienz und Robustheit ihres zugehörigen Optimierungsprozesses als auch in Bezug auf ihre praktische Anwendbarkeit zu evaluieren.

Abbildung 1.1 zeigt die in dieser Arbeit zu realisierende Verknüpfung zwischen Simulation und Optimierung. Bei der Vorgehensweise wird zwischen globalen und lokalen Optimierungsalgorithmen unterschieden. *Globale* Optimierungsmethoden sollen vor allem dann Einsatz finden, wenn keine geeigneten Startparameter gefunden werden können. Hier werden konkret ein stochastisch basierter evolutionärer Algorithmus und eine auf Interpolation beruhende Metamodellierung betrachtet, da diese mithilfe von nur wenigen Auswertungen der zu minimierenden Zielfunktion dazu in der Lage sind, in die Nähe eines globalen Minimums zu gelangen. Ein Ansatz, eine effizientere *lokale* Optimierung zu erhalten, wird mithilfe von gradientenbasierten Verfahren realisiert werden, da sich diese durch sehr gute Konvergenzeigenschaften auszeichnen. Um nicht nur mit möglichst wenig Iterationen, sondern auch Funktionsauswertungen auszukommen, wird weiterhin untersucht werden, ob und inwieweit ein Gradient beziehungsweise eine Hesse-Matrix ohne zusätzlichen Simulationsaufwand berechnet werden kann. Anhand der Struktur

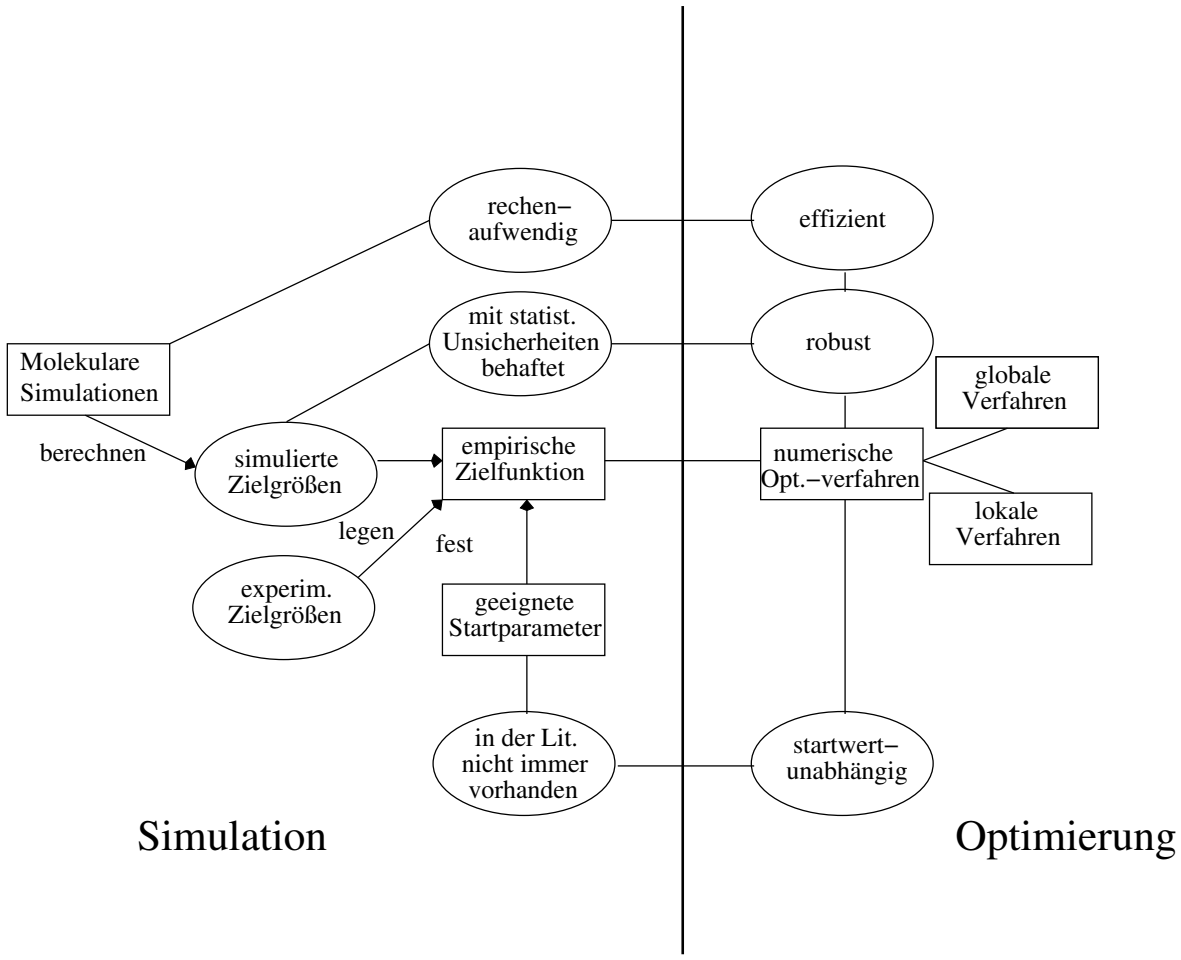


Abbildung 1.1: In dieser Arbeit zu realisierende Verknüpfung zwischen Simulation und Optimierung. Die Gegebenheiten auf der Simulationsseite erfordern die angegebenen Eigenschaften für die einzusetzenden numerischen Optimierungsverfahren.

der Zielfunktion wird ebenfalls eingehend analysiert werden, inwieweit sich eine Kombination aus globalen und lokalen Optimierungsmethoden eignet.

Da sich die Berechnung eines Gradienten beziehungsweise einer Hesse-Matrix aufgrund des statistischen Rauschens besonders in der Nähe eines globalen oder lokalen Minimums als problematisch erweist, wird auch die Suche nach einem effizienten ableitungsfreien Verfahren äußerst bedeutsam. Im Rahmen dieser Dissertation wird daher ein neuartiges lokal konvergentes ableitungsfreies Verfahren entwickelt, welches im Vergleich zu den betrachteten gradientenbasierten Verfahren zum Ziel hat, noch näher und robuster an das Minimum zu gelangen sowie die Anzahl an zur Optimierung notwendiger Molekularer Simulationen noch zu verringern. Inwieweit dies realisiert werden kann, ist in dieser Dissertation eingehend zu studieren. Ein Konvergenzbeweis wird ebenfalls durchgeführt. Sämtliche in dieser Dissertation behandelten Optimierungsverfahren werden sowohl theoretisch als auch praktisch in Bezug auf ihre Robustheit gegenüber statistischem Rauschen analysiert.

1.3 Gliederung

In Kapitel 2 werden zunächst die Grundlagen Molekularer Simulationen vorgestellt. Dabei stehen insbesondere die in dieser Arbeit verwendeten Methoden und deren Effizienz im Vordergrund. Die beiden wichtigsten Kategorien, Molekulardynamik und Monte-Carlo-Simulationen, werden detailliert eingeführt. Kapitel 3 behandelt die Kraftfeldoptimierung durch auf Molekulare Simulationen angepaßte Optimierungsverfahren. Dabei werden verschiedene globale und gradientenbasierte lokale Optimierungsverfahren dargestellt, nachdem die Nachteile bereits verwendeter anderer Methoden diskutiert wurden. Die Erhöhung der Effizienz derartiger Verfahren sowie die Behandlung von statistischem Rauschen werden in diesem Kapitel detailliert behandelt. In Abschnitt 3.5 wird dargelegt, wie die Anpassung gradientenbasierter Verfahren an Molekulare Simulationen in dieser Dissertation erfolgt. Sämtliche folgenden Algorithmen und Ideen sind, falls nicht anders angegeben, ab diesem Abschnitt in der vorliegenden Arbeit entwickelt worden. Um Verfahren praktisch detailliert zu evaluieren, sind Molekulare Simulationen zu rechenintensiv. Daher werden diese zunächst geeignet ersetzt werden. In Kapitel 4 werden die einzelnen eingesetzten Verfahren und deren Varianten aus Kapitel 3 anhand derartiger simulierter Simulationen eingehend bewertet. Die Anwendbarkeit auf Molekulare Simulationen wird dann in Kapitel 5 diskutiert. Dort werden auch die in dieser Arbeit erhaltenen optimalen Kraftfelder angegeben und evaluiert. Das neuartige ableitungsfreie Verfahren wird in Kapitel 6 eingeführt, wobei die Nachteile gradientenbasierter Verfahren sowie die Eigenschaften und Vorteile dieses Verfahrens auf theoretischer Ebene detailliert analyisiert werden. Zum Schluß wird der bereits erwähnte Konvergenzbeweis durchgeführt. Kapitel 7 behandelt die praktische Anwendung der ableitungsfreien Methode sowie die Fragestellung, inwieweit die an das Verfahren gesetzten Ziele in der Praxis erreicht werden können. In Kapitel 8 werden schließlich die Ergebnisse der vorliegenden Dissertation zusammengefaßt sowie kritisch beleuchtet, und es wird ein Ausblick in Bezug auf weiterführende Untersuchungen gegeben.

2 Molekulare Simulationen

Dieses Kapitel befaßt sich mit den Grundlagen im Bereich der Molekularen Simulationen und geht auf Allen u. Tildesley (1987), Jensen (1999), Kuypers (1993) und Müller-Plathe (2001) zurück. Es erläutert schrittweise, wie makroskopische physikalische Eigenschaften aus Zuständen auf atomistischer Ebene zu berechnen sind. In Abschnitt 2.1 wird beschrieben, wie man von der Lösung der Schrödinger-Gleichung in der Quantenmechanik zur Lösung der Newtonschen Bewegungsgleichung in der Klassischen Mechanik gelangt. Zur Lösung letzterer Differentialgleichung ist zunächst die potentielle Energie $U(r)$ zu determinieren, da diese von zentraler Bedeutung ist. Da die Entwicklung der Zustände im Raum aller möglichen Zustände eines Systems, dem sogenannten *Phasenraum*, durch Kraftfelder approximiert wird, werden in Abschnitt 2.2 zunächst verschiedene Kraftfelder und Potentiale eingeführt. Es wird dabei zwischen intra- und intermolekularen Wechselwirkungen unterschieden, wobei bezüglich letzteren das Lennard-Jones-Potential (Abschnitt 2.2.2) in dieser Arbeit im Vordergrund steht. Ist die potentielle Energie definiert, so kann diese zur numerischen Lösung der zeitabhängigen Newtonschen Bewegungsgleichung verwendet werden: In Abschnitt 2.3 wird das Verfahren der *Molekulardynamik* im Detail beschrieben, welches die zeitliche Entwicklung von Teilchenkoordinaten berechnet. Die daraus resultierenden zeitabhängigen Raumkurven werden *Trajektorien* genannt. In Abschnitt 2.4 werden *Monte-Carlo-Methoden* eingeführt, welche Zustände im Phasenraum per Zufall verändern. Zu jedem Zeitschritt (bei MD-Simulationen) beziehungsweise jedem Zustand (bei MC-Simulationen) lassen sich statische und im Falle von MD auch dynamische Eigenschaften ableiten. Die Berechnung der wichtigsten Systemeigenschaften wird schließlich in Abschnitt 2.5 erläutert. Sie ergeben sich als Durchschnitte über einzelne zeitliche oder örtliche Zustände aus dem Verlauf der jeweiligen Simulation. Da die Effizienz numerischer Verfahren in dieser Arbeit im Vordergrund steht, werden praktische Aspekte in Bezug auf die Durchführung Molekularer Simulationen am Ende des Kapitels in Abschnitt 2.6 diskutiert.
Die Umsetzung von Simulationen sämtlicher in dieser Arbeit relevanten Ensembles sowie einige moderne Simulationstechniken sind in Anhang A beschrieben. Methoden zur effizienten Berechnung von Kräften und Potentialen werden in Anhang E dargestellt. Die Simulationspakete, welche in dieser Arbeit eine Rolle spielen, sind in Anhang F angegeben.

2.1 Von der Quantenmechanik zur Klassischen Mechanik

Die Theorie der Quantenmechanik führt auf die sogenannte *Schrödinger-Gleichung*, die auf Erwin Schrödinger (1881–1961) zurückgeht und erstmals im Jahre 1926 veröffentlicht wurde. Die erste Darstellung der Schrödinger-Gleichung, die heute als *Fundamentalgleichung der Quantenmechanik* bezeichnet wird, ist in Schrödinger (1926) zu finden. Es handelt sich hierbei um eine lineare partielle Differentialgleichung zweiter Ordnung. Für den einfachen Fall eines Wasser-

stoffatoms lautet ihre Form:

$$i\hbar\frac{\partial}{\partial t}\Psi(r,t) = \left(-\frac{\hbar^2}{2m}\Delta + \hat{U}(r,t)\right)\Psi(r,t) \tag{2.1}$$

Dabei ist $\hbar := \frac{h}{2\pi}$, wobei $h \approx 6.63 \times 10^{-34}$ Js das *Plancksche Wirkungsquantum*, m die Masse eines Teilchens, r sein Ort, $\hat{U}$ der Operator der potentiellen Energie, Δ der Laplace-Operator und t die Zeit. Die komplexwertige Lösung $\Psi(r,t)$ wird *Wellenfunktion* genannt. Der sogenannte *Hamilton-Operator*

$$\hat{H}(r,t) := -\frac{\hbar^2}{2m}\Delta + \hat{U}(r,t) \tag{2.2}$$

bestimmt die Dynamik und die möglichen Energien des Systems. Für den Hamilton-Operator eines Gesamtsystems aus N Teilchen gilt weiterhin:

$$\hat{H} = \hat{K} + \hat{U}, \tag{2.3}$$

wobei $\hat{K} := \sum_{i=1}^{N} \frac{\hbar^2}{2m_i}\Delta_i$ der Operator der kinetischen und $\hat{U} := \sum_{i=1}^{N}\sum_{j>i}^{N} \hat{U}_{ij}$ der Operator der potentiellen Energie des Systems sind. Letztere beschreibt die Interaktionen sämtlicher Teilchen. Der Index i bezieht sich dabei auf ein einzelnes Teilchen i und der Index ij auf die Interaktionen zweier Teilchen i und j. Die Schrödinger-Gleichung beschreibt allgemein das Verhalten eines *quantenmechanischen* Systems. Ist der Hamilton-Operator unabhängig von t, gilt also $\bar{H}(r,t) = \bar{H}(r)$, so erhält man mit dem Separationsansatz $\Psi(r,t) = \Phi(r)\exp(-iEt/h)$ die *zeitunabhängige Schrödinger-Gleichung*:

$$\hat{H}(r)\Phi(r) = E\Phi(r). \tag{2.4}$$

Diese beschreibt die Teilchen-Wellen-Dualität der Elektronen. Es ist zu beachten, daß der zeitabhängige Faktor $\exp(-iEt/h)$ ein Phasenfaktor ist. Gleichung (2.4) ist eine Eigenwertaufgabe, wobei E die Energieeigenwerte und $\Phi(r)$ die zugehörigen Eigenfunktionen sind. Die Lösungen Ψ sind physikalisch zunächst nicht interpretierbar. Durch die Normierung

$$\int_{\mathbb{R}^3} |\Psi(r,t)|^2 d^3r = 1 \tag{2.5}$$

ist $|\Psi|^2$ die Wahrscheinlichkeit, ein Teilchen an einem bestimmten Ort r zu finden. Gleichung (2.5) sagt aus, daß die Wahrscheinlichkeit, ein Teilchen irgendwo im Raum zu finden, bei 100% liegt. Dies klingt zunächst plausibel, ist aber nur aufgrund des *Noether-Theorems* gewährleistet, welches besagt, daß diese Wahrscheinlichkeit eine Erhaltungsgröße ist.

Die Schrödinger-Gleichung ist allerdings lediglich für einfache Potentiale exakt lösbar und auch nur für einfache Teilchen, wie zum Beispiel Wasserstoffmoleküle. Wäre für jedes beliebige System der Hamilton-Operator und die Lösung der Schrödinger-Gleichung bekannt, so wäre jedes System exakt simulierbar und jegliches Approximationsmodell wäre unnötig. Dies ist jedoch in der Realität leider nicht der Fall: In der Praxis ist es unmöglich, die Schrödinger-Gleichung auch nur näherungsweise für 'große' Systeme zu lösen, das heißt für Systeme, die aus einigen hundert Atomen bestehen. Außerdem müssen in der Quantenmechanik sowohl sämtliche Kerne als auch Elektronen mitberücksichtigt werden. Die erste Approximation der Schrödinger-Gleichung geht auf Born u. Oppenheimer (1927) zurück. Diese sogenannte *Born-Oppenheimer-Approximation*

beruht auf der Annahme, die Bewegungen der Kerne von denen der Elektronen trennen zu können:

$$\Psi_{tot} = \Phi_n(R) \cdot \Phi_e(R, r), \tag{2.6}$$

wobei e für Elektronen und n für Kerne (Nuclei) steht. Sie erfolgt in zwei aufeinanderfolgenden Schritten:

1. Die Konfiguration der Kerne wird fixiert, und es wird nur die *elektronische* Schrödinger-Gleichung
$$\hat{H}_e \Phi_e(R, r) = E_e(R) \Phi_e(R, r) \tag{2.7}$$
gelöst. Dabei steht R für die Positionen der Kerne und r für die der Elektronen. Es resultiert Φ_e. Aufgrund von Gleichung (2.3) gilt mit derselben Indizierung für den Hamilton-Operator:
$$\hat{H}_{tot} = \hat{H}_e + \hat{K}_n, \ \hat{H}_e = \hat{K}_e + \hat{U}_{ne} + \hat{U}_{ee} + \hat{U}_{nn}, \tag{2.8}$$
wobei zum Beispiel $\hat{U}_{ne}$ die potentielle Energie bezüglich den Interaktionen zwischen Kernen und Elektronen ist. Die sich nur auf die Kerne beziehende kinetische Energie $\hat{K}_n$ wird von $\hat{H}_{tot}$ subtrahiert.
2. Nun wird $\hat{K}_n$ wieder eingeführt, und man erhält die *nukleare* Schrödinger-Gleichung:
$$(\hat{K}_n + E_e(R)) \Phi_n(R) = E_{tot} \Phi_n(R) = \hat{H}_{tot} \Phi_n(R). \tag{2.9}$$

Die Lösungen von Gleichung (2.9) sind sogenannte *Energiehyperflächen*. Sie beschreiben die potentielle Energie in Abhängigkeit von den Positionen der Kerne. Die Berechnung von Energiehyperflächen und Wellenfunktionen mit quantenmechanischen Methoden stellt einen eigenen Wissenschaftszweig dar. Es gibt dabei verschiedene Ansätze, um den damit verbundenen sehr hohen Rechenaufwand in gewisser Weise zu reduzieren: Bei der Born-Oppenheimer-Approximation werden die Abstände der Protonen innerhalb eines Kerns fixiert, das heißt, Protonenbewegungen werden vernachlässigt. Bei der sogenannten *Hartree-Fock*-Methode (siehe zum Beispiel Roothaan (1951)) werden anstatt Interaktionen einzelner Elektronen Wechselwirkungen zwischen einem Elektron und Elektronenwolken betrachtet. Die *Møller-Plasset-(MP)*-Methode (Møller u. Plesset, 1934) hingegen sieht vor, daß die Elektronen einzeln miteinander interagieren. Unterschieden werden dabei verschiedene *Korrelationen*: Bei der MP(2)-Methode werden zweifache, bei der MP(3)-Methode dreifache und allgemein bei der MP(n)-Methode n-fache Elektronenkorrelationen berücksichtigt. MP-Methoden sind demzufolge im Vergleich zum Hartree-Fock-Verfahren äußerst rechenaufwendig, dafür jedoch auch viel genauer. Den bislang besten Kompromiß zwischen Rechenaufwand und Genauigkeit liefern sogenannte *Coupled-Cluster-Methoden* (Coester u. Kümmel, 1960). Dabei wird die Wellenfunktion mithilfe der Reihendarstellung einer Exponentialfunktion approximiert. Das erste nichtkonstante Glied der Reihe beschreibt Einfachanregungen, das zweite Doppelanregungen, das dritte Dreifachanregungen von Elektronen und so weiter. Je mehr Anregungen in die Darstellung miteinbezogen werden, desto genauer ist das Verfahren, allerdings können oftmals höhere Anregungen als Doppelanregungen vernachlässigt werden. Die exakte Lösung von Problemen im Bereich der Quantenmechanik ist hochgradig nichttrivial und nur von wenigen Autoren, wie zum Beispiel Baxter (1982), in Angriff genommen worden.

Bei Molekularen Simulationen, die auf der Klassischen Mechanik beruhen, wird nicht mehr zwischen Kernen und Elektronen unterschieden, sondern es werden ganze Atome als kleinste Teilchen betrachtet. Anstatt der Darstellung von Teilchen mithilfe von Wellenfunktionen, welche die Positionen der Teilchen charakterisieren, werden die Teilchen durch Bewegungsgleichungen beschrieben. In der Klassischen Mechanik wird eine andere Fundamentalgleichung betrachtet, und zwar die aus dem Jahr 1687 stammende Bewegungsgleichung von Isaac Newton (1642–1726). Die erste Publikation dieser Gleichung und der zugehörigen Newtonschen Gesetze ist in Newton (1726) zu finden. Nach dem zweiten Newtonschen Gesetz gilt:

$$F = m \cdot a, \tag{2.10}$$

also in Worten: *Kraft ist gleich Masse mal Beschleunigung.* Da die Beschleunigung die zweite Ableitung des Ortes nach der Zeit und die Kraft der negative Gradient der (ortsabhängigen) potentiellen Energie ist, gilt in der Klassischen Mechanik die *Newtonsche Bewegungsgleichung*:

$$-\nabla_i U_{\text{pot}}(r^N) = m_i \frac{d^2 r_i^N}{dt^2}(r^N), \;\; i = 1, ..., N. \tag{2.11}$$

Dabei umfaßt $r^N \in \mathbb{R}^{3N}$ die räumlichen Positionen aller $N \in \mathbb{N}^{>0}$ Teilchen des Systems. Der Index i bezieht sich auf ein bestimmtes Teilchen $i \in \{1, ..., N\}$. Demnach sind $-\nabla_i U(r) = f_i(r)$ die auf das Teilchen i wirkende Kraft, also der i-te Eintrag des negativen Potentialgradienten, und m_i dessen Masse. Anstelle des Hamilton-Operators $\hat{H}$ wird hier ein Funktional $U_{\text{tot}} : \mathbb{R}^{3N} \to \mathbb{R}$ betrachtet, und zwar eine Gesamtenergie der Form

$$U_{\text{tot}} = U_{\text{pot}} + U_{\text{kin}}. \tag{2.12}$$

Dabei ist U_{pot} die potentielle Energie und $U_{\text{kin}} := \frac{1}{2} \sum_{i=1}^{N} m_i v_i^2$ die kinetische Energie, abhängig von den Teilchengeschwindigkeiten $v_i, \;\; i = 1, ..., N$.

In der Klassischen Mechanik werden molekulare Systeme nicht immer atomar beschrieben. Es gibt auch Betrachtungsweisen auf sogenannten *Mesoskalen.* Der Übergang von atomistischen Skalen zu Mesoskalen wird als *Coarse Graining* bezeichnet, welches einen Vergröberungsprozeß beschreibt. Dabei werden mehrere Atome zu Kugeln (sogenannten *beads*) gruppiert. Dies verringert wiederum die Genauigkeit der Darstellung, jedoch auch die Komplexität und den damit verbundenen Rechenaufwand. Mesoskalen werden daher oft bei großen Molekülen wie Polymeren (Reith u. a., 2001, 2003; Fukunaga u. a., 2002; Faller, 2004) oder Proteinen (Tozzini, 2005; Clementi, 2008; Hills Jr. u. a., 2010) verwendet. Methoden, welche auf derartigen mesoskopischen Skalen angewandt werden, sind meist stochastischer Natur. Da das *Coarse Graining* in dieser Arbeit nicht im Vordergrund steht, wird es an dieser Stelle der Vollständigkeit halber nur kurz erwähnt.

In dieser Arbeit stehen Simulationen von Systemen der kondensierten Materie im Vordergrund. Insbesondere liegt hierbei der Fokus auf Flüssigkeiten, deren Systeme mittels mathematischer und physikalischer Methoden realitätsnah beschrieben werden können. Aus thermodynamischer Sicht sind Gase und Festkörper einfacher zu beschreiben als Flüssigkeiten. Deshalb sind Systeme kondensierter Materie von besonderem Interesse. Die ersten Flüssigkeitsmodelle gehen auf Morrell u. Hildebrand (1936) zurück, basierend auf der physikalischen Manipulation und Analyse

einer großen Anzahl Gelatinekugeln. Diese Modelle führten zu einer sehr guten dreidimensionalen Darstellung von Flüssigkeitsstrukturen und fanden beispielsweise in den experimentellen Studien von Bernal u. King (1968) Verwendung. Computersimulationen wurden erst Anfang der fünfziger Jahre durchgeführt: Das erste *in silico* durchgeführte Experiment, also die erste Molekulare Simulation, basierte auf dem von Metropolis u. Ulam (1949) eingeführten *Monte-Carlo-Verfahren* und wurde im Rahmen der experimentellen Studien von Metropolis u. a. (1953) veröffenticht.
Erst später wurde das Verfahren der Molekulardynamik entwickelt, welches die Newtonsche Bewegungsgleichung auf molekularer Ebene näherungsweise löst und zum Ziel hat, die dynamischen Eigenschaften eines Vielteilchensystems zu erhalten. Zunächst wurde dieses Verfahren nur für harte Kugeln eingeführt, und zwar in den Arbeiten von Alder u. Wainwright (1957) und Alder u. Wainwright (1959). Dabei bewegen sich die Teilchen mit konstanter Geschwindigkeit zwischen perfekt elastischen Kollisionen, was es ermöglicht, das dynamische Problem ohne Approximationen im Rahmen der Maschinengenauigkeit exakt zu lösen. Ein Jahr später wurde von Alder u. Wainwright (1960) eine Studie über eine kleine Anzahl an elastischen Kugeln veröffentlicht. Bei sogenannten *Lennard-Jones-Teilchen* ändert sich die Kraft stetig, wenn die Teilchen in Bewegung sind. Dies hat zur Folge, daß die Newtonsche Bewegungsgleichung numerisch, also zeitschrittweise, gelöst werden muß. Dies wurde als erstes von Rahman (1964) erfolgreich durchgeführt. Es folgten zahlreiche Forschungen im Bereich Simulationen von Lennard-Jones-Teilchen, beispielsweise von Verlet (1967), Verlet (1968) und Nicolas u. a. (1979). Die ersten vergröberten Modelle wurden in den 1970er Jahren entwickelt, zum Beispiel von Levitt (1976).

Für die Durchführung von Molekularen Simulationen, welche auf den Gesetzen der Klassischen Mechanik beruhen, sind also die folgenden Punkte von erheblicher Bedeutung:

- **Wahl der Granulariät:** Molekulare Simulationstechniken können sich, wie bereits erwähnt, sowohl der Quantenmechanik als auch der Klassischen Mechanik bedienen. Bei quantenmechanischen Berechnungen ist zu beachten, daß die zu lösenden Bewegungsgleichungen zu immensem Rechenaufwand führen. Die detaillierte Systemdarstellung, also die separate Betrachtung von Kernen und Elektronen, ist ein Grund dafür, daß die Schrödinger-Gleichung für komplexe Systeme praktisch nicht lösbar ist. Bei Berechnungen mit Mitteln der Klassischen Mechanik ist zu erwähnen, daß es sich hierbei lediglich um Approximationen des Systems handelt, ohne Berücksichtigung der Quantennatur. Dennoch sind die Ergebnisse oftmals ausreichend gut, um mit weitaus geringerem Rechenaufwand das System realitätsnah zu beschreiben. Gleiches gilt für mesoskalige Ansätze, die insbesondere bei großen Molekülen verwendet werden. Der Rechenaufwand Molekularer Simulationen bleibt trotz derartiger Ansätze immer noch sehr groß: Um ein sehr kleines System, bestehend aus beispielsweise 1000 kleinen Molekülen, nur eine Nanosekunde lang auf einem modernen Parallelrechner zu simulieren, werden in der Regel etwa zwei bis vier Stunden benötigt.

Sowohl in der Quantenmechanik als auch in der Klassischen Mechanik gibt es verschiedene Verfahren:

1. **Verfahren aus der Quantenmechanik:** Hierbei wird vor allem die Energiefläche als Lösung der Born-Oppenheimer-Approximation lokal minimiert. Daraus resultieren ebenfalls optimierte Geometrien.
2. **Verfahren aus der Klassischen Mechanik:** Hierbei gibt es zwei bekannte Kategorien von Simulationen. Die eine ist die sogenannte *Molekulardynamik (MD)*, bei der die Newtonsche Bewegungsgleichung gelöst wird. Die andere basiert auf zufälligen Veränderungen von Systemzuständen, deren Auswirkung auf die potentielle Energie gemessen wird. Der Phasenraum, also die Menge aller möglichen Zustände, wird dabei durch einen Markov-Prozeß abgetastet. Hierbei handelt es sich um sogenannte *Monte-Carlo-(MC-)Simulationen.*

Diese Verfahren können grob betrachtet mit den folgenden drei Granularitätsebenen in Verbindung gebracht werden:

1. **Quantenmechanische Darstellung:** Hierbei gehen sämtliche Kerne und Elektronen in die Beschreibung des Systems mit ein.
2. **Atomistische Darstellung:** Hierbei werden alle Atome des Systems einzeln betrachtet. Man spricht dabei von einem *All-Atom-Modell.* Eine Art von Vergröberung stellt das sogenannte *United-Atom-Modell* dar, welches zum Beispiel Wasserstoffatome nicht explizit berücksichtigt, sondern zum Beispiel gemeinsam mit dem daran gebundenen Kohlenstoffatom zu einer Methylgruppe zusammenfaßt.
3. **Mesoskalige Darstellung:** Der Begriff *Coarse Graining* steht für den Übergangsprozeß von der atomistischen zu einer vergröberten Darstellung. Vergröberung bezieht sich sowohl auf die Längen- als auch auf die Zeitskala.

Es sind auch Kombinationen dieser Simulationstechniken denkbar, die in den folgenden Kapiteln kurz angesprochen werden. Es ist zu beachten, daß MD-Simulationen auch für vergröberte Darstellungen und MC-Simulationen ebenfalls für atomistische Darstellungen anwendbar sind. Eine eindeutige Abgrenzung ist also nicht gegeben.

- **Wahl des Ensembles:** Von entscheidender Bedeutung ist bei Simulationen die Wahl des Ensembles. In der Quantenmechanik ist ein Ensemble lediglich als imaginäre Gesamtheit von Systemen zu verstehen, die aus einem oder mehreren Teilchen bestehen können. In der Klassischen Mechanik gibt es verschiedene Arten von Ensembles. Vier der wichtigsten Ensembles seien im folgenden definiert:
  Definition 2.1.1 (Ensembles)**.** *In der Statistischen Mechanik wird unter anderem zwischen den folgenden Arten von* Ensembles *unterschieden:*

  (i) Ein abgeschlossenes System mit konstanter Teilchenzahl N, konstantem Volumen V und konstanter innerer Energie E heißt NVE- oder mikrokanonisches *Ensemble.*

  (ii) Ein geschlossenes System mit konstanter Teilchenzahl N, konstantem Volumen V und konstanter Temperatur T heißt NVT- oder kanonisches *Ensemble. Ein kanonisches Ensemble wird auch* Gibbs-Ensemble *genannt.*

  (iii) Ein geschlossenes System mit konstanter Teilchenzahl N, konstantem Druck P und konstanter Temperatur T heißt NPT- oder isotherm-isobares *Ensemble.*

(iv) Ein offenes System mit konstantem chemischem Potential μ, konstantem Volumen V und konstanter Temperatur T heißt μVT- oder großkanonisches *Ensemble. Es wird auch* makrokanonisches *oder* superadditiv-kanonisches *Ensemble genannt.*

- **Anfangsbedingungen und räumliche Randbedingungen:** Partielle Differentialgleichungen sind nur dann eindeutig lösbar, wenn bestimmte Anfangs- und Randbedingungen gegeben sind. Die Lösung der Schrödinger-Gleichung bei vorgegebenen Randbedingungen erweist sich in der Praxis zumeist als äußerst schwierig, da $\hat{H}$ für komplexe Systeme gar nicht bekannt ist. Im Falle der Newtonschen Bewegungsgleichung ist eine Startkonfiguration des Systems, das heißt zum Zeitpunkt 0, zu definieren. Die Gleichung wird dann nach der Festlegung einer Zeitschrittweite numerisch gelöst. Außerdem sind räumliche periodische Randbedingungen hier von großer Bedeutung. Das bedeutet, daß das Gesamtsystem in kleine Boxen gleicher Kantenlänge eingeteilt wird, in denen die molekularen Bewegungen identisch sind. Näheres hierzu befindet sich in Anhang D.
- **Nebenbedingungen:** Falls man ein offenes System betrachtet, so ist der energetische Austausch zwischen System und Umgebung mitzuberücksichtigen. Ausschlaggebend sind hierbei insbesondere die Temperatur und der Druck, falls man diese konstant halten und somit an die Umgebung anpassen möchte. Das bedeutet, daß auch Nebenbedingungen eine große Rolle spielen. Gleiches gilt, wenn beispielsweise Bindungslängen oder -winkel während der Simulation konstant gehalten werden müssen.
- **Wahl des Potentials:** Wie oben bereits erwähnt, ist die Wahl des Potentials zur Beschreibung von Wechselwirkungen von der Komplexität des Systems abhängig. Verschiedene Formen von Potentialtermen werden in Abschnitt 2.2 näher erläutert.

2.2 Kraftfelder und Potentiale

In der Physik handelt es sich bei einem *Kraftfeld* um einen Bereich des Raums, in dem auf einen Körper eine ortsabhängige Kraft, ausgehend von einem oder mehreren anderen Körpern, wirkt. Mathematisch gesehen ist ein Kraftfeld ein Vektorfeld, dessen Schnittmenge mit einer Ebene durch Feldlinien darstellbar ist. Bei Molekularen Simulationen versteht man jedoch unter diesem Begriff eine Ansatzfunktion einschließlich der Parameter, die notwendig sind, um eine Kraft als negativen Ortsgradienten eines Potentials zu berechnen. Im folgenden wird der Begriff *Kraftfeld* nur in dieser letzten Definition verwendet.
Das Hauptziel dabei besteht darin, Parameter für Atomtypen zu bestimmen, um daraus Bindungslängen, Winkel, uneigentliche Torsionen und Diederwinkel innerhalb eines Moleküls sowie intermolekulare Wechselwirkungen zwischen Molekülen, die beispielsweise auf Dispersions- und elektrostatischen Kräften beruhen, zu beschreiben. Es wird sofort klar, daß die Anzahl dieser Atomtypen weitaus größer sein muß als die Anzahl an Elementen im Periodensystem, da es dabei vor allem auf die chemische Umgebung ankommt. Der Winkel zwischen Kohlenstoffatomen von Alkylgruppen und der zwischen Kohlenstoffatomen innerhalb eines Benzolrings beispielsweise unterscheidet sich signifikant.
Die einzelnen Komponenten eines Kraftfeldes werden in diesem Abschnitt im Detail vorgestellt. Abschnitt 2.2.1 behandelt zunächst intramolekulare Potentiale bezüglich Bindungen, Winkel, uneigentliche und eigentliche Diederwinkel. Intermolekulare Potentiale werden in Abschnitt

2.2.2 eingeführt: Das im Rahmen dieser Arbeit wichtigste intermolekulare Potential ist dabei das *Lennard-Jones-Potential*, welches kurzreichweitige Wechselwirkungen beschreibt. Langreichweitige Kräfte werden durch das *Coulomb-Potential* beschrieben, welches elektrostatische Wechselwirkungen darstellt und ebenfalls in Abschnitt 2.2.2 vorgestellt wird. Weiterhin gibt es Wechselwirkungen, die beispielsweise auf Dipol- und Quadrupolmomente zurückzuführen sind: In Abschnitt 2.2.3 wird die sogenannte *Multipolentwicklung* hergeleitet, aus der Dipol- und Quadrupolpotentiale leicht abzuleiten sind. Das Gesamtkraftfeld ergibt sich zum Schluß aus der Summe der einzelnen Kraftfeldkomponenten. Dies kann auf unterschiedlichste Art und Weise geschehen. Die drei wichtigsten Kraftfelder *Amber*, *Gromos* und *OPLS* werden in Abschnitt 2.2.4 vorgestellt.

2.2.1 Intramolekulare Potentiale

Dieser Abschnitt befaßt sich mit *intramolekularen* Wechselwirkungen. Diese sind vor allem geometrischer Natur und sind zum Beispiel auf Änderungen von Bindungslängen und Bindungswinkel zwischen Atomen innerhalb eines Moleküls zurückzuführen. Es werden sowohl Bindungswinkel zwischen drei Atomen als auch Diederwinkel zwischen vier Atomen betrachtet. Weiterhin werden auch Spezialfälle in Betracht gezogen. Es liegt zum Beispiel ein Sonderfall vor, wenn ein Atom sp^2-hybridisiert ist. Je nach Hybridisierung verändert sich die Geometrie innerhalb der chemischen Verbindung. Außer im Falle von Diederwinkeln werden intramolekulare Wechselwirkungen in vielen Fällen durch *harmonische* Potentiale $U(r)$ beschrieben, für die gilt:

$$\Delta U(r) = 0, \tag{2.13}$$

wobei Δ den Laplace-Operator bezeichnet.

Das Bindungslängenpotential wird wie folgt definiert:

Definition 2.2.1 (Bindungslängenpotential)**.** *Das* Bindungslängenpotential *mit Gleichgewichtsabstand r^0 ist für eine gegebene Bindungslänge r und Kraftkonstante k_r gegeben durch:*

$$U_b(r) := \frac{k_r}{2}(r - r^0)^2. \tag{2.14}$$

Analog wird das Potential für Bindungswinkel folgendermaßen definiert:

Definition 2.2.2 (Bindungswinkelpotential)**.** *Das* Bindungswinkelpotential *mit Gleichgewichtswinkel Φ^0 ist für einen gegebenen Bindungswinkel Φ und Kraftkonstante k_Φ gegeben durch*

$$U_a(\Phi) := \frac{k_\Phi}{2}(\Phi - \Phi^0)^2 \tag{2.15}$$

Bindungslänge und Bindungswinkel sind in Abbildung 2.1 dargestellt.

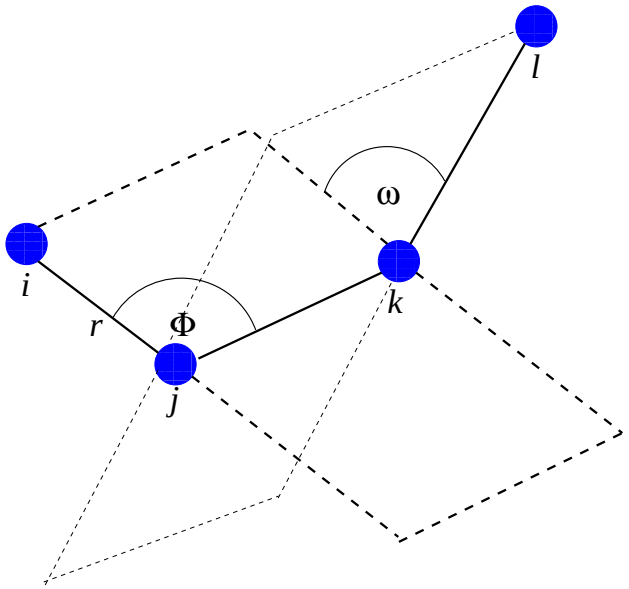


Abbildung 2.1: Bindungslänge r zwischen zwei Atomen i und j, Bindungswinkel Φ zwischen drei Atomen i, j und k sowie Diederwinkel ω zwischen vier Atomen i, j, k und l. Bei letzterem handelt es sich um den Winkel zwischen der Ebene, die von den Atomen i, j und k und der Ebene, die von den Atomen j, k und l aufgespannt wird.

Der Diederwinkel ω zwischen vier Atomen i, j, k und l ist ebenfalls in Abbildung 2.1 veranschaulicht: Es handelt sich dabei um den Winkel zwischen der Ebene, die von den Atomen i, j und k und der, die von den Atomen j, k und l aufgespannt wird. Derartige Winkel werden durch ein spezielles Potential beschrieben, welches von $\omega \in [0°, 360°]$ oder $\omega \in [-180°, 180°]$ abhängt. Da es sich um ein Rotationspotential handelt, muß es außerdem periodisch sein. Die chemische Motivation ist dabei die folgende: Im Falle von sp^3-hybridisierten endständigen C-Atomen von Kohlenwasserstoffen können die daran gebundenen H-Atome um das C-Atom gleichmäßig rotieren. Nach einer Rotation um 120° ist die Geometrie äquivalent zur ursprünglichen Geometrie. Bei sp^2-hybridisierten C-Atomen ist dies nach einer Rotation um 180° der Fall. Da die Rotationsbarriere im allgemeinen nicht sehr hoch ist, sind starke Abweichungen vom Minimum ω_0 möglich. Somit wäre eine Taylorentwicklung um ω_0, die nach dem ersten Glied abgeschnitten wird, zu ungenau, und Taylorentwicklungen höherer Ordnungen würden zu höherem Rechenaufwand führen. Aufgrund der Periodizität ist das Potential jedoch durch eine Fourierreihe beschreibbar:

Definition 2.2.3 (Diederwinkelpotential)**.** *Das* Diederwinkelpotential *mit Perioden* $\frac{2\pi}{n}$ *und Rotationskonstanten* $V_n,\ n = 1, ..., m,\ m \in \mathbb{N}$, *ist für einen Diederwinkel* ω *gegeben durch:*

$$U_d(\omega) := \sum_{n=1}^{m} V_n \cos(n\omega),\ m \in \mathbb{N}. \tag{2.16}$$

Die Rotationskonstanten $V_n,\ n = 1, ..., m$, beschreiben dabei die Größe der Rotationsbarriere um die Achse j-k.
Im allgemeinen gilt beim Diederwinkelpotential $n = ak,\ k \in \mathbb{N},\ a \leq 3$. Terme höherer Ordnung

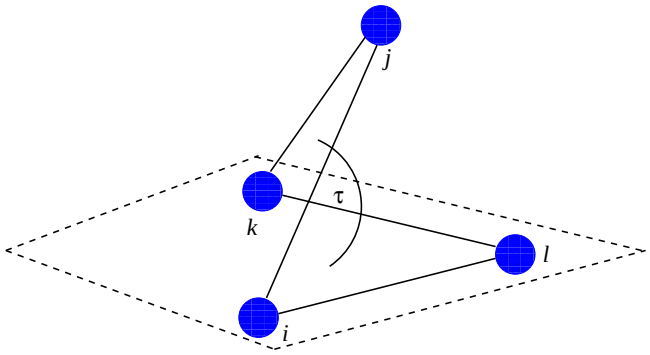


Abbildung 2.2: Uneigentlicher Diederwinkel τ im Falle eines sp^2-hybridisierten Atoms j. Das Atom j nennt man auch *out-of-plane*-Atom, da es aus der Ebene, die von den Atomen i, k und l aufgespannt wird, herausragt.

sind bei repulsiven und attraktiven van-der-Waals-Interaktionen notwendig, die beispielsweise bei oktaedrisch koordinierten Metallen und dinuklaren Komplexen auftreten.
Üblicherweise wird beim Diederwinkelpotential der Nullpunkt verschoben:

$$U_d(\omega) = \frac{1}{2}V_1\left(1+\cos(\omega)\right) + \frac{1}{2}V_2\left(1-\cos(2\omega)\right) + \frac{1}{2}V_3\left(1+\cos(3\omega)\right). \tag{2.17}$$

Dabei wird das Vorzeichen des Kosinusterms so gewählt, daß beim einfachen Rotationsterm ein Minimum bei $\omega = 180°$, beim zweifachen bei $\omega \in \{0°, 180°\}$ und beim dreifachen bei $\omega \in \{60°, 180°, 300°\}$ beziehungsweise $\omega \in \{-60°, 60°, 180°\}$ vorliegt. Durch den Faktor $\frac{1}{2}$ sind die Fourierkoeffizienten direkte Rotationsbarrieren. Dabei ist $V_n < 0$ nicht ausgeschlossen.

Ein Spezialfall ergibt sich, falls beispielsweise ein Atom j sp^2-hybridisiert ist und Bindungen zwischen Atomen i und l, k und l sowie i und j und k und j bestehen, wie in Abbildung 2.2 dargestellt. Da das Atom j aus der Ebene herausragt, die von i, j und k aufgespannt wird, spricht man von einem *out-of-plane*-Atom (oop), das zu einer Winkelverzerrung und zu hohen Kraftkonstanten führt. Daher führt man für derartige Spezialfälle einen separaten Potentialterm ein, der mit dem Begriff *uneigentliche Diederwinkel (englisch: improper dihedrals)* bezeichnet wird. Man betrachtet dabei den Diederwinkel τ zwischen den Atomen i, j, k und l, der in Abbildung 2.2 dargestellt ist, und betrachtet wieder dessen Differenz zum Gleichgewichtsdiederwinkel τ^0 mit einem harmonischen Potential:

Definition 2.2.4 (Potential für uneigentliche Diederwinkel)**.** *Das* Potential für uneigentliche Diederwinkel *mit Gleichgewichtsdiederwinkel τ^0 ist für einen gegebenen uneigentlichen Diederwinkel τ und Kraftkonstante k_τ gegeben durch:*

$$U_t(\tau) := \frac{k_\tau}{2}(\tau - \tau^0)^2. \tag{2.18}$$

Einige Simulationsprogramme verwenden auch Kombinationen aus den bisher eingeführten Potentialtermen. Derartige *Kopplungsterme* seien anhand des folgenden einfachen Beispiels motiviert:

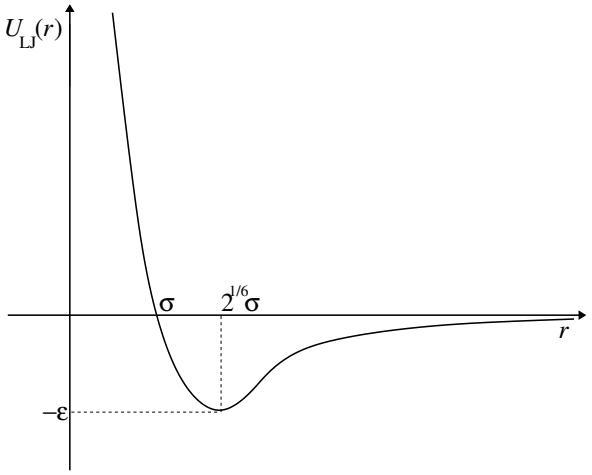


Abbildung 2.3: Das Lennard-Jones-(12,6)-Potential: Auf der x-Achse ist der Atomabstand r und auf der y-Achse das intermolekulare Potential $U_{LJ}(r)$ aufgetragen. Für $r \to \infty$ geht das Potential gegen 0, für $r \to 0$ gegen $+\infty$. Die Nullstelle liegt bei $r = \sigma$, das Minimum liegt bei $r = \sqrt[6]{2}\sigma$ und hat den Potentialwert $-\varepsilon$.

Ein Wassermolekül weist im thermodynamischen Gleichgewicht im Mittel einen HOH-Winkel von 104.5° und eine OH-Bindungslänge von 0.958 Å auf. Verkleinert man den Winkel durch äußere Einflüsse auf 90°, so erhöht sich die Bindungslänge auf 0.968 Å. Bei Vergrößerung des Winkels wird die Bindungslänge kleiner. Dieses Phänomen ist auf das *Elektronenpaarabstoßungsmodell* zurückzuführen, auch VSEPR-Modell (Gillespie u. Robinson, 2005) genannt, welches den räumlichen Aufbau eines Moleküls auf die abstoßenden Kräfte zwischen den Eletronenpaaren in der Valenzschale zurückführt. Es ist also durchaus sinnvoll, derartige synchrone Veränderungen der Potentialterme mittels Kombinationstermen zu berücksichtigen. Kombinationsterme sind charakterisiert durch eine multiplikative Abhängigkeit einzelner harmonischer Potentiale. Für weitere Details siehe zum Beispiel Jensen (1999).

2.2.2 Intermolekulare Potentiale

Vor der Einführung des in dieser Arbeit im Vordergrund stehenden intermolekularen Lennard-Jones-Potentials seien zunächst zwei grundlegende Definitionen gegeben:

Definition 2.2.5 (Lang- und kurzreichweitig)**.** *Sei $U \propto r^{-\ell}$ ein Potential, welches durch einen Abstandsterm r beschrieben wird, und $\ell \in \mathbb{N}^{>0}$. Die zugehörigen Interaktionen beziehungsweise Kräfte heißen* langreichweitig, *falls $\ell \leq \dim(\text{System}) - 1$. Ansonsten heißen sie* kurzreichweitig.

Langreichweitige Interaktionen stehen zum Beispiel für elektrostatische Kräfte und kurzreichweitige für Kräfte, die auf Dispersion und Repulsion zurückzuführen sind. Dispersionskräfte werden nicht durch permanente, sondern durch temporäre Polaritäten erzeugt.

Lennard-Jones-Potential Um kurzreichweitige Wechselwirkungen zwischen nicht chemisch gebundenen Atomen, die nicht geladen sind, zu beschreiben, also Interaktionen, die auf Dis-

persion und Repulsion zurückzuführen sind, gibt es verschiedene Alternativen. Ein sehr bekanntes Beispiel hierfür ist das sogenannte *Lennard-Jones-(LJ-)Potential*, das in Abbildung 2.3 graphisch dargestellt ist und auf Lennard-Jones (1931) zurückgeht: Wie deutlich zu erkennen ist, besteht es aus einem anziehenden und einem repulsiven Teil. Bei großen Abständen der Wechselwirkungszentren dominiert der anziehende Teil, der auch dafür sorgt, daß das Potential bei unendlicher Entfernung gleich 0. Je mehr sich Teilchen annähern, desto mehr Energie wird frei, was bedeutet, daß das Potential immer negativer wird. Das Potential erreicht im negativen Bereich ein Minimum und jenseits dieses Minimums eine Nullstelle, ab der der repulsive Anteil überwiegt, welcher auf die sogenannte *Pauli-Repulsion* zurückzuführen ist. Aufgrund letzterer stoßen sich Elektronen mit entgegengesetztem Spin und ansonsten gleichen Quantenzahlen bei einer Orbitalüberlagerung gegenseitig ab. Das Potential wächst somit an und geht für sehr kleine Abstände gegen unendlich.
Der anziehende Teil wird durch die sogenannte *London-Formel* beschrieben, bei der davon ausgegangen wird, daß die Anziehung durch einen negativen r^{-6}-Term dargestellt werden kann:

$$U_L(r) = -\frac{C_6}{r^6}. \tag{2.19}$$

Der Term, der hier als C_6 bezeichnet wird, ist oft schwer zugänglich und hängt beispielsweise von der Ionisierungsenergie ab. Insbesondere für Wasserstoffbrückenbindungen wird oftmals auch anstatt des r^{-6}-Terms ein r^{-10}-Term verwendet, um stärkere Anziehungskräfte zu beschreiben. Allerdings wurde in Ferguson u. Kollman (1991) gezeigt, daß bei geeigneter Wahl der Stauchungsparameter die resultierenden Potentiale Ergebnisse gleicher Güte liefern.
Der repulsive Teil wird durch einen positiven r^{-12}-Term ausgedrückt:

$$U_R(r) = +\frac{C_{12}}{r^{12}}. \tag{2.20}$$

Dieser Term ist nicht physikalisch, sondern rein numerisch motiviert, da $\left(r^{-6}\right)^2 = r^{-12}$. Die einfache numerische Handhabung des (12,6)-LJ-Potentials ist neben der Tatsache, daß es sich in der Praxis in äußerst vielen Fällen bewährt hat, ein weiterer Grund, daß es in dieser Arbeit zur Beschreibung von Dispersionswechselwirkungen verwendet wird.
Insgesamt entsteht aus den Gleichungen (2.19) und (2.20) das (12,6)-Lennard-Jones-Potential:

Definition 2.2.6 (Lennard-Jones-Potential)**.** *Das* (12,6)-Lennard-Jones-Potential *ist für einen Abstand r zwischen zwei Wechselwirkungszentren gegeben durch:*

$$U_{LJ}(r) := 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \ \sigma, \varepsilon \geq 0. \tag{2.21}$$

Das (12,6)-Lennard-Jones-Potential, das im folgenden mit *LJ-Potential* abgekürzt wird, hat die folgenden Eigenschaften:

- $U_{LJ}(\sigma) = 0$, das heißt, σ ist die Nullstelle des LJ-Potentials und kann als Interaktionsradius eines LJ-Wechselwirkungszentrums interpretiert werden.
- $U'_{LJ}(\sqrt[6]{2}\sigma) = 0 \wedge U''_{LJ}(\sqrt[6]{2}\sigma) > 0$, das heißt an der Stelle $r = \sqrt[6]{2}\sigma$ nimmt das Lennard-Jones-Potential ein Minimum an.
- $U_{LJ}(\sqrt[6]{2}\sigma) = -\varepsilon$, das heißt ε steht für die Tiefe des Potentialminimums.

Die Parameter ε_{ij} und σ_{ij} werden durch eine Kombinationsvorschrift aus den Parametern ε_i und σ_i, die sich auf ein Teilchen i beziehen, erhalten. Weit verbreitet sind die sogenannten *Lorentz-Bertelot-Formeln*, die auf Lorentz (1881) sowie Berthelot (1898) zurückgehen und für ε das geometrische sowie für σ das arithmetische Mittel als Kombination verwenden:

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j} \tag{2.22}$$

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j). \tag{2.23}$$

Es sind jedoch auch andere Kombinationen möglich. Auch für σ wird oft das geometrische Mittel verwendet. Dies ist abhängig vom zugrundegelegten Kraftfeld.
In dieser Arbeit wird in den meisten Fällen das (12,6)-Lennard-Jones-Potential für die Beschreibung von Interaktionen zwischen ungeladenen, nicht chemisch gebundenen Teilchen verwendet, da es sich als sehr gute Approximation der tatsächlichen intermolekularen Interaktionen erwiesen hat (siehe zum Beispiel Allen u. Tildesley (1987)). Außerdem ist es mathematisch gesehen leicht zu handhaben und benötigt keine numerisch aufwendigen Funktionsauswertungen. Ein anderes Potential, welches mithilfe eines exponentiellen Terms beschrieben wird und somit einen höheren Rechenaufwand benötigt, ist das sogenannte *Morse-Potential* (Morse, 1929). Der exponentielle Term sorgt dafür, daß auch Wechselwirkungen höherer Ordnung als Dipolwechselwirkungen berücksichtigt werden. Ein weiteres bekanntes intermolekulares Potential ist das sogenannte *Buckingham-Potential* (Buckingham, 1938). Die Motivation dieses Potentials besteht darin, daß die Abstoßung auf der Überlappung von Wellen beruht und die Elektronendichte proportional zu $\exp(-r)$ ist. Es unterscheidet sich vom Lennard-Jones-Potential nur bezüglich des Abstoßungsterms. Details zum Morse- und Buckingham-Potential sind zum Beispiel auch in Jensen (1999) zu finden. Der Vollständigkeit halber sei hier noch das sogenannte *FENE-Potential* erwähnt, welches jedoch vor allem im Bereich *Coarse Graining*, das heißt insbesondere für Polymersimulationen, eingesetzt wird und auf Kremer u. Grest (1990) zurückgeht.

Elektrostatisches Potential Um langreichweitige Interaktionen, die auf positiven und negativen Partialladungen $q_i,\ i = 1, ..., N$, von Atomen beruhen, zu beschreiben, wird das sogenannte *elektrostatische Potential*, auch *Coulomb-Potential* genannt, verwendet. Es handelt sich im Gegensatz zum Lennard-Jones-Potential um einen r^{-1}-Term.

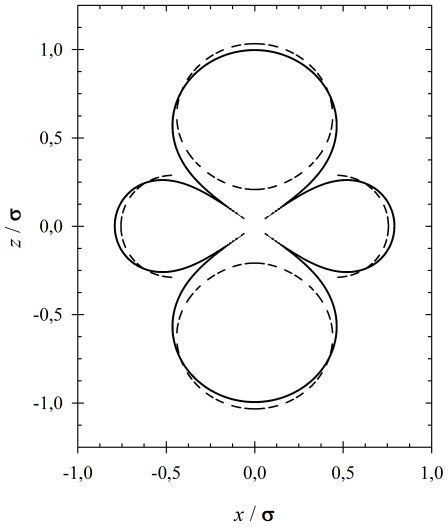


Abbildung 2.4: Projektionsdarstellung eines Quadrupolpotentials in der xz-Ebene: Das Potential ergibt sich als Überlagerung zweier Dipolpotentiale. Angegeben sind reduzierte Koordinaten, das heißt, die x- und die z-Koordinate wurden jeweils durch den LJ-Parameter σ dividiert. Das Bild ist, mit freundlicher Genehmigung, aus Engin u. a. (2011) entnommen.

Definition 2.2.7 (Elektrostatisches Potential)**.** *Das* Elektrostatische *oder* Coulomb-Potential *für zwei Atome mit Ladungen q_1 und q_2 ist für einen Abstand r gegeben durch:*

$$U_{El}(r) := \frac{q_1 q_2}{4\pi\varepsilon_0 r} \tag{2.24}$$

Dabei ist $\varepsilon_0 = 8.854 \cdot 10^{-12}\ \mathrm{Fm}^{-1}$ die Permittivität des Vakuums. Das elektrostatische Potential basiert auf dem Coulombschen Gesetz (Coulomb, 1788), ein bis heute immer wieder bestätigtes Naturgesetz für Punktladungen. Gemäß Definition 2.2.5 beschreibt das Coulomb-Potential langreichweitige Interaktionen.
Partialladungen, also Teilladungen innerhalb eines Moleküls, deren Summe dessen Gesamtladung ist, ändern sich während einer Simulation nicht. Daher muß deren Berechnung im Gleichgewichtszustand erfolgen und in den Kernzentren lokalisiert werden. Eine Methode zur Berechnung heißt **CH***arges from* **EL***ectrostatic* **P***otentials using a* **G***rid based method (CHELPG)* und geht auf Breneman u. Wiberg (1990) zurück. Sie basiert auf quantenmechanischen Ansätzen und ist zum Beispiel im Softwarepaket *Gaussian03* (Frisch u. a., 2004) enthalten. Weiterhin können Partialladungen quantenmechanisch mithilfe des Softwarepakets *GÅMESS* (Guest u. a., 2005) bestimmt werden.

2.2.3 Multipolentwicklung

Die *Multipolentwicklung* nähert lokalisierte Ladungsdichteverteilungen durch Punktladungen an (siehe zum Beispiel Buckingham (1959)). Die bekanntesten und praxisrelevanten Spezialfälle von Multipolen sind Monopol, Dipol und Quadrupol. Ein Monopol beschreibt das elektrische

Feld einer positiven oder negativen Punktladung, ein Dipol besteht aus zwei räumlich voneinander getrennten Punktladungen mit entgegengesetzten Vorzeichen und ein Quadrupol, dessen Potential in Abbildung 2.4 dargestellt ist, ergibt sich aus der Überlagerung von zwei Dipolen. Es bezeichne $\rho(\hat{r})$ die Ladungsdichteverteilung in Abhängigkeit vom dreidimensionalen Positionsvektor $\hat{r}$. Dann ist das allgemeine Potential für Multipole wie folgt definiert:

Definition 2.2.8 (Multipolpotential)**.** *Das* Multipolpotential *mit Ladungsdichteverteilung ρ ist für den dreidimensionalen Positionsvektor $\hat{r}$ eines Multipols gegeben durch:*

$$U_M(\hat{r}) := \frac{1}{4\pi\epsilon_0}\int d^3r' \frac{\rho(r')}{||\hat{r}-r'||}.$$

Durch Taylorentwicklung ergibt sich:

$$\begin{aligned}\frac{1}{||\hat{r}-r'||} &= \sum_{n=0}^{\infty}\frac{1}{n!}\left(-r'\nabla\right)^n\left(\frac{1}{||\hat{r}||}\right)\\ &= \frac{1}{||\hat{r}||} - \left\langle r', \nabla\left(\frac{1}{||\hat{r}||}\right)\right\rangle + \frac{1}{2}(r')^T\nabla\nabla\left(\frac{1}{||\hat{r}||}\right)r' + \mathcal{O}(r')^3.\end{aligned}$$

Es gilt:

$$\begin{aligned}\nabla\left(\frac{1}{||\hat{r}||}\right) &= -\frac{\hat{r}}{||\hat{r}||^3}\\ \nabla\nabla\left(\frac{1}{||\hat{r}||}\right) &= -\left(\hat{r}\left[\nabla\left(\frac{1}{||\hat{r}||^3}\right)\right]^T + \frac{1}{||\hat{r}||^3}I\right)\\ &= -\left(3\frac{-\hat{r}}{||\hat{r}||^5}\right)\hat{r}^T - \frac{1}{||\hat{r}||^3}I = 3\frac{1}{||\hat{r}||^5}\hat{r}\hat{r}^T - \frac{||\hat{r}||^2}{||\hat{r}||^5}I.\end{aligned}$$

Somit ergibt sich insgesamt:

$$\frac{1}{||\hat{r}-r'||} = \frac{1}{||\hat{r}||} + \frac{1}{||\hat{r}||^3}\langle\hat{r},r'\rangle + \frac{1}{2}(r')^T\left(3\frac{1}{||\hat{r}||^5}rr^T - \frac{||\hat{r}||^2}{||\hat{r}||^5}I\right)r' + \mathcal{O}(r')^3.$$

Für das Multipolpotential folgt dann:

$$\begin{aligned}U_M(\hat{r}) &= \frac{1}{4\pi\epsilon_0}[\underbrace{\frac{1}{||\hat{r}||}\int\rho(r')d^3r'}_{\text{Monopolpotential}} + \underbrace{\frac{1}{||\hat{r}||^3}\int\langle\hat{r},r'\rangle\rho(r')d^3r'}_{\text{Dipolpotential}}\\ &+ \underbrace{\frac{1}{2}\frac{1}{||\hat{r}||^5}\int(r')^T\left(3\hat{r}\hat{r}^T - ||\hat{r}||^2I\right)r'\rho(r')d^3r' + ...}_{\text{Quadrupolpotential}}].\end{aligned} \tag{2.25}$$

Der erste Term in Gleichung (2.25) entspricht gerade dem elektrostatischen Potential aus Gleichung (2.24), in diesem Abschnitt auch *Monopolpotential* genannt. Es beschreibt das Feld einer Punktladung, wobei diesbezüglich erwähnt sei, daß für große Abstände ein Feld von Ladungswolken genauso aussieht wie ein Feld von Punktladungen.

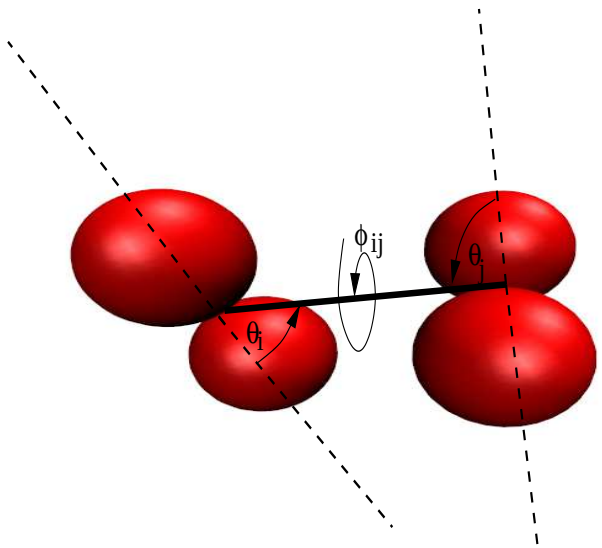


Abbildung 2.5: Orientierungswinkel bei zwei Molekülen i und j innerhalb eines Dipol- oder Quadrupolpotentials. Es gilt $\omega_{ij} = (\phi_{ij}.\theta_i)^T$.

Terme höherer Ordnung wie *Dipol-* und *Quadrupolpotential* sind Korrekturen für kleinere Abstände, bei denen die Approximation von Ladungswolken durch Punktladungen nicht mehr exakt genug ist.

Als nächstes wird die folgende Polarkoordinatentransformation betrachtet:

$$\frac{1}{||\hat{r} - r'||} = \frac{1}{\sqrt{||\hat{r}||^2 + ||\hat{r}'||^2 - 2\langle \hat{r}, r' \rangle \cos\phi_{ij}}} \tag{2.26}$$

$$= \frac{1}{||\hat{r}||} \frac{1}{\sqrt{1 + \left(\frac{||\hat{r}'||}{||\hat{r}||}\right)^2 - 2\frac{||\hat{r}'||}{||\hat{r}||}\cos\phi_{ij}}}. \tag{2.27}$$

Dabei ist ϕ_{ij} der Winkel zwischen den Molekülachsen (siehe Abbildung 2.5) zweier Moleküle i und j, in deren Richtungen zwei Dipolmomente μ^i und μ^j beziehungsweise zwei Quadrupolmomente Q^i und Q^j zeigen. Mit der Transformation $x := \frac{||\hat{r}'||}{||\hat{r}||}$ und $y := \cos\phi_{ij}$ folgt durch Taylorentwicklung:

$$\frac{1}{||\hat{r} - r'||} = \frac{1}{||\hat{r}||} \frac{1}{\sqrt{1 + x^2 - 2yx}} \tag{2.28}$$

$$= \frac{1}{||\hat{r}||} \left(1 + yx + \left(\frac{3}{2}y^2 - \frac{1}{2}\right)x^2 + \left(\frac{5}{2}y^3 - \frac{3}{2}y\right)x^3 + ...\right) \tag{2.29}$$

$$= \frac{1}{||\hat{r}||} \sum_{l=0}^{\infty} P_l(y) x^l, \tag{2.30}$$

wobei P_l die *Legendre-Polynome* vom Grad l sind. Nach Integration, Rücktransformation und

Einführung der Molekülwinkel θ_i und θ_j (siehe Abbildung 2.5) erhält man aus Gleichung (2.25) die folgenden Darstellungen für Dipol- und Quadrupolpotential:

Definition 2.2.9 (Dipolpotential). *Das* Dipolpotential *mit Dipolmomenten μ^i und μ^j sowie den Molekülwinkeln θ_i, θ_j und ϕ_{ij} ist für einen Dipolabstand r gegeben durch:*

$$U_D(r) := \frac{\mu^i \mu^j}{4\pi\epsilon_0 r^3} \left(\cos\phi_{ij} - 3\cos\theta_i \cos\theta_j\right) \tag{2.31}$$

Definition 2.2.10 (Quadrupolpotential). *Das* Quadrupolpotential *mit Quadrupolmomenten Q^i und Q^j sowie den Molekülwinkeln θ_i, θ_j und ϕ_{ij} ist für einen Quadrupolabstand r gegeben durch:*

$$\begin{aligned} U_Q(r) &:= \frac{3}{4}\frac{Q^i Q^j}{4\pi\epsilon_0 r^5}(1 - 5(\cos^2\theta_i + \cos^2\theta_j) && (2.32)\\ &- 15\cos^2\theta_i \cos^2\theta_j + 2(\sin\theta_i \sin\theta_j \cos\phi_{ij} - 4\cos\theta_i \cos\theta_j)) && (2.33) \end{aligned}$$

Dabei ist der Dipol- beziehungsweise Quadrupolabstand die Distanz der beiden windschiefen Molekülachsen. Dipole beziehungsweise Quadrupole werden oftmals in das Molekülzentrum plaziert. Dabei kann es sich sowohl um das geometrische als auch um das Massenzentrum handeln. Es ist zu beachten, daß Definition 2.2.10 nur für den Fall gilt, daß ein Quadrupol aus der Überlagerung zweier antiparalleler Dipole entsteht.

2.2.4 Beispiele für generalisierte Kraftfelder: Amber, Gromos und OPLS

Ein Kraftfeld setzt sich nun aus allen in den vorherigen Abschnitten eingeführten Potentialtermen zusammen. Die Kombination besteht dabei aus einer einfachen Aufsummierung der einzelnen Komponenten. Dabei werden die einzelnen Potentialterme als unabhängig vorausgesetzt. Kreuzabhängigkeiten können gegebenenfalls durch Kopplungsterme beschrieben werden. In diesem Abschnitt werden die drei wichtigsten Kraftfeldtypen vorgestellt.

Das sogenannte **A***ssisted-***M***odel-***B***uilding-and-***E***nergy-***R***efinement-(Amber-)Potential* hat die folgende Gestalt:

$$\begin{aligned} U_{\text{Amber}}(r^N) &:= \sum_{\text{Bindungen zw. } i,j} K_r^{ij}(r_{ij} - r_{ij}^0)^2 + \sum_{\text{Winkel zw. } i,j,k}^{N} K_\phi^{ijk}(\phi_{ijk} - \phi_{ijk}^0)^2 \\ &+ \sum_{\text{Diederwinkel } \omega} \sum_{n=1}^{m} \frac{V_n}{2}(1 + \cos(n\omega) - \gamma) \\ &+ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left\{ 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}. \end{aligned} \tag{2.34}$$

Dabei ist r^N ein $3N$-dimensionaler Positionsvektor. Es ist zu beachten, daß die Kraftkonstanten für Bindungen und Winkel K_r und K_ϕ anstatt $\frac{k_r}{2}$ und $\frac{k_\phi}{2}$ genommen werden. Das bedeutet

jedoch nicht, daß der Faktor $\frac{1}{2}$ weggelassen wird, er ist bereits in den Konstanten enthalten. Beim Diederwinkelterm wird die in Abschnitt 2.2.1 angesprochene Verschiebung des Nullpunkts vorgenommen. Zusätzlich verwendet man in den Kosinustermen einen Verschiebungsparameter γ. Das Amber-Potential wurde von Weiner u. a. (1984) entwickelt. Das zugehörige Kraftfeld wurde vor allem für Biomoleküle wie Proteine und Nukleinsäuren entwickelt.

Ein weiteres wichtiges Kraftfeld ist das sogenannte *Gromos-Kraftfeld*, welches auf Berendsen u. van Gunsteren (1987) zurückgeht und in der Software **GRO***ningen* **MA***chine for* **C***hemical* **S***imulations* (*Gromacs*, siehe auch Anhang F.1) implementiert ist. Das Gromos-Kraftfeld hat die folgende Gestalt:

$$\begin{aligned} U_{\text{Gromos}}(r^N) \;:=\; & \sum_{\text{Bindungen zw. } i,j} \frac{k_r}{2}(r_{ij}-r_{ij}^0)^2 + \sum_{\text{Winkel zw. } i,j,k}^{N} \frac{k_\phi}{2}(\phi_{ijk}-\phi_{ijk}^0)^2 \\ & + \sum_{\text{uneigentl. Diederwinkel zw. } i,j,k,l}^{N} \frac{k_\tau^{ijkl}}{2}(\tau_{ijkl}-\tau_{ijkl}^0)^2 \\ & + \sum_{\text{Diederwinkel } \omega} \sum_{n=1}^{m} \frac{V_\omega}{2}(1-\cos(n\omega-\omega_0)) \\ & + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \left\{ 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\epsilon_0}\left(\frac{1}{r_{ij}}\frac{\epsilon_{\text{RF}}-1}{2\epsilon_{\text{RF}}+1}\frac{r_{ij}^2}{r_c^3}\right)\right\}. \end{aligned} \tag{2.35}$$

Im Gromos-Potential ist auch ein Term für uneigentliche Diederwinkel enthalten. Beim eigentlichen Diederwinkelpotential ist zu beachten, daß die Kraftkonstante V_ω vom Diederwinkel und nicht von der Periode abhängig ist. Außerdem wird hier ein Gleichgewichtsdiederwinkel ω_0 angegeben. Für die intermolekularen Terme wird ein sogenannter sphärischer Abschneideradius r_C (C steht für *cutoff*) angegeben. Das bedeutet, daß das Potential ab einem bestimmten Wert gleich 0 gesetzt wird oder es bis auf den Wert 0 stetig fortführt. Dann wird das verschobene intermolekulare Potential

$$U_{\text{nb,verschoben}}(r_{ij}) := U_{\text{nb}}(r_{ij}) - U_{\text{nb}}(r_C)$$

verwendet, welches an der Stelle r_C stetig verschwindet. Dabei ist U_{nb} das ursprüngliche intermolekulare Potential. Näheres hierzu ist in Abschnitt 2.6.2 angegeben. Die Wirkung eines Reaktionsfelds mit elektrischer Feldkonstante ϵ_{RF} außerhalb des Abschneideradius r_C auf die Wechselwirkung eines Dipols mit seiner Umgebung wird durch den zusätzlichen Faktor in Gleichung (2.35) modelliert. Für $\epsilon_{\text{RF}} = 1$ ist dieser Term gleich 0. Bei einem Reaktionsfeld handelt es sich um ein elektrisches Feld, dessen Kräfte durch das elektrostatische Potential gegeben sind. Näheres zu Dielektrizitätskonstanten wird in Anhang C.2 besprochen.

Ein jüngeres Potential ist das **O***ptimized-***P***otentials-for-***L***iquid-***S***imulations-(OPLS-)Potential*, welches von Jorgensen u. a. (1996) entwickelt wurde und ebenfalls Einsatz im Softwarepaket

Gromacs findet. Es hat die folgende Form:

$$
\begin{aligned}
U_{\mathrm{OPLS}}(r^N) \quad := \quad & \sum_{\text{Bindungen zw. } i,j} K_r^{ij}(r_{ij} - r_{ij}^0)^2 + \sum_{\text{Winkel zw. } i,j,k}^{N} K_\phi^{ijk}(\phi_{ijk} - \phi_{ijk}^0)^2 \\
+ \quad & \sum_{\text{Diederwinkel } \omega} \frac{V_1}{2}(1+\cos(\omega)) + \frac{V_2}{2}(1-\cos(2\omega)) + \frac{V_3}{2}(1+\cos(3\omega)) \\
+ \quad & \frac{V_4}{2}(1-\cos(4\omega)) + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \left\{ 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\} f_{ij}
\end{aligned}
\tag{2.36}
$$

Es werden wie beim Amber-Potential die bereits halbierten Kraftkonstanten K_r und K_ϕ verwendet und beim Diederwinkelpotential wird explizit $m = 4$ angenommen, das heißt, Potentialterme geringerer Periodizität werden vernachlässigt. Der wesentliche Unterschied zu den beiden anderen Kraftfeldern ist der Skalierungsfaktor f_{ij}. Es ist zu beachten, daß intermolekulare Wechselwirkungen auch innerhalb eines Moleküls auftreten können, und zwar genau dann, wenn Atome durch drei oder mehr Bindungen voneinander getrennt sind. Da an drei Bindungen vier Atome beteiligt sind, spricht man dabei von 1-4-Interaktionen. In diesem Fall gilt $f_{ij} = 0.5$. Sind die Atome durch mehr als drei Bindungen voneinander getrennt, so gilt $f_{ij} = 1$.

Die meisten Kraftfelder unterscheiden sich nur unwesentlich bezüglich der Definition gewisser Potentialterme. Weitere wichtige Kraftfelder sind zum Beispiel *CHARMM* (Brooks u. a., 1983) und *COSMOS* (Möllhoff u. Sternberg, 2001). Im Falle polarisierbarer Kraftfelder wie *PFF* (Kaminski u. a., 2002) oder bei Mehrkörperpotentialen wie beispielsweise beim Tersoff- (Tersoff, 1988) oder Brenner-Potential (Brenner, 1990), welche nicht nur binäre intermolekulare Wechselwirkungen betrachten, kommen neue Potentialterme hinzu. Auch gibt es Kombinationsansätze zwischen Quantenmechanik und atomistischen Betrachtungen. Derartige Ansätze finden zum Beispiel bei großen biomolekularen Systeme Anwendung, wo bestimmte Systembereiche quantenmechanisch und andere nur atomistisch behandelt werden. Mithilfe des Amber-Potentials zur Bestimmung der LJ-Parameter wurde dies beispielsweise von Freindorf u. a. (2005) realisiert. Da die genannten Potentiale im Rahmen dieser Arbeit jedoch nicht verwendet werden, wird darauf nicht weiter eingegangen.

2.3 Molekulardynamik

Die Molekulardynamik (MD) ist neben der Monte-Carlo-Methode eine der beiden Hauptkategorien der Computersimulationen auf der Nanoskala und findet einen sehr breiten Einsatz in wichtigen wissenschaftlichen Anwendungen, siehe zum Beispiel Allen u. Tildesley (1987), van Gunsteren u. Berendsen (1990), Frenkel u. Smit (2006) und Jensen (1999).
Sämtliche Verfahren innerhalb einer MD-Simulation basieren auf der Lösung der zeitdiskretisierten Newtonschen Bewegungsgleichung (2.11). Dies bedeutet, Ortskoordinaten und Geschwindigkeiten in Abhängigkeit von der Zeit zu berechnen, das heißt, eine Trajektorie innerhalb des Phasenraums anzugeben. Dies geschieht mit numerischen Algorithmen, zum Beispiel mit den ursprünglich in der MD verwendeten *Prädiktor-Korrektor-Methoden*, die allgemein in Abschnitt

2.3.1 eingeführt werden. Eine weit verbreitete Integrationsmethode ist der *Verlet-Algorithmus*, der in Abschnitt 2.3.2 dargestellt wird.
MD-Simulationen sollten im allgemeinen stets die folgenden Anforderungen erfüllen:

1. Sie sollten den Energie- und Impulserhaltungssätzen genügen und zeitreversibel sein.
2. Sie sollten der realen Trajektorie so nahe wie möglich kommen.
3. Sie sollten auch für größere Zeitschrittweiten Δt verwendbar sein.
4. Sie sollten wenig rechenaufwendig sein und möglichst wenig Speicher verbrauchen.
5. Sie sollten möglichst einfach und mit wenig Aufwand zu implementieren sein.

Damit die Methode der MD diese Eigenschaften erfüllt, werden effiziente numerische Algorithmen angewandt, um die Bewegungsgleichung zu lösen.

2.3.1 Prädiktor-Korrektor-Verfahren

Bei der numerischen Lösung von Anfangswertproblemen im Zusammenhang mit einer gewöhnlichen zeitabhängigen Differentialgleichung werden stets Ort und Zeit in gewisser Weise diskretisiert und die Lösung zu einem Zeitpunkt t aus Funktionswerten an vorherigen Zeitpunkten vorhergesagt. Derartige *Finite-Differenzen-Methoden* gehen im allgemeinen auf eine Taylorentwicklung zurück, welche die Ordnung des numerischen Algorithmus festsetzt. Eine große Kategorie derartiger Verfahren stellen in der Numerik die sogenannten *Prädiktor-Korrektor-Methoden* dar. Dabei wird zunächst – meist durch einen heuristischen Ansatz – eine Vorhersage (p) für die Lösung des Problems ermittelt, welche dann durch einen Korrekturterm (Korrektor (c)) im darauffolgenden Schritt verbessert wird. Die hier vorgestellten Prädiktor-Korrektor-Verfahren gehen auf Gear (1966) und Gear (1971) zurück.
Es seien im folgenden $r(t)$ die Ortskoordinate zum Zeitpunkt t, $v(t)$ die Geschwindigkeit, $a(t)$ die Beschleunigung und $b(t) = \dot{a}(t)$ die Ableitung der Beschleunigung nach der Zeit. Der Prädiktionsschritt ergibt sich durch einfache Taylorentwicklung:

$$\begin{aligned} r^p(t+\Delta t) &= r(t) + \Delta t v(t) + \frac{1}{2}\Delta t^2 a(t) + \frac{1}{6}\Delta t^3 b(t) + \mathcal{O}(\Delta t^4) \\ v^p(t+\Delta t) &= v(t) + \Delta t a(t) + \frac{1}{2}\Delta t^2 b(t) + \mathcal{O}(\Delta t^3) \end{aligned} \tag{2.37}$$

Entsprechende Formeln gelten für $a^p(t+\Delta t)$ und $b^p(t+\Delta t)$. Als nächstes werden aus $r^p(t+\Delta t)$ die Kräfte zum neuen Zeitpunkt $t+\Delta t$ berechnet. Aufgrund der Newtonschen Bewegungsgleichung erhält man dadurch eine Korrekturbeschleunigung $a^c(t+\Delta t)$, die mit der Prädiktionsbeschleunigung $a^p(t+\Delta t)$ aus Gleichung (2.37) verglichen wird:

$$\Delta a(t+\Delta t) := a^c(t+\Delta t) - a^p(t+\Delta t). \tag{2.38}$$

Somit ergeben sich die folgenden Korrekturterme:

$$\begin{aligned} r^c(t+\Delta t) &= r^p(t+\Delta t) + c_0 \Delta a(t+\Delta t) \\ v^c(t+\Delta t) &= v^p(t+\Delta t) + c_1 \Delta a(t+\Delta t) \end{aligned} \tag{2.39}$$

Entsprechende Korrekturen ergeben sich für $a^p(t+\Delta t)$ und $b^p(t+\Delta t)$ mit entsprechenden Koeffizienten c_2 und c_3. Die Koeffizienten $c_0, ..., c_3 \in \mathbb{R}$ werden optimiert, um eine optimale Stabilität und Genauigkeit der Trajektorie zu erhalten. Die Koeffizientenoptimierung und der Konvergenzbeweis der Prädiktor-Korrektor-Methode sind in Gear (1966) und Gear (1971) zu finden. Der hier beschriebene Algorithmus läßt sich in vier Schritte zusammenfassen:

1. Prädiktion,
2. Berechnung der Kräfte und Beschleunigungen gemäß der Newtonschen Bewegungsgleichung (2.11),
3. Korrektur,
4. Aktualisierung der Ensemble-Mittelwerte, siehe Abschnitt 2.5.

Ein konkretes Beispiel für eine Prädiktor-Korrektor-Methode ist das sogenannte *Rahman-Verfahren*, welches auf Rahman (1964) zurückgeht. Der Prädiktionsschritt sieht bei diesem Verfahren wie folgt aus:

$$r^p(t+\Delta t) = r(t-\Delta t) + 2\Delta t v(t)$$

Aus den mit diesen Werten ermittelten Kräften erhält man $a(t+\Delta t)$. Daraus wird die Geschwindigkeit im nächsten Zeitschritt berechnet und anschließend der Korrektorterm r^c:

$$v(t+\Delta t) = v(t) + \frac{\Delta t}{2}\left(a(t+\Delta t) + a(t)\right) \tag{2.40}$$

$$r^c(t+\Delta t) = r(t) + \frac{\Delta t}{2}\left(v(t+\Delta t) + v(t)\right). \tag{2.41}$$

Dabei sollte die Beschleunigung $a(t+\Delta t)$ zwei- bis dreimal aktualisiert werden.
Das Rahman-Verfahren wurde bei den ersten in der Praxis durchgeführten MD-Simulationen verwendet.

2.3.2 Verlet-Algorithmus

Der auf Verlet (1967) und Verlet (1968) zurückgehende *Verlet-Algorithmus* und dessen Varianten sind die meistverwendeten numerischen Verfahren zur Lösung der Newtonschen Bewegungsgleichung im Rahmen von MD-Simulationen. Es wird wieder ein System aus N Teilchen betrachtet, wobei $r_i(t)$ für die Ortskoordinaten des Teilchens i zur Zeit t steht. Der Verlet-Algorithmus basiert auf einer Vorwärts- und Rückwärts-Taylorentwicklung von r_i:

$$r_i(t+\Delta t) = r_i(t) + \Delta t\frac{dr_i}{dt}(t) + \frac{1}{2}\Delta t^2\frac{d^2r_i}{dt^2}(t) + \frac{1}{6}\Delta t^3\frac{d^3r_i}{dt^3}(t) + \mathcal{O}(\Delta t^4) \tag{2.42}$$

$$r_i(t-\Delta t) = r_i(t) - \Delta t\frac{dr_i}{dt}(t) + \frac{1}{2}\Delta t^2\frac{d^2r_i}{dt^2}(t) - \frac{1}{6}\Delta t^3\frac{d^3r_i}{dt^3}(t) + \mathcal{O}(\Delta t^4). \tag{2.43}$$

Addiert man die Gleichungen (2.42) und (2.43) und setzt die Newtonsche Bewegungsgleichung ein, so ergibt sich die Rechenvorschrift für den klassischen Verlet-Algorithmus:

$$r_i(t+\Delta t) = 2r_i(t) - r_i(t-\Delta t) + \Delta t^2 \underbrace{\frac{d^2r_i}{dt^2}(t)}_{=-\frac{1}{m_i}\nabla U_i(t)} + \mathcal{O}(\Delta t^4). \tag{2.44}$$

Dabei ist m_i die Masse des Teilchens i und $U_i(t) := U(r_i(t))$ das zugrundegelegte (ortsabhängige) Potential.
Es ist zu beachten, daß im klassischen Verlet-Algorithmus keine Geschwindigkeiten auftauchen, da sie sich durch die Addition herauskürzen. Der Verlet-Algorithmus hat die Ordnung 4.
In einem zweiten Schritt lassen sich die Geschwindigkeiten bestimmen, allerdings nur im aktuellen Zeitschritt t: Wird Gleichung (2.43) von Gleichung (2.42) subtrahiert, so erhält man die zentralen Differenzen der Ordnung 2 für $v(t)$:

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2). \tag{2.45}$$

Da $r(t+\Delta t)$ und $r(t-\Delta t)$ dieselbe Rolle in den Gleichungen (2.44) und (2.45) spielen, ist der klassische Verlet-Algorithmus zeitreversibel.

Eine in vielen Simulationsprogrammen verwendete Variante des Verlet-Algorithmus ist der sogenannte *Bocksprung-Verlet* (englisch *Leap Frog Verlet*), welcher von Hockney (1970) und Potter (1972) entwickelt wurde. Dabei wird die Geschwindigkeit zum aktuellen Zeitpunkt t durch die Geschwindigkeit zum Zeitpunkt $t + \frac{\Delta t}{2}$ ersetzt, die mittels der zentralen Differenz

$$v\left(t + \frac{\Delta t}{2}\right) = v\left(t - \frac{\Delta t}{2}\right) + \Delta t a(t) \tag{2.46}$$

geschätzt wird. Man erhält dann als Rechenvorschrift:

$$r(t+\Delta t) = r(t) + \Delta t v\left(t + \frac{\Delta t}{2}\right) = r(t) + \Delta t \left(v\left(t - \frac{\Delta t}{2}\right) + \Delta t a(t)\right). \tag{2.47}$$

Die aktuelle Geschwindigkeit wird dann als Durschnittswert aus den beiden halben Zeitschritten berechnet:

$$v(t) = \frac{v\left(t + \frac{\Delta t}{2}\right) + v\left(t - \frac{\Delta t}{2}\right)}{2}. \tag{2.48}$$

Der Algorithmus heißt Bocksprung-Verlet, weil die Geschwindigkeiten den aktuellen Zeitpunkt t 'überspringen'. Sie gehen dabei von $t - \frac{\Delta t}{2}$ nach $t + \frac{\Delta t}{2}$ über.

Eine weitere Variante des Verlet-Algorithmus ist auf Swope u. a. (1982) zurückzuführen und befaßt sich nur mit den Geschwindigkeiten. Dieser sogenannte *Geschwindigkeits-Verlet* hat in verkürzter Form die folgende Darstellung:

$$v(t+\Delta t) = v(t) + \frac{\Delta t}{2}(a(t) + a(t+\Delta t)). \tag{2.49}$$

Die eigentliche Rechenvorschrift besteht aus zwei Teilschritten:

$$\begin{aligned} v\left(t + \frac{\Delta t}{2}\right) &= v(t) + \frac{\Delta t}{2} a(t) \\ v(t+\Delta t) &= v\left(t + \frac{\Delta t}{2}\right) + \frac{\Delta t}{2} a(t+\Delta t). \end{aligned}$$

Dabei ergibt sich die zweite Gleichung aus

$$v\left(t + \frac{\Delta t}{2}\right) = v(t+\Delta t) - \frac{\Delta t}{2} a(t+\Delta t),$$

wobei die Beschleunigung a zum Zeitpunkt $t + \Delta t$ gemäß dem klassischen Verlet-Algorithmus (2.44) ermittelt wird. Der Verlet-Algorithmus und dessen Varianten erfüllen die in Abschnitt 2 angegebenen Anforderungen an eine MD-Simulation (siehe zum Beispiel Allen u. Tildesley (1987)) und werden auch aufgrund ihrer geringen Komplexität in dieser Arbeit verwendet.

Das letzte hier vorgestellte Verfahren ist der von Beeman (1976) entwickelte *Beeman-Verlet*. Es handelt sich dabei um ein Verfahren höherer Ordnung, bei dem sich andere Koeffizienten für die Beschleunigungen ergeben, welche auf einen Koeffizientenvergleich basierend auf verschiedenen Taylorentwicklungen zurückzuführen sind. Dabei werden Werte zu vorherigen Zeitpunkten miteinbezogen. Dies ist in der Numerik eine übliche Methode, um Verfahren höherer Ordnung zu erhalten. Die Rechenvorschrift für Koordinaten und Geschwindigkeiten des Beeman-Algorithmus sieht wie folgt aus:

$$\begin{aligned} r(t+\Delta t) &= r(t) + \Delta t v(t) + \frac{2}{3}\Delta t^2 a(t) - \frac{1}{6}\Delta t^2 a(t-\Delta t) \\ v(t+\Delta t) &= v(t) + \frac{1}{3}\Delta t a(t+\Delta t) + \frac{5}{6}\Delta t a(t) - \frac{1}{6}\Delta t a(t-\Delta t) \end{aligned} \tag{2.50}$$

Allerdings hat die Erhöhung der Genauigkeit auch eine Erhöhung der Komplexität zur Folge. Daher gehört der Beeman-Verlet nicht zu den klassischen Methoden, die von den meisten Simulationsprogrammen verwendet werden.

2.4 Monte-Carlo-Simulationen

Zusammen mit Molekulardynamik sind *Monte-Carlo-Verfahren (MC-Verfahren)* die bekanntesten Methoden im Bereich Molekularer Simulationen. Monte-Carlo-Verfahren wurden von Metropolis u. Ulam (1949) entwickelt und basieren auf stochastischen Prozessen, also auf einer Folge von Zufallsereignissen. Die Hauptidee dabei ist, für sämtliche Teilchen Zufallsbewegungen zu definieren, welche mit einer gewissen Wahrscheinlichkeit akzeptiert oder verworfen werden, wodurch unwichtige Pfade durch den Phasenraum umgangen werden sollen, was zu einer Erhöhung der Effizienz führt. Ein wichtiger Bestandteil dabei ist die sogenannte *Stichprobenentnahme (Sampling)*, mit der der Phasenraum per Zufallsprinzip sukzessive abgetastet wird und aus dem resultierenden Zustand die Systemeigenschaften mittels finiter thermodynamischer Durchschnitte (siehe Abschnitt 2.5) berechnet werden. Im Gegensatz zu MD-Simulationen sind MC-Schritte keine Zeitschritte, sondern zufällig gewählte Zustände $q_i,\ i = 1, ..., M,\ M << 2^N$. Näheres zur Stichprobenentnahme befindet sich in Abschnitt 2.4.1. Ein Schema zur Berechnung sinnvoller Übergangswahrscheinlichkeiten von Zuständen ist das sogenannte *Metropolis-Schema*, welches sehr eng mit dem sogenannten *Importance Sampling* verbunden ist. Dieses wird in Abschnitt 2.4.2 eingeführt.

2.4.1 Stichprobenentnahme

Im folgenden werden Schnappschüße eines Systems als Zustände im Phasenraum betrachtet. Die Wahl dieser Zustände ist für die Effizienz von MC-Simulationen von ausschlaggebender Bedeutung und geschieht über ein sogenanntes *Sampling*, also eine Stichprobenentnahme durch

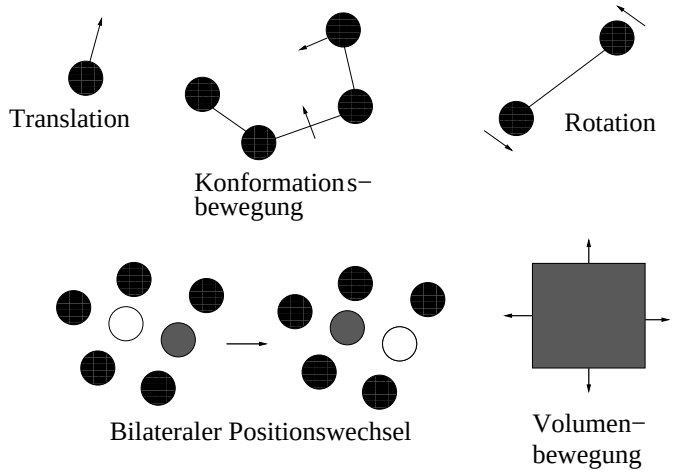


Abbildung 2.6: Typische symmetrische Monte-Carlo-Bewegungen: Translation, Konformationsbewegung, Rotation, bilateraler Positionswechsel und Volumenbewegung.

ein sukzessives, effizientes Abstasten des Phasenraums. Jede Stichprobenentnahme ist gegeben durch ein sogenanntes *Schema*. Die einfachste Möglichkeit ist das sogenannte *Simple Sampling*, gegeben durch ein simples Zufallsschema, nach dem statistisch unabhängige Zustände gemäß einem einfachen Zufallsprinzip gewählt werden. In den meisten Fällen leisten die derart gewählten Zustände jedoch nur einen kleinen Beitrag zum multidimensionalen Integral, wodurch Systemeigenschaften berechnet werden (siehe Abschnitt 2.5.1). Daher ist es sinnvoller, nur diejenigen Zustände zu betrachten, die einen signifikanten Beitrag leisten (sogenanntes *Importance Sampling*).
Stichprobenentnahmen basieren im allgemeinen auf *Markov-Ketten*, das heißt, die Wahrscheinlichkeit eines Zustandes hängt ausschließlich vom vorherigen Zustand ab. Es seien q_i, $i = 1, ..., M$, zufällig gewählte Zustände im Phasenraum. Dann werden Übergangswahrscheinlichkeiten $P(q_i \rightarrow q_j)$ zwischen zwei Zuständen q_i und q_j betrachtet, und es wird zwischen symmetrischen und asymmetrischen Versuchsbewegungen unterschieden. Die Wahrscheinlichkeit einer Versuchsbewegung wird mit α_{ij} bezeichnet. Die Wahrscheinlichkeit, diese Bewegung letztendlich zu akzeptieren, ist P_{ij}. Das *Importance Sampling* basiert auf symmetrischen Versuchsbewegungen ($\alpha_{ij} = \alpha_{ji}$), das heißt, die Wahrscheinlichkeit der Versuchsbewegung ist unabhängig von deren Richtung und die Umkehrung ist gleich wahrscheinlich. Typische MC-Bewegungen, welche auf derartigen Wahrscheinlichkeiten beruhen, sind in Abbildung 2.6 angegeben. Es handelt sich dabei um Translation, Konformationsbewegung, Rotation, bilaterale Positionswechsel und Volumenbewegung.
Das sogenannte *Biased Sampling* oder *Präferentielles Sampling* hingegen basiert auf asymmetrischen Versuchsbewegungen ($\alpha_{ij} \neq \alpha_{ji}$). Diese Methode wird insbesondere bei Systemen langer Kettenmoleküle, wie zum Beispiel Polymerschmelzen, verwendet (siehe auch Binder (1995)). Typische MC-Bewegungen sind dabei einzelne Monomerbewegungen und 'kriechende' Bewegungen der gesamten Kette (sogenannte *reptation*).
Die endgültige Berechnung der Versuchs- und Akzeptanzwahrscheinlichkeiten, die auf verschiedenen Arten von Stichprobenentnahmen basieren, variiert bei den unterschiedlichen MC-Verfahren.

2.4.2 Metropolis-Schema

Zur Konstruktion einer Trajektorie im Phasenraum innerhalb eines kanonischen Ensembles wird eine Übergangsmatrix Π benötigt, deren Einträge π_{ij} die bedingten Wahrscheinlichkeiten sind, vom Zustand q_i nach q_j zu wechseln. Ist $P(q_i)$ die *a-priori*-Wahrscheinlichkeit eines Zustands q_i, so gilt für die Übergangswahrscheinlichkeit:

$$P(q_i \to q_j) = P(q_i)\pi_{ij}.$$

Ist α_{ij} die Wahrscheinlichkeit einer Versuchsbewegung $q_i \to q_j$ und P_{ij} die Wahrscheinlichkeit, diese Bewegung zu akzeptieren, so gilt

$$\pi_{ij} = \alpha_{ij} P_{ij}. \tag{2.51}$$

Die Übergangsmatrix Π muß weiterhin die folgenden Eigenschaften erfüllen:

- Sie muß den Eigenwert 1 haben: $\exists_P\ \Pi P = P \Leftrightarrow \forall_i\ \sum_j \pi_{ij} P_j = P_i$.
- Sie muß stochastisch sein: $\forall_i\ \sum_j \pi_{ij} = 1$.

Lemma 2.4.1 (Mikroskopische Umkehrbarkeit)**.** *Ist die Bedingung der* mikroskopischen Umkehrbarkeit*,*

$$\forall_{ij}\ P_i \pi_{ij} = P_j \pi_{ji},$$

erfüllt, und ist Π *stochastisch, so hat* Π *den Eigenwert 1.*

Beweis. Es gilt:

$$\sum_i P_i \pi_{ij} = \sum_i P_j \pi_{ji} = P_j \underbrace{\sum_i \pi_{ji}}_{=1} = P_j.$$

□

Das erste zur Bestimmung einer Übergangsmatrix Π entwickelte Schema ist das *Metropolis-Schema* und geht auf Metropolis u. a. (1953) zurück. Die Akzeptanzwahrscheinlichkeiten P_{ij} aus Gleichung (2.51) werden dabei wie folgt bestimmt:

$$P_{ij} = \begin{cases} 1, & P_j \geq P_i \\ \frac{P_j}{P_i}, & P_j < P_i \end{cases} \tag{2.52}$$

Dabei gilt stets $i \neq j$. Es soll jedoch auch möglich sein, den aktuellen Zustand beizubehalten:

$$\pi_{ii} = 1 - \sum_{j \neq i} \pi_{ij}. \tag{2.53}$$

Das wichtigste Resultat aus dem Metropolis-Schema ist die Tatsache, daß α eine symmetrische Matrix ist:

Lemma 2.4.2. *Gilt die mikroskopische Umkehrbarkeit, so ist α ist symmetrisch.*

Beweis. Aus der mikroskopischen Umkehrbarkeit aus Lemma 2.4.1 folgt:

$$\forall_{ij}\pi_{ij} = \pi_{ji}\frac{P_j}{P_i}.$$

1. Fall: $(P_j \geq P_i)$
Dann ist $\pi_{ij} = \alpha_{ij}$, und es gilt:

$$\alpha_{ij} = \pi_{ji}\frac{P_j}{P_i} \overset{P_i<P_j,\ (2.52)}{=} \alpha_{ji}\frac{P_i}{P_j}\frac{P_j}{P_i} = \alpha_{ji}.$$

2. Fall: $(P_j < P_i)$
Dann ist $\pi_{ij} = \alpha_{ij}\frac{P_j}{P_i}$, und es gilt:

$$\alpha_{ij} = \pi_{ij}\frac{P_i}{P_j} = \pi_{ji}\frac{P_j}{P_i}\frac{P_i}{P_j} = \pi_{ji} \overset{P_j\leq P_i,\ 1.\ \text{Fall}}{=} \alpha_{ji}.$$

□

Die Umkehrung gilt ebenfalls:

Lemma 2.4.3. *Falls α symmetrisch ist, so gilt die mikroskopische Umkehrbarkeit.*

Beweis. Es wird wieder eine Fallunterscheidung vorgenommen:
1. Fall $(P_j \geq P_i)$. Dann gilt

$$\pi_{ij} = \alpha_{ij} = \alpha_{ji} \overset{\text{wie oben}}{=} \pi_{ji}\frac{P_j}{P_i},$$

woraus die mikroskopische Umkehrbarkeit folgt.
2. Fall $(P_j < P_i)$
Dann gilt

$$\pi_{ij} = \alpha_{ij}\frac{P_j}{P_i}.$$

Da α symmetrisch ist, gilt $\alpha_{ij} = \alpha_{ji} = \pi_{ji}$, woraus die mikroskopische Umkehrbarkeit folgt. □

Wird also im Metropolis-Schema die Matrix α als symmetrisch vorausgesetzt, so sind die gewünschten Eigenschaften für Π erfüllt. Mithilfe von Gleichung (2.53) läßt sich erreichen, daß Π stochastisch ist.
Die durch das Metropolis-Schema gegebene Stichprobenentnahme ist das *Importance Sampling*, bei dem die Zustände q_i so gewählt werden, daß sie mit zugehöriger Wahrscheinlichkeit $P(q_i)$ im Metropolis-Schema auftauchen. Die Markov-Kette $q = \{q_1, q_2, ..., q_i, q_j, ...\}$ enthält dann nur

die wichtigsten Zustände. Das Metropolis-Schema ist ein Beispiel, durch das das *Importance Sampling* gegeben ist. Wichtig ist, daß die mikroskopische Umkehrbarkeit erfüllt ist, also daß die Übergangsmatrix Π gemäß Lemma 2.4.1 den Eigenwert 1 hat. Wird nämlich der Phasenraum sukzessive nach wichtigen Zuständen abgetastet, so muß nach sukzessiver Anwendung der Übergangsmatrix wieder die gesamte Verteilung $P = P_\infty := P(q)$ aller Zustände aus q herauskommen:

$$P_\infty = \lim_{\tau \to \infty} \Pi^\tau P_1, \tag{2.54}$$

wobei $P_1 = P(q_1)$ die *a-priori*-Wahrscheinlichkeit des Anfangszustands q_1 ist.

Das zum Metropolis-Schema zugehörige Standard-MC-Verfahren läßt sich wie folgt beschreiben: Seien U_i und U_j die den Zuständen q_i und q_j entsprechenden potentiellen Energien. Dann gilt für konservative Systeme:

$$\frac{P(q_i)}{P(q_j)} = \exp\left(-\frac{U_j - U_i}{k_B T}\right). \tag{2.55}$$

Die Idee dabei besteht darin, Zustände mit geringer potentieller Energie zu bevorzugen. Für die Akzeptanzwahrscheinlichkeit P_{ij} gilt dann nach dem Metropolis-Schema (2.52):

$$P_{ij} = \min\left(1, \exp\left(-\frac{U_j - U_i}{k_B T}\right)\right). \tag{2.56}$$

Im Falle $U_j > U_i$ gilt $P_{ij} = \exp\left(-\frac{U_j - U_i}{k_B T}\right)$ und $P_{ji} = 1$. Das bedeutet, daß im MC-Algorithmus eine gleichverteilte Zufallszahl $R \in [0, 1]$ berechnet wird, und falls $R \leq \exp\left(-\frac{U_j - U_i}{k_B T}\right)$, so wird die entsprechende Versuchsbewegung akzeptiert. Andernfalls, das heißt für $U_j \leq U_i$, wird die Versuchsbewegung in jedem Fall akzeptiert. Falls eine Versuchsbewegung verworfen wird, so wird die alte Konfiguration erneut verwendet.

2.5 Berechnung von Systemeigenschaften

In den folgenden Abschnitten wird darauf eingegangen, wie aus Molekularen Simulationen physikalische Eigenschaften ermittelt werden können. Dies geschieht über sogenannte thermodynamische Durchschnitte, die in Abschnitt 2.5.1 eingeführt werden. In Abschnitt 2.5.2 wird auf die Berechnung der freien Energie und des chemischen Potentials eingegangen, was für die in dieser Arbeit äußerst wichtigen Simulationen im Phasengleichgewicht zwischen flüssiger und gasförmiger Phase von entscheidender Bedeutung ist.

2.5.1 Ensemble-Mittelwerte

Ensemble-Durchschnitte stammen aus der Statistischen Mechanik und sind statistische Mittelungen im Phasenraum. Hergeleitet werden sie mittels *generalisierter Koordinaten*, welche im Zusammenhang mit dem Hamilton-Formalismus stehen und der Beschreibung verallgemeinerter Bewegungsgleichungen dienen. Es seien im folgenden q^N generalisierte Koordinaten und p^N

generalisierte Momente. Der sogenannte *Ensemble-Durchschnitt* einer Eigenschaft A ist ein normierter Erwartungswert, welcher über Wahrscheinlichkeiten von Zuständen (q^N, p^N) berechnet wird:

Definition 2.5.1 (Ensemble-Durchschnitt (kanonisch)). *Der* Ensemble-Durchschnitt einer Eigenschaft A in einem kanonischen Ensemble *ist gegeben durch*

$$\begin{aligned}\langle A\rangle_{\mathrm{NVT}} &:= \frac{\int A(q^N,p^N)\exp\left(-H(q^N,p^N)/k_BT\right)dq^N dp^N}{\int \exp\left(-H(q^N,p^N)/k_BT\right)dq^N dp^N}\\ &= \frac{1}{Z}\int A(q^N,p^N)\exp\left(-H(q^N,p^N)/k_BT\right)dq^N dp^N.\end{aligned}$$

Dabei ist $Z := \int \exp\left(-H(q^N,p^N)/k_BT\right)dq^N dp^N$ *ein Normalisierungsterm, welcher der Gesamtheit aller Zustände entspricht.*

Es ist zu beachten, daß sich die Definitionen in anderen Ensembles von Definition 2.5.1 unterscheiden (vergleiche hierzu Anhänge A.3 und A.4).
Der Hamilton-Operator H setzt sich zusammen aus potentieller und kinetischer Energie:

$$H = U(q^N) + K(p^N).$$

Aufgrund dieser Separation in q^N und p^N kürzt sich bei konservativen Systemen, das heißt bei Systemen, in denen die Energie erhalten wird, der Integralterm, der sich auf die kinetische Energie bezieht, heraus, denn für diesen gilt

$$\int \exp\left(-\frac{K}{k_BT}\right)dp^N = \frac{\left(\frac{2\pi m k_BT}{h^2}\right)^{\frac{3}{2}}V}{N!} \equiv const.$$

Außerdem gilt $A(q^N,p^N) = A(q^N)$. Es folgt daher

$$\langle A\rangle_{NVT} := \frac{\int A(q^N)\exp\left(-U(q^N)/k_BT\right)dq^N}{\int \exp\left(-U(q^N)/k_BT\right)dq^N}.$$

Der *normalisierte Boltzmann-Faktor* $\frac{\exp\left(-U(q^N)/k_BT\right)}{\int \exp(-U(q^N)/k_BT)dq^N}$ ist die Wahrscheinlichkeitsdichte für q^N. Der Ensemble-Durchschnitt kann somit mit der Wahrscheinlichkeit des Zustands q^N ausgedrückt werden: Es gilt

$$\langle A\rangle_{NVT} = \int A(q^N)P(q^N)dq^N. \tag{2.57}$$

Wird eine diskrete Anzahl von Punkten $(q_1^N, ..., q_M^N)$ mit $M << 2^N$ betrachtet, so läßt sich der Ensemble-Durchschnitt durch die folgende Summe approximieren:

$$\langle A\rangle_{NVT} = \frac{1}{M}\sum_{i=1}^{M} A(q_i^N)P(q_i^N). \tag{2.58}$$

Die Berechnung von $P(q_i^N)$ ist allerdings nur durch wiederholtes Abtasten des Phasenraums möglich, was zunächst zu einem sehr hohen Rechenaufwand führt, jedoch durch *Monte-Carlo-Verfahren* (siehe Abschnitt 2.4) effizient gelöst werden kann. Bei MD-Simulationen bedient man

sich der sogenannten *Ergodenhypothese*, welche besagt, daß der zeitliche Mittelwert gleich dem Ensemble-Mittelwert ist. Es gilt dann:

$$\langle A \rangle = \bar{A} := \lim_{t \to \infty} \frac{1}{t} \int_0^t A(t')dt' \approx \lim_{M \to \infty} \sum_{i=1}^{M} A_i, \tag{2.59}$$

wobei M in diesem Fall die endliche Anzahl an MD-Zeitschritten ist.
Das Verfahren zur Berechnung physikalischer Eigenschaften wird somit sehr einfach: Es werden die Eigenschaften zu jedem Zeitschritt t berechnet und abschließend der Durchschnitt über alle Zeitschritte gebildet. Man weiß jedoch *a priori* nicht, wie lange simuliert werden muß, um korrekte Werte zu erhalten. Die Berechnung von Eigenschaften als Durchschnitte über einen gewissen Zeitraum, der sogenannten *Produktionsphase*, kann erst dann erfolgen, wenn das System näherungsweise *äquilibriert* ist, sich also in einem Gleichgewichtszustand befindet. Dies läßt sich feststellen, wenn sich die berechneten Systemeigenschaften über einen langen Zeitraum hinweg nur unwesentlich verändert haben. Allerdings liegt die Schwierigkeit darin, die Begriffe 'langer Zeitraum' und 'unwesentliche Veränderung' genau zu spezifizieren. Letzteres gilt auch für die Länge der Produktionsphase. Näheres dazu wird in Abschnitt 3.5.4 angesprochen. Von entscheidender Bedeutung ist auch die Wahl der Anfangskonfiguration, welche sich nicht zu weit vom Äquilibrium entfernt befinden darf. Es sei noch bemerkt, daß die sogenannte *Lyapunov-Instabilität* besagt, daß kleine Unterschiede in den Anfangswerten zu exponentiell divergenten Trajektorien führen können.

Formeln zur Berechnung der in dieser Arbeit relevanten physikalischen Eigenschaften sind in Anhang C angegeben.

2.5.2 Freie Energie und chemisches Potential

Dieser Abschnitt behandelt die Berechnung entropsicher Größen, die sich in der Praxis vor allem bei großen Systemen oftmals als schwierig erweist. Die freie Energie ist die zur Generierung eines Systems im thermischen Gleichgewicht mit seiner Umgebung bei gegebener Temperatur T benötigte Energie.
Für die freie Helmholtz-Energie F gilt:

$$F = U - T \cdot S, \tag{2.60}$$

und für die freie Gibbs-Energie G gilt:

$$G = H - T \cdot S. \tag{2.61}$$

Dabei ist $H = U + PV$ die Enthalpie, die sich aus innerer Energie U und Volumenarbeit PV zusammensetzt, und S die Entropie des Systems. Die Entropie ist in einem NVE-Ensemble gegeben durch

$$S = k_B \ln(\Omega), \tag{2.62}$$

wobei Ω eine Verteilungsfunktion ist und für die Anzahl der Zustände steht, welche die Teilchen annehmen können. In der Praxis ist die Entropie und somit die freie Energie schwierig zu berechnen, da die Multiplizität Ω im allgemeinen nicht bekannt ist.

Chemiker interessieren sich oft für die Änderung an freier Energie ΔG. Diese gibt beispielsweise an, ob Reaktionen freiwillig ablaufen (exergonisch, $\Delta G < 0$) oder erst nach Aufwendung eines bestimmten Energiebetrags (endergonisch, $\Delta G > 0$). Nach der *Gibbs-Helmholtz-Gleichung* gilt analog zu Gleichung (2.61):

$$\Delta G = \Delta H - T\Delta S. \tag{2.63}$$

Bei Molekularen Simulationen ist die freie Energie über thermodynamische Durchschnitte von potentiellen Energien oder deren Änderungen zu ermitteln. Es sei im folgenden A allgemein die freie Energie, also entweder die Gibbs- oder die Helmholtz-Energie. Es gilt:

$$A(N,V,T) = -k_B T \ln Q(N,V,T), \tag{2.64}$$

wobei $Q(N,V,T) := \int\int \exp\left(-\frac{\mathcal{H}(p,r)}{k_B T}\right) dr^N dp^N$ eine kanonische Zustandssumme ist. Dabei beschreibt $\mathcal{H}$ im Falle der Helmholtz-Energie die innere Energie und im Falle der Gibbs-Energie die Enthalpie. Die freie Energie kann weiterhin folgendermaßen in zwei Terme aufgespalten werden:

$$A(N,V,T) = A_{\mathrm{id}}(N,V,T) + A_{\mathrm{ex}}(N,V,T).$$

Dabei setzt sich $A(N,V,T)$ aus einem idealen Gasterm A_{id} und einem Realanteil A_{ex} zusammen. Es kann stattdessen auch deren Änderung betrachtet werden:

$$\begin{aligned}\Delta A(N,V,T) &= A_1(N,V,T) - A_0(N,V,T)\\ &= A_{\mathrm{id}1}(N,V,T) + A_{\mathrm{ex}1}(N,V,T) - \left(A_{\mathrm{id}0}(N,V,T) + A_{\mathrm{ex}0}(N,V,T)\right).\end{aligned}$$

Es handelt sich dabei um die Differenz an freier Energie zweier Zustände 0 und 1 (ungestörtes und gestörtes System). Der ideale Gasterm G_{id} ist im allgemeinen bekannt.
Für die Funktion Q gilt:

$$\begin{aligned}Q(N,V,T) &= Q_{\mathrm{id}}(N,V,T)\cdot Q_{\mathrm{ex}}(N,V,T)\\ &= \int \exp\left(-\frac{U_{\mathrm{kin}}}{k_B T}\right) dp^N \cdot \frac{1}{V^N} \int \exp\left(-\frac{U_{\mathrm{pot}}}{k_B T}\right) dr^N,\end{aligned}$$

und somit folgt für $A_{\mathrm{ex}}(N,V,T)$:

$$\begin{aligned}A_{\mathrm{ex}}(N,V,T) &= -k_B T \ln Q_{\mathrm{ex}}(N,V,T)\\ &= -k_B T \ln\left(V^{-N}\left\langle \exp\left(-\frac{U_{\mathrm{pot}}}{k_B T}\right)\right\rangle\right).\end{aligned} \tag{2.65}$$

Analog ergibt sich für ΔA:

$$\begin{aligned}\Delta A &= A_{\mathrm{ex}1}(N,V,T) - A_{\mathrm{ex}0}(N,V,T) = -k_B T \ln \frac{Q_{\mathrm{ex}0}(N,V,T)}{Q_{\mathrm{ex}1}(N,V,T)}\\ &\overset{U_{\mathrm{pot}1}=U_{\mathrm{pot}0}+\Delta U_{\mathrm{pot}}}{=} -k_B T \ln\left\langle \exp\left(-\frac{\Delta U_{\mathrm{pot}}}{k_B T}\right)\right\rangle_{U0_{\mathrm{pot}}}.\end{aligned} \tag{2.66}$$

Dabei stehen $U_{\mathrm{pot}0}$ und $U_{\mathrm{pot}1}$ für die potentiellen Energien der Zustände 0 und 1. Aufgrund der künstlichen Einführung einer *Störung* in das System, welche es von Zustand 0 in Zustand 1 überführt, wird dieses Verfahren zur Bestimmung der freien Energie auch als *Thermodynamische Perturbation* bezeichnet.

Eine weitere Möglichkeit zur Bestimmung von ΔA besteht in der Verwendung des sogenannten *chemischen Potentials* und geht auf Widom (1963) zurück. Das chemische Potential $\mu = \mu_{\mathrm{id}} + \mu_{\mathrm{ex}}$ beschreibt die Möglichkeit einer chemischen Substanz, mit anderen Substanzen zu reagieren (chemische Reaktionen), Aggregatszustände zu wechseln (Phasenübergänge) und sich im Raum zu verteilen (Diffusion). Es ergibt sich aus der Veränderung der freien Energie in Abhängigkeit von der Teilchenzahl N bei konstantem Volumen und konstanter Temperatur:

$$\mu = \left(\frac{\partial A}{\partial N}\right)_{V,T}, \quad \mu_{\mathrm{id}} = \left(\frac{\partial A_{\mathrm{id}}}{\partial N}\right)_{V,T}, \quad \mu_{\mathrm{ex}} = \left(\frac{\partial A_{\mathrm{ex}}}{\partial N}\right)_{V,T}. \tag{2.67}$$

Dabei kann μ_{ex} durch einen Differenzenquotienten approximiert werden, indem in das System ein sogenanntes *Testteilchen* $(N+1)$ eingeführt wird:

$$\begin{aligned} \mu_{\mathrm{ex}} &\approx A_{\mathrm{ex}}(N+1,V,T) - A_{\mathrm{ex}}(N,V,T) \\ &= -k_B T \log \left(\frac{V^{-(N+1)} \int \exp\left(-\frac{U_{\mathrm{pot}}(r^N, r_{N+1})}{k_B T}\right) dr^N dr_{N+1}}{V^{-N} \int \exp\left(-\frac{U_{\mathrm{pot}}(r^N)}{k_B T}\right) dr^N} \right). \end{aligned} \tag{2.68}$$

Mit $U_{\mathrm{pot}}(r^N, r_{N+1}) = U_{\mathrm{pot}}(r^N) + \Delta U_{\mathrm{pot}}(r^N, r_{N+1})$ und $V = \int dr_{N+1}$ folgt aus (2.68):

$$\mu_{\mathrm{ex}} \approx -k_B T \log \left\langle \exp\left(-\frac{\Delta U_{\mathrm{pot}}(r^N, r_{N+1})}{k_B T}\right) \right\rangle_{V, r^N, r_{N+1}}. \tag{2.69}$$

Der Algorithmus zur Bestimmung von ΔA beziehungsweise μ_{ex} besteht nunmehr aus den folgenden Schritten:

1. Simulation eines Systems aus N Teilchen.
2. Einfügen eines Testteilchens $N+1$ in jede Konfigurations an zufälliger Position mit zufälliger Orientierung.
3. Berechnung der potentiellen Energie der Interaktion des Testteilchens mit allen N anderen:
$$\Delta U_{\mathrm{pot}}(r^N, r_{N+1}) = \sum_{i=1}^{N} U_{\mathrm{pot}}(r_{N+1}, r_i).$$
4. Berechnung des *Boltzmann-Faktors*
$$\left\langle \exp\left(-\frac{\Delta U_{\mathrm{pot}}(r^N, r_{N+1})}{k_B T}\right) \right\rangle_{V, r^N, r_{N+1}}.$$

Der Nachteil der Methode nach Widom (1963) liegt in der Tatsache, daß sie bei hohen Dichten kaum anwendbar ist, da durch das zufällige Einsetzen eines Testteilchens die Wahrscheinlichkeit der Überlappung mit zunehmender Dichte steigt. Somit trägt nur eine geringe Anzahl an Testteilchen zum thermodynamischen Durchschnitt in Gleichung (2.69) bei, was zu äußerst schlechten Statistiken führt.

Ein etwas aufwendigeres Verfahren, das jedoch auch auf hohe Dichten anwendbar ist, ist die

auf Nezbeda u. Kolafa (1991) zurückzuführende *Methode der graduellen Einsetzung*. Sie ist zunächst nur für NVT-Ensembles verwendbar. Die Hauptidee besteht darin, ein fluktuierendes Teilchen einzuführen, welches in verschiedenen Kopplungszuständen mit allen anderen Teilchen auftreten kann. Es werden dabei k NPT-Subensembles $[N+\psi_j],\ j=0,...,k$, eingeführt, welche von einer kompletten Entkopplung des Teilchens ($[N+\psi_0]=[N]$) bis zu einer Gleichwertigkeit mit allen anderen Teilchen ($[N+\psi_k]=[N+1]$) übergehen. Dabei bezeichnen $\psi_j,\ j=0,...,k$, die Interaktionsenergien des fluktuierenden Teilchens mit allen anderen. Es gilt $\psi_0=0$. Weiterhin wird jeder Zustand j mit einem Gewicht w_j versehen.
Graduelle Einsetzung wird in dieser Arbeit nur in Verbindung mit MC eingesetzt. Sämtliche MC-Bewegungen (siehe zum Beispiel Abbildung 2.6) sind auch für das fluktierende Teilchen möglich. Hinzu kommt der Übergang von Zustand q_i nach Zustand q_j $(i,j\in\{0,...,k\},)$ dessen Wahrscheinlichkeit durch

$$P_{\text{flukt}}(q_i\to q_j):=\min\left(1,\frac{w_j\alpha_{ji}}{w_i\alpha_{ij}}\exp\left(-\beta_T(\psi_j-\psi_i)\right)\right) \tag{2.70}$$

gegeben ist. Dabei sind α_{ij} die Versuchswahrscheinlichkeiten im Metropolis-Schema (siehe Abschnitt 2.4.2). Außerdem gilt $\beta_T:=(k_BT)^{-1}$. Das chemische Potential in einem NVT-Ensemble berechnet sich dann gemäß

$$\mu=\mu_{\text{id}}(T)+\beta_T^{-1}\ln\left(\frac{N}{V}\left\langle\frac{w_k}{w_0}\frac{P(N)}{P(N+1)}\right\rangle\right), \tag{2.71}$$

wobei $P(N)$ beziehungsweise $P(N+1)$ die Wahrscheinlichkeiten sind, ein Ensemble mit N beziehungsweise $N+1$ Teilchen zu beobachten. Weiterhin ist $\mu_{\text{id}}(T)$ der temperaturabhängige ideale Anteil des chemischen Potentials, welcher wie G_{id} im allgemeinen bekannt ist. Zur Erhöhung der Effizienz wird die Methode der graduellen Einsetzung meist zusammen mit präferentiellem Sampling verwendet. Die mikroskopische Umkehrbarkeit aus Lemma 2.4.1 muß auch für das fluktuierende Teilchen gewährleistet sein.
In einem NPT-Ensemble sind zusätzlich zu den Teilchenbewegungen auch Volumenbewegungen zu berücksichtigen. Die Wahrscheinlichkeit einer Volumenbewegung $V_m\to V_n$ ist gegeben durch

$$P(V_m\to V_n):=\min\left(1,\left(\frac{V_n}{V_m}\right)^N\exp\left(-\beta_T(P(V_n-V_m)+U_n-U_m)\right)\right). \tag{2.72}$$

Dabei steht U_n für die neue und U_m für die alte potentielle Energie des Systems. Das Volumen V in Gleichung 2.71 muß nun in den Ensembledurchschnitt miteinbezogen werden, und es ergibt sich die folgende bislang nur numerisch bewiesene Gleichung für das chemische Potential in einem NPT-Ensemble:

$$\mu=\mu_{\text{id}}(T)+\beta_T^{-1}\ln\left\langle\frac{N}{V}\frac{w_k}{w_0}\frac{P(N)}{P(N+1)}\right\rangle. \tag{2.73}$$

In Vrabec u. a. (2002) wurde die Methode der graduellen Einsetzung auf quadrupolare Zweizentren-Lennard-Jones-Teilchen angewandt und mit dem Verfahren von Widom (1963) verglichen. Graduelle Einsetzung ist numerisch deutlich aufwendiger als die Widom-Methode, führt jedoch zu besseren Statistiken und ist auch bei hohen Dichten anwendbar. In einem NPT-Ensemble sind die Ergebnisse der beiden Methoden sehr ähnlich, was als numerischer Beweis für Gleichung (2.73) angesehen werden kann.

2.6 Ausgewählte praktische Aspekte

In diesem Abschnitt wird erläutert, wie die in den Abschnitten 2.2 bis 2.5 eingeführten Theorien in der Praxis angewandt werden. Es ist keineswegs trivial, aus den theoretischen Überlegungen bezüglich Simulationen effiziente Programme zu entwickeln. Daher beschäftigt sich dieser Abschnitt mit praktischen Aspekten. Zunächst muß eine initiale Konfiguration für das System festgelegt werden. Diese sowie der anschließend durchzuführende Simulationslauf und die damit verbundenen praktischen Problematiken werden in Abschnitt 2.6.1 kurz erläutert.
Die in Abschnitt 2.2 eingeführten Kräfte als negative Gradienten der entsprechenden Potentiale sollten möglichst effektiv berechnet werden, denn sie benötigen zunächst sämtliche Teilcheninteraktionen, was ein $\mathcal{O}(N^2)$-Problem ist. Wie die Komplexität durch numerische und physikalische Überlegungen reduziert werden kann, wird in Abschnitt 2.6.2 kurz beschrieben.
Weitere Details bezüglich der technischen Umsetzung bei Molekularen Simulationen befindet sich in Anhang D.

2.6.1 Von der initialen Konfiguration bis zur Berechnung von Systemeigenschaften

Im diesem Abschnitt geht es um die Struktur eines Computerprogramms für Molekulare Simulationen. Dabei ist es zunächst notwendig, eine sinnvolle initiale Konfiguration festzulegen. Diese beinhaltet die Wahl des Ensembles, die Definition der Anfangs- und Randbedingungen für die Teilchenkoordinaten und, im Falle von MD, Teilchengeschwindigkeiten sowie die Festlegung einer Systemtopologie. Letztere legt Atome, Atomgruppen, Bindungen und Kraftfeldparameter innerhalb eines Moleküls fest. Erst nach Festlegung einer derartigen Startkonfiguration kann eine Simulation ausgeführt werden. Diese besteht, wie in Abschnitt 2.3 erläutert, bei MD-Simulationen aus der numerischen, iterativen Lösung der zugrundeliegenden Bewegungsgleichung, bei der sämtliche Kräfte und Potentiale zu jedem Zeitschritt berechnet werden müssen und bei MC-Simulationen aus einer Stichprobenenthahme innerhalb des Phasenraums, was in Abschnitt 2.4 beschrieben wurde. Um den Phasenraum genauestens abzutasten, sind theoretisch unendlich viele Stichprobenentnahmen erforderlich. Um dennoch nach einer endlichen Simulationsdauer akkurate Systemeigenschaften ermitteln zu können, sind einige praktische Fragen zu berücksichtigen. Daher muß nach einem bestimmten Kriterium entschieden werden, wann die Simulation abgebrochen werden kann und verläßliche Werte für alle betrachteten Eigenschaften berechnet werden können. Hierzu wird eine Zweiteilung des Simulationsverlaufs vorgenommen, und zwar in *Äquilibrierung* und *Produktion*. Die Bestandteile der initialen Konfiguration und der eigentliche Simulationsverlauf werden in Anhang D im Detail diskutiert.

2.6.2 Technische Umsetzung der Kräfteberechnungen

Um in einem System, bestehend aus N Teilchen, die Kräfte, welche auf jedes Teilchen i, $i = 1, ..., N$, durch den negativen Gradienten des zugrundeliegenden Potentials zu bestimmen, sind einige Vorüberlegungen durchzuführen. Es ist keineswegs trivial, sämtliche Parameter und Konstanten für das entsprechende Kraftfeld zu ermitteln und die Kräfte effizient zu berechnen. Daher werden im folgenden einige Quellen für Kraftfeldinformationen angegeben und kurz dar-

gestellt, wie die Berechnung der Kräfte implementationstechnisch effizient umgesetzt werden kann. Details zu effizienten Kräfteberechnungen befinden sich in Anhang E.

Quellen für Kraftfeldinformationen Im folgenden sind die experimentellen und theoretischen Quellen für Kraftfeldparameter und -konstanten sowie physikalische Eigenschaften aufgeführt:

1. **Experimentell:**
   - *Molekularstruktur:* Spektroskopie, Kristallographie, Elektronenbeugung
   - *Kraftkonstanten:* Schwingungsspektroskopie, Raman-Spektroskopie
   - *Struktur von Flüssigkeiten:* Röntgen- und Neutronenbeugung
   - *Dynamische Eigenschaften:* NMR-Spektroskopie, Fluoreszenz
   - *Intermolekulare Interaktionen:* Kristallographie, Elektronen-, Röntgen- und Neutronenbeugung
2. **Theorie:**
   - *Bindungslängen und -winkel:* Semi-empirische Methoden, das heißt quantenmechanische Methoden, bei denen die Wechselwirkungen zwischen Elektronen approximiert oder vernachlässigt werden. Als Korrektur werden die entsprechenden Modellparameter so eingestellt, daß experimentelle Geometrien reproduziert werden.
   - *Diederwinkel, intermolekulare elektrostatische Konstanten und partielle Atomladungen: Ab-initio*-Methoden, das heißt quantenchemische Methoden ohne jegliche Form von empirischen Daten. Partialladungen können auch an experimentelle Zielgrößen angepaßt werden.

In Praprotnik u. a. (2008a) beispielsweise wurden bei Verwendung der CHARMM-Energiefunktion (Brooks u. a., 1983) die Atompositionen innerhalb eines Aluminiumphosphatkristalls aus röntgendiffraktometrischen Daten und alle anderen Kraftfeldparameter mithilfe von quantenmechanischen Methoden ermittelt. Beispiele, bei denen ein Teil der Parameter an experimentelle Geometrien und der andere Teil an quantenmechanische Daten angepaßt wurden, sind Foloppe u. MacKerell (2000) für biologische Substanzen wie DNA und RNA, Chen u. a. (2001b) für kleine Moleküle wie Amine und Fluorethane sowie Krieger u. a. (2002) für Proteine. Ein neues Kraftfeld für Proteine, welches von Liu u. a. (2011) entwickelt wurde, basiert auf der Einstellung der Torsionsenergieparameter. Das mathematische Problem, die Potentialhyperfläche zu minimieren, wurde dabei zu einem Regressionsproblem umformuliert, welches mit einer *Support Vector Machine* (siehe Abschnitt 6.2.2) gelöst wurde.

Nicht alle Parameter können jedoch auf derartige Art und Weise bestimmt werden. Vor allem Kraftfeldparameter für dispersive Interaktionen wie zum Beispiel die LJ-Parameter ε und σ sind dadurch nur schwierig zu bestimmen. Außerdem sind experimentelle Geometrien bei weitem nicht für jede Substanz bekannt. Daher bedient man sich in der Praxis empirischer Methoden. Hierzu sind lediglich experimentelle physikochemische Eigenschaften für bestimmte Temperaturen und Drücke erforderlich. Für einige wenige sehr kleine Moleküle hingegen gibt es sogar Zustandsgleichungen. Mithilfe dieser Gleichungen, welche auf experimentellen Messungen beruhen und mittels Regression erhalten worden sind, ist eine Vielzahl an physikalischen

Eigenschaften zu jeder Temperatur und zu jedem Druck über einen sehr großen Bereich hinweg berechenbar. Prominente Beispiele hierfür sind eine Zustandsgleichung für Kohlenstoffdioxid (Span u. a., 1996) und Stickstoff (Span u. a., 2000).

Berechnung der intramolekularen Wechselwirkungen Zur Berechnung der intramolekularen Wechselwirkungen sind Bindungslängen aus Atomabständen, Bindungs- und Diederwinkel zu bestimmen. Die Berechnung der Kräfte erfolgt über die Differentiation der jeweiligen Potentiale. Wie dies im einzelnen geschieht, ist in Anhang E.2 erläutert.

Effiziente Berechnung der intermolekularen Wechselwirkungen Die Komplexität der Berechnnung von intermolekularen Wechselwirkungen liegt in $\mathcal{O}(N^2)$. Sie ist der Hauptgrund für die hohe Komplexität Molekularer Simulationen. Bei der Berechnung der zugehörigen Potentialterme kann beispielsweise ein Abschneideradius $r_C > 0$ eingeführt werden, bei dem das Potential künstlich auf 0 gesetzt wird. Die Idee dabei besteht darin, daß die Wechselwirkungen zwischen weit voneinander entfernt liegenden Teilchen vernachlässigbar gering sind. Um eine Unstetigskeitsstelle zu vermeiden, kann das Potential nach oben verschoben oder stetig und differenzierbar, beispielsweise durch Splines, auf 0 gesetzt werden. Der Rechenaufwand kann dadurch auf $\mathcal{O}(N)$ reduziert werden. Diese Methode ist allerdings insbesondere bei langreichweitigen Interaktionen problematisch. Hier sind in den meisten Fällen Korrekturterme erforderlich, welche nachträglich aufaddiert werden müssen. Bei der Berechnung des elektrostatischen Potentials bedient man sich komplexerer Methoden wie beispielsweise der sogenannten *Ewald-Summation* oder der P^3M*-Methode*. Erstere kann den Rechenaufwand lediglich auf $\mathcal{O}(N^{\frac{3}{2}})$ und letztere auf $\mathcal{O}(N \log N)$ reduzieren. Details zu den genannten Methoden befinden sich ebenfalls in Anhang E.2.

3 Optimierung von Kraftfeldparametern für Molekulare Simulationen

Die Parametrisierung von Kraftfeldern ist als ein eigenständiges Forschungsfeld zu betrachten, welches erst in den letzten Jahren zwar von einzelnen Autoren behandelt worden, aber noch lange nicht als vollständig entwickelt zu sehen ist. Daher werden im Rahmen dieser Dissertation Optimierungsmethoden von Kraftfeldparametern in besonderem Maße ergründet und neue mathematisch-numerisch basierte Lösungswege vorgeschlagen. Optimierte Kraftfelder, welche die inter- und intramolekularen Wechselwirkungen zwischen Molekülen beschreiben, sind sowohl in wissenschaftlicher als auch in industrieller Hinsicht unentbehrlich. Denn sämtliche zu berechnenden physikalischen Eigenschaften sind vollkommen abhängig von der Wahl des zugrundegelegten Kraftfelds mit seinen zugehörigen Parametern. Weiterhin sind Molekulare Simulationen für Systeme beliebiger Größe und zu im Labor schwer zu realisierenden Temperaturen und Drücken nur dann akkurat durchführbar, wenn vorher die entsprechenden Kraftfelder sinnvoll eingestellt wurden.
Die enorme Wichtigkeit der Wahl eines sinnvollen und möglichst genauen Kraftfeldes ergibt sich aus der großen Breite des Anwendungsgebiets von Molekularen Simulationen. In der Literatur findet man eine Vielzahl von Anwendungsbereichen, in denen Kraftfelder eine wesentliche Rolle spielen. Insbesondere sind hierbei thermodynamische Eigenschaften von Fluiden (zum Beispiel Zhou u. Stell (1992), Siepmann u. a. (1993) und Kolafa u. a. (2001)), Verhalten von Fluiden in Membranen (zum Beispiel Singer u. Nicolson (1972), Ho u. Baumgärtner (1990) und Valiullin u. a. (2006)), mechanische Eigenschaften von Feststoffen (zum Beispiel Batra u. a. (1975), Fehlner (2000) und Della u. Dongwei (2008)), Phasenübergänge (zum Beispiel Lin u. a. (2003), Bien u. Chiriac (2004) und Vrabec u. Gross (2008)), Transportprozesse in der Biologie (zum Beispiel Hodgkin u. Huxley (1952) und Barkla u. Pantoja (1996)), Faltung von Proteinen (zum Beispiel Levitt u. Warshe I (1975), Gsponer u. Caflisch (2002) und Snow u. a. (2005)), Transportprozesse in Flüssigkeiten (zum Beispiel Müller-Plathe u. Reith (1999), Bordat u. a. (2001) und Guevara-Carrion u. a. (2008)), Eigenschaften von Polymeren bei verschiedenen Längenskalen (zum Beispiel Grest u. Kremer (1986), Müller-Plathe (1994) und Kremer u. Müller-Plathe (2002)) sowie generelle statistische Eigenschaften kondensierter Materie (zum Beispiel Praprotnik u. a. (2008b)) zu nennen.

Das vorliegende Kapitel über die Optimierung von Kraftfeldparametern ist wie folgt aufgebaut: Zunächst wird in Abschnitt 3.1 kurz wiederholt, welche Parameter bei der Definition von Kraftfeldern eine wesentliche Rolle spielen. Weiterhin wird ein kurzer Überblick über bisherige Ansätze zur Parametrisierung von Kraftfeldern gegeben, und es wird der in dieser Arbeit verwendete Ansatz eingeführt. Dabei handelt es sich um die Minimierung einer quadratischen Fehlerfunktion zwischen simulierten und experimentellen Daten. In Abschnitt 3.2 werden bisher angewandte Verfahren zur Lösung dieses Optimierungsproblems zusammengefaßt, und die Vor-

und Nachteile der wichtigsten Ansätze innerhalb des letzten Jahrzehnts werden formuliert. Globale, stochastische Verfahren, welche als Voroptimierer einzusetzen sind, wenn keine geeigneten Startparameter für ein Optimierungsproblem gefunden werden können, werden in Abschnitt 3.3 angesprochen. In dieser Arbeit stehen jedoch zunächst lokale Optimierungsverfahren im Vordergrund, die gradientenbasiert sind und in Abschnitt 3.4 vorgestellt werden. Es handelt sich hierbei um die im Bereich der Numerischen Optimierung bekanntesten Methoden, welche zudem sehr gute Konvergenzeigenschaften aufweisen. Deren Anwendung auf Optimierungsprobleme im Bereich Molekularer Simulationen ist allerdings keineswegs trivial. Aufgrund der statistischen Unsicherheiten, mit denen die zu optimierenden Zielgrößen behaftet sind, und der Tatsache, daß keine analytische Form der zu minimierenden Zielfunktion existiert und somit Eigenschaften wie Stetigkeit und Differenzierbarkeit mathematisch nicht bewiesen werden können, sind in dieser Arbeit zusätzliche Überlegungen in Bezug auf die Anwendbarkeit zu treffen. Eine eingehende und detaillierte Studie, inwieweit welche der hier betrachteten Verfahren auf die vorliegende Problemstellung anwendbar sind, wird in Abschnitt 3.5 durchgeführt. Da, wie in Abschnitt 2.6 bereits angesprochen, Molekulare Simulationen im allgemeinen äußerst aufwendig sind, ergibt sich die Notwendigkeit nach einer Erhöhung der Effizienz der Optimierungabläufe. Methoden zur Effizienzerhöhung, darunter auch neuartige, werden in Abschnitt 3.6 aufgeführt und detailliert erläutert. Die praktische Umsetzung, welche die Anwendung sämtlicher vorgestellter Algorithmen auf Molekulare Simulationen umfaßt, mit Ausnahme der globalen Optimierungsverfahren, wurde mit einem automatisierten Optimierungsworkflow namens *GROW* realisiert. Dessen Aufbau und Umfang werden schließlich in Anhang G.1 angesprochen.

3.1 Motivation eines numerischen Optimierungsablaufs

Zur Motivation eines mathematischen Optimierungsproblems zur Parametrisierung von Kraftfeldern wird das folgende vereinfacht dargestellte Kraftfeld betrachtet, welches durch die potentielle Energie beschrieben wird:

$$\begin{aligned} U_{\mathrm{pot}}(r^N) \quad := \quad & \sum_{\mathrm{B.}} \frac{k_r}{2} \left(r - r_0\right)^2 + \sum_{\mathrm{W.}} \frac{k_\phi}{2} \left(\phi - \phi_0\right)^2 + \sum_{\mathrm{D.}} \sum_{n=1}^{m} V_n \cos(n\omega) \\ + \quad & \sum_{i<j}^{N} \left\{ 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \right\}. \end{aligned} \tag{3.1}$$

Dabei stehen die Abkürzungen 'B.', 'W.' beziehungsweise 'D.' für 'Bindungen', 'Winkel' beziehungsweise 'Diederwinkel'. Intramolekulare Kraftfeldparameter sind blau, intermolekulare sind rot dargestellt.
Die zu bestimmenden Parameter sind in diesem Fall also die Kraftfeldkonstanten k_r und k_ϕ für jede Bindung und jeden Winkel, die Rotationskonstanten V_n und deren Anzahl m für jeden Diederwinkel, die Bezugsbindungslängen und -winkelgrößen r_0, ϕ_0 und ω_0, die Lennard-Jones-Parameter ε_{ij} und σ_{ij} sowie sämtliche Ladungen q_i für alle Atome. Es ist zu beachten, daß nach den Lorentz-Berthelot-Relationen aus Gleichung (2.23) nur die Lennard-Jones-Parameter ε_i und σ_i für einzelne Atomtypen zu bestimmen sind, nicht aber ε_{ij} und σ_{ij}.
In Abschnitt 3.1.1 wird zunächst motiviert, wie die Kraftfeldparameter, die in Gleichung (3.1)

in unterschiedlichen Farben dargestellt sind, mit verschiedenen Methoden bestimmt werden önnen. Insbesondere für die Ermittlung der intermolekularen Parameter wird eine empirische Fehlerfunktion zwischen experimentellen und simulierten Daten zu minimieren sein. In Abschnitt 3.1.2 wird ein allgemeiner Ablauf zur Lösung eines derartigen mathematischen Optimierungsproblems dargestellt.

3.1.1 Quantenmechanische und empirische Methoden

Zunächst sei bemerkt, daß innerhalb einer Simulation auch externe Kräfte eine Rolle spielen, zum Beispiel aufgrund von Thermostaten oder Barostaten (siehe Anhänge A.1 bis A.3), bei deren Verwendung ebenfalls Parameter zu bestimmen sind. Diese sollten allerdings aufgrund der in Anhang A beschriebenen Studien korrekt eingestellt sein. Daher liegt der Fokus hier nur auf der Optimierung der internen Kräfte. In der Literatur wurde oftmals gezeigt, daß viele Kraftfelder die molekularen Wechselwirkungen sowohl qualitiativ als auch quantitativ adäquat beschreiben, was an einer Vielzahl von Anwendungen belegt worden ist (vergleiche Guevara-Carrion u. a. (2012)). Daher sind Simulationen, welche auf Kraftfeldern basieren, stets irgendwo zwischen Theorie und Experiment anzusiedeln (Allen u. Tildesley, 1987; Poling u. a., 2000; Leach, 2001).
Intramolekulare Kraftfeldparameter können durch quantenmechanische Methoden bestimmt werden, wobei die Justierung der Parameter zumeist über die Minimierung einer Potentialhyperfläche geschieht. Auch Partialladungen können durch die Position von Kernen und Elektronen ermittelt werden. Die Minimierung der Potentialhyperfläche erfolgt durch numerische Optimierungsverfahren, zum Beispiel über Konjugierte Gradientenmethoden. Quantenmechanische Methoden bedeuten jedoch gerade bei größeren Molekülen einen erheblichen Rechenaufwand. Daher werden oftmals Vereinfachungen vorgenommen und für die Anpassung der Parameter experimentell bestimmte spektroskopische Daten zu Hilfe genommen. Dabei handelt es sich um sogenannte *semiempirische* Verfahren. Auf quantenmechanischen und semiempirischen Methoden beruhen eine Vielzahl von intramolekularen Kraftfeldern, deren Parameter auf diese Weise bereits sehr genau bestimmt wurden. Die bekanntesten Kraftfelder wurden von Weiner u. a. (1984), Berendsen u. van Gunsteren (1987) und Jorgensen u. a. (1996) entwickelt, um nur einige Beispiele zu nennen. Quantenmechanische Ladungsberechnungen wurden vor allem von Frisch u. a. (2004) und Duan u. a. (2003) realisiert. Am Fraunhofer-Institut SCAI wurde ein automatisierter Optimierungsworkflow namens $WOLF_2PACK$ erstellt (Reith u. Kirschner, 2011), welcher quantenmechanisch basierte Verfahren mit atomistischen Modellen kombiniert und in der Lage ist, sowohl optimale intramolekulare Kraftfeldparameter als auch Partialladungen zu berechnen. Intramolekulare Parameter, die durch quantenmechanische Methoden bestimmt werden können, sind in Gleichung (3.1) in blau dargestellt. Eine quantenmechanische Ermittlung von Partialladungen wird ebenfalls in den meisten Fällen durchgeführt, sie können allerdings auch mit empirischen Methoden bestimmt werden, was im folgenden erläutert wird. Sie sind daher in Gleichung (3.1) in cyanblau dargestellt.
Die Bestimmung von intermolekularen Kraftfeldparametern, abgesehen von Partialladungen, durch derartige Methoden hingegen erwies sich als äußerst schwierig und komplex. Bei kurz- und langreichweitigen Potentialen sind sämtliche Parameter nur durch eine ausreichend hohe Anzahl an Systemteilchen zu bestimmen, welche miteinander interagieren. Daher wurden hierzu zumeist empirische Methoden herangezogen, das heißt, die Kraftfeldparameter wurden

an gewisse experimentell bestimmte Zielgrößen angepaßt. Hier sind insbesondere die Studien von Jorgensen u. a. (1984), Martin u. Siepmann (1998), Oostenbrink u. a. (2004) und Eckl u. a. (2008a) zu nennen. Allerdings sind die so gewonnenen intermolekularen Kraftfeldparameter keinesfalls als optimal zu betrachten, da sie nicht an eine Vielzahl von experimentellen Daten angepaßt sind. Außerdem wurden die Parameter zumeist manuell angepaßt, was stets arbeits- und zeitintensiv ist. Es lassen sich jedoch derartige Kraftfeldparameter auch auf andere Substanzen transferieren. In den meisten Fällen ist dann allerdings eine manuelle Nachjustierung erforderlich: In Peguin u. a. (2009) beispielsweise wurden Standardparameter für Ethan auf Fluor direkt auf die Substanz 1,1,1,2-Tetrafluorethan transferiert und zur akkuraten Reproduktion von VLE-Daten nachjustiert. Viele Anwender verwenden in ihren Simulationen die für bestimmte Systeme ermittelten Standard-Kraftfeldparameter, die zwar zu zufriedenstellenden, aber keinesfalls optimalen physikalischen Zielgrößen führen.
Für die Lennard-Jones-Parameter ist bekannt, daß σ in direktem Bezug zur Dichte und ε zur Verdampfungsenthalpie des Systems steht. Daher ist es sinnvoll, diese Parameter möglichst automatisiert an experimentell bestimmte Dichten und Verdampfungsenthalpien anzupassen. Dies ist bereits in den letzten Jahren von einigen Autoren, teilweise simultan zu verschiedenen Temperaturen und Drücken, durchgeführt worden (Faller u. a., 1999; Ungerer u. a., 1999; Bourasseau u. a., 2003; Stoll u. a., 2003b; Sun, 2004). Durch die Definition einer Fehlerfunktion zwischen experimentellen und simulierten Daten, zum Beispiel zu verschiedenen Temperaturen, entstand ein mathematisches Optimierungsproblem. Die Fehlerfunktion wurde bezüglich der betrachteten Kraftfeldparameter durch iterative Verfahren minimiert. Dabei wurden oft Standard-Kraftfeldparameter aus der Literatur als Startwerte eingesetzt. Intermolekulare Parameter, die durch derartige empirische Methoden bestimmt werden, sind in Gleichung (3.1) in rot dargestellt.
Da die Minimierung einer empirischen Fehlerfunktion nach Meinung des Autors der einzig sinnvolle Ansatz ist, um optimale intermolekulare Kraftfeldparameter zu erhalten, wird er im Rahmen dieser Dissertation weiterverfolgt. Allerdings sind in jedem Optimierungsschritt stets Simulationen durchzuführen, welche auch auf atomistischer Ebene noch sehr rechenaufwendig sind. Daher werden hier effiziente numerische Optimierungsverfahren evaluiert und eingesetzt, welche sich durch äußerst gute Konvergenzeigenschaften auszeichnen, zum Beispiel gradientenbasierte Abstiegsverfahren. Inwieweit derartige Optimierungsverfahren einsetzbar sind, wird eingehend zu untersuchen sein, da Simulationsdaten im allgemeinen statistisches Rauschen aufweisen und die Anwendung gradientenbasierter Verfahren somit mathematisch gesehen nicht trivial ist. Diese Untersuchungen sind jedoch nach Meinung des Autors unentbehrlich, da die bisher verwendeten Verfahren Nachteile aufweisen und der Erhalt optimierter Kraftfelder sowohl aus wissenschaftlicher als auch aus industrieller Sicht äußerst wichtig ist.
Ein weiterer wichtiger Aspekt ist, daß ein Algorithmus zur Kraftfeldparametrisierung flexibel und generisch sein muß, das heißt, er sollte nicht auf spezielle Kraftfeldparameter und Zielgrößen ausgerichtet sein. Eine wesentliche Frage ist, ob die Parameter aus Gleichung (3.1) voneinander unabhängig sind. Die Bezugsgrößen, welche sich auf die Geometrien innerhalb eines Moleküls beziehen, können zunächst als unabhängig von intermolekularen Wechselwirkungen angesehen werden. Oftmals werden auch starre Moleküle betrachtet, indem MD-Simulationen mit Nebenbedingungen (siehe Abschnitt B) durchgeführt werden. In diesen Fällen ist der entsprechende intramolekulare Potentialterm gleich Null. Da jedoch äußere Kräfte auf die Bindungen innerhalb eines Moleküls einwirken, besteht eine gewisse Abhängigkeit zwischen intra- und intermolekularen Kraftfeldparametern. Diese Abhängigkeiten sind jedoch oftmals vernachlässigbar gering.

Korrekt wäre es jedoch, bei einer quantenmechanischen Bestimmung von intramolekularen Parametern mehrere Moleküle in Betracht zu ziehen und die Auswirkungen von intermolekularen Kräften miteinfließen zu lassen. Mit dem resultierenden intramolekularen Kraftfeld können dann wiederum die intermolekularen Parameter unabhängig optimiert werden, so daß ein Optimierungskreislauf entsteht.
Ein Problem stellen jedoch die Partialladungen dar: Werden diese quantenmechanisch für ein Molekül berechnet, so wird der Einfluß intermolekularer Wechselwirkungen nicht berücksichtigt. Für mehrere Moleküle wird der Rechenaufwand zu groß. Allerdings verursachen beispielsweise Dispersionskräfte kurzzeitige Dipole. Insofern können elektrostatische Effekte entstehen, die auch von der Wahl von ε und σ in irgendeiner Form abhängig sind. Wie groß diese Abhängigkeiten jedoch tatsächlich sind beziehungsweise wie hoch der Fehler ist, wenn Ladungen und Lennard-Jones-Parameter als unabhängig betrachtet werden, ist *a priori* nicht bekannt. Ein generischer Optimierungsablauf sollte es daher ermöglichen, sämtliche Kraftfeldparameter in die Optimierung miteinzubeziehen. Je mehr Parameter man gleichzeitig optimiert, desto höher wird die Genauigkeit des resultierenden Kraftfelds sein, allerdings steigt dann auch der Rechenaufwand. In dieser Arbeit werden in einigen Fällen auch Partialladungen mit empirischen Methoden optimiert. Sämtliche Problematiken bezüglich multilateraler Abhängigkeiten von Kraftfeldparametern wurden in Hünenberger u. van Gunsteren (1997) diskutiert.
Zunächst wird im folgenden Abschnitt ein allgemeiner Optimierungsablauf, basierend auf einer empirischen Fehlerfunktion, vorgestellt.

3.1.2 Allgemeiner Optimierungsablauf

Da es keine direkte, analytische Beziehung zwischen den Kraftfeldparametern und Zielgrößen gibt und die Berechnung beispielsweise über eine Simulation erfolgt, kann in der Regel kein direktes Optimierungsverfahren angewandt werden, welches zum Beispiel in nur einem Schritt den Gradienten über ein nichtlineares oder lineares Gleichungssystem gleich Null setzt. Die einzige Mögichkeit, dies ansatzweise zu realisieren, wird in Abschnitt 3.2.2 durch die Einführung eines von Ungerer u. a. (1999) und Bourasseau u. a. (2003) eingesetzten Verfahrens diskutiert. Aber auch dieses Verfahren basiert auf einer lokalen Approximation in der Umgebung eines geeigneten Startwertes.
Somit sind sämtliche hier betrachtete Verfahren abhängig von geeigneten Startparametern, welche zum Beispiel aus der Literatur, das heißt aus Standardkraftfeldern, genommen oder aber durch geeignete physikalische und chemische Überlegungen erhalten werden können. Der Optimierungsablauf soll nun mathematisch beschrieben werden, die Startparameter werden als Vektor $x^0 \in \mathbb{R}^N$ bezeichnet. Abbildung 3.1 zeigt einen allgemeinen Optimierungsablauf, der im folgenden detaillierter diskutiert wird:
Der Startvektor x^0 wird als erster aktueller Parametervektor bezeichnet, aus dem die betrachteten physikalischen Zielgrößen berechnet werden. Dies geschieht in der Regel über ein Simulationstool, welches eine oder mehrere MD- oder MC-Simulationen durchführt (siehe Abschnitte 2.3, 2.4 und 2.6) und dann die physikalischen Eigenschaften über die in Abschnitt 2.5 beschriebenen Methoden als thermodynamische Durchschnitte berechnet. Diese werden dann mit den Referenzgrößen verglichen. Bei diesen kann es sich sowohl um experimentelle als auch um theoretische Werte, das heißt um aus quantenmechanischen Methoden bestimmte Werte, handeln. Liegen die berechneten Eigenschaften noch nicht nahe genug an den Referenzwerten, was

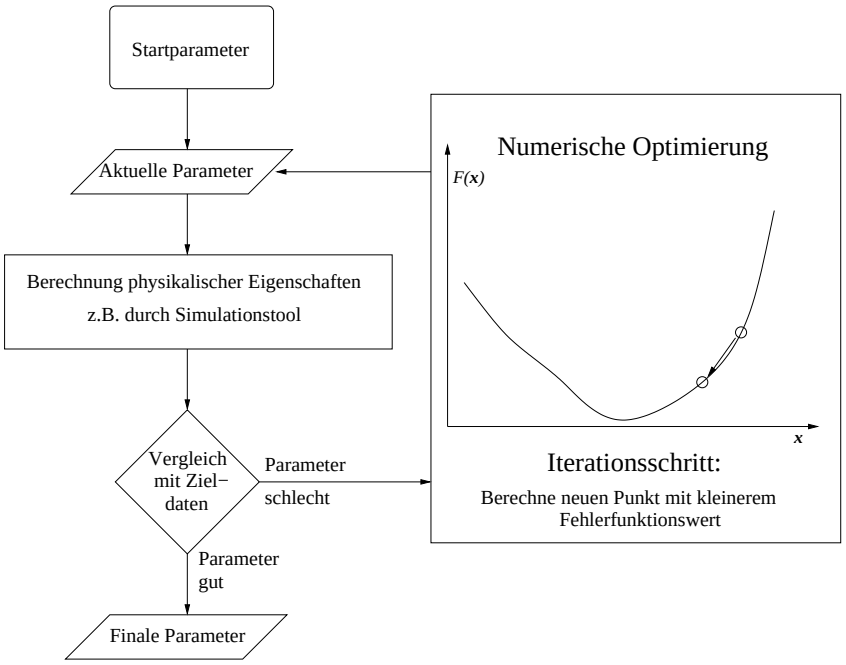


Abbildung 3.1: Allgemeiner Optimierungsablauf: Aus geeigneten Startwerten werden zum Beispiel mithilfe eines Simulationstools die betrachteten physikalischen Eigenschaften bestimmt. Diese werden mit den entsprechenden Referenzwerten verglichen. Liegen die berechneten Eigenschaften noch nicht nahe genug an den Referenzwerten, so wird die Optimierung in Gang gesetzt und neue Parameter mit einem geringeren Fehlerfunktionswert bestimmt, aus denen dann wiederum neue physikalische Eigenschaften berechnet werden. Stimmen die Eigenschaften mit den Referenzwerten bis auf eine gewisse Toleranz überein, so wird der aktuelle Parametervektor als final betrachtet, und der Optimierungsablauf endet. Die zu minimierende Fehlerfunktion $F(x)$ in Abhängigkeit eines Parametervektors x ist in Gleichung (3.2) definiert.

bei den Startparametern im allgemeinen der Fall ist, so wird die eigentliche Optimierung in Gang gesetzt und ein neuer, besserer Parametervektor x^1 bestimmt, aus dem dann wiederum physikalische Eigenschaften neu berechnet werden. Stimmen die Eigenschaften jedoch mit den Referenzwerten bis auf eine gewisse, vorher festzulegende Toleranz überein, so wird der aktuelle Parametervektor als final betrachtet, und der Optimierungsablauf endet. Ist x^1 auch noch nicht als optimal zu betrachten, so wird ein neuer Parametervektor x^2 bestimmt und so weiter. Es sind hierbei zwei Schlüsselfragen zu beantworten:

1. Wie ist ein neuer, besserer Parametervektor möglichst effizient zu bestimmen?

2. In welchem Fall ist ein Parametervektor *gut genug*, das heißt, wie ist der Vergleich sämtlicher betrachteter physikalischer Zielgrößen mit den jeweiligen Referenzwerten in der Praxis zu realisieren?

Mit der ersten Frage wird sich dieses Kapitel in den folgenden Abschnitten hauptsächlich befassen. In diesem Abschnitt soll zunächst kurz die zweite Frage erörtert werden. Die Antwort ist im Prinzip sehr einfach: Da die meisten numerischen Optimierungsverfahren auf der Minimierung einer Funktion $F : \mathbb{R}^N \to \mathbb{R}$ beruhen, wird eine quadratische Fehlerfunktion zwischen den berechneten und experimentellen Daten definiert, so daß sämtliche Zielgrößen simultan angepaßt werden können.

Definition 3.1.1 (Quadratische Fehlerfunktion)**.** *Die* quadratische Fehlerfunktion *zwischen den simulierten und vorgegebenen, zumeist experimentellen Daten* $F : \mathbb{R}^N \to \mathbb{R}_0^+$ *ist gegeben durch*

$$F(x) = \sum_{i=1}^{n} w_i^2 \left(\frac{f_i^{\exp} - f_i^{\mathrm{sim}}(x)}{f_i^{\exp}} \right)^2 . \tag{3.2}$$

Dabei ist $x = (x_1, ..., x_N)^T$ ein Vektor bestehend aus Kraftfeldparametern, N die Dimension des Parameterraums, n die Anzahl an betrachteten physikalischen Zielgrößen, womöglich zu unterschiedlichen Temperaturen. Weiterhin stehen $f_i^{\mathrm{sim}}(x),\ i = 1, ..., n$, für die von der Simulation berechneten Zielgrößen, welche abhängig vom Parametervektor x sind, und $f_i^{\exp},\ i = 1, ..., n$, für die experimentellen Daten. Die Gewichte $w_i^2,\ i = 1, ..., n$, berücksichtigen die Tatsache, daß manche Eigenschaften leichter zu reproduzieren oder genauer vermessen sind als andere. Die Fehlerfunktion F muß bezüglich x in einem kompakten Gebiet $\Omega \subset \mathbb{R}^N$ minimiert werden.
Die Fehlerfunktion aus Definition 3.1.1 ist nur ein Beispiel zur Bewertung der simulierten Zielgrößen. Wegen des Vorhandenseins von statistischem Rauschen ist stets sicherzustellen, daß das Endergebnis nicht durch Zufall erreicht wurde. Dies kann einerseits durch mehrere Replikationen des Optimierungsablaufs erzielt werden, was allerdings bei der Verwendung von MD- oder MC-Simulationen äußerst aufwendig ist. Außerdem werden auch experimentelle Daten zumeist mit Unsicherheiten, das heißt in einem bestimmten Konfidenzintervall, angegeben. Somit ist deren exakte Vorhersage niemals möglich, so daß ein Optimierungsalgorithmus nur bis zu einem bestimmten Punkt erfolgreich sein kann, welcher durch die statistischen Unsicherheiten sowohl bei den simulierten als auch experimentellen Daten bestimmt ist. Um gemäß dem *Maximum-Likelihood-Prinzip* zu garantieren, daß das Ergebnis nicht zufällig ist, sondern aus statistischen Gründen nicht verbessert werden kann, besteht die Möglichkeit, die Fehlerfunktion mithilfe von Standardabweichungen zu gewichten.
Es sind auch andere Arten von Fehlerfunktionen denkbar, welche nicht die euklidische Norm betrachten, sondern zum Beispiel die L_1-Norm oder die L_∞-Norm. Dann ist F jedoch nicht differenzierbar, und die in diesem Kapitel angegebenen gradientenbasierten Verfahren sind nicht anwendbar.

3.2 Vor- und Nachteile bisher angewandter lokaler Optimierungsverfahren

Bislang sind zwei verschiedene Optimierungsmethoden im Bereich der Kraftfeldparametrisierung angewandt worden. Diese werden im folgenden vorgestellt und deren Vor- und Nachteile im einzelnen diskutiert. Weiterhin wird motiviert, inwiefern noch Verbesserungsbedarf besteht und die Forschung bezüglich optimierter Kraftfelder keineswegs als abgeschlossen betrachtet werden kann. Das in Abschnitt 3.2.1 vorgestellte Verfahren ist ein ableitungsfreies Verfahren, der sogenannte *Nelder-Mead-Simplex-Algorithmus*. Die Abschnitte 3.2.2 und 3.2.3 befassen sich mit einem von verschiedenen Autoren angewandten gradientenbasierten Gauß-Newton-Verfahren, welches den Gradienten der Fehlerfunktion mithilfe einer Taylorentwicklung in der Nähe des Minimums lokal linearisiert.

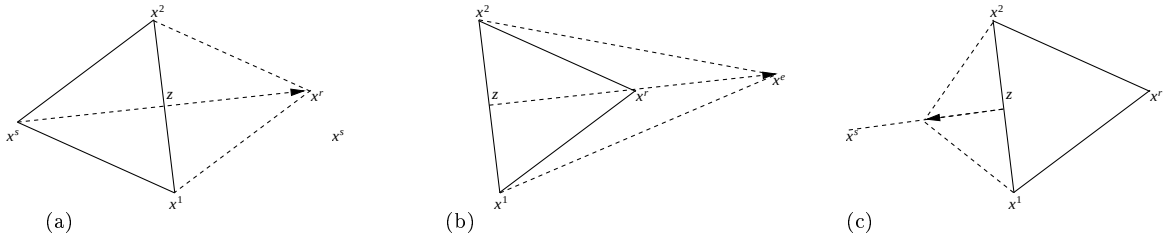


Abbildung 3.2: Die drei Schritte des Simplex-Algorihtmus: Reflexion (a), Expansion (b) und Kontraktion (c).

3.2.1 Simplex-Verfahren nach Nelder und Mead

Das auf Nelder u. Mead (1965) zurückgehende Simplexverfahren zur multidimensionalen nichtlinearen Optimierung ist eine robuste Methode zur Bestimmung eines lokalen Minimums. Im Gegensatz zu vielen anderen numerischen Optimierungsverfahren stellt es keinerlei Anforderungen in Bezug auf Differenzierbarkeit, Monotonie oder Konvexität an die zu minimierende Funktion. Das Verfahren ist somit auf jede beliebige Funktion anwendbar. Ein großer Nachteil ist jedoch, daß es keine direkt zum Minimium weisende Suchrichtung gibt, so daß das Verfahren oftmals dazu neigt, sich zunächst vom Minimum wegzubewegen, um erst später lokal zu konvergieren. In der Nähe des Minimums springt das Simplexverfahren dann um das Minimum herum, anstatt sich direkt daraufzuzubewegen. Im allgemeinen erfordert die Methode eine Vielzahl an Iterationen und Funktionsauswertungen, was sich wegen der langen Rechenzeiten in der Anwendung auf Molekulare Simulationen wegen des sehr hohen Rechenaufwands als äußerst problematisch erweist.
Das Verfahren selbst wird nun kurz eingeführt. Anschließend wird dessen bisherige Anwendung auf die vorliegende Problemstellung dargestellt, und die Unzulänglichkeiten werden anhand dieses Beispiels hervorgehoben.
Es sei zunächst die allgemeine Definition eines *Simplex* gegeben:

Definition 3.2.1 (Simplex). *Sei $N \geq 2$. Ein* Simplex $\mathcal{S}$*, auch* Polyeder *genannt, ist eine Menge von N-dimensionalen Punkten, bestehend aus $N+1$ Elementen,*

$$\mathcal{S} := \{x^1, ..., x^{N+1}\} \subset \mathbb{R}^N,$$

wobei die oBdA von x^1 ausgehenden Kanten $\{x^2 - x^1, ..., x^{N+1} - x^1\} \subset \mathbb{R}^N$ linear unabhängig sind.

Im allgemeinen wird ein Simplex auch als konvexe Hülle von $\mathcal{S}$ definiert. Dann ist ein Simplex für $N = 2$ ein nichtentartetes Dreieck und für $N = 3$ ein nichtentarteter Tetraeder. Innerhalb eines Simplex werden zunächst der schlechteste, der zweitschlechteste und der beste Punkt bestimmt. Diese sind folgendermaßen definiert:

- Schlechtester Punkt: $x^s := \arg\max\{F(x), x \in \mathcal{S}\}$,
- Zweitschlechtester Punkt: $x^a := \arg\max\{F(x), x \in \mathcal{S}, a \neq s\}$,
- Bester Punkt: $x^b := \arg\min\{F(x), x \in \mathcal{S}\}$.

Ein Punkt wird somit umso *besser* angesehen, je kleiner sein Funktionswert ist.
Als nächstes wird das Zentrum z der N besten Punkte berechnet:

$$z := \frac{1}{N} \sum_{i \neq s} x^i.$$

Der Simplex-Algorithmus besteht aus den drei Teilschritten *Reflexion*, *Expansion* und *Kontraktion*:

1. **Reflexion:** Im ersten Schritt des Simplexalgorithmus wird der schlechteste Punkt x^s am Zentrum z *reflektiert*, was in Abbilding 3.2(a) veranschaulicht ist. Der reflextierte Punkt ist gegeben durch:

   $$x^r := z + (z - x^s) = \frac{2}{N} \sum_{j=1}^{N+1} x^j - \left(\frac{2}{N} + 1 \right) x^i.$$

   Falls x^r mindestens genauso gut ist wie der zweitschlechteste Punkt, falls also $F(x^r) \leq F(x^a)$, so wird x^s durch x^r ersetzt, und es resultiert ein neuer Simplex. Im Falle $F(x^b) \leq F(x^r)$ ist der resultierende Simplex der Simplex der nächsten Iteration.

2. **Expansion:** Im Falle $F(x^r) < F(x^b)$ wird x^r noch weiter in dieselbe Richtung *expandiert*, da die heuristische Überlegung zugrundegelegt wird, daß man sich mit x^r bereits auf dem richtigen Weg befindet und diesen Weg weiterverfolgen möchte. Eine Expansion, dargestellt in Abbildung 3.2(b), wird beschrieben durch

   $$x^e := z + \lambda(z - x^s) = \frac{1 + \lambda}{N} \sum_{j=1}^{N+1} x^j - \left(\frac{1 + \lambda}{N} + 1 \right) x^i,$$

   wobei $\lambda > 1$. Eine Reflexion ist somit eine Expansion mit $\lambda = 1$. Im allgemeinen setzt man $\lambda = 2$. Falls $F(x^e) < F(x^r)$, so ersetzt man x^s durch x^e, andernfalls durch x^r.

3. **Kontraktion:** Im Falle einer Verschlechterung durch x^r, also falls $F(x^r) > F(x^a)$, wird der Simplex *kontrahiert*, das heißt, es wird ein kontrahierter Punkt x^k berechnet. Es gilt dann für x^k:

   $$x^k := \begin{cases} z + \frac{1}{2}(x^s - z), & F(x^s) < F(x^r) \\ z + \frac{1}{2}(z - x^s), & F(x^r) \leq F(x^s). \end{cases}$$

   Es handelt sich somit um eine Expansion mit $\lambda = 0.5$ beziehungsweise $\lambda = -0.5$. Eine Kontraktion für $\lambda = 0.5$ ist in Abbildung 3.2(c) dargestellt. Falls dadurch eine Verbesserung erzielt wurde, also im Falle $F(x^k) < F(x^s)$, wird x^s durch x^k ersetzt. Ansonsten ist mit keinem der bisher berechneten Punkte eine Verbesserung erzielt worden, und die letzte Möglichkeit besteht nun darin, den ganzen Simplex in Richtung des besten Punktes zu kontrahieren. Diese Gesamtkontraktion ist gegeben durch

   $$\forall_{i \neq b}\ x^{i'} := \frac{1}{2}(x^i + x^b).$$

Das Simplexverfahren endet, falls eines der folgenden Abbruchkriterien erfüllt ist:

- $\forall_{i,j}\ ||x_i - x_j|| \leq \tau$ oder
- $\frac{1}{N+1}\sum_{i=1}^{N+1}\left(f(x^i) - f(z)\right)^2 \leq \tau^2$,

mit einer Toleranz $\tau > 0$. Ein Startsimplex mit Ecke x^1 und Kantenlänge c läßt sich folgendermaßen konstruieren:

$$\forall_j\ j = 2, ..., N+1\ x^j := x^i + \frac{c}{\sqrt{2}}\frac{\sqrt{N+1}-1}{N}(1, ..., 1)^T + \frac{c}{\sqrt{2}}e_{j-1},$$

wobei $e_1, ..., e_N \subset \mathbb{R}^N$ die kanonischen Einheitsvektoren sind.
Eine Implementation des Simplexverfahrens wird zum Beispiel in Press u. a. (1992) angegeben.

Die verwendete Fehlerfunktion in Faller u. a. (1999) ist:

$$F(x) = \sqrt{\sum_{i=1}^{n} w_i^2 \left(\frac{f_i^{\exp} - f_i^{\mathrm{sim}}(x)}{f_i^{\exp}}\right)^2}, \tag{3.3}$$

wobei die spezifischen physikalischen Eigenschaften mit Gewichten w_i^2 versehen wurden. Die Gewichte hingen stets von den statistischen Unsicherheiten ab. Bei dieser Fehlerfunktion wird die Wurzel gezogen, da sie steiler in Richtung des Minimums abfällt. Bei jeglichen gradientenbasierten Verfahren ist diese Fehlerfunktion jedoch nicht geeignet, da die Wurzelfunktion bei Null nicht differenzierbar ist.
Die in Faller u. a. (1999) optimierten Zielgrößen waren die Dichte ρ und die Verdampfungsenthalpie $\Delta_v H$ zu der Temperatur $T = 293$ K. Die betrachteten chemischen Substanzen waren Methylpentan, Tetrahydrofuran und zyklische Kohlenwasserstoffe. Da stets mehr als zwei Kraftfeldparameter optimiert wurden, handelte es sich stets um unterbestimmte Optimierungsprobleme. Für jedwede Simplex-Iteration war eine Molekulare Simulation durchzuführen. In diesem Fall handelte es sich dabei um MD-Simulationen mit Nebenbedingungen, das heißt, die Bindungslängen wurden mithilfe des SHAKE-Algorithmus (siehe Anhang B) konstant gehalten. Weiterhin wurden kubische periodische Randbedingungen angenommen, und die Simulationen wurden zu Standardbedingungen durchgeführt. Die Verlet-Nachbarliste (siehe Anhang E.1) wurde bis zu einem Radius $r_N = 1$ nm alle 10–15 Zeitschritte bestimmt. Zum Erhalt eines NPT-Ensembles wurden das Berendsen-Thermostat beziehungsweise -Barostat verwendet, welche in den Anhängen A.1 und A.2 beschrieben sind. Die Kopplungszeiten betrugen 0.2 ps für den Druck und 2 ps für die Temperatur. Pro Äquilibrierungslauf (siehe Anhang D) wurden 50 ps gerechnet, mit einem Zeitschritt von $\Delta t = 1$ fs. Ausgewertet wurde das Simplexverfahren zunächst anhand von $N = 125$ Methylpentanteilchen. Nach 11 Simplexiterationen, die etwa zwei Wochen auf einem DEC-433MHz-Prozessor benötigten, ergab sich eine Genauigkeit von +0.77% bezüglich der Verdampfungsenthalpie und von -0.11% bezüglich der Dichte, wobei lediglich die Lennard-Jones-Parameter ε und σ für je ein Kohlenstoff- und Wasserstoffatom optimiert wurden. Im Falle von Tetrahydrofuran wurden zusätzlich zu ε und σ für das Sauerstoffatom auch die Ladungen des Sauerstoff- und eines Kohlenstoffatoms optimert. Dabei wurde die Verdampfungsenthalpie nach 53 Iterationen äußerst genau vorhergesagt, die Dichte allerdings nur ungenau. Daher wurden nach einem durchgeführten Optimierungsablauf sämtliche Parameter bis auf σ festgehalten, und nach einer zusätzlichen Optimierung wurde

die Dichte bis auf -0.34% vorhergesagt. In einer weiteren Applikation anhand von zyklischen Kohlenwasserstoffsystemen wurden sowohl Dichte als auch Verdampfungsenthalpie genauestens vorhergesagt. Diese Ergebnisse mögen zunächst sehr gut erscheinen, allerdings wurden sie nur für unterbestimmte Systeme, das heißt in diesem Fall für experimentelle Daten zu nur einer Temperatur, erzielt. Nach einer Validierung der auf diese Art erhaltenen Kraftfeldparameter, also für $T = 293$ K, bezüglich Ethylenoxid in Bezug auf experimentelle Daten in einem Temperaturbereich vom 230–400 K ergab sich in Müller u. a. (2008) ein durchschnittlicher Fehler von 11.4% für die Dichte, also ein äußerst schlechtes Ergebnis.
Das Simplexverfahren ist ein sehr robustes Verfahren bezüglich lokaler Optimierung und ist geometrisch erklärbar. Es ist jedoch ein sehr heuristisches Verfahren, dessen Konvergenz äußerst langsam ist. Bei anderen Applikationen (vergleiche Müller u. a. (2008)) benötigte es 70–100 Iterationen.

3.2.2 Angepaßtes Gauß-Newton-Verfahren, Variante 1

Das zweite hier vorgestellte Verfahren ist ein auf Molekulare Simulationen angepaßtes Gauß-Newton-Verfahren. Es handelt sich um einen direkten gradientenbasierten Ansatz, das heißt, die Methode läuft auf die Lösung eines Linearen Gleichungssystems (LGS) hinaus, das dadurch entsteht, daß der Gradient gleich 0 gesetzt wird. Weiterhin werden physikalisch basierte Formeln für die partiellen Ableitungen $\frac{\partial f_i^{\text{sim}}}{\partial x_j}(x)$ aus Gleichung (3.52) verwendet. Das Verfahren wurde in der Literatur zur Optimierung von Kraftfeldparametern von Ungerer u. a. (1999) und Bourasseau u. a. (2003) eingesetzt. Daher wird innerhalb dieser Arbeit zumeist vom *Verfahren nach Ungerer und Bourasseau* gesprochen.
Die Fehlerfunktion bei dieser Methode ist:

$$\bar{F}(x) := \frac{1}{n} \sum_{i=1}^{n} \frac{\left(f_i^{\text{sim}}(x) - f_i^{\text{exp}}\right)^2}{s_i^2}, \tag{3.4}$$

wobei s_i ein Schätzwert für die statistische Unsicherheit auf $\left(f_i^{\text{sim}}(x) - f_i^{\text{exp}}\right)$ ist, welcher aus den Unsicherheiten auf den simulierten und experimentellen Daten gemäß

$$s_i^2 = \left(s_i^{\text{sim}}\right)^2 + \left(s_i^{\text{exp}}\right)^2 \tag{3.5}$$

erhalten wird.
Auch dieses Verfahren basiert auf einem Startvektor x^0, aus dem ein nächster Vektor $x^1 = x^0 + \Delta x$ bestimmt wird. Hierzu wird von einer Taylorentwicklung von f_i^{sim} um x^0 ausgegangen:

$$f_i^{\text{sim}}(x^0 + \Delta x) \doteq f_i^{\text{sim}}(x^0) + \sum_{k=1}^{N} \frac{\partial f_i^{\text{sim}}}{\partial x_k} \Delta x_k, \;\; i = 1, ..., n. \tag{3.6}$$

Wird dies in Gleichung (3.4) eingesetzt und der Gradient der Fehlerfunktion gleich 0 gesetzt, so ergibt sich:

$$\sum_{i=1}^{n} \frac{\left(f_i^{\text{sim}}(x^0) - f_i^{\text{exp}} + \sum_{k=1}^{N} \frac{\partial f_i^{\text{sim}}}{\partial x_k} \Delta x_k\right) \frac{\partial f_i^{\text{sim}}}{\partial x_j}}{s_i^2} = 0, \;\; j = 1, ..., N. \tag{3.7}$$

Dies ist ein LGS für $\Delta x_k,\ k = 1, ..., N$. Die partiellen Ableitungen $\frac{\partial f_i^{\text{sim}}}{\partial x_j}$ können zum Beispiel durch finite Differenzen approximiert werden. Allerdings ist dies äußerst aufwendig, da hierfür N weitere Simulationen benötigt werden, und aufgrund der statistischen Unsicherheiten wird der Gradient oft sehr ungenau berechnet. Dies wird in Abschnitt 3.5 noch näher erläutert.
Von Bourasseau u. a. (2003) wurde die Verwendung statistischer Fluktuationen vorgeschlagen: Sei X eine beliebige physikalische Eigenschaft und $\langle X \rangle$ deren thermodynamischer Durchschnitt. Dann gilt:

$$\frac{\partial \langle X \rangle}{x_k} = \left\langle \frac{\partial X}{\partial x_k} \right\rangle - \beta_T \left(\left\langle X \frac{\partial U_{\text{pot}}}{\partial x_k} \right\rangle - \langle X \rangle \left\langle \frac{\partial U_{\text{pot}}}{\partial x_k} \right\rangle \right). \tag{3.8}$$

Für die Dichte ρ, welche aus Gleichung (C.8) errechnet werden kann, gilt dann nach Gleichung (3.8):

$$\frac{\partial \langle \rho \rangle}{x_k} = -\frac{\beta_T M N}{\langle V \rangle^2 N_A} \left(\left\langle V \frac{\partial U_{\text{pot}}}{\partial x_k} \right\rangle - \langle V \rangle \left\langle \frac{\partial U_{\text{pot}}}{\partial x_k} \right\rangle \right). \tag{3.9}$$

Dabei ist die Ableitung des momentanen Volumens nach den Kraftfeldparametern gleich Null, und somit entfällt der erste Term auf der rechten Seite von Gleichung (3.8).
Für die Verdampfungsenthalpie $\Delta_v H$, welche mittels Gleichung (C.3) approximiert werden kann, resultiert aus Gleichung (3.8):

$$\frac{\partial \langle \Delta_v H \rangle}{x_k} = -\frac{N_A}{N} \left[\left\langle \frac{\partial U_{\text{nb}}}{\partial x_k} \right\rangle - \beta_T \left(\left\langle U_{\text{nb}} \frac{\partial U_{\text{pot}}}{\partial x_k} \right\rangle - \langle U_{\text{nb}} \rangle \left\langle \frac{\partial U_{\text{pot}}}{\partial x_k} \right\rangle \right) \right]. \tag{3.10}$$

Die partiellen Ableitungen der potentiellen Energie werden mittels finiter Differenzen approximiert.

Nach dem folgenden Lemma ist das Verfahren von Ungerer und Bourasseau nicht auf unterbestimmte Probleme anwendbar:

Lemma 3.2.2. *Im Falle $N > n$ ist die Matrix $A = (a_{kj})_{k,j=1,...,N}$ des LGS aus Gleichung (3.7) singulär.*

Beweis: Aus Gleichung (3.7) folgt, daß

$$a_{kj} = \sum_{i=1}^{n} \frac{1}{s_i^2} \frac{\partial f_i^{\text{sim}}(x)}{\partial x_k} \frac{\partial f_i^{\text{sim}}(x)}{\partial x_j}.$$

Es seien nun $w_{ij} := \frac{1}{s_i^2} \frac{\partial f_i^{\text{sim}}(x)}{\partial x_j},\ i = 1, ..., N,\ j = 1, ..., n$, und

$$v_i := \begin{pmatrix} \frac{\partial f_i^{\text{sim}}(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f_i^{\text{sim}}(x)}{\partial x_N} \end{pmatrix} \in \mathbb{R}^N,\ i = 1, ..., n.$$

Dann gilt für jede Spalte a_j von A:

$$a_j = \sum_{i=1}^{n} w_{ij} v_i, \Rightarrow \forall_{j=1,...,N}\ a_j \in \text{span}(v_1, ..., v_n).$$

Falls $N > n$, so gilt:

$$\forall_{j>n}\ a_j \in \mathrm{span}(a_1, ..., a_n).$$

Dann ist A singulär. □

Dies ist im mathematischen Sinne als erheblicher Nachteil anzusehen, denn es ist ein Optimierungsablauf anzustreben, welcher für sämtliche Optimierungsprobleme anwendbar ist.
Ist die Matrix aus Gleichung (3.7) jedoch regulär, so ist es möglich, weitere sukzessive Iterationen durchzuführen. Auf den ersten Blick ist das Verfahren von Ungerer und Bourasseau ein direktes Verfahren, allerdings ist es letztendlich eine iterative Methode, da mit der Lösung des LGS aufgrund der Taylor-Entwicklung nur eine approximative Lösung erhalten wird. Es stellt sich jedoch das folgende Problem: Einige Kraftfeldparameter können physikalisch unsinnige, zum Beispiel negative Werte, annehmen, oder die simulierten Daten f_i^{sim} können nicht durch eine Taylorentwickung wie in Gleichung (3.6) linear approximiert werden. In diesem Fall ist eine weitere Optimierung sinnlos, da schon der erste Schritt sinnlos war, was auf die schlechte Wahl der Startparameter zurückzuführen ist. Das grundsätzliche Problem des Verfahrens von Ungerer und Bourasseau liegt in der Wahl der initialen Kraftfeldparameter. Eine nach dem ersten Glied abgeschnittene Taylorentwicklung wie in Gleichung (3.6) ist nur lokal in einer kleinen Umgebung der Startparameter ausreichend genau, und somit ist das Verfahren direkt nur in einer kleinen Umgebung des Optimums anwendbar. Das bedeutet, daß die Startparameter prinzipiell sehr nah an den optimalen Parametern liegen müssen. Ansonsten ist die Konvergenz mit der eines einfachen Newton-Verfahrens zur Nullstellenbestimmung vergleichbar, welches ebenfalls äußerst startwertabhängig ist.
Die praktischen Ergebnisse des Verfahrens sind eher als schlecht anzusehen. Angewandt wurde es auf Kohlenstoffatome innerhalb von π-Bindungen von Alkenen (Bourasseau u. a., 2003). Dabei wurden sowohl die Lennard-Jones-Parameter ε und σ als auch die Distanz δ zwischen einem Kohlenstoffatom und dem Kraftzentrum an experimentelle Werte angepaßt. Bei letzteren handelte es sich um *Vapor-Liquid-Equlibrium-(VLE-)*Daten, also um Zielgrößen des Phasengleichgewichts zwischen Flüssigkeit und Dampf. Betrachtet wurden dabei die Sättigungsdichte ρ_l, die Verdampfungsenthalpie $\Delta_v H$ und der Dampfdruck p_σ. Es wurden MC-Simulationen mit einer Konfigurationsverzerrungsmethode durchgeführt. Für verschiedene Alkene bewegten sich die relativen Fehler bei den erzielten optimalen Kraftfeldparametern in Bezug auf die Sättigungsdichte im Bereich 0.6–3.2%, in Bezug auf die Verdampfungsenthalpie im Bereich 0.5–5.6% und in Bezug auf den Dampfdruck im Bereich 2.9–22.5%. Die Anzahl an benötigten Iterationen ist in der Publikation nicht angegeben. Aufgrund der schlechten Approximation durch das Taylor-Modell gerade bei Kraftfeldparametern, die weit vom Minimum entfernt sind, wird diese jedoch vermutlich im Bereich von Faller u. a. (1999) gewesen sein. Zudem waren auch sukzessive Optimierungen erforderlich, das heißt, zunächst wurden die LJ-Parameter und dann die Partialladungen angepaßt. Das Verfahren von Ungerer und Bourasseau hat somit ebenfalls für die vorliegende Problemstellung erhebliche Nachteile, obwohl es in der Regel bessere Ergebnisse liefert als das Simplex-Verfahren.

Als Vorteil des Verfahrens nach Ungerer und Bourasseau wird angesehen, daß es als direktes Verfahren betrachtet werden kann, wenn die Taylorentwicklung aus Gleichung (3.6) für die physikalischen Zielgrößen eine ausreichend gute Approximation ist. Dies ist der Fall, wenn die Zielgrößen zumindest in einer Umgebung linear und die Fehlerfunktion aus Gleichung (3.4)

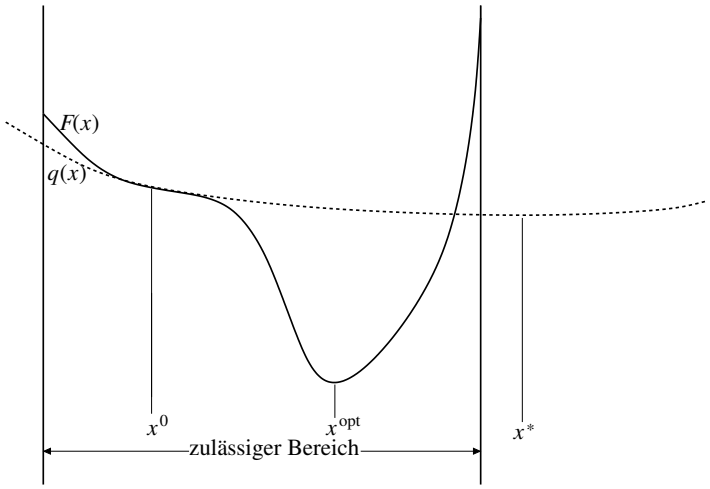


Abbildung 3.3: Problem beim von Ungerer u. a. (1999) und Bourasseau u. a. (2003) sowie Stoll (2005) eingesetzten Gauß-Newton-Verfahren bei Wahl eines ungeeigneten Startvektors x^0: Das quadratische Modell q muß die Fehlerfunktion F nicht notwendigerweise global wiedergeben. Das globale Minimum von q, $x^\star$, kann somit außerhalb des zulässigen Bereichs liegen, und das Verfahren konvergiert nicht gegen das Minimum von F, x^{opt}.

quadratisch von den Kraftfeldparametern abhängt. Dann kann die Fehlerfunktion in dieser Umgebung durch ein quadratisches Modell approximiert und dessen Minimum durch Lösung des LGS direkt bestimmt werden. Dies kann jedoch nur in einer Umgebung des lokalen Optimums vorausgesetzt werden. Ansonsten, besonders bei der Wahl ungeeigneter Startwerte, ist es möglich, daß das Minimum des quadratischen Modells nicht mehr im zulässigen Bereich für die Kraftfeldparameter liegt. Dieser Nachteil ist in Abbildung 3.3 veranschaulicht.

3.2.3 Angepaßtes Gauß-Newton-Verfahren, Variante 2

Eine zweite Variante eines angepaßten Gauß-Newton-Verfahrens wurde in Stoll (2005) beschrieben. In dieser Arbeit wird die Methode zumeist als *Verfahren nach Stoll* bezeichnet. Der Algorithmus ist iterativ und stellt sicher, daß die Iterationen stets im zulässigen Bereich liegen und gegen ein lokales Minimum konvergieren. Dies geschieht zum einen durch eine akkurate Wahl der Startparameter, die bereits sehr nahe am Optimum liegen, was jedoch zumeist eine hohe Anzahl an vorgeschalteten Simulationen erfordert. Zum anderen wird die folgende Fehlerfunktion minimiert:

$$F_{\mathrm{Stoll}}(x) := \sum_{i=1}^{n} \sum_{T \in \mathcal{T}_{\mathrm{Stoll}}} w_{i,T} \left(\frac{f_{i,T}^{\mathrm{exp}} - f_{i,T}^{\mathrm{sim}}(x)}{f_{i,T}^{\mathrm{exp}}} \right)^2 . \tag{3.11}$$

Zumeist ist dabei $n = 3$, das heißt es werden Siededichte, Verdampfungsenthalpie und Dampfdruck, also VLE-Daten, optimiert. Der Temperaturbereich $\mathcal{T}_{\mathrm{Stoll}}$ enthält sämtliche Temperaturen aus einem bestimmten Temperaturintervall im Abstand von 1 K. Weiterhin sind $w_{i,T}, f_{i,T}^{\mathrm{exp}}$ beziehungsweise $f_{i,T}^{\mathrm{sim}}(x)$ die Gewichte sowie die experimentellen beziehungsweise simulierten

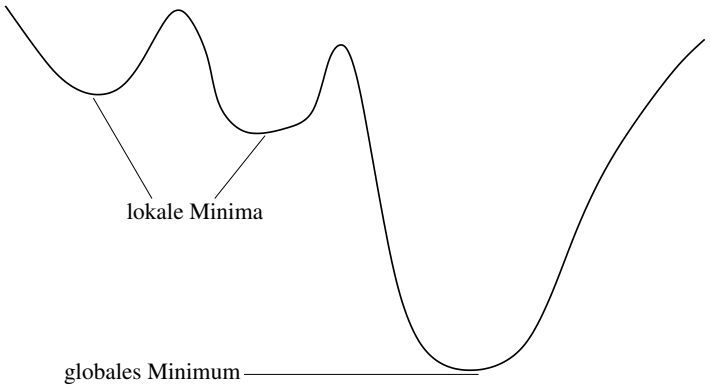


Abbildung 3.4: Suche nach einem globalen Minimum, welches sich deutlich von anderen lokalen Minima unterscheidet.

Werte einer Zielgröße i zur Temperatur T. Dies macht die Fehlerfunktion konvexer, wodurch diese besser durch ein quadratisches Modell approximierbar ist. Ein praktischer Nachteil ergibt sich jedoch dadurch, daß auch die experimentellen Daten zu jeder Temperatur zur Verfügung stehen müssen. In Stoll u. a. (2003b) wurde die Methode für Kohlenstoffdioxid und halogenierte Kohlenwasserstoffe angewandt. Dabei wurden keine Molekularen Simulationen durchgeführt. Die VLE-Daten wurden mithilfe der in Abschnitt 3.5.2 vorgestellten Korrelationsfunktionen ermittelt, welche bestimmte Molekulare Simulationen ersetzen können. Sämtliche Eigenschaften konnten innerhalb ihrer vorher festgelegten Toleranzbereiche reproduziert werden. Die Anzahl an Funktionsauswertungen konnte zwar durch eine effiziente Gradientenberechnung für die physikalischen Zielgrößen $f_{i,T}^{\mathrm{sim}}(x)$ (siehe Abschnitte 3.6.2 und 3.6.5) von $\mathcal{O}(N)$ auf $\mathcal{O}(1)$ reduziert werden, allerdings ist in Stoll u. a. (2003b) nicht erwähnt, wie viele Iterationen tatsächlich notwendig waren. Aus demselben Grund wie beim Verfahren von Ungerer und Bourasseau ist jedoch auch hier zu vermuten, daß diese relativ hoch gewesen sein muß. Geeignete Startparameter werden vermutlich nur durch Probieren zu finden gewesen sein.

Alles in allem hat die Methode nach Stoll dieselben Nachteile wie die Methode nach Ungerer und Bourasseau. Erst eine adaptierte Variante, welche in Abschnitt 3.6.5 vorgestellt wird, macht das Verfahren nach Stoll zu einem auf die vorliegende Problemstellung generisch anwendbaren Algorithmus.

3.3 Globale Optimierungsverfahren

Sämtliche lokalen numerischen Optimierungsverfahren konvergieren nur dann gegen ein globales Minimum, wenn die zu minimierende Funktion konvex ist, oder aber, wenn sich die Startparameter im Einzugsbereich des globalen Minimums befinden. Die Eigenschaft der Konvexität kann jedoch im allgemeinen nicht vorausgesetzt werden. Bei der vorliegenden Problemstellung ist eine lokale Konvergenz in vielen Fällen jedoch ausreichend, da es in der Literatur bereits eine Vielzahl an Standardkraftfeldparametern gibt, die sich sehr gut als Startparameter für gradientenbasierte Verfahren eignen, welche in den Abschnitten 3.4 und 3.5 vertieft werden. Ansonsten besteht immer noch die Möglichkeit, Verfahren zur groben Startwertbestimmung

vorzuschalten. Es gibt Verfahren, welche unter bestimmten Voraussetzungen gegen ein globales Minimum konvergieren. Die Grundidee derartiger Verfahren besteht entweder darin, innerhalb der Iterationen auch größere Funktionswerte zuzulassen, um nicht in einem lokalen Minimum hängenzubleiben, oder aber, die Fehlerfunktion global zu modellieren, das heißt zu interpolieren oder approximieren, und dann das globale Minimum des (meist differenzierbaren) Modells zu bestimmen. Es ist abzuwägen, ob globale Optimierungsverfahren trotz des damit verbundenen höheren Rechenaufwands besser geeignet sind als lokale und wie sie sich bei statistischem Rauschen verhalten. Weiterhin ist zu untersuchen, ob eine Kombination globaler und lokaler Minimierungsalgorithmen besser geeignet ist als die Verwendung eines globalen Minimierers alleine.
Globale Optimierungsverfahren basieren stets auf den folgenden drei Grundprinzipien:

1. Das globale Minimum muß sich deutlich von anderen lokalen Minima unterscheiden, was in Abbildung 3.4 veranschaulicht ist.
2. Das Verfahren muß so gestaltet sein, daß es stets ein lokales Minimum verlassen kann, im globalen Minimum jedoch hängenbleibt.
3. Wenn ein Minimum gefunden wird, welches nicht mehr verlassen werden kann, so muß es sich um das globale handeln, und das Verfahren wird abgebrochen.

Grundsätzlich sind globale Optimierungsverfahren in zwei Kategorien einzuteilen, in stochastische Methoden und solche, die eine Metamodellierung der Fehlerfunktion durchführen. Aus jeder dieser Kategorien wird in dieser Arbeit jeweils ein vielversprechendes Verfahren selektiert. Aus den stochastischen Methoden wird das sogenannte *CMA-ES* gewählt, ein Evolutionärer Algorithmus, der auf der Streckung von Ellipsoiden in Richtung des globalen Minimums beruht und bei geeigneter Größe der Ellipsoide dazu in der Lage ist, intermediäre lokale Minima zu überspringen. Gemäß Hansen u. Ostermeier (2001) ist CMA-ES äußerst effizient und robust auch bei nichtkonvexen Funktionen einsetzbar. *Evolutionäre Algorithmen* werden zunächst allgemein in Abschnitt 3.3.1 eingeführt. Sie imitieren aus der Biologie stammende Evolutionsprozesse innerhalb einer Population. Ein größerer Funktionswert wird dabei stets mit einer gewissen Wahrscheinlichkeit akzeptiert. CMA-ES im speziellen wird dann in Abschnitt 3.3.2 beschrieben.
Aus der zweiten Kategorie wird das am Fraunhofer SCAI entwickelte *DesParO* gewählt, das in Abschnitt 3.3.3 diskutiert wird. Nach Bäuerle u. a. (2004) ist DesParO sehr gut geeignet für multikriterielle Optimierungsaufgaben und wurde bereits erfolgreich für Anwendungen in der Automobilindustrie eingesetzt (Stork u. a., 2008). Es ist daher äußerst interessant, zu untersuchen, inwieweit die Software auch für die vorliegende Problemstellung nutzbar ist.

3.3.1 Evolutionäre Algorithmen

Bei *Evolutionären Algorithmen (EAs)* handelt es sich um globale Optimierungsverfahren, welche den Prozeß der biologischen Evolution imitieren. Die Idee dabei ist, daß die Individuen innerhalb einer *Population* derart selektiert werden, daß nur diejenigen überleben, die sich als möglichst geeignet erweisen. Dann dürfen diese ihre Eigenschaften vererben, und die Population rückt immer näher an das globale Optimum.
Geprägt wurden die EAs insbesondere durch Schwefel (1995). Auf diesem Buch basiert der

gesamte vorliegende Abschnitt. EAs sind stochastische Verfahren, innerhalb derer der einzige deterministische Schritt die Selektion der Individuen ist. Die wichtigsten Schritte der EAs sind Repräsentation der Individuen, Rekombination, Mutation und Selektion, welche im folgenden im einzelnen vorgestellt werden:

1. **Repräsentation der Individuen:** Sogenannte *Eltern* und *Kinder* einer Population werden als Vektoren $x = (x_1, ..., x_N; \sigma_1, ..., \sigma_N)$ repräsentiert, wobei $x_i,\ i = 1, ..., N$, die zu optimierenden Parameter und $\sigma_i,\ i = 1, ..., N$, zufällige Abweichungen sind, welche die Suche im Parameterraum und somit die biologische Evolution kontrollieren. Es wird zwischen zwei verschiedenen *Evolutionsstrategien (ES)* unterschieden: der (μ, λ)-ES und der $(\mu + \lambda)$-ES, wobei μ für die Anzahl Eltern und λ für die Anzahl Kinder steht. Die (μ, λ)-Strategie ist *nicht-überlappend*, das heißt die μ Eltern der aktuellen Generation werden ausschließlich aus den λ Nachkommen der vorangegangenen Generation selektiert. Die $(\mu + \lambda)$-Strategie hingegen ist *überlappend*, das heißt, die μ Eltern der aktuellen Generation werden aus der gesamten vorangegangenen Generation selektiert, bestehend aus $\mu + \lambda$ Individuen. Somit ist es möglich, daß bestimmte Individuen über mehrere Generationen hinweg, ja sogar ewig überleben können.
   Bei der Initialisierung werden die Parameter x_i zufällig gemäß einer bestimmten Verteilung aus dem Intervall $[x_i^{\min}, x_i^{\max}]$, welches Nebenbedingungen für die einzelnen Komponenten von x festlegt, gewählt. Für die σ_i wird ein Startwert $\sigma_0 < 1$ definiert, und die σ_i werden dann gemäß $\sigma_i = \sigma_0(x_i^{\max} - x_i^{\min})$ bestimmt. Dies erlaubt die Variation der x_i über verschiedene Größenordnungen.

2. **Rekombination:** Unter *Rekombination* versteht man in der Genetik den Austausch von Allelen beziehungsweise die Neuanordnung von genetischem Material, das heißt, es ist der Prozeß der Produktion von Kindern aus Eltern. Zusammen mit der Mutation ist die Rekombination verantwortlich für die genetische Variabilität. Nach Schwefel (1995) gibt es zwei verschiedene Klassifikationen von Rekombinationen:

   a) *Lokal/Global: Lokale* Operatoren generieren ein Kind vollständig aus zwei zufällig gewählten Eltern, *globale* Operatoren wählen zufällig ein neues Elternpaar für jede Komponente des Kindes.

   b) *Diskret/Intermediär: Diskrete* Operatoren weisen jeder Kindkomponente denselben Wert einer der beiden Elternkomponenten zu, *intermediäre* Operatoren bilden den Durchschnitt der entsprechenden Elternkomponenten.

   Rekombinationen erfolgen ohne Zurücklegen, das heißt, es ist nicht möglich, daß ein Kind aus zwei Kopien desselben Elternteils entsteht. Zumeist werden lokale, diskrete Rekombinationen verwendet.
   Im allgemeinen können EAs auch ohne Rekombination auskommen, da zumeist die Mutation der entscheidende Schritt ist. In diesem Fall wird von einer *asexuellen Fortpflanzung* innerhalb der Population ausgegangen. Die Verwendung von Rekombinationen führt jedoch meistens zu einer Erhöhung der Effizienz.

3. **Mutation:** Bei der Mutation handelt es sich um zufällige Veränderungen von Kindkomponenten $\tilde{x}_i := x_i + z_i$, wobei die z_i so gewählt werden müssen, daß große Veränderungen selten und kleine häufig auftreten. Im diskreten Fall stammen die z_i aus einer Binomialverteilung, im kontinuierlichen Fall aus einer Normalverteilung mit Standardabweichung σ_i.

Gleichwahrscheinliche Änderungen liegen dann auf einem Hyperellipsoid mit Gleichung

$$\sum_{i=1}^{N} \left(\frac{z_i}{\sigma_i} \right)^2 \equiv const.$$

um den aktuellen Parametervektor x mit Semiachsen $\sigma_i,\ i = 1, ..., N$, wobei es sich im Falle $\forall_{i=1,...,N}\ \sigma_i := \sigma \equiv const.$ um eine Kugel mit Radius σ handelt. Die σ_i können unabhängig voneinander im Laufe des Algorithmus variieren oder aber mittels eines Referenzparameters σ_R gemäß $\sigma_i = \sigma_R(x_i^{\max} - x_i^{\min})$ kontrolliert gewählt werden. Dann variiert jedoch der Referenzparameter über den gesamten Algorithmus hinweg.
Rekombination alleine kann nicht zum Erfolg führen, da der Parameterraum dadurch nicht genügend exploriert wird. Beispielsweise werden bei intermediärer Rekombination durch die Durchschnittsbildung niemals Werte in der Nähe des Randes des zulässigen Intervalls angenommen. Der entscheidende Schritt, welcher die Konvergenz gegen ein globales Minimum ermöglicht, ist daher die Mutation. Die einfachste Variante ist dabei, den Referenzparameter σ zu fixieren und im Laufe des Algorithmus nicht zu verändern. Eine etwas effizientere Methode besteht darin, σ in jeder Generation um einen Faktor $0 < c_0 < 1$ zu verkleinern (sogenanntes *Einfaches Abkühlen – Simple Annealing*). Dies entspricht der Idee, zu Beginn des Verfahrens größere Mutationen zuzulassen, um möglichst schnell in die Nähe des Minimums zu gelangen, und zum Ende hin, wo nur noch lokale Verfeinerungen wünschenswert sind, nur noch kleine Mutationen.
Eine populäre Methode, welche diesen Ansatz ebenfalls jedoch auf effizientere Weise verfolgt, geht auf Beyer u. Schwefel (2002) zurück: Dabei werden die Standardabweichungen folgendermaßen mutiert:

$$\tilde{\sigma}_i := \sigma_i \ell_g \ell_l,$$

wobei

$$\begin{aligned} \ell_g &:= \exp\left(\frac{1}{\sqrt{2N}} \mathcal{N}(0,1) \right), \\ \ell_l &:= \exp\left(\frac{1}{\sqrt{2\sqrt{N}}} \mathcal{N}(0,1) \right), \end{aligned}$$

wobei $\mathcal{N}$ für die Normalverteilung steht. Dabei fungiert ℓ_g als globale Skalierung der Mutationsgröße, wird also unabhängig für jedes Kind und für alle σ_i berechnet, wohingegen ℓ_l unabhängig für jede Komponente berechnet wird.
Weiterhin gibt es komplexere Mutationsverfahren, die die Entwicklung der zu optimierenden Funktion über eine bestimmte Anzahl an Generationen mitberücksichtigen und dementsprechend die Mutationsgröße festlegen. Ein Beispiel hierfür ist die sogenannte $\frac{1}{5}$-*Erfolgsregel*, welche die Anzahl an Erfolgen, das heißt Verkleinerungen der zu minimierenden Funktion, über eine gewisse Anzahl an Generationen speichert und diese ins Verhältnis zu der Anzahl an Versuchen, die Funktion zu verkleinern, setzt. Dieses Verhältnis sei mit r bezeichnet. Falls $r > \frac{1}{5}$, so wird σ vergrößert (noch weit weg vom Optimum), andernfalls wird sie verkleinert (nahe am Optimum $\Rightarrow$ lokale Verfeinerungen gesucht). Präziser läßt sich die $\frac{1}{5}$-Erfolgsregel folgendermaßen formulieren: *Für alle n Mutationen wird überprüft, wie viele Erfolge über die vorangegangenen* $10n$ *Mutationen hinweg erzielt*

wurden. Ist diese Anzahl $\leq 2n$*, so wird* σ *mit 0.85 multipliziert, ansonsten durch 0.85 dividiert.*

4. **Selektion:** Die Selektion ist der einzige deterministische Schritt innerhalb von EAs. Sie legt fest, welche Kinder der aktuellen Generation Eltern der nächsten Generation sein werden. Dies geschieht mittels Evaluation gemäß ihrem Funktionswert. Nur die fittesten sollen überleben, das heißt diejenigen mit den kleinsten Funktionswerten. Es ist jedoch weiterhin die globale Konvergenz zu berücksichtigen, das heißt, die einfache Wahl der λ beziehungsweise $\mu + \lambda$ Individuen mit kleinstem Funktionswert kann dazu führen, daß der Algorithmus in einem lokalen Minimum hängenbleibt, da keine größeren Funktionswerte zugelassen werden. Effizienter erscheint die Möglichkeit, durch sogenannte *Abschneidemethoden* nur bestimmte Individuen per Zufallsprinzip auszuwählen, diese paarweise anzuordnen und aus jedem Paar das jeweils beste Individuum zu selektieren. Dies kann jedoch zur Folge haben, daß das beste Individuum innerhalb der aktuellen Population gar nicht überlebt. Um dies zu vermeiden, bedient man sich sogenannter *semi-überlappender* Selektionsmethoden, welche garantieren, daß stets das beste Kind oder Elternteil in der aktuellen Generation überlebt.

Ein EA kann eins oder mehrere der folgenden Abbruchkriterien haben:

- Die zu minimierende Funktion hat einen bestimmten Toleranzwert τ_0 unterschritten.
- Die zu minimierende Funktion bleibt innerhalb einer Generation bis auf einen bestimmten Toleranzwert τ_1 konstant.
- Die zu optimierenden Parameter bleiben innerhalb einer Generation bis auf einen bestimmten Toleranzwert τ_2 konstant.
- Die Standardabweichungen fallen innerhalb einer Generation alle unter einen bestimmten Toleranzwert τ_3.
- Eine gewisse Anzahl g an Generationen ist überschritten.
- Eine gewisse Anzahl G an Geburten ist überschritten.

Diese Abbruchkriterien führen zwar zu einer Reduktion des Rechenaufwands, schließen aber nicht aus, daß man sich lediglich durch ein *enges Tal* bewegt, sich jedoch noch weit vom globalen Optimum entfernt befindet. Aufgrund dieser Tatsache schlägt Schwefel (1995) das folgende Abbruchkriterium vor: *Die zu minimierende Funktion bleibt innerhalb einer bestimmten Anzahl an Generationen* $g \leq 2n$*, wobei* n *die betrachtete Anzahl an Mutationen ist, bis auf einen gewissen Toleranzwert* τ *konstant.* Es ist zu beachten, daß EAs auch Nebenbedingungen für die Eingabeparameter berücksichtigen können.

Eine Variante der EAs sind die sogenannten *Genetischen Algorithmen (GAs)*, welche auf Goldberg (1989) zurückzuführen sind. Der wesentliche Unterschied der beiden Verfahren besteht darin, daß GAs den *Genotyp* und EAs den *Phenotyp* betrachten. Bei ersterem handelt es sich um das Erbbild eines Organismus und bei letzterem um das sich daraus ergebende Erscheinungsbild. Die Eingabeparameter bei GAs sind *Bit-Strings*, welche die genetische Information repräsentieren. Der Hauptvorteil besteht darin, daß kleinere Änderungen im Genotyp große Auswirkungen auf den Phenotyp haben, so daß viele verschiedene Phenotypen auftreten, was wiederum zu einer ausgiebigen Abtastung des Suchraums und somit zu guten globalen Kon-

vergenzeigenschaften führt. Bei lokalen Verfeinerungen erweist sich dies jedoch als erheblicher Nachteil, da die Suche in der Nachbarschaft der aktuellen Individuen einer Population nicht fokussiert werden kann. Lokale Verfeinerungen in Phenotyp sind im Genotyp kaum wahrzunehmen. Im Bereich Kraftfeldoptimierung sind GAs bereits erfolgreich zur Anpassung von simulierten an experimentell bestimmte Geometrien beziehungsweise Molekülkonfigurationen eingesetzt worden (siehe zum Beispiel Hunger u. Huttner (1999) für Metallkomplexe). Auch von Barnes u. Gelb (2007) wurden hierzu EAs verwendet, angewandt auf wäßrige Silikatsysteme. Dabei wurden verschiedene Arten von Populationen und auch Rekombinations-, Mutations- sowie Selektionsmethoden getestet. Dabei haben sich folgende Methoden als die besten herausgestellt:

1. **Population:** Nicht-überlappende Strategien mit einem kleinen Eltern-Kind-Verhältnis $\frac{\mu}{\lambda}$ haben sich als effizienter im Vergleich zu anderen Methoden erwiesen. In Barnes u. Gelb (2007) wurden $\mu = 8$ und $\lambda = 96$ empfohlen.

2. **Rekombination:** Globale intermediäre Operatoren waren deutlich effizienter als lokale diskrete Operatoren. Dies gilt insbesondere für die ersten 250 Generationen.

3. **Mutation:** Erst nach mehr als 1000 Generationen waren die verschiedenen Mutationsalgorithmen voneinander unterscheidbar. Innerhalb der ersten 1000 Generationen sollte die Methode nach Beyer u. Schwefel (2002) verwendet werden. Danach erwiesen sich diejenigen Methoden als effizienter, die mehrere Generationen berücksichtigen, was jedoch auch zu einer erheblichen Erhöhung des Rechenaufwands führt.

4. **Selektion:** Deterministische Methoden führten zu einer schnelleren Verringerung der zu minimierenden Fehlerfunktion als stochastische. Semiüberlappende Methoden stellten sich als am effizientesten heraus.

Auch eine allgemeine Aussage bezüglich EAs wird durch die Studien von Barnes u. Gelb (2007) bestätigt: Innerhalb einer gewissen Anzahl an Generationen (in der Studie 250) verringert sich der Wert der zu minimierenden Fehlerfunktion um mehrere Größenordnungen (in der Studie von $\propto 10^6$ auf $\propto 10^2$). Man kann davon ausgehen, daß der Algorithmus sich dann in der Nähe eines globalen Minimums befindet. Innerhalb der ersten Generationen dominiert die Rekombination, wohingegen danach die Mutation dominiert. Lokale Verfeinerungen werden allerdings nur langsam und mit erheblichem Rechenaufwand erzielt. Es empfiehlt sich somit stets, globale Optimierungsverfahren wie SA, EA oder GA vorzuschalten, so daß lokale Minimierer gegen ein globales Optimum konvergieren.
Eine derartige Kombination ist effizienter als die alleinige Verwendung von globalen Verfahren, da lokale Verfahren in der Nähe des Minimums deutlich schneller konvergieren als globale. Dies wurde auch im Rahmen einer anderen Applikation bestätigt, wo ein *Recursive-Random-Search*-Algorithmus zur Optimierung des Dampfdrucks und der Siededichte als globales Verfahren vorgeschaltet wurde, um geeignete Startwerte für einen gradientenbasierten lokalen Optimierer (Levenberg-Marquardt (Roweis, 1996)) zu finden (Elliott, 2011). Da hier allerdings keine Molekularen Simulationen, sondern approximative analytische Modelle für Dampfdruck und Siededichte in Abhängigkeit von den LJ-Parametern verwendet wurden, konnten mehrere tausend Funktionsauswertungen bei der globalen Voroptimierung in Kauf genommen werden, was bei der vorliegenden Problemstellung nicht der Fall ist. Hier sind somit effizientere Algorithmen zu verwenden.

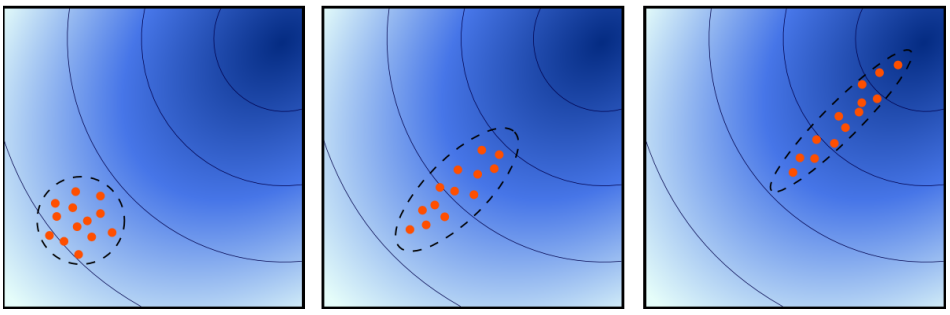


Abbildung 3.5: Suchmechanismus von CMA-ES (Wikipedia, 2011a): Durch die Aufdatierung der Kovarianzmatrix und die Schrittweitensteuerung liegen die Individuen innerhalb von Ellipsoiden. Die aktuelle Suchrichtung entspricht dem breiten Durchmesser des aktuellen Ellipsoids.

3.3.2 Evolutionärer Algorithmus mit Adaption einer Kovarianzmatrix: CMA-ES

Die hier zur globalen Optimierung eingesetzte **C***ovariance* **M***atrix* **A***daptation* **E***volution* **S***trategy (CMA-ES)* geht auf Hansen u. Ostermeier (2001) zurück und ist ein spezieller evolutionärer Algorithmus zur globalen Optimierung nichtlinearer, nichtkonvexer Funktionen. Auch bei diesem Verfahren werden per Zufallsprinzip Populationen innerhalb einer Generation generiert. Hierbei sind die möglichen Lösungen multivariat normalverteilt. Rekombination erfolgt durch die Berechnung eines gleitenden Mittelwerts über die μ besten Individuen. Eine neue Iteration innerhalb einer neuen Generation wird durch die Aufdatierung einer Kovarianzmatrix und einer Schrittweitensteuerung erhalten. Dadurch liegen die Individuen nach und nach innerhalb von Ellipsoiden, deren breiter Durchmesser der aktuellen Suchrichtung entspricht (siehe Abbildung 3.5). Die Kovarianzmatrix ist vergleichbar mit der inversen Hesse-Matrix eines Newton- oder Quasi-Newton-Verfahrens. Eine weitere Besonderheit von CMA-ES liegt in der Speicherung von zwei verschiedenen Evolutionspfaden, welche wichtige Informationen über die Korrelation zweier aufeinanderfolgender Iterationsschritte enthalten. Ein Pfad sorgt für die Aufdatierung der Kovarianzmatrix und somit für die Berechnung einer effizienten Suchrichtung. Der zweite Pfad wird für eine effiziente Schrittweitensteuerung verwendet, um schnelle Konvergenz in die Nähe eines globalen Optimums zu realisieren.

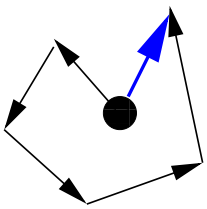
Standardmäßig wird eine $\left(\frac{\mu}{\mu_w}, \lambda\right)$-CMA-ES betrachtet, wobei

$$\mu_w := \left(\sum_{i=1}^{\mu} w_i\right)^{-1} \tag{3.12}$$

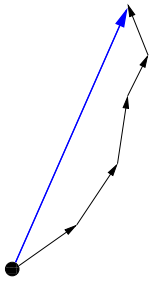
aus Gewichten $w_1, ..., w_\mu \geq 0$ für die Ermittlung eines gleitenden Mittelwerts einer Generation berechnet wird. Es gilt dabei $\sum_{i=1}^{\mu} w_i = 1$. Innerhalb einer Generation g ist

$$m^{(g)} := \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g)} \tag{3.13}$$

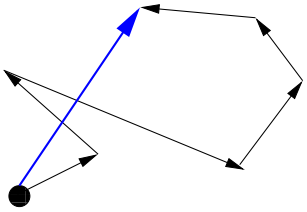
der gleitende Mittelwert über die μ besten Individuen $x_{i:\lambda},\ i = 1, ..., \mu$, von Generation g, bestehend aus λ Individuen. Dabei sind die $x_{i:\lambda},\ i = 1, ..., \lambda$, so geordnet, daß $F(x_{1:\lambda}) \leq F(x_{2:\lambda}) \leq ... \leq F(x_{\mu:\lambda}) \leq F(x_{\mu+1:\lambda}) \leq ... \leq F(x_{\lambda:\lambda})$. Je größer die Populationsgröße λ gewählt wird,



(a) Kurzer Evolutionspfad: Die Einzelpfade sind antikorreliert. Die Schrittweite wird verkleinert.



(b) Langer Evolutionspfad: Die Einzelpfade sind korreliert. Die Schrittweite wird vergrößert.



(c) Idealfall: Die Einzelpfade sind unkorreliert. Die Schrittweite wird beibehalten.

Abbildung 3.6: Evolutionspfade bei CMA-ES: antikorrelierte (a), korrelierte (b) und unkorrelierte (c) Einzelpfade und deren Auswirkungen auf die Schrittweite. Die jeweils resultierenden Schritte sind mit blauen Pfeilen markiert.

desto verstreuter kann der Algoritmus im Parameterraum suchen und desto mehr Individuen können pro Generation überleben. Das bedeutet, daß CMA-ES mit steigendem λ schneller, das heißt mit weniger Iterationen, in die Nähe eines globalen Minimums gelangt. In Hansen u. Ostermeier (2001) wird $\mu \leq \frac{\lambda}{2}$ und $\mu_w \approx \frac{\lambda}{4}$ empfohlen.

Es seien im folgenden $\sigma^{(g)} > 0$ eine Schrittweite und $C^{(g)}$ eine $N \times N$-Kovarianzmatrix, deren Einträge Korrelationen innerhalb der Population beschreiben. Es gilt $C^{(0)} = I$. Die Individuen $x_i \in \mathbb{R}^N$, $i = 1, ..., \lambda$, werden aus einer multivariaten Normalverteilung per Zufallsprinzip gewählt. Es gilt

$$x_i \sim \mathcal{N}\left(m^{(g)}, \left(\sigma^{(g)}\right)^2 C^{(g)}\right) \sim m^{(g)} + \sigma^{(g)} \mathcal{N}\left(0, C^{(g)}\right), \; i = 1, ..., \lambda, \tag{3.14}$$

wobei $\mathcal{N}$ wieder für die Normalverteilung steht. Es werden sowohl die Schrittweite $\sigma^{(g)}$ als auch die Kovarianzmatrix $C^{(g)}$ mithilfe von sogenannten *Evolutionspfaden* aufdatiert. Die innerhalb von CMA-ES verwendete Schrittweitensteuerung ist auch bekannt unter dem Namen **C***umulative* **S***tep Length* **A***daptation (CSA)*. Die Länge des aktuellen Evolutionspfads für die Schrittweite ist die Summe beziehungsweise die Bilanz der bislang ermittelten Einzelpfade. Abbildung 3.6 zeigt drei mögliche Fälle:

1. Ist der Evolutionspfad kurz (Abbildung 3.6(a)), das heißt, heben sich die Einzelpfade bei Aufsummierung gegenseitig auf, so sind diese antikorreliert, und die Schrittweite sollte verkleinert werden, da in diesem Fall nur noch lokale Verfeinerungen vonnöten sind.
2. Ist der Evolutionspfad lang (Abbildung 3.6(b)), so sind die Einzelpfade korreliert, und die Schrittweite sollte vergrößert werden, da hierbei viele kleine Einzelschritte durch einen einzigen langen Schritt ersetzt werden können.
3. Im idealen Fall (Abbildung 3.6(c)) sind die Einzelpfade unkorreliert. Dann kann die aktuelle Schrittweite beibehalten werden.

Um zu entscheiden, ob ein Evolutionspfad kurz oder lang ist, wird seine tatsächliche Länge mit seiner *erwarteten Länge bei zufälliger Selektion* verglichen. Der Evolutionspfad p_σ für die

Schrittweitensteuerung ist gemäß Hansen u. Ostermeier (2001) durch die Iterationsvorschrift

$$p_\sigma^{(g+1)} := (1-c_\sigma)p_\sigma^{(g)} + \sqrt{1-(1-c_\sigma)^2}\left(C^{(g)}\right)^{-\frac{1}{2}} \frac{m^{(g+1)}-m^{(g)}}{\sigma^{(g)}} \tag{3.15}$$

gegeben, wobei $p_\sigma^{(0)} := 0$. Dabei gibt $c_\sigma^{-1} \approx \frac{N}{3}$ an, wie viele bisher durchgeführte Iterationen in den Evolutionspfad miteinbezogen werden. Die Schrittweite wird nun gemäß

$$\sigma^{(g+1)} := \sigma^{(g)} \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\frac{||p_\sigma||}{E||\mathcal{N}(0,1)||-1}\right) \tag{3.16}$$

aufdatiert. Dabei sind d_σ ein Dämpfungsparameter, und für den angegebenen Erwartungswert gilt $E||\mathcal{N}(0,1)|| = \sqrt{2}\Gamma\left(\frac{N+1}{2}\right)/\Gamma\left(\frac{N}{2}\right)$. Durch Gleichung (3.16) definiert die Schrittweite annähernd $\left(C^{(g)}\right)^{-1}$-konjugierte Differenzen von aufeinanderfolgenden Mittwelwerten, das heißt, es gilt

$$\forall_{g\geq 1} \quad \left(m^{(g+1)}-m^{(g)}\right)^T \left(C^{(g)}\right)^{-1} \left(m^{(g)}-m^{(g-1)}\right) \approx 0. \tag{3.17}$$

Auch die Aufdatierung der Kovarianzmatrix geschieht über einen Evolutionspfad. Dieser wird gemäß

$$p_C^{(g+1)} := (1-c_C)p_C^{(g)} + 1_{[0,\alpha\sqrt{N}]}(||p_\sigma||)\sqrt{1-(1-c_C)^2}\sqrt{\mu_w}\frac{m^{(g+1)}-m^{(g)}}{\sigma^g} \tag{3.18}$$

festgesetzt. Dabei hat c_C dieselbe Bedeutung wie c_σ in Gleichung (3.16). Hierbei wird jedoch von Hansen u. Ostermeier (2001) $c_C^{-1} \approx \frac{N}{4}$ empfohlen. Weiterhin gilt $\alpha \approx 1.5$, was zur Folge hat, daß p_C nur dann aufdatiert wird, wenn p_σ nicht allzu lang ist. Bei korrelierten Einzelpfaden von p_σ muß die Suchrichtung selbst nicht geändert werden, das heißt, das Ellipsoid braucht nicht in andere Richtungen verzerrt zu werden.
Die Kovarianzmatrix unterliegt sowohl einer *Rang-1-* als auch einer *Rang-μ-*Aufdatierung. Erstere ergibt sich aus einer gewichteten Summe von einer und letztere als gewichtete Summe von μ Rang-1-Matrizen. Die Rang-μ-Aufdatierung führt zu einer Matrix vom Rang $\min(\mu, N)$. Details hierzu findet man in Hansen u. Ostermeier (2001). Der Parameter c_1 sei dabei das Gewicht der Rang-1-Aufdatierung und c_μ das Gewicht der Rang-μ-Aufdatierung. Beides sind sogenannte *Lernraten*, und es wird $c_1 \approx \frac{2}{N^2}$ sowie $c_\mu \approx \frac{\mu_w}{N^2}$ empfohlen. In jedem Fall muß die Bedingung $c_\mu \geq 1 - c_1$ erfüllt sein. Dann wird die Kovarianzmatrix gemäß

$$C^{(g+1)} := (1-c_1-c_\mu)C^{(g)} + c_1 p_C p_C^T + c_\mu \sum_{i=1}^{\mu} w_i \frac{x_{i:\lambda}^{(g)}-m^{(g)}}{\sigma^{(g)}}\left(\frac{x_{i:\lambda}^{(g)}-m^{(g)}}{\sigma^{(g)}}\right)^T \tag{3.19}$$

aufdatiert.
Die meisten Parameter innerhalb von CMA-ES sind festgesetzt. Eine wichtige vom Benutzer festzulegende Variable ist die Populationsgröße λ, da diese sehr stark von der Gestalt der zu optimierenden Funktion abhängig ist. Bei großem λ ist globale Konvergenz und bei kleinem λ eher lokale Konvergenz zu erwarten. Eine hohe Populationsgröße hat zwar, wie bereits erwähnt,

eine geringere Anzahl an Iterationen zur Folge, allerdings auch eine höhere Anzahl an Funktionsauswertungen. Daher ist bei der vorliegenden Problemstellung vorher zu evaluieren, wie die Populationsgröße geeignet zu wählen ist.
Je nach Gestalt der zu minimierenden Zielfunktion und Wahl der Verfahrensparameter ist bei CMA-ES eine mehr als lineare, teilweise sogar exponentielle Konvergenz beobachtbar Hansen u. Ostermeier (1997). In dieser Arbeit wird CMA-ES als globaler Voroptimierer eingesetzt, da das Verfahren gemäß Hansen u. Ostermeier (2001) bei geeigneter Parametrisierung dazu in der Lage ist, schnell und effizient in die Nähe eines globalen Minimums zu gelangen, unabhängig von der Gestalt der Fehlerfunktion. Weiterhin ist es aufgrund der Mittelwertbildung zum Erhalt der Evolutionspfade äußerst robust in Bezug auf Rauschen und aufgrund der Verzerrung der Ellipsoide auf ein Minimum gerichtet. Ein weiterer Vorteil von CMA-ES besteht darin, daß die meistens Verfahrensparameter aus Hansen (2011) übernommen werden können und nicht problemspezifisch zu wählen sind. Ob und inwieweit CMA-ES bei der vorliegenden Problemstellung mit akzeptablem Rechenaufwand in die Nähe eines globalen Minimums gelangen kann, wird in Abschnitt 4.4 detailliert diskutiert. Verwendet wurde dazu eine Java-Implementierung, die unter der *GNU Public License (GPL)* im Internet frei verfügbar ist (siehe Anhänge G.2 und G.11 für weitere Details).

3.3.3 Erstellung eines Metamodells und multikriterielle Optimierung: DesParO

In der am Fraunhofer-Institut SCAI entwickelten **Des***ign* **Par***ameter* **O***ptimisation Toolbox (DesParO)* ist ein globales Interpolationsverfahren implementiert, welches anhand von Simulationsergebnissen, wobei die entsprechenden Eingangsmodelle zufallsbasiert sind, ein nichtlineares Metamodell erstellt und dadurch für multikriterielle Optimierungsaufgaben geeignet ist, also Pareto-Fronten bestimmen kann. DesParO wurde von Bäuerle u. a. (2004) entwickelt und basierte zunächst auf einer stochastisch basierten *Kriging*-Interpolation, siehe hierzu zum Beispiel Armstrong (1998). Später wurde es von Thole u. a. (2007) weiterentwickelt, basierend auf einer Interpolation mittels *Radialer Basisfunktionen (RBFs)*, welche in Abschnitt 6.2.2 näher erläutert werden. Hauptanwendung von DesParO waren multikriterielle Optimierungsaufgaben in der Automobilindustrie (siehe Stork u. a. (2008)), wobei das Ziel darin bestand, mittels Crash-Simulationen eine maximale Eindringung bei maximaler Steifigkeit des Blechs und gleichzeitig minimaler Masse des Fahrzeugs zu erhalten. DesParO eignet sich jedoch für jedwede Applikation, bei der ein funktionaler Zusammenhang zwischen Eingabeparameter und Zielgrößen detektiert werden soll. Es hat weiterhin folgende Eigenschaften:

- Ein sehr wichtiges Merkmal von DesParO ist die sogenannte *Robuste Toleranzvorhersage*. Das erstellte Metamodell gibt nicht nur eine Schätzung für eine bestimmte Zielgröße, sondern auch für deren Toleranzgrenzen. Jede Zielgröße $f_i,\ i = 1, ..., n$, sollte sich in einem Bereich $f_i < f_i^{\max}$ befinden, am besten jedoch jede Zielgröße zusammen mit ihrer Toleranz $\eta(f_i)$, also $f_i + \eta(f_i) < f_i^{\max}$. Zur Schätzung von $\eta(f_i)$ wird eine *Leave-One-Out*-Kreuzvalidierung verwendet, siehe hierzu beispielsweise Hülsmann (2006): Dabei wird stets ein Datenpunkt ausgelassen und ein Metamodell über die restlichen Datenpunkte erstellt. Die mittels Interpolation vorhergesagte Zielgröße für den ausgelassenen Datenpunkt wird dann mit der wahren Zielgröße verglichen. Die Differenz ist gleich der Toleranz des Metamodells an diesem Datenpunkt.

- Eine weitere wichtige Eigenschaft ist die *Sensitivitätsanalyse*: Eine Korrelationsmatrix zeigt an, wie hoch die Interpendenz zwischen einem Eingabeparameter und einer Zielgröße ist. Die Größe der Interpendenz wird dabei durch Quadrate verschiedener Größe angezeigt. Ob die betrachtete Zielgröße bei Erhöhung des Eingabeparameters sinkt oder steigt, wird durch verschiedene Farben gekennzeichnet: Blau bedeutet, daß die Zielgröße sinkt und rot, daß sie steigt. Schwarze Quadrate stehen für alternierende Veränderungen.

DesParO ist ein interaktives Tool, das heißt, der Benutzer kann mithilfe eines roten Schiebers die Eingabeparameter verändern und dann direkt die Auswirkung dieser Änderung auf sämtliche Zielgrößen sehen. Hierzu sind auf der Seite der Zielgrößen zunächst mithilfe von zwei grünen Schiebern Intervalle einzustellen, in denen sich die gewünschten Zielgrößen befinden. Dann wird die Verfügbarkeit von Lösungen als 'grüne Inseln' auf der Seite der Eingabeparameter gekennzeichnet. Durch Veränderung der Eingabeparameter auf diesen grünen Inseln mit roten Schiebern können die Auswirkungen auf sämtliche Zielgrößen beobachtet werden, da sich rote Schieber auf der Seite der Zielgrößen dadurch ebenfalls bewegen. Weiterhin kann der Benutzer ein durch die robuste Toleranzvorhersage erhaltenes Konfidenzintervall auf der Seite der Zielgröße direkt ablesen. Dieses Konfidenzintervall ist durch einen waagerechten Strich unter dem jeweiligen roten Schieber gekennzeichnet.

Werden als Zielgrößen physikalische Eigenschaften zu verschiedenen Temperaturen und als Eingabeparameter die Kraftfeldparameter verwendet, so kann DesParO direkt auf die Bestimmung optimaler Kraftfelder für Molekulare Simulationen angewandt werden. Dafür muß jedoch zunächst eine ausreichend große Anzahl an Simulationen durchgeführt werden, die aufgrund des hohen Rechenaufwandes nicht zu hoch sein sollte. Die Anzahl an Simulationen steigt gemäß Thole u. a. (2007) quadratisch mit der Anzahl an Kraftfeldparametern. Nach einer ausreichenden Anzahl an Simulationen von Zufallsmodellen, durch die die entsprechenden Zielgrößen bestimmt werden, erstellt DesParO ein Metamodell mittels Interpolation, so daß eine direkte Abhängigkeit zwischen Kraftfeldparametern und physikalischen Eigenschaften gegeben ist. Wie viele Zufallsmodelle dazu tatsächlich gewählt werden müssen, stellt in der Praxis allerdings zunächst ein Problem dar. Zur Optimierung sämtlicher Zielgrößen können die entsprechenden grünen Schalter auf der rechten Seite die experimentellen Werte eingrenzen. Auf der linken Seite erhält der Benutzer die Lösungsbereiche für die Kraftfeldparameter. Es ist jedoch zu beachten, daß es sich nach wie vor um ein multikriterielles Optimierungsproblem handelt. Eine exakte Lösung kann nicht garantiert werden. Das bedeutet, daß die Einstellung einer bestimmten Zielgröße auf ihren experimentellen Wert durch Verschiebung der roten Schalter auf der linken Seite zur Folge haben kann, daß der Lösungsbereich derart eingeschränkt wird, daß andere Zielgrößen nicht mehr auf ihre experimentellen Werte gesetzt werden können. In diesem Fall muß der Benutzer nach einem Kompromiß suchen, so daß sämtliche experimentelle Eigenschaften zwar nicht exakt, aber möglichst genau vorhergesagt werden. Abbildung 3.7(a) zeigt dies anhand der Siededichten und Verdampfungsenthalpien zu verschiedenen Temperaturen von Ethylenoxid (Maaß u. a., 2010). Betrachtet wurden dabei die LJ-Parameter von Kohlenstoff und Sauerstoff. Die Verdampfungsenthalpie zu $T = 375$ K befindet sich zwar annähernd im Zentrum des durch die grünen Schalter festgesetzten Intervalls, dafür befindet sich die Siededichte zu $T = 260$ K am linken Rand. Die Auswirkungen der LJ-Parameter auf die Zielgrößen sind in Abbildung 3.7(b) durch die DesParO-Korrelationsmatrix dargestellt.

(a) Interaktive Eigenschaft (rote und grüne Schieber) von DesParO: Der Benutzer kann durch zwei grüne Schieber auf der Seite der Zielgrößen Intervalle festesetzen, in denen sich die jeweiligen gewünschten Zielgrößen befinden. Durch die Bewegung der roten Schieber auf der Seite der Eingabeparameter kann die Auswirkung dieser Änderungen auf sämtliche Zielgrößen beobachtet werden. Bei $T = 400$ K sind die Konfidenzintervalle ziemlich groß, was durch die roten Striche unter den roten Schiebern gekennzeichnet ist und für hohe statistische Ungenauigkeiten steht.

(b) Korrelationsmatrix von DesParO: Die Zeilen sind die Zielgrößen und die Spalten die Eingabeparameter. Die Größe der Quadrate gibt an, in welchem Ausmaß sich die Änderung eines Eingabewerts auf eine Zielgröße auswirkt. Blau bedeutet, daß sie ansteigt und rot, daß sie abfällt. Schwarze Quadrate stehen für alternierende Veränderungen.

Abbildung 3.7: Interaktive Eigenschaft, das heißt rote und grüne Schieber, (a) und Korrelationsmatrix (b) von DesParO. Betrachtet wurden die LJ-Parameter ε (e) und σ (s) des Kohlenstoffs (c) sowie Sauerstoffs (o) von Ethylenoxid und deren Auswirkungen auf Siededichte (d) und Verdampfungsenthalpie (h) zu den Temperaturen 260, 300, 330, 375 und 400 K.

Das Verfahren ist für die Anwendung auf Molekulare Simulationen wie folgt zu bewerten: DesParO muß stets auf einen ausreichend großen Eingabeparameterbereich angewandt werden. Da die Anzahl an Molekularen Simulationen aufgrund des hohen Rechenaufwands gering gehalten werden soll und die einzelnen Funktionen f_i auf einem großen zulässigen Gebiet sehr komplex sein können, kann es sein, daß das Metamodell den funktionalen Zusammenhang nicht akkurat wiedergibt. Hinzu kommt die Tatsache, daß die physikalischen Eigenschaften mit statistischem Rauschen behaftet sind. Da das Metamodell auf Interpolation beruht und somit die Trainingsdaten exakt vorhergesagt werden, wird auch jede statistische Unsicherheit im Modell mitberücksichtigt, was das Modell noch ungenauer macht. Aufgrund der Sensitivitätsanalyse eignet sich DesParO jedoch äußerst gut, um die Auswirkungen bestimmter Kraftfeldparameter auf bestimmte Zielgrößen zu beschreiben. Somit kann beispielsweise eine *Merkmalsselektion (Feature Selection)* durchgeführt werden. Falls es einen oder mehrere Kraftfeldparameter gibt, die sich gar nicht oder nur geringfügig auf die Zielgrößen auswirken, brauchen sie nicht optimiert zu werden. Weiterhin ergibt sich durch DesParO die Möglichkeit der *sukzessiven Optimierung*, das heißt, es werden zunächst die einflußreichsten Kraftfeldparameter angepaßt, diese in ei-

nem nächsten Schritt festgehalten und nur noch diejenigen Parameter angepaßt, die sich nur geringfügig auf die Zielgrößen auswirken. Es sei jedoch hervorgehoben, daß das Ergebnis von DesParO aufgrund der oben genannten Tatsachen keinesfalls als optimal angesehen werden kann. Die resultierenden Kraftfeldparameter eignen sich lediglich als Startparameter für einen nachfolgenden Optimierungsablauf.

3.4 Gradientenbasierte Verfahren zur Parameteroptimierung

Die in Abschnitt 3.2.2 vorgestellte Methode von Ungerer und Bourasseau ist durch viele Nachteile geprägt, liefert jedoch einen praktischen Beweis dafür, daß gradientenbasierte Verfahren grundsätzlich auf die vorliegende Problemstellung anwendbar sind. Ein Ziel dieser Arbeit besteht darin, effiziente gradientenbasierte numerische Optimierungsalgorithmen mit äußerst guten Konvergenzeigenschaften zu verwenden. Aufgrund der in jedem Iterationsschritt durchzuführenden Molekularen Simulationen sollen diese mit möglichst wenigen Iterationen zu einem lokalen Optimum gelangen. Dabei bleibt das Ergebnis der Optimierung startwertabhängig, was in Abschnitt 3.5.1 noch eingehender diskutiert wird. Weiterhin haben die hier verwendeten Methoden den Vorteil, daß in jeder Iteration bessere Kraftfeldparameter erhalten werden und daß sie gerichtet sind, sich also stets in Richtung des Minimums bewegen. Außerdem sind die Algorithmen sowohl auf über- als auch auf unterbestimmte Optimierungsprobleme anwendbar. Gradientenbasierte Verfahren sind in *Abstiegs-* und *Trust-Region-Verfahren* einzuteilen (siehe zum Beispiel Nocedal u. Wright (1999) und Press u. a. (1992)). In dieser Arbeit werden alle bekannten gradientenbasierten Verfahren in Betracht gezogen. Aufgrund der statistischen Unsicherheiten werden in diesem Abschnitt keine komplexeren Verfahren verwendet, da eine Erhöhung der Genauigkeit in diesem Fall keinen Nutzen mit sich bringen würde. Abstiegsverfahren werden in Abschnitt 3.4.1 eingeführt und in Abschnitt 3.4.2 mit einer geeigneten Schrittweitensteuerung versehen. Gradientenbasierte Trust-Region-Verfahren werden in Abschnitt 3.4.3 vorgestellt.

3.4.1 Abstiegsverfahren

Sämtliche Abstiegsverfahren lassen sich durch die Iterationsvorschrift

$$x^{k+1} = x^k + t_k d^k, \tag{3.20}$$

wobei d^k die sogenannte *Abstiegsrichtung* ist, welche in Richtung des Minimums zeigen muß. Mathematisch ausgedrückt bedeutet dies

$$\langle \nabla F(x^k), d^k \rangle < 0. \tag{3.21}$$

Die Schrittweite t_k legt fest, wie weit der Abstiegsrichtung gefolgt wird. Näheres zur Schrittweitensteuerung wird in den Abschnitten 3.4.2 und 3.5.1 erläutert. Sämtliche Abstiegsverfahren unterscheiden sich nur in Bezug auf die Wahl der Abstiegsrichtung d^k. Einige der hier betrachteten Methoden konvergieren q-überlinear:

Definition 3.4.1 (q-überlineare Konvergenz)**.** *Ein iteratives Verfahren konvergiert q-überlinear gegen $x^{\text{opt}} \in \mathbb{R}^N$, falls*

$$\forall_{k\geq 0}\ ||x^{k+1} - x^{\text{opt}}|| \leq c_k ||x^k - x^{\text{opt}}||, \tag{3.22}$$

wobei $(c_k)_{k\in\mathbb{N}}$ eine Nullfolge ist.

Im folgenden werden die verwendeten Abstiegsverfahren im einzelnen aufgezählt:

- **Verfahren des steilsten Abstiegs:** $d^k = -\nabla F(x^k)$. Dabei zeigt d^k in Richtung des steilsten Abstiegs von F. Es handelt sich hierbei um ein heuristisches Verfahren, dessen Konvergenz nur dann garantiert ist, wenn die Iterationsfolge $(F(x^k))_{k\in\mathbb{N}}$ streng monoton fallend ist. Dann ist jeder Häufungspunkt von $(x^k)_{k\in\mathbb{N}}$ ein stationärer Punkt von F.
- **Newton-Raphson-Verfahren:** $d^k = -(D^2F(x^k))^{-1}\nabla F(x^k)$. Es handelt sich hierbei um die mehrdimensionale Erweiterung des eindimensionalen Newton-Verfahrens zur Nullstellenbestimmung. Der Vorteil ist, daß es zusätzlich zur Steigung auch Krümmungseigenschaften von F miteinbezieht. Die Newton-Raphson-Richtung ist jedoch genau dann eine Abstiegsrichtung, wenn die Hesse-Matrix $D^2(F(x^k))$ spd ist. Symmetrisch ist sie nach dem Satz von Schwarz, allerdings ist die Eigenschaft der positiven Definitheit oftmals nicht erfüllt. Das Newton-Raphson-Verfahren eignet sich zumeist nur in der Nähe des Minimums, konvergiert dann jedoch äußerst schnell: Falls D^2F in einer Umgebung des Minimums Lipschitz-stetig ist, so konvergiert das Verfahren quadratisch gegen x^{opt}. Daher wird der höhere Rechenaufwand, welcher durch die Bestimmung der Hesse-Matrix entsteht, durch die höhere Konvergenzgeschwindigkeit ausgeglichen. Allerdings kann D^2F in der Praxis auch singulär sein. Falls dies der Fall ist oder der Abstieg nicht ausreichend klein, das heißt falls
  $$\langle \nabla F(x^k), d^k \rangle > -\rho ||d^k||^p \tag{3.23}$$
  für ein $\rho > 0$ und ein $p \in \mathbb{N}^{>2}$, so wird wie beim Verfahren des Steilsten Abstiegs der negative Gradient verwendet. In der Praxis sind stets zwei Aspekte zu berücksichtigen:
  1. Die Hesse-Matrix muß in einer angemessenen Rechenzeit bestimmbar sein.
  2. Die Hesse-Matrix muß robust in Bezug auf Rauschen sein. Nähres hierzu wird in Abschnitt 3.5.6 diskutiert.
- **Quasi-Newton-Verfahren:** $d^k = -H_k^{-1}\nabla F(x^k)$. Dabei ist H_k eine Approximation der Hesse-Matrix $D^2F(x^k)$. Ein Quasi-Newton-Verfahren kommt zumeist dann zum Einsatz, falls einer der beiden oben genannten Punkte in Bezug auf die Hesse-Matrix nicht erfüllt ist. Da jedoch die Bestimmung von H_k stets iterativ erfolgt, das heißt $H_k \to H_{k+1}$ mit $H_0 = D^2F(x^0)$, sind Quasi-Newton-Verfahren stets startwertabhängig und somit auch nur in der Nähe des Optimums anwendbar. Ist jedoch H_k spd, so ist auch H_{k+1} spd. Letzteres gilt nach dem Theorem von Dennis und Moré, allerdings muß hierfür die sogenannte *Sekantenbedingung*
  $$H_{k+1} \underbrace{\left(x^{k+1} - x^k\right)}_{:=s_k} = \underbrace{\nabla F(x^{k+1}) - \nabla F(x^k)}_{:=y_k} \tag{3.24}$$
  erfüllt sein.

Die drei hier betrachteten Quasi-Newton-Verfahren erfüllen diese Bedingung und unterscheiden sich lediglich in der Aufdatierung von H_k:

- *Powell Symmetric Broyden (PSB):*

$$H_{k+1}^{\mathrm{PSB}} := H_k + \frac{1}{\langle s_k, s_k\rangle}\left((y_k - H_k s_k)s_k^T + s_k(y_k - H_k s_k)^T\right) - \frac{(y_k - H_k s_k)^T s_k}{\langle s_k, s_k\rangle^2} s_k s_k^T. \tag{3.25}$$

  Mit dieser Aufdatierung ist die Sekantenbedingung $H_{k+1}^{\mathrm{PSB}} s_k = y_k$ automatisch erfüllt.
  Falls F in einer Umgebung des Minimums Lipschitz-stetig ist und $\sum_{k=0}^{\infty} ||x^k - x^{\mathrm{opt}}|| < \infty$, so ist die Konvergenz q-überlinear.
- *Davidon Fletcher Powell (DFP):*

$$H_{k+1}^{\mathrm{DFP}} := H_k + \frac{1}{\langle y_k, s_k\rangle}\left((y_k - H_k s_k)y_k^T + y_k(y_k - H_k s_k)^T\right) - \frac{(y_k - H_k s_k)^T s_k}{\langle y_k, s_k\rangle^2} y_k y_k^T. \tag{3.26}$$

  Mit dieser Aufdatierung ist die Sekantenbedingung $H_{k+1}^{\mathrm{DFP}} s_k = y_k$ automatisch erfüllt. Die Konvergenz ist unter denselben Voraussetzungen wie beim PSB-Verfahren q-überlinear.
- *Broyden Fletcher Goldfarb Shanno (BFGS):* Hierbei wird eine Approximation der inversen Hesse-Matrix $B_k \to B_{k+1}$ aufdatiert, wobei $B_0 = (D^2F(x^0))^{-1}$:

$$B_{k+1}^{\mathrm{BFGS}} := B_k + \frac{1}{\langle y_k, s_k\rangle}\left((s_k - B_k y_k)s_k^T + s_k(s_k - B_k y_k)^T\right) - \frac{(s_k - B_k y_k)^T y_k}{\langle y_k, s_k\rangle^2} s_k s_k^T. \tag{3.27}$$

  Mit dieser Aufdatierung ist die Sekantenbedingung $B_{k+1}^{\mathrm{BFGS}} y_k = s_k$ automatisch erfüllt.
  Dies hat zwar den Nachteil, daß im ersten Schritt die Inverse der Hesse-Matrix bestimmt werden muß, die nächsten Schritte jedoch nur noch aus Matrix-Vektor-Multiplikationen erhalten werden und nicht mehr aus der Lösung eines LGS. Falls F in einer Umgebung des Minimums Lipschitz-stetig ist und für alle $k \geq 0$ die Frobenius-Norm der Hesse-Matrix beschränkt ist, konvergiert das BFGS-Verfahren ebenfalls q-überlinear.

- **Verfahren der Konjugierten Gradienten (CG-Verfahren):** $d^{k+1} = -\nabla F(x^{k+1}) + \beta_k d^k$, wobei $d^0 = -\nabla F(x^0)$. Die Abstiegsrichtung wird in jedem Schritt mithilfe des neuen Gradienten aufdatiert. Dieser muß somit vorher berechnet werden. Die Methode ist ein simpler Transfer des CG-Verfahrens für LGS, allerdings mit neuen Konvergenzbeweisen. Beim Standard-CG-Verfahren sind die Residuen, das heißt, die negativen Gradienten einer zu minimierenden quadratischen Form, konjugiert, also

$$\forall_{k\leq 0}\ \langle \nabla F(x^{k+1}), \nabla F(x^k)\rangle = 0, \tag{3.28}$$

  was dazu führt, daß es sich im Prinzip um ein direktes Verfahren handelt. Es ist lediglich aufgrund von Rundungsfehlern ein iteratives Verfahren. Im mehrdimensionalen Fall, das heißt bezogen auf nichtlineare Optimierungsprobleme, ist die Konjugiertheit jedoch nicht mehr gegeben.

Die zwei bekanntesten CG-Verfahren, welche sich nur in der Wahl von β_k unterscheiden, werden hier betrachtet:

- *Fletcher-Reeves-Verfahren:* $\beta_k^{\mathrm{FR}} = ||\nabla F(x^{k+1})||^2 / ||\nabla F(x^k)||^2$. Der Ansatz besteht darin, die Gradienten zunächst als konjugiert zu betrachten, das heißt, es handelt sich um einen direkten Transfer des CG-Verfahrens für LGS. Falls die Levelmenge $\mathcal{L}(x^0) := \{x | F(x) < F(x^0)\}$ kompakt ist und F gleichmäßig konvex auf $\mathcal{L}(x^0)$ ist, konvergiert das Verfahren gegen das eindeutig bestimmte globale Minimum von F. Zur lokalen Konvergenz ist der zulässige Bereich für x einzuschränken.
- *Polak-Ribière-Verfahren:* $\beta_k^{\mathrm{PR}} = \langle \nabla F(x^{k+1}) - \nabla F(x^k), \nabla F(x^{k+1}) \rangle / ||\nabla F(x^k)||^2$. Sind die Gradienten konjugiert, so gilt $\beta_k^{\mathrm{FR}} = \beta_k^{\mathrm{PR}}$. Da dies nicht erfüllt ist, spricht zunächst beim Transfer nichts dagegen, β_k^{PR} anstatt β_k^{FR} zu verwenden. Diese heuristische Betrachtung führt zu einem weiteren CG-Verfahren mit guten Konvergenzeigenschaften: Ist die Levelmenge kompakt und ∇F in einer Kugel, welche die Levelmenge enthält, Lipschitz-stetig, so ist die Konvergenz gegen ein globales Minimum garantiert. Zur lokalen Konvergenz ist der zulässige Bereich für x einzuschränken.

Die Konvergenz der CG-Verfahren läßt sich nur dann beweisen, wenn bestimmte funktionsangepaßte Schrittweiten t_k verwendet werden. Diese sind allerdings in der Praxis entweder nur durch hohen Rechenaufwand oder gar nicht erhältlich. Daher wird in dieser Arbeit von der Verwendung derartiger Schrittweiten abgesehen, obwohl die Konvergenz dann theoretisch nicht mehr garantiert ist.

Sämtliche Abstiegsverfahren haben eine gemeinsame Voraussetzung: Sie sind abhängig von den Startwerten, das heißt x^0 muß sich in einer Umgebung von x^{opt} befinden. Weiterhin muß sich bei den Quasi-Newton-Verfahren die Hesse-Approximation H_0 bezüglich einer Matrixnorm, beispielsweise der Frobenius-Norm, in der Nähe von $D^2F(x^0)$ befinden. Da es kein allgemeines Verfahren zur Bestimmung eines geeigneten Startwerts gibt, ist dies das größte Problem bei der Anwendung derartiger Verfahren. Weiterhin muß die Existenz eines lokalen beziehungsweise globalen Optimums gewährleistet sein, das heißt es muß ein x^{opt} existieren mit

$$\nabla F(x^{\mathrm{opt}}) = 0, \quad D^2F(x^{\mathrm{opt}}) \text{ spd.} \tag{3.29}$$

Die Eindeutigkeit des Minimums kann durch die Einschränkung des zulässigen Bereichs erzielt werden. Bei Verfahren, die eine Hesse-Matrix verwenden, muß die Funktion F zumindest in der Nähe des Minimums konvex sein, denn dann ist die Hesse-Matrix spd. In den folgenden Kapiteln wird deutlich gemacht, inwiefern die positive Definitheit der Hesse-Matrix von ausschlaggebender Bedeutung zum Erfolg der Verfahren ist.

3.4.2 Schrittweitensteuerung

Eine funktionangepaßte Schrittweitensteuerung ist für jedes numerische Optimierungsverfahren unentbehrlich. In steilen Bereichen der Funktion sind große Schritte zu bevorzugen, und in flachen Bereichen, das heißt in einer Umgebung des Minimums, kleine Schritte, so daß das Minimum nicht verfehlt wird und der Algorithmus um das Minimum herumspringt. Die Schrittweitensteuerung sollte so aufgebaut sein, daß die Schrittweite in jedem Schritt kleiner wird, um letzteres ebenfalls zu vermeiden. Wegen der Iterationsvorschrift (3.20) bestimmt sowohl die

Schrittweite t_k als auch die Norm der Abstiegsrichtung $||d^k||$ die Länge des Iterationsschritts. Ist $||d^k||$ groß, so sollte t_k klein sein und umgekehrt. Insgesamt sollte $t_k||d^k||$ in steilen Funktionsbereichen groß und in flachen Berechen klein sein. Um eine langsame Konvergenz der Schrittweitensteuerung bereits in den ersten Iterationsschritten zu vermeiden, wird anstatt der Iterationsvorschrift (3.20) die folgende Iterationsvorschrift verwendet:

$$x^{k+1} = x^k + t_k \frac{d^k}{||d^k||}. \tag{3.30}$$

Somit kontrolliert nur die Schrittweite t_k die Länge des Iterationsschritts, und zu kleine t_k, die bereits zu Beginn des Iterationsverfahrens auftreten, werden vermieden. Da t_k nicht zu klein werden kann, ist auch eine langsame Konvergenz der Schrittweitensteuerung zu Beginn ausgeschlossen. Die Vorschrift (3.30) wird jedoch nur dann verwendet, wenn $||\nabla F|| > 1$, denn ansonsten ist die Konvergenz des Iterationsverfahrens nicht mehr garantiert, da so in einer Umgebung des Minimums genügend große t_k nicht mehr gewährleistet sind.
Die Schrittweitensteuerung sollte ebenfalls dafür sorgen, daß die Monotonieeigenschaft

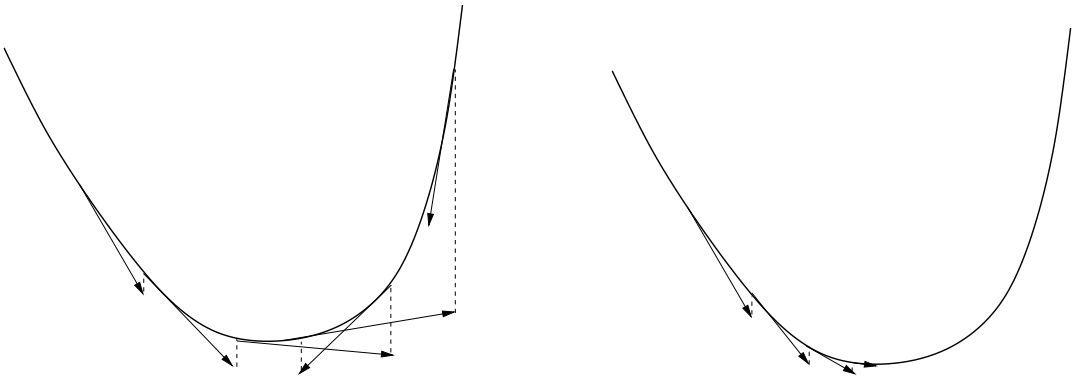
$$F(x^{k+1}) < F(x^k) \tag{3.31}$$

erfüllt ist. Um ein auf das Minimum gerichtetes iteratives Verfahren zu realisieren, ist diese Bedingung von entscheidender Bedeutung.
Die einfachste Möglichkeit besteht darin, eine geeignete Schrittweite *durch Probieren* zu finden. Es wird dabei von der heuristischen Überlegung ausgegangen, große t_k in steilen und kleine t_k in flachen Bereichen zu verwenden. Das bedeutet, daß diese sogenannte *heuristische* Schrittweite t_H ab einem bestimmten Schwellenwert $\vartheta > 0$ mit $F(x^{k_0}) < \vartheta$ oder $||\nabla F(x^{k_0})|| < \vartheta$ deutlich verkleinert wird, wobei $k_0 >> 1$. Vorstellbar sind auch mehrere Schwellenwerte $\vartheta_1, ..., \vartheta_p,\ p \in \mathbb{N}$. Dies ist jedoch bei rechenaufwendigen Funktionsauswertungen wie bei der vorliegenden Problemstellung undenkbar. Außerdem wird der Algorithmus in flacheren Bereichen um das Minimum herumspringen und größere Fehlerfunktionswerte liefern. Eine heuristische Schrittweite ist nur dann von Interesse, wenn die Funktion derart zerklüftet ist, daß irrelevante intermediäre lokale Minima übersprungen werden sollen, um einen besseren Startwert zu finden.
Eine funktionsangepaßte Schrittweitensteuerung benötigt sogenannte *effiziente* Schrittweiten:

Definition 3.4.2 (Effiziente Schrittweite)**.** *Eine Schrittweite t_k heißt* effizient*, falls*

$$\exists_{\theta, \theta \neq \theta(x^k, d^k)}\ F(x^k + t_k d^k) \leq F(x^k) - \theta \left(\frac{\langle \nabla F(x^k), d^k \rangle}{||x^k||} \right)^2. \tag{3.32}$$

Die Schrittweite sollte also vom Abstieg $\langle \nabla F(x^k), d^k \rangle$ abhängen und für signifikant kleinere Funktionswerte sorgen. Die Konstante θ ist also eine globale Konstante.
Der Unterschied in Bezug auf den Verlauf des Optimierungsverfahrens bei Verwendung einer heuristischen beziehungsweise effizienten Schrittweite ist in Abbildung 3.8 veranschaulicht.



(a) Heuristische Schrittweite: Da die Schrittweite konstant ist, ist die Wahrscheinlichkeit sehr groß, daß das Verfahren um das Minimum hin- und herspringt sowie größere Fehlerfunktionswerte gefunden werden. Die Größe der Schrittweite ist somit dynamisch zu wählen, was jedoch in der Praxis oft nicht möglich ist, da der Verlauf der Fehlerfunktion *a priori* nicht bekannt ist.

(b) Effiziente Schrittweite: Die Schrittweite ist funktionsangepaßt. Es werden nur kleinere Fehlerfunktionswerte gefunden, so daß die Wahrscheinlichkeit, daß das Verfahren um das Minimum herumspringt, reduziert wird.

Abbildung 3.8: Unterschied zwischen heuristischer (a) und effizienter (b) Schrittweite. Bei der effizienten Schrittweite sind nur kleinere Fehlerfunktionswerte zugelassen, so daß sich das Optimierungsverfahren gerichtet auf das Minimum zubewegt.

Effiziente Schrittweiten werden iterativ bestimmt, das heißt mithilfe eines Algorithmus $t_k^\ell \to t_k^{\ell+1},\ \ell \in \mathbb{N}$, mit Startwert t_k^0. Ein Beispiel ist die sogenannte *skalierte Armijo*-Schrittweite:

Definition 3.4.3 (Skalierte Armijo-Schrittweite)**.** *Eine Schrittweite t_A heißt* skalierte Armijo-Schrittweite*, falls*

$$t_A = \max\{s\beta_A^\ell | \ell = 0, 1, ..., F(x + s\beta_A^\ell d) \leq F(x) + \zeta_A s\beta_A^\ell \langle \nabla F(x), d\rangle\}, \tag{3.33}$$

wobei $\beta_A, \zeta_A \in (0, 1)$, d eine Abstiegsrichtung und $s > 0$.

Es ist zu beachten, daß in Gleichung (3.33) im Gegensatz zu Gleichung (3.32) ein '+' steht. Dies ist dadurch gerechtfertigt, daß d eine Abstiegsrichtung ist, also $\langle \nabla F(x), d\rangle < 0$.

Satz 3.4.4 (Effizienz der skalierten Armijo-Schrittweite)**.** *Falls $F : \mathbb{R}^N \to \mathbb{R} \in C^1$ und ∇F Lipschitz-stetig auf $\mathcal{L}(x^0)$ ist, und falls*

$$s \geq -c\frac{\langle \nabla F, d\rangle}{||d||^2} \tag{3.34}$$

bei fest vorgegebenem $c > 0$, so ist die skalierte Armijo-Schrittweite für alle $x \in \mathcal{L}(x^0)$ effizient. Weiterhin ist t_A wohldefiniert, falls d eine Abstiegsrichtung ist.

Beweis: Der Beweis ist in Geiger u. Kanzow (1999) zu finden. □

Der Nachteil der Armijo-Schrittweitensteuerung ist, daß sie in der Nähe des Minimums äußerst langsam konvergiert und die resultierende Schrittweite $t_A = \beta^{\ell^\star}$ mit sehr großem $\ell^\star$ kaum noch

Gewinn bringt, da sie zu klein ist. Da es in der Praxis äußerst schwierig ist, globale Werte für c und s zu finden, kann die Effizienz der Armijo-Schrittweite im allgemeinen nicht vorausgesetzt werden. Daher ist das Abbruchkriterium $||\nabla F(x^{\text{opt}})|| < \tau$ für die vorliegende Problemstellung ungeeignet.

Zu kleine Schrittweiten werden durch die sogannte *Wolfe-Powell-Schrittweitensteuerung* vermieden:

Definition 3.4.5 (Wolfe-Powell-Schrittweite)**.** *Eine* Wolfe-Powell-Schrittweite t_{WP} *ist eine Armijo-Schrittweite mit* $\zeta_A \in (0, \frac{1}{2})$ *und der Zusatzbedingung*

$$\langle \nabla F(x + t_{WP} d), d\rangle \geq \rho \langle \nabla F(x), d\rangle, \;\; \rho \in [\zeta_A, 1). \tag{3.35}$$

Die Bedingung (3.35) stellt eine zusätzliche Anforderung an den Abstieg in der neuen Iteration. Die Idee dabei ist, daß die Funktion an der neuen Iteration deutlich flacher sein muß. Der Nachteil der Wolfe-Powell-Schrittweite ist jedoch, daß in jedem Schritt t^{ℓ}_{WP} eine Gradientenberechnung $\nabla F(x + t^{\ell}_{WP} d)$ erforderlich ist.

Satz 3.4.6 (Effizienz der Wolfe-Powell-Schrittweite)**.** *Falls F nach unten beschränkt ist, so ist die Wolfe-Powell-Schrittweite t_{WP} wohldefiniert. Falls ∇F auf $\mathcal{L}(x^0)$ Lipschitz-stetig ist, so ist t_{WP} für alle $x \in \mathcal{L}(x^0)$ effizient.*

Beweis: Der Beweis ist in Nocedal u. Wright (1999) zu finden. □

3.4.3 Trust-Region-Verfahren

Die zweite Klasse der hier betrachteten gradientenbasierten numerischen Optimierungsverfahren sind die sogenannten *Trust-Region-(TR-)Verfahren*. Die Schrittweitensteuerung ist bei diesen Verfahren im Algorithmus selbst enthalten. In jeder Iteration k wird eine Schrittweite Δ_k bestimmt, so daß die Funktion F auf $B_{\Delta_k}(x^k)$ durch eine quadratische Form approximiert werden kann. Die kompakte Kugel $B_{\Delta_k}(x^k)$ heißt *Vertrauensgebiet* für F (englisch: *Trust Region*). Für jedes Δ_k wird dann eine geeignete Abstiegsrichtung d^k bestimmt und $x^{k+1} = x^k + d^k$ berechnet.
Die Idee ist dabei die folgende: Solange d^k als nicht geeignet eingestuft wird, wird Δ_k um ein $\gamma_1 \in (0, 1)$ verkleinert. Ansonsten wird Δ_k vorsichtshalber, das heißt um den Suchbereich zu vergrößern, um ein $\gamma_2 > 1$ vergrößert. Was *geeignet* in diesem Fall bedeutet, sei im folgenden erläutert: Die Funktion F wird auf dem Vertrauensgebiet durch eine quadratische Form approximiert, die aus einer Taylorentwicklung der Ordnung 2 herzuleiten ist:

$$q_{x^k}(d^k) := F(x^k + d^k) \doteq F(x^k) + \langle \nabla F(x^k), d^k\rangle + \frac{1}{2}(d^k)^T D^2 F(x^k) d^k. \tag{3.36}$$

Dann wird $q_{x^k}(d^k)$ auf dem Vertrauensgebiet minimiert, das heißt es wird

$$\min_{||d^k|| \leq \Delta_k} q_{x^k}(d^k) \tag{3.37}$$

bestimmt. Dann wird die Güte der Approximation mittels des Verhältnisses

$$r^k := \frac{F(x^{k+1}) - F(x^k)}{q_{x^k}(d^k) - q_{x^k}(0)} \overset{(3.36)}{=} \frac{F(x^{k+1}) - F(x^k)}{q_{x^k}(d^k) - F(x^k)} \tag{3.38}$$

geschätzt. Ist $r^k \approx 0$, so stimmt das quadratische Modell nur schlecht mit der originalen Funktion F überein, da es den Abfall von F überschätzt. In diesem Fall wird die Abstiegsrichtung d^k als nicht geeignet eingestuft. Andererseits, falls $r^k \approx 1$ oder sogar $r^k > 1$, so stimmt das quadratische Modell gut mit F überein oder F fällt sogar stärker ab als q_{x^k}. In diesem Fall wird d^k als geeignet eingestuft, und Δ_k kann vergrößert werden. In der Praxis werden zwei Schwellenwerte $0 < \eta_1 < \eta_2$ eingeführt. Falls $r^k \geq \eta_1$, wird d^k als Abstiegsrichtung akzeptiert, und falls sogar $r^k > \eta_2$, so wird Δ_k vergrößert.
Sämtliche Trust-Region-Verfahren unterscheiden sich in der Lösung des *Trust-Region-Teilproblems* (3.37). Ist dieses lösbar, so besitzen die Trust-Region-Verfahren äußerst gute Konvergenzeigenschaften:

Satz 3.4.7 (Konvergenz der Trust-Region-Verfahren). *Falls das Trust-Region-Teilproblem* (3.37) *lösbar ist und $D^2F(x^*)$ spd ist, wobei x^* ein Häufungspunkt der durch das Verfahren definierten Folge $(x^k)_{k\in\mathbb{N}}$ ist, so konvergiert das Verfahren gegen x^*, und es gilt*

$$\forall_{m\in(0,1)}\ \forall_{k>0}\ ||x^k - x^{\text{opt}}|| \leq c_k m^k, \tag{3.39}$$

wobei $(c_k)_{k\in\mathbb{N}}$ eine Nullfolge ist, das heißt, die Konvergenz ist r-überlinear. Falls D^2F in x^* Lipschitz-stetig ist, so ist die Konvergenz sogar q-quadratisch.

Beweis: Der Beweis ist in Nocedal u. Wright (1999) zu finden. □

Allerdings ist in jedem Iterationsschritt die Berechnung der Hesse-Matrix D^2F erforderlich.
Eine weitere Problematik liegt in der Lösung des Trust-Region-Teilproblems. Es handelt sich dabei um ein nichtlineares Optimierungsproblem mit Nebenbedingungen. Da eine Lösung mithilfe des Satzes von Lagrange bei der vorliegenden Problemstellung als zu aufwendig angesehen wird, werden hier zwei verschiedene Lösungsmethoden des Trust-Region-Teilproblems betrachtet.

Zur Lösung des Trust-Region-Teilproblems Im allgemeinen reicht es aus, das Trust-Region-Teilproblem (3.37) nur approximativ zu lösen. Die Lösung muß im aktuellen Vertrauensgebiet liegen und ein signifikantes Abfallen des quadratischen Modells garantieren. Ein erster Ansatz zur Lösung des Minimierungsproblems mit der Nebenbedingung $||d^k|| \leq \Delta$ ist der sogenannte *Cauchy-Punkt*:

Definition 3.4.8 (Cauchy-Punkt). *Es seien $\Delta > 0$ und*

$$\phi : [0,1] \to \mathbb{R},\ \phi(t) := q_x\left(-t\frac{\Delta\nabla F(x)}{||\nabla F(x)||}\right).$$

Dann ist der Cauchy-Punkt d_C *gegeben als das Minimum von* ϕ *auf* $[0,1]$*, also*

$$d_C = -t_C \frac{\Delta \nabla F(x)}{||F(x)||}, \text{ wobei } \phi'(t_C) = 0 \vee t_C = 1. \tag{3.40}$$

Es ist

$$t_C = \min\left(\frac{||\nabla F(x)||^3}{\Delta \nabla F(x)^T D^2 F(x) \nabla F(x)}, 1\right).$$

Der Cauchy-Punkt zeigt somit in die Richtung des negativen Gradienten und garantiert daher einen Abstieg des quadratischen Modells.
Obwohl der Cauchy-Punkt leicht zu berechnen ist, ist man stets an einer besseren Lösung von (3.37) interessiert. Denn durch die Verwendung des Cauchy-Punkts unterscheidet sich das entsprechende Trust-Region-Verfahren nicht vom Verfahren des steilsten Abstiegs. Um jedoch eine schnellere Konvergenz, also die Konvergenzeigenschaft (3.39), zu garantieren, muß ein besserer Punkt als der Cauchy-Punkt gefunden werden. Da Trust-Region-Verfahren die Hesse-Matrix D^2F verwenden, sollte dies auch dazu genutzt werden, um die Konvergenzgeschwindigkeit zu erhöhen.
Es werden hier zwei verschiedene verbesserte Lösungsansätze für das Trust-Region-Teilproblem betrachtet. Einer davon ist geometrisch motiviert und heißt *Doppeltes Hundebein*. Die unrestringierte Lösung von (3.37) ist $d_N := -(D^2F(x))^{-1}\nabla F(x)$, also die Newton-Richtung. Die heuristische Überlegung besteht nun darin, daß bei kleinem Δ eher der Cauchy-Punkt als Lösung in Frage kommt und bei großem Δ eher die Newton-Richtung. Das quadratische Modell wird entlang des folgenden Pfades monton fallen:

$$\tilde{d}(t) := \begin{cases} td_C, & 0 \le t \le 1 \\ d_C + (t-1)(d_N - d_C), & 1 \le t \le 2. \end{cases} \tag{3.41}$$

Dies zeigt das folgende Lemma:

Lemma 3.4.9. *Falls* $D^2F(x)$ *positiv definit ist, dann gilt:*

- $||\tilde{d}(\cdot)||$ *ist monoton wachsend,*
- $q_x(\tilde{d}(\cdot))$ *ist monton fallend.*

Beweis: Der Beweis ist in Nocedal u. Wright (1999) zu finden. □

Falls $||d_N|| \ge \Delta$, so schneidet $\tilde{d}(t)$ den Rand des Vertrauensgebiets an genau einer Stelle, und zwar für $t = t_0$ mit

$$||d_C + (t_0 - 1)(d_N - d_C)||^2 = \Delta^2. \tag{3.42}$$

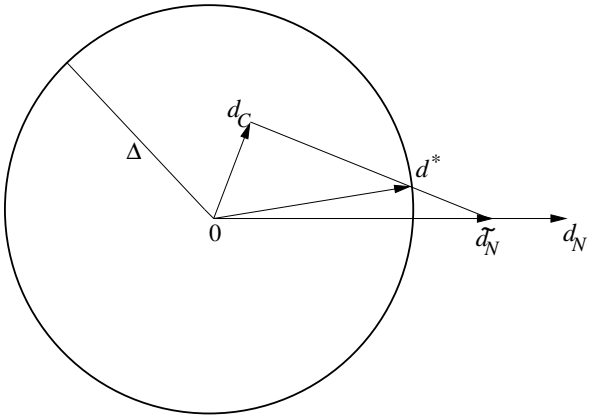


Abbildung 3.9: Geometrische Veranschaulichung des Doppelten Hundebeins: Die Suchrichtung d^* ergibt sich aus einer Linearkombination der skalierten Newton-Richtung $\tilde{d}_N = \gamma d_N$ und des Cauchy-Punkts d_C. Dadurch, daß d^* am Rand von $B_\Delta(0)$ liegen muß, ist es eindeutig bestimmt.

Es gibt nun drei verschiedene Fälle zum Erhalt einer Lösung d^* des Trust-Region-Teilproblems (3.37):

1. $D^2F(x)$ nicht positiv definit $\Rightarrow d^* := d_C$.
2. $D^2F(x)$ positiv definit, $||d_N|| < \Delta \Rightarrow d^* := d_N$.
3. $D^2F(x)$ positiv definit, $||d_N|| \geq \Delta \Rightarrow$ bestimme d^* aus Gleichung (3.42).

Dies ist der Algorithmus des sogenannten *Einfachen Hundebeins*. Mit der zusätzlichen Skalierung $\tilde{d}_N = \gamma d_N$ resultiert das *Doppelte Hundebein*. Dabei ist

$$\gamma \in \left(\frac{\Delta}{||(D^2F(x))^{-1}\nabla F(x)||}, 1 \right) = \left(\frac{\Delta}{||d_N||}, 1 \right).$$

Die Motivation dafür ist, daß das quadratische Modell entlang der Strecke von d_C nach $\tilde{d}_N$ weiter abfällt. Abbildung 3.9 veranschaulicht geometrisch die Idee des Doppelten Hundebeins. Es ist zu beachten, daß für $\gamma = 1$ wieder das Einfache Hundebein resultiert.

Eine Voraussetzung dafür, daß die Methode des Doppelten Hundebeins konvergiert, ist die positive Definitheit der Hesse-Matrix. Da letztere allerdings in der Praxis oftmals nicht gewährleistet werden kann, wird ein Verfahren benötigt, das nicht von der positiven Definitheit abhängig ist. Es wird somit hier noch ein weiterer Algorithmus zur Lösung des Trust-Region-Teilproblems betrachtet, welcher eine nahezu exakte Lösung bestimmt. Es seien $g := \nabla F(x)$, $B := D^2F(x)$ und $\lambda \geq 0$. Nach den sogenannten *Karush-Kuhn-Tucker-(KKT-)Bedingungen*, siehe zum Beispiel Nocedal u. Wright (1999) oder Hülsmann (2006), ist d^* genau dann eine Lösung des Teilproblems (3.37), wenn gilt:

$$\begin{aligned} (B + \lambda I)d^* &= -g \\ \lambda(\Delta - ||d^*||) &= 0. \end{aligned} \tag{3.43}$$

Falls das Minimum d^* nicht am Rand des Vertrauensgebiets liegt, so ist $\lambda = 0$ Lösung von (3.43). Dann muß B jedoch positiv definit sein. Ansonsten wird $\lambda > 0$ so gewählt, daß $B + \lambda I$ positiv definit ist, für das gleichzeitig

$$||d(\lambda)|| = || - (B + \lambda I)^{-1} g|| = \Delta \tag{3.44}$$

gilt. Das Verfahren beruht auf einer Eigenwertzerlegung von $B + \lambda I$. Da B symmetrisch ist, ist B diagonalisierbar, das heißt, es existiert eine orthogonale Matrix Q und eine Diagonalmatrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N)$ mit $B = Q\Lambda Q^T$, wobei $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ die Eigenwerte von B sind. Da B symmetrisch ist, sind diese reell. Wegen $B + \lambda I = Q(\Lambda + \lambda I)Q^T$ gilt für $\lambda \neq -\lambda_j$

$$d(\lambda) = -Q(\Lambda + \lambda I)Q^T g = -\sum_{j=1}^{N} \frac{\langle q_j, g \rangle}{\lambda_j + \lambda} q_j, \tag{3.45}$$

wobei q_j die Spaltenvektoren von Q sind. Da diese orthogonal sind, folgt

$$||d(\lambda)||^2 = \sum_{j=1}^{N} \frac{\langle q_j, g \rangle^2}{(\lambda_j + \lambda)^2}. \tag{3.46}$$

Falls $\lambda > -\lambda_1$, so gilt $\forall_{j=1,...,N}\ \lambda_j + \lambda > 0$. Damit ist $||d(\lambda)||$ stetig und nicht abfallend mit λ auf dem Intervall $(-\lambda_1, \infty)$. Es gilt $\lim_{\lambda \to \infty} ||d(\lambda)|| = 0$, und falls $\langle q_j, g \rangle \neq 0$, so gilt ebenfalls $\lim_{\lambda \to \lambda_j} ||d(\lambda)|| = \infty$. Es folgt nach dem Zwischenwertsatz, daß Gleichung (3.44) auf dem Intervall $(-\lambda_1, \infty)$ eindeutig lösbar ist.
Falls B bereits positiv definit ist und $||B^{-1}g|| \leq \Delta$, so kann d^* als Newton-Richtung gewählt werden. Falls B positiv definit ist und $||B^{-1}g|| > \Delta$ oder B indefinit ist, wird d^* mithilfe der eindeutig bestimmten positiven Lösung von Gleichung (3.44) bestimmt. Im ersten Fall gilt $\lambda \in (0, \infty)$ und im zweiten Fall $\lambda \in (-\lambda_1, \infty)$. Dies wird mit einem Newton-Verfahren zur Nullstellenbestimmung durchgeführt. Anstatt $\Phi_1(\lambda) := ||d(\lambda)|| - \Delta$ wird jedoch

$$\Phi_2(\lambda) := \frac{1}{\Delta} - \frac{1}{||d(\lambda)||} \tag{3.47}$$

betrachtet, da Φ_1 für $\lambda \approx \lambda_1$ hochgradig nichtlinear ist, was zu einer sehr langsamen Konvergenz des Newton-Verfahrens führt.
Bei der Newton-Iteration $\lambda^{(\ell)} \to \lambda^{(\ell+1)}$ sollte der Startwert $\lambda^{(0)}$ so gewählt werden, daß $B + \lambda^{(0)} I$ positiv definit ist, also in jedem Fall $\lambda^{(0)} > -\lambda_1$. Falls B positiv definit ist und $||B^{-1}g|| > \Delta$, so ist sogar $\lambda > 0$ zu wählen. Ist $B + \lambda I$ positiv definit, so kann auf diese Matrix eine Cholesky-Zerlegung $B + \lambda^{(\ell)} I = R^T R$ angewandt werden, was zu einer Erhöhung der Effizienz führt. Dann werden zwei lineare Gleichungssysteme gelöst:

1. $R^T R p_\ell = -g$
2. $R^T q_\ell = p_\ell$

Lemma 3.4.10 (Nahezu exakte Lösung des TR-Teilproblems)**.** *Es sei $\Delta > 0$, und $\lambda^{(0)} > 0$ sei so gewählt, daß $B + \lambda^{(0)} I$ positiv definit (in jedem Fall $\lambda > -\lambda_1$). Falls B indefinit, so sei $\langle q_1, g \rangle \neq 0$. Dann konvergiert die Folge $\left(d(\lambda^{(\ell)})\right)_{\ell \in \mathbb{N}}$, wobei*

$$\lambda^{(\ell+1)} = \lambda^{(\ell)} + \frac{||p_\ell|| - \Delta}{\Delta} \left(\frac{||p_\ell||}{||q_\ell||} \right)^2, \tag{3.48}$$

gegen die eindeutig bestimmte Lösung des TR-Teilproblems.

Beweis: Die positive Definitheit von $B + \lambda I$ ist nach den Karush-Kuhn-Tucker-Bedingungen notwendig für die Existenz eines eindeutig bestimmten Minimums am Rand des Vertrauensgebiets. Weiterhin ist dann eine Cholesky-Zerlegung durchführbar, und die Invertierbarkeit von $B + \lambda I$ ist ebenfalls gewährleistet. Die Voraussetzung $\langle q_1, g\rangle \neq 0$ garantiert die Eindeutigkeit einer Lösung von Gleichung (3.44) im Intervall $(-\lambda_1, \infty)$. Aus Stetigkeitsgründen bleibt nur noch zu zeigen, daß die Iterationsvorschrift (3.48) mit der Newton-Iteration übereinstimmt. Es gilt:

$$\begin{aligned} ||q_\ell||^2 &= ||R^{-T}p_\ell||^2 = p_\ell^T R^{-1}R^{-T}p_\ell \\ &= p_\ell^T \left(R^T R\right)^{-1} p_\ell = p_\ell^T \left(B + \lambda^{(\ell)} I\right)^{-1} p_\ell \\ &\overset{(3.46)}{=} \sum_{j=1}^{N} \frac{\langle q_j, g\rangle^2}{(\lambda_j^{(\ell)} + \lambda^{(\ell)})^3}. \end{aligned} \tag{3.49}$$

Es ist zu beachten, daß $p_\ell = -(R^T R)^{-1} g = -(B + \lambda^{(\ell)} I)^{-1} g = d(\lambda^{(\ell)})$. Weiterhin gilt innerhalb der Iterationsvorschrift des Newton-Verfahrens, welches die Nullstelle der Funktion Φ_2 aus Gleichung (3.47) findet:

$$\begin{aligned} -\frac{\Phi_2(\lambda^{(\ell)})}{\Phi_2'(\lambda^{(\ell)})} &= -\frac{\frac{1}{\Delta} - \frac{1}{||d(\lambda^{(\ell)})||}}{\frac{1}{||d(\lambda^{(\ell)})||^2}\left(-\frac{1}{||d(\lambda^{(\ell)})||}\sum_{j=1}^{N}\frac{\langle q_j, g\rangle^2}{(\lambda_j^{(\ell)}+\lambda^{(\ell)})^3}\right)} \overset{(3.49)}{=} \frac{\frac{1}{\Delta} - \frac{1}{||d(\lambda^{(\ell)})||}}{\frac{1}{||d(\lambda^{(\ell)})||^3}||q_\ell||^2} \\ &= \frac{\frac{||p_\ell||^3}{\Delta} - ||p_\ell||^2}{||q_\ell||^2} = \left(\frac{||p_\ell||}{||q_\ell||}\right)^2 \frac{||p_\ell|| - \Delta}{\Delta}. \end{aligned} \tag{3.50}$$

Die Iterationsvorschrift (3.48) stimmt mit Gleichung (3.50) überein. □

Im Gegensatz zur Methode des Doppelten Hundebeins wird die Lösung hier nahezu exakt bestimmt, das heißt, die zu minimierende Funktion wird nicht nur entlang einer Richtung betrachtet, auf der sie monoton fallend ist, sondern es wird das eindeutig bestimmte Minimum am Rand des Vertrauensgebiets ermittelt. Nahezu exakt bedeutet, daß es dennoch über ein Iterationsverfahren bestimmt wird. Befindet sich das Minimum von q_x innerhalb des Vertrauensgebiets, so ist die Lösung des TR-Teilproblems eine Newton-Raphson-Richtung. Da Δ von Anfang an so klein gewählt ist, daß das quadratische Modell möglichst gut mit F übereinstimmt, kann am Schluß des Trust-Region-Verfahrens einfach ein Newton-Raphson-Verfahren nachgeschaltet werden, von dem erwartet werden kann, daß es nach nur sehr wenigen Iterationen zum Minimum von F gelangt.
In Nocedal u. Wright (1999) sind weitere Verfahren dargestellt, welche zum Beispiel dann angewandt werden können, wenn die Hesse-Matrix nicht als positiv definit vorausgesetzt werden kann, und eines für den denkbar schlechtesten Fall, sprich wenn auch nicht die Voraussetzung $\langle q_1, g\rangle \neq 0$ erfüllt ist.

3.5 Anpassung gradientenbasierter Verfahren an Molekulare Simulationen

Es ist keineswegs trivial, die in Abschnitt 3.4 vorgestellten gradientenbasierten numerischen Optimierungsverfahren auf Molekulare Simulationen anzuwenden, denn aufgrund der nicht analytisch darstellbaren Fehlerfunktion und des statistischen Rauschens ist eine akkurate Gradientenberechnung nur schwierig zu realisieren. Einige oder ähnliche Verfahren sind bislang lediglich zur Minimierung von Potentialhyperflächen angewandt worden, zum Beispiel ein präkonditionierter BFGS-Algorithmus in Jiang u. a. (2004) und eine Kombination des Simplex-Verfahrens nach Nelder und Mead mit einem anderen ableitungsfreien Verfahren in Fan (2002). Letztere Kombination wurde mit gradientenbasierten Verfahren wie der Methode des steilsten Abstiegs und dem BFGS-Verfahren verglichen.

In diesem und den folgenden Abschnitten werden selbst entwickelte Verfahren detailliert dargestellt, die der effizienten Anpassung gradientenbasierter Verfahren an Molekulare Simulationen dienen. Bei der Verwendung gradientenbasierter numerischer Optimierungsverfahren sind zunächst an F gewisse Glattheitsvoraussetzungen zu stellen. Falls $\forall_{i=1,...,n}\, f_i^{\mathrm{sim}} \in C^m(\Omega),\ m \in \mathbb{N}$, so ist auch $F \in C^m(\Omega)$. Die Summierung quadratischer Fehlerterme sorgt für die Beibehaltung der Glattheitseigenschaften und verhindert die Auslöschung von einzelnen Fehlern mit entgegengesetztem Vorzeichen. Durch die Division durch f_i^{exp} werden stets relative Fehler betrachtet, so daß die physikalischen Einheiten der Zielgrößen keinerlei Auswirkung auf die Fehlerfunktion haben. Das primäre Ziel des Optimierungsablaufs ist, eine optimale Lösung x^{opt} zu finden, so daß

$$\nabla F(x^{\mathrm{opt}}) = 0,$$

also einen stationären Punkt der Fehlerfunktion zu finden. Ein lokales Minimum liegt vor, falls zusätzlich die Hesse-Matrix $D^2F(x^{\mathrm{opt}})$ symmetrisch positiv definit (spd) ist. Im Idealfall sollte es sich um ein globales Minimum handeln, es sollte also gelten: $F(x^{\mathrm{opt}}) = 0$. Die letzte Bedingung ist bei überbestimmten Optimierungsproblemen jedoch oftmals nicht zu erfüllen. Ob der Algorithmus gegen ein lokales oder globales Minimum konvergiert, ist von der Wahl von x^0 und Ω abhängig. Diese Problematik wird in Abschnitt 3.5.1 detailliert erörtert.
Der Gradient ist gegeben durch die partiellen Ableitungen

$$\frac{\partial F}{\partial x_j}(x) = -2 \sum_{i=1}^{n} w_i^2 \frac{f_i^{\mathrm{exp}} - f_i^{\mathrm{sim}}(x)}{\left(f_i^{\mathrm{exp}}\right)^2} \frac{\partial f_i^{\mathrm{sim}}}{\partial x_j}(x),\ j = 1, ..., N. \tag{3.51}$$

Das Hauptproblem liegt in der Bestimmung der partiellen Ableitungen der f_i^{sim}. Dies ist deshalb so schwierig, weil keine analytische Form dieser Funktionen bekannt ist. Die Frage nach der Glattheit muß dennoch diskutiert werden, um einen Erfolg der Algorithmen auch theoretisch zu belegen. In jedem Fall sind diese Funktionen mit statistischem Rauschen behaftet, was das Problem umso schwieriger macht. Dies wird in diesem Abschnitt detaillierter behandelt. An dieser Stelle sei nur erwähnt, daß eine Möglichkeit darin besteht, die partiellen Ableitungen der f_i^{sim} durch finite Differenzen zu approximieren:

$$\frac{\partial f_i^{\mathrm{sim}}}{\partial x_j}(x) = \frac{f_i^{\mathrm{sim}}(x_1, ..., x_j + h, ..., x_N) - f_i^{\mathrm{sim}}(x)}{h} + \mathcal{O}(h), \tag{3.52}$$

wobei $i = 1, ..., n,\ j = 1, ..., N$ und $h > 0$. Die zweiten partiellen Ableitungen können gemäß

$$\begin{aligned}\frac{\partial^2 f_k^{\text{sim}}}{\partial x_i \partial x_j}(x) &= \frac{1}{h}\left(\frac{\partial f_k^{\text{sim}}}{\partial x_i}(x_1, ..., x_j + h, ..., x_N) - \frac{\partial f_k^{\text{sim}}}{\partial x_i}(x)\right) + \mathcal{O}(h) \\ &= \frac{1}{h^2}(f_k^{\text{sim}}(x_1, ..., x_i + h, ..., x_j + h, ..., x_N) - f_i^{\text{sim}}(x_1, ..., x_j + h, ..., x_N) \\ &- f_k^{\text{sim}}(x_1, ..., x_i + h, ..., x_N) + f_k^{\text{sim}}(x)) + \mathcal{O}(h), \end{aligned} \tag{3.53}$$

wobei $k = 1, ..., n$, und $i, j = 1, ..., N$, angenähert werden.

Auf Diskretisierungen höherer Ordnung, die zum Beispiel durch zentrale Differenzen realisiert werden können, wird aufgrund des höheren Rechenaufwands verzichtet.
Ein allgemeines Problem von Iterationsverfahren ist die Wahl geeigneter Startparameter. Diese bestimmen die Konvergenz des jeweiligen Verfahrens und auch dessen Grenzwert, das heißt sie legen fest, ob das Verfahren gegen ein lokales beziehungsweise globales Minimum oder gar einen Sattelpunkt konvergiert. Für die Iterationsverfahren sind somit gewisse Anfangsbedingungen festzulegen, das heißt, bei den Startparametern Molekularer Simulationen muß es sich um physikalisch sinnvolle Parameter handeln, die bereits einen relativ kleinen Wert für die Fehlerfunktion aus Gleichung (3.2) liefern. Weiterhin gibt es für die Kraftfeldparameter gewisse Einschränkungen, was zur Festlegung von Randbedingungen führt. Die Definition und Behandlung von Anfangs- und Randbedingungen sowie die damit verbundene Güte des berechneten Minimums werden in Abschnitt 3.5.1 erläutert. Es wird dort auch eine eingehende Diskussion über den Erhalt lokaler beziehungsweise globaler Minima gegeben.
Ein weiteres Problem ist die Tatsache, daß die von Simulationsprogrammen errechneten thermodynamischen Durchschnitte stets mit statistischem Rauschen behaftet sind. Da es *a priori* nicht klar ist, inwieweit die hier betrachteten Verfahren mit Rauschen umgehen können, ist eine Vorabanalyse in Bezug auf Rauschen unentbehrlich. In Abschnitt 3.5.2 wird dargelegt, wie Simulationen durch glatte Funktionen ersetzt werden können, die bestimmte Zielgrößen in Abhängigkeit von bestimmten Kraftfeldparametern beschreiben und auf welche dann künstliches Rauschen addiert wird. Dieses Hilfskonstrukt beziehungsweise Ersatzverfahren reduziert den Rechenaufwand im Gegensatz zu Molekularen Simulationen quasi auf 0. Dies liefert die Grundlage einer eingehenden Bewertung der einzelnen Verfahren bezüglich Rauschen sowie die Darstellung einer geglätteten Fehlerfunktion, durch die ein geeignetes Abbruchkriterium festgelegt werden kann, was in Abschnitt 3.5.3 diskutiert wird. Wie Rauschen dann bei Simulationen selbst behandelt werden kann, wird in Abschnitt 3.5.4 dargestellt. Bei der Behandlung des Rauschens spielt die Grobheit der Diskretisierung des Gradienten und der Hesse-Matrix eine entscheidende Rolle. Die Diskretisierungsschrittweite h aus Gleichung (3.52) muß in Abhängigkeit von der Auswirkung des Rauschens auf die zu minimierende Funktion vergrößert werden. Der Effekt des Rauschens auf die Fehlerfunktion wird in Abschnitt 3.5.5 hergeleitet, woraus sich eine theoretische Schlußfolgerung ergibt, wie h geeignet gewählt werden kann. In Abschnitt 3.5.6 wird eine praktische Analyse zur geeigneten Wahl von h für die Hesse-Matrix durchgeführt.

3.5.1 Anfangs- und Randwerte: Lokale und globale Minima

Minima, die aus lokal konvergenten gradientenbasierten Verfahren abhängen, sind von der Wahl der Anfangsbedingungen abhängig, sprich von den Startparametern. Befinden sich diese in der

Nähe eines globalen Minimums, so wird das entsprechende Verfahren auch dagegen konvergieren. Globale Konvergenz ist auch garantiert, wenn die Fehlerfunktion konvex ist, was jedoch in der Praxis nicht vorausgesetzt werden kann. Aber auch in diesem Falle ist der Startvektor in einer Umgebung des Minimums zu lokalisieren. Die initialen Kraftfeldparameter müssen physikalisch sinnvoll gewählt werden. Für viele Substanzen gibt es bereits Standardkraftfelder, welche in der Literatur zu finden sind. Standardkraftfelder sind beispielsweise die in Abschnitt 2.2.4 beschriebenen Kraftfelder OPLS, Gromos oder Amber. Sind jedoch keine Standardkraftfeldparameter verfügbar oder erweisen sich diese als ungeeignet, ist die Fehlerfunktion auf geeignete Art und Weise diskret zu analysieren, so daß ein approximatives Minimum bestimmt werden kann und die zugehörigen Parameter als Startparameter verwendet werden. Darauf wird in Abschnitt 3.6.6 näher eingegangen.

Die meisten Kraftfeldparameter sind aus einem zulässigen Definitionsbereich zu wählen. Die Lennard-Jones-Parameter σ und ε beispielsweise müssen in jedem Fall positiv sein, um physikalisch sinnvoll und interpretierbar zu sein. Viele Parameter sind auch nach oben beschränkt. Dies führt zu Randbedingungen für die Kraftfeldparameter, welche innerhalb des Algorithmus in jedem Fall eingehalten werden müssen. Führt das Verfahren zu einem Punkt außerhalb des zulässigen Bereiches, ist also beispielsweise einer der Kraftfeldparameter negativ, so ist eine Simulation mit den neuen Parametern nicht möglich. Oftmals werden MD-Simulationen mit Nebenbedingungen (siehe Abschnitt B) durchgeführt. Werden dabei beispielsweise Bindungslängen konstant gehalten, so darf der Längenparameter σ nicht zu groß gewählt werden, da ansonsten beim Versuch des Erhalts eines kleineren Atomabstandes die Abstoßungskräfte zu hoch werden, und das System wird mechanisch instabil. Für sämtliche Kraftfeldparameter gilt, daß sinnvolle Simulationen nur in einem kompakten zulässigen Bereich durchführbar sind. Jeder einzelne Parameter muß bestimmten vorher festgelegten Nebenbedingungen genügen. Dies führt zu einer zusätzlichen Schwierigkeit für die numerischen Optimierungsverfahren. Eine Möglichkeit besteht darin, Optimierungsverfahren mit Nebenbedingungen zu betrachten. Es handelt sich in diesem Fall jedoch um nichtlineare Optimierungsprobleme mit zumeist linearen Nebenbedingungen, die mit Lagrange- und KKT-Verfahren behandelt werden können. Zur Bestimmung der Lagrange-Multiplikatoren ist jedoch eine erhebliche Erhöhung des Rechenaufwands erforderlich, da das Problem höherdimensional wird. Dadurch steigt die Anzahl an durchzuführenden Simulationen. Außerdem ist es möglich, daß die zu minimierende Funktion im unzulässigen Bereich ausgewertet werden muß, was bei Molekularen Simulationen wie oben erläutert nicht möglich ist. Optimierungsverfahren mit Nebenbedingungen sind aus diesen Gründen bei der vorliegenden Applikation nicht geeignet.
Eine in Abschnitt 3.4.2 angesprochene funktionsadaptierte Schrittweitensteuerung wird daher für das Problem der Randbedingungen bei Abstiegsverfahren verwendet. Die Schrittweite soll dabei dafür sorgen, daß das Verfahren nicht aus dem kompakten zulässigen Gebiet herausläuft. Da die Wolfe-Powell-Schrittweite aus den genannten Gründen nicht praktikabel ist, werden nur heuristische und Armijo-Schrittweiten betrachtet. Die heuristische Schrittweite ist jedoch nur in zwei Fällen sinnvoll:

1. Falls der Startwert vom Minimum zu weit entfernt ist, was anhand einer sehr hohen Gradientennorm oder einem langsamen Fortschreiten bereits zu Beginn des Verfahrens festzustellen ist, können 3–4 heuristische Schritte äußerst nützlich sein, zumal dadurch auch mögliche intermediäre lokale Minima übersprungen werden können.

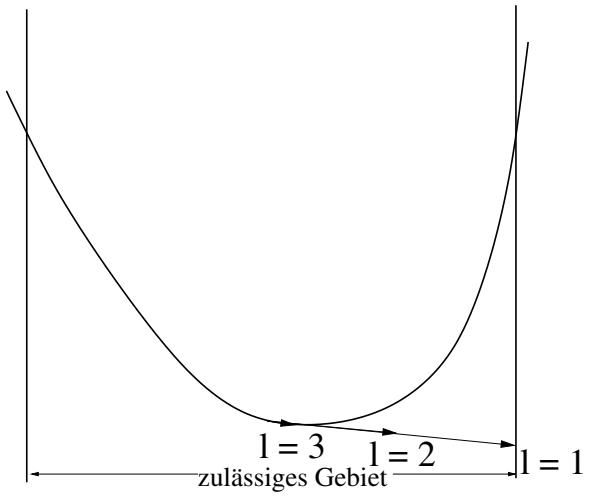


Abbildung 3.10: Veranschaulichung der Armijo-Schrittweitensteuerung: Da β_A *a priori* nicht bekannt ist, wird es so bestimmt, daß das Optimierungsverfahren genau den Rand des zulässigen Gebiets trifft. Da dies nicht gewünscht ist, wird das so erhaltene β_A lediglich als Startwert ($\ell = 1$) für die Armijo-Schrittweitensteuerung verwendet, welche bei $\ell = 2$ beginnt.

2. Falls das Verfahren an den Rand des zulässigen Gebiets konvergiert, sind einige heuristische Schritte nützlich, um einen neuen Startwert in einem neuen zulässigen Gebiet zu finden.

Bei sämtlichen Abstiegsverfahren wird zunächst die Armijo-Schrittweite verwendet. Dabei wird in Gleichung (3.34) zunächst $s = 1$ gesetzt. Falls einer der beiden oben genannten Fälle eintritt, werden einige Versuche mit der heuristischen Schrittweite unternommen. Die Armijo-Schrittweite hat den Nachteil, daß sie schnell gegen Null konvergiert und in der Nähe des Optimums irgendwann keinen Gewinn mehr bringt, da eine Vielzahl an Armijo-Schritten notwendig ist, um als Resultat eine äußerst kleine Schrittweite zu erhalten. Außerdem kann durch die Wahl $s = 1$ nicht garantiert werden, daß die Schrittweite effizient ist. Da die Fehlerfunktion jedoch mit Rauschen behaftet ist, kann ohnehin nicht erwartet werden, daß das Minimum exakt bestimmt wird. Daher ist die Armijo-Schrittweite bei dieser Anwendung geeignet.

Zur Einhaltung der Randbedingungen ist die Definition einer zulässigen Schrittweite erforderlich, welche genau an den Rand des zulässigen Gebiets führt:

Definition 3.5.1 (Zulässige Schrittweite)**.** *Es sei $[c_i^k, C_i^k]$ das zulässige Gebiet für die i-te Komponente der Iteration x^k. Dann ist die* zulässige Schrittweite t_k^{adm} *gegeben durch*

$$t_k^{\mathrm{adm}} := \max\{t_k > 0 | \forall_{i=1,\ldots,N}\ c_i^k \leq x_i^k + t_k d_i^k \leq C_i^k\}, \tag{3.54}$$

wobei d^k die Abstiegsrichtung des Verfahrens ist.

Durch die Maximumsbedingung gilt dann:

$$\exists_i\ \left(x_i^k + t_k^{\mathrm{adm}} d_i^k = c_i^k\right) \vee \left(x_i^k + t_k^{\mathrm{adm}} d_i^k = C_i^k\right). \tag{3.55}$$

Daher führt die zulässige Schrittweite an den Rand des zulässigen Gebiets. Bei der Armijo-Schrittweite wird $\beta_A := t_k^{\text{adm}}$ gesetzt, und die Armijo-Iteration startet bei $\ell = 1$, wobei β_A^1 nicht als Armijo-Schrittweite verwendet wird, da diese an den Rand führen würde. Dies ist in Abbildung 3.10 veranschaulicht. Falls $\beta_A > 1$, das heißt, falls die Abstiegsrichtung nicht aus dem zulässigen Gebiet hinaus führt, ist die Schrittweite an die Länge der Abstiegsrichtung anzupassen. Denn führt die Abstiegsrichtung nicht über das Minimum hinaus, so ist $t_A > 1$ eine sinnvolle Wahl. Allerdings konvergiert die Folge $(\beta_A^\ell)_{\ell \in \mathbb{N}}$ nicht mehr gegen 0. Im Falle von $\beta_A > 1$ wird somit gemäß Satz 3.4.4 folgendes gesetzt:

$$\begin{aligned} c &:= 0.01 - \frac{||d||^2 \beta_A}{\langle \nabla F, d \rangle} \\ s &:= -c \frac{\langle \nabla F, d \rangle}{||d||^2}. \end{aligned} \tag{3.56}$$

Damit gilt $s > \beta_A$ und $\tilde{\beta}_A := \frac{\beta_A}{s} < 1$. Als skalierte Armijo-Schrittweite wird dann $t_A := s\tilde{\beta}_A^\ell$ gewählt, damit zu kleine Armijo-Schritte vermieden werden.
Bei der heuristischen Schrittweite ist vom Benutzer problemabhängig ein Faktor $\kappa < 1$ zu wählen, und die resultierende heuristische Schrittweite ist dann $t_H := \kappa t_k^{\text{adm}}$.
Bei den Trust-Region-Verfahren wird ein entsprechendes maximales $\Delta_k^{\text{adm}} > 0$ berechnet, so daß $B_{\Delta_k^{\text{adm}}}(x^k) \subset \Omega$, wobei Ω das zulässige Gebiet ist. Der Startwert $\Delta_k^0 := \tilde{\kappa} \Delta_k^{\text{adm}}$ wird dann durch ein vorher festgelegtes $\tilde{\kappa} << 1$ bestimmt.

3.5.2 Möglichkeiten zur Verfahrensevaluation und künstliches Rauschen

Um sowohl einen Eindruck der Gestalt der Fehlerfunktion zu gewinnen als auch eine adäquate Bewertung der Optimierungsverfahren in Bezug auf statistisches Rauschen durchzuführen, ist es sehr hilfreich, zeitintensive Simulationen geeignet zu ersetzen. Physikalische Eigenschaften wie Siededichte, Dampfdichte und Dampfdruck können mittels sogenannter *Korrelationsfunktionen*, welche auf Stoll u. a. (2001) zurückgehen, in Abhängigkeit von bestimmten Kraftfeldparametern ausgedrückt werden. Es handelt sich dabei um Hilfskonstrukte, welche im Vergleich zu Molekularen Simulationen einen verschwindend geringen Rechenaufwand benötigen. Sie sind allerdings nur für spezielle Probleme einsetzbar und können keinesfalls Molekulare Simulationen generell ersetzen. Falls $x_i,\ i = 1, ..., N$, die Kraftfeldparameter sind und $f_j,\ j = 1, ..., n$, für obige physikalische Eigenschaften stehen, so existieren explizite funktionale Abhängigkeiten $y_{ji} = f_j(x_i),\ i = 1, ..., N,\ j = 1, ..., n$. Die Korrelationsfunktionen f_j sind differenzierbar und nicht mit Rauschen behaftet. Daher ist es möglich, künstlich ein kontrolliertes Rauschen auf f_j zu addieren und somit das Verhalten der einzelnen Optimierungsverfahren in Bezug auf Rauschen zu studieren. Es ist jedoch zu beachten, daß diese Korrelationsfunktionen nur für bestimmte physikalische Eigenschaften und ein bestimmtes Kraftfeldmodell hergeleitet worden sind. Sie sind nur anwendbar auf Phasenübergangsdaten, das heißt, das System wird im Gleichgewicht von flüssiger und gasförmiger Phase, also auf einer bestimmten Temperatur-Druck-Kurve, betrachtet. Bevor die Vorgehensweise zur Bewertung der Optimierungsverfahren angesprochen wird, werden im folgenden die Korrelationsfunktionen im einzelnen beschrieben.

Das chemische Modell, auf dem die Korrelationsfunktionen basieren, ist das sogenannte *Quadrupolare Zweizentren-Lennard-Jones-Modell* (englisch: *Two–Center Lennard–Jones Quadrupolar (2CLJQ) Model*). Es ist anwendbar auf kleine Moleküle, die aus zwei Lennard-Jones-Zentren bestehen, deren Abstand gleich der *Elongation* L ist, und ein quadrupolares Moment Q besitzen, welches am Massenzentrum liegt und entlang der Molekülachse orientiert ist. Die Herleitung von Quadrupolpotentialtermen wurde bereits in Abschnitt 2.2.3 durchgeführt. Es gibt auch Korrelationsfunktionen für Zweizentren-Lennard-Jones-Teilchen mit einem Dipolmoment (2CLJD, siehe Stoll u. a. (2003a)). Da die beiden Modelle jedoch ähnlich sind und in dieser Arbeit nur das 2CLJQ-Modell verwendet wird, wird in diesem Abschnitt nur letzteres vorgestellt.
Das 2CLJQ-Modell beinhaltet vier zustandsunabhängige Parameter: die LJ-Parameter σ und ε sowie L und Q. Das Paarpotential u_{2CLJQ} ist gegeben durch

$$
\begin{aligned}
u_{\mathrm{2CLJQ}}(r_{ij}, \omega_i, \omega_j, L, Q^2) &:= \sum_{a=1}^{2}\sum_{b=1}^{2} 4\varepsilon \left[\left(\frac{\sigma}{r_{ab}}\right)^{12} - \left(\frac{\sigma}{r_{ab}}\right)^{6}\right] && (3.57)\\
&+ \frac{3}{4}\frac{Q^2}{||r_{ij}||^5}[1 - 5(\cos^2\theta_i + \cos^2\theta_j) - 15\cos^2\theta_i\cos^2\theta_j && (3.58)\\
&+ 2(\sin\theta_i \sin\theta_j \cos\phi_{ij} - 4\cos\theta_i\cos\theta_j)^2]. && (3.59)
\end{aligned}
$$

Dabei ist r_{ij} der Abstandsvektor zwischen den Zentren zweier Moleküle i und j, r_{ab} einer der vier LJ-Abstände, wobei a sich auf die beiden LJ-Zentren von Molekül i und b auf die von Molekül j bezieht. Weiterhin stellen ω_i und ω_j die Orientierungen der beiden Moleküle dar, wobei θ_i der azimuthale Winkel zwischen der Achse von Molekül i und der Verbindungslinie der beiden Molekülzentren und ϕ_{ij} der Winkel zwischen den Achsen von Molekül i und j ist. Dies ist in Abbildung 2.5 veranschaulicht.

Kritische Werte für Temperatur und Dichte sowie die Sättigungsdichte der flüssigen und gasförmigen Phase und der Dampfdruck sind mithilfe von Korrelationsfunktionen in Abhängigkeit von ε, σ, Q^2 und L berechenbar. In Stoll u. a. (2001) sind die Phasenübergangsdaten mittels Molekularer Simulationen für 30 verschiedene Zweizentren-LJ-Fluide, unter anderem Stickstoff und Kohlenstoffmonoxid, erhalten und durch glatte Korrelationsfunktionen approximiert worden. Die Koeffizienten sind für jede der oben erwähnten Eigenschaften verschieden und wurden durch nichtlineare Regression über einen großen Temperaturbereich ermittelt. Sei $y \in \{T_c, \rho_c, \rho_l, \rho_v, p_\sigma\}$. Sowohl y als auch die Kraftfeldparameter werden in reduzierter Form betrachtet, das heißt anstelle von $y(\varepsilon, \sigma, Q^2, L)$ wird $y^*(Q^{*2}, L^*)$ berechnet, wobei $Q^{*2} := Q^2/(\varepsilon\sigma^5)$ und $L^* := L/\sigma$. Der Erhalt von reduzierten Zielgrößen ist in Abschnitt 3.6.4 angegeben. Die reduzierten Funktionen $T_c^*(Q^{*2}, L^*)$ und $\rho_c^*(Q^{*2}, L^*)$ wurden als lineare Kombinationen von Elementarfunktionen angenommen, von denen eine eine Konstante c ist und alle anderen entweder von Q^{*2}, also $\psi_i\left(Q^{*2}\right)$, von L^*, also $\xi_i\left(L^*\right)$, oder von beiden Parametern, also $\chi_i\left(Q^{*2}, L^*\right)$, abhängen. Die Anzahl an Elementarfunktionen ist sowohl für die Q^{*2}- als auch für die L^*-Abhängigkeit auf zwei beschränkt. Für $y \in \{T_c^*, \rho_c^*\}$ gilt:

$$
y\left(Q^{*2}, L^*\right) = c + \sum_{i=1}^{\leq 2} \alpha_i \cdot \psi_i(Q^{*2}) + \sum_{j=1}^{\leq 2} \beta_j \cdot \xi_j(L^*) + \sum_{k=1}^{\leq 4} \gamma_k \cdot \chi_k(Q^{*2}, L^*), \qquad (3.60)
$$

wobei $c, \alpha_i, \beta_j, \gamma_k \in \mathbb{R}$. Der entscheidende Schritt besteht in der Wahl der Elementarfunktionen

ψ_i, ξ_i und χ_i. Für $\ell := L^*$ und $q := Q^{*2}$ können sie allgemein in der Form

$$o(\ell, q, k, l, m, \hat{c}) := \frac{\ell^k q^l}{\ell^m + \hat{c}} \tag{3.61}$$

dargestellt werden. Die Wahl von k, l, m und $\hat{c}$ hängt von y ab. Die Parameter wurden per Hand eingestellt, so daß die Korrelationsfunktion möglichst gut mit den Simulationsdaten übereinstimmte. Weiterhin gilt stets $c \in \{0, 1\}$. Eine Tabelle sämtlicher Elementarfunktionen ist im Anhang von Stoll u. a. (2001) zu finden.
Nach Guggenheim (1945) ist die Dichte-Temperatur-Abhängigkeit in der Nähe des kritischen Punkts gegeben durch $\rho \sim (T_c - T)^{1/3}$. In reduzierten Größen, also $\rho^* = \rho\sigma^3$ und $T^* = Tk_B/\varepsilon$, können die Korrelationsfunktionen für ρ_l und ρ_v folgendermaßen geschrieben werden:

$$\rho_l^* = \rho_c^* + C_1 \cdot (T_c^* - T^*)^{1/3} + C_2' \cdot (T_c^* - T^*) + C_3' \cdot (T_c^* - T^*)^{3/2}, \tag{3.62}$$
$$\rho_v^* = \rho_c^* - C_1 \cdot (T_c^* - T^*)^{1/3} + C_2'' \cdot (T_c^* - T^*) + C_3'' \cdot (T_c^* - T^*)^{3/2}. \tag{3.63}$$

Durch das simultane Anpassen von ρ_l und ρ_v werden nicht nur die Regressionskoeffizienten C_1, C_2', C_3', C_2'', $C_3'' \in \mathbb{R}$, sondern auch die kritischen Daten ρ_c^* und T_c^* erhalten. Sowohl die Koeffizienten als auch die kritischen Daten sind gemäß Gleichung (3.60) als Linearkombinationen aus den Elementarfunktionen (3.61) zu erhalten, so daß letztendlich eine Abhängigkeit von L^* und Q^{*2} besteht.
Für den Dampfdruck sieht die Korrelationsfunktion wie folgt aus:

$$\ln p_\sigma^*(Q^{*2}, L^*, T^*) = c_1(Q^{*2}, L^*) + \frac{c_2(Q^{*2}, L^*)}{T^*} + \frac{c_3(Q^{*2}, L^*)}{T^{*4}}, \tag{3.64}$$

wobei $p_\sigma^* = p\sigma^3/\varepsilon$. Die Koeffizienten $c_1(Q^{*2}, L^*)$, $c_2(Q^{*2}, L^*)$, $c_3(Q^{*2}, L^*) \in \mathbb{R}$ werden wieder wie oben ermittelt.
Auch die reduzierte Verdampfungsenthalpie $\Delta_v H^* := \Delta_v H/\varepsilon$ kann nun berechnet werden, und zwar mithilfe der *Clausius-Clapeyron-Gleichung*

$$\frac{\partial \ln p_\sigma^*}{\partial T^*} = \frac{\Delta h_v^*}{p_\sigma^* T^* \left(1/\rho_v^* - 1/\rho_l^*\right)}, \tag{3.65}$$

wobei $(\partial \ln p_\sigma^*)/(\partial T^*)$ analytisch gemäß Gleichung (3.64) berechnet werden kann. Es ist zu beachten, daß die Korrelationsfunktion für die Dampfdichte nach Gleichung (3.63) nur für $T/T_c \geq 0.7$ gültig ist. Daher wird für niedrigere Tempetaturen die ideale Gasgleichung $\rho_v^* = p_\sigma^*/T^*$ verwendet und dies anstelle von ρ_v^* in die Clausius-Clapeyron-Gleichung (3.65) eingesetzt.

Zum effizienten Testen der verschiedenen Optimierungsverfahren werden Molekulare Simulationen durch diese Korrelationsfunktionen ersetzt. Zusätzlich wird statistisches Rauschen aufaddiert. Typische Unsicherheiten, das heißt Fehler in Bezug auf die wahren thermodynamischen Durchschnitte, liegen bei 0.5% für Dichten, 1.0% für Enthalpien und 3.0% bei Drücken (vergleiche Stoll u. a. (2003a)). Die aus einer Simulation errechneten Durchschnitte sind typischerweise normalverteilt um den eigentlichen Mittelwert mit spezifischen Standardabweichungen. Da die Aufaddierung von künstlichem Rauschen jedoch zur Bewertung der Optimierungsverfahren dienen soll, werden gleichverteilte Zufallszahlen verwendet, um auch in diesem viel ungünstigeren

Fall das Verhalten der einzelnen Verfahren zu studieren. Eine Normalverteilung kann nämlich oft nur theoretisch angenommen werden. In der Praxis müssen die Verfahren auch bei ungünstigerem Rauschen anwendbar sein. Also wird anstatt einer Eigenschaft y, welche mittels der Korrelationsfunktionen berechnet wird, ein verrauschtes $y+\Delta y$ genommen, wobei Δy gleichverteilt in $[y-\vartheta y, y+\vartheta y]$, mit $\vartheta \in \{0.005, 0.01, 0.03\}$, je nachdem, ob Dichte, Verdampfungsenthalpie oder Dampfdruck betrachtet wird.

3.5.3 Gestalt der Fehlerfunktion und Abbruchkriterium

Die Fehlerfunktion aus Gleichung (3.2) ist eine nicht-negative reellwertige Funktion $F : \mathbb{R}^N \to \mathbb{R}_0^+$, deren Verlauf *a priori* nicht erkennbar ist. Die Problematik liegt in der Verwendung der Teilfunktionen $f_i^{\text{sim}}(x),\ i = 1, ..., n$, die für die betrachteten physikalischen Eigenschaften in Abhängigkeit von den zu optimierenden Kraftfeldparametern x stehen. Diese Funktionen werden im allgemeinen mittels Molekularer Simulationen ausgewertet, was über thermodynamische Durchschnitte erfolgt. Bei MD-Simulationen werden diese Durchschnitte über einen gewissen Zeitraum und bei MC-Simulationen über eine gewisse Anzahl an Konfigurationen im Phasenraum (Stichproben) ermittelt. Daher sind diese Funktionen mit statistischem Rauschen behaftet und besitzen keinen analytischen Funktionsterm. Somit sind auch Eigenschaften wie Stetigkeit oder Differenzierbarkeit mathematisch nicht beweisbar, sondern bestenfalls argumentativ als gegeben vorauszusetzen, falls das Rauschen in irgendeiner Form eliminiert werden kann. In Bourasseau u. a. (2003) wurden für gewisse physikalische Zielgrößen Formeln für deren partielle Ableitungen nach den Kraftfeldparametern gefunden. Falls alle $f_i^{\text{sim}},\ i = 1, ..., n$, stetig differenzierbar sind, so ist auch F trivialerweise differenzierbar, da es sich um eine gewichtete Summe aus quadratischen Funktionen handelt.

Um dennoch einen groben Eindruck von F zu gewinnen, wurden die Funktionen $f_i^{\text{sim}},\ i = 1, ..., n$, mithilfe der Korrelationsfunktionen aus Abschnitt 3.5.2 ausgewertet. Bei der Fehlerfunktion handelt es sich dabei um eine Funktion

$$\begin{aligned} F: \quad \mathbb{R}^4 &\to \mathbb{R}_0^+ \\ (\varepsilon, \sigma, Q^2, L)^T &\mapsto F\left((\varepsilon, \sigma, Q^2, L)^T\right) \geq 0. \end{aligned}$$

Die Korrelationsfunktionen sind auf ihrem Definitionsbereich stetig differenzierbar, da die Elementarfunktionen (3.61) für $\ell > 0$ stetig differenzierbar sind. Die Glattheit von F ist in diesem Fall also gewährleistet. Es wird nun zwischen acht Optimierungsproblemen unterschieden, welche die folgenden physikalischen Eigenschaften simultan betrachten:

1. ρ_l und $\Delta_v H$ bei einer Temperatur,
2. ρ_l und $\Delta_v H$ bei einer Temperatur mit künstlichem Rauschen,
3. ρ_l und $\Delta_v H$ bei mehreren Temperaturen,
4. ρ_l und $\Delta_v H$ bei mehreren Temperaturen mit künstlichem Rauschen,
5. ρ_l und p_σ bei einer Temperatur,
6. ρ_l und p_σ bei einer Temperatur mit künstlichem Rauschen,

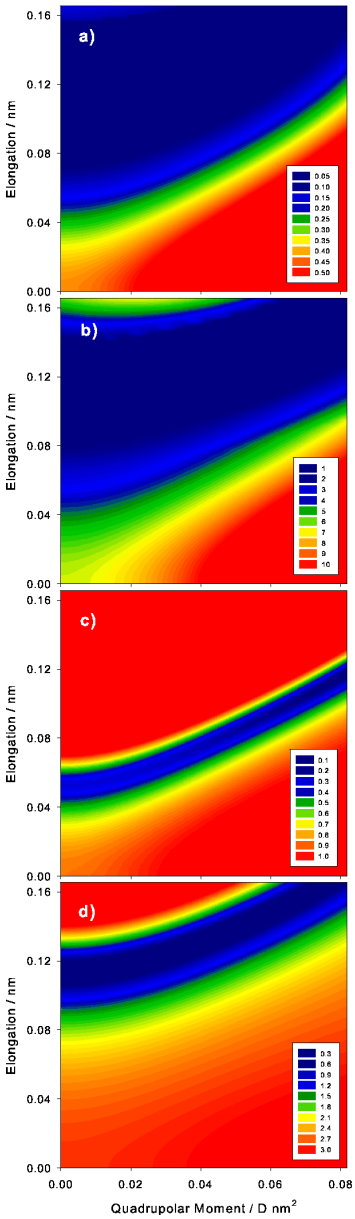


Abbildung 3.11: Konturplots der projizierten Fehlerfunktion im Falle des 2CLJQ-Potentials, entnommen aus Hülsmann u. a. (2010b). Die Lennard-Jones-Parameter wurden auf $\varepsilon = 0.3101$ kJ/mol und $\sigma = 0.331$ nm festgesetzt. Das quadratische quadrupolare Moment Q^2 und die Elongation L sind variabel. Die Einzelgraphiken a)-d) entsprechen den Optimierungsproblemen 1, 3, 5 und 7. Im Falle von mehreren Temperaturen nimmt die Fehlerfunktion viel höhere Werte an als im Falle von einer Temperatur. Eine dreidimensionale steile Regenrinne, wo sich womöglich das Minimum befindet, kann in allen vier Fällen beobachtet werden.

7. ρ_l und p_σ bei mehreren Temperaturen,

8. ρ_l und p_σ bei mehreren Temperaturen mit künstlichem Rauschen.

Abbildung 3.11 zeigt die Konturplots der in den $\mathbb{R}^2$ projizierten Fehlerfunktion. Dabei wurden σ und ε festgehalten sowie Q^2 und L variiert. Die Graphiken a)-d) entsprechen den Optimierungsproblemen 1, 3, 5 und 7. Es ist jedoch zu beachten, daß derartige Projektionen noch nichts über die allgemeine Fehlerfunktion aussagen. Allerdings wird ein Eindruck darüber gewonnen, wie groß der Wertebereich sein kann. Nennenswert sind weiterhin die Bereiche, wo F die Gestalt einer steilen Regenrinne besitzt, was für quadratische Fehlerfunktionen nicht untypisch ist. Am Boden dieser Regenrinne können mehrere lokale Minima und ein globales Minimum liegen.

Die Gestalt der Fehlerfunktion führt zu der Definition eines geeigneten Abbruchkriteriums für die numerischen Optimierungsalgorithmen: Wie bereits in Abschnitt 3.4.2 angesprochen, wird die Armijo-Schrittweite schnell zu klein. Da $\beta_A^\ell \overset{\ell\to\infty}{\to} 0$ und aufgrund der Randbedingungen bereits β_A^1 in Relation zur Größenordnung der Kraftfeldparameter äußerst klein sein wird, wird die resultierende Schrittweite t_A irgendwann zu keiner signifikanten Reduktion der Fehlerfunktion mehr führen. Ohne einen sehr hohen Rechenaufwand ist somit bei Verwendung der Armijo-Schrittweite keine Konvergenz gegen ein x^{opt} mit $\nabla F(x^{\text{opt}}) = 0$ zu erwarten. Da es in der Nähe des Minimums für die Armijo-Schrittweite umso schwieriger ist, eine signifikante Reduktion zu erzielen, wird die Schrittweitensteuerung ab einem bestimmten Punkt nicht mehr innerhalb einer akzeptablen Anzahl an Iterationen zum Ziel führen. Die nach einer sehr hohen Anzahl an Iterationen gefundenen Schrittweiten würden in Relation zur Norm der Abstiegsrichtung zu klein sein, was zu einer extrem langsamen Konvergenz des gesamten Optimierungsalgorithmus führen würde.
Das oben angesprochene Regenrinnenphänomen stellt diesbezüglich ein weiteres Problem dar: Gradientenbasierte Verfahren realisieren lediglich den funktionalen Abstieg an den Wänden der Regenrinne. Ein Gefälle am Boden wird somit gar nicht oder nur schwach registriert, was bereits in Roweis (1996) diskutiert wurde. Der Grund dafür liegt in der Tatsache, daß die Abstiegskomponente in Richtung der Wände viel größer ist und somit einen viel höheren Einfluß hat. Eine Möglichkeit, die Konvergenz in der Nähe des Minimums zu beschleunigen, ist die Kombination mit dem auf Levenberg (1944) und Marquardt (1963) zurückgehenden *Levenberg-Marquardt-Algorithmus*. In der vorliegenden Dissertation werden jedoch andere Verfahren eingesetzt, um dieses Problem zu lösen. Ein weiteres großes Problem besteht darin, daß der Gradient in der Nähe des Minimums aufgrund des statistischen Rauschens von einem gewissen Punkt an nicht mehr korrekt berechnet werden kann. Zeigt er in eine falsche Richtung, so sind sämtliche gradientenbasierte Optimierungsverfahren erfolglos. Daher wird später auch ein ableitungsfreies Verfahren als Alternative zu gradientenbasierten Verfahren verwendet.
Aus den genannten Gründen kann keine schnelle Konvergenz erwartet werden, wenn das Abbruchkriterium

$$||\nabla F(x)|| \leq \tau \tag{3.66}$$

lautet, wobei $\tau > 0$, zum Beispiel $\tau = 10^{-3}$. Bei der vorliegenden Problemstellung ist es ratsamer, mit weniger Rechenaufwand zu einem weniger strikten Abbruchkriterium zu gelangen. Daher wird hier ein Abbruchkriterium verwendet, welches sich nur auf die Fehlerfunktion selbst

und nicht auf ihren Gradienten bezieht, und zwar

$$F(x) \leq \tau, \tag{3.67}$$

wobei $\tau = 10^{-4}$ oder sogar $\tau = 10^{-5}$. Der erreichbare Wert τ hängt von der Gestalt der Fehlerfunktion ab, das heißt von den betrachteten physikalischen Eigenschaften, deren Anzahl und der Anzahl an betrachteten Temperaturen sowie von der Größenordnung des Rauschens.

3.5.4 Behandlung von Rauschen in den Systemeigenschaften bei Molekularen Simulationen

Wie aus Abschnitt 3.5.3 hervorgeht, besteht die Hauptproblematik der Optimierung darin, daß die zu minimierende Fehlerfunktion nicht analytisch darstellbar ist. Eigenschaften wie Stetigkeit und Differenzierbarkeit sind nicht *a priori* gegeben, und der Verlauf des Funktionsgraphen kann sehr komplex sein, das heißt, die Fehlerfunktion ist möglicherweise stark zerklüftet. Hinzu kommt die Tatsache, daß die Funktion mit statistischem Rauschen behaftet ist, was die Verwendung gradientenbasierter Verfahren zusätzlich erschwert. Dies muß sowohl bei der Auswertung der Simulation als auch bei der Auswahl des Algorithmus selbst berücksichtigt werden. Dieser Abschnitt befaßt sich mit der Behandlung des Rauschens in den Simulationsläufen. Die Idee dabei ist, das Rauschen möglichst gering zu halten, ohne unnötig lange zu simulieren. Resultiert aus der in Abschnitt 3.5.3 angesprochenen Verfahrensbewertung, daß gewisse Schranken für die statistischen Unsicherheiten toleriert werden können, so können diese Schranken in das Abbruchkriterium der Simulation einfließen. Das verbleibende geringfügige Rauschen wird innerhalb des Optimierungsalgorithmus selbst behandelt, was in Abschnitt 3.5.5 diskutiert wird. Innerhalb einer Simulation ist es zunächst das Ziel, das zugrundeliegende System zu äquilibrieren, so daß die gewünschten physikalischen Eigenschaften mittels thermodynamischer Durchschnitte in einem anschließenden Schritt berechnet werden können. Die Theorie hierzu wird in Anhang D beschrieben, die praktische Umsetzung sei im folgenden erläutert.

Zunächst wird eine geeignete Startkonfiguration gewählt, in der Regel eine räumliche Gitterstruktur, auf der die Teilchen angeordnet sind. Sinnvoll sind auch randomisierte Konfigurationen, da eine Gitterstruktur von Flüssigkeiten unphysikalisch ist und gegebenenfalls eine zu lange Simulation durchzuführen ist, bis eine angemessene Struktur erreicht wird. Bei einer randomisierten Struktur ist jedoch zu beachten, daß Überlappungen zu vermeiden sind.
Nachdem die aktuellen Kraftfeldparameter in die Topologie aufgenommen wurden, wird eine Simulation gestartet, welche aus den folgenden Teilschritten besteht, die zunächst für MD-Simulationen beschrieben werden:

1. **Energieminimierung:** Aufgrund von ungeeigneten initialen Startkonfigurationen können die Kräfte innerhalb des Systems und somit die Beschleunigung der Teilchen zu groß sein, so daß überschüssige Energien vorhanden sind. Um dem entgegenzuwirken, wird stets eine Energieminimierung vorgeschaltet, bei der die Anfangsstruktur verlassen und stattdessen eine Struktur berechnet wird, bei der ein lokales Minimum der Potentialhyperfläche vorliegt. Bei letzterer handelt es sich um eine Funktion

$$\begin{aligned} \mathcal{E} : \mathbb{R}^{3N} &\rightarrow \mathbb{R} \\ r^N &\mapsto \mathcal{E}(r^N), \end{aligned} \tag{3.68}$$

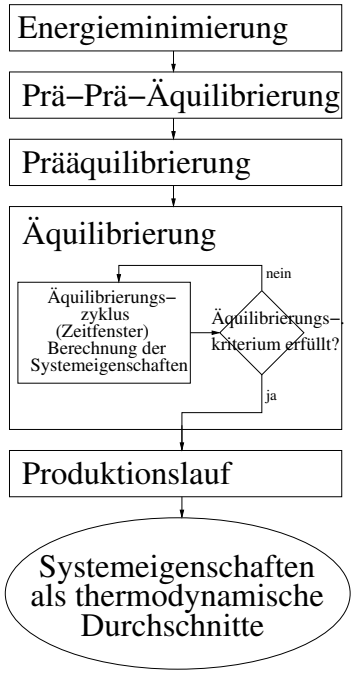


Abbildung 3.12: Teilsimulationen innerhalb einer Gesamtsimulation zur Berechnung von äquilibrierten Systemeigenschaften als thermodynamische Durchschnitte zur Minimierung des statistischen Rauschens.

wobei N die Anzahl an Molekülen und r^N deren Positionsvektor bezeichnet. Die Geometrie wird bei der Energieminimierung iterativ verändert. Es wird wie gesagt stets nach einem lokalen Minimum gesucht, was mit einer Kombination aus der Methode des steilsten Abstiegs, einem CG-Verfahren und dem Newton-Raphson-Verfahren erzielt werden kann.

2. **Prä-Prä-Äquilibrierung:** Hierbei handelt es sich um kein Standardverfahren, sondern um einen Zusatz, der sich in der Praxis zum Beispiel bei ionischen Systemen als günstig erwiesen hat (vergleiche Köddermann (2008)): Um lange Rechenzeiten bei der nachfolgenden Präääquilibrierung zu vermeiden, werden zunächst wenige Zeitschritte mit einer kleinen Zeitschrittweite betrachtet. Eine feinere Diskretisierung erhöht die Genauigkeit der Simulation und verhindert, daß das System mechanisch instabil wird. Dabei wird ein NVT-Ensemble angenommen, das heißt, das Volumen wird vorgegeben. Bei Systemen, die in der Nähe von Phasenübergängen simuliert werden, muß dies jedoch angepaßt werden, zum Beispiel durch die Erhöhung der Anzahl an Zeitschritten, da hierbei die Wahrscheinlichkeit eines mechanisch instabilen Systems größer ist. Im Falle einer Veränderung des Aggregatszustands muß den Teilchen die Gelegenheit gegeben werden, wieder in den flüssigen Zustand zurückzukehren. Die Simulationszeit muß dafür erhöht und das Boxvolumen groß genug gewählt werden, so daß der Druck und somit die Kräfte innerhalb des Systems nicht derart hoch werden, daß zu Beginn einer anschließenden NPT-Simulation, bei der das Volumen wieder variiert wird, ein instabiler Zustand zu erwarten ist.

3. **Präääquilibrierung:** Um das System in eine Konfiguration zu überführen, die möglichst weit weg von einer möglicherweise ungünstigen initialen Struktur und idealerweise bereits

nahe am Gleichgewicht liegt, wird eine Prääquilibrierung durchgeführt. Es wird dabei ein NPT-Ensemble auf einem Zeitfenster geeigneter Länge simuliert. Letztere ist systemabhängig.

4. **Äquilibrierung:** Die nachfolgende Äquilibrierung besteht aus verschiedenen Zyklen. Dabei handelt es sich um Zeitfenster derselben Größe, bei kleinen Molekülen zumeist $2.5 \cdot 10^5$ Zeitschritte mit einer Zeitschrittweite von 2 fs. Nach jedem Äquilibrierungszyklus wird ein Äquilibrierungstest durchgeführt, welcher entscheidet, ob sich die aktuellen Durchschnitte von den tatsächlichen Durchschnitten noch signifikant unterscheiden oder nicht. Ist der Fehler gering, so endet der Äquilibrierungslauf, und die aus dem nachfolgenden Produktionslauf errechneten Durchschnitte können als gute Näherungen verwendet werden. Das folgende Lemma liefert ein geeignetes Äquilibrierungskriterium:

Lemma 3.5.2 (Zeitabhängige statistische Fehlerfunktion)**.** *Es seien X eine bestimmte physikalische Eigenschaft und $\mu := \langle X \rangle$ deren thermodynamischer Durchschnitt. Es seien weiterhin $E(X,M) = \frac{1}{M}\sum_{i=1}^{M} X_i$ der für μ näherungsweise berechnete Mittelwert und die Stichprobenfolge $X_1, X_2, ..., X_M$, $M \in \mathbb{N}$, so gewählt, daß X_i und X_j für alle $i < j$, $i, j = 1, ..., M$, statistisch unabhängig sind. Dann ist die zeitabhängige statistische Fehlerfunktion $\mathcal{F}(X,M) := \sqrt{\langle (E(X,M) - \mu)^2 \rangle}$ gegeben durch*

$$\mathcal{F}(X,M) = \frac{\sigma(X)}{\sqrt{M}}, \tag{3.69}$$

wobei $\sigma(X)$ die Standardabweichung der Eigenschaft X bezeichnet.

Beweis: Es gilt:

$$\begin{aligned}
\mathcal{F}(X,M)^2 &= \langle (E(X,M) - \mu)^2 \rangle = \langle E(X,M)^2 \rangle - 2\underbrace{\langle E(X,M) \rangle}_{=\mu}\mu + \mu^2 \\
&= \left\langle \frac{1}{M^2} \sum_{i,j=1}^{M} X_i X_j \right\rangle - \mu^2 = \left\langle \frac{1}{M^2} \sum_{i=1}^{M} X_i^2 \right\rangle + 2\left\langle \frac{1}{M^2} \sum_{i<j} X_i X_j \right\rangle - \mu^2 \\
&= \frac{1}{M}\langle X^2 \rangle + \frac{2}{M^2} \sum_{i<j} \langle X_i \rangle \langle X_j \rangle - \mu^2 = \frac{1}{M}\langle X^2 \rangle + 2\frac{M(M-1)}{2M^2}\underbrace{\langle X \rangle^2}_{=\mu^2} - \mu^2 \\
&= \frac{1}{M}\langle X^2 \rangle + \left(1 - \frac{1}{M}\right)\mu^2 - \mu^2 = \frac{1}{M}\left(\langle X^2 \rangle - \langle X \rangle^2\right) \\
&= \frac{1}{M}\mathrm{Var}(X).
\end{aligned}$$

Da die Wurzel aus der Varianz gleich der Standardabweichung ist, ist Gleichung (3.69) bewiesen. □

Da nach Lemma 3.5.2 $\mathcal{F}(X,M) \overset{M\to\infty}{\to} 0$, wird als Äquilibrierungskriterium

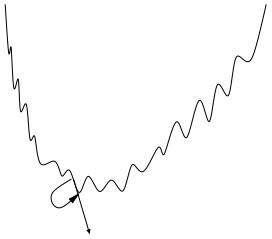
$$\frac{\sigma(X)}{\sqrt{M}} \leq \vartheta E(X,M) \tag{3.70}$$

verwendet, wobei nach Abschnitt 3.5.2 in der Regel $\vartheta = 0.005$ für Dichten, $\vartheta = 0.01$ für Energien und Enthalpien und $\vartheta = 0.03$ für Drücke gewählt werden. Es sind stets diejenigen physikalischen Eigenschaften in das Äquilibrierungskriterium miteinzubeziehen, die optimiert werden sollen. Weiterhin wird stets die potentielle Energie betrachtet, da das System nicht als äquilibriert bezeichnet werden kann, solange diese noch erheblich schwankt.
Ein weiteres Problem stellt die statistische Unabhängigkeit der Stichproben dar, die in die Berechnung des Mittelwerts $E(X, M)$ miteinbezogen werden. Wird die Folge $X_t,\ t = 1, 2, ...$ als Zeitreihe aufgefaßt, so kann mithilfe ihrer Autokorrelationsfunktion festgestellt werden, ab welcher Verschiebung unabhängige Zeitreihenwerte zu erhalten sind. Da die Überprüfung auf statistische Unabhängigkeit im allgemeinen jedoch aufwendig ist und es bei der vorliegenden Problematik nicht von Belang ist, eine genaue Korrelationszeit zu bestimmen, wird ein bestimmter Schwellenwert hierfür verwendet. Im allgemeinen kann angenommen werden, daß nach jedem tausendsten Zeitschritt die entsprechenden Eigenschaften nahezu unabhängig sind. Bei einem Äquilibrierungszyklus der Länge $2.5 \cdot 10^5$ bedeutet dies, daß 250 Werte in die Berechnung des Mittelwerts eingehen. Verfahren zur genauen Bestimmung dieses Schwellenwerts b werden in Allen u. Tildesley (1987) diskutiert. In der Praxis hat sich $b = 10^3$ stets als geeignet herausgestellt.

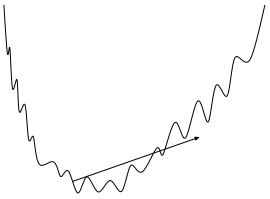
5. **Produktionslauf:** Da die (Prä-)Äquilibrierung äußerst akkurat durchgeführt wurde, ist ein langer Produktionslauf in vielen Fällen nicht mehr notwendig. Die endgültigen Durchschnitte werden nun über eine einzige Simulation bestimmt, die gegebenenfalls sogar kürzer als ein Äquilibrierungszyklus sein kann. Allerdings ist zu beachten, daß auch hierbei eine ausreichend gute Statistik vorhanden sein muß, um einen akkuraten Mittelwert zu bestimmen.

Die gesamte Simulation zur Berechnung eines thermodynamischen Durchschnitts von der Energieminimierung bis zur Produktion ist in Abbildung 3.12 veranschaulicht. Um den Äquilibrierungsprozeß zu beschleunigen, werden zumeist Bindungen und zum Teil auch Winkel starr gehalten. Vor allem bei kleinen Molekülen können starre Bindungen angenommen werden, was dazu führt, daß eine Äquilibrierung der intramolekularen Kräfte nicht mehr erforderlich ist, so daß das Gesamtsystem schneller ins Gleichgewicht gebracht werden kann. Hierzu werden im Falle von MD-Simulationen die in Anhang B erläuterten Verfahren eingesetzt, wie zum Beispiel SHAKE oder LINCS.

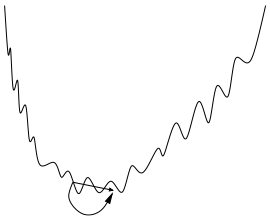
Eine Besonderheit ergibt sich beim Selbstdiffusionskoeffizienten D: Gemäß Gleichung (C.18) wird dieser mithilfe der mittleren quadratischen Verschiebung $\langle (r(t) - r(0))^2 \rangle$ berechnet. Das bedeutet, daß die Berechnung von D erst nach einer ausreichenden Anzahl an Zeitschritten möglich ist, denn bei der mittleren quadratischen Verschiebung handelt es sich um einen thermodynamischen Durchschnitt über einen bestimmten Zeitraum. Ein Auflisten von D zu jedem beliebigen Zeitpunkt t ist daher nicht möglich. Daher wird D zunächst nach dem Produktionslauf für die anderen Eigenschaften berechnet, indem über einen ausreichend langen Zeitraum simuliert wird, welcher um ein Vielfaches größer als ein Äquilibrierungszyklus ist. Eine weitere Möglichkeit ergibt sich durch das Auflisten von D über Zeitblöcke der Länge B, welche disjunkt und aufeinanderfolgend gewählt werden können. Eine andere Möglichkeit besteht darin, Zeitblöcke festzulegen, die stets bis zum Anfangszeitpunkt zurückgehen und somit monoton wachsen. Ein Nachteil letzterer Methode ist die Tatsache, daß in die Berechnung des Mittel-



(a) Wahl eines zu kleinen $h > 0$: Die Abstiegsrichtung ist überangepaßt und sorgt für die Konvergenz gegen ein durch Oszillationen hervorgerufenes lokales Minimum.



(b) Wahl eines zu großen $h > 0$: Dabei besteht die Gefahr, daß keine Abstiegsrichtung gefunden werden kann, da diese komplett falsch berechnet wird.



(c) Wahl eines idealen $h > 0$: Der Trend der Fehlerfunktion wird korrekt reproduziert, und das Verfahren konvergiert gegen das lokale Minimum.

Abbildung 3.13: Verhalten von gradientenbasierten Optimierungsverfahren bei unterschiedlicher Wahl von h: Wahl eines zu kleinen (a), zu großen (b) und eines idealen (c) $h > 0$.

werts stets die Anfangswerte miteinbezogen werden, welche weit vom eigentlichen Durchschnitt entfernt sind, was zu einer langsameren Äquilibrierung führt. Nur im Falle einer Auflistung über Zeitblöcke kann das Äquilibrierungskriterium (3.70) auch auf den Diffusionskoeffizienten angewandt werden. Diese Überlegungen gelten auch für andere Transporteigenschaften wie zum Beispiel die Viskosität.

Bei MC-Simulationen erfolgt eine Äquilibrierung nach demselben Prinzip mit dem einzigen Unterschied, daß Stichproben von bestimmten Systemkonfigurationen anstelle von Zeitfenstern betrachtet werden. Eine Prä-Prä-Äquilibrierung ist in obigem Sinne jedoch nicht möglich, da diese die Reduktion der Zeitschrittweite in Betracht zieht. Stattdessen ist es möglich, anfangs für kleinere Akzeptanzwahrscheinlichkeiten zu sorgen.

3.5.5 Effekt des Rauschens auf die Fehlerfunktion: Wahl der Diskretisierung für den Gradienten

Sei $\mu(x)$ ein beliebiger Mittelwert in Abhängigkeit eines Kraftfeldparametervektors x. Angenommen, die Funktion $x \mapsto \mu(x)$ sei mindestens zweimal stetig differenzierbar, und es existiere ein kompaktes Gebiet Ω, so daß die Funktion in Ω ein globales Minimum besitzt. Liegt der Startvektor x^0 im Einzugsbereichs dieses Minimums, so werden die hier betrachteten numerischen Optimierungsverfahren nur dann zum globalen Optimum führen, wenn kein Rauschen in den Funktionswerten vorhanden ist. Ansonsten werden die Algorithmen gegen ein durch Oszillationen hervorgerufenes lokales Minimum konvergieren, was in Abbildung 3.13 veranschaulicht ist. Das würde jedoch bedeuten, daß $E(x) = \mu(x)$ gelten müßte, das heißt, es müßte unendlich lange simuliert werden. Da jedoch der Gradient und die Hesse-Matrix durch finite Differenzen approximiert werden, ist es durch eine gröbere Diskretisierung möglich, die Verfahren auch bei Vorhandensein von Rauschen zum Ziel zu führen. Die Wahl von h in Gleichung (3.52) ist also ausschlaggebend. Abbildung 3.13 zeigt diesen Sachverhalt: Wird h zu klein gewählt, so ist die Abstiegsrichtung überangepaßt, und das Verfahren konvergiert gegen ein durch Oszillationen hervorgerufenes lokales Minimum (Abbildung 3.13(a)). Wird es zu groß gewählt, so kann die Richtung des negativen Gradienten komplett falsch berechnet werden (Abbildung 3.13(b)). Der Parameter h muß so gewählt werden, daß der absteigende Trend der Fehlerfunktion korrekt wie-

dergegeben wird und somit durch Rauschen hervorgerufene lokale Minima übersprungen werden (Abbildung 3.13(c)). Dies funktioniert allerdings nur in ausreichend steilen Bereichen. Die Richtung des Gradienten kann komplett falsch berechnet werden. Dies macht die Verwendung von gradientenbasierten Methoden äußerst schwierig, und es ist *a priori* keinesfalls gewährleistet, daß diese Verfahren überhaupt auf die vorliegende Problematik anwendbar sind. Die in Abschnitt 3.5.2 angesprochene Verfahrensbewertung kann zwar gewisse eigenschaftsabhängige obere Schranken für die statistischen Unsicherheiten liefern, es kann jedoch nicht ohne weiteres vorhergesagt werden, ab welchem Ausmaß an statistischem Rauschen die Verfahren nicht mehr zum Ziel führen. Eine obere Grenze ist stets abhängig von der Gestalt der Fehlerfunktion und könnte höchstens durch geeignete statistische Analysen, das heißt durch vielfache Wiederholung von Experimenten mit Unsicherheiten verschiedenen Ausmaßes, geschätzt werden. Da jedoch für die meisten hier betrachteten physikalischen Zielgrößen allgemeine Toleranzwerte bekannt sind, ist lediglich sicherzustellen, daß das dadurch schätzbare Ausmaß an Rauschen für gradientenbasierte Verfahren tolerabel ist. Daher wird in dieser Arbeit von einer detaillierten statischen Analyse bezüglich des möglichen Ausmaßes an Rauschen abgesehen. In diesem Abschnitt wird vielmehr eine theoretische Betrachtung geliefert, das heißt, es wird analysiert, inwieweit sich Unsicherheiten auf den betrachteten physikalischen Eigenschaften auf die gesamte Fehlerfunktion auswirken. Anschließend wird hergeleitet, wie h aufgrund dieser Auswirkung geeignet gewählt werden kann, und es wird ein Algorithmus zur Optimierung von h angegeben, welcher jedoch bei Molekularen Simulationen wegen des zu hohen Rechenaufwands in der Praxis nicht durchgeführt werden kann. Es ist klar, daß die numerischen Optimierungsverfahren im Falle von Rauschen nie exakt gegen ein Minimum konvergieren werden. In der Nähe des Minimums tritt irgendwann der in Abbildung 3.13(b) veranschaulichte Fall ein. Das Ziel ist es daher, in einen Bereich zu gelangen, wo sämtliche physikalische Eigenschaften bis auf ihre statistischen Unsicherheiten genau erhalten werden können.

Es wird die Fehlerfunktion $F : \Omega \to \mathbb{R}_0^+$ betrachtet, wobei $\Omega \subset \mathbb{R}^N$ ein kompaktes Gebiet ist. Es sei weiterhin $f : \Omega \to \mathbb{R}_0^+$ eine bestimmte physikalische Eigenschaft. Der Einfachheit halber werden hier nur Eigenschaften betrachtet, die positive Werte annehmen. Ansonsten können Verschiebungen $f + z,\ z \in \mathbb{R}$, anstelle von f verwendet werden. Es wird das Rauschen an einer bestimmten Stelle $x \in \Omega$ untersucht. Die entsprechende verrauschte Funktion $\tilde{f}$ sei gegeben durch $\tilde{f}(x) := f(x) + \Delta f_x$, wobei $\Delta f_x = cf(x)$ mit $|c| < 1$. Da $f(x) \geq 0$, gilt $\operatorname{sign}(\Delta f_x) = \operatorname{sign}(c)$. Sei $0 < C < 1$ eine obere Schranke für $|c|$. Dann ist auch Δf_x beschränkt, denn es gilt:

$$-Cf(x) \leq \Delta f_x \leq Cf(x). \tag{3.71}$$

Die Länge der Simulation, welche durch das Äquilibrierungskriterium (3.70) kontrolliert wird, sorgt dafür, daß C möglichst klein ist.
Sei R der Referenzwert für $f(x)$. Dann ist die Fehlerfunkion F bezüglich der Eigenschaft f gegeben durch $F(x) := \left(\frac{f(x)-R}{R}\right)^2$. Wird der verrauschte Wert $\tilde{f}$ in F eingesetzt, so ergibt sich

die verrauschte Fehlerfunktion $\tilde{F}$ durch:

$$\begin{aligned}\tilde{F}(x) &:= \left(\frac{\tilde{f}(x)-R}{R}\right)^2 = \left(\frac{f(x)+\Delta f_x - R}{R}\right)^2 \\ &= \underbrace{\left(\frac{f(x)-R}{R}\right)^2}_{=F(x)} + 2\frac{(f(x)-R)\Delta f_x}{R^2} + \frac{\Delta f_x^2}{R^2}. \end{aligned} \tag{3.72}$$

Somit ergibt sich für den Fehler bezüglich F:

$$\begin{aligned}\Delta F_x := \tilde{F}(x) - F(x) &= 2\frac{(f(x)-R)cf(x)}{R^2} + \frac{c^2 f(x)^2}{R^2} \\ &= 2c\frac{f(x)^2 - 2Rf(x) + R^2}{R^2} + 2c\frac{f(x)}{R} - 2c + \frac{c^2 f(x)^2}{R^2} \\ &= 2c\left(F(x) + \frac{f(x)}{R} - 1\right) + \frac{c^2 f(x)^2}{R^2}. \end{aligned} \tag{3.73}$$

Das Ergebnis ist keinesfalls überraschend. Es handelt sich hierbei um die gewöhnliche Fehlerfortpflanzung für die Multiplikation: Der Parameter c wird verdoppelt, wirkt sich also additiv auf die Fehlerfuntion F aus. Weiterhin steht der Faktor $2c$ vor dem Funktionswert $f(x)$, allerdings auch mit negativem Vorzeichen vor einer Konstanten (in diesem Fall 1), so daß dieser zusätzliche Term teilweise wieder ausgelöscht wird. Ist c so klein, daß der quadratische Term $c^2 f(x)^2$ zu vernachlässigen ist, so ist der resultierende Fehler analog zur gewöhnlichen multiplikativen Fehlerfortpflanzung als gutartig zu bezeichnen. Es ist jedoch zu beachten, daß $\text{sign}(\Delta F_x)$ sowohl von $\text{sign}(c)$ als auch von c selbst sowie von $\frac{f(x)}{R}$ abhängig ist. Es kommt hierbei also auch darauf an, ob die simulierte physikalische Eigenschaft $f(x)$ größer oder kleiner ist als ihr zugehöriger Referenzwert. Wird Gleichung (3.73) umgestellt, so ergibt sich

$$\Delta F_x = 2cF(x) - 2c + \left(\frac{cf(x)}{R} + 1\right)^2 - 1. \tag{3.74}$$

Falls ein $0 < \bar{C}(x) < 1$ existiert mit

$$\left|\left(\frac{cf(x)}{R} + 1\right)^2 - 1\right| \leq \bar{C}(x), \tag{3.75}$$

so ist ΔF_x beschränkt, denn es gilt:

$$-2CF(x) - 2C - \bar{C}(x) \leq \Delta F_x \leq 2CF(x) + 2C + \bar{C}(x). \tag{3.76}$$

Falls $c \approx 0$ ist und $f(x)$ nicht zu weit von R entfernt, so ist auch $\bar{C}(x) \approx 0$, und die Auswirkung des Rauschens auf F ist als numerisch gutartig zu bezeichnen. Wegen

$$\left|\left(\frac{cf(x)}{R} + 1\right)^2 - 1\right| \leq \left(\frac{Cf(x)}{R} + 1\right)^2 - 1$$

wird $\bar{C}(x) := \left(\frac{Cf(x)}{R} + 1\right)^2 - 1 > 0$ gesetzt.
Für die relativen Fehler gilt:

$$\begin{aligned}
\frac{\Delta f}{f} &= \frac{cf}{f} = c && (3.77)\\
\frac{\Delta F}{F} &= \frac{\left(\frac{f+\Delta f-R}{R}\right)^2 - \left(\frac{f-R}{R}\right)^2}{\left(\frac{f-R}{R}\right)^2} \\
&= \frac{(f+\Delta f)^2 - 2R(f+\Delta f) + R^2 - f^2 + 2Rf - R^2}{(f-R)^2} \\
&= \frac{2f\Delta f + \Delta f^2 - 2R\Delta f}{(f-R)^2} = \frac{2(f-R)\Delta f + \Delta f^2}{(f-R)^2} \\
&= \frac{2\Delta f}{f-R} + \frac{\Delta f^2}{(f-R)^2}, && (3.78)
\end{aligned}$$

was ebenfalls die Gutartigkeit des resultierenden Rauschens bestätigt. Ist der quadratische Term vernachlässigbar, so ist der resultierende relative Fehler additiv. Es ist zu beachten, daß bis auf R alle Größen in Abschätzung (3.78) von x abhänging sind, was jedoch der Einfachheit halber weggelassen wurde.
Im folgenden sei $F^i := \left(\frac{f^i(x)-R^i}{R^i}\right)^2,\ i \in \{1, ..., n\}$, die quadratische Fehlerfunktion, welche sich auf eine bestimmte physikalische Eigenschaft $f^i(x)$ bezieht, und ΔF_x^i deren statistischer Fehler in Bezug auf x. Die oberen Schranken C^i und $\bar{C}^i(x)$ seien wie oben definiert und beziehen sich ebenfalls auf die Eigenschaft $f^i(x)$. Dann gilt für die verrauschte Gesamtfehlerfunktion, welche sich auf sämtliche Eigenschaften bezieht:

$$\begin{aligned}
\tilde{F}(x) &:= \sum_{i=1}^{n} \left(F^i(x) + \Delta F_x^i\right) \\
&= F(x) + \sum_{i=1}^{n} \Delta F_x^i, && (3.79)
\end{aligned}$$

das heißt, die statistischen Fehler addieren sich lediglich. Entsprechend ergibt sich für den relativen Fehler:

$$\begin{aligned}
\left|\frac{\Delta F}{F}\right| &= \frac{\left|\sum_{i=1}^n \Delta F^i\right|}{\sum_{i=1}^n F^i} \le \frac{\sum_{i=1}^n \left|\Delta F^i\right|}{\sum_{i=1}^n F^i} \\
\Rightarrow \left|\frac{\Delta F}{F}\right| &\le \frac{\sum_{i=1}^n \left|\Delta F^i\right|}{F^j} \\
&\overset{(3.76),\,(3.78)}{\le} \left|\frac{2\Delta f^j}{f^j - R^j} + \frac{(\Delta f^j)^2}{(f^j - R^j)^2}\right| + \frac{(R^j)^2\left(\sum_{i\neq j} 2CF^i + (n-1)(2C + \bar{C})\right)}{(f^j - R^j)^2} \\
&\le \frac{2|\Delta f^j||f^j - R^j| + (\Delta f^j)^2 + (R^j)^2\left(\sum_{i\neq j} 2CF^i + (n-1)(2C + \bar{C})\right)}{(f^j - R^j)^2}, \\
& j = 1, ..., n, && (3.80)
\end{aligned}$$

wobei $C := \max_{i=1}^{n} C^i$ und $\bar{C} = \bar{C}(x) := \max_{i=1}^{n} \bar{C}^i(x)$. Damit ist auch der relative Fehler durch einen additiven Fehlerterm sowie durch kleine obere Fehlerschranken abschätzbar. Es ist zu beachten, daß bis auf C und $R^i,\ i = 1, ..., n$, alle Größen in Abschätzung (3.80) von x abhänging sind, was jedoch der Einfachheit halber weggelassen wurde.
Die Auswirkung des Rauschens auf die Fehlerfunktion selbst ist also nicht gravierend, so daß bisher nichts gegen die Anwendung gradientenbasierter Verfahren spricht. Es bleibt jedoch zu zeigen, inwieweit sich das Rauschen auf die partiellen Ableitungen von F und somit auf den Gradienten auswirkt. Es sei zunächst wieder $F(x) := \left(\frac{f(x)-R}{R}\right)^2$ und $\Delta f_x, \tilde{f}$ sowie $\tilde{F}$ entsprechend. Es gilt zunächst:

$$\begin{aligned}
\frac{\partial \tilde{f}}{\partial x_i}(x) &= \frac{\tilde{f}(x_1, ..., x_i + h, ..., x_N) - \tilde{f}(x)}{h} + \mathcal{O}(h) \\
&= \frac{f(x_1, ..., x_i + h, ..., x_N) + \Delta f_{(x_1,...,x_i+h,...,x_N)} - f(x) - \Delta f_x}{h} + \mathcal{O}(h) \\
&= \frac{\partial f}{\partial x_i}(x) + \frac{\partial \Delta f_x}{\partial x_i},\ i = 1, ..., N. \qquad (3.81)
\end{aligned}$$

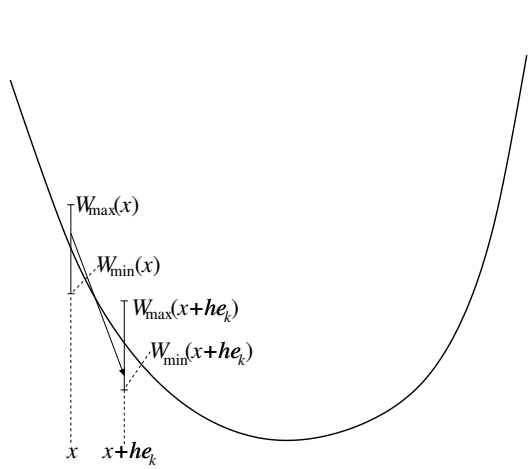
Somit folgt für die partielle Ableitung von $\tilde{F}$:

$$\begin{aligned}
\frac{\partial \tilde{F}}{\partial x_i}(x) &= \frac{2}{R^2}\left(\tilde{f}(x) - R\right)\frac{\partial \tilde{f}}{\partial x_i}(x) = \frac{2}{R^2}\left(f(x) + \Delta f_x\ - R\right)\frac{\partial \tilde{f}}{\partial x_i}(x) \\
&\overset{(3.81)}{=} \frac{2}{R^2}\left(f(x) - R\right)\frac{\partial f}{\partial x_i}(x) + \frac{2}{R^2}\left(f(x) - R\right)\frac{\partial \Delta f_x}{\partial x_i} \\
&+ \Delta f_x \frac{\partial f}{\partial x_i}(x) + \Delta f_x \frac{\partial \Delta f_x}{\partial x_i} \\
&= \frac{\partial F}{\partial x_i}(x) + \Delta f_x \frac{\partial f}{\partial x_i}(x) + \frac{\partial \Delta f_x}{\partial x_i}\left(\frac{2}{R^2}\left(f(x) - R\right) + \Delta f_x\right), \\
&\quad i = 1, ..., N. \qquad (3.82)
\end{aligned}$$

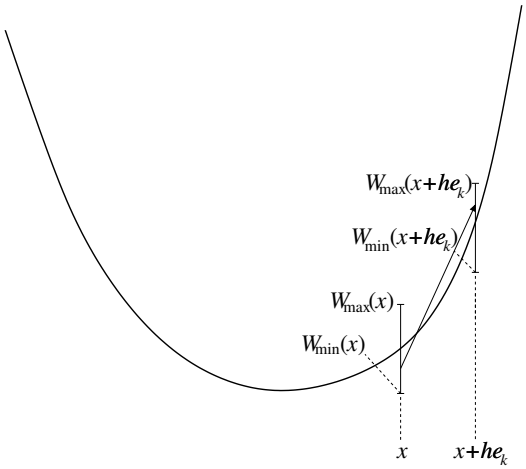
Somit gilt für den statistischen Fehler in Bezug auf $\frac{\partial F}{\partial x_i}(x)$:

$$\Delta\frac{\partial F}{\partial x_i}(x) = c\frac{\partial F}{\partial x_i}(x) + \left(c + c^2\right) f(x)\frac{\partial f}{\partial x_i}(x),\ i = 1, ..., N. \qquad (3.83)$$

Das Rauschen in Bezug auf die partiellen Ableitungen der Gesamtfehlerfunktion ist analog zu Gleichung (3.79) wieder gleich der Summe aus den einzelnen statistischen Unsicherheiten.
Der Fehler hängt also insbesondere von den Termen $c\frac{\partial F}{\partial x_i}(x)$ und $c\frac{\partial f}{\partial x_i}(x)$ ab. Da die partiellen Ableitungen mittels Differenzenquotienten approximiert werden, muß h so gewählt werden, daß diese Terme möglichst klein werden. Das bedeutet zum einen, daß c möglichst klein sein muß, aber auch, daß auch die Steigungen von F und f in Richtung der Achsen des Koordinatensystems sowie der Funktionswert von f betraglich klein sein müssen. In allzu steilen Bereichen von F ist eine genaue Gradientenberechnung somit nicht möglich. Ansonsten muß h genügend groß gewählt werden, so daß der Trend der Fehlerfunktion möglichst gut wiedergegeben wird (vergleiche Abbildung 3.13(c)). Die Verfahren werden allerdings nur an einen Parametervektor gelangen, welcher in der Nähe des Minimums liegt, was vollkommen genügen kann, da die physikalischen Eigenschaften nur bis auf ihre statistischen Unsicherheiten genau bestimmbar sind.



(a) Lage zweier benachbarter Fehlerbalken im Falle eines idealen h für $F(x) > F(x + he_k)$.



(b) Lage zweier benachbarter Fehlerbalken im Falle eines idealen h für $F(x) < F(x + he_k)$.

Abbildung 3.14: Die Fehlerbalken an den Stellen x und $x + he_k$, $k = 1, ..., N$, dürfen nicht überlappen. Ansonsten ist es möglich, daß der approximierte negative Gradient den abfallenden Trend der Fehlerfunktion nicht korrekt wiedergibt und somit keine Abstiegsrichtung ist. Es bezeichnen $W_{\max}$ und $W_{\min}$ das obere beziehungsweise untere Ende des jeweiligen Fehlerbalkens. Das untere Ende des höher gelegenen Fehlerbalkens muß stets echt größer sein als das obere Ende des tiefer gelegenen, was durch geeignete Wahl von h erreicht werden kann. Dies ist hier für zwei Fälle dargestellt, und zwar für $F(x) > F(x + he_k)$ (a) und für $F(x) < F(x + he_k)$ (b).

Um auch theoretisch zu belegen, daß die Wahl eines größeren h sinnvoll ist, wird zunächst der in Abbildung 3.14 dargestellte Sachverhalt diskutiert: Es sei e_k der k-te Einheitsvektor. Die Gesamtfehlerfunktion $F(x) = \sum_{i=1}^{n} F^i(x)$ wird nun entlang e_k betrachtet. Der Diskretisierungsparameter h muß so gewählt werden, daß der Steigungstrend korrekt reproduziert wird, das heißt, daß $\langle -\nabla F(x), e_k \rangle < 0$ beziehungsweise daß $-\nabla F(x)$ eine Abstiegsrichtung bleibt. Dies kann dadurch erzielt werden, daß die Sekante durch die Punkte $(x, \tilde{F}(x))$ und $(x + he_k, \tilde{F}(x + he_k))$, welche durch die Fehlerbalken bei x und $x + he_k$ verläuft, eine negative Steigung aufweist. Das bedeutet wiederum, daß sich die beiden Fehlerbalken nicht überlappen dürfen: Im Falle von $F(x) > F(x + he_k)$ (Abbildung 3.14(a)) muß das obere Ende des Fehlerbalkens bei $x + he_k$ echt kleiner sein als das untere Ende des Fehlerbalkens bei x. Im Falle von $F(x) < F(x + he_k)$ (Abbildung 3.14(b)) muß dies genau umgekehrt sein. Es sei allgemein $W_{\min}(y)$ als das untere und $W_{\max}(y)$ als das obere Ende des Fehlerbalkens an der Stelle y definiert, also nach Abschätzung (3.76)

$$W_{\min}(y) := F(y) - 2 \sum_{i=1}^{n} \left[C^i(F^i(y) + 1) - \frac{1}{2} \bar{C}^i(y) \right] \tag{3.84}$$

$$W_{\max}(y) := F(y) + 2 \sum_{i=1}^{n} \left[C^i(F^i(y) + 1) + \frac{1}{2} \bar{C}^i(y) \right]. \tag{3.85}$$

Weiterhin seien

$$W^k_{\min} := \max(W_{\min}(x), W_{\min}(x + he_k)) \tag{3.86}$$

$$W^k_{\max} := \min(W_{\max}(x), W_{\max}(x + he_k)). \tag{3.87}$$

Dann ist

$$\forall_{k=1,...,N}\ W^k_{\min} > W^k_{\max} \tag{3.88}$$

eine Voraussetzung dafür, daß ein gradientenbasiertes Verfahren zum nächsten Iterationsschritt führt. Der Einfachheit halber wird die folgende Notation verwendet:

$$C := \sum_{i=1}^{n} C^i,\ \bar{C}(x) := \sum_{i=1}^{n} \bar{C}^i(x). \tag{3.89}$$

Wegen $CF(x) \geq \sum_{i=1}^{n} C^i F^i(x)$ bleibt Abschätzung (3.76) erhalten, wenn $CF(x)$ anstelle von $\sum_{i=1}^{n} C^i F^i(x)$ verwendet wird.
Zum Erhalt der Bedingung (3.88) muß somit gemäß den Gleichungen (3.84), (3.85), (3.86) und (3.87) für alle $k = 1, ..., N$ gelten:

$$\begin{aligned}
& \max(F(x) - 2CF(x) - 2C - \bar{C}(x), F(x + he_k) - 2CF(x + he_k) - 2C - \bar{C}(x + he_k)) \\
> \ & \min(F(x) + 2CF(x) + 2C + \bar{C}(x), F(x + he_k) + 2CF(x + he_k) + 2C + \bar{C}(x + he_k)) \\
\Leftrightarrow \ & \max(F(x), F(x + he_k)) - 2C \min(F(x), F(x + he_k)) - 2C - \min(\bar{C}(x), \bar{C}(x + he_k)) \\
> \ & \min(F(x), F(x + he_k)) + 2C \min(F(x), F(x + he_k)) + 2C + \min(\bar{C}(x), \bar{C}(x + he_k)) \\
\Leftrightarrow \ & \max(F(x), F(x + he_k)) > (1 + 4C) \min(F(x), F(x + he_k)) + 4C \\
& + 2 \min(\bar{C}(x), \bar{C}(x + he_k)).
\end{aligned} \tag{3.90}$$

Also muß h so gewählt werden, daß die Bedingung (3.90) erfüllt ist, falls dies aufgrund des Ausmaßes an statistischem Rauschen überhaupt möglich ist. Falls kein Rauschen vorhanden ist, so ist $C = 0$ und somit auch $\bar{C} = 0$. Dann wird Bedingung (3.90) zu:

$$\max(F(x), F(x + he_k)) > \min(F(x), F(x + he_k)). \tag{3.91}$$

Dies ist für $h \neq 0$ stets erfüllt. Es ist klar, daß h auch nicht zu groß gewählt werden darf, da dies auf Kosten der numerischen Genauigkeit gehen würde. Letztere kann im Falle von Rauschen allerdings nicht als so hoch erwartet werden wie bei der Minimierung glatter Funktionen.
Es stellt sich jedoch die Frage, ob ein $h > 0$ überhaupt existiert, welches Bedingung (3.90) erfüllt. Da bei Molekularen Simulationen keine analytische Form für die Fehlerfunktion F bekannt ist, kann dies jedoch nur auf theoretischer Ebene beantwortet werden. Der folgende Satz gibt Aufschluß über die Existenz von h und stellt deutlich dar, daß h nicht zu klein gewählt werden darf:

Satz 3.5.3 (Existenz von h). *Falls für ein Variablenpaar (H, h) mit $0 < H < h < 1$ die Bedingung*

$$\forall_{k=1,...,N}\ \left| \frac{F(x + he_k) - F(x)}{h} \right| \geq \frac{1}{H} \left(4CF(x) + 4C + 2\bar{C}(x) \right) \tag{3.92}$$

erfüllt ist, so ist auch die Fehlerbalkenbedingung (3.88) *erfüllt.*

Beweis: Angenommen, es gilt

$$\begin{aligned}
\forall_{0<H<h<1}\ \exists_{k_0 \in \{1,...,N\}} \quad & \max(F(x), F(x + he_{k_0})) \leq (1 + 4C) \min(F(x), F(x + he_{k_0})) + 4C \\
& + 2 \min(\bar{C}(x), \bar{C}(x + he_{k_0})).
\end{aligned}$$

Dann folgt einerseits:

$$\begin{aligned} F(x+he_{k_0}) &\leq F(x)+4CF(x)+4C+2\bar{C}(x) \\ \Rightarrow \frac{F(x+he_{k_0})-F(x)}{h} &\leq \frac{4CF(x)+4C+2\bar{C}(x)}{h} \\ &\overset{h>H}{<} \frac{1}{H}\left(4CF(x)+4C+2\bar{C}(x)\right). \end{aligned}$$

Andererseits gilt analog:

$$\frac{F(x)-F(x+he_{k_0})}{h} < \frac{1}{H}\left(4CF(x)+4C+2\bar{C}(x)\right).$$

Insgesamt folgt:

$$\left|\frac{F(x+he_{k_0})-F(x)}{h}\right| < \frac{1}{H}\left(4CF(x)+4C+2\bar{C}(x)\right). \qquad \lightning$$

Dies ist ein Widerspruch zu Ungleichung (3.92), und somit folgt die Behauptung. □

Eine Betrachtung für $h \to 0$ ist also im Falle von Rauschen nicht möglich. Es kann somit von den gradientenbasierten Verfahren keine Konvergenz in dem Sinne erwartet werden, daß das Minimum bis auf einen Term der Größenordnung $\mathcal{O}(h)$ oder $\mathcal{O}(h^2)$ genau bestimmt wird. Somit wird ein Verfahren hier als *konvergent* betrachtet, falls das Abbruchkriterium (3.67) erfüllt ist, wobei $\tau > 0$ so klein ist, daß die physikalischen Eigenschaften bis auf ihre statistischen Unsicherheiten genau bestimmt sind.
Wie bereits geometrisch motiviert, zeigt auch Satz 3.5.3, daß die Optimierungsverfahren nur dann zum Erfolg führen können, wenn die Steigung der Funktion in jeder Koordinatenrichtung im Vergleich zum Ausmaß des Rauschens ausreichend groß ist.

Die Schrittweite h ließe sich folgendermaßen iterativ bestimmen: Es werde zum Beispiel eine Iterationsfolge $h_{ij} := j \cdot 10^{-3+i},\ i = 0,1,2,\ j = 1,...,9$ gewählt, und zwar solange bis die Bedingung (3.90) für ein $h_{i_0 j_0}$ erfüllt ist. Ist die Bedingung für kein Zahlenpaar (i,j) erfüllt, so ist die Steigung im Gegensatz zum Rauschen nicht genügend groß, so daß kein h gefunden werden kann. In diesem Fall sollte das Äquilibrierungskriterium (3.70) strenger gewählt und/oder die Simulationsdauer erhöht werden, so daß das Rauschen reduziert wird.
Ein Algorithmus zur Bestimmung von h bedeutet jedoch auch erheblich mehr Rechenaufwand: Für die vorgeschlagene iterative Bestimmung von h würde ein im Gegensatz zur Gradientenbestimmung mit vorbestimmtem h zusätzlicher möglicher Rechenaufwand von $27N - N = 26N$ Iterationen entstehen. Im Falle von Molekularen Simulationen ist dies aufgrund der hohen Rechenzeiten in der Praxis nicht durchführbar, allerdings besteht die Möglichkeit zur Parallelisierung für die verschiedenen h_{ij}. Dann lohnt es sich jedoch eher, das Optimierungsverfahren abzubrechen, falls nach einer gewissen Anzahl an Armijo-Schritten (zum Beispiel zehn) kein verbesserter Fehlerfunktionswert gefunden ist, und die Simulation mit einem strikteren Äquilibrierungskriterium, einer längeren Simulationsdauer und/oder der Wahl eines größeren h zu wiederholen.
Ein weiteres Problem liegt in der Tatsache, daß $F(x)$ und $F(x+he_k)$ in Bedingung (3.90)

nicht bekannt sind, sondern nur $\tilde{F}(x)$ und $\tilde{F}(x + he_k)$. Im schlimmsten Fall kann jedoch nach Abschätzung (3.76)

$$\tilde{F}(x) = F(x) \pm 2CF(x) + D(x)$$

angenommen werden, wobei $D(x) := 2C + \bar{C}(x)$. Daher gilt stets

$$F(x) \in \left[\frac{\tilde{F}(x) - D(x)}{1 + 2C}, \frac{\tilde{F}(x) + D(x)}{1 - 2C} \right], \tag{3.93}$$

und man erhält eine Fehlerschranke um den simulierten, mit Rauschen behafteten Funktionswert $\tilde{F}(x)$, in der sich der tatsächliche Funktionswert $F(x)$ garantiert befindet. Diese Fehlerschranke kann in Bedingung (3.90) verwendet werden, indem stets die Extremfälle für $F(x)$ beziehungsweise $F(x + he_k)$ angenommen werden. Dies kann allerdings dazu führen, daß kein geeignetes h gefunden wird, obwohl eins existiert, da man so die statistischen Fehler überschätzt. Allerdings wird Abschätzung (3.93) später für eine lokale Approximation von F mit radialen Basisfunktionen (siehe Abschnitt 6.2.2) verwendet, wobei die statistischen Unsicherheiten von F in die Gewichtung der im Algorithmus enthaltenen Regression miteingehen können.

Bemerkung 3.5.4. *Die in Abschnitt 4.2 dargestellte Verfahrensbewertung hat ergeben, daß bei der vorliegenden Problemstellung h =0.01 eine geeignete Wahl ist. Weitere praktische Analysen haben ergeben, daß h zur Beschleunigung in den ersten Verfahrensschritten gegebenenfalls vergrößert werden (zum Beispiel auf h =0.02, siehe Simulations- und Optimierungsergebnisse aus Kapitel 5) und in der Nähe des Minimums in Abhängigkeit vom Ausmaß an statistischem Rauschen um eine Größenordnung verkleinert werden sollte (h =0.001).*

3.5.6 Wahl der Diskretisierung für die Hesse-Matrix

Im Falle der zweiten partiellen Ableitungen taucht beim statistischen Fehler der Term $\frac{c^2}{h^2}$ auf. Um eine akkurate (kleinere) Krümmung zu berechnen, müßte h demnach noch mehr vergrößert werden. Allerdings kann dies die Quadrierung von c in diesem Falle wieder ausgleichen. Da bei den meisten Optimierungsalgorithmen die Hesse-Matrix spd sein muß, was bei der vorliegenden Problemstellung nicht vorausgesetzt werden kann, und aus Komplexitätsgründen sind die meisten Hesse-Matrix-basierten Verfahren hier nicht empfehlenswert. Lediglich ganz in der Nähe des Minimums können sich derartige Verfahren eignen, falls dort die positive Definitheit erfüllt ist. Allerdings können Hesse-Matrizen aufgrund des Rauschens aus demselben Grund wie Gradienten dort nicht mehr genau bestimmt werden (vergleiche Abschnitt 3.5.5). Lediglich das exakte Trust-Region-Verfahren könnte dann angewandt werden, wenn andere Verfahren keine signifikant kleineren Fehlerfunktionswerte mehr liefern. Die Verfahrensbewertung, welche in Abschnitt 4.2 beschrieben wird, hat ergeben, daß das exakte Trust-Region-Verfahren für $h = 0.01$ oftmals näher an das Minimum gelangt als andere. Daher werden in diesem Abschnitt verschiedene Hesse-Matrizen von glatten und verrauschten Korrelationsfunktionen aus Abschnitt 3.5.2 in der Nähe des Minimums bezüglich ihrer Frobenius-Norm miteinander verglichen. Im Falle der verrauschten Funktionen wird h variiert. Anschließend wird in Abhängigkeit diskutiert, wie h im Falle des Trust-Region-Verfahrens, in Abhängigkeit von dessen Konvergenzverhalten, geeignet zu wählen ist.

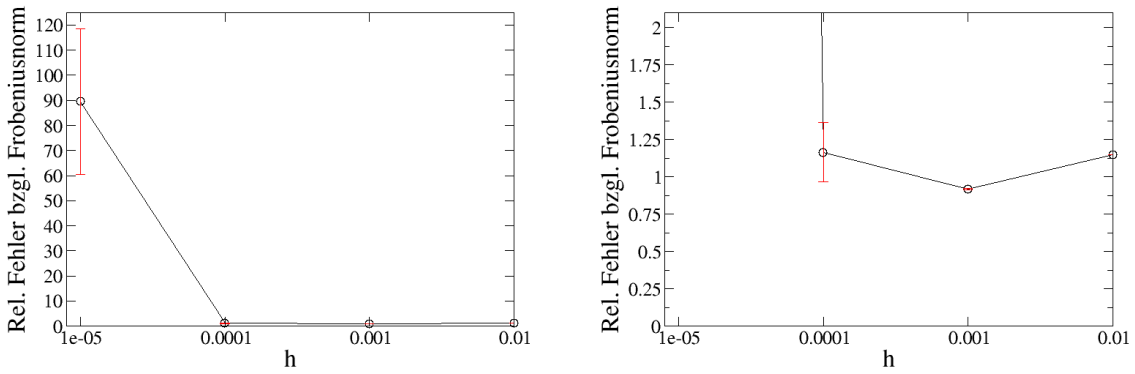


Abbildung 3.15: Relative durch statistisches Rauschen verursachte Fehler in der Hesse-Matrix bezüglich der Frobenius-Norm in Abhängigkeit von h. Das Beispiel bezieht sich auf Kraftfeldparameter in der Nähe des Minimums im Falle von glatten und verrauschten Korrelationsfunktionen aus Abschnitt 3.5.2. Die zu optimierenden Größen sind Dampfdruck und Siededichte von Stickstoff zu sechs verschiedenen Temperaturen. Wird h zu klein, so steigt der Fehler in Bezug auf die tatsächliche Hesse-Matrix drastisch an oder aber das Verfahren wird weniger robust. Letzteres ist an der Größe der Fehlerbalken zu erkennen, welche sich auf zehn unabhängige Zufallsreplikate beziehen. Bei $h = 0.001$ ist der Fehler am kleinsten, und $h = 0.01$ ist ebenfalls vor allem in den ersten Verfahrensschritten eine gute Wahl. Die rechte Abbildung ist analog zur linken, lediglich die Auflösung der Ordinate ist detaillierter.

Es werden Analysen für $h \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ betrachtet: Abbildung 3.15 zeigt den Vergleich verschiedener Hesse-Matrizen bezüglich ihrer Frobenius-Norm in Abhängigkeit von h. Angegeben sind die relativen Unterschiede in der Frobenius-Norm, also

$$\delta_h(H^h) := \frac{\|H^h(x) - D^2F(x)\|_F}{\|D^2F(x)\|_F},$$

wobei x der betrachtete Parametervektor in der Nähe des Minimums ist, $D^2F(x)$ die durch finite Differenzen approximierte Hesse-Matrix und $H^h(x)$ die verrauschte Hesse-Matrix in Abhängigkeit von h. Die zu optimierenden Zielgrößen sind Dampfdruck und Siededichte von Stickstoff zu sechs verschiedenen Temperaturen, welche mithilfe der Korrelationsfunktionen aus Abschnitt 3.5.2 berechnet wurden, also mittels der Gleichungen (3.62) und (3.64). Der betrachtete Temperaturbereich lautet $\mathcal{T}/K := \{65, 75, 85, 95, 105, 115\}$. Näheres zum Stickstoffmodell, basierend auf Korrelationsfunktionen, befindet sich in Abschnitt 4.1.
Der Parametervektor in der Nähe des Minimums wurde folgendermaßen festgelegt: Für $h = 0.01$ wurde das Trust-Region-Verfahren solange angewandt, bis $\Delta < \Delta^{\min}$. Auf die zu optimierenden Größen wurde gemäß Abschnitt 3.5.2 künstliches Rauschen addiert. Das Zufallsexperiment wurde zehnmal wiederholt, so daß zehn verschiedene Parametervektoren resultierten. Diese Vektoren wurden gemittelt, was zu dem hier betrachteten Parametervektor führte, welcher das für diese Optimierungsaufgabe festgelegte Abbruchkriterium $F(x) \leq 0.015$ (siehe Abschnitt 4.1) erfüllte.

Abbildung 3.15 zeigt, daß für $h = 10^{-5}$ die verrauschte Hesse-Matrix stark von der Hesse-Matrix im glatten Fall abweicht. Dies entspricht den in Abschnitt 3.5.5 getroffenen Überlegungen, welche besagen, daß bei zu kleinem h die durch Rauschen verursachten Oszillationen zu einem falschen Gradienten und zu einer falschen Hesse-Matrix führen können. Die Fehlerbalken in

h	$\mu(\delta_h)$	$\sigma(\delta_h)$
10^{-2}	1.146	$1.464 \cdot 10^{-4}$
10^{-3}	0.919	$3.856 \cdot 10^{-3}$
10^{-4}	1.164	0.198
10^{-5}	89.488	28.957

Tabelle 3.1: Relative durch statistisches Rauschen verursachte Fehler in der Hesse-Matrix bezüglich der Frobenius-Norm in Abhängigkeit von h, entsprechend Abbildung 3.15. Angegeben sind die Ergebnisse von zehn Zufallsreplikaten $\delta_h := \left(\delta_h(H_i^h)\right)_{i=1,\ldots,10}$. Dabei sind $\mu(\delta_h)$ der Mittelwert und $\sigma(\delta_h)$ die zugehörige Standardabweichung über die 10 Replikate.

Abbildung 3.15 beziehen sich auf zehn unabhängige Zufallsreplikate. Im Falle von $h = 10^{-4}$ ist die Abweichung bezüglich der Frobenius-Norm zwar nicht mehr so groß, allerdings obliegen die Hesse-Matrizen hohen Schwankungen, was an der Größe des Fehlerbalkens zu erkennen ist. Dies wiederum hat Auswirkung auf die Robustheit des Trust-Region-Verfahrens. Daher ist auch $h = 10^{-4}$ nicht empfehlenswert. Lediglich $h = 10^{-3}$ und $h = 10^{-2}$ liefern relativ geringe Abweichungen und sehr kleine Schwankungen. Tabelle 3.1 gibt die in Abbildung 3.15 dargestellten Werte für die relativen Abweichungen bezüglich der Frobenius-Norm an. Die Wahl $h = 10^{-3}$ lieferte die besten Ergebnisse für die Hesse-Matrix mit nur sehr geringen Schwankungen. Am robustesten ist jedoch die Wahl $h = 10^{-2}$, allerdings ist dies aus genannten Gründen nur für die ersten Verfahrensschritte sinnvoll, um die Konvergenz des Trust-Region-Verfahrens zu beschleunigen.

Die Wahl $h = 10^{-2}$ sowohl für den Gradienten als auch für die Hesse-Matrix hat sich für das Trust-Region-Verfahren als äußerst geeignet erwiesen, um in die Nähe eines lokalen Minimums zu gelangen, sowohl im Hinblick auf die Konvergenzgeschwindigkeit als auch auf die Robustheit des Verfahrens. Es ist nicht sinnvoll, ein h zu wählen, welches noch eine Größenordnung höher ist, da ansonsten der Approximationsfehler zu groß ist. Die Wahl $h = 0.02$ für die Hesse-Matrix und $h = 0.01$ für den Gradienten bleibt allerdings hier nicht ausgeschlossen und ist in Einzelfällen zu testen.
Es seien im folgenden mit h_G das h für den Gradienten und mit h_H das h für die Hesse-Matrix bezeichnet. Es sind einige Untersuchungen durchgeführt worden, wie sich das Trust-Region-Verfahren in der Nähe des Minimums verhält, das heißt, inwieweit es den oben erwähnten Parametervektor noch verbessern kann. Dabei sind folgende Einstellungen gegenübergestellt worden:

- $h_G = 10^{-2},\ h_H = 10^{-2}$: In diesem Fall sind keine Verbesserungen erzielt worden, da der Parametervektor zu dem Zeitpunkt festgesetzt wurde, als das Trust-Region-Verfahren aufgrund einer zu kleinen Schrittweite keine kleineren Fehlerfunktionswerte mehr erreichen konnte.
- $h_G = 10^{-3},\ h_H = 10^{-2}$
- $h_G = 10^{-4},\ h_H = 10^{-3}$
- $h_G = 10^{-3},\ h_H = 10^{-3}$
- $h_G = 10^{-4},\ h_H = 10^{-4}$

Nur für $h_G = 10^{-3}$ und $h_H = 10^{-3}$ konnte die Fehlerfunktion innerhalb von wenigen Iterationen um eine Größenordnung verkleinert werden, ansonsten waren nur geringfügige Verbesserungen

zu verzeichnen. Dies entspricht genau den Ergebnissen aus Abbildung 3.15 beziehungsweise Tabelle 3.1 sowie aus Bemerkung 3.5.4: In der Nähe des Minimums sind $h_G = 10^{-3}$ und $h_H = 10^{-3}$ die beste Wahl.

Bemerkung 3.5.5. *Für alle h_H liegen gemäß Tabelle 3.1 die Abweichungen bezüglich der Frobenius-Norm zumeist nahe bei 1, was einer Abweichung von der Hesse-Matrix im glatten Fall von etwa 100% entspricht. Es ist daher durchaus überraschend, daß das Trust-Region-Verfahren mit derart inkorrekten Hesse-Matrizen umgehen kann. Allerdings ist zu sagen, daß die Idee dieses Verfahrens darin besteht, die Hesse-Matrix so zu verändern, daß sie spd wird und mittels Lagrange-Theorie ein höherdimensionales Approximationsmodell entsteht, welches nur einen kleineren Fehlerfunktionswert finden muß, was in steilen Bereichen der Fehlerfunktion aufgrund der positiven Definitheit der modifizierten Hesse-Matrix äußerst wahrscheinlich ist.*

3.6 Erhöhung der Effizienz des Optimierungsablaufs

Die in Abschnitt 3.4 vorgestellten gradientenbasierten Verfahren haben sehr gute Konvergenzeigenschaften. Sie sind also in Bezug auf die Anzahl an Iterationen äußerst effizient, vor allem im Vergleich zum Simplex-Algorithmus aus Abschnitt 3.2.1. Allerdings ist für den Erhalt einer schnellen Konvergenz auch die Wahl der Startparameter ausschlaggebend. Es seien als Beispiel die Startparameter so gewählt, daß die gradientenbasierten Verfahren durchschnittlich in $P = 10$ Iterationen zum Minimum führen. Jede Iteration benötigt jedoch eine gewisse, meist hohe Anzahl an Molekularen Simulationen, was den Rechenaufwand erheblich erhöht: Verfahren, die lediglich den Gradienten verwenden, also die Methode des steilsten Abstiegs und die CG-Verfahren, benötigen in jedem Schritt N Simulationen zu dessen Berechnung, wobei N die Dimension des Optimierungsproblems ist. Wird wie im Falle von Newton-Raphson und Trust-Region in jedem Schritt eine Hesse-Matrix ermittelt, so kommen $\binom{N}{2} + N = \frac{N(N+1)}{2}$ Simulationen hinzu. Durch die Symmetrie der Hesse-Matrix kann das obere Dreieck ohne die Diagonale eingespart werden, es handelt sich aber immer noch um $\mathcal{O}(N^2)$ Simulationen. Weiterhin kommen Simulationen bei der Schrittweitensteuerung hinzu. Die Armijo-Schrittweite für $\ell = 1$ würde an den Rand des zulässigen Gebiets führen. Daher kann die Schrittweitensteuerung bei $\ell = 2$ begonnen werden. Angenommen, es werden durchschnittlich $A = 3$ Armijo-Schritte benötigt. Dann braucht man im Falle von $N = 4$ in jeder Iteration

- $(1+N+(A-1))*P = (1+4+2)*10 = 70$ Simulationen bei Verfahren, die nur Gradienten verwenden, und
- $(1 + N + N(N+1)/2 + (A-1)) * P = (1 + 4 + 10 + 2) * 10 = 170$ Simulationen bei Verfahren, die zusätzlich Hesse-Matrizen verwenden.

Es ist zu beachten, daß der jeweils letzte Armijo-Schritt die neue Iteration festlegt. Daher ist stets eine Funktionsauswertung abzuziehen.
Auf einem Rechencluster mit 16 zur Verfügung stehenden Quadcore-Opteron-Knoten mit jeweils 16 GB Arbeitsspeicher, die über einen schnellen Infiniband-Netzwerk mit jeweils 16 Gb/s Double Date Rate (DDR) miteinander verbunden sind, braucht eine Simulation, parallelisiert auf vier Prozessoren, im Falle von Systemen von nur etwa 1000 kleinen Molekülen (siehe Abschnitt 5.1) etwa drei bis vier Stunden. 70 Simulationen würden daher auf diesem Rechencluster etwa acht bis zwölf Tage benötigen, für 170 Simulationen werden bereits etwa drei bis vier Wochen

gebraucht. Somit stellt sich die Notwendigkeit, die Rechenzeit soweit als möglich zu reduzieren. Dies sollte jedoch nicht dadurch realisiert werden, daß die Simulationen selbst verkürzt werden, da ansonsten die statistischen Unsicherheiten zu groß sein werden, so daß die gradientenbasierten Verfahren nicht mehr zum Ziel führen. Da die Reduktion der Unsicherheiten, wie in den Abschnitten 3.5.4 und 3.5.5 eingehend diskutiert, eine entscheidende Rolle spielt, sollte lediglich die Gesamtrechenzeit reduziert werden, jede einzelne Simulation jedoch so akkurat wie möglich durchgeführt werden. Eine kürzere Simulationsdauer ist nur zu Beginn des Optimierungsprozesses denkbar, also in steilen Bereichen der Fehlerfunktion, in der Nähe des Minimums allerdings nicht mehr. Dynamische Simulationszeiten werden in den Abschnitten 5.1.4 und 5.1.5 diskutiert.
Die Gesamtrechenzeit kann auf verschiedene Weise reduziert werden, zum einen durch Parallelisierung unabhängiger Rechnungen und zum anderen durch mathematische Manipulationen, welche Molekulare Simulationen einsparen oder gar ersetzen. Möglichkeiten zur Parallelisierung werden in Abschnitt 3.6.1 angesprochen. Anschließend wird in den Abschnitten 3.6.2, 3.6.3 und 3.6.4 diskutiert, inwieweit Simulationen zur Gradienten- und Hesse-Matrix-Berechnung komplett ersetzt werden können. Dies kann mittels bereits vorhandener Simulationsergebnisse oder über die Verwendung von reduzierten Parametern erfolgen. Der Gradient wird dann über ein LGS mithilfe von Richtungsableitungen berechnet, was auch für die Spaltenvektoren der Hesse-Matrix funktioniert. Da letzteres mit reduzierten Parametern in der Praxis nicht durchführbar ist, wird die Idee der Verwendung reduzierter Parameter lediglich in der Nähe des Minimums angewandt. Umgesetzt wird diese Idee mit der *Methode der reduzierten Einheiten*, welche ebenfalls in Abschnitt 3.6.4 beschrieben wird. Eine weitere Möglichkeit zur Effizienzerhöhung besteht in der Kombination verschiedener Verfahren und in der Wahl geeigneter Startparameter. Bezüglich Kombination wird in Abschnitt 3.6.5 eine Variante des in Abschnitt 3.2.3 eingeführten Verfahrens nach Stoll vorgestellt. Eine Kombination von GROW (siehe Anhang G.1) mit der Methode der reduzierten Einheiten und der Variante des Verfahrens nach Stoll sowie die geeignete Wahl von Startparametern werden in Abschnitt 3.6.6 näher erläutert. Sämtliche in diesem Abschnitt dargestellten Methoden und Ideen sind, falls nicht anders angegeben, im Rahmen dieser Dissertation entwickelt worden.

3.6.1 Parallelisierung

Parallelisierungsmöglichkeiten ergeben sich sowohl bei Molekularen Simulationen selbst als auch bei den gradientenbasierten Optimierungsverfahren. Im ersten Fall ist zwischen MD- und MC-Simulationen zu unterscheiden: Bei MD-Simulationen geschieht die Parallelisierung über die Betrachtung der in Anhang E.1 beschriebenen voneinander unabhängigen Nachbarschaftslisten. Für jedes Molekül werden nur seine nächsten Nachbarn in die Berechnung der Kräfte miteinbezogen. Sämtliche Kräfte zwischen zwei Teilchen können völlig unabhängig voneinader berechnet werden, siehe dazu zum Beispiel Hofmann (1992). Mithilfe der Nachbarschaftslisten kann die Parallelisierung noch effizienter erfolgen. Bei MC-Simulationen ist die Parallelisierung denkbar trivial: Es werden per Zufallsprinzip verschiedene Stichproben parallel aus dem Phasenraum entnommen und die entsprechenden Ergebnisse (zum Beispiel die potentielle Energie) anschließend gemittelt.
Durch Parallelisierung ist es also bei Vorhandensein entsprechender Hardware-Ressourcen möglich, daß $3 + N + N(N+1)/2$ Simulationen pro Iteration parallel ausgeführt werden. Somit

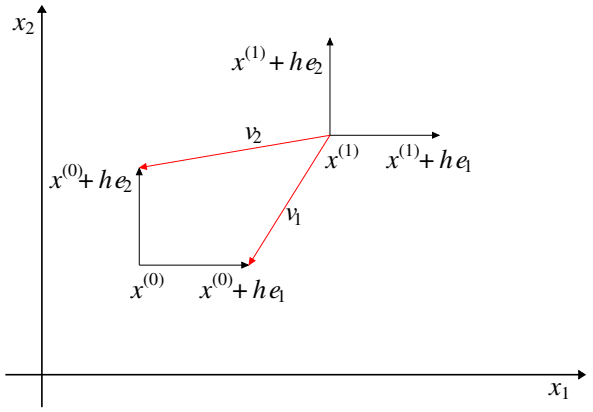


Abbildung 3.16: Effiziente Gradientenberechnung in 2D: Für die Iteration $x^{(1)}$ ist der Gradient effizient zu berechnen. Anstatt zusätzliche Simulationen für $x^{(1)}+he_i,\ i=1,2$, durchzuführen, kann auf die Funktionswerte von $x^{(0)}+he_i,\ i=1,2$, zurückgegriffen werden. Die Vektoren v_1 und v_2, die von $x^{(1)}$ nach $x^{(0)}+he_i,\ i=1,2$, sind mit roten Pfeilen gekennzeichnet und bilden eine neue Basis. Mithilfe einer Basistransformation kann der tatsächliche Gradient approximiert werden.

liegt die Gesamtkomplexität der Optimierung zumindest scheinbar bei $\mathcal{O}(P)$ Simulationen. Für CMA-ES (siehe Abschnitt 3.3.2) können die einzelnen Individuen innerhalb einer Population und im Falle von DesParO (siehe Abschnitt 3.3.3) sämtliche Zufallsmodelle parallelisiert werden. Notwendig für sämtliche Parallelisierungen ist selbstverständlich ein leistungsfähiger Rechencluster mit einer ausreichenden Anzahl an Knoten. Zusätzlich muß berücksichtigt werden, daß bei der Anpassung physikalischer Eigenschaften zu mehreren verschiedenen Temperaturen die entsprechenden Simulationen in jedem Fall parallel durchgeführt werden sollten. Da sich die Moleküle bei geringeren Temperaturen und hohen Drücken deutlich langsamer bewegen als bei höheren Temperaturen und niedrigen Drücken, dauert es im ersten Fall bis zur Äquilibrierung auch deutlich länger. Die langsamste Simulation ist diejenige, die die Gesamtrechenzeit innerhalb einer Iteration bestimmt.

3.6.2 Effiziente Gradientenberechnung mittels bereits durchgeführter Simulationen

Unter bestimmten Voraussetzungen ist es tatsächlich möglich, rechenaufwendige Simulationen durch mathematische Umrechnungen zu ersetzen. In diesem und im folgenden Abschnitt wird dargestellt, wie Gradient und Hesse-Matrix alternativ ziemlich genau berechnet werden können, wobei hier zunächst die Gradienten betrachtet werden. Die zugehörigen Algorithmen und Ideen sind im Rahmen dieser Dissertation entwickelt worden.
Die sogenannte *klassische* Gradientenberechnung geschieht über die partiellen Ableitungen, welche als die Richtungsvektoren in Richtung der Einheitsvektoren $e_j,\ j=1,...,N$, definiert sind. Der Gradient einer Iteration $x^{(k)},\ k\in\mathbb{N}$, des Optimierungsalgorithmus wird numerisch über die finiten Differenzen aus Gleichung (3.52) berechnet. In einem Optimierungsschritt wird implizit eine Basistransformation durchgeführt. Betrachtet wird zunächst die Basis $\{he_j\}_{j=1,...,N}$ mit $x^{(k)}$ als Ursprung. Die Division durch h bewirkt die Rücktransformation in kartesische Koordinaten. Der Ursprung ist jedoch immer noch $x^{(k)}$. Erst durch die Iterationsvorschrift

$x^{(k+1)} = x^{(k)} + t_k d^k \left(\nabla F(x^{(k)})\right)$ wird eine Rücktransformation in kartesische Koordinaten mit dem Nullpunkt als Ursprung erzielt.
Die Motivation besteht darin, im nächsten Optimierungsschritt Richtungsableitungen in Richtung von Vektoren $v_j^{(k+1)},\ j = 1, ..., N$, zu bestimmen und den Gradienten zunächst bezüglich derjenigen Basis zu berechnen, die von diesen Vektoren aufgespannt wird, mit $x^{(k+1)}$ als Ursprung. Die Rücktransformation erfolgt dann über ein LGS. Die Idee besteht nun darin, die Vektoren $v_j^{(k+1)}$ so zu wählen, daß die Linearkombination $x^{(k+1)} + v_j^{(k+1)}$ zu Vektoren führt, für die bereits Simulationen durchgeführt worden sind. Dies können beispielsweise die Vektoren $x^{(k)} + he_j$ aus dem vorherigen Optimierungsschritt sein, wie in Abbildung 3.16 für den zweidimensionalen Fall veranschaulicht. Es gilt dann:

$$\forall_{j=1,...,N}\ v_j^{(k+1)} = x^{(k)} + he_j - x^{(k+1)}. \tag{3.94}$$

Die Richtungsableitungen in Richtung der Vektoren $v_j^{(k+1)}$ werden dann gemäß

$$\begin{aligned} r_j^{(k+1)} &= \left\langle \nabla F(x^{(k+1)}), \frac{v_j^{(k+1)}}{||v_j^{(k+1)}||} \right\rangle \\ &= \frac{F\left(x^{(k+1)} + ||v_j^{(k+1)}|| \frac{v_j^{(k+1)}}{||v_j^{(k+1)}||}\right) - F(x^{(k+1)})}{||v_j^{(k+1)}||} + \mathcal{O}(||v_j^{(k+1)}||), \\ & j = 1, ..., N, \end{aligned} \tag{3.95}$$

als finite Differenzen berechnet. Wegen Gleichung (3.94) sind sämtliche Funktionsauswertungen in Gleichung (3.95) bereits bekannt, so daß keine zusätzlichen Simulationen durchgeführt werden müssen. Auch die Berechnung der Norm kann umgangen werden. Es werden lediglich

$$r_j^{(k+1)} ||v_j^{(k+1)}|| = F\left(x^{(k+1)} + v_j^{(k+1)}\right) - F(x^{(k+1)}),\ j = 1, ..., N, \tag{3.96}$$

berechnet. Sind $v_{ji}^{(k+1)},\ i = 1, ..., N$, die Komponenten des Vektors $v_j^{(k+1)}$ und $x_i^{(k+1)},\ i = 1, ..., N$, die von $x^{(k+1)}$, so ist zum Erhalt des Gradienten das folgende LGS zu lösen:

$$\begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{NN} \end{pmatrix} \begin{pmatrix} \frac{\partial F}{\partial x_1^{(k+1)}}(x^{(k+1)}) \\ \frac{\partial F}{\partial x_2^{(k+1)}}(x^{(k+1)}) \\ \vdots \\ \frac{\partial F}{\partial x_N^{(k+1)}}(x^{(k+1)}) \end{pmatrix} = \begin{pmatrix} r_1^{(k+1)} ||v_1^{(k+1)}|| \\ r_2^{(k+1)} ||v_2^{(k+1)}|| \\ \vdots \\ r_N^{(k+1)} ||v_N^{(k+1)}|| \end{pmatrix}. \tag{3.97}$$

Dieser Sachverhalt sei anhand des folgenden Beispiels veranschaulicht:

Beispiel 3.6.1. *Es wird im folgenden die Funktion* $H(x_1, x_2) := x_1^2 + x_2^3$ *mit Gradient* $\nabla H(x_1, x_2) = (2x_1, 3x_2^2)^T$ *betrachtet. Der Startwert sei* $x^{(0)} := (-1, 1)^T$ *mit Gradient* $\nabla H(x^{(0)}) = (-2, 3)^T$. *Weiterhin seien* e_1 *und* e_2 *die Einheitsvektoren in* $\mathbb{R}^2$ *sowie* $h = 10^{-3}$. *Es soll der Gradient bei*

$$x^{(1)} = x^{(0)} + h \begin{pmatrix} \frac{1}{2} \\ \frac{2}{3} \end{pmatrix}$$

bestimmt werden.
Für die Basisvektoren gilt

$$\begin{aligned} v_1^{(1)} &= x^{(0)} + he_1 - x^{(1)} = \begin{pmatrix} -1+h \\ 1 \end{pmatrix} - \begin{pmatrix} -1+\frac{h}{2} \\ 1+\frac{2h}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{2}{3} \end{pmatrix} h \\ v_2^{(1)} &= x^{(0)} + he_2 - x^{(1)} = \begin{pmatrix} -1 \\ 1+h \end{pmatrix} - \begin{pmatrix} -1+\frac{h}{2} \\ 1+\frac{2h}{3} \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{3} \end{pmatrix} h. \end{aligned}$$

Für die Richtungsableitungen in Richtung v_1 und v_2 gilt dann:

$$\begin{aligned} r_1^{(1)}||v_1^{(1)}|| &\doteq H(x^{(0)} + he_1) - H(x^{(1)}) = 1.998009 - 2.0010015 = -0.0030006 \\ r_2^{(1)}||v_2^{(1)}|| &\doteq H(x^{(0)} + he_2) - H(x^{(1)}) = 2.003003 - 2.0010015 = 0.0020014. \end{aligned}$$

Somit ist das folgende LGS zu lösen:

$$\begin{pmatrix} \frac{h}{2} & -\frac{2h}{3} \\ -\frac{h}{2} & \frac{h}{3} \end{pmatrix} \begin{pmatrix} \frac{\partial H}{\partial x_1}(x^{(1)}) \\ \frac{\partial H}{\partial x_2}(x^{(1)}) \end{pmatrix} = \begin{pmatrix} -0.0030006 \\ 0.0020014 \end{pmatrix}.$$

Die Lösung dieses LGS ist $\nabla H(x^{(1)}) = (-2.0045, 2.9975)^T$. Der analytisch bestimmte Gradient lautet im Vergleich dazu $\nabla H(x^{(1)}) = (-1.999, 3.002)^T$ Die Differenz ist gemäß Gleichung (3.95) *wie zu erwarten in $\mathcal{O}(h)$, da $||v_1^{(1)}|| = \frac{5}{6}h$ und $||v_2^{(1)}|| = \frac{\sqrt{13}}{6}h$.*

Das in diesem Abschnitt vorgestellte Verfahren zur effizienten Gradientenberechnung ist allerdings nur unter den folgenden Voraussetzungen anwendbar:

- Die finiten Differenzen aus Gleichung (3.95) müssen eine ausreichend gute Näherung für die Richtungsableitungen sein, das heißt, es muß gelten:

  $$\forall_{j=1,\ldots,N}\ ||v_j^{(k+1)}|| \leq Ch \tag{3.98}$$

  für ein benutzerdefiniertes $C > 0$. Dann ist $\forall_{j=1,\ldots,N}\ ||v_j^{(k+1)}|| = \mathcal{O}(h)$ gewährleistet.

- Die Vektoren $v_j^{(k+1)},\ j = 1, ..., N$, müssen eine Basis bilden, das heißt, es muß gelten:

  $$\det \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{NN} \end{pmatrix} > \epsilon_{\det} \tag{3.99}$$

  für ein benutzerdefiniertes $\epsilon_{\det} > 0$.
  Ein besseres Maß für die lineare Unabhängigkeit liefert die Konditionszahl $\kappa > 0$. Gilt $\kappa \leq 1$, so gilt das LGS als gut konditioniert. Anstelle des Determinantenkriteriums (3.99) sollte also das Konditionszahlkriterium

  $$\kappa \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \cdots & v_{NN} \end{pmatrix} < \epsilon_\kappa \tag{3.100}$$

verwendet werden, wobei $\epsilon_{\text{det}} > 0$.

Ist eine der beiden Bedingungen (3.98) oder 3.100 nicht erfüllt, so ist der Gradient auf klassische Weise zu bestimmen. Ist das Verfahren bei mehreren aufeinanderfolgenden Iterationen $x^{(k)}, x^{(k+1)}, x^{(k+2)}, ...$ anwendbar, so sind die Ergebnisse der Simulationen an den Stellen $x^{(k)} + he_j,\ j = 1, ..., N$, mehrmals hintereinander zu verwenden. Zu nahe am Minimum wird das hier beschriebene Verfahren problematisch, da die zusätzlichen Ungenauigkeiten keine exakte Gradientenberechnung mehr zulassen, was jedoch gerade kurz vorm Minimum absolut erforderlich wäre. In der Nähe des Minimums ist das Normkriterium in jedem Fall strikter zu wählen als zu Beginn der Optimierung. Wie oft der Gradient tatsächlich effizient berechnet werden kann, hängt vom Optimierungsablauf ab und wird für bestimmte Beispiele im einzelnen zu studieren sein. In Abschnitt 4.3.1 wird eine detaillierte Evaluation der effizienten Gradientenberechnung durchgeführt.

Es ist nicht unbedingt erforderlich, auf die Funktionswerte von $x^{(k)} + he_j,\ j = 1, ..., N$, zurückzugreifen. Im Prinzip können beliebige Vektoren verwendet werden, für die bereits Simulationen durchgeführt wurden. Dabei sollten Vektoren gewählt werden, bei denen sowohl das Norm- als auch das Konditionszahlkriterium am günstigsten sind. Die Auswahl dieser Vektoren sollte jedoch ebenfalls effizient durchgeführt werden, das heißt, es sollte nicht jede beliebige Kombination überprüft werden. Es können beispielsweise diejenigen Vektoren von der Suche ausgeschlossen werden, die weit von der aktuellen Iteration entfernt liegen.

3.6.3 Effiziente Berechnung der Hesse-Matrix mittels bereits durchgeführter Simulationen

Auch die Hesse-Matrix kann genau wie der Gradient mithilfe von bereits durchgeführten Simulationen effizient bestimmt werden. Gemäß Gleichung (3.53) sind hierzu neben $\nabla F(x^{(k+1)})$ auch $\nabla F(x^{(k+1)} + he_i),\ i = 1, ..., N$, notwendig. Letztere sind wie $\nabla F(x^{(k+1)})$ gemäß Abschnitt 3.6.2 effizient zu bestimmen:
Wegen

$$\forall_{i=1,...,N}\ \forall_{j \geq i}\ F(x^{(k+1)} + he_i + he_j) = F(x^{(k+1)} + he_i + w_j^{(k+1)}),$$

mit

$$w_j^{(k+1)} := x^{(k)} + he_i + he_j - (x^{(k+1)} + he_j) = x^{(k)} + he_j - x^{(k+1)} = v_j^{(k+1)}$$

(vergleiche Abbildung 3.17 für den zweidimensionalen Fall) sind auch die zweiten Ableitungen $\frac{\partial^2 F}{\partial x_i \partial x_j}(x^{(k+1)}),\ i, j = 1, ..., N$, effizient bestimmbar. Allerdings sind die Funktionswerte $F(x^{(k+1)} + he_i),\ i = 1, ..., N$, nur über Simulationen zu ermitteln. Es wäre denkbar, eine Taylorentwicklung der Form

$$F(x^{(k+1)} + he_i) = F(x^{(k)}) + h\frac{\partial F}{\partial x_i}(x^{(k)}) + \mathcal{O}(h^2),\ j = 1, ..., N,$$

zu verwenden, allerdings können sich kleine Fehler erheblich auf die zweiten partiellen Ableitungen auswirken: Ist der Zähler aus Gleichung (3.53) durch numerische und/oder statistische

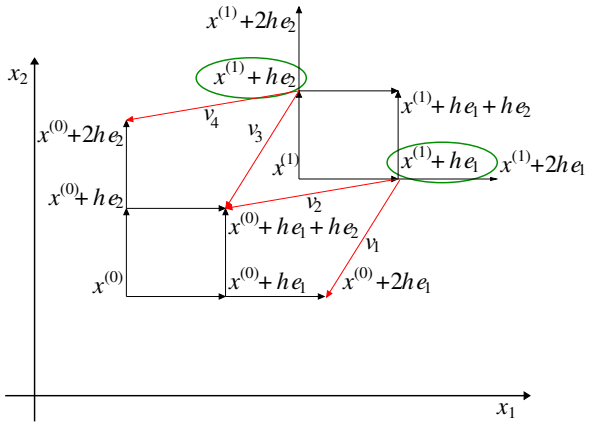


Abbildung 3.17: Effiziente Hesse-Berechnung in 2D: Für die Iteration $x^{(1)}$ ist die Hesse-Matrix effizient zu berechnen. Notwendig hierfür sind $\nabla F^(x^{(1)} + he_i),\ i = 1,2$. Diese Gradienten können mithilfe der Methode aus Abschnitt 3.6.2 effizient berechnet werden: Anstatt zusätzliche Simulationen für $x^{(1)}+he_i+he_j,\ i,j = 1,2$, durchzuführen, kann auf die Funktionswerte von $x^{(0)}+he_i+he_j,\ i,j = 1,2$, zurückgegriffen werden. Die Vektoren $v_1 = v_3$ und $v_2 = v_4$, die von $x^{(1)} + he_i,\ i = 1,2$, nach $x^{(0)} + he_i + he_j,\ j = 1,2$, sind mit roten Pfeilen gekennzeichnet. v_1 und v_2 bilden eine neue Basis. Mithilfe einer Basistransformation können die tatsächlichen Gradienten und damit die Hesse-Matrix approximiert werden. Die Funktionswerte für die beiden Vektoren $x^{(1)} + he_i,\ i = 1,2$ (grün umrahmt), müssen allerdings mithilfe von Simulationen ermittelt werden.

Fehler in $\mathcal{O}(h)$, so bewirkt die Division durch h^2, daß der Gesamtfehler in den partiellen Ableitungen in $\mathcal{O}(h^{-1})$ ist. In Abschnitt 3.5.6 wurde die Auswirkung von statistischem Rauschen auf die Frobenius-Norm der Hesse-Matrix dargestellt. Somit ist es zu empfehlen, die Funktionswerte $F(x^{(k+1)} + he_i),\ i = 1,...,N$, über Simulationen zu bestimmen.

Das Beispiel aus Abschnitt 3.6.2 wird nun auf die Hesse-Matrix erweitert:

Beispiel 3.6.2 (Fortsetzung von Beispiel 3.6.1)**.** *Für die Funktion* $H(x_1,x_2) = x_1^2 + x_2^3$ *gilt gemäß effizienter Gradientenberechnung* $\nabla H(x^{(1)}) = (-2.0045, 2.9975)^T$. *Die Funktionswerte* $H(x^{(1)} + he_1)$ *und* $H(x^{(1)} + he_2)$ *werden aus der Funktionsgleichung berechnet, also simuliert. Für die Richtungsableitungen in Richtung* $w_1^{(1)} = v_1^{(1)}$ *und* $w_2^{(1)} = v_2^{(1)}$ *gilt:*

$$\begin{aligned}
r_1^{(1)}||w_1^{(1)}|| &\doteq H(x^{(1)} + he_1 + w_1^{(1)}) - H(x^{(1)} + he_1) \\
&= 1.996004 - 1.9990035 = -0.0029996 \\
r_1^{(1)}||w_2^{(1)}|| &\doteq H(x^{(1)} + he_1 + w_2^{(1)}) - H(x^{(1)} + he_1) \\
&= 2.001004 - 1.9990035 = 0.0020004 \\
r_2^{(1)}||w_1^{(1)}|| &\doteq H(x^{(1)} + he_2 + w_1^{(1)}) - H(x^{(1)} + he_2) \\
&= 2.001004 - 2.0040085 = -0.0030046 \\
r_2^{(1)}||w_2^{(1)}|| &\doteq H(x^{(1)} + he_2 + w_2^{(1)}) - H(x^{(1)} + he_2) \\
&= 2.006012 - 2.0040085 = 0.0020034.
\end{aligned}$$

Die Richtungsableitungen $r_i^{(1)}||w_j^{(1)}|| = \left\langle \nabla H(x^{(1)} + he_i), w_j^{(1)} \right\rangle$ *bilden die rechte Seite von zwei Gleichungssystemen* $i = 1,2$. *Wegen* $\forall_{i=1,2}\ w_i^{(1)} = v_i^{(1)}$, *ist die Matrix dieselbe wie bei der*

effizienten Gradientenberechnung (siehe Beispiel 3.6.1). Die Lösungen der beiden Gleichungssysteme sind $\nabla H(x^{(1)} + he_1) = (-2.003, 2.997)^T$ *und* $\nabla H(x^{(1)} + he_2) = (-2.005, 3.004)^T$. *Die Komponenten der Hesse-Matrix können nun gemäß Gleichung* (3.53) *bestimmt werden. Es gilt:*

$$D^2H(x^{(1)}) = \begin{pmatrix} 2 & 0 \\ -0.004 & 6.002 \end{pmatrix}.$$

Im Vergleich zu der analytisch bestimmten Hesse-Matrix

$$D^2H(x^{(1)}) = \begin{pmatrix} 2 & 0 \\ 0 & 6.004 \end{pmatrix}$$

ist dies wie erwartet eine $\mathcal{O}(h)$*-Approximation.*

Da $\forall_{i=1,...,N}\ w_i^{(k+1)} = v_i^{(k+1)}$ gilt, ist das in diesem Abschnitt vorgestellte Verfahren zur effizienten Hesseberechnung unter denselben Voraussetzungen anwendbar wie das zur effizienten Gradientenberechnung. Es sind jedoch die folgenden Zusatzüberlegungen zu treffen:

- Die finiten Differenzen aus Gleichung (3.53) müssen eine $\mathcal{O}(h)$-Näherung für die zweiten partiellen Ableitungen sein. Kleine Störungen haben große Auswirkungen auf die Hesse-Matrix. Die Normbedingung für die $v_j^{(k+1)},\ j = 1, ..., N$, muß daher in jedem Fall lauten:

$$\forall_{j=1,...,N}\ ||v_j^{(k+1)}|| \leq Ch,\ 0 < C \leq 1. \tag{3.101}$$

  Dann ist $\forall_{j=1,...,N}\ ||v_j^{(k+1)}|| = \mathcal{O}(h)$ gewährleistet.
- Auch die Determinantenbedingung (3.99) beziehungsweise das Konditionszahlkriterium (3.100) müssen verschärft werden. Die Matrix der Basisvektoren $v_j^{(k+1)},\ j = 1, ..., N$, muß besser konditioniert sein als im Falle der effizienten Gradientenberechnung, denn kleine durch Rauschen verursachte Störungen auf der rechten Seite des LGS wirken sich sonst erheblich auf die Hesse-Matrix aus.

Ist eine effiziente Hesseberechnung möglich, so ist auch eine effiziente Gradientenberechnung möglich, so daß sämtliche für die Hesse-Matrix notwendigen Simulationen, mit Ausnahme der Funktionswerte $F(x^{(k+1)} + he_i),\ i = 1, ..., N$, eingespart werden können. Die Umkehrung gilt nicht, das heißt, es ist möglich, daß sämtliche Simulationen für die Hesse-Matrix durchgeführt werden müssen. Dann sind jedoch auch alle Gradientenkomponenten mittels Simulation zu bestimmen.

3.6.4 Methode der reduzierten Einheiten

Mithilfe von reduzierten, das heißt dimensionslosen Kraftfeldparametern, ist es ebenfalls möglich, unter bestimmten Voraussetzungen rechenaufwendige Simulationen zu ersetzen. Aufgrund des statistischen Rauschens sei allerdings von vornherein gesagt, daß auch dieser Abschnitt genau wie Abschnitt 3.5.5 zunächst nur theoretisch geprägt ist und der hier beschriebene Algorithmus in der Praxis nur unter ganz bestimmten Voraussetzungen anwendbar ist. Eine Effizienzerhöhung im Rahmen des hier beschriebenen Algorithmus ist unter den folgenden Voraussetzungen möglich:

Physikalische Größe	Reduktionsformel
Partialladung	$q^* = q/(\sqrt{4\pi\varepsilon_0}\varepsilon_R\sigma_R)$
Dipolmoment	$\mu^{*2} = \mu^2/(\sqrt{4\pi\varepsilon_0}\varepsilon_R\sigma_R^3)$
Quadrupolmoment	$Q^{*2} = Q^2/(\sqrt{4\pi\varepsilon_0}\varepsilon_R\sigma_R^5)$
Elongation	$L^* = L/\sigma_R$
Temperatur	$T^* = Tk_B/\varepsilon_R$
Druck	$P^* = P\sigma_R^3/\varepsilon_R$
Simulationszeitschritt	$\Delta t^* = \Delta t\sqrt{\varepsilon_R/m_R}/\sigma_R$
Dichte	$\rho^* = \rho\sigma_R^3$
Verdampfungsenthalpie	$\Delta_v H^* = \Delta_v H/\varepsilon_R$
Diffusionskoeffizient	$D^* = D/(\sigma_R\sqrt{m_R/\varepsilon_R})$

Tabelle 3.2: Reduktionsformeln verschiedener physikalischer Größen.

- Es muß eine Beziehung zwischen den LJ-Parametern σ und ε und den physikalischen Zielgrößen existieren. Mithilfe von Längen, Energien und Massen können die physikalischen Einheiten jeder Zielgröße herausgekürzt werden. Diese Voraussetzung ist also stets erfüllt.
- Die ersetzten Simulationen werden, wie im folgenden noch motiviert wird, zu etwas anderen Bedingungen durchgeführt, das heißt zu einer etwas anderen Temperatur und zu einem etwas anderen Druck. Auch das molekulare Modell wird verändert. Es muß daher gewährleistet sein, daß die Fehler, die sich aufgrunddessen für die Berechnung der physikalischen Eigenschaften ergeben, möglichst gering gehalten werden. Dies hängt selbsverständlich auch vom Ausmaß der statistischen Unsicherheiten ab.
- Es müssen entweder Möglichkeiten zur Korrektur der Fehler bestehen, die bezüglich Temperatur und Druck entstehen, oder die zu minimierende Fehlerfunktion muß verändert werden. Letzteres bedeutet, daß experimentelle Daten zu anderen Temperaturen und Drücken in Gleichung (3.2) eingesetzt werden müssen. Dies hat jedoch den Nachteil, daß diese für jede Temperatur und jeden Druck vorhanden sein müssen. Im Falle der Existenz einer Zustandsgleichung ist dies möglich, was allerdings nicht für jede Substanz vorausgesetzt werden kann.

Der im folgenden vorgestellte, in dieser Dissertation entwickelte Algorithmus basiert sowohl auf mathematischen Umrechnungen als auch auf physikalischen Gesetzmäßigkeiten. Anstelle des Gradienten werden analog zu Abschnitt 3.6.2 Richtungsableitungen berechnet, aus denen der Gradient über ein LGS bestimmt werden kann. Der Algorithmus basiert auf der Verwendung von *reduzierten Parametern*, also Parametern, deren physikalische Einheit mithilfe von σ, ε und m herausgekürzt wurde. Reduzierte Parameter werden durch einem Stern (*) gekennzeichnet. Die Reduktion von σ und ε selbst erfolgt durch die Division durch geeignete Referenzwerte σ_R, ε_R und m_R. Die Reduktion von bestimmten physikalischen Größen erfolgt dann mittels der in Tabelle 3.2 angegebenen Reduktionsformeln für Dipol- und Quadrupolmoment, Elongation, Temperatur, Druck, Simulationszeitschritt, Dichte, Verdampfungsenthalpie und Diffusionskoeffizient. Es ist jedoch zu beachten, daß je nach physikalischer Einheit ein bestimmter multiplikativer Faktor zu berücksichtigen ist.
Werden nun reduzierte Werte für σ und ε als Eingabewerte für eine Simulation betrachtet, so

erhält man als Ausgabe auch reduzierte Werte für die berechneten physikalischen Größen:

$$\sigma, \varepsilon \rightarrow \left\{ \begin{array}{rcl} \frac{\sigma}{\sigma_R} & = & \sigma^* \\ \frac{\varepsilon}{\varepsilon_R} & = & \varepsilon^* \end{array} \right\} \overset{\text{Sim.},T,P,q}{\rightarrow} \left\{ \begin{array}{rcccl} \Delta_v H^* & = & \frac{\Delta_v H}{\varepsilon_R} & \rightarrow & \Delta_v H \\ \rho^* & = & \rho\sigma_R^3 & \rightarrow & \rho \\ D^* & = & \frac{D}{\sigma_R\sqrt{m_R/\varepsilon_R}} & \rightarrow & D \end{array} \right\}.$$

Bei der Berechnung des Gradienten werden σ und ε nacheinander um ein $h > 0$ verändert. Im folgenden werden somit veränderte Kraftfeldparameter $\tilde{\sigma} := \sigma + h$ beziehungsweise $\tilde{\varepsilon} := \varepsilon + h$ betrachtet. Die Idee besteht nun darin, auch die Referenzwerte σ_R und ε_R so zu verändern, daß dieselben reduzierten Kraftfeldparameter wie oben verwendet werden:

$$\tilde{\sigma}, \tilde{\varepsilon} \rightarrow \left\{ \begin{array}{rcl} \frac{\tilde{\sigma}}{\tilde{\sigma}_R} & = & \sigma^* \\ \frac{\tilde{\varepsilon}}{\tilde{\varepsilon}_R} & = & \varepsilon^* \end{array} \right\} \overset{\not{\text{Sim.}},\tilde{T},\tilde{P},\tilde{q}}{\rightarrow} \left\{ \begin{array}{rcccl} \Delta_v H^* & = & \frac{\tilde{\Delta_v H}}{\tilde{\varepsilon}_R} & \rightarrow & \tilde{\Delta_v H} \\ \rho^* & = & \tilde{\rho}\tilde{\sigma}_R^3 & \rightarrow & \tilde{\rho} \\ D^* & = & \frac{\tilde{D}}{\tilde{\sigma}_R\sqrt{m_R/\tilde{\varepsilon}_R}} & \rightarrow & \tilde{D} \end{array} \right\}.$$

Das bedeutet, es kann eine komplette Simulation eingespart werden, und dennoch können die neuen physikalischen Größen $\tilde{\Delta_v H}$, $\tilde{\rho}$ und $\tilde{D}$ berechnet werden. Allerdings ändern sich gemäß Tabelle 3.2 auch andere Eingabegrößen. Die problematischen Eingabeparameter sind vor allem die Temperatur T und der Druck P. Die hier ausgelassene Simulation ist in Wahrheit zu einer anderen Temperatur $\tilde{T}$ und zu einem anderen Druck $\tilde{P}$ durchgeführt worden, welche sich gemäß Tabelle 3.2 aus den geänderten Referenzwerten σ_R und ε_R ergeben. Das gleiche gilt auch für Ladungen, wobei die Reduktionsformeln anders aussehen, je nachdem, ob es sich um einen Monopol, Dipol oder Quadrupol handelt (siehe Abschnitt 2.2.3). Es stehe im folgenden q für eine dieser drei Ladungsarten. Dann kann bei Änderung von σ und ε eine Simulation folgendermaßen umgangen werden:

$$\tilde{\sigma}, \tilde{\varepsilon}, \tilde{T}, \tilde{P}, \tilde{q} \rightarrow \left\{ \begin{array}{rcl} \frac{\tilde{\sigma}}{\tilde{\sigma}_R} & = & \sigma^* \\ \frac{\tilde{\varepsilon}}{\tilde{\varepsilon}_R} & = & \varepsilon^* \\ \frac{\tilde{T}k_B}{\tilde{\varepsilon}_R} & = & T^* \\ \frac{\tilde{P}\tilde{\sigma}_R^3}{\tilde{\varepsilon}_R} & = & P^* \\ \frac{q}{\sqrt{4\pi\varepsilon_0\sigma_R\varepsilon_R^\eta}} & = & q^* \end{array} \right\} \overset{\not{\text{Sim.}},\tilde{T},\tilde{P},\tilde{q}}{\rightarrow} \left\{ \begin{array}{rcccl} \Delta_v H^* & = & \frac{\tilde{\Delta_v H}}{\tilde{\varepsilon}_R} & \rightarrow & \tilde{\Delta_v H} \\ \rho^* & = & \tilde{\rho}\tilde{\sigma}_R^3 & \rightarrow & \tilde{\rho} \\ D^* & = & \frac{\tilde{D}}{\tilde{\sigma}_R\sqrt{m_R/\tilde{\varepsilon}_R}} & \rightarrow & \tilde{D} \end{array} \right\}.$$

Dabei ist $\eta \in \{1, 3, 5\}$, je nachdem, ob es sich bei der Ladung um einen Monopol, Dipol oder Quadrupol handelt.

Es ist somit möglich, die Funktionswerte von Gradientenkomponenten oder Einträgen der Hesse-Matrix ohne Simulationen zu berechnen. In der Praxis hat sich der zugehörige Algorithmus jedoch nicht als geeignet erwiesen (vergleiche Abschnitt 4.3.2). Es existieren zwar für bestimmte Zielgrößen Korrekturen in Bezug auf Temperatur und Druck, allerdings wird insbesondere zu Beginn der Optimierung das molekulare Modell zu stark verändert, was nicht mehr zu korrigieren ist. Ein weiteres großes Problem ist das statistische Rauschen, was im folgenden anhand der Dichte motiviert werden soll: Eine Simulation berechnet eine verrauschte reduzierte

Dichte $\hat{\rho}^* = \rho^* + \Delta\rho^*$, wobei ρ^* für die korrekte reduzierte Dichte und $\Delta\rho^*$ für deren statistische Ungenauigkeit steht. Die Dichte $\tilde{\rho}$ eines veränderten Kraftfeldparameters ergibt sich dann gemäß

$$\tilde{\rho} = \frac{\hat{\rho}^*}{\tilde{\sigma}_R^3} = \frac{\rho^*}{\tilde{\sigma}_R^3} + \frac{\Delta\rho^*}{\tilde{\sigma}_R^3}. \tag{3.102}$$

Da $\tilde{\sigma}_R^3$ im Nenner steht, ist der Fehler bezüglich $\tilde{\rho}$ umso größer, je kleiner h gewählt wird. Also muß auch hier analog zu Abschnitt 3.5.5 h größer gewählt werden, um den Abstiegstrend der Fehlerfunktion zu reproduzieren. Wird h jedoch zu groß gewählt, unterscheiden sich auch $\tilde{T}, \tilde{P}$ und $\tilde{q}$ zu sehr von T, P und q.
Weitere theoretische Details zum entsprechenden Algorithmus befinden sich in den Anhängen H.1 und H.2.

Die Methode kann bestenfalls in der Nähe des Minimums angewandt werden, das heißt genau dann, wenn sämtliche andere Optimierungsverfahren keine Verbesserungen mehr erzielen. Dann können σ und ε allerdings direkt ohne eine weitere Simulation bestimmt werden, und es ist kein Optimierungsschritt und somit keine Gradienten- beziehungsweise Hesse-Matrix-Berechnung mehr erforderlich. Dieses Verfahren wird in dieser Arbeit als *Methode der reduzierten Einheiten* bezeichnet (Merker u. a., 2012). Allgemein läßt sich diese Methode wie folgt beschreiben:
Es sei X eine beliebige physikalische Zielgröße und y ein beliebiger Kraftfeldparameter mit zulässigem Bereich $\Omega(y)$. Weiterhin seien X^* beziehungsweise y^* deren reduzierte Größen. Es existiere eine auf $\Omega(y)$ stetige, umkehrbare Funktion $h(y)$ sowie eine stetige Funktion $g(X, y^*)$, so daß

$$X^* = g(X, y^*) \cdot h(y). \tag{3.103}$$

Soll nun einen zu einer experimentellen Zielgröße X^{exp} zugehörigen Kraftfeldparameter y_R^{exp} bestimmt werden, so geschieht dies mittels

$$y_R^{\text{exp}} = h^{-1}\left(\frac{X^*}{g(X^{\text{exp}}, y^*)}\right), \tag{3.104}$$

wobei X^* und y^* aus einer vorherigen Simulation bestimmt worden sind. Beispielsweise können so sämtliche Zielgrößen aus Tabelle 3.2 ermittelt werden. Für die Dichte gilt beispielsweise $h(\sigma) = \sigma^3$ und $g(\rho, \sigma^*) = \frac{\rho}{(\sigma^*)^3}$. Dann kann σ_R^{exp} gemäß

$$\sigma_R^{\text{exp}} = \sqrt[3]{\frac{\rho^*(\sigma^*)^3}{\rho^{\text{exp}}}} \tag{3.105}$$

berechnet werden. Ähnliches gilt für die Verdampfungsenthalpie: Hier gilt $h(\varepsilon) = \frac{1}{\varepsilon}$ und $g(\Delta_v H, \varepsilon^*) = \Delta_v H \cdot \varepsilon^*$. Dann kann $\varepsilon_R^{\text{exp}}$ gemäß

$$\varepsilon_R^{\text{exp}} = \frac{\Delta_v H^{\text{exp}}}{\Delta_v H^*}\varepsilon^* \tag{3.106}$$

berechnet werden. Es ist zu beachten, daß die ersetzte Simulation zu einer anderen Temperatur und zu einem anderen Druck (bei VLE-Simulationen nur zu einer anderen Temperatur) durchgeführt wurde. Da die LJ-Parameter temperatur- und druckunabhängig sind, sind

sie so einzustellen, daß Gleichung (3.104) für möglichst viele Temperaturen beziehungsweise Drücke gilt. Hier werden nur VLE-Simulationen, das heißt nur temperaturabhängige Zielgrößen, betrachtet. Dies erfolgt in dieser Arbeit durch eine Approximation, wobei das zugehörige Minimierungsproblem mithilfe eines Newton-Verfahrens gelöst wird. Da sich die Temperatur bei Variation von ε_R ebenfalls ändert, muß dies im Algorithmus folgendermaßen berücksichtigt werden: Die experimentellen Zielgrößen X^{exp} sind in Abhängigkeit von der Temperatur zu betrachten: $X^{\text{exp}} = X^{\text{exp}}(T) = X^{\text{exp}}(T^* \varepsilon_R / k_B)$ (vergleiche Tabelle 3.2). Nur dann können die resultierenden Kraftfeldparameter die experimentellen Zielgrößen zwar zu anderen, aber korrekten Temperaturen reproduzieren. Dazu müssen jedoch experimentelle Temperaturfits für die betrachtete Substanz exisitieren, was allerdings an der Phasenübergangskurve in den meisten Fällen, zumindest für kleine Moleküle, gewährleistet ist. Für die Siededichte wird hier der Guggenheim-Fit aus Gleichung (3.62) und der Verdampfungsenthalpie-Fit aus Gleichung (H.2) verwendet.

Die Argumentation, in welchen Fällen die Methode der reduzierten Einheiten anwendbar ist, ist dieselbe wie oben: Die reduzierten Größen X^* und y^* werden in einer vorherigen Simulation bestimmt, der eine Zielgröße X und ein Kraftfeldparameter y zugrundeliegen. Der Übergang $y \to y^{\text{exp}}$ darf das molekulare Modell nur vernachlässigbar modifizieren. Daher kann diese Methode nur als letzter Schritt angewandt werden. Es ist zu beachten, daß sich auch Partialladungen, Dipol- oder Quadrupolmoment geringfügig ändern.

Die Methode der reduzierten Einheiten ist bereits in Merker u. a. (2012) anhand von Cyclohexanol in der Nähe des Minimums erfolgreich angewandt worden. Der vorgeschaltete Optimierungsalgorithmus war dabei das Verfahren nach Stoll aus Abschnitt 3.2.3.

3.6.5 Angepaßtes Gauß-Newton-Verfahren, eigene Variante

Ein weiterer Unterschied zwischen dem in Abschnitt 3.2.2 vorgestellten Verfahren nach Ungerer und Bourasseau und der in Abschnitt 3.2.3 vorgestellten Methode nach Stoll liegt in der Tatsache, daß letztere die effiziente Gradientenberechnung aus Abschnitt 3.6.2 verwendet. Die Bedingungen (3.98) oder (3.100) werden jedoch vom Verfahren nach Stoll nicht überprüft, so daß Fehler in den Gradientenkomponenten nicht ausgeschlossen sind.
Die partiellen Ableitungen $\frac{\partial f_i^{\text{sim}}}{\partial x_j}$ werden für $i = 1, ..., n$ und $j = 1, ..., N$ mittels der finiten Differenzen $f_i^{\text{sim}}(x^{(k)} + v_j) - f_i^{\text{sim}}(x^{(k)})$ für eine Iteration $x^{(k)}$ berechnet. Die Lösung des LGS ist der Kraftfeldparametervektor $x_{\mathcal{V}}^{(k+1)} := \left(\left(x_1^{(k+1)} \right)_{\mathcal{V}}, ..., \left(x_N^{(k+1)} \right)_{\mathcal{V}} \right)^T$ bezüglich der Basis $\mathcal{V} := \{v_1, ..., v_N\}$. Den kartesischen Parametervektor erhält man durch die Basistransformation

$$x^{(k+1)} = x^{(k)} + \sum_{j=1}^{N} \left(x_j^{(k+1)} \right)_{\mathcal{V}} \cdot v_j. \tag{3.107}$$

Um jedoch eine Situation wie in Abbildung 3.3 auszuschließen, ist es notwendig, das Verfahren zu modifizieren. Zwar wird das Verfahren in dieser Arbeit nur in der Nähe des Minimums ver-

wendet werden, allerdings könnte unter Umständen auch dort eine derartige Situation auftreten. Daher wird das Verfahren nach Stoll hier mit einem Trust-Region-Ansatz (siehe Abschnitt 3.4.3) verknüpft: Für jede Iteration $x^{(k)}$ wird ein Vertrauensbereich $U_{\Delta_k}(x^{(k)})$ mit $x^{(k)} \in U_{\Delta_k}(x^{(k)})$ mit $\Delta_k > 0$ betrachtet. Falls die kartesische Lösung des LGS $x^{(k+1)}$ nicht in $U_{\Delta_k}(x^{(k)})$ liegt, so wird

$$\tilde{x}^{(k+1)} := \min_{y \in \partial U_{\Delta_k}(x^{(k)})} q_{\text{Stoll}}(y) \tag{3.108}$$

berechnet und gemäß Gleichung (3.38) mit $F_{\text{Stoll}}(y)$ verglichen. Dabei ist $q_{\text{Stoll}}(y)$ das innerhalb des Verfahrens nach Stoll verwendete quadratische Approximationsmodell, welches auf einer Taylorentwicklung der zu optimierenden Zielgrößen f_i^{sim} (siehe Gleichung (3.6)) basiert. Die Akzeptanz von $\tilde{x}^{(k+1)}$ als neue Iteration geschieht genauso wie beim Trust-Region-Verfahren aus Abschnitt 3.4.3. Wird die neue Iteration akzeptiert, so wird der Vertrauensbereich vorsichtshalber vergrößert, ansonsten wird er verkleinert.

Die Methode wird in dieser Arbeit stets als *Variante des Verfahrens nach Stoll* bezeichnet. Es ist zu beachten, daß auch die Methode ebenfalls gradientenbasiert ist. Da sie nur in einer kleinen Umgebung des Minimums angewendet werden soll, muß gewährleistet sein, daß das Ausmaß an statistischem Rauschen so gering ist, daß die Richtung des Gradienten korrekt berechnet wird. Durch die Kombination mit der effizienten Gradientenberechnung, welche sich jedoch nicht auf F selbst, sondern auf die Zielgrößen bezieht, durch die ein erheblicher Rechenaufwand eingespart werden kann, lohnt es sich, bei der Berechnung von $F(x^k)$ einen längeren Produktionslauf durchzuführen, um glattere Funktionswerte zu erhalten. Weiterhin werden die in Abschnitt 3.5.2 und Anhang H.1 angesprochenen temperaturabhängigen Fits der zu optimierenden physikalischen Zielgrößen verwendet. Diese glätten die Fehlerfunktion zusätzlich.
Ein weiterer Vorteil der hier beschriebenen Modifikation des Verfahrens nach Stoll besteht in der Tatsache, daß diese auch im Falle von $N > n$ anwendbar ist. Das zu lösende LGS ist nach Lemma 3.2.2 zwar nach wie vor singulär, allerdings ist ein Minimum von q_{Stoll} am Rand des aktuellen Vertrauensgebiets stets bestimmbar. Das Abbruchkriterium für die Variante des Verfahrens nach Stoll ist dann $\Delta_k < \Delta_{\min}$.

3.6.6 Startwerte und Kombination

Die Bestimmung geeigneter Startwerte ist keineswegs trivial, jedoch unentbehrlich für sämtliche lokale Optimierungsverfahren, vor allem wenn sie gegen ein globales Minimum konvergieren sollen. Geeignete Startwerte erhöhen weiterhin die Konvergenzgeschwindigkeit der numerischen Verfahren, denn je näher der Startwert am Minimum liegt, desto weniger Iterationen sind notwendig. Die Entwicklung eines effizienten Algorithmus für eine generische Startwertbestimmung ist ein schwieriges Problem, welches nicht generisch gelöst werden kann, da die zugrundeliegende Funktion beliebig kompliziert und zeklüftet sein kann.

Die einfachste, aber aufwendigste Möglichkeit besteht darin, Startwerte per Hand zu bestimmen. Im Falle der LJ-Parameter beispielsweise kann angenommen werden, daß eine Erhöhung von σ auch bei größeren Teilchenabständen zu Abstoßungen führt und somit die Dichte sinkt,

und daß eine Erhöhung des Energieparameters ε stärkere intermolekulare Anziehungen bewirkt und somit die Verdampfungsenthalpie steigt. Dies sind physikalisch fundierte Annahmen, die sich auch zumeist als korrekt herausgestellt und zu zufriedenstellenden Ergebnissen geführt haben, allerdings kann die Unabhängigkeit der LJ-Parameter in der Praxis nicht vorausgesetzt werden. Es kann nicht erwartet werden, daß ρ nur von σ und $\Delta_v H$ nur von ε abhängt. In einigen wenigen Fällen hat eine derartige manuelle Parametereinstellung zu einer guten bis sehr guten Reproduktion von experimentellen Zielgrößen zu verschiedenen Temperaturen geführt, zum Beispiel im Falle von Ionischen Flüssigkeiten (Köddermann, 2008). Es ist jedoch stets eine extrem hohe Anzahl an Testsimulationen durchzuführen, ohne daß der Erhalt optimaler Kraftfeldparameter garantiert ist. Bei großem N ist dieser Ansatz in der Praxis nicht realisierbar. Es ist bestenfalls möglich, eine überschaubare Anzahl an Testsimulationen parallel durchzuführen, um eine grobe Vorstellung von der Fehlerfunktion beziehungsweise den physikalischen Eigenschaften in Abhängigkeit von den Kraftfeldparametern zu gewinnen und somit geeignete Startparameter zu erhalten.

Eine weitere Möglichkeit zur Startwertbestimmung besteht in der Verwendung der in Abschnitt 3.4.2 angesprochenen *heuristischen* Schrittweite. Durch große Schritte in steilen Bereichen der Fehlerfunktion können schnell neue Startwerte bestimmt werden. Bei zu großen Schritten jedoch besteht die Gefahr, über das Minimum hinauszuzielen oder aber an den Rand des zulässigen Gebiets zu gelangen. Im letzten Fall ist es jedoch auch denkbar, den so erhaltenen Randpunkt als Zentrum eines neuen zulässigen Gebiets zu verwenden. Falls beispielsweise die Fehlerfunktion im vorher festgelegten zulässigen Gebiet kein lokales Minimum besitzt, so werden die Optimierungsverfahren stets in Richtung des Rands laufen und ab einem bestimmten Punkt keine signifikanten Verbesserungen mehr erzielen. Bei ungünstigen Startparametern empfiehlt es sich daher, zunächst eine moderate heuristische Schrittweite zu verwenden, um festzustellen, ob mit wenig Aufwand bessere Startwerte in flacheren Bereichen erzielt werden können oder ob die Algorithmen an den Rand des zulässigen Gebiets gelangen, welches dann verändert werden muß, so daß diese überhaupt zum Erfolg führen können.

Die in dieser Arbeit vorgeschlagene Lösung besteht darin, die Stärken verschiedener Optimierungsverfahren auszunutzen und die Algorithmen geeignet miteinander zu kombinieren. Derartige *hybride Verfahren* sind an den aktuellen Funktionsbereich adaptiert. Es werden dabei drei verschiedene Funktionsbereiche betrachtet:

1. **Große Distanz zum globalen Minimum:** Sind keine geeigneten Startwerte in der Nähe des globalen Minimums vorhanden, so ist ein startwertunabhängiges globales Optimierungsverfahren vorzuschalten, welches effizient in die Nähe des globalen Minimums gelangt. Stochastische Optimierungsverfahren wie Evolutionäre Algorithmen (siehe Abschnitte 3.3.1 und 3.3.2) nehmen per Zufallsprinzip mehrere Stichproben auf einem bestimmten Bereich innerhalb des Parameterraums. Dadurch, daß innerhalb einer Population nicht nur derjenige Parametervektor mit dem kleinsten Fehlerfunktionswert, sondern mehrere Parametervektoren mit kleinen Fehlerfunktionswerten betrachtet werden, können intermediäre lokale Optima übersprungen werden. Durch das effiziente Abtasten des Parameterraums sind globale Optimierungsverfahren sehr schnell dazu in der Lage, Bereiche

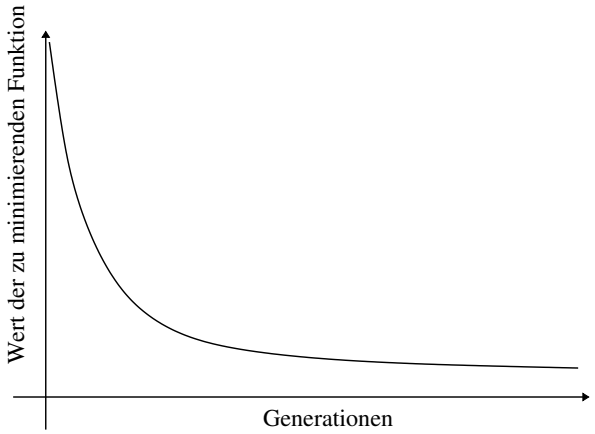


Abbildung 3.18: Typischer Verlauf der Werte, die die zu minimierende Funktion innerhalb eines Evolutionären Algorithmus annimmt, in Abhängigkeit von der Zahl der Generationen: Im Falle eines großen Suchraums werden schnell und effizient Punkte mit kleineren Funktionswerten gefunden. Bei lokalen Verfeinerungen wird die zufallsbasierte Suche immer ineffizienter, so daß nur noch mit hohem Rechenaufwand kleinere Funktionswerte gefunden werden können. Dies führt dazu, daß die Kurve ab einer gewissen Generationszahl stagniert.

zu finden, in denen die Fehlerfunktion abfällt. Besitzt die Fehlerfunktion nicht allzu viele intermediäre lokale Minima mit Funktionswerten in derselben Größenordnung wie der Funktionswert des globalen Minimums, so wird der Algorithmus schnell und effizient in die Nähe eines globalen Minimums gelangen. Auch DesParO (siehe Abschnitt 3.3.3) kann als globaler Optimierer vorgeschaltet werden, wenn die zu interpolierende Fehlerfunktion nicht zu komplex ist.

2. **Geringe Distanz zum globalen Minimum, abfallende Fehlerfunktion:** Je näher ein globaler Optimierungsalgorithmus zum Minimum gelangt, desto mehr unnötige Funktionsauswertungen sind zu verzeichnen. Das liegt daran, daß ein derartiges Verfahren nicht gerichtet ist, und die zufällige Wahl von Parametervektoren macht es in der Nähe des Minimums ineffizient. Bereiche, in denen die Fehlerfunktion weiter abfällt, sind hier nur noch mit hohem Rechenaufwand zu identifizieren. Der Suchbereich muß in jedem Fall eingeschränkt werden. Allerdings sind für lokale Verfeinerungen zum Minimum gerichtete Verfahren besser geeignet. Abbildung 3.18 zeigt, daß die Konvergenzgeschwindigkeit eines evolutionären Algorithmus typischerweise mit zunehmender Generationszahl abnimmt. Als Beispiel hierzu ist die Arbeit von Barnes u. Gelb (2007) zu nennen, die bereits in Abschnitt 3.3.1 angesprochen wurde: Die dort angegebenen Kurven werden ab einer Generationszahl von 100 stets deutlich flacher. Für lokale Verfeinerungen sind die in Abschnitt 3.4 vorgestellten numerischen Optimierungsverfahren besser geeignet. Für den vorgeschalteten globalen Optimierer ist ein geeignetes Abbruchkriterium zu finden, und die finalen Parameter sind als Startwerte für die anschließende gradientenbasierte Optimierung zu verwenden.

Jedes einzelne gradientenbasierte Verfahren hat in bestimmten Funktionsbereichen stets gewisse Vor- und Nachteile, so daß sich auch hierbei eine Kombination der Methoden

empfiehlt. Kombination bedeutet in diesem Fall, daß die letzte Iteration der einen Methode als Startwert für die andere verwendet wird. Somit kann stets der für ein Verfahren am besten geeignete Funktionsbereich optimal ausgeschöpft werden. Zu Beginn, das heißt in steilen Funktionsbereichen, eignet sich am besten die Methode des steilsten Abstiegs, da es zunächst das Ziel ist, möglichst schnell in die Nähe des Minimums zu gelangen. Der negative Gradient ist in diesem Fall die effizienteste Abstiegsrichtung, allerdings können in flacheren Bereichen kleinere Änderungen mithilfe dieses Verfahrens kaum noch erkannt werden, so daß oftmals weit über das Minimum hinausgezielt wird. Das Resultat ist eine langsame Konvergenz der Armijo-Schrittweite, deren Ergebnis im Vergleich zum Rechenaufwand keine nennenswerte Verbesserung mehr beinhaltet. Daher sollte die Schrittweitensteuerung nach einer bestimmten Anzahl an Armijo-Schritten abgebrochen und die letzte Iteration als Startwert für ein anderes Verfahren verwendet werden. Oftmals ist das Minimum zu diesem Zeitpunkt immer noch so weit entfernt, daß Verfahren, die eine Hesse-Matrix benötigen, aufgrund des Rechenaufwands noch nicht verwendet werden sollten. Die CG-Verfahren berücksichtigen zwar keine Krümmungseigenschaften, haben jedoch im Vergleich zur Methode des steilsten Abstiegs den Vorteil, daß sie zwei aufeinanderfolgende Abstiegsrichtungen miteinander kombinieren und somit den Funktionsverlauf besser einschätzen können. Daher empfiehlt sich an dieser Stelle der Einsatz eines CG-Verfahrens. Erst wenn das CG-Verfahren nach einer bestimmten Anzahl an Armijo-Schritten keinen kleineren Fehlerfunktionswert mehr findet, sollte über die Wahl von Methoden nachgedacht werden, die eine Hesse-Matrix verwenden. Das Newton-Raphson-Verfahren, die Quasi-Newton-Verfahren und die Trust-Region-Methode zusammen mit dem doppeltem Hundebein führen in den meisten Fällen nicht zum Ziel, da die Hesse-Matrix nicht positiv definit ist. Die positive Definitheit ist bestenfalls in einer kleinen Umgebung des Minimums erfüllt, so daß die Hesse-Matrix in anderen Fällen oftmals unnötig berechnet wird. Daher sollte zunächst ein Trust-Region-Verfahren zum Einsatz kommen, welches das Teilproblem exakt löst (siehe Abschnitt 3.4.3). Dieses verwendet gegebenenfalls später einen Newton-Raphson-Schritt.

In Abschnitt 4.2 wird eine detaillierte Bewertung der einzelnen Verfahren anhand von simulierten Simulationen durchgeführt. Die zugrundeliegende Fehlerfunktion ist zunächst glatt, und in einem zweiten Schritt wird künstliches Rauschen auf die betrachteten physikalischen Eigenschaften addiert, um Molekulare Simulationen zu imitieren. Im glatten Fall haben sich die CG-Verfahren und das Trust-Region-Verfahren mit exakter Lösung des Teilproblems als am besten geeignet herausgestellt, das heißt, sie sind sowohl in steilen als auch in flachen Bereichen der Funktion anwendbar. Am Anfang sollte jedoch stets die Methode des steilsten Abstiegs verwendet werden, um mit möglichst geringem Rechenaufwand möglichst schnell in die Nähe des Minimums zu gelangen. Die Konvergenz der CG-Verfahren und des Trust-Region-Verfahrens sind ähnlich, allerdings sollte sich aufgrund des geringeren Rechenaufwands zunächst für ein CG-Verfahren entschieden werden. Erst wenn bei Verwendung des Trust-Region-Verfahrens das Vertrauensgebiet so klein ist, daß das Minimum im Inneren angenommen wird, ist es denkbar, einen Newton-Raphson- oder Quasi-Newton-Schritt anzuwenden. Auch die Trust-Region-Methode mit dem doppelten Hundebein wird in diesem Fall eine Newton-Richtung benutzen. Es ist jedoch zu beachten,

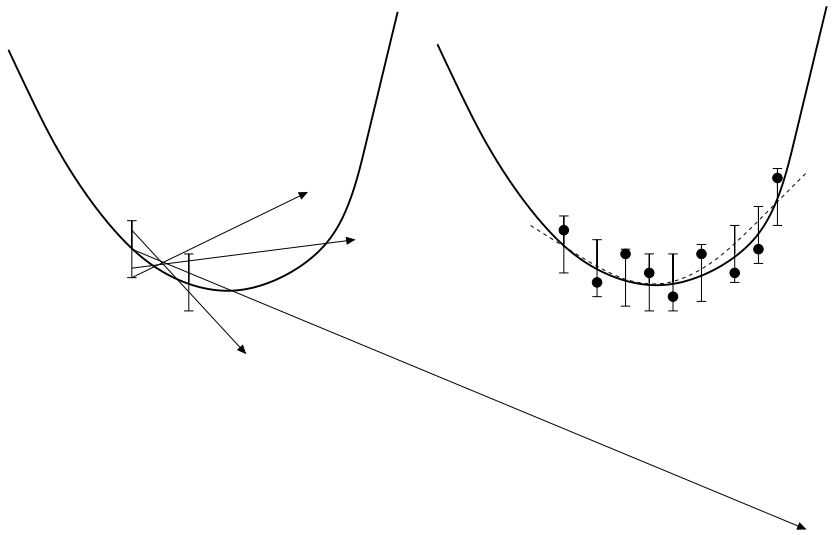


Abbildung 3.19: Motivation zur Optimierung in der Nähe des Minimums bei verrauschten Daten: Gradientenbasierte Verfahren werden von einem gewissen Punkt an nicht mehr erfolgreich sein. Daher ist eine Modellierung der Fehlerfunktion notwendig, um das Rauschen herauszufiltern. Dies leistet zum Beispiel die Variante des Verfahrens nach Stoll.

daß aufgrund der in Abschnitt 3.5.5 theoretisch fundierten Sachverhalte sämtliche gradientenbasierte Verfahren im Falle von Rauschen ab einer gewissen Iteration in der Nähe des Minimums nicht mehr anwendbar sind. Daher ist *a priori* nicht festgelegt, bis wohin eine derartige Kombination verwendet werden kann. Dies wird im Einzelfall getestet werden. Die positive Definitheit der Hesse-Matrix kann bei der vorliegenden Fehlerfunktion nicht vorausgesetzt werden, noch nicht einmal lokal. Daher müssen das Newton-Verfahren und die Quasi-Newton-Verfahren für die vorliegende Problemstellung als ungeeignet eingestuft werden. Sie sind bestenfalls als 'letzter Test' am Ende des Kombinationsalgorithmus einsetzbar. Allerdings werden in dieser Arbeit andere Verfahren betrachtet, die mit weniger Rechenaufwand zum Ziel führen.

3. **Umittelbare Umgebung des globalen Minimums:** Erst in einer umittelbaren Umgebung des globalen Minimums, das heißt, um noch sehr kleine Verfeinerungen durchzuführen, können das Verfahren von Ungerer und Bourasseau (Abschnitt 3.2.2) oder die Methode von Stoll (Abschnitt 3.2.3) eingesetzt werden. In dieser Arbeit werden die in Abschnitt 3.6.5 vorgestellte Variante des Verfahrens nach Stoll sowie das exakte Trust-Region-Verfahren kombiniert mit den Temperaturfits aus Abschnitt 3.5.2 und Anhang H.1 verwendet. Gradientenbasierte Verfahren werden irgendwann aufgrund des statistischen Rauschens keine Verbesserungen mehr erzielen, da die Richtung des Gradienten falsch berechnet wird, oder aber, es ist eine Vielzahl an Armijo-Schritten notwendig, um einen kleineren Fehlerfunktionswert zu erhalten. Das Rauschen muß mittels einer Modellierung beziehungsweise Approximation in der Nähe des Minimums herausgefiltert werden, um dann das Minimum des Modells zu bestimmen, entweder das globale oder das Minimum innerhalb eines Vertrauensbereichs. Die Motivation hierzu ist in Abbildung 3.19 dargestellt. Aufgrund der Taylor-Entwicklungen wird eine derartige Modellierung durch die Variante des Verfahrens nach Stoll und auch durch das exakte Trust-Region-Verfahren realisiert.

Allerdings ist auch hier die Berechnung eines Gradienten erforderlich, allerdings beinhaltet das Verfahren eine vorgeschaltete Glättung über den jeweiligen Temperaturbereich. Kapitel 6 behandelt den Ersatz von gradientenbasierten Verfahren durch die neuartige, ableitungsfreie Methode DGAFO. Diese führt ebenfalls eine Modellierung der Fehlerfunktion durch. Ein späterer Vergleich ist somit unentbehrlich.
Ganz zum Schluß ist, falls möglich, die Methode der reduzierten Einheiten aus Abschnitt 3.6.4 anzuwenden.

4 Bewertung der eingesetzten Optimierungsmethoden anhand von simulierten Simulationen

Es sollen nun die in Kapitel 3 beschriebenen Methoden praktisch ausgewertet werden, und zwar unter den Aspekten Automatisierbarkeit, Anwendbarkeit auf die Problemstellung, Effizienzerhöhung und Startwertbestimmung beziehungsweise Kombination zum möglichst exakten Erhalt des globalen Minimums. Eine detaillierte Verfahrensevaluation ist aufgrund des hohen Rechenaufwands allerdings nicht anhand von Molekularen Simulationen selbst durchführbar. Daher müssen diese in geeigneter Weise ersetzt werden.
Wie bereits in Abschnitt 3.5.2 motiviert, werden die Korrelationsfunktionen aus Stoll u. a. (2001) verwendet, um aufwendige Simulationen geeignet zu ersetzen. Da diese nur auf Zweizentren-Lennard-Jones-Teilchen mit einem dipolaren (Stoll u. a., 2003a) oder quadrupolaren Wechselwirkungsbeitrag (Stoll u. a., 2001) anwendbar sind, wird die Verfahrensbewertung am Beispiel von Stickstoff (N_2) an der Phasenübergangskurve flüssig $\rightarrow$ gasförmig durchgeführt. Stickstoff ist aufgrund der Dreifachbindung zwischen den beiden N-Atomen ein interessanter Testfall, was die molekulare Modellbeschreibung anbelangt. Allerdings ist letztere auf der anderen Seite auch nicht zu komplex, da Stickstoff nur aus zwei Atomen besteht und somit ein sehr kleines Teilchen ist. Außerdem ist eine Vielzahl an experimentellen Daten bekannt und verfügbar, und das Zweizentren-Lennard-Jones-Modell mit quadrupolarem Moment hat sich in vielen Fällen als geeignet erwiesen (Buckingham, 1959; Murthy u. a., 1980; Kriebel u. a., 1996; Vrabec u. a., 2001). Somit eignet sich Stickstoff ideal für eine eingehende und detaillierte Verfahrensbewertung.
Um Molekulare Simulationen zu imitieren, wurde künstlich gleichverteiltes Rauschen addiert, dessen Ausmaß von der jeweiligen anzupassenden physikalischen Eigenschaft abhing. Weiterhin wurden acht verschiedene Optimierungsaufgaben betrachtet. Näheres hierzu wurde bereits in Abschnitt 3.5.2 erläutert. In Abschnitt 4.1 wird zum einen die Konzipierung der Verfahrensbewertung diskutiert, und zum anderen werden die daraus erhaltenen optimalen Ergebnisse für unverrauschte simulierte Simulationen zu verschiedenen Temperaturen angegeben. Abschnitt 4.2 befaßt sich mit der detaillierten Verfahrensbewertung selbst, zeigt, daß eine automatisierte Optimierung von Kraftfeldparametern mit gradientenbasierten Verfahren möglich ist und vergleicht die verschiedenen eingesetzten gradientenbasierten Optimierungsverfahren. Es wird eingehend diskutiert, welche Verfahren auf die vorliegende Problemstellung anwendbar sind und welche nicht. Da sich die Bewertung der Konvergenzgeschwindigkeit zunächst wie üblich auf die Anzahl an Iterationen bezieht, werden in Abschnitt 4.3 die Ergebnisse der in den Abschnitten 3.6.2 und 3.6.3 eingeführten Methoden zur Reduktion der Funktionsauswertungen beziehungsweise Simulationen bewertet. Es wird praktisch dargelegt, daß auch die Effizienz der gradientenbasierten Verfahren erhöht werden kann. Es ist zu beachten, daß die Effizienz, wie in

Abschnitt 3.6.1 dargelegt, zusätzlich durch Parallelisierung erhöht werden kann, falls genügend Rechenressourcen zur Verfügung stehen. Wie in Abschnitt 3.6.6 motiviert, sind gradientenbasierte Verfahren alleine nicht die Lösung für die Optimierung von Kraftfeldparametern. Vielmehr wird in bestimmten Fällen ein Kombinationsalgorithmus anzuwenden sein, um einerseits geeignete Startparameter zu finden und andererseits so nah wie möglich an das globale Minimum zu gelangen. In Abschnitt 4.4 werden zunächst einige der in Abschnitt 3.3 beschriebenen globalen Optimierungsverfahren angewandt. Dabei wird diskutiert, ob und inwieweit derartige Verfahren zur Startwertbestimmung für gradientenbasierte Verfahren eingesetzt werden können. Schließlich wird in Abschnitt 4.5 ein kompletter Kombinationsalgorithmus durchgeführt.

4.1 Simulierte Simulationen

Eine detaillierte Beschreibung des hier eingesetzten generischen Optimierungsablaufs und dessen praktische Umsetzung befinden sich in Abschnitt 3.1 und Anhang G.1 sowie in Hülsmann u. a. (2010a). Jeder iterative gradientenbasierte Algorithmus beginnt mit einem Startvektor $x^{(0)}$, welcher innerhalb des Optimierungsablaufs stetig verbessert wird. Für jede Iteration $x^{(k)}$, $k \in \mathbb{N}$, wird innerhalb des Verfahrens stets die Siededichte ermittelt, entweder zusammen mit der Verdampfungsenthalpie oder dem Dampdruck. Da es sich hierbei gemäß Abschnitt 3.5.2 zunächst um reduzierte Größen handelt, werden die in Tabelle 3.2 angegebenen Formeln verwendet, um daraus physikalische Einheiten zu berechnen. Die Siededichte wird dabei in $\mathrm{kg/m^3}$, die Verdampfungsenthalpie in kJ/mol und der Dampdruck in MPa gemessen.
Die hier betrachtete Fehlerfunktion bezieht sich auf die physikalischen Größen in einem gewissen Temperaturbereich $\mathcal{T}$, das heißt, es gilt hier:

$$F(x) := \sum_{i=1}^{n} \sum_{T \in \mathcal{T}} w_{i,T} \left(\frac{f_{i,T}^{\mathrm{exp}} - f_{i,T}^{\mathrm{sim}}(x)}{f_{i,T}^{\mathrm{exp}}} \right)^2 . \tag{4.1}$$

Dabei ist $x = (\varepsilon, \sigma, Q^2, L)^T \in \mathbb{R}^4$ der Kraftfeldparametervektor, $f_{i,T}^{\mathrm{sim}}(x)$ die i-te physikalische Eigenschaft zu einer bestimmten Temperatur T und $f_{i,T}^{\mathrm{exp}}$ die entsprechende experimentelle Größe, an die angepaßt wird. Die Gewichte $w_{i,T}$ sind ebenfalls temperaturabhängig definiert, allerdings wird hier oftmals $\forall_{i,T}\ w_{i,T} = 1$ gesetzt, da innerhalb der Verfahrensbewertung sämtliche Eigenschaften als gleichwertig angesehen werden. In diesem Kapitel gilt stets $n = 2$. Es seien $w_{\rho_l,T}$ das Gewicht der Siededichte zu einer Temperatur, $w_{\Delta_v H,T}$ das entsprechende Gewicht der Verdampfungsenthalpie und $w_{p_\sigma,T}$ das des Dampfdrucks. Aufgrund des in Abschnitt 3.5.2 angegebenen Ausmaßes an statistischem Rauschen gilt hier entweder $\forall_{i,T}\ w_{i,T} = 1$ oder $\forall_T\ w_{\rho_l,T} = 2w_{\Delta_v H,T} = 6w_{p_\sigma,T}$, wobei im letzten Fall $\sum_{i=1}^{2} \sum_{T \in \mathcal{T}}\ w_{i,T} = 1$ gilt.
Die aus der Korrelationsfunktion berechneten, das heißt simulierten, Größen werden in physikalischen Einheiten in die Fehlerfunktion aus Gleichung (4.1) eingesetzt. Sämtliche experimentellen Daten wurden aus der NIST-Datenbank (NIST, 2011) genommen. Die Abbruchkriterien wurden für die einzelnen acht Optimierungsaufgaben folgendermaßen festgesetzt: Es konnte beobachtet werden, daß die Methode des steilsten Abstiegs oftmals schnell zu signifikanten Verbesserungen in den Fehlerfunktionswerten führte, ab einem bestimmten Punkt jedoch die Armijo-Schrittweitensteuerung keine kleineren Funktionswerte mehr mit akzeptablem Rechen-

aufwand finden konnte. Eine Optimierung wurde stets abgebrochen, wenn die Schrittweitensteuerung mehr als 100 Iterationen benötigte. Sobald also die Methode des steilsten Abstiegs keine Verbesserungen mehr lieferte, wurde der aktuelle Funktionswert für die Festlegung eines Abbruchkriteriums verwendet. Dieses Abbruchkriterium wurde für den Verfahrensvergleich benutzt. Andererseits wurde stets überprüft, wie nahe die besten Optimierungsmethoden an das Optimum gelangen, das heißt, es wurde solange optimiert, bis die Armijo-Methode mehr als 100 Iterationen benötigte.
Die Startparameter basierten auf der Optimierung in Vrabec u. a. (2001): $Q^2 = 0.020727$ $(\mathrm{Dnm})^2$ und $L = 0.10464$ nm. Weiterhin wurden $\varepsilon = 0.3101$ kJ/mol und $\sigma = 0.331$ nm gewählt, welche sich von den Werten in Vrabec u. a. (2001) unterscheiden, denn für eine Verfahrensbewertung ist es nicht sinnvoll, bereits optimale Parameter als Startwerte zu verwenden. Es ist zu beachten, daß die Anzahl an zu optimierenden Parameter von der Methode her unbeschränkt ist. Allerdings reichen hier, auch um unnötig hohen Rechenaufwand zu vermeiden, vier Parameter völlig aus, um eine Bewertung durchführen zu können. Um physikalisch sinnvolle Parameter während der gesamten Optimierung zu gewährleisten, wurde das zulässige Gebiet folgendermaßen festgesetzt: Eine maximale Veränderung von σ lag bei 10% und von ε bei 40%. Da gemäß Stoll u. a. (2001) die Korrelationsfunktionen nur für $L^* \in [0, 0.8]$ und $Q^{2*} \in [0, 4]$ angewendet werden kann, konnte aus dem zulässigen Bereich für σ und ε der für L und Q^2 berechnet werden. Es ergab sich $0.2979 \leq \sigma/\mathrm{nm} \leq 0.3641$, $0.18606 \leq \varepsilon/(\mathrm{kJ/mol}) \leq 0.43414$, $0 \leq (Q/\mathrm{Dnm})^2 \leq 0.029$ und $0 \leq L/\mathrm{nm} \leq 0.14895$. Es ist zu beachten, daß verschiedene Startparameter zu verschiedenen lokalen Minima führen können, was zunächst für die Verfahrensbewertung nicht von Belang ist. Ob es sich bei den hier erhaltenen lokalen Optima um globale Optima handelt, wird in Abschnitt 4.4 diskutiert werden.
Da die Korrelationsfunktionen nur im Temperaturbereich $T/T_\mathrm{c} \in [0.55, 0.95]$ gütlig sind, wobei T_c die kritische Temperatur ist, wurde $T/T_\mathrm{c} \in \{0.52, 0.59, 0.67, 0.75, 0.83, 0.91\}$, also $T/\mathrm{K} \in \{65, 75, 85, 95, 105, 115\}$, festgelegt, wobei aus Stetigkeitsgründen angenommen werden kann, daß $T = 65$ K ebenfalls verwendbar ist, da der Tripelpunkt von Stickstoff bei etwa $T = 63$ K liegt. Wurde nur bei einer Temperatur optimiert, so handelte es sich dabei um $T = 75$ K.

Eine eingehende Verfahrensbewertung bei allen acht Optimierungsaufgaben wurde in Hülsmann u. a. (2010b) für $\forall_{i,T}\ w_{i,T} = 1$ durchgeführt. Abbildung 4.1 zeigt die Ergebnisse bei Verwendung der besten Optimierungsverfahren zu allen sechs betrachteten Temperaturen, bei denen simultan optimiert wurde. Dabei wurde kein künstliches Rauschen addiert. Alle drei experimentellen physikalischen Größen konnten sehr gut wiedergegeben werden. Die experimentellen Daten über den gesamten Temperaturbereich wurden, zum Erhalt der schwarzen Kurven, aus Span u. a. (2000) entnommen. Die beiden schwarzen Kurven repräsentieren jeweils Kraftfeldparameter, die am Rand des zulässigen Gebiets liegen. Dabei wurden einerseits Startwerte für ε und σ sowie minimale Werte für Q^2 und L und andererseits maximale Werte für σ und ε sowie Startwerte für Q^2 und L verwendet. Dies führt gemäß Abbildung 4.1 jedesmal zu sehr schlechten Vorhersagen. Somit ist eine automatisierte Parameteroptimierung stets unentbehrlich.

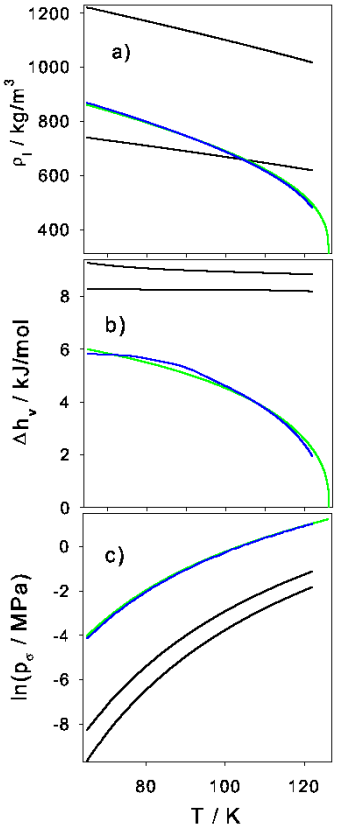


Abbildung 4.1: Optimierungsergebnisse der besten Verfahren zu allen sechs Temperaturen (grüne Kurven) im Vergleich zu den experimentellen Daten (blaue Kurven). Bei allen drei physikalischen Größen (Siededichte, Verdampfungsenthalpie und Dampfdruck) konnten sehr gute Übereinstimmungen gefunden werden. Im Gegensatz dazu führen Randwerte des zulässigen Gebiets zu völlig falschen Ergebnissen (schwarze Kurven). Die Abbildung wurde aus Hülsmann u. a. (2010b) übernommen.

4.2 Bewertung und Vergleich gradientenbasierter Optimierungsverfahren

Eine detaillierte Verfahrensbewertung für alle acht Optimierungsaufgaben wurde in Hülsmann u. a. (2010b) durchgeführt. Da die Vorstellung sämtlicher Ergebnisse den Rahmen dieser Arbeit sprengen würde, werden hier nur die beiden schwierigsten Optimierungsprobleme betrachtet, und zwar die Optimierungsaufgaben 7 und 8. Der Dampfdruck ist im allgemeinen eine schwer zu reproduzierende Größe, da er oftmals mit einem großen Ausmaß an Rauschen behaftet ist. Daher wurde hier eine statistische Unsicherheit von 3% angenommen. Bei Molekularen Simulationen ist es sinnvoll, ein Kraftfeld zu generieren, welches physikalische Daten zu verschiedenen Temperaturen wiedergibt. Da es bei den hier bewerteten Methoden um lokale Optimierungsverfahren handelt, welche lokale Verfeinerungen vornehmen müssen, ist es äußerst wichtig, so viele Informationen wie möglich in die zu optimierende Funktion miteinfließen zu lassen. Bei einer Temperatur ist die Fehlerfunktion wesentlich flacher als bei mehreren Temperaturen (vergleiche

Abschnitt 3.5.3), was dazu führt, daß eine exakte Bestimmung des Minimums schwieriger ist, da es viel mehr Punkte gibt, deren Funktionswerte unter einen gewissen Schwellenwert fallen. Eine ungenaue Bestimmung des Minimums bei nur einer Temperatur kann erhebliche Auswirkungen auf Zielgrößen zu anderen Temperaturen haben. Daher werden hier die beiden Dampfdruck-Optimierungsaufgaben bei sechs Temperaturen diskutiert. Im Gegensatz zu Hülsmann u. a. (2010b), wo $\forall_{i,T}\ w_{i,T} = 1$ gesetzt wurde, wird hier $\forall_T\ w(\rho_l, T) = 6w(p_\sigma, T)$ angenommen. Die hier vorgestellten Ergebnisse sind mit denen in der Veröffentlichung konsistent, das heißt, es stellen sich dieselben Verfahren als geeignet beziehungsweise ungeeignet heraus.

Um die Algorithmen im einzelnen zu diskutieren und zu evaluieren, ist zunächst die Anzahl an Iterationen bis zum Abbruchkriterium ausschlaggebend, da diese ein Indikator für die Konvergenzgeschwindigkeit ist. Die Anzahl an Funktionsauswertungen, welche der Anzahl an durchzuführenden Molekularen Simulationen entspricht, ist ebenfalls von hoher Bedeutung, spielt aber bei der Verfahrensbewertung selbst zunächst keine große Rolle. In Müller u. a. (2008) benötigte das Simplex-Verfahren nach Nelder und Mead aus Abschnitt 3.2.1 zusammen mit dem MD-Simulationstool YASP (Müller-Plathe, 1993) 70 bis 100 Iterationen für die Anpassung von vier Modellparametern an die Siededichte und Verdampfungsenthalpie zu einer Temperatur bis zu einem vergleichbaren Abbruchkriterium. Eine Reduktion der Iterationszahl auf 30 bis 50 stellt somit bereits eine signifikante Verbesserung dar. Konvergiert ein Verfahren innerhalb von zehn oder weniger Schritten, so ist dies ein sehr gutes Ergebnis, und das Verfahren kann als für Molekulare Simulationen besonders gut geeignet bewertet werden. Es ist jedoch zu beachten, daß im Falle des Simplex-Algorithmus die Anzahl an Iterationen gleich der Anzahl an Funktionsauswertungen ist. Da dies bei gradientenbasierten Verfahren nicht der Fall ist, müssen Zusatzüberlegungen getroffen werden, welche die Effizienz der eingesetzten Algorithmen erhöhen. Darauf wird in Abschnitt 4.3 eingehend eingegangen werden.

Bei Optimierungen zu einer Temperatur, also im Falle von unterbestimmten Problemen, kann eine nahezu perfekte Übereinstimmung der zu optimierenden physikalischen Größen mit den experimentellen Werten erwartet werden. Es hat sich jedoch herausgestellt (siehe auch Abschnitt 4.4), daß es in diesem Fall unendlich viele globale Minima geben kann, welche sich am Boden einer Art Regenrinne befinden. Je nach Startwert werden daher von den Optimierungsalgorithmen verschiedene globale Minima gefunden.
Auch bei mehreren Temperaturen kann es mehr als ein lokales Minimum geben. Gemäß Abbildung 3.11 kann auch hier eine Regenrinnengestalt der Fehlerfunktion erwartet werden, allerdings können die lokalen Minima am Boden der Regenrinne punktueller verteilt sein. Derartige Optimierungsprobleme sind gemäß den Ausführungen in den Abschnitten 3.4.2 und 3.6.6 schwieriger zu lösen. Das globale Minimum mit Fehlerfunktionswert 0 kann, wenn ein solches überhaupt existiert, in der Regel nicht gefunden werden. In Hülsmann u. a. (2010b) konnten niemals optimale, aber stets gute und zufriedenstellende Ergebnisse erzielt werden. Bei den hier vorliegenden Gewichtungen wurde die Fehlerfunktion insgesamt flacher, und die Algorithmen konnten näher an das jeweilige lokale Minimum gelangen. Ein globales Minimum kann jedoch nur von einem globalen Optimierer gefunden werden.
Im Falle von künstlichem statistischen Rauschen können sich die Vorhersagen für die zu optimierenden Eigenschaften nur innerhalb eines Fehlerbalkens um den jeweiligen experimentellen

	$x^{(0)}$	ρ_l	p_σ	$F(x^{(0)})$	$\|\|\nabla F(x^{(0)})\|\|$
Optimierungsaufgabe 7: ρ_l, p_σ, sechs Temperaturen, ohne Rauschen	0.310100 0.331000 0.020727 0.104640	5.49%	34.60%	0.054	8.5484
Optimierungsaufgabe 8: ρ_l, p_σ, sechs Temperaturen, mit Rauschen	0.310100 0.331000 0.020727 0.104640	5.48%	34.52%	0.053	9.045

Tabelle 4.1: Startparameter und deren Ergebnisse für die Optimierungsaufgaben 7 und 8: Die Ergebnisse beinhalten neben den Parametern die relativen Fehler bezüglich Siededichte ρ_l und Dampfdruck p_σ sowie die Werte der Fehlerfunktion $F(x^{(0)})$ und die Gradientennormen $||\nabla F(x^{(0)})||$. Im Falle der relativen Fehler sind die MAPE-Werte über alle sechs Temperaturen angegeben. Im Falle von Rauschen (Optimierungsaufgabe 8) wurden sämtliche Ergebnisse über zehn Replikate gemittelt.

Wert befinden. Eine relative Abweichung, die kleiner als die relative aufaddierte statistische Unsicherheit ist, kann daher als optimales Ergebnis betrachtet werden. Die Schwellenwerte für die Abbruchkriterien waren daher bei Rauschen stets etwas größer gewählt worden. Um Zufallsergebnisse auszuschließen, wurden für jeden Optimierungsablauf zehn statistisch unabhängige Replikate ausgeführt und deren Ergebnisse gemittelt. Ein Replikat heißt *erfolgreich*, falls das zugehörige Optimierungsverfahren das Abbruchkriterium erreicht, ohne daß zu einem früheren Zeitpunkt die Schrittweitensteuerung nicht innerhalb von 100 Schritten zu einem kleineren Fehlerfunktionswert geführt hat. Dadurch konnten statistisch repräsentative Ergebnisse erzielt werden.

Bei den unterbestimmten Problemen (Optimierungsaufgaben zu einer Temperatur) konnten die am besten geeigneten Verfahren die experimentellen Daten nahezu perfekt reproduzieren, das heißt, die Fehlerfunktion und die Norm des Gradienten waren stets nahe bei 0. Im Falle von Rauschen wurden die Eigenschaften innerhalb ihrer Fehlerbalken vorhergesagt. Bei mehreren Temperaturen konnte die Siededichte mit weniger als 1%, die Verdampfungsenthalpie und der Dampfdruck mit ca. 3% Abweichung vorhergesagt werden, sowohl mit als auch ohne Rauschen. Dies sind keine optimalen, aber zufriedenstellende Ergebnisse. Die Verfahren konvergierten nur gegen ein lokales und nicht gegen ein globales Minimum. Um letzteres zu garantieren, sind andere Methoden vonnöten (siehe Abschnitte 3.6.6, 4.4 und 4.5). Für die Verfahrensbewertung der gradientenbasierten Verfahren, von denen sowieso lediglich lokale Konvergenz zu erwarten ist, ist dies jedoch nicht von Belang. Oftmals konnte das Abbruchkriterium mit weniger als 20 Iterationen erreicht werden, was zu dem Schluß führte, daß einige der hier angewandten gradientbasierten Verfahren eine effiziente Lösung für die vorliegende Problemstellung sind.

Tabelle 4.1 gibt die Ergebnisse für die Optimierungsaufgaben 7 und 8 mit den in Abschnitt 4.1 angegebenen Startparametern an. Die über alle Temperaturen gemittelten absoluten prozentualen Abweichungen vom Experiment (sogenannte **M***ean* **A***bsolute* **P***ercentage* **E***rror-(MAPE-) Werte*) waren sowohl im Falle der Siededichte als auch im Falle des Dampfdrucks äußerst schlecht. Dies zeigt nochmals deutlich, wie wichtig eine Parameteroptimierung ist. Tabelle 4.2

Algorithmus	# It.	# Eval.	x^{opt}	ρ_l	p_σ	$F(x^{\text{opt}})$	$\|\|\nabla F(x^{\text{opt}})\|\|$
Steilster Abstieg	6	44	0.3044 0.3280 0.0178 0.1122	0.95%	2.83%	0.000441	0.07
Newton-Raphson	12	296	0.3074 0.3272 0.0163 0.1137	0.96%	2.88%	0.000459	0.11
PSB (Quasi-Newton)	99⋆	798	0.3093 0.3339 0.0204 0.1057	2.30%	33.78%	0.050576	8.13
DFP (Quasi-Newton)	9⋆	71	0.3102 0.3308 0.0201 0.1049	5.40%	33.60%	0.051063	8.42
BFGS (Quasi-Newton)	17⋆	98	0.3117 0.3299 0.0152 0.1059	5.03%	29.00%	0.039047	7.78
Fletcher-Reeves (CG)	6	44	0.3043 0.3282 0.0178 0.1125	0.57%	2.90%	0.000433	0.14
Polak-Ribière (CG)	5	38	0.3044 0.3281 0.0178 0.1123	0.87%	2.84%	0.000436	0.08
Trust-Region (DD)	Keine Lösung des Teilproblems gefunden: Hesse-Matrix nicht spd.						
Trust-Region (exakt)	12	185	0.3054 0.3274 0.0186 0.1132	1.10%	2.71%	0.000416	0.10

Tabelle 4.2: Optimierungsergebnisse für Optimierungsaufgabe 7 (Siededichte und Dampfdruck bei sechs Temperaturen, ohne Rauschen): Das Abbruchkriterium war $F(x) \leq 0.000442$. Die Ergebnisse beinhalten neben den Parametern die Anzahl an Iterationen (# It.), die Anzahl an Funktionsevaluationen (# Eval.), die relativen Fehler bezüglich Siededichte (ρ_l) und Dampfdruck (p_σ) sowie die Werte der Fehlerfunktion ($F(x^0)$) und die Gradientennormen ($||\nabla F(x^0)||$). Im Falle der relativen Fehler sind die MAPE-Werte über alle sechs Temperaturen angegeben. Der Stern (*) gibt an, daß das Verfahren das Abbruchkriterium nicht erreicht hat, sondern nach 100 Armijo-Schritten keinen kleineren Fehlerfunktionswert mehr gefunden hat. Dies war für die Quasi-Newton-Verfahren der Fall, bei denen sogar die Akzeptanzbedingung für die Armijo-Schrittweitensteuerung abgeschwächt werden mußte (DFP: $\zeta_A = 0.1$, PSB: $\zeta_A = 0.001$). Die Trust-Region-Methode zusammen mit dem doppelten Hundebein war gar nicht geeignet, da die Hesse-Matrix nicht spd war. Somit konnte keine Lösung des Teilproblems gefunden werden. Es handelte sich um Steilster-Abstieg-Iterationen, allerdings war die Größe des Vertrauensgebiets Δ stets so klein, daß keine signifikant kleineren Fehlerfunktionswerte ermittelt werden konnten. Alle anderen Verfahren konvergierten gegen das gleiche Minimum, was an den Parameterwerten deutlich erkennbar ist. Das Newton-Raphson-Verfahren erreichte das Abbruchkriterium zwar nicht exakt, die Abweichungen bezüglich ρ_l und p_σ lagen dennoch in der gleichen Größenordnung wie bei den anderen Verfahren.

Algorithmus	# Repl.	# It.	# Eval.	x^{opt}	ρ_l	p_σ	$F(x^{\mathrm{opt}})$	$\|\|\nabla F(x^{\mathrm{opt}})\|\|$
Steilster Abstieg	9	5–6	44–45	0.3052 0.3282 0.0174 0.1124	0.84%	3.90%	0.000797	0.77
Newton-Raphson	5	12–13	233–234	0.3068 0.3275 0.0187 0.1138	1.09%	3.32%	0.000695	0.56
PSB (Quasi-Newton)				Kein erfolgreiches Replikat.				
DFP (Quasi-Newton)				Kein erfolgreiches Replikat.				
BFGS (Quasi-Newton)				Kein erfolgreiches Replikat.				
Fletcher-Reeves (CG)	10	4–5	29–30	0.3051 0.3282 0.0174 0.1129	0.56%	4.14%	0.000795	0.74
Polak-Ribière (CG)	9	4–5	28–29	0.3049 0.3280 0.0173 0.1130	0.54%	3.86%	0.000721	0.80
Trust-Region (DD)	10	2–3	35–36	0.3048 0.3280 0.0171 0.1128	0.59%	3.98%	0.000695	0.73
Trust-Region (exakt)	6	3–4	56	0.3014 0.3291 0.0172 0.1100	1.18%	3.52%	0.000687	0.28

Tabelle 4.3: Optimierungsergebnisse für Optimierungsaufgabe 8 (Siededichte und Dampfdruck bei sechs Temperaturen, mit Rauschen): Das Abbruchkriterium war $F(x) \leq 0.001$. Die Ergebnisse beinhalten neben den Parametern die Anzahl an erfolgreichen Replikaten (# Repl.), die Anzahl an Iterationen (# It.), die Anzahl an Funktionsevaluationen (# Eval.), die relativen Fehler bezüglich Siededichte (ρ_l) und Dampfdruck (p_σ) sowie die Werte der Fehlerfunktion ($F(x^0)$) und die Gradientennormen ($||\nabla F(x^0)||$). Im Falle der relativen Fehler sind die MAPE-Werte über alle sechs Temperaturen angegeben. Sämtliche Ergebnisse wurden über zehn statistisch unabhängige Replikate gemittelt. Im Falle des Newton-Raphson-Verfahrens mußte die Akzeptanzbedingung für die Armijo-Schrittweitensteuerung abgeschwächt werden ($\zeta_A = 0.001$). Alle geeigneten Verfahren konvergierten gegen das gleiche Minimum. Bei der Trust-Region-Methode zusammen mit dem doppelten Hundebein waren hier alle zehn Replikate erfolgreich, allerdings ist dies nur als Zufallsergebnis zu werten, da es sich bei den Iterationen um Steilster-Abstieg-Iterationen handelte und durch das Rauschen irgendwann ein signifikant kleinerer Fehlerfunktionswert gefunden werden konnte.

zeigt die Optimierungsergebnisse für Optimierungsaufgabe 7 für alle angewandten gradientenbasierten Verfahren und Tabelle 4.3 für Optimierungsaufgabe 8. Letztere enthält Mittelwerte über zehn statistisch unabhängige Zufallsreplikate. Die aus den Replikaten erhaltenen Resultate sind mithilfe von *Box-Plots* in Abbildung 4.2 veranschaulicht.
Ohne künstliches Rauschen (Tabelle 4.2) wurde oftmals eine Abweichung bezüglich ρ_l von weniger als 1% und bezüglich p_σ von weniger als 3% erzielt. Das Abbruchkriterium $F(x) \leq 0.000442$ wurde von der Methode des steilsten Abstiegs, den CG-Verfahren und dem Trust-Region-Verfahren mit exakter Lösung des Teilproblems erreicht. Das Newton-Raphson-Verfahren war kurz vorher abgebrochen, allerdings lagen die Abweichungen bezüglich ρ_l und p_σ in der gleichen Größenordnung wie bei den anderen Verfahren. All diese Verfahren gelangten in die Nähe des gleichen Minimums. Die finalen Kraftfeldparameter waren sehr ähnlich. Die drei Quasi-Newton-Verfahren brachen weit vor Erfüllung des Abbruchkriteriums ab. Hierbei konvergierte die Armijo-Schrittweitensteuerung nicht innerhalb von 100 Iterationen. Da die Hesse-Matrix nirgends spd war, konnte nicht garantiert werden, daß eine geeignete Abstiegsrichtung gefunden wurde. Die Trust-Region-Methode zusammen mit dem doppeltem Hundebein gelangte zu keiner einzigen weiteren Iteration. Es konnte keine Lösung des Teilproblems gefunden werden, da die Hesse-Matrix von vornherein nicht spd war. Die Methode des steilsten Abstiegs und die CG-Verfahren benötigten deutlich weniger als zehn Iterationen mit einer akzeptablen Anzahl an Funktionsauswertungen, um die Fehlerfunktion um eine Größenordnung und die Norm des Gradienten um zwei Größenordnungen zu verringern. Dies ist als äußerst gutes Ergebnis zu werten, da auch experimentelle Dampfdrücke stets mit derart hohen Unsicherheiten behaftet sind.

Mit künstlichem Rauschen (Tabelle 4.3) war die Abweichung bezüglich p_σ mit 3–4% etwas schlechter. Auch Anwendersicht ist dies jedoch immer noch als gutes Ergebnis zu werten. Das Abbruchkriterium $F(x) \leq 0.001$ wurde von der Methode des steilsten Abstiegs, den CG-Verfahren und beiden Trust-Region-Verfahren in den meisten Fällen erreicht. Bei den Quasi-Newton-Verfahren war aus den oben erwähnten Gründen kein Replikat erfolgreich. Da die Hesse-Matrix nie spd war, konnte die Trust-Region-Methode zusammen mit dem doppelten Hundebein gemäß Abschnitt 3.4.3 lediglich die Richtung des steilsten Abstiegs verwenden. Der Radius Δ des Vertrauensgebiets war so eingestellt, daß ohne Rauschen keine Iteration gefunden werden konnte, so daß $r_k \geq \eta_1 = 0.3$. Im Falle von Rauschen war dies möglich, da irgendwann im Laufe der Lösung des Teilproblems eine Zufallszahl generiert wurde, welche diese Bedingung erfüllte. Daher sind die Ergebnisse der Trust-Region-Methode zusammen mit dem doppelten Hundebein als Zufallsergebnisse zu werten. Bei der detaillierten Verfahrensbewertung in Hülsmann u. a. (2010b) hat sich dieses Verfahren auch im Falle von Rauschen als ungeeignet erwiesen. Die Methode des steilsten Abstiegs und die CG-Verfahren lieferten eine Erfolgsquote von 90-100%. Das Newton-Raphson-Verfahren lieferte 50% und das Trust-Region-Verfahren mit exakter Lösung des Teilproblems 60%. Bei ersterem kam es jedoch oft vor, daß entweder die Richtung des steilsten Abstiegs verwendet wurde oder die Newton-Richtung keine geeignete Abstiegsrichtung war. Bei letzterem stellt die relativ niedrige Erfolgsquote bei dieser Optimierungsaufgabe zwar ein Nachteil dar, allerdings wurde in Hülsmann u. a. (2010b) dargelegt, daß es sich dennoch um ein robustes Optimierungsverfahren handelt, welches teilweise näher an das Minimum gelangt als die anderen Verfahren. Das Newton-Raphson-Verfahren hingegen hat

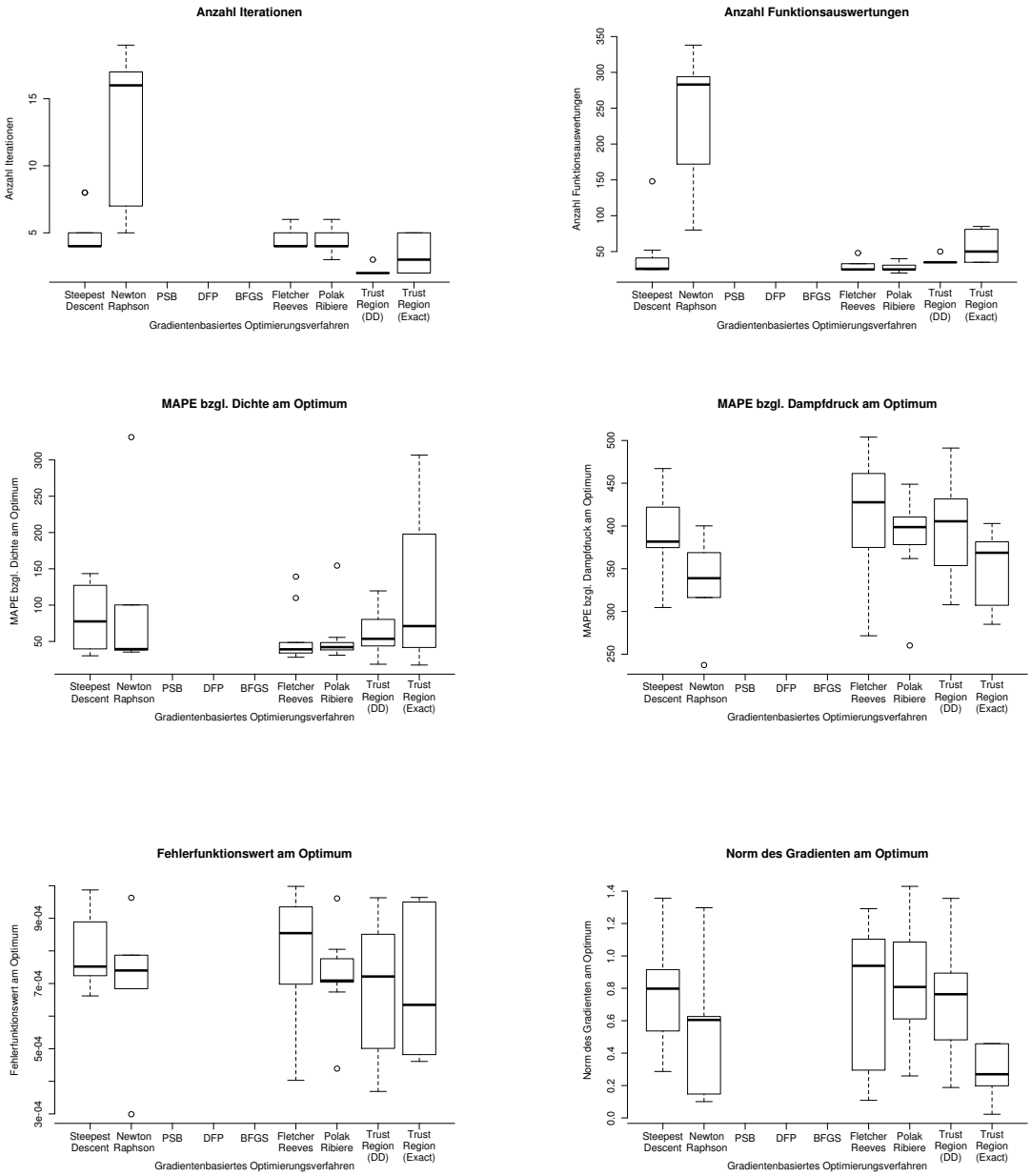


Abbildung 4.2: Box-Plots entsprechend den Spalten von Tabelle 4.3: Das robusteste und schnellste Verfahren ist in diesem Fall das Polak-Ribière-Verfahren. Allerdings konnten sich auch Fletcher-Reeves, die Trust-Region-Methode mit exakter Lösung des Teilproblems und die Methode des steilsten Abstiegs als robust erweisen. Die detaillierte Verfahrensbewertung hat ergeben, daß die Newton-Raphson-Methode und das Trust-Region-Verfahren zusammen mit dem doppeltem Hundebein im Falle von Rauschen nicht geeignet waren. Somit sind die vorliegenden Ergebnisse als Zufallsergebnisse zu werten. Zudem ist die Streuung der Resultate sehr hoch.

sich im Falle von Rauschen als eher ungeeignet herausgestellt. Die Anzahl an benötigten Iterationen und Funktionsauswertungen war teilweise geringer als im unverrauschten Fall. Dies hängt damit zusammen, daß hierbei für die Diskretisierung von Gradient und Hesse-Matrix $h = 10^{-2}$ anstelle von $h = 10^{-5}$ gewählt wurde. Somit wurde lediglich der Trend von F reproduziert und kleinen lokalen Änderungen keine Bedeutung beigemessen. Mögliche Zerklüftungen wurden übersprungen. Außerdem war der Schwellenwert des Abbruchkriteriums größer, und die Armijo-Schrittweitensteuerung neigt dazu, Fehlerfunktionswerte zu akzeptieren, zu denen negative Zufallszahlen addiert wurden. Daher kann innerhalb nur einer Iteration eine viel deutlichere Verbesserung erzielt werden als im unverrauschten Fall, was dann allerdings nur auf das Rauschen zurückzuführen ist. Die erfolgreichen Verfahren konvergierten wie erwartet gegen das gleiche Minimum. Die finalen Kraftfeldparameter unterschieden sich nur geringfügig. Auch im Falle von Rauschen wurden die Fehlerfunktion und die Norm des Gradienten jeweils um eine Größenordnung verringert. Die Norm des Gradienten war um eine Größenordnung höher als im unverrauschten Fall, was darauf zurückzuführen ist, daß der Schwellenwert für das Abbruchkriterium größer war. Dennoch konnte durch die Approximation des Gradienten der Trend der Fehlerfunktion korrekt reproduziert werden. Auf die Hesse-Matrix wirken sich statistische Unsicherheiten, wie bereits in Abschnitt 3.5.6 motiviert und getestet, enorm aus. Dies führt dazu, daß Verfahren wie Newton-Raphson, Quasi-Newton und die Trust-Region-Methode zusammen mit dem doppelten Hundebein bei Rauschen nicht geeignet sind. Lediglich das Trust-Region-Verfahren mit exakter Lösung des Teilproblems kann mit der extrem verrauschten Hesse-Matrix umgehen (siehe Bemerkung 3.5.5).
Die Box-Plots in Abbildung 4.2 zeigen deutlich, daß das Polak-Ribière-Verfahren bei Optimierungsaufgabe 8 am robustesten in Bezug auf Rauschen ist und die besten Ergebnisse liefert. Aber auch das Fletcher-Reeves-Verfahren und die Methode des steilsten Abstiegs sind robust und somit geeignete Verfahren. Das Trust-Region-Verfahren mit exakter Lösung des Teilproblems ist zumindest in Bezug auf die Anzahl an Iterationen und Funktionsauswertungen robust. Bei der detaillierten Verfahrensbewertung hat es sich jedoch als geeignet herausgestellt, vor allem weil es oft sehr nahe an das Minimum gelangt. Die Ergebnisse der Methode des doppelten Hundebeins sind, wie bereits diskutiert, als Zufallsergebnisse zu werten. Weitere Details sind in Hülsmann u. a. (2010b) nachzulesen.

Im folgenden wird die Anwendbarkeit der einzelnen gradientenbasierten Verfahren diskutiert und bewertet. Es ist zu beachten, daß für eine erfolgreiche Armijo-Schrittweitensteuerung eine geeignete Abstiegsrichtung gefunden werden muß, da die Armijo-Schrittweite ansonsten sehr schnell so klein wird, daß keine signifikanten Verbesserungen mehr erzielbar sind. Falls eine Gradientenrichtung falsch ist, beispielsweise aufgrund des Rauschens oder aus numerischen Gründen, oder die Komponenten einer Hesse-Matrix zu verrauscht sind, so kann die zugehörige Abstiegsrichtung für die Armijo-Schrittweitensteuerung ungeeignet sein.

- **Methode des steilsten Abstiegs:** Dieses Verfahren hat sich insbesondere in den ersten Optimierungsschritten, das heißt in Bereichen, wo die Fehlerfunktion steil ist, als geeignet und effizient erwiesen. Auch im Falle von Rauschen konnte durch den Gradienten ein korrekter Trend ermittelt werden. Die Methode des steilsten Abstiegs hat zwar eine geringe Konvergenzgeschwindigkeit und gelangt nicht so nah an das Minimum wie andere

Verfahren, ist aber für die vorliegende Problemstellung auf jeden Fall geeignet.

- **Newton-Raphson-Verfahren:** Im unverrauschten Fall ist dieses Verfahren zwar als geeignet einzustufen, da es stets zum Abbruchkriterium gelangte, allerdings wurde oftmals anstelle der Newton-Richtung die Richtung des steilsten Abstiegs verwendet, so daß stets die Gefahr besteht, daß die Hesse-Matrix unnötig berechnet wird. Die Methode benötigt zwar nicht die Voraussetzung der positiven Definitheit der Hesse-Matrix, da die Newton-Richtung gemäß Gleichung (3.23) verworfen oder akzeptiert wird. Allerdings ist es nicht gewährleistet, daß die Abstiegsrichtung für die Armijo-Schrittweitensteuerung geeignet ist, was insbesondere im Falle von Rauschen dazu geführt hat, daß sehr viele Replikate nicht erfolgreich waren. In Hülsmann u. a. (2010b) wurde das Newton-Raphson-Verfahren zwar als geeignet eingestuft, da es im unverrauschten Fall geeignet war und im verrauschten Fall zumindest manchmal zum Ziel führte. Allerdings war es nie robust. Da in dieser Arbeit jedoch der Fokus auf der Effizienz liegt, gerade im Falle von Rauschen, und die Hesse-Matrix teilweise unnötig berechnet wird, wird es hier als ungeeignet bewertet. Lediglich als letzter Schritt des Trust-Region-Verfahrens mit exakter Lösung des Teilproblems, das heißt, falls das Minimum des Modells innerhalb des Vertrauensbereichs liegt und die Hesse-Matrix dort positiv definit ist, könnte es unter Umständen sinnvoll sein. Dies wird jedoch noch im einzelnen zu prüfen sein (siehe Abschnitt 4.5).

- **Quasi-Newton-Verfahren:** Diese Verfahren sind für die vorliegende Problemstellung nicht geeignet, auch nicht im unverrauschten Fall. Um Quasi-Newton-Verfahren mit einer effizienten Schrittweitensteuerung zu kombinieren, ist die positive Definitheit von H_0 erforderlich, was jedoch nie der Fall war. Ist H_k positiv definit, dann ist auch H_{k+1} positiv definit. Auf diesen Satz sind alle Quasi-Newton-Verfahren angewiesen. Selbst wenn H_0 positiv definit wäre, könnte immer noch nicht garantiert werden, daß die zugehörige Abstiegsrichtung geeignet ist, denn H_k ist nur eine Approximation für $D^2F(x^{(k)})$. Im Falle von Rauschen liefert das Verfahren nur noch völlig falsche Berechnungen.

- **Konjugierte Gradienten:** Sowohl das Fletcher-Reeves- als auch das Polak-Ribière-Verfahren sind geeignete Methoden, wobei sich letzteres als robuster herausgestellt hat. Die CG-Verfahren verwenden zwei aufeinanderfolgende Gradienten, haben somit mehr Information über den Verlauf von F und sind somit auch dazu in der Lage, näher an das Minimum zu gelangen als die Methode des steilsten Abstiegs, was am Ende dieses Abschnitts beispielhaft dargestellt wird.

- **Trust-Region-Verfahren mit doppeltem Hundebein:** Dieses Verfahren ist aus demselben Grund wie bei den Quasi-Newton-Verfahren nicht geeignet: Da die Hesse-Matrix nicht spd war, wurde weder die Newton-Richtung noch die Richtung des doppelten Hundebeins akzeptiert. Der Cauchy-Punkt war stets Lösung des Teilproblems, so daß das Verfahren der Methode des steilsten Abstiegs gleichkam. Dann führt aber eine ungeeignete Wahl von Δ dazu, daß kein signifikant kleinerer Fehlerfunktionswert gefunden werden konnte. Im Falle von Rauschen wurde dieser aufgrund der Zufallszahlen meistens gefunden, allerdings war dadurch das Verfahren keinesfalls robust. Selbst wenn die Newton-Richtung akzeptiert würde, würden sowohl das Problem des ungeeigneten Δ als auch die Schwächen des Newton-Raphson-Verfahrens im Falle von Rauschen auftreten. Nur bei der Akzeptanz der Richtung des doppelten Hundebeins würde das Verfahren erfolgreicher sein, da F in dieser Richtung noch weiter absteigt. Da dies aber nur der Fall sein kann,

Alg.	# It.	# Eval.	$\rho_l(x^\tau)$	$p_\sigma(x^\tau)$	$F(x^\tau)$	# It.	# Eval.	$\rho_l(x^{\mathrm{opt}})$	$p_\sigma(x^{\mathrm{opt}})$	$F(x^{\mathrm{opt}})$	sign.
					Optimierungsaufgabe 7						
FR	6	44	0.57%	2.90%	0.000433	20	123	0.45%	2.62%	0.000358	ja
PR	5	38	0.87%	2.84%	0.000436	19	137	0.30%	2.50%	0.000306	ja
TR	12	185	1.10%	2.71%	0.000416	65	1001	0.59%	2.56%	0.000336	ja
					Optimierungsaufgabe 8						
FR	4–5	29–30	0.56%	4.14%	0.000795	12	> 60	0.26%	2.99%	0.000287	ja
PR	4–5	28–29	0.54%	3.86%	0.000721	4	> 20	0.57%	3.05%	0.000431	nein
TR	3–4	56	1.18%	3.52%	0.000687	7	> 100	0.30%	3.86%	0.000346	ja

Tabelle 4.4: Lokale Verfeinerung im Falle der Konjugierten Gradienten (Fletcher-Reeves (FR) und Polak-Ribière (PR)) und des Trust-Region-Verfahrens mit exakter Lösung des Teilproblems (TR) für die Optimierungsaufgaben 7 und 8. Zunächst sind für jedes Verfahren die Ergebnisse aus den Tabellen 4.2 und 4.3 angegeben und dann im Vergleich dazu die Ergebnisse der lokalen Verfeinerungen. In der letzten Spalte ist angegeben, ob eine signifikante Verbesserung erzielt wurde oder nicht. Im unverrauschten Fall ist dabei jede kleinste Verbesserung signifikant. Mit x^τ wurde die erste Iteration bezeichnet, für die die Abbruchbedingung $F(x) \leq \tau$ erfüllt war und mit x^{opt} die nach den lokalen Verfeinerungen erhaltene (optimale) Iteration. Im unverrauschten Fall haben alle drei Verfahren zu signifikanten Verbesserungen geführt, im verrauschten Fall alle bis auf das Polak-Ribière-Verfahren. Es wurde stets nur ein Replikat durchgeführt. Daher ist keine genaue Anzahl an Funktionsauswertungen angegeben, da die Anzahl an Armijo-Schritten stets variieren kann. Somit reicht es zu sagen, daß sich die Anzahl an Funktionsauswertungen in etwa verdoppelte. Bei den Konjugierten Gradienten wurden für x^{opt} einige Replikate durchgeführt, um die Signifikanz zu prüfen. Da bei beiden Verfahren alle Abweichungen von ρ_l und p_σ in den oben angegebenen Bereichen lagen, konnte geschlossen werden, daß das Fletcher-Reeves-Verfahren zu einer Verbesserung geführt hat und das Polak-Ribière-Verfahren nicht. Im Falle des Trust-Region-Verfahrens waren mehrere Replikate nicht notwendig, da die Verbesserung in der Siededichte signifikant war.

wenn die Hesse-Matrix spd ist, wurde diese Richtung nie akzeptiert und die Hesse-Matrix stets unnötig berechnet.

- **Trust-Region-Verfahren mit exakter Lösung:** Dieses Verfahren war zwar nicht so robust wie die CG-Verfahren, allerdings konvergierte es meistens innerhalb weniger Iterationen bei den erfolgreichen Replikaten. Da stets eine positiv definite Näherung für die Hesse-Matrix berechnet wird, ist dieses Verfahren auch im Falle von extrem verrauschten Hesse-Matrizen geeignet, da die durch das approximative Modell berechnete Abstiegsrichtung meistens den Trend der Fehlerfunktion reproduzieren konnte. Die Methode gelangte oftmals näher an das Minimum als die Methode des steilsten Abstiegs, was am Ende des Abschnitts beispielhaft aufgeführt ist. Da die Berechnung einer Hesse-Matrix sehr aufwendig ist, ist es daher insbesondere in der Nähe des Minimums anzuwenden. Dies wird in Abschnitt 4.5 eingehender studiert.

Ein weiterer wichtiger Aspekt ist die Frage, wie nahe an das Minimum ein Verfahren gelangen kann, das heißt ohne daß aufgrund des Regenrinnenproblems oder des Rauschens die Armijo-Schrittweitensteuerung mehr als 100 Iterationen benötigt. Im Falle von Rauschen ist dabei zu evaluieren, ob die erhaltenen Verbesserungen tatsächlich signifikant sind, das heißt nicht im Bereich des Rauschens liegen. Da das Abbruchkriterium stets so gewählt worden war, daß die Methode des steilsten Abstiegs keine Verbesserungen mehr liefern konnte, kommen nur noch die Konjugierten Gradienten und das Trust-Region-Verfahren mit exakter Lösung des Teilproblems für derartige lokale Verfeinerungen in Frage.

Tabelle 4.4 zeigt die Ergebnisse der lokalen Verfeinerungen an. Die Vorgehensweise und die Bewertung sind in der Legende beschrieben. Im unverrauschten Fall führten alle Verfahren zu signifikanten Verbesserungen, im verrauschten Fall das Fletcher-Reeves- und das Trust-Region-Verfahren bei etwa doppelt so vielen Funktionsauswertungen. Es wurde stets nur ein Replikat durchgeführt, da in diesem Abschnitt nur gezeigt werden soll, daß derartige Verfeinerungen mithilfe der drei Verfahren grundsätzlich möglich sind. Die Ergebnisse einer detaillierteren Analyse werden in Abschnitt 4.5.2 angegeben.
Ein als optimal anzusehendes Kraftfeld, welches mit vier Replikaten evaluiert wurde, konnte vom Fletcher-Reeves-Verfahren erzielt werden: Nach zwölf Iterationen und mehr als 60 Funktionsauswertungen waren die optimalen Kraftfeldparameter

$$\varepsilon = 0.30490 \text{ kJ/mol}, \ \sigma = 0.32873 \text{ nm}, \ Q^2 = 0.01791 \text{ (Dnm)}^2 \text{ und } L = 0.11271 \text{ nm}.$$

Außer bei $T = 65$ K, wo die Abweichung des Dampfdrucks bei über 7% lag, wichen alle Siededichten um weniger als 0.5% und alle Dampfdrücke um weniger als 3% vom Experiment ab.

4.3 Reduktion der Anzahl an Simulationen mittels effizienter Gradienten- und Hesse-Matrix-Berechnung

Wie den Tabellen 4.2 und 4.3 zu entnehmen ist, erreichen die für die vorliegende Problemstellung geeigneten Verfahren das Abbruchkriterium oftmals zwar in weniger als zehn Iterationen, allerdings werden zumeist immer noch mehr als 50 Funktionsevaluationen benötigt. Die Effizienz der gradientenbasierten Verfahren hat sich somit in Bezug auf die schnelle Konvergenz bestätigt, allerdings sollte versucht werden, den Gradienten beziehungsweise die Hesse-Matrix möglichst ohne weitere Simulationen zu berechnen. Daher werden in diesem Abschnitt die Methoden aus den Abschnitten 3.6.2 bis 3.6.4 verwendet, um die Effizienz noch weiter zu erhöhen. Abschnitt 4.3.1 befaßt sich dabei mit der Verwendung bereits durchgeführter Simulationen, das heißt mit den in den Abschnitten 3.6.2 und 3.6.3 vorgestellten Methodiken, und Abschnitt 4.3.2 mit der Verwendung reduzierter Parameter, das heißt mit den in Abschnitt 3.6.4 sowie den Anhängen H.1 und H.2 vorgestellten Methodiken.

4.3.1 Verwendung bereits durchgeführter Simulationen

Ein Gradient kann nur dann mithilfe des in Abschnitt 3.6.2 beschriebenen Algorithmus effizient, das heißt durch Verwendung bereits durchgeführter Simulationen, berechnet weden, wenn sowohl die Normbedingung (3.98) als auch die Determinantenbedingungen (3.99) beziehungsweise das Konditionszahlkriterium (3.100) erfüllt sind. Es sei $x^\star$ diejenige Iteration, bei der zuletzt der Gradient auf klassische Weise, das heißt mit einer Simulation pro Dimension, ermittelt wurde. Je größer der Index k wird, desto weiter entfernt sich $x^{(k)}$ von $x^\star$, was ab einem gewissen Punkt zur Verletzung der Normbedingung führt. Da die $x^\star + he_i, \ i = 1, ..., N$, in einer kleinen Umgebung von $x^\star$ liegen, werden die Differenzvektoren $x^{(k)} - (x^\star + he_i), \ i = 1, ..., N$, irgendwann linear abhängig sein, was wiederum ab einem gewissen Punkt zur Verletzung der Determinantenbedingung beziehungsweise des Konditionszahlkriteriums (3.100) führt. Es wird

daher keinesfalls möglich sein, daß für alle $k > 1$ der Gradient an der Stelle $x^{(k)}$ effizient berechnet wird.
Ein weiteres Problem ist die Wahl von h. Aufgrund von Gleichung (3.98) kann durch die effiziente Gradientenberechnung nur eine Verbesserung in $\mathcal{O}(h)$ in jeder Dimension erzielt werden. Im unverrauschten Fall, wo $h = 10^{-5}$ gewählt wurde, ist dieses Verfahren daher uninteressant. Der Gradient wurde bei den Optimierungsaufgaben 1, 3, 5 und 7 niemals effizient berechnet. Die Methode ist somit nur im Falle von Rauschen einsetzbar. Dabei wurde stets $h = 10^{-2}$ gewählt. Es ist allerdings zu beachten, daß das Rauschen auf der Fehlerfunktion zu fehlerhaften Gradientenrichtungen führen kann. Da die effiziente Gradientenberechnung eine zusätzliche Approximation darstellt, kann der Fehler unter Umständen noch größer sein. Bei der effizienten Hesse-Berechnung kann sich dies noch extremer auswirken. Ob sich diese Verfahren tatsächlich lohnen, ist daher anhand der Korrelationsfunktionen zu prüfen. Falls keine zusätzlichen Simulationen für Gradienten und Hesse-Matrix mehr notwendig sind, dafür aber aufgrund von falschem Gradient oder falscher Hesse-Matrix die Armijo-Schrittweitensteuerung viel mehr Iterationen benötigt, sind die Methodiken unbrauchbar.

Da hierbei alle Optimierungsaufgaben in Betracht gezogen werden, wird wieder $\forall_{i,T}\ w_{i,t} = 1$ gesetzt, um die Anzahl an benötigten Funktionsauswertungen direkt mit den Ergebnissen aus Hülsmann u. a. (2010b) vergleichen zu können. Tabelle 4.5 zeigt die Anzahl an erfolgreichen Replikaten, Iterationen und Funktionsauswertungen sowie die Anzahl an Funktionsauswertungen pro Iteration für diejenigen Optimierungsverfahren, die sich innerhalb der Bewertung als geeignet erwiesen haben. Außerdem ist jeweils der finale Parametervektor, hier wieder mit x^{opt} bezeichnet, angegeben, um sicherzustellen, daß das jeweilige Verfahren in die gleiche Richtung konvergiert ist. Somit konnte die Berechnung eines falschen Gradienten auch praktisch ausgeschlossen werden. Die Werte der Fehlerfunktion sowie die relativen Fehler der Zielgrößen in Bezug auf die experimentellen Daten sind nicht angegeben, da sich diese sowieso stets ähnlich sind, wenn das Abbruchkriterium erreicht wird. Dargestellt sind erneut die Durchschnittsergebnisse über zehn statistisch unabhängige Zufallsreplikate im Falle der Optimierungsaufgaben 2, 4, 6 und 8, wobei bei Optimierungsaufgabe 6 kein Replikat erfolgreich war. Es ist zu beachten, daß in Hülsmann u. a. (2010b) die Verwendung einer heuristischen Schrittweitensteuerung notwendig war, um überhaupt geeignete Startparameter zu erhalten. Die anschließende Optimierung erwies sich ebenfalls als schwierig, was auf die derartige Wahl der Startparameter zurückzuführen ist. Mit besseren Startparametern wären auch bei dieser Aufgabe bessere Ergebnisse erzielt worden. In allen anderen Fällen sank die Anzahl an Funktionsauswertungen oftmals um den Faktor 2, teilweise sogar um den Faktor 4. In manchen Fällen konnte sogar die Anzahl an Iterationen reduziert werden. Aufgrund der höheren Ungenauigkeit im effizienten Fall wurde die Richtung des Gradienten zwar zumeist gut reproduziert, allerdings war die Norm des approximierten Gradienten zumeist etwas größer. Gelangt das Verfahren im klassischen Fall mit einer Abstiegsrichtung d und einer Schrittweite β an den Rand des zulässigen Gebiets und im effizienten Fall mit einer Abstiegsrichtung $\tilde{d}$ und einer Schrittweite $\tilde{\beta}$, so gilt in den meisten Fällen $||d|| < ||\tilde{d}||$ und somit $\beta > \tilde{\beta}$. Falls die Gradientenrichtung korrekt reproduziert wird, so gilt $\beta d = \tilde{\beta}\tilde{d}$ und somit $\forall_\ell\ \beta^\ell ||d|| > \tilde{\beta}^\ell ||\tilde{d}||$. Somit ist im allgemeinen bei der effizienten Gradientenberechnung eine kleinere Anzahl $\bar{\ell}$ an Armijo-Schritten erforderlich, um einen kleineren Fehlerfunktionswert zu finden. Dies kann im Endeffekt oft dazu führen, daß $\tilde{\beta}^{\bar{\ell}} ||\tilde{d}|| < \beta^{\tilde{\ell}} ||d||$,

Alg.	# Repl.	# It.	# Eval.	# Eval./It.	x^{opt}	# Repl.	# It.	# Eval.	# Eval./It.	x^{opt}
					Optimierungsaufgabe 2					
SA	8	7–8	63–64	8–9	0.3055 0.3314 0.0165 0.1102	6	4	16–17	4–5	0.3059 0.3305 0.0185 0.1097
FR	3	10–11	191–192	18–19	0.3067 0.3317 0.0170 0.1086	6	10–11	77-78	7–8	0.3066 0.3314 0.0170 0.1092
PR	8	6–7	52	8	0.3056 0.3314 0.0166 0.1104	6	9–10	49	5–6	0.3054 0.3310 0.0163 0.1100
					Optimierungsaufgabe 4					
SA	5	9–10	84	8–9	0.3039 0.3291 0.0166 0.1149	8	7–8	17–18	2–3	0.3027 0.3288 0.0158 0.1131
FR	8	8–9	61–62	7–8	0.3039 0.3277 0.0165 0.1158	9	9–10	47–48	5–6	0.3030 0.3293 0.0158 0.1140
PR	7	7–8	81–82	10–11	0.3040 0.3289 0.0166 0.1148	10	7–8	17–18	2–3	0.3028 0.3287 0.0160 0.1130
					Optimierungsaufgabe 6					
					Eine effiziente Gradientenberechnung führte bei dieser Anwendung zu keinem Erfolg.					
					Optimierungsaufgabe 8					
SA	9	6	46–47	7–8	0.3056 0.3285 0.0176 0.1120	9	5	20–21	4–5	0.3054 0.3286 0.0172 0.1110
FR	10	6–7	82	12–13	0.3055 0.3285 0.0174 0.1122	10	6–7	27–28	4–5	0.3051 0.3288 0.0170 0.1114
PR	10	5–6	55–66	10–11	0.3056 0.3284 0.0176 0.1120	9	5–6	18	3–4	0.3054 0.3283 0.0171 0.1109

Tabelle 4.5: Ergebnisse der effizienten Gradientenberechnung für die Methode des Steilsten Abstiegs (SA) und die Konjugierten Gradienten (FR und PR) im Falle der Optimierungsaufgaben 2, 4, 6 und 8. Angegeben sind Durchschnittswerte über zehn statistisch unabhängige Zufallsreplikate. Optimierungsaufgabe 6 erwies sich für die Bewertung der effizienten Gradientenberechnung als ungeeignet. Sonst konnte die durchschnittliche Anzahl an Funktionsauswertungen, insbesondere bei mehreren Temperaturen (Optimierungsaufgaben 4 und 8), meist um den Faktor 2-4 reduziert werden. Es wurde von Beginn des Optimierungsprozesses an effizient berechnet, weswegen in jedem der angegebenen Optimierungsabläufe der Gradient höchstens zweimal klassisch und ansonsten ausschließlich effizient berechnet wurde.

Alg.	# Repl.	# It.	# Eval.	# Eval./It.	x^{opt}	# Repl.	# It.	# Eval.	# Eval./It.	x^{opt}
					Optimierungsaufgabe 2					
					Es wurde erst dann versucht, die Hesse-Matrix effizient zu berechnen, als die physikalischen Eigenschaften bereits bis auf das jeweilige Rauschen optimal bestimmt wurden.					
					Optimierungsaufgabe 4					
					Nur 3 Replikate erfolgreich, ansonsten zulässigen Bereich verlassen. Optimierungsaufgabe für Bewertung ungeeignet.					
					Optimierungsaufgabe 6					
					Die Hesse-Matrix wurde nie effizient berechnet.					
					Optimierungsaufgabe 8					
TR	8	7	147	21	0.3026 0.3282 0.0172 0.1101	10	5–6	96–97	17–18	0.3022 0.3273 0.0176 0.1100

Tabelle 4.6: Ergebnisse der effizienten Hesse-Matrix-Berechnung für das Trust-Region-Verfahren mit exakter Lösung des Teilproblems (TR) im Falle der Optimierungsaufgaben 2, 4, 6 und 8. Nur Optimierungsaufgabe 8 konnte die dadurch ermöglichte Reduktion des Rechenaufwands aufzeigen. Angegeben sind hierbei wieder Durchschnittswerte über zehn statistisch unabhängige Zufallsreplikate. Auch bei allen anderen Optimierungsaufgaben wurden zehn Replikate durchgeführt. Es konnte eine signifikante Reduktion der Anzahl an Funktionsauswertungen beobachtet werden.

wobei $\tilde{\ell} > \bar{\ell}$ die Anzahl an Armijo-Schrittweiten im klassischen Fall gewesen wäre. Somit ist es möglich, daß das jeweilige Optimierungsverfahren im effizienten Fall größere Schritte macht, was zu weniger Iterationen bis zum Erreichen des Abbruchkriteriums führt.

Im Falle der Verfahren, die eine Hesse-Matrix benötigen, hat sich gemäß Abschnitt 4.2 lediglich das Trust-Region-Verfahren mit exakter Lösung des Teilproblems als geeignet herausgestellt. Da das in Abschnitt 3.6.3 vorgestellte Verfahren zur effizienten Berechnung der Hesse-Matrix aufgrund der strikteren Akzeptanzbedingung (3.101) stets nur in der Nähe des Minimums verwendet wurde und aufgrund der Argumentationen in Abschnitt 3.6.6, sollte das Trust-Region-Verfahren auch nur dort in Betracht gezogen werden. Tabelle 4.6 zeigt die Ergebnisse der effizienten im Vergleich zur klassischen Hesse-Matrix-Berechnung analog zu Tabelle 4.5 für alle vier Optimierungsaufgaben: Eine effiziente Berechnung wurde nur bei mehreren Temperaturen durchgeführt, was darauf zurückzuführen ist, daß die Fehlerfunktion bei einer Temperatur flacher ist und somit aufgrund des Rauschens benachbarte Funktionswerte schwieriger zu unterscheiden sind. Bei Optimierungsaufgabe 2 wurde erst in einem Bereich versucht, die Hesse-Matrix effizient zu berechnen, in dem die Fehlerfunktion bereits optimal war. Alle physikalischen Zielgrößen stimmten bis auf statistische Unsicherheiten mit ihren experimentellen Werten überein. Insofern machte die in Abschnitt 3.6.3 dargestellte Methodik keinen Sinn mehr. Optimierungsaufgabe 4 erwies sich als ungeeignet für die Bewertung der effizienten Berechnungsweise. In Hülsmann u. a. (2010b) waren nur drei Replikate erfolgreich. Das

Trust-Region-Verfahren konvergierte in die Nähe des Randes des zulässigen Gebietes. Zur klassischen Hesse-Matrix-Berechnung war eine Veränderung der Kraftfeldparameter im Falle der Diagonalen um $h = 2 \cdot 10^{-2}$ erforderlich, was dazu führte, daß das zulässige Gebiet verlassen wurde und somit die entsprechenden Korrelationsfunktionen nicht mehr ausgewertet werden konnten. Zur Berechnung der Verdampfungsenthalpie wurde für höhere Temperaturen die Clausius-Clapeyron-Gleichung (3.65) und für niedrigere Temperaturen die ideale Gasgleichung verwendet. Dies führte zu einem Artefakt in der temperaturabhängigen Verdampfungsenthalpiekurve, was in Abbildung 4.1 deutlich zu erkennen ist. Dies hatte wiederum Auswirkungen auf die Differenzierbarkeit der Fehlerfunktion, was bei Optimierungsaufgabe 4 zu einer zusätzlichen Schwierigkeit führte. Für die allgemeine Verfahrensbewertung war diese Aufgabe daher nur bedingt, für die Effizienzbewertung gar nicht geeignet. Als weiterer Versuch wurde $h = 10^{-3}$ gewählt, so daß das zulässige Gebiet nicht verlassen werden konnte. Um dieselbe Genauigkeit wie vorher zu erhalten, wurde bei der Normbedingung (3.101) $C = 10$ gewählt. Allerdings war die Genauigkeit zu hoch, also $h = 10^{-3}$ zu klein, wann immer die Hesse-Matrix klassisch berechnet wurde (vergleiche Abschnitt 3.5.6), was dazu führte, daß das Abbruchkriterium nicht mehr erreicht werden konnte. Im Falle von Optimierungsaufgabe 6 wurde genau wie bei der effizienten Gradientenberechnung die Hesse-Matrix nie effizient berechnet. Erst anhand von Optimierungsaufgabe 8 konnte gezeigt werden, daß eine Beschleunigung möglich ist: Die Anzahl an Funktionsauswertungen sank deutlich, die Anzahl an Iterationen wurde im Durchschnitt ebenfalls reduziert, und die Anzahl an erfolgreichen Replikaten stieg auf 100% an. Dies hat dieselben Ursachen wie bei der Gradientenberechnung.

Die Bewertung hat praktisch bewiesen, daß eine effiziente Gradientenberechnung im allgemeinen zu einer sehr großen Reduktion des Rechenaufwands führt, ohne den Verlauf der Optimierungsverfahren zu verändern. Bei der Hesse-Matrix war die Verbesserung nicht so deutlich, es kann jedoch geschlossen werden, daß im Falle des Trust-Region-Verfahrens mit exakter Lösung des Teilproblems eine effiziente Hesse-Matrix-Berechnung möglich und somit lohnenswert ist.

Es ist allerdings auszuschließen, daß der Gradient in eine völlig andere Richtung zeigt als der auf klassische Weise berechnete. Auch die Einträge der Hesse-Matrix sollten sich tendentiell nicht allzu sehr von denen der klassisch ermittelten Hesse-Matrix unterscheiden. Daher wurden innerhalb einer Optimierung mit der Methode des steilsten Abstiegs zwei Gradienten und mit dem Trust-Region-Verfahren zwei Hesse-Matrizen miteinander verglichen. Nachdem der Gradient effizient berechnet war, wurde dieser herausgeschrieben und danach an der aktuellen Iteration der Gradient auf klassische Weise ermittelt. Der klassische Gradient lautete

$$(\nabla F)_{\text{klass}} = (19.8999, 5.9671, 13.9760, -33.5199)^T, \ ||(\nabla F)_{\text{klass}}|| = 41.84,$$

und der effizient berechnete Gradient

$$(\nabla F)_{\text{eff}} = (25.1876, 6.7406, 17.9736, -30.3034)^T, \ ||(\nabla F)_{\text{eff}}|| = 43.83.$$

Die Richtungen der beiden Gradienten sind ähnlich, was in beiden Fällen zu einer Abstiegsrichtung führte. Daß die Norm im effizienten Fall meist etwas größer ist, wurde oben bereits

motiviert. Es ist zu beachten, daß beide Gradienten im Falle von Rauschen nur grobe Approximationen sind. Weiterhin ist zu vermerken, daß in der Normbedingung (3.98) $C = 5$ gewählt wurde. Nur der Optimierungsablauf selbst konnte zeigen, daß beide Gradienten den Trend der Fehlerfunktion approximativ reproduzieren konnten.
Das gleiche gilt für die Hesse-Matrix: Eine klassisch ermittelte Hesse-Matrix lautete

$$(D^2F)_{\text{klass}} = \begin{pmatrix} 2968 & 2068 & 2325 & -9034 \\ 2068 & 2204 & 637 & -6083 \\ 2325 & 638 & 3692 & -6569 \\ -9034 & -6084 & -6569 & 21940 \end{pmatrix}, \; ||(D^2F)_{\text{klass}}||_F = 29196,$$

und eine effizient berechnete Hesse-Matrix

$$(D^2F)_{\text{eff}} = \begin{pmatrix} 6645 & 4360 & 4519 & -959 \\ 5923 & 5058 & 3429 & 2556 \\ 5957 & 3304 & 5864 & 1686 \\ -3503 & -1548 & -2296 & 33869 \end{pmatrix}, \; ||(D^2F)_{\text{eff}}||_F = 37605.$$

Letztere ist nicht symmetrisch, stimmt aber bis auf die letzte Spalte zumindest tendentiell mit der klassischen überein. Auch hier konnte ein Optimierungsablauf mit dem Trust-Region-Verfahren mit exakter Lösung des Teilproblems zeigen, daß beide Hesse-Matrizen die Krümmung der Fehlerfunktion approximativ reproduzieren konnten. Die nachfolgende Iteration und deren Funktionswert unterschieden sich stets, das heißt auch nach einer effizienten Gradienten- oder Hesse-Berechnung, signifikant von der vorherigen und stellte stets eine Verbesserung dar.

In der Nähe des Minimums, das heißt unterhalb der Abbruchkriterien, mußten die Normbedingungen (3.98) beziehungsweise (3.101) strenger gewählt werden, da dort eine genauere Approximation des Gradienten und der Hesse-Matrix vonnöten war. Wie bereits in Abschnitt 3.5.5 dargestellt, können weder Steigung noch Krümmung ansonsten akkurat wiedergegeben werden. Dies führte jedoch dazu, daß die Normkriterien in der Nähe des Minimums nie erfüllt waren und somit weder Gradient noch Hesse-Matrix dort effizient berechnet wurden. Eine effiziente Berechnung ist also lediglich zu Beginn des Optimierungsprozesses und nicht in der Nähe des Minimums geeignet. Damit wird das Trust-Region-Verfahren aufgrund des zu hohen Rechenaufwands in der Nähe des Minimums nicht empfehlenswert und ist somit geeignet zu ersetzen, was in den Abschnitten 4.5.2 und 5.2.2 eingehend diskutiert wird.

Ob und inwieweit die hier verwendeten Verfahren zur Effizienzerhöhung gradientenbasierter Verfahren auf Molekulare Simulationen angewandt werden können, wird in Abschnitt 5.1.3 am Beispiel von Phosgen diskutiert.

4.3.2 Verwendung reduzierter Parameter

Die in Abschnitt 3.6.4 sowie den Anhängen H.1 und H.2 vorgestellten Methoden zur effizienten Gradienten- beziehungsweise Hesse-Matrix-Berechnung sind ein Beispiel dafür, daß theoretisch

fundierte Vorgehensweisen in der Praxis nicht notwendigerweise brauchbar sind. Die Veränderungen der Kraftfeldparameter um h bewirken eine zu große Änderung des molekularen Modells. Somit ist der Ansatz (H.4) praktisch nicht anwendbar, jedenfalls nicht ohne die Möglichkeit, die gewünschten physikalischen Zielgrößen für sämtliche molekulare Modellparameter zurückzurechnen. Da bislang allerdings keine funktionalen Abhängigkeiten der Zielgrößen von beispielsweise Ladung oder Kraftkonstanten bekannt ist, ist dies leider nicht umzusetzen. Die Methode wäre nur dann sinnvoll, wenn h so klein gewählt wird, daß die Änderungen nur geringfügige Auswirkungen auf das molekulare Modell haben, und gleichzeitig so groß ist, daß aufgrund des statistischen Rauschens die entsprechenden Funktionswerte unterscheidbar sind. Die Existenz eines derartigen optimalen h kann weder theoretisch noch praktisch bewiesen werden. Selbst wenn es existiert, kann es lediglich durch eine erhebliche Erhöhung des Rechenaufwands gefunden werden, denn es wären in jedem Fall weitere Simulationen erforderlich. Weiterhin wirkt sich das Rauschen zum Beispiel aufgrund von Gleichung (3.102) hierbei stärker aus als gewöhnlich.

Daß die effiziente Gradientenberechnung in der Praxis nicht anwendbar ist, belegt der folgende Vergleich: Bei der Anpassung von sechs Kraftfeldparametern für Phosgen an Siededichte und Verdampfungsenthalpie zu sieben verschiedenen Temperaturen (siehe Abschnitt 5.1.3) ergab sich für den initialen Parametervektor der folgende klassisch berechnete Gradient:

$$(\nabla F)_{\text{klass}} = (-1.3596, 0.3525, 11.2082, -0.2961, -0.1368, -0.5332)^T,\ ||(\nabla F)_{\text{klass}}|| = 11.31,$$

Im Vergleich dazu ergab sich mithilfe der Methoden aus den Anhängen H.1 und H.2 die folgende Approximation:

$$(\nabla F)_{\text{eff}} = (2810.0660, 1802.6604, -2639.5849, 60.0632, -14.2931, 0.5718)^T,\ ||(\nabla F)_{\text{eff}}|| = 4256.44.$$

Der relative Fehler bezüglich der euklidischen Norm beträgt dabei

$$\frac{||(\nabla F)_{\text{klass}} - (\nabla F)_{\text{eff}}||}{||(\nabla F)_{\text{klass}}||} = 37692\%,$$

wohingegen der relative Fehler im Falle der in Abschnitt 4.3.1 angegebenen Gradienten nur 17.7% beträgt.

Die Methodik ist höchstens in der Nähe des globalen Minimums anwendbar, wo kleine Veränderungen nur geringe Auswirkungen auf das molekulare Modell haben. Dort stellt sich jedoch die Frage, ob der Gradient dann überhaupt noch korrekt berechnet werden kann. Allerdings ist dies wiederum auch nicht sinnvoll, denn in diesem Fall kann auch die in Abschnitt 3.6.4 eingeführte Methode der reduzierten Einheiten eingesetzt werden. Bei starren Molekülen, bei denen keine intramolekularen Wechselwirkungen auftreten und bei denen weder die Partialladungen noch sonstige Modellparameter außer den Lennard-Jones-Parametern einen Einfluß haben, könnte die Methode unter Umständen zum Erfolg führen. Da der effizient berechnete Gradient schon derart falsch war, wird die Hesse-Matrix auf diese Art und Weise völlig unmöglich zu ermitteln sein. Aufgrund dessen ist für die Hesse-Matrix kein Vergleich durchgeführt worden.

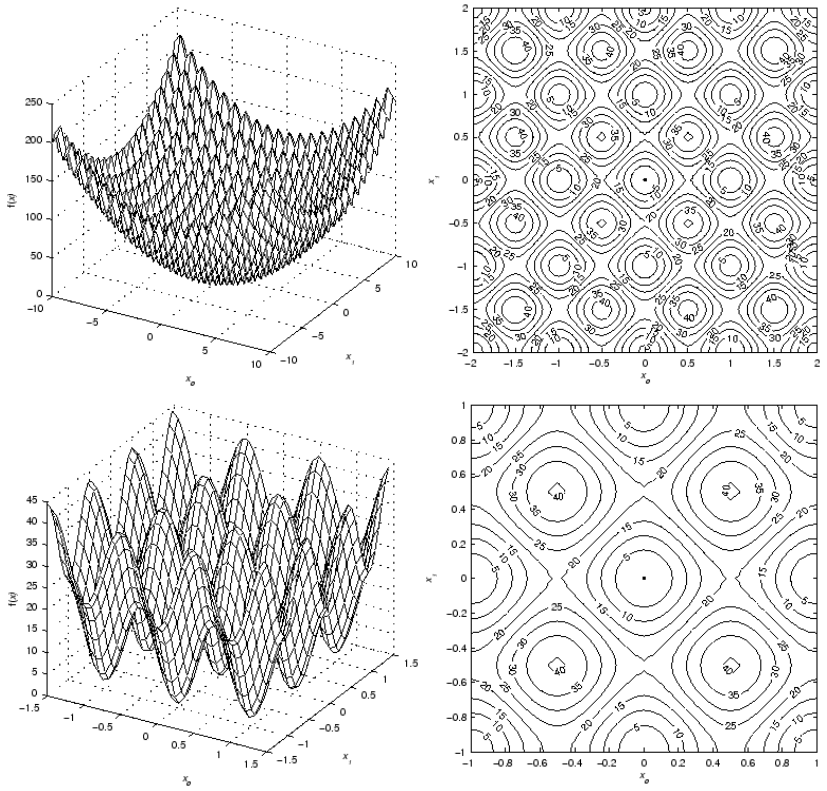


Abbildung 4.3: Rastrigin-Funktion auf den Gebieten $[-10, 10]^2$ und $[-1.5, 1.5]^2$. Dargestellt ist die Funktion jeweils dreidimensional und mit Höhenlinien. Auf dem größeren Gebiet ist das globale Minimum $(0, 0)$ deutlich zu erkennen, wohingegen es auf dem kleineren Gebiet so gut wie gar nicht von den benachbarten lokalen Minima unterscheidbar ist. Letzteres führt zu erheblichen Problemen für globale Optimierungsalgorithmen. Die Graphiken sind, mit freundlicher Genehmigung, von LUT (2011) übernommen.

4.4 Einsatz globaler Optimierungsverfahren

Eine weitere Verfahrensbewertung anhand der Korrelationsfunktionen wurde für die in Abschnitt 3.3.2 vorgestellte globale Optimierungsmethode CMA-ES durchgeführt. Abschnitt 4.4.1 befaßt sich mit der Anwendung von CMA-ES als globales und Abschnitt 4.4.2 mit dessen Anwendung als lokales Optimierungsverfahren.

4.4.1 CMA-ES als globaler Optimierer

Das globale Konvergenzverhalten des Optimierungsalgorithmus CMA-ES soll anhand der folgenden Kriterien bewertet werden:

- Wahl des Startvektors $x^{(0)}$, des initialen Vektors der Standardabweichungen $\sigma^{(0)}$ und der dadurch definierten initialen Kovarianzmatrix $C^{(0)} := I\sigma^{(0)}$.
- Wahl der Populationsgröße λ.
- Wahl des zulässigen Gebiets Ω.

- Wahl des Schwellenwertes $\tau > 0$ für das Abbruchkriterium $F(x) \leq \tau$.
- Erfolg von Zufallsreplikaten bei gleichverteiltem statistischem Rauschen Ψ.
- Konvergenzgeschwindigkeit, welche sich hier lediglich auf die Anzahl an Funktionsauswertungen bezieht.

Ein Maß für die Güte der Konvergenz sei gegeben durch den sogenannten *Gütequotienten*:

Definition 4.4.1 (Gütequotient)**.** *Es sei ν die relative Anzahl an erfolgreichen Zufallsreplikaten, welche das vorgegebene Abbruchkriterium erfüllen, und M die Anzahl an Funktionsauswertungen bis zum Erreichen des Abbruchkriteriums. Dann ist der* Gütequotient *G gegeben durch*

$$G = G\left(x^{(0)}, \sigma^{(0)}, \lambda, \Omega, \Psi, \tau\right) := \frac{(100 \cdot \nu)^2}{M}. \tag{4.2}$$

Das Erfolgsmaß ν wird dabei höher bewertet als die Anzahl an Funktionsauswertungen, da das Ziel darin bestehen soll, CMA-ES als globalen Voroptimierer für GROW einzusetzen. Ist CMA-ES nicht erfolgreich, so wird die anschließende gradientenbasierte Optimierung nicht durchführbar sein, und der aufgebrachte Rechenaufwand war unnötig.

Um eine Vorhersage darüber zu erhalten, inwieweit CMA-ES für die vorliegende Problemstellung geeignet ist, wurde eine Vorabbewertung anhand der *Rastrigin-Funktion* durchgeführt:

Definition 4.4.2 (Rastrigin-Funktion)**.** *Die Funktion*

$$\begin{aligned} R: \quad \mathbb{R}^2 &\to \mathbb{R} \\ (x_1, x_2) &\mapsto 20 + x_1^2 + x_2^2 - 10\left(\cos(2\pi x_1) + \cos(2\pi x_2)\right) \end{aligned} \tag{4.3}$$

heißt Rastrigin-Funktion.

Die Rastrigin-Funktion hat ein globales Minimum bei $(0, 0)$ mit Funktionswert $R(0, 0) = 0$ sowie eine Vielzahl an lokalen Minima. Abbildung 4.3 zeigt die Funktion sowohl mithilfe einer dreidimensionalen Darstellung als auch von Höhlenlinien. Außerdem ist anhand der Abbildung die Schwierigkeit einer globalen Minimierung der Rastrigin-Funktion erkennbar: Je nach Größe des zulässigen Gebiets ist das globale Minimum kaum von seinen benachbarten lokalen Minima zu unterscheiden. Daher wird die Rastrigin-Funktion sehr häufig als Bewertungsfunktion für globale Optimierer eingesetzt.

Tabelle 4.7 zeigt einige Ergebnisse der Anwendungen von CMA-ES zur globalen Minimierung der Rastrigin-Funktion. Als Abbruchkriterium wurde dabei $F(x) \leq 0.1$ gewählt. Es ist zu beachten, daß CMA-ES nicht dann abbrach, wenn ein Funktionswert unter das Abbruchkriterium fiel, sondern erst dann, wenn innerhalb verschiedener aufeinanderfolgender Generationen ein Großteil der Individuen einen Funktionswert kleiner als 0.1 hatte. Dies hatte zur Folge, daß die meisten Funktionswerte weit unter 0.1 lagen, also sehr nahe bei 0. Es ist deutlich zu sehen, daß für $\Omega = \mathbb{R}^2$, also im unrestringierten Fall, der Gütefaktor stark von der Lage der Startvektoren

$x^{(0)}$	$\sigma(0)$	λ	Ω	Ψ	ν	M	G
(0.5,0.5)	(3,3)	50	$\mathbb{R}^2$	0%	87%	911	8.3
(1,1)	(5,5)	45	$\mathbb{R}^2$	0%	87%	968	7.8
(4,4)	(5,5)	55	$\mathbb{R}^2$	0%	90%	1090	7.4
(100,100)	(100,100)	60	$\mathbb{R}^2$	0%	93%	1500	5.8
(5,4)	(1.7,1.2)	8	[4.5,7.8] × [3.5,5.8]	0%	95%	181	50
(5,4)	(1.7,1.2)	8	[4.5,7.8] × [3.5,5.8]	5%	91%	190	44
(5,4)	(1.7,1.2)	7	[4.5,7.8] × [3.5,5.8]	10%	75%	204	28
(13.8,8.8)	(6.2,1.2)	7	[7.5,20] × [7.5,10]	0%	90%	200	40
(13.8,8.8)	(6.2,1.2)	5	[7.5,20] × [7.5,10]	3%	66%	180	24
(13.8,8.8)	(6.2,1.2)	30	[7.5,20] × [7.5,10]	3%	90%	500	16.2

Tabelle 4.7: Ergebnisse der Anwendungen von CMA-ES auf die Rastringin-Funktion aus Gleichung (4.3) für $\tau = 0.1$. Im unrestringierten Fall ($\Omega = \mathbb{R}^2$) wurden nur sehr kleine Gütequotienten erreicht. Je weiter der Startwert vom Minimum entfernt war, desto kleiner wurde der Gütequotient. Eine hohe Erfolgsquote konnte nur mit einer hohen Standardabweichung, großen Populationen und hohem Rechenaufwand erzielt werden. Im restringierten Fall war die Erfolgsquote weitgehend unabhängig vom Ausmaß an statistischem Rauschen, lediglich die Anzahl an Funktionsauswertungen stieg mit größerem Ψ. War das Ausmaß an Rauschen jedoch so groß, daß durch die dadurch entstandenen Oszillationen die lokalen Minima und Maxima nicht mehr voneinander unterscheidbar waren, so konnte eine hohe Erfolgsquote wiederum nur mit hohem Rechenaufwand erzielt werden. Letzteres war bei der Wahl $\Omega = [7.5, 20] \times [7.5, 10]$ der Fall. Der Gütequotient sank jedoch drastisch mit der Größe der Population.

abhängig war: Je weiter diese von (0,0) entfernt waren, desto mehr Funktionsauswertungen waren für die Konvergenz erforderlich. Eine hohe Erfolgsquote war dabei nur mithilfe einer großen Standardabweichung zu gewährleisten. Der entscheidende Faktor ist dabei jedoch die Populationsgröße: Diese muß so eingestellt werden, daß einerseits die Konvergenz ins globale Minimum garantiert ist und andererseits der Rechenaufwand nicht zu groß wird. Im restringierten Fall, wo der Startvektor stets Mittelpunkt des zulässigen Gebiets war und die Standardabweichung stets genau auf dessen Rand führte, war der Gütequotient vor allem davon abhängig, wie viele lokale Minima in unmittelbarer Umgebung des globalen Minimums waren und wie sich diese voneinander unterschieden. Waren diese unterscheidbar, so wirkte sich auch statistisches Rauschen nicht negativ auf das Konvergenzverhalten aus. Führten die durch das Rauschen hervorgerufenen Oszillationen allerdings zum gegenteiligen Effekt oder war die Restriktion so gewählt, daß bereits ohne Rauschen im zulässigen Gebiet die Minima nur kaum unterscheidbar waren, so sank der Gütefaktor drastisch, und eine hohe Erfolgsquote war nur noch mit einer hohen Populationsgröße und somit mit weitaus höherem Rechenaufwand erreichbar. Zusammenfassend läßt sich aufgrund der praktischen Konvergenzanalyse anhand der Rastrigin-Funktion folgendes schließen: Die Güte der Konvergenz von CMA-ES ist abhängig von den initialen Parametern und Standardabweichungen, sehr stark von der Populationsgröße und der Wahl der Restriktion, welche bestimmt, ob statistisches Rausches Einfluß hat oder nicht. War die Wahl der Restriktion günstig, so hat sich CMA-ES auch in weiteren Analysen generell als sehr robust in Bezug auf Rauschen erwiesen. Es ist deutlich robuster als gradientenbasierte Verfahren. Außerdem konvergierte CMA-ES im Falle der Rastrigin-Funktion bei günstigem zulässigen Gebiet mit hoher Erfolgsquote und robust in die Nähe des globalen Minimums. Ein weiterer Faktor vor allem für die Konvergenzgeschwindigkeit ist die Wahl des Abbruchkriteriums: War τ zu klein gewählt, so stagnierte CMA-ES unnötig in der Nähe des globalen Minimums, ohne noch signifikante Verbesserungen zu erzielen. Da das Verfahren nicht gerichtet ist, werden nur noch zufällig in einer Umgebung des Minimums Datenpunkte ausgewertet und die Konvergenz ist im allgemeinen langsamer als bei gradientenbasierten Verfahren. Daher sollte im Hinblick auf die in Abschnitt

3.6.6 dargestellte Kombinationsidee das Abbruchkriterium so gewählt werden, daß einerseits garantiert ist, daß CMA-ES in die Nähe des globalen Minimums gelangt ist und andererseits unnötiger Rechenaufwand vermieden wird, so daß der Einsatz von GROW an dieser Stelle dazu führt, daß das Minimum effizienter erreicht wird. Inwieweit diese Kombination tatsächlich eine Effizienzerhöhung darstellt, wird in Abschnitt 4.5.1 analysiert. Dieser Abschnitt befaßt sich nur mit der globalen Konvergenz von CMA-ES alleine. Es wird daher im folgenden das globale Konvergenzverhalten des Optimierers anhand der Korrelationsfunktionen evaluiert.

Aufgrund der mithilfe der Analysen anhand der Rastrigin-Funktion getroffenen Vorüberlegungen, ist die Vermutung, daß CMA-ES zur globalen Optimierung oder zumindest als Voroptimierer für die vorliegende Problemstellung geeignet ist, naheliegend: Das zulässige Gebiet ist aufgrund von chemischer und physikalischer Intuition so zu wählen, daß das globale Minimum darin enthalten ist. Aufgrund der Überlegungen in Abschnitt 3.5.3 und der Bewertung der gradientenbasierten Verfahren in Abschnitt 4.2 läßt sich sagen, daß die Fehlerfunktion zumeist sehr steil in Richtung des globalen Optimums abfällt. Zumeist hat die Funktion die Gestalt einer steilen Regenrinne. Auch statistisches Rauschen zerstört nicht den Trend der Fehlerfunktion, ansonsten wären die gradientenbasierten Verfahren gar nicht anwendbar. Die Restriktion in Abschnitt 4.2 ist somit gutartig. Es bleiben jedoch zwei Fragen offen:

1. **Kann CMA-ES auch dann zum Ziel führen, wenn man keine Vorstellung davon hat, wie die Startparameter und das zulässige Gebiet gewählt werden können?** Diese Frage kann jedoch mithilfe der Korrelationsfunktionen nicht beantwortet werden, da deren zulässiges Gebiet zu klein ist. Generell wird sich diese Frage nicht mit ja beantworten lassen, da man die Gestalt einer unrestringierten Fehlerfunktion im allgemeinen nicht kennt.
2. **Ist die Restriktion so gutartig, daß CMA-ES zuverlässig in die Nähe eines globalen Minimums führt?** Es ist möglich, daß sich am Boden der Regenrinne verschiedene lokale und nur ein globales Minimum befinden. Dann könnten bei CMA-ES dieselben Probleme auftreten wie im Falle der Rastrigin-Funktion. Diese Frage wird im folgenden näher untersucht.

Tabelle 4.8 zeigt die Ergebnisse von mehreren globalen Anwendungen auf die Korrelationsfunktionen. Im Gegensatz zu Abschnitt 4.1 wurde der folgende zulässige Bereich gewählt: $0.29 \leq \sigma/\mathrm{nm} \leq 0.4$, $0.3 \leq \varepsilon/(\mathrm{kJ/mol}) \leq 0.4$, $0 \leq (Q/\mathrm{Dnm})^2 \leq 0.0468$ und $0 \leq L/\mathrm{nm} \leq 0.24$. Dadurch konnten die zulässigen Intervalle für alle Kraftfeldparameter außer für ε erheblich vergrößert werden, was bei einer globalen Optimierung sinnvoll ist. Die in Abschnitt 4.1 angegebenen von GROW gefundenen lokalen Minima lagen auch innerhalb des neuen zulässigen Gebiets. Der Startvektor war stets das Zentrum des zulässigen Gebiets, und die initialen Standardabweichungen wurden wieder so gewählt, daß der Rand des Gebiets erreicht wurde. Auf die simulierten Daten beider zu optimierender physikalischer Eigenschaften wurde ein gleichverteiltes Rauschen von maximal 5% addiert, da gerade bei Startwerten, die weit weg vom Minimum liegen, ein derart hohes Rauschen angenommen werden sollte. Im Falle eines erfolglosen Optimierungsablaufs oder einer zu hohen Anzahl an benötigten Funktionsauswertungen wurden die Zufallszahlen wieder auf die in Abschnitt 3.5.3 angegebenen Werte gesetzt. Da es sich bei den zu untersuchenden Optimierungsaufgaben nur um die verrauschten handelt, zeigt die Tabelle die Ergebnisse der Aufgaben 2, 4, 6 und 8. CMA-ES wurde abgebrochen, sobald ein

Optimierungsaufgabe	τ	λ	Ψ	ν	M	G
2	$3.5{\cdot}10^{-3}$	4	(5%,5%)	100%	31	322.6
		8	(5%,5%)	100%	38	263.2
4	0.013	versch.	(5%,5%)	20%	> 100	< 4
			(0.5%,1%)	20%	> 100	< 4
6	$5{\cdot}10^{-3}$	20	(5%,5%)	100%	30	333.33
		8	(5%,5%)	100%	56	178.6
8	0.015	8	(5%,5%)	100%	210	47.6
		15	(5%,5%)	100%	210	47.6
		8	(0.5%,3%)	100%	126	79.4

Tabelle 4.8: Ergebnisse der Anwendungen von CMA-ES als globaler Optimierer auf die Korrelationsfunktionen aus Abschnitt 3.5.2: Angegeben sind die Resultate von zehn unabhängigen Zufallsreplikaten. Bei den Optimierungsaufgaben 2 und 6 (eine Temperatur) lieferte CMA-ES bei großem Rauschen sehr gute Ergebnisse. Allerdings waren die gefundenen Minima aufgrund der Regenrinnenstruktur der Fehlerfunktion weit verstreut. Bei den Optimierungsaufgaben 4 und 8 (sechs Temperaturen) sind die Gütequotienten wesentlich kleiner, vor allem bei Optimierungsaufgabe 4. Bei Optimierungsaufgabe 8 konnte G mit moderatem Rauschen um etwa den Faktor 2 erhöht werden, allerdings war die Anzahl an Funktionsauswertungen mit 126 immer noch sehr hoch. Bemerkenswert ist, daß die Erfolgswahrscheinlichkeit von CMA-ES in den meisten Fällen bei 100% liegt.

Fehlerfunktionswert kleiner als τ war, da im Gegensatz zur Rastrigin-Funktion hierbei bereits geeignete Erfahrungswerte für τ bekannt waren.
Bei einer Temperatur, also bei den Optimierungsaufgaben 2 und 4, war die globale Optimierung von CMA-ES bemerkenswert gut. Die Gütequotienten lagen weit über 100. Allerdings waren die von CMA-ES gefundenen globalen Minima weit über den zulässigen Bereich hinweg verstreut, was eine weitere Bestätigung der Regenrinnenstruktur der Fehlerfunktion darstellt. Daß sich am Boden der Regenrinne weitere Minima befinden, konnte ausgeschlossen werden, da die ermittelten Fehlerfunktionswerte stets nahezu gleich 0 waren. Im Falle der Optimierungsaufgaben 4 und 8 waren die Gütequotienten deutlich kleiner. Bei Optimierungsaufgabe 4 war er stets kleiner als 4. Allerdings gelangte CMA-ES stets in eine Umgebung des Vektors $(0.3781, 0.3143, 0.0053, 0.2242)^T$. Von GROW konnte dieser Vektor nicht gefunden werden, da er nicht innerhalb des zulässigen Gebiets lag. Dies ist jedoch eine weitere Motivation dafür, CMA-ES mit GROW zu kombinieren, um das globale Minimum möglichst effizient und exakt zu bestimmen. Bei Optimierungsaufgabe 8 konnte der Gütequotient bei moderatem Rauschen um etwa den Faktor 2 erhöht werden, allerdings waren immer noch zu viele Funktionsauswertungen nötig. Auch hierbei konvergierte CMA-ES stets in die Umgebung desselben Minimums. Bei mehreren Temperaturen können also am Boden der Regenrinne mehrere Minima auftreten. CMA-ES konvergierte in den meisten Fällen sehr zuverlässig in die Nähe eines dieser Minima. Die Restriktion war somit gutartig. Eine Kombination mit GROW scheint aufgrund des hohen Rechenaufwands jedoch bereits zu diesem Zeitpunkt äußerst angemessen zu sein. Es sei allerdings noch hervorgehoben, daß CMA-ES nicht in die Nähe des globalen Minimums konvergierte. Weitere Analysen haben ergeben, daß es einen Bereich gibt, wo die Fehlerfunktion noch kleinere Werte annimmt. Dieser Bereich wurde jedoch nie von CMA-ES gefunden.

Optimierungsaufgabe	τ	Ψ	Methode	λ	ν	M	$\tilde{G}$
2	$3.5{\cdot}10^{-3}$	(0.5%,1%)	CMA-ES	4	100%	12	8.33
			PR	–	80%	52	1.54
4	0.013	(0.5%,1%)	CMA-ES	8	100%	104	0.96
			FR	–	80%	61	1.31
6	$5{\cdot}10^{-3}$	(0.5%,3%)	CMA-ES	8	100%	80	1.25
			PR	–	70%	79	0.87
8	0.015	(0.5%,3%)	CMA-ES	8	100%	132	0.76
			PR	–	100%	55	1.82

Tabelle 4.9: Ergebnisse der Anwendungen von CMA-ES als lokaler Optimierer auf die Korrelationsfunktionen aus Abschnitt 3.5.2. Bei flachen Fehlerfunktionen, also im Falle der Optimierungsaufgaben 2 und 6, ist CMA-ES besser als lokaler Optimierer geeignet als GROW, für das stets die Ergebnisse von zehn unabhängigen Zufallsreplikaten des jeweils besten gradientenbasierten Verfahrens angegeben sind. Bei den relevanten Optimierungsverfahren 4 und 8 ist es für CMA-ES problematischer, durch zufälliges Abtasten des Parameterraums Verbesserungen zu erzielen. Gerichtete Verfahren sind hier besser geeignet. GROW liefert in diesen Fällen deutlich bessere Ergebnisse als CMA-ES.

4.4.2 CMA-ES als lokaler Optimierer

Wie bereits mehrfach motiviert, stellt sich die Frage, inwieweit eine Kombination von CMA-ES mit GROW eine Verbesserung darstellt. Dazu ist zunächst zu überprüfen, wie sich CMA-ES als lokaler Optimierer im Einzugsbereich eines Minimums verhält. Anschließend ist der entsprechende Kombinationsalgorithmus zu evaluieren, womit sich Abschnitt 4.5.1 befaßt.
Zum Vergleich von CMA-ES mit GROW wurden dieselben Startparameter gewählt wie bei GROW (siehe Abschnitt 4.1). Weiterhin wurden die initialen Standardabweichungen klein gewählt. Im folgenden sei $\sigma^{(0)} := (0.007, 0.007, 0.004, 0.02)^T$, entsprechend den Größenordnungen der zulässigen Intervalle für die einzelnen Kraftfeldparameter.
Tabelle 4.9 zeigt den direkten Vergleich zwischen CMA-ES und GROW als lokale Optimierer im Falle der verrauschten Optimierungsaufgaben. Im Hinblick auf Simulationen sind nur diese vier Aufgaben interessant. Am relevantesten sind die Optimierungsaufgaben 4 und 8, da hierbei zu mehreren Temperaturen simultan optimiert wird. Gemäß Abschnitt 4.4.1 weist CMA-ES eine sehr hohe Erfolgsquote auf, zumeist 100%, was sich auch hier widerspiegelte. Bei der lokalen Optimierung sind jedoch Erfolgsquote und Konvergenzgeschwindigkeit gleich zu werten, denn in diesem Falle würde es keinen Gewinn darstellen, robust zum Ziel zu gelangen, dafür jedoch eine Vielzahl an Simulationen durchführen zu müssen. Im globalen Fall wäre dies hingegen tolerierbar, wenn man sich sehr weit vom globalen Minimum entfernt befindet und die Fehlerfunktion erheblich verrauscht und zerklüftet ist. Daher wird hier anstelle von G (siehe Abschnitt 4.4.1) der modifizierte Gütequotient

$$\tilde{G} = \tilde{G}\left(x^{(0)}, \sigma^{(0)}, \lambda, \Omega, \Psi, \tau\right) := \frac{100 \cdot \nu}{M} \tag{4.4}$$

betrachtet. Es ist zu beachten, daß $\tilde{G}$ im Falle von GROW nicht von $\sigma^{(0)}$ und λ abhängig ist. Bei den Optimierungsaufgaben 2 und 6 ist $\tilde{G}$ bei CMA-ES größer als bei GROW. In diesen Fällen ist die Fehlerfunktion flacher, so daß CMA-ES bessere Chancen hat, durch das zufällige Abtasten des Parameterraums Verbesserungen zu erzielen. Bei den im Hinblick auf Simulationen relevanten Optimierungsaufgaben 4 und 8 jedoch, wo die Fehlerfunktion steiler ist, ist GROW als lokaler Optimierer deutlich besser geeignet als CMA-ES. Theoretisch ist dies damit zu

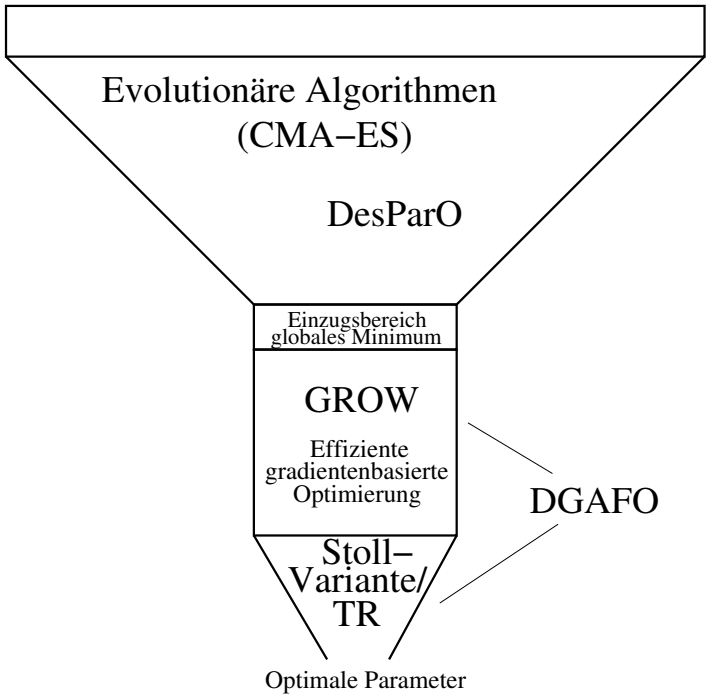


Abbildung 4.4: Kombinationsidee als Filterungsprozeß: Zu Beginn wird mithilfe globaler Optimierungsmethoden wie zum Beispiel Evolutionären Algorithmen oder DesParO der Parameterraum effizient abgetastet, um in den Einzugsbereich eines globalen Minimums zu gelangen. Dort konvergieren gradientenbasierte Verfahren wie in GROW implementiert oder die ableitungsfreie Methode DGAFO wesentlich schneller. In der Nähe des Minimums sind spezielle Methoden zu verwenden, welche das Minimum möglichst exakt bestimmen. Zum Schluß erhält man dann die optimalen Parameter.

erklären, daß gradientenbasierte Verfahren gerichtet sind und sich eine zufällige Parametersuche in der Nähe eines Minimums als ineffizient erweist. Im Einzugsbereich des globalen Minimums sollte somit GROW anstelle von CMA-ES verwendet werden.

4.5 Anwendung eines Kombinationsalgorithmus

Aufgrund der Überlegungen in Abschnitt 3.6.6 ist die Anwendung eines Optimierungsverfahrens alleine in vielen Fällen nicht ausreichend, um das Minimum möglichst exakt zu bestimmen. Gradientenbasierte Verfahren können zum einen nur dann konvergieren, wenn die Startparameter im Einzugsbereich des Minimums liegen, und zum anderen ist es möglich, daß die Richtung des Gradienten in der Nähe des Minimums falsch berechnet wird, oder aber die Armijo-Schrittweiten werden zu schnell zu klein. In solchen Fällen sind die hier betrachteten Verfahren miteinander zu kombinieren, und zwar so, daß die Stärken der einzelnen Verfahren in bestimmten Funktionsbereichen ausgenutzt werden. Abbildung 4.4 zeigt die Kombinationsidee: Sind keinerlei Startparameter aus der Literatur zugänglich oder sind diese für eine gradientenbasierte Optimierung ungeeignet, so wird zunächst ein globales Optimierungsverfahren vorgeschaltet, zum Beispiel ein Evolutionärer Algorithmus wie CMA-ES oder DesParO (siehe Abschnitt 3.3). Da gemäß Abschnitt 4.4.2 GROW aufgrund der höheren Konvergenzgeschwindigkeit als lokaler Op-

timierer besser geeignet ist als CMA-ES, wird in Abschnitt 4.5.1 die Kombination von CMA-ES mit GROW als globale Optimierung anhand der Korrelationsfunktionen aus Abschnitt 3.5.2 bewertet. Der Ersatz von GROW durch das ableitungsfreie DGAFO-Verfahren aus Kapitel 6 ist an dieser Stelle ebenfalls denkbar, wird jedoch später diskutiert. Wie in Abschnitt 3.6.6 bereits motiviert, scheint in der Nähe des Minimums eine Modellierung beziehungsweise Approximation der Fehlerfunktion am besten geeignet zu sein. In Abschnitt 4.5.2 werden zwei gradientenbasierte Verfahren bewertet, das Trust-Region-Verfahren mit exakter Lösung des Teilproblems (siehe Abschnitt 3.4.3) und eine Variante des Verfahrens nach Stoll (siehe Abschnitt 3.6.5). Ob diese Verfahren trotz der notwendigen Gradientenberechnung näher ans Minimum gelangen als GROW, wird im einzelnen zu testen sein. Allerdings wird die in Abschnitt 4.5.2 durchgeführte Bewertung nicht wie bisher auf Molekulare Simulationen übertragbar sein. Daher wird später eine nachfolgende Bewertung anhand Molekularer Simulationen erfolgen, die auch zum Vergleich mit dem DGAFO-Verfahren verwendet wird. Der vorliegende Abschnitt stellt insgesamt den Verlauf eines in Abschnitt 3.6.6 motivierten Kombinationsalgorithmus dar.

4.5.1 Kombination von CMA-ES mit GROW

Aufgrund der Ergebnisse aus Abschnitt 4.5.1 bleibt nunmehr zu überprüfen, wie sich die Kombination von CMA-ES mit GROW in Bezug auf die Konvergenz gegen das globale Minimum verhält. Die Problematik bei einer derartigen Kombination liegt in der geeigneten Wahl des Abbruchkriteriums für CMA-ES. Der globale Optimierer sollte terminiert werden, sobald er in den Einzugsbereich des globalen Minimums gelangt ist. Dabei sind selbstverständlich überflüssige Funktionsauswertungen zu vermeiden. Es wird davon abgesehen, das Abbruchkriterium so einzustellen, daß die Konvergenz gegen das globale Minimum garantiert ist, da die Evaluation anhand der Rastrigin-Funktion aus Abschnitt 4.4.1 ergeben hat, daß dies zu viele Funktionsauswertungen erfordern würde. Somit wurden verschiedene Schwellenwerte getestet, um die Abhängigkeit der Konvergenz von der Wahl des Abbruchkriteriums zu überprüfen.

Die Ergebnisse der Kombination von CMA-ES mit GROW sind in Tabelle 4.10 aufgezeigt. Es wurden lediglich die Optimierungsaufgaben 4 und 8 betrachtet, da nur diese relevant in Bezug auf Molekulare Simulationen sind. Die Tabelle zeigt die Erfolgsquote, die Anzahl an benötigten Funktionsevaluationen sowie den Gütequotienten $\tilde{G}$ aus Gleichung (4.4) in Abhängigkeit vom Abbruchkriterium für CMA-ES

$$F(x) \leq \tau_{\mathrm{CMA-ES}},\ 0 < \tau < \tau_{\mathrm{CMA-ES}}.$$

GROW ist wie erwartet dazu in der Lage, den Rechenaufwand drastisch zu reduzieren. Daß dabei die Erfolgsquote sinkt, ist tolerierbar, zumal hier überschätztes, gleichverteiltes statistisches Rauschen angenommen wurde, was bei Molekularen Simulationen in dieser Form nicht vorliegt. Bei Optimierungsaufgabe 4 war die Erfolgsquote von CMA-ES alleine auch sehr gering, bei Optimierungsaufgabe 8 lag sie bei 100%, die der Kombination für $\tau_{\mathrm{CMA-ES}} = 0.1$ immerhin noch bei 80%. Weniger zufriedenstellend ist jedoch die starke Abhängigkeit des Gütequotienten vom Abbruchkriterium für CMA-ES, welcher je nach Wahl von $\tau_{\mathrm{CMA-ES}}$ stark variiert. Die Frage, inwieweit die Kombination von CMA-ES mit GROW tatsächlich auf Simulationen anwendbar ist, bleibt demzufolge noch offen, denn es ist möglich, daß CMA-ES in einen Bereich des Parameterraums gelangt, in dem die Fehlerfunktion derart zerklüftet und trendlos ist, daß gradientenbasierte Verfahren kaum noch zum Erfolg führen.

Optimierungsaufgabe	Ψ	Methode	τ	$\tau_{\mathrm{CMA-ES}}$	ν	M	$\tilde{G}$
4	(0.5%,1%)	CMA-ES + PR	0.013	0.1	50%	584	0.09
				> 0.04	0%	–	0
				0.04	**30%**	**56**	**0.54**
				0.03	40%	104	0.38
				0.02	10%	151	0.07
		CMA-ES	0.013	–	20%	> 100	< 0.2
8	(0.5%,3%)	CMA-ES + PR	0.015	1.0	20%	73	0.27
				0.5	40%	74	0.54
				0.4	50%	72	0.69
				0.3	50%	100	0.50
				0.2	70%	48	1.46
				0.15	20%	35	0.57
				0.1	**80%**	**79**	**1.01**
				0.05	70%	80	0.88
				0.04	70%	103	0.68
				0.03	90%	130	0.69
				0.02	100%	116	0.86
		CMA-ES	0.015	–	100%	126	0.79

Tabelle 4.10: Ergebnisse der Kombination von CMA-ES mit GROW bei den Optimierungsaufgaben 4 und 8. Es ist deutlich zu sehen, daß bei einem bestimmtem $\tau_{\mathrm{CMA-ES}}$ die Kombination signifikant besser abschneidet als die Verwendung von CMA-ES alleine. GROW sorgt dabei für eine merkliche Reduktion an Funktionsauswertungen, senkt allerdings auch die Erfolsquote. Die höchsten Gütekoeffizienten wurden bei Optimierungsaufgabe 4 für $\tau_{\mathrm{CMA-ES}} = 0.04$ und bei Optimierungsaufgabe 8 für $\tau_{\mathrm{CMA-ES}} = 0.1$ erzielt. Im Fall von $\tau_{\mathrm{CMA-ES}} = 0.2$ war $\tilde{G} = 1.46$ ein Zufallsergebnis. Außer bei einem Replikat wurde stets ein Schwellenwert < 0.1 erreicht, daher wird $\tau_{\mathrm{CMA-ES}} = 0.1$ hierbei favorisiert.

Aufg.	GROW			CMA-ES			Kombination			
	ν	M	$\tilde{G}$	ν	M	$\tilde{G}$	ν	M	$\tilde{G}$	$\tau_{\mathrm{CMA-ES}}$
1	3/3	385	0.26	3/3	202	0.50	3/3	48	2.08	0.01
2	2/3	41	1.63	3/3	32	3.13	3/3	10	10.0	
3	0/3	–	0	2/3	536	0.12	0/3	–	0	versch.
4	0/3	–	0	0/3	–	0	0/3	–	0	versch.
5	3/3	186	0.54	3/3	200	0.50	3/3	114	0.88	0.01
6	2/3	42	1.59	3/3	86	1.16	2/3	48	1.39	0.01
7	3/3	270	0.37	3/3	186	0.54	2/3	112	0.60	0.3
8	0/3	–	0	3/3	93	1.08	2/3	56	1.19	0.3

Tabelle 4.11: Vergleich der Ergebnisse von GROW, CMA-ES und dem Kombinationsalgorithmus bei allen Optimierungsaufgaben. In diesem Fall wurden nur drei statt zehn statistisch unabhängige Zufallsreplikate verwendet. Im Falle der Kombination sind die Ergebnisse für das jeweils optimale $\tau_{\mathrm{CMA-ES}}$ angegeben. Außer bei den Optimierungsaufgaben 3 und 4, wo die Erfolgsquote im allgemeinen sehr gering war, war der Gütequotient der Kombination stets größer als der der Anwendung von CMA-ES alleine. Bei Optimierungsaufgabe 6 war er kleiner als der von GROW alleine, was jedoch nur Zufall war.

Auch Tabelle 4.11 bestätigt, daß die Kombination von CMA-ES mit GROW am besten geeignet ist. Angegeben sind dort die Ergebnisse von drei statistisch unabhängigen Zufallsreplikaten für alle acht Optimierungsaufgaben aus Abschnitt 3.5.3. Bei der Kombination wurde stets das jeweils optimale $\tau_{\mathrm{CMA-ES}}$ gewählt. In den Fällen, wo die Erfolgsquote aller Algorithmen im allgemein hoch war, war der Gütequotient der Kombination am höchsten, außer im Falle von Optimierungsaufgabe 6, wo der von GROW alleine dominierte. Bei dieser Aufgabe war GROW auch als globaler Optimierer besser geeignet als CMA-ES. Anscheinend hat die Fehlerfunktion in diesem Fall einen klaren Trend ins globale Minimum. Daß allerdings GROW alleine im Durchschnitt nur 42 und die Kombination 48 Funktionsevaluationen benötigte, ist jedoch als Zufallsergebnis zu werten.

Trotz der oben beschriebenen Nachteile ist es ratsam, CMA-ES mit GROW zu kombinieren, auch wenn dadurch die Zuverlässigkeit des Optimierungsalgorithmus sinkt. Insgesamt war jedoch eine signifikante Reduktion an Funktionsauswertungen zu beobachten. Somit läßt sich der Kombinationsalgorithmus in jedem Fall auf Molekulare Simulationen hin testen. Dabei wird jedoch ein anderes Abbruchkriterium für CMA-ES gewählt als in diesem Abschnitt, da die Abhängigkeit des Erfolgs der Kombination von $\tau_{\mathrm{CMA-ES}}$ zu groß ist. Es sei hier bereits erwähnt, daß CMA-ES nicht dazu geeignet ist, mit akzeptablem Rechenaufwand robust in die Nähe des globalen Minimums zu gelangen. Auch in diesem Abschnitt konnte das globale Minimum nicht gefunden werden. Anschließende Untersuchungen haben ergeben, daß es einen Bereich gibt, wo die Fehlerfunktion noch kleinere Werte aufweist. Dieser Bereich wurde von CMA-ES allerdings nie erreicht.

4.5.2 Kombination von GROW mit einem gradientenbasierten Verfahren zur möglichst exakten Bestimmung des globalen Minimums

Es bleibt nun zu untersuchen, welches Verfahren dazu geeignet ist, das Minimum möglichst exakt zu bestimmen, das heißt dann angewendet werden kann, wenn GROW keine Verbesserungen mehr liefert. Hierzu eignen sich Verfahren, welche die Fehlerfunktion modellieren beziehungsweise approximieren. Gradientenbasierte Verfahren, die dafür in Frage kommen, sind das in Abschnitt 3.4.3 dargestellte Trust-Region-Verfahren mit exakter Lösung des Teilproblems mit gegebenenfalls effizienter Hesse-Berechnung (siehe Abschnitt 3.6.3) und die in Abschnitt 3.6.5 eingeführte Variante des Verfahrens nach Stoll. Um eine geeignete Approximation zu erstellen, muß die Fehlerfunktion möglichst glatt sein. Daher werden die in Anhang H.1 angesprochenen temperaturabhängigen Fits für die zu optimierenden physikalischen Zielgrößen verwendet. Aufgrund dessen wird das exakte Trust-Region-Verfahren hier von GROW losgelöst betrachtet. Ob eine korrekte Gradienten- beziehungsweise Hesse-Matrix-Berechnung dadurch ermöglicht wird, wird in diesem Abschnitt getestet. Für eine erste Bewertung, die sich sowohl auf die Optimalität der Ergebnisse als auch auf den Rechenaufwand bezieht, werden wieder die in Abschnitt 3.5.2 behandelten Korrelationsfunktionen in Betracht gezogen. Es werden dabei nur die relevanten Optimierungsaufgaben 4 und 8 verwendet. Es sei allerdings darauf hingewiesen, daß bei diesen die Berechnung der Systemeigenschaften durch die temperaturabhängigen Fitfunktionen erfolgt. Das anschließend addierte künstliche Rauschen wird durch die gewichtete nichtlineare Regression, die für die Berechnung der Temperaturfits notwendig ist, wieder herausgefiltert, so daß die Betrachtung der Optimierungsaufgaben 4 und 8 der der Optimierungsaufgaben 3 und 7 gleichkäme. Somit ist in jedem Fall eine nachfolgende Evaluation der Verfahren bei Molekularen Simulationen erforderlich. Die insgesamt erzielten Ergebnisse werden später mit denen des DGAFO-Verfahrens in der Nähe des Minimums verglichen.
Die in Abschnitt 3.6.4 dargstellte Methode der reduzierten Einheiten ist im Falle der Korrelationsfunktionen nicht anwendbar, da sie ausschließlich für Lennard-Jones-Parameter geeignet ist. Daher werden im folgenden nur die Variante des Verfahrens nach Stoll und das exakte Trust-Region-Verfahren in Bezug auf ihr Verhalten in der Nähe des Minimums evaluiert.

	# It.	# Eval.	x^{opt}	ρ_l	p_σ	# It.	# Eval.	x^{opt}	ρ_l	p_σ
	Variante des Verfahrens nach Stoll					Trust-Region-Verfahren				
$\bar{R}$	2–3	20–21	0.3713 0.3130 0.0229 0.1871	0.69% (0.09%)	1.89% (0.47%)	2–3	60–61	0.3705 0.3131 0.0239 0.1879	0.62% (0.13%)	1.62% (0.46%)
$R_{\min}$	4	23	0.3733 0.3124 0.0205 0.1845	0.51%	1.93%	3	70	0.3704 0.3132 0.0242 0.1879	0.41%	1.32%

Tabelle 4.12: Vergleich der Variante des Verfahrens nach Stoll mit dem exakten Trust-Region-Verfahren basierend auf Temperaturfits in der Nähe des Minimums. Angegeben sind Durchschnittswerte über zehn statistisch unabhängige Replikate ($\bar{R}$) und die Ergebnisse für den jeweils kleinsten MAPE-Wert bezüglich ρ_l ($R_{\min}$). Standardabweichungen sind in Klammern angegeben. Bezüglich Siedichte (ρ_l) und Dampfdruck (p_σ) sind jeweils die erreichten MAPE-Werte angezeigt. Beide Verfahren gelangten äußerst robust näher an das Minimum heran als GROW, wobei die Variante des Verfahrens nach Stoll aufgrund der effizienten Gradientenberechnung viel weniger Simulationen benötigte und stärkere Veränderungen der Kraftfeldparameter erzielte als das exakte Trust-Region-Verfahren. Bei letzterem wurde die Hesse-Matrix nie effizient berechnet.

Da in diesem Abschnitt ein Kombinationsalgorithmus evaluiert werden soll, der gegen das globale Minimum der Fehlerfunktion konvergiert, sind zunächst geeignete Startwerte für die beiden Verfahren zu bestimmen, welche sich aus der Kombination von CMA-ES mit GROW aus Abschnitt 4.5.1 ergeben haben. Die in Hülsmann u. a. (2010b) erhaltenen Kraftfeldparameter lagen im Durchschnitt bei $x^\tau := (0.3056, 0.2384, 0.0175, 0.1120)^T$, gemittelt über zehn statistisch unabhängige Replikate, wohingegen CMA-ES, wie Abschnitt 4.4.1 zu entnehmen ist, gegen ein Minimum konvergierte, welches sich innerhalb von $\Omega^{\mathrm{opt}} := [0.36, 0.38] \times [0.30, 0.32] \times [0.01, 0.03] \times [0.17, 0.19]$ befand. Diese Menge überschneidet sich nicht mit dem zulässigen Gebiet aus Abschnitt 4.1, das heißt, weder in Hülsmann u. a. (2010b) noch in der Verfahrensbewertung aus Abschnitt 4.2 hätte das Minimum in Ω^{opt} gefunden werden können. Sowohl bei der globalen Optimierung durch CMA-ES alleine als auch bei der Kombination von CMA-ES mit GROW konnte festgestellt werden, daß die Streuung der Kraftfeldparameter, die das jeweilige Abbruchkriterium erfüllten, zwar deutlich geringer war als bei den Optimierungsaufgaben 2 und 6, wo nur eine Temperatur betrachtet wurde, allerdings immer noch einige Ausreißer vorhanden waren. Es war ein geeigneter Startvektor aus denjenigen zehn Replikaten auszuwählen, der den Ergebnissen aus Tabelle 4.10 im Falle $\tau_{\mathrm{CMA-ES}} = 0.1$ entsprach. Das Abbruchkriterium für GROW war $F(x) \leq 0.015$. Um einen nahtlosen Anschluss zu gewährleisten, wurden nur die drei Replikate in Betracht gezogen, für die $F(x) \approx 0.015$ galt. Von diesen drei Replikaten konnten zwei identifiziert werden, die zu weit weg von den durchschnittlichen durch die Kombination erhaltenen Kraftfeldparametern lagen. Somit wurde als Startvektor $x^{(0)} := (0.3694, 0.3094, 0.0243, 0.1879)^T$ selektiert. Wie bereits in Abschnitt 4.2 dargelegt, ist die dort verwendete Gewichtung der Fehlerfunktion praktisch unentbehrlich, um ein Minimum möglichst exakt zu bestimmen. Obwohl also in Abschnitt 4.5.1 $\forall_{i,T}\, w_{i,T} = 1$ gesetzt wurde, wird in diesem Abschnitt wieder die Gewichtung verwendet. Als nächstes ist zu überprüfen, wie weit GROW von diesem Startvektor aus noch in die Nähe des Minimums gelangt. Der Dampfdruck war bereits fast bis auf statistische Unsicherheiten genau vorhergesagt worden, der MAPE-Wert

Verfahren	τ	# It.	# Eval.	# It.	# Eval.
	durchschnittlich			ein Replikat	
CMA-ES	0.1	7–8	59	3	24
GROW(1)	0.015	2–3	20	4	22
GROW(2)	–	5–6	46	5	39
Stoll	–	2–3	20	4	23
TR	–	2–3	60	3	70
Gesamt Stoll		18	145	17	108
Gesamt TR		18	185	16	155

Tabelle 4.13: Anzahl an Iterationen und Funktionsevaluationen für die einzelnen Verfahren innerhalb der beiden Kombinationsalgorithmen. Angegeben sind Durchschnittswerte und die Ergebnisse des besten Replikats sowie die Gesamtanzahl an Iterationen und Funktionsevaluationen für die beiden Kombinationsalgorithmen. GROW(1) bezieht sich auf die Kombination von CMA-ES mit GROW und GROW(2) auf die Untersuchung, wie weit GROW noch an das Minimum herankommt. Die Variante des Verfahrens nach Stoll brauchte durchschnittlich 40 Simulationen weniger als das exakte Trust-Region-Verfahren basierend auf Temperaturfits.

bezüglich der Siededichte lag allerdings noch im Bereich von 2–3%. Weder das Fletcher-Reeves- noch das Polak-Ribière-Verfahren konnten optimale Ergebnisse bezüglich beider Eigenschaften erzielen. Das beste Ergebnis von 1.60% bezüglich Dampfdruck und 0.84% bezüglich Siededichte, lieferte nach fünf Iterationen und 39 Simulationen eines von zehn Replikaten des Polak-Ribière-Verfahrens. Der zugehörige Parametervektor war $x^{(0)}_{\text{Stoll,TR}} := (0.3699, 0.3133, 0.0239, 0.1879)^T$, welcher als Startvektor für die Variante des Verfahrens nach Stoll und das exakte Trust-Region-Verfahren basierend auf Temperaturfits verwendet wurde.

Tabelle 4.12 zeigt die Ergebnisse von zehn statistisch unabhängigen Zufallsreplikaten der beiden Verfahren: Anhand der MAPE-Werte bezüglich ρ_l, der sehr geringen Standardabweichungen sowie der Veränderungen der Kraftfeldparameter läßt sich feststellen, daß beide Algorithmen robust näher an das Minimum gelangten als GROW: Zum einen benötigte das Trust-Region-Verfahren jedoch deutlich mehr Simulationen als das Verfahren nach Stoll, da die Hesse-Matrix nie effizient berechnet wurde, und zum anderen waren die Veränderungen in Bezug auf die Kraftfeldparameter eher gering, so daß hier nicht definitiv ausgeschlossen werden kann, daß es sich um Zufallsergebnisse handelt. Die hohe Robustheit spricht allerdings dagegen. In Tabelle 4.13 ist die Anzahl an Iterationen und Funktionsevaluationen, sprich Simulationen, für den gesamten Kombinationsalgorithmus und die darin enthaltenen einzelnen Verfahren angegeben. Da GROW in den meisten Fällen nicht signifikant näher an das Minimum gelangte, nachdem das Abbruchkriterium $F(x) \leq 0.015$ erreicht war, sind die hierzu benötigten durchschnittlichen 46 Simulationen nicht zu werten. Da jedoch die Möglichkeiten von GROW ausgeschöpft werden mußten, um wirklich sicherzugehen, daß das Verfahren nach Stoll und das Trust-Region-Verfahren wirklich näher an das Minimum gelangten als GROW, war das Einzelreplikat mit 39 Simulationen auf jeden Fall erforderlich.

Es konnte gezeigt werden, daß die hier verwendeten beiden Verfahren für die vorliegende Problemstellung geeignet sind und ein robustes Verhalten aufweisen, was jedoch insbesondere an

der Verwendung der Temperaturfits liegt. Diese führen zu einer glatteren Fehlerfunktion, wie oben bereits motiviert wurde. Es kann daher hier nicht geschlossen werden, ob die Verfahren tatsächlich geeignet sind. Außerdem wurde mit $x^{(0)}_{\mathrm{Stoll,TR}}$ durch Zufall auch ein sehr guter Parametervektor mit GROW erreicht. Zwar sind die beiden hier betrachteten Verfahren robust in einen anderen Bereich gelangt, wo kleinere Fehlerfunktionswerte erreicht wurden, allerdings reicht die hier durchgeführte Analyse nicht aus, um endgültige Schlußfolgerungen zu ziehen. Eine Anwendung auf Molekulare Simulationen ist somit unentbehrlich und wird in Abschnitt 5.2.2 nachgeholt. Es kann jedoch folgendes geschlossen werden: Beide Verfahren sind bei glatter zu minimierender Funktion in der Nähe des Minimums einsetzbar. Sollte sich herausstellen, daß beide Verfahren näher an das Minimum gelangen als GROW, so ist die Variante des Verfahrens nach Stoll dem Trust-Region-Verfahren vorzuziehen, da letzteres bei dieser Untersuchung dreimal so viele Simulationen benötigte, es sei denn, es ist in der Nähe des Minimums stets eine effiziente Berechnung der Hesse-Matrix gemäß Abschnitt 3.6.3 möglich. Dies ist jedoch aufgrund des strengeren Normkriteriums (3.101) auch bei Molekularen Simulationen eher unwahrscheinlich.

5 Anwendung der eingesetzten Optimierungsmethoden auf Molekulare Simulationen

In diesem Kapitel werden alle gradientenbasierten Optimierungsverfahren, die sich in Kapitel 4 als geeignet herausgestellt haben, auf Molekulare Simulationen angewandt. Da hier stets zu verschiedenen Temperaturen optimiert wird, ist eine Parallelisierung der Simulationen erforderlich. Ein sequentieller Ablauf und die Optimierung zu einer Temperatur sind jedoch *a priori* nicht ausgeschlossen.
In Abschnitt 5.1 werden Kraftfeldparameter für Benzol, Phosgen, Methanol und Kohlenstoffdisulfid bestimmt. Es handelt sich dabei um vier chemisch sehr unterschiedliche Moleküle, deren Simulationen bis zur Äquilibrierung mit der in Abschnitt 3.6 angegebenen Prozessorleistung nur etwa zwei bis vier Stunden in Anspruch nehmen, deren Kraftfeldparameter allerdings aufgrund spezifischer Eigenschaften nicht trivial zu optimieren sind. Startparameter wurden dabei jeweils aus der Literatur genommen, da für derartige Moleküle Standard-Kraftfelder existieren. In jedem der Unterabschnitte werden verschiedene Optimierungsprobleme betrachtet.
Da es nicht für jedes Optimierungsproblem möglich ist, geeignete Startparameter in der Literatur zu finden, wird in Abschnitt 5.2 mit CMA-ES (siehe Abschnitt 3.3.2) ein globaler Voroptimierer auf Molekulare Simulationen angewandt. Weiterhin ist die Kombination von CMA-ES mit GROW praktisch zu evaluieren, was anhand der Korrelationsfunktionen bereits in Abschnitt 4.5.1 durchgeführt wurde. In der Nähe des Minimums werden die Variante des Verfahrens nach Stoll aus Abschnitt 3.6.5 sowie das Trust-Region-Verfahren mit exakter Lösung des Teilproblems aus Abschnitt 3.4.3 evaluiert und miteinander verglichen, so daß geeignete Algorithmen für jeden Schritt innerhalb des in Abschnitt 4.4 dargestellten Filterungsprozesses vorhanden sind.
Eine weitere Kombination von Optimierungsalgorithmen wird anhand des sehr toxischen Moleküls Ethylenoxid in Abschnitt 5.3 ausgeführt. Dabei werden nicht nur approximative, sondern genaue VLE-Simulationen verwendet. Zwar sind für Ethylenoxid Standard-Kraftfelder aus der Literatur vorhanden, allerdings ist am Fraunhofer-Institut SCAI mit dem in Abschnitt 3.3.3 beschriebenen Optimierungstool DesParO eine globale Voroptimierung durchgeführt worden, welche in Maaß u. a. (2010) veröffentlicht ist. Als Simulationstool wurde dabei das MC-Program *Towhee* (Towhee, 2011) eingesetzt, mit welchem koexistierende flüssige und gasförmige Phasen im Gibbs-Ensemble simuliert wurden. Somit eignet sich Ethylenoxid hervorragend für eine Kombination zwischen DesParO und GROW. Die vorgeschaltete DesParO-Optimierung wurde in dieser Arbeit mit *ms2* (siehe Anhang F.3) durchgeführt. Dieses basiert auf einer NPT-Simulation der Flüssigkeit und der anschließenden Simulation eines großkanonischen Ensembles in der Gasphase (vergleiche Anhang A.5).
In Abschnitt 5.4 werden Optimierungen für Ionische Flüssigkeiten, als Beispiel einer komplexen,

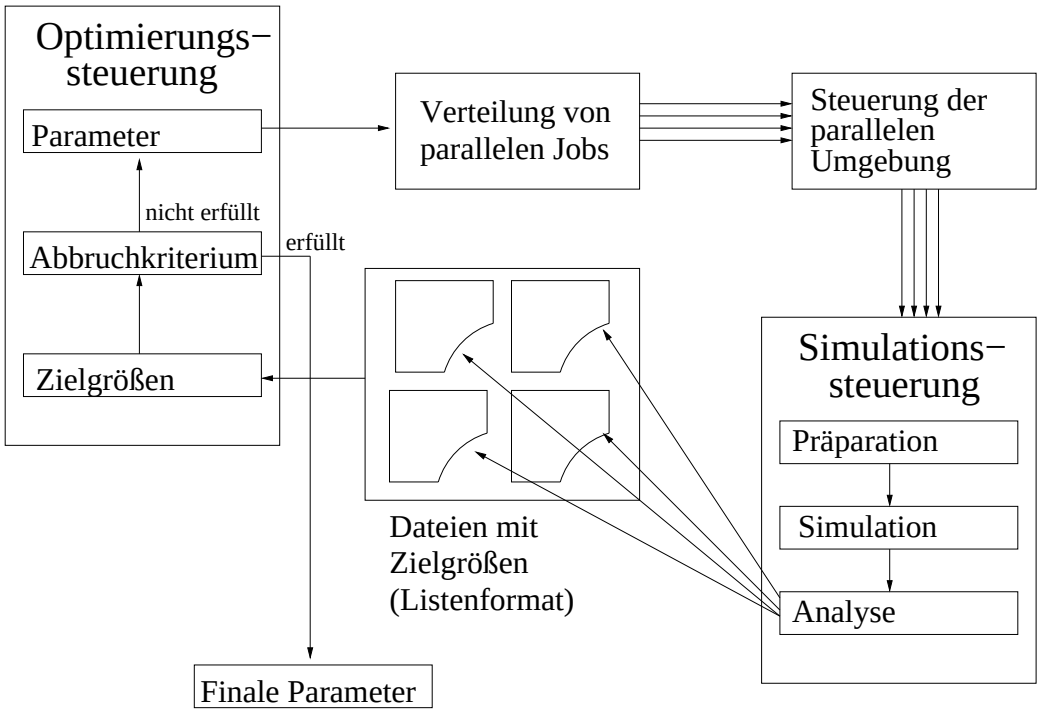


Abbildung 5.1: GROW als Schnittstelle zwischen Optimierung und Simulation im Falle einer Parallelisierung über verschiedene Temperaturen: Auf der linken Seite wird die Optimierung selbst gesteuert. Ist das Abbruchkriterium nicht erfüllt, werden die aktuellen Kraftfeldparameter einem Shell-Skript übergeben, welches die einzelnen Simulationen zu verschiedenen Temperaturen parallelisiert. Nach der Ausführung eines Kontrollskripts für die parallele Umgebung wird ein Simulationskontrollskript aufgerufen, welches sämtliche Vorbereitungen, die einzelnen Simulationen selbst sowie die Berechnung der notwendigen Systemeigenschaften durchführt. Letztere werden in separate Dateien geschrieben, welche wiederum vom Steuerskript der Optimierung eingelesen werden. Nach der Auswertung der Fehlerfunktion und der Überprüfung des Abbruchkriteriums wird der Ablauf fortgesetzt.

größeren Molekülklasse, durchgeführt, welche sowohl industriell als auch wissenschaftlich interessant sind und innerhalb der Arbeitsgruppe *Computational Chemistry Engineering (CoChE)* des Fraunhofer-Instituts SCAI von sehr hoher Bedeutung sind.
In Abschnitt 5.5 werden schließlich die hier erhaltenen Kraftfelder für Methanol und Epoxid validiert, das heißt auf andere physikalische Eigenschaften angewandt, um die allgemeine Verwendbarkeit eines auf bestimmte Zielgrößen angepaßten Kraftfelds sowohl aus physikalischer als auch aus mathematischer Sicht zu diskutieren.

Die meisten in diesem Kapitel beschriebenen Simulationen wurden mit *Gromacs* durchgeführt (siehe Anhang F.1). Es handelte sich um MD-Simulationen in einem NPT-Ensemble. Transporteigenschaften und intermolekulare Kräfte wurden mit *Moscito* (siehe Anhang F.2) berechnet, da dies mit *Gromacs* entweder nicht möglich oder zu kompliziert war. Da die meisten experimtellen physikalischen Eigenschaften jedoch VLE-Daten sind, handelt es sich bei den mit *Gromacs* und *Moscito* ermittelten Zielgrößen lediglich um Approximationen (siehe Anhang A.5). Wurde eine anderes Simulationstool verwendet, so ist dies im folgenden explizit angegeben. Es ist zu beachten, daß in dieser Arbeit mit den Begriffen *Diffusion* beziehungsweise *Diffusionskoeffizient*

stets der Selbstdiffusionskoeffizient gemeint ist.

Bevor die Ergebnisse im einzelnen vorgestellt werden, wird kurz auf die technische Realisierung der Kraftfeldparametrisierung eingegangen, welche für alle in diesem Kapitel dargestellten Optimierungsläufe relevant ist:

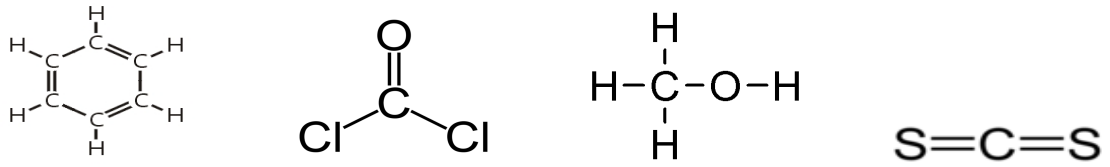
Technische Realisierung der Optimierungsprozesse Der Gesamtablauf des Optimierungsprozesses mit parallelen Simulationen ist in Abbildung 5.1 veranschaulicht: Die Optimierungssteuerung ist auf der linken Seite von Abbildung 5.1 gezeigt. Die aktuellen Kraftfeldparameter werden an die Simulationssteuerung übergeben, wobei vorher ein Verteilungsskript die Parallelisierung über die verschiedenen Temperaturen organisiert. Dieses ruft wiederum ein Kontrollskript für die parallele Umgebung auf, welches dann das eigentliche Steuerskript für die Simulationen startet. Alle genannten Skripte sind in *shell (bash)* geschrieben. Das Steuerskript für die Simulationen regelt den gesamten in Abschnitt 3.5.4 angegebenen Simulationsverlauf, das heißt, es steuert mehrere aufeinanderfolgende, voneinander abhängige Molekulare Simulationen. Es führt alle notwenigen Präparationsroutinen aus, die Simulationen selbst, das heißt die Erstellung der Trajektorien, sowie innerhalb einer Analyse der Simulationsergebnisse die Berechnung und Speicherung der Systemeigenschaften. Letztere werden in Form von Listen gespeichert, das heißt, zu jedem Simulationsschritt (zum Beispiel alle tausend Zeitschritte bei MD-Simulationen und nach jeder tausendsten Stichprobenentnahme bei MC-Simulationen) wird ein Wert herausgeschrieben. Diese Listendateien werden von der Optimierungsssteuerung eingelesen, die Fehlerfunktion wird ausgewertet, das Abbruchkriterium überprüft und der Optimierungsprozeß fortgeführt.
Die Simulationen selbst werden ebenfalls auf verschiedenen Prozessoren parallel ausgeführt, und zwar mit der Standard-Kommunikationsschnittstelle *MPI* (siehe Anhang G.8). Die Parallelisierung erfolgt dabei im Falle von MD-Simulationen über die in Anhang E.1 angesprochenen Nachbarschaftslisten, und im Falle von MC-Simulationen werden einfach P unabhängige Simulationsläufe parallel ausgeführt, wobei P die Anzahl an verwendeten Prozessoren ist, und deren Ergebnisse gemittelt. Ist M die Gesamtanzahl an MC-Schritten, so werden auf jedem Prozessor $\frac{M}{P}$ MC-Schritte durchgeführt.

5.1 Verfahrenstests: Benzol, Phosgen, Methanol und Kohlenstoffdisulfid

Dieser Abschnitt zeigt die Ergebnisse verschiedener mit GROW durchgeführter Optimierungsläufe für Benzol, Phosgen, Methanol und Kohlenstoffdisulfid. Die Eigenschaften und Besonderheiten dieser Moleküle sowie die Schwierigkeiten, die sowohl bei den Simulationen als auch bei den Optimierungen entstehen, werden in Abschnitt 5.1.1 erläutert. Die durch die einzelnen Optimierungsprozesse erhaltenen Kraftfelder sind sowohl mathematisch als auch physikalisch und chemisch zu bewerten. Insbesondere stellt sich dabei die Frage, inwieweit die Kraftfelder reproduzierbar und eindeutig sind. Dabei spielen das statistische Rauschen, die zu optimierenden Kraftfeldparameter und Zielgrößen sowie die davon abhängige Gestalt der Fehler-

funktion eine Rolle. Auch die Struktur der Optimierung ist von erheblicher Bedeutung, das heißt ob sämtliche Kraftfeldparameter oder Zielgrößen simultan oder sukzessive in der Optimierung verwendet werden. All diese Fragestellungen sollen hier anhand der ersten Anwendungen von GROW auf Molekulare Simulationen geklärt werden. Dies ist möglich, da es sich hierbei um kleine Moleküle handelt, deren Simulationen nicht allzu aufwendig sind, so daß gewisse Optimierungsanalysen durchführbar sind. Bei den Simulationen in diesem Abschnitt handelt es sich um MD-Simulationen im NPT-Ensemble, ausgeführt mit *Gromacs*. Als Integrator wurde der Bocksprung-Verlet-Algorithmus aus Abschnitt 2.3.2 benutzt. Es wurden stets die in Abschnitt 3.5.4 beschriebenen Teilsimulationen verwendet, um möglichst exakte thermodynamische Durchschnitte zu erhalten. Die Länge der Äquilibrierung und Produktion war dabei auch teilweise abhängig vom aktuellen Fehlerfunktionsbereich: Zu Beginn der Optimierung können kürzere Simulationszeiten in Kauf genommen werden, da GROW gemäß Abschnitt 3.5.5 mit einem gewissen Ausmaß an statistischem Rauschen umgehen kann. Die Anzahl an Zeitschritten und Zeitschrittweiten sind für alle Teilsimulationen in den entsprechenden Unterabschnitten angegeben. Kurzreichweitige intermolekulare Wechselwirkungen wurden mithilfe des (12,6)-Lennard-Jones-Potentials aus Abschnitt 2.2.2 zusammen mit den Lorentz-Berthelot-Mischungsregeln (Gleichung (2.23)) beschrieben. Bei sämtlichen Simulationen erfolgte die Berechnung der elektrostatischen Kräfte mithilfe der in Anhang E.2 angesprochenen Ewald-Summation. Der langreichweitige Term wurde dabei durch eine Fast-Fourier-Transformation bestimmt, so daß die Ladungsdichte im Fourierraum diskret zu evaluieren war (sogenannte *Particle-Mesh-Ewald*-Methode). Als Abschneideradien wurden $r_N = r_C = 0.9$ nm (siehe Anhang E.1) sowohl für die Lennard-Jones- als auch für die Coulomb-Wechselwirkungen verwendet. Im Falle der Prä-Prä-Äquilibrierung, wo ausgeschlossen werden mußte, daß das System mechanisch instabil wurde und daher in einem festen Volumen genauer simuliert werden mußte, wurde $r_N = r_C = 1.2$ nm gesetzt.
Zum Erhalt starrer Bindungen und Winkel wurden drei Iterationen des sogenannten *LINCS-Algorithmus* verwendet (siehe Anhang B). Es handelt sich dabei um ein Verfahren dritter Ordnung zur Lösung der Newtonschen Bewegungsgleichung mit Nebenbedingungen. Werden von diesem Verfahren mehr als 30 Warnungen ausgegeben, so liegt eine schlechte Konvergenz vor, so daß die Nebenbedingungen gar nicht oder nur schwer erfüllbar sind. In diesem Fall ist davon auszugehen, daß das System trotz Prä-Prä-Äquilibrierung mechanisch instabil geworden ist, da nur noch sehr schwache intermolekulare Kräfte vorhanden sind. Die Simulation wurde dann abgebrochen, und die Einstellungen waren erneut zu überprüfen.
Zur Durchführung der MD-Simulationen bei konstanter Temperatur wurde ein Nosé-Hoover-Thermostat mit Kopplungsparameter $\tau_T = 0.5$ eingesetzt (vergleiche Anhang A.1). Für MD-Simulationen bei konstantem Druck wurde ein Parinello-Rahman-Barostat mit einer Kopplungszeit von 2 fs eingesetzt (vergleiche Anhang A.2). Initiale Geschwindigkeiten wurden stets per Zufallsprinzip festgesetzt.

Im Falle von Benzol (Abschnitt 5.1.2) wurden verschiedene physikalische Zielgrößen betrachtet, zunächst Siededichte und Diffusion und dann Siededichte und Verdampfungsenthalpie. Bei Phosgen (Abschnitt 5.1.3) wurden die Zielgrößen Siededichte und Verdampfungsenthalpie auch sukzessive optimiert. Weiterhin wurden die effizienten Gradienten- und Hesse-Matrix-Berechnungen aus den Abschnitten 3.6.2 und 3.6.3 angewandt. Für Methanol (Abschnitt 5.1.4) wurden auch Partialladungen mitoptimiert, und es wurde eine temperaturbasierte Kreuzvalidierung durchgeführt. Kohlenstoffdisulfid schließlich (Abschnitt 5.1.5) zeigt die Grenzen der



(a) Benzol (b) Phosgen (c) Methanol (d) Kohlenstoffdisulfid

Abbildung 5.2: Strukturformeln von Benzol (Wikipedia, 2011b) (a), Phosgen (Wikipedia, 2011c) (b), Methanol (Wikipedia, 2011d) (c) und Kohlenstoffdisulfid (Wikipedia, 2011e) (d).

Kraftfeldoptimierung auf: Spiegelt das molekulare Modell die Realität nicht auf geeignete Art und Weise wider, so ist kein optimales Kraftfeld zu erwarten.

5.1.1 Allgemeines zu Benzol, Phosgen, Methanol und Kohlenstoffdisulfid

Bei Benzol, Phosgen, Methanol und Kohlenstoffdisulfid handelt es sich um kleine Moleküle (Abbildung 5.2), die mit nicht allzu großem Rechenaufwand zu simulieren sind. Allerdings weist jedes der vier Teilchen gewisse Herausforderungen auf, die die Erstellung des molekularen Modells erschweren. Dies wirkt sich ebenfalls auf die Optimierung aus, denn ein strukturell ungeeignetes molekulares Modell kann nicht zielführend parametrisiert werden. Im folgenden werden die Substanzen im einzelnen vorgestellt und die Besonderheiten sowie Problematiken für Simulation und Optimierung hervorgehoben.

Benzol Benzol (Abbildung 5.2(a)) ist eine farblose, klare und bei Raumtemperatur flüssige sowie leicht flüchtige Substanz. Abbildung 5.3 zeigt die Molekülgeometrie: Benzol besteht aus sechs ringförmig angeordneten Kohlenstoff- und sechs Wasserstoffatomen, woraus sich die Summenformel C_6H_6 ergibt. Durch die Delokalisierung der π-Orbitale im Kohlenstoffring sind alle C–H- und C–C-Bindungen äquivalent und gleich lang, woraus sich ein regelmäßiges Sechseck und eine hohe Symmetrie ergibt. Benzol gehört zu der Gruppe der aromatischen Kohlenwasserstoffe und ist der Grundbaustein für viele Aromaten. Es ist toxisch und krebserregend. Da es im Körper zu Ethylenoxid oxidiert wird, kann das Erbgut beschädigt werden. Weiterhin ist das Molekül brennbar und weist eine geringere Viskosität als Wasser auf. Es hat eine Schmelztemperatur von 5.5 °C und eine Siedetemperatur von 80.1 °C. Chemisch gesehen ist Benzol quadrupolar, verbrennt mit gelber Flamme unter Rußentwicklung zu Wasser und Kohlenstoffdioxid, hat einen charakteristischen Geruch und eine besonders hohe aromatische Stabilität. Dadurch wird im Gegensatz zu Cyclohexan und Cyclohexadien kein Bromwasserstoff addiert. Benzol kommt in Steinkohleteer und Erdöl vor. Im Rauch einer Zigarette und bei Vulkanausbrüchen entstehen geringe Mengen an Benzoldampf. Hergestellt werden kann es durch Dampfspaltung, wobei Hexan über Cyclohexan zu Benzol dehydriert wird. In der Kraftstoffindustrie wird es zur Erhöhung der Klopffestigkeit von Benzin angewandt und in der chemischen Industrie zur Synthese von Kunststoffen, Kautschuk, Nylon, Insektiziden und Farbstoffen. Früher wurde es als Lösungsmittel für Wachse, Harze und Öle eingesetzt. Da es in Deutschland, außer in Laboratorien, nur in einer Konzentration von bis zu 0.1% zulässig ist, wurde Benzol als Lösungsmittel durch weniger gesundheitsschädliche Stoffe wie Toluol und Aceton ersetzt.

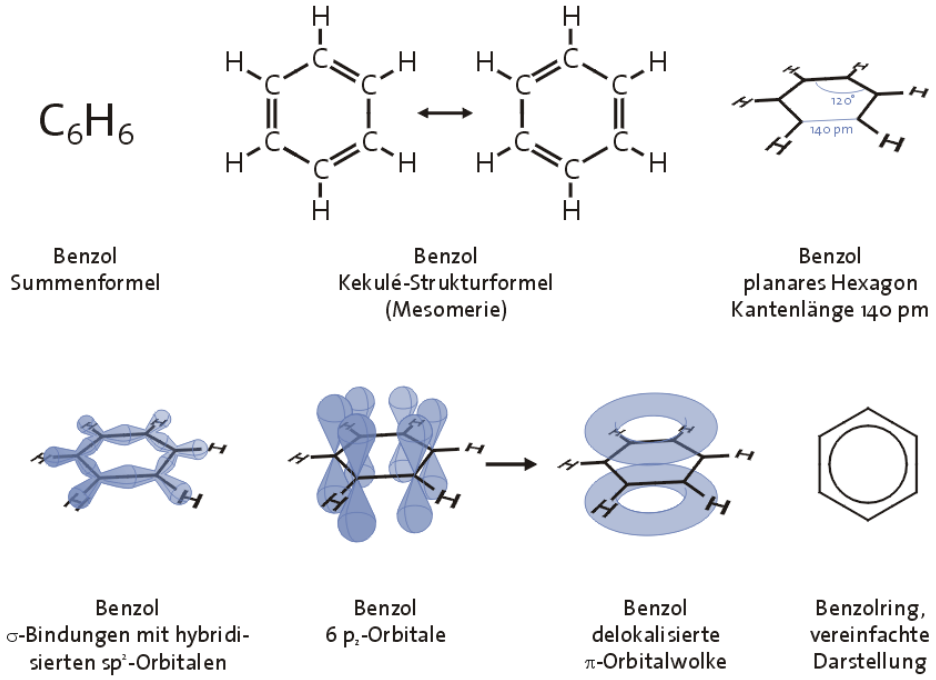
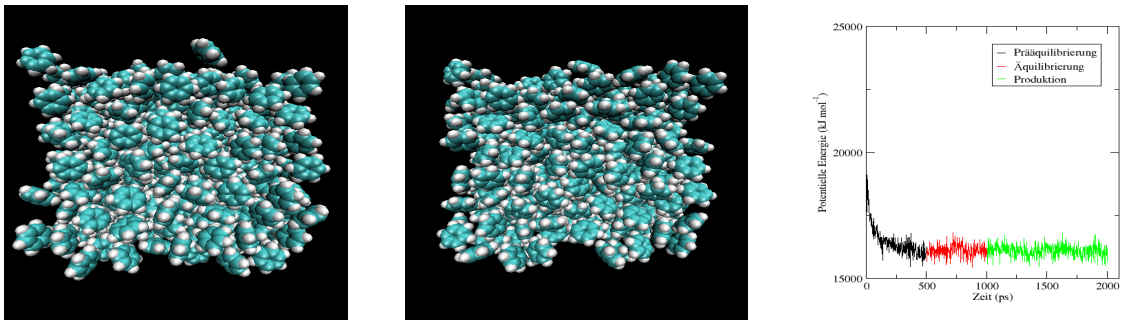


Abbildung 5.3: Molekülgeometrie von Benzol (Wikipedia, 2011b): Das Molekül besteht aus sechs in einem Ring angeordneten Kohlenstoffatomen, die jeweils an ein Wasserstoffatomen einfach gebunden sind (Summenformel: C_6H_6). Es gibt verschiedene Darstellungsmöglichkeiten der Benzolstruktur: Die Kekulé-Strukturformel sieht zwei mesomere Geometrien vor, bestehend aus drei Einfach- und drei Doppelbindungen im Kohlenstoffring. Jedes Kohlenstoffatom ist sp^2-hybridisiert, woraus σ-Bindungen enstehen. Die verbleibenden sechs p^z-Orbitale bilden eine über den gesamten Ring delokalisierte π-Orbitalwolke, die vereinfacht durch ein planares Sechseck mit Kreis dargestellt werden kann. Die Delokalisierung führt zu sechs gleich langen C–C-Bindungen der Länge 140 pm und somit zu einer regelmäßigen Sechsechsstruktur. Bei der Simulation sind zwar Verzerrungen erlaubt, allerdings wird vorwiegend das planare Hexagon favorisiert.

Da Benzol eine symmetrische Struktur aufweist, quadrupolar ist und nur zwei Atomtypen besitzt, ist die Simulation und Kraftfeldparametrisierung relativ einfach. Nichtsdestotrotz ist die Erstellung eines geeigneten molekularen Modells keineswegs trivial. Schwierigkeiten entstehen durch die delokalisierten π-Elektronen. Bei der Simulation wurden sechs Punktladungen für den Kohlenstoffring verwendet, allerdings ist auch ein Quadrupol im Massenzentrum denkbar. Weiterhin wird das Modell durch die Existenz von 10 Winkeln und 30 Diederwinkeln komplexer. Es ist zu beachten, daß auch in *Gromacs* Dipole beziehungsweise Quadrupole zumindest approximativ modellierbar sind: In Engin u. a. (2011) wurden ein Dipol mithilfe von zwei entgegengesetzten Punktladungen $+q$ und $-q$ und ein Quadrupol mithilfe von drei Punktladungen $+q$, $-2q$ und $+q$ angenähert. Allerdings wäre hierzu ein weiterer zu optimierender Parameter vonnöten, und zwar der Abstand d zwischen den Punktladungen. In der oben genannten Veröffentlichung konnte gezeigt werden, daß bei zu großem d die Appriximation zu schlecht ist und bei zu kleinem d numerische Probleme bei der Berechnung des Dipol- beziehungsweise Quadrupolmoments auftreten. Daher wird in dieser Arbeit von einer derartigen Approximation abgesehen, und Dipol- beziehungsweise Quadrupolmodelle werden in diesem Kapitel stets mit dem Softwarepaket *ms2* (siehe Anhang F.3) simuliert.

Phosgen Phosgen (Abbildung 5.2(b)), auch bekannt unter den Namen Kohlenoxiddichlorid oder Carbonylchlorid, ist äußerst toxisch und bei Raumtemperatur gasförmig. Die Summenformel lautet $COCl_2$. Das Molekül ist wie Benzol planar und besteht aus einem zentralen Kohlenstoffatom, an das ein Sauerstoffatom doppelt und zwei Chloratome einfach gebunden ist. Die Länge der C–O-Bindung beträgt 118 pm und die der C–Cl-Bindungen 174 pm. Es besteht eine Achsensymmetrie bezüglich der C–O-Achse, und der Winkel zwischen den Chloratomen beträgt 111.8 °. Es schmilzt bei -128°C und siedet bei 7.5 °C.
Phosgen ist ein Derivat der Kohlensäure, nämlich das Dichlorid. Es ist schlecht wasserlöslich und kann somit beim Einatmen bis in die Lungenbläschen gelangen, wo es einerseits die Sauerstoffzufuhr blockiert und andererseits mit Wasser langsam zu Kohlenstoffdioxid und Salzsäure reagiert. Letztere verätzt das Lungengewebe und führt somit zum Tod. In geringer Konzentration weist Phosgen einen süßlichen Geruch auf, und in hoher Konzentration hat es einen charakteristischen faulen Obstgeruch. Es löst sich gut in organischen Lösungsmitteln.
Phosgen wird industriell aus Kohlenstoffmonoxid und Chlorgas hergestellt. Die chemische Reaktion ist stark exotherm und muß somit drastisch gekühlt werden, was die Herstellung kostenintensiv und aufwendig macht. Eine andere Möglichkeit ist die Synthese aus Tetrachlormethan und rauchender Schwefelsäure. Verwendet wurde Phosgen als militärischer Gaskampfstoff, vor allem im Ersten Weltkrieg. Heute findet es vor allem Verwendung als Grundbaustein für Carbonsäurechloride, aus denen Polyurethane und Polycarbonat-Kunststoffe synthetisiert werden, die wiederum für die Herstellung von Medikamenten, Farbstoffen und Insektiziden eingesetzt werden.
Die Schwierigkeit bei der Modellierung von Phosgen liegt in der Polarität und der hohen Polarisierbarkeit des Moleküls. Das bedeutet, es kann erwartet werden, daß die Interaktionsfläche temperaturabhängig ist. Somit müßten je nach Temperatur andere Partialladungen gewählt werden. Ob ein generisches, temperaturunabhängiges Kraftfeld für Phosgen erhalten werden kann, welches die Realität gut wiedergibt, bleibt daher abzuwarten.

Methanol Methanol (Abbildung 5.2(c)) ist bei Raumtemperatur eine farblose, klare und leicht flüchtige Flüssigkeit. Die Summenformel lautet CH_4O oder genauer CH_3OH, das heißt, eine Methylgruppe ist mit einer Hydroxygruppe verbunden. Daher ist Methanol der am einfachsten strukturierte Alkohol. Die Substanz ist aufgrund des elektronegativen Sauerstoffs polar und somit sehr gut wasserlöslich. Der Schmelzpunkt liegt bei -98 °C und der Siedepunkt bei 65 °C. Methanol ist leicht entzündlich und verbrennt mit einer schwach blauen Flamme. Bei einer Konzentration von 6–50% bilden Methanoldämpfe mit Luft explosionsfähige Gemische. Weiterhin ist Methanol toxisch und kann zur Erblindung führen. In der Natur ist es in zahlreichen Früchten an Methylestern und -ethern chemisch gebunden. Außerdem entsteht es bei der Maischung durch die Hydrolyse eines speziellen Esters mithilfe von Enzymen. Auch Tabakrauch enthält geringe Spuren von Methanoldampf. Industriell wird das Molekül aus Synthesegas hergestellt. Dabei handelt es sich um ein 1:2-Gemisch von Kohlenstoffmonoxid und Wasserstoff. Verwendet wird es insbesondere als Energieträger und Wasserstofflieferant. Außerdem können aus Methanol zahlreiche andere industriell bedeutende chemische Substanzen gewonnen werden, wie zum Beispiel Essigsäure oder Formaldehyd, ein Edukt für Harze. Auch als Kraftstoff ist Methanol gut geeignet. Aufgrund seiner Giftigkeit wird es jedoch als Kraftstoffzusatz in lediglich geringer Konzentration verwendet.
Methanol ist aufgrund seiner hohen Polarität und Wasserstoffbrückenbindungen ein interessan-



(a) (b) (c)

Abbildung 5.4: Startkonfiguration (a), äquilibrierte Konfiguration (b) und Äquilibrierung der potentiellen Energie (c) im Falle von Benzol. Die Kantenlänge der Box beträgt 40 nm.

tes Testbeispiel. Es ist zu untersuchen, ob die Optimierung von LJ-Parametern alleine zu einem geeigneten Methanol-Kraftfeld führen kann, da diese keine Dipol-Dipol-Wechselwirkungen beschreiben. Es ist sehr wahrscheinlich, daß die Partialladungen in die Optimierung miteinbezogen werden müssen.

Kohlenstoffdisulfid Kohlenstoffdisulfid (Abbildung 5.2(d)) ist bei Raumtemperatur eine farblose, klare und leicht flüchtige Flüssigkeit. Die Summenformel lautet CS_2. Ein Kohlenstoff ist dabei von zwei doppelt gebundenen Schwefelatomen umgeben. Die Molekülstruktur ist linear. CS_2 ist hochentzündlich und leicht fettlöslich. Dies führt dazu, daß es durch Lunge und Haut aufgenommen werden und zu einer Vielzahl von Vergiftungserscheinungen führen kann. Da Kohlenstoffdisulfid eine positive Standardbildungsenthalpie aufweist, ist die Synthese aus den Elementen eine endotherme Reaktion. Daher kann es industriell durch die Leitung von Schwefeldämpfen über glühende Holzkohle gewonnen werden. Eine weitere Möglichkeit besteht darin, Erdgas mit Schwefel katalytisch reagieren zu lassen. Der dabei entstehende Schwefelwasserstoff ist ebenfalls industriell verwendbar. Genutzt wird CS_2 vor allem bei chemischen Teilreaktionen wie zum Beispiel bei der Herstellung von Zellulosefasern.
Die Schwierigkeiten bei der Simulation von Kohlenstoffdisulfid sind ähnlich wie bei Benzol: Ein einfaches Stäbchenmodell wird aufgrund der Doppelbindungen nicht ausreichend sein. Es ist abzuwägen, ob ein Quadrupolmodell besser geeignet ist.

5.1.2 Benzol: Verschiedene Arten von Zielgrößen

Für Benzol wurde ein relativ einfaches molekulares Modell angenommen, bestehend aus sechs ringförmig angeordneten Kohlenstoffatomen, die jeweils an ein Wasserstoffatom gebunden sind. Zwei Simulationsboxen sind in Abbildung 5.4 dargestellt. Ein zusätzlicher Quadrupol im Massenzentrum ist ebenfalls denkbar, allerdings ist dies mit *Gromacs* nicht simulierbar. Die delokalisierten π-Elektronen können mit einem derartigen Modell allerdings nicht beschrieben werden. Lediglich verschiedene Bindungslängen für C−C und C=C unterscheiden Einzel- von Doppelbindungen. Die Bindungen wurden mithilfe des LINCS-Algorithmus (siehe Anhang B) starr gehalten. Die Bindungslängen stammen aus dem OPLS-Kraftfeld von Jorgensen u. a. (1996).

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	C, H	*All-Atom*-Modell
	Dipole/Quadrupole	–	auch möglich: zentraler Quadrupol
	Molare Masse	78.12 g/mol	m(C) = 12.01 g/mol, m(H) = 1.01 g/mol
	Molekülstruktur	verzerrbarer Sechsring	3 π-Bindungen im Ring (delokalisiert)
	Intramolekulares Kraftfeld	12 starre Bindungen (LINCS) 10 Winkel 30 Diederwinkel	Quelle: OPLS (Jorgensen u. a., 1996)
	Ladungen	12 Punktladungen	Quelle: OPLS (Jorgensen u. a., 1996) oder werden mitoptimiert
	Besonderheiten	1,4-Wechselwirkungen exkludiert	
Simulationsbox	Startkonfiguration	randomisiert	siehe Abbildung 5.4(a)
	Anzahl Moleküle	1000	
	Kantenlänge	40 nm	
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung	10000	Schrittweite: 0.2 fs
	Prääquilibrierung	250000	Schrittweite: 2 fs
	Äquilibrierung	250000	Schrittweite: 2 fs
	Produktion	500000	Schrittweite: 2 fs
Optimierungsrelevantes	zu optimierende Parameter	σ(C), (σ(H),) ε(C), (ε(H),) (q(C))	Quelle Startwerte: OPLS (Jorgensen u. a., 1996)
	Temperaturen	303, 313, 323	in K
	Dampfdrücke	0.16, 0.25, 0.32	in bar (aus Antoine-Gleichung (5.1))
Ausgaben:			
Zielgrößen	experimentell	D, ρ_l	Quellen: D, ρ_l: Literatur (Yoshida u. a., 2008)
		$\Delta_v H$, ρ_l	$\Delta_v H$: NIST-Datenbank (NIST, 2011)
	simuliert	D: Gleichung (C.18)	Einstein-Darstellung
		ρ_l: Gleichung (C.8)	
		$\Delta_v H$: Gleichung (C.3)	Approximation (ideales Gas)
	Toleranzen	D: unbekannt	
		ρ_l: 0.5%	
		$\Delta_v H$: 1%	

Tabelle 5.1: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Benzol.

Winkel- und Diederpotentiale wurden gemäß dem Gromos-Potential aus Gleichung (2.35) definiert. Die intramolekularen Potentialparameter wurden ebenfalls aus dem OPLS-Kraftfeld entnommen. Dadurch, daß Winkel- und Diederwinkel variieren konnten, wurde als molekulares Modell ein verzerrbarer Sechsring mit starren Bindungen verwendet. Weiterhin wurden zwölf Punktladungen angenommen, sechs gleich große positive Partialladungen für Kohlenstoff und sechs für Wasserstoff. Sämtliche 1,4-Wechselwirkungen wurden exkludiert.

Simulations- und Optimierungseinstellungen Unterschieden wurden zwei verschiedene Optimierungsläufe: Zunächst wurden Diffusion und Siededichte an der Phasengrenzkurve zwischen

Flüssigkeit und Gas an experimentelle Werte angepaßt. Die Optimierung wurde bereits in Hülsmann u. a. (2011b) veröffentlicht und wird im folgenden mit (D,ρ_l) markiert. Die aus dieser Optimierung erhaltenen Kraftfeldparameter wurden für eine zweite Optimierung verwendet, bei der Verdampfungsenthalpie und Siededichte betrachtet wurden. Diese wird im folgenden mit ($\Delta_v H$,ρ_l) markiert. Bei der ersten Optimierung waren die anzupassenden Kraftfeldparameter lediglich die Lennard-Jones-Parameter für Kohlenstoff, bei der zweiten wurden die für Wasserstoff und die Partialladungen für Kohlenstoff hinzugefügt. Sämtliche Startparameter stammen wieder aus dem OPLS-Kraftfeld. Simuliert wurde stets an der Phasenübergangskurve zwischen Flüssigkeit und Gas zu den Temperaturen 303, 313 und 323 K. Da es sich um NPT-Simulationen der Flüssigkeit handelte, waren die entsprechenden experimentellen Dampfdrücke (0.16, 0.25 und 0.32 bar) ebenfalls anzugeben. Sie wurden aus der NIST-Datenbank (NIST, 2011) entnommen, laut welcher sie mittels der sogenannten *Antoine-Gleichung*

$$\log_{10}(p_\sigma) = A - \frac{B}{T + C}, \tag{5.1}$$

berechenbar sind. Die Koeffizienten A, B und C stammen dabei aus Deshpande u. Pandya (1967).

Tabelle 5.1 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Benzol. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen und Toleranzen für die simulierten Zielgrößen angegeben.
Abbildung 5.4(a) zeigt die Startkonfiguration der Simulationsbox: In einem Würfel mit einer Kantenlänge von 40 nm wurden 1000 Benzolmoleküle per Zufall so positioniert, daß keine Überlappungen auftraten. Nach einer Energieminimierung und kurzen NVT-Simulation von 10000 Zeitschritten mit kleinerer Zeitschrittweite (Prä-Prä-Äquilibrierung) wurden 250000 Zeitschritte verwendet, um die Moleküle aus ihren originalen Positionen herauszutreiben (Präääquilibrierung). Die Äquilibrierung bestand ebenfalls aus 250000 Zeitschritten. Nach meistens nur einem Äquilibrierungszyklus waren die Äquilibrierungskriterien (3.70) für potentielle Energie, Siededichte und intermolekulare Energie erfüllt. Abbildung 5.4(b) zeigt eine äquilibrierte Konfiguration: Es ist deutlich zu sehen, daß die Moleküle mehr ineinander verschachtelt waren als zu Beginn, was auf die zunehmende Solvatisierung während des Simulationsverlaufs zurückzuführen ist. Abbildung 5.4(c) zeigt die Äquilibrierung der potentiellen Energie: Im Laufe der Präääquilibrierung sank die potentielle Energie drastisch ab, um dann um einen konstanten Mittelwert herum zu oszillieren, welcher auch in der Produktionsphase erhalten blieb.

Optimierungsergebnisse für D und ρ_l Tabelle 5.2 und Abbildung 5.5 zeigen die Optimierungsergebnisse im Falle von D und ρ_l. Verwendet wurde dabei die Methode des steilsten Abstiegs mit $h = 0.01$, $\zeta_A = 0.2$ und $\forall_{i,T}\ w_{i,T} = 1$. Alle Eigenschaften sollten zunächst als gleichwertig betrachtet werden. Der zulässige Bereich war $\Omega := (10, 20)$, das heißt, σ wurde um maximal 10% und ε um maximal 20% verändert. Im allgemeinen ist bei der Änderung von σ eine größere Auswirkung auf das System und die zu berechnenden Zielgrößen zu beobachten. Daher wurde der zulässige Bereich für ε auch im folgenden meist größer gewählt als der für σ. Innerhalb von nur zwei Iterationen konnte die Fehlerfunktion um fast zwei Größenordnungen verkleinert werden, wobei ρ_l im Bereich der statistischen Unsicherheiten gleich blieb und D um

k	$x^{(k)}$	MAPE D	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
		Steilster Abstieg, $h = 0.01$, $\Omega = (10, 20)$				
0	0.3550 0.2929	22.05%	0.81%	0.1463	17.0692 8.2676	18.9660
1	0.3424 0.2868	7.77%	0.85%	0.0205	-9.9735 -6.0080	11.6433
2	0.3447 0.2882	2.15%	0.85%	0.0017	-2.7582 -1.6602	3.2193

Tabelle 5.2: Optimierungsergebnisse für Benzol im Falle von D und ρ_l: Angegeben sind die MAPE-Werte für D und ρ_l, die Fehlerfunktionswerte $F\left(x^{(k)}\right)$, die Gradienten $\nabla F\left(x^{(k)}\right)$ und deren Normen für jede Iteration $x^{(k)}$. Innerhalb von zwei Iterationen mit der Methode des steilsten Abstiegs wurde die Fehlerfunktion um fast zwei Größenordnungen verkleinert. Während die Siededichte im Bereich der statistischen Unsicherheiten gleich blieb, wurde der Diffusionskoeffizient um etwa eine Größenordnung verbessert. Weitere Verbesserungen konnten nicht erzielt werden, auch nicht mit einem CG-Verfahren. Die zu optimierenden Kraftfeldparameter waren σ(C) und ε(C), das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{C})^{(k)}, \varepsilon(\mathrm{C})^{(k)}\right)$.

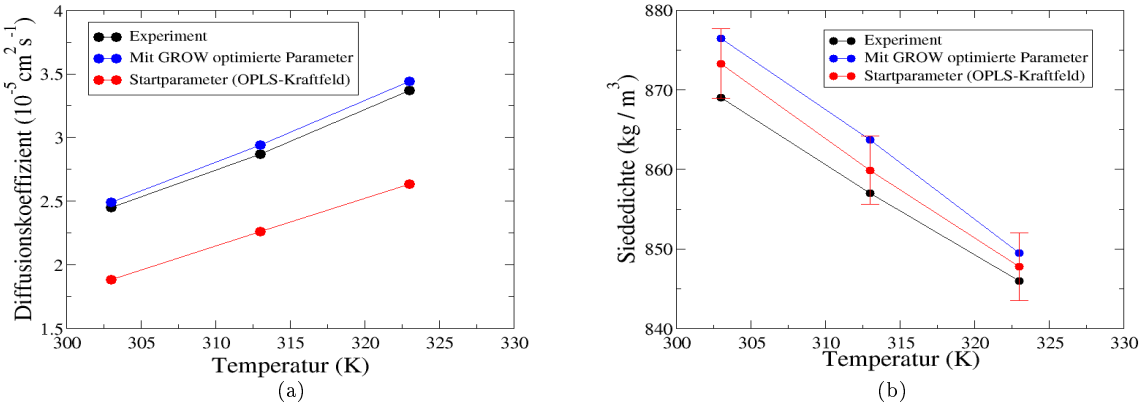


Abbildung 5.5: Optimierung von D (a) und ρ_l (b) im Falle von Benzol. Die Siededichte blieb innerhalb der statistischen Unsicherheiten gleich, wohingegen der Diffusionskoeffizient deutlich verbessert wurde.

etwa eine Größenordnung verbessert wurde. Abbildung 5.6 zeigt die Entwicklung der Fehlerfunktion.

Da bei $k = 0$ und $k = 1$ jeweils nur eine Armijo-Schrittweite erforderlich war, ergab sich die Anzahl an Simulationen für die Optimierung und Gradientenberechnung bei $k = 2$ zu $3 \cdot 3 = 9$. Die maximale Anzahl an Armijo-Schritten wurde dabei auf 10 gesetzt, das heißt, es kamen noch 9 Simulationen hinzu, um feststellen zu können, daß keine Verbesserung mehr möglich war. Es ist dabei zu beachten, daß die Schrittweitensteuerung stets bei $l = 2$ beginnt. Ein anschließendes CG-Verfahren konnte keine Verbesserungen mehr erzielen, weder das FR- noch das PR-Verfahren. Auch die Einstellung $h = 0.001$ führte zu keinem weiteren Erfolg. Aus Abbildung 5.5(a) ist zu erkennen, daß die Diffusionskoeffizienten im Falle der Startparameter aus dem OPLS-Kraftfeld noch sehr stark vom Experiment abwichen, wohingegen die Optimierung Werte für die Diffusionskoeffizienten lieferte, die deutlich näher am Experiment lagen. Auch der Trend der experimentellen Kurve wurde korrekt reproduziert, das heißt, qualitativ gesehen konnte ein gutes Kraftfeld erzielt werden. Da allerdings mithilfe von *Gromacs* keine statistischen Unsicherheiten für D geschätzt werden konnten, ist das Kraftfeld nur schwierig abschließend zu bewerten. Um auszuschließen, daß es sich nur um ein Zufallsergebnis handelt, wurden jedoch zehn Replikate mit der vierfachen Produktionszeit ($2 \cdot 10^6$ Zeitschritte) für die

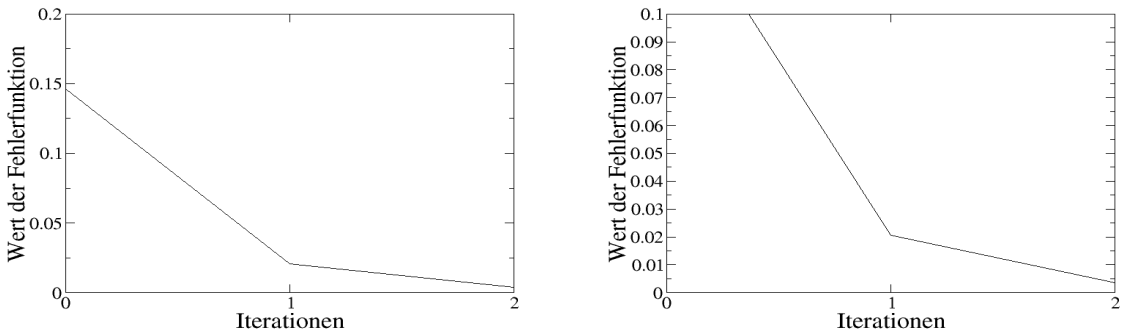


Abbildung 5.6: Entwicklung der Fehlerfunktion im Falle von Benzol (D,ρ_l) unterhalb der Schwellenwerte 0.2 (a) und 0.1 (b). Innerhalb von nur zwei Iterationen wurde diese um fast zwei Größenordnungen verkleinert.

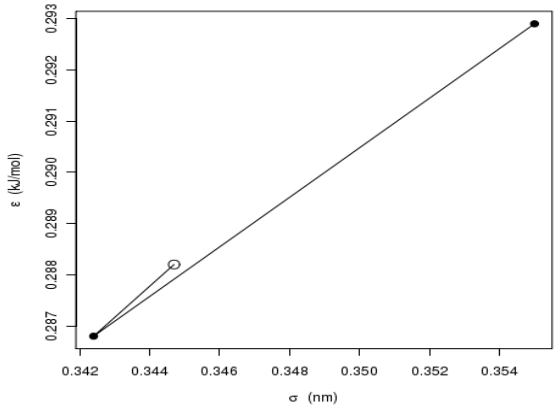


Abbildung 5.7: Entwicklung der LJ-Parameter im Falle von Benzol (D,ρ_l): σ(C) und ε(C). Der Kreis zeigt die optimalen Parameter an. Beide Parameter wurden im Laufe der Optimierung verkleinert.

optimalen Kraftfeldparameter durchgeführt. Im Mittel lag der MAPE-Wert für D bei 1.72% mit einer Standardabweichung von 0.88%, so daß ein Zufallsergebnis ausgeschlossen werden kann. Im Falle von ρ_l konnte das gewünschte Ergebnis, sämtliche Abweichungen vom Experiment zu jeder Temperatur auf unter 0.5% zu bringen, nicht erzielt werden. Abbildung 5.5(b) zeigt zwar eine Verschlechterung von ρ_l, allerdings lag diese im Bereich des statistischen Rauschens. Der Optimierungsalgorithmus legte deutlich mehr Wert auf die Diffusion als auf die Siededichte. Verbesserungen wären höchstens mit einer anderen Gewichtung zu erzielen, allerdings ist zum einen eine geeignete Gewichtung für D nicht bekannt, und zum anderen kann eine Verbesserung der Siededichte auf Kosten der Diffusion gehen. Da alle Siededichten bis auf 1% Genauigkeit vorhergesagt wurden und die Diffusion eine vergleichsweise schwer einzustellende Größe ist, bei der nur selten ein Fehler von unter 3% erzielt werden kann, wird das hier erhaltene Kraftfeld als zufriedenstellend und final betrachtet, zumal nur 18 Simulationen (effektiv neun, das heißt ohne Miteinbeziehung der letzten neun Armijo-Schritte) zum Erhalt notwendig waren. Im folgenden wird es für die Optimierung einer weiteren Größe, der Verdampfungsenthalpie, verwendet.

k	$x^{(k)}$	MAPE $\Delta_v H$	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
	Steilster Abstieg, $h = 0.02$, $\Omega = (30, 80, 30)$					
0	0.2420 0.3447 0.1255 0.2882 -0.1150	17.00%	0.40%	0.0868	1.6587 -6.0382 -2.1178 -3.3283 2.0692	7.6848
1	0.2385 0.3576 0.1301 0.2953 -0.1194	4.82%	2.28%	0.0086	-0.1232 -1.5291 -0.5585 -0.8385 0.5271	1.9095
2	0.2392 0.3671 0.1336 0.3005 -0.1227	3.18%	1.39%	0.0036	-0.3920 0.5153 0.4515 0.7093 -0.4803	1.1648
3	0.2396 0.3666 0.1331 0.2999 -0.1223	2.45%	1.20%	0.0023	-0.3412 0.3712 0.3548 0.6052 -0.3694	0.9396
4	0.2401 0.3661 0.1326 0.2990 -0.1217	1.70%	1.09%	0.0013	-0.2978 0.2448 0.2189 0.3931 -0.2748	0.6531
5	0.2411 0.3653 0.1319 0.2977 -0.1208	0.26%	0.64%	$1.7{\cdot}10^{-4}$	-0.1438 -0.0262 0.0379 0.0813 -0.0602	0.1818
	Polak-Ribière, $h = 0.01$, $\Omega = (30, 80, 30)$					
6	0.2422 0.3655 0.1316 0.2970 -0.1203	0.14%	0.33%	$5.5{\cdot}10^{-5}$	-0.1247 -0.1223 -0.0086 -0.0002 -0.0112	0.1753
7	0.2427 0.3659 0.1316 0.2969 -0.1202	0.09%	0.10%	$8{\cdot}10^{-6}$	0.0106 -0.0004 -0.0035 -0.0104 0.0089	0.0177

Tabelle 5.3: Optimierungsergebnisse für Benzol im Falle von $\Delta_v H$ und ρ_l: Innerhalb von fünf Iterationen mit dem Verfahren des steilsten Abstiegs und zwei Iterationen des Polak-Ribière-Verfahrens wurde die Fehlerfunktion um vier Größenordnungen verkleinert. Die bereits zu Beginn nahezu optimale Siededichte wurde zunächst etwas verschlechtert. Am Schluß der Optimierung lagen jedoch die Fehler bezüglich aller Siedichten weit unter 0.5%. Der durchschnittliche Fehler bezüglich der Verdampfungsenthalpie wurde von 17% auf weit unter 0.5% verbessert. Das erhaltene Kraftfeld ist somit in Bezug auf $\Delta_v H$ und ρ_l optimal. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{H}), \sigma(\mathrm{C}), \varepsilon(\mathrm{H}), \varepsilon(\mathrm{C})$ und $q(\mathrm{C})$, das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{H})^{(k)}, \sigma(\mathrm{C})^{(k)}, \varepsilon(\mathrm{H})^{(k)}, \varepsilon(\mathrm{C})^{(k)}, q(\mathrm{C})^{(k)}\right)$.

Abbildung 5.7 zeigt die Entwicklung der Kraftfeldparameter im Laufe der Optimierung: Sowohl σ als auch ε wurden im ersten Optimierungsschritt deutlich verkleinert. Im zweiten Schritt wurden sie noch etwas vergrößert, was auch an der Richtungsänderung des Gradienten in Tabelle 5.2 zu erkennen ist. Dieses Verhalten ist keineswegs ungewöhnlich: Zunächst werden die Änderungen überschätzt, um in die Nähe des Minimums zu gelangen. Dann kann sich aufgrund der lokalen Verfeinerungen die Verfahrensrichtung ändern.

Bei der Hinzunahme von $\Delta_v H$ ergab sich, daß alle drei Zielgrößen simultan nicht an die entsprechenden experimentellen Werte angepaßt werden konnten. Der MAPE-Wert bezüglich $\Delta_v H$ lag bei etwa 17% und konnte nicht verkleinert werden, auch nicht nach Hinzunahme der Wasserstoffatome und Mitoptimierung der Partialladungen. Eine mögliche Ursache hierfür ist, daß das molekulare Modell nicht geeignet ist. Es ist zu beachten, daß das intramolekulare Modell auch einen erheblichen Teil zur Genauigkeit der Simulation beiträgt. Ist das intramolekulare Modell zu ungenau, kann der intermolekulare Teil alleine nicht mehr zu einem optimalen Modell

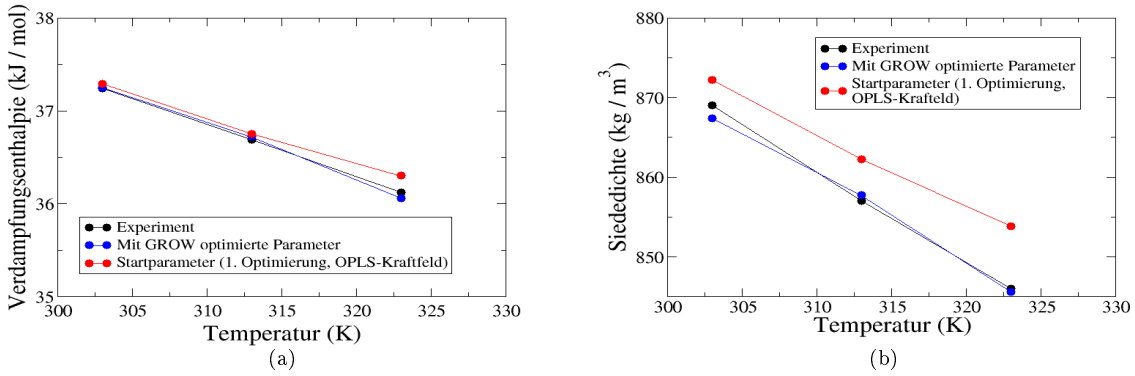


Abbildung 5.8: Optimierung von $\Delta_v H$ (a) und ρ_l (b) im Falle von Benzol. Beide Eigenschaften konnten optimal reproduziert werden.

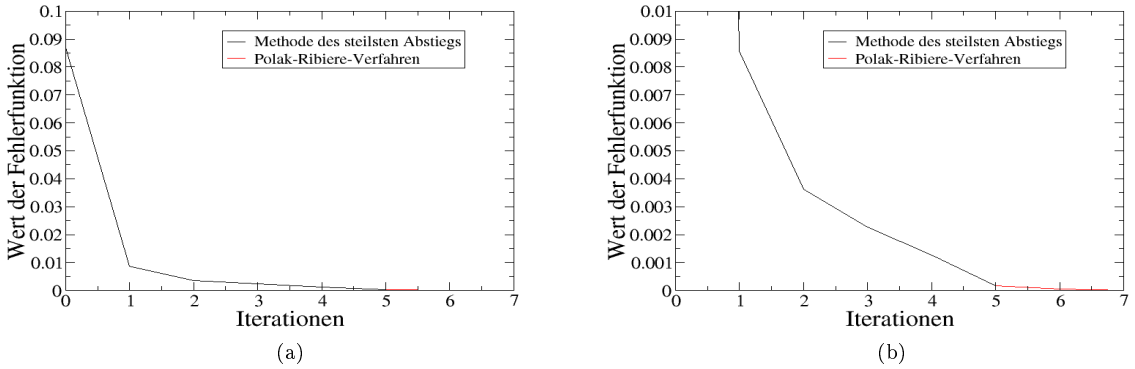
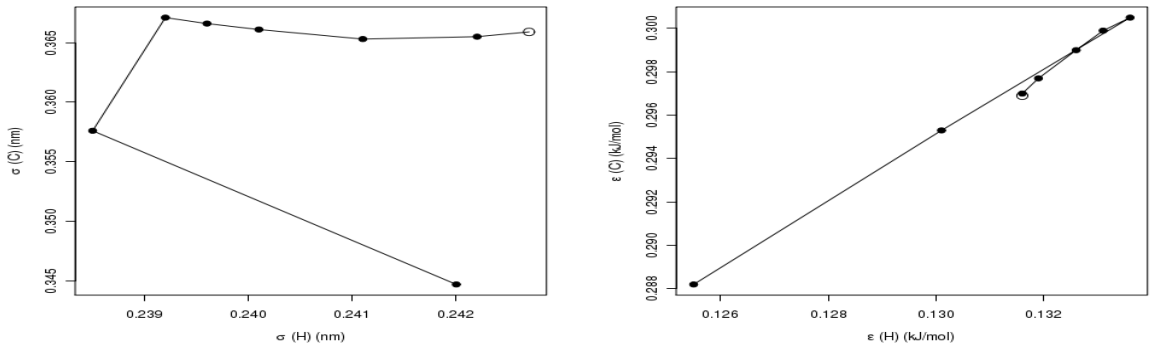


Abbildung 5.9: Entwicklung der Fehlerfunktion im Falle von Benzol ($\Delta_v H$,ρ_l) unterhalb der Schwellenwerte 10^{-1} (a) und 10^{-2} (b). Innerhalb von sieben Iterationen wurde diese um vier Größenordnungen verkleinert, und es ist ein sehr schneller Abfall zu erkennen.

führen. Wie oben bereits motiviert, ist die Delokalisierung der π-Elektronen nicht im Modell berücksichtigt. Daher kann nicht erwartet werden, daß das vorliegende einfache Modell sämtliche Eigenschaften korrekt reproduzieren kann. Ein quadrupolares Modell wäre wahrscheinlich besser geeignet, was allerdings mit *Gromacs* nicht simulierbar ist. Das vorliegende Modell, basierend auf zwölf Punktladungen, konnte nur jeweils zwei der drei Zielgrößen zufriedenstellend treffen.

Optimierungsergebnisse für $\Delta_v H$ und ρ_l Eine weitere Optimierung bezog sich auf $\Delta_v H$ und ρ_l, beginnend mit den finalen Parametern aus der Optimierung von D und ρ_l. Tabelle 5.3 und Abbildung 5.8 zeigen die Optimierungsergebnisse: Es konnten in diesem Fall optimale Ergebnisse erzielt werden, das heißt, sämtliche Siededichten und Verdampfungsenthalpien wichen im



(a) (b)

Abbildung 5.10: Entwicklung der LJ-Parameter im Falle von Benzol ($\Delta_v H$,ρ_l): σ(H) und σ(C) (a) sowie ε(H) und ε(C) (b). Die unausgefüllten Kreise zeigen die optimalen Parameter an. Man sieht deutlich, daß die Optimierung zunächst in eine falsche Richtung verlief.

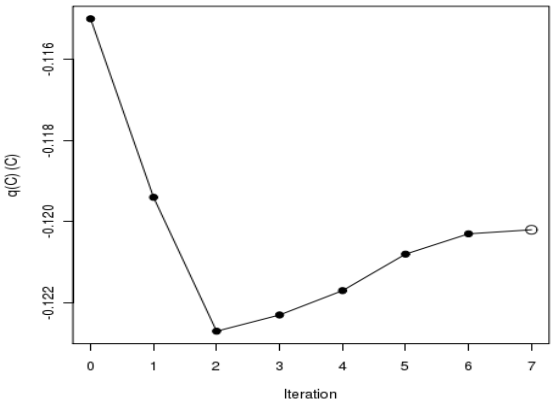
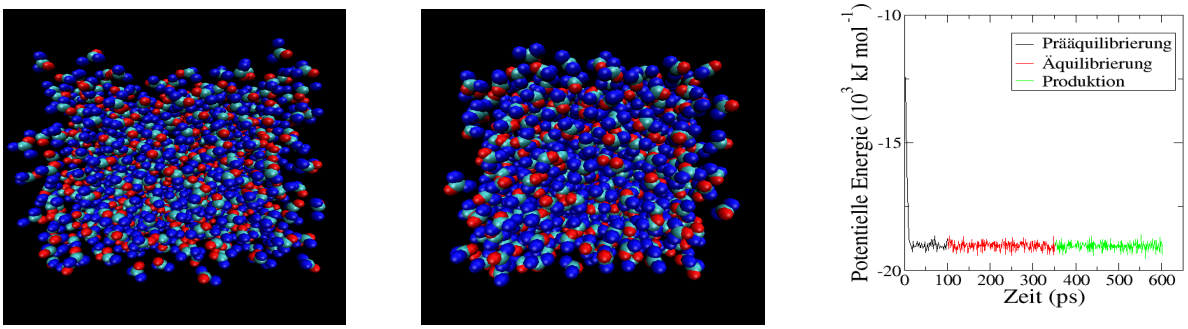


Abbildung 5.11: Partialladung des Kohlenstoffs q(C) im Falle von Benzol ($\Delta_v H$,ρ_l). Der Kreis zeigt die optimale Partialladung an. Auch hier ist ein deutlicher Umweg im Laufe der Optimierung zu erkennen.

Falle der optimalen Kraftfeldparameter um weniger als 0.5% vom Experiment ab. Es ist allerdings zu beachten, daß hierbei die Wasserstoffatome und Partialladung des Kohlenstoffs mit in die Optimierung einbezogen wurden. Die Gewichtung, der Armijo-Parameter ζ_A und die maximale Anzahl an Armijo-Schrittweiten wurden wie oben gewählt. Der zulässige Bereich war $\Omega := (30, 80, 30)$, das heißt, σ wurde um höchstens 30%, ε um höchstens 80% und die Partialladung des Kohlenstoffs um höchstens 30% geändert. Außerdem wurde zunächst $h = 0.02$ gewählt. Der Grund für die Vergrößerung des zulässigen Gebiets und der gröberen Diskretisierung lag in der langsamen Konvergenz. Der allgemeine Trend der Fehlerfunktion konnte mit einem größeren h zu Beginn der Optimierung besser reproduziert werden, und die Vergrößerung des zulässigen Gebiets ließ größere Armijo-Schritte zu. Nach fünf Iterationen mit der Methode des steilsten Abstiegs wurde die Fehlerfunktion um zwei Größenordnungen verbessert. Insgesamt wurden sechs Iterationen durchgeführt, wobei bei $k = 0$ und $k = 1$ jeweils eine, bei $k = 2$, $k = 3$ und $k = 4$ jeweils zwei und bei $k = 5$ sieben Armijo-Schritte notwendig waren.



(a) (b) (c)

Abbildung 5.12: Startkonfiguration (a), äquilibrierte Konfiguration (b) und Äquilibrierung der potentiellen Energie (c) im Falle von Phosgen. Die Kantenlänge der Box beträgt 6.27 nm.

Inklusive der Gradientenberechnung für $k = 6$ und den neun anschließenden Armijo-Schritten waren insgesamt 51 Simulationen notwendig (effektiv 42). Als nächstes wurde wieder $h = 0.01$ gesetzt, womit jedoch keine weitere Verbesserung erzielt werden konnte. Erst nach Einsatz des Polak-Ribière-Verfahrens konnte die Fehlerfunktion um zwei Größenordnungen verbessert werden. Hierzu waren zwei Iterationen, bei $k = 0$ sieben, bei $k = 1$ sechs Armijo-Schritte und somit insgesamt 29 Simulationen (effektiv 20) erforderlich. Das Kraftfeld war in Bezug auf die vorliegende Aufgabenstellung optimal. Für die gesamte Optimierung waren effektiv 62 Simulationen erforderlich. Abbildung 5.9 zeigt die Entwicklung der Fehlerfunktion für die Methode des steilsten Abstiegs und das Polak-Ribière-Verfahren.

Abbildung 5.10 gibt die Entwicklung der LJ-Parameter (Abbildung 5.10(a) bezieht sich auf σ und Abbildung 5.10(b) auf ε) und Abbildung 5.11 die der Partialladung des Kohlenstoffs im Laufe der Optimierung an: ε(H) und ε(C) zeigten ein ähnliches Verhalten wie σ(C) und ε(C) bei der Optimierung von D und ρ_l, allerdings wurden beide zunächst vergrößert und dann verkleinert. Sowohl σ(H) als auch σ(C) gingen zunächst in die falsche Richtung, um dann später gegen das Minimum zu konvergieren. Wie aus Abbildung 5.11 hervorgeht, sank die Partialladung des Kohlenstoffs zu Beginn der Optimierung drastisch ab, um dann wieder etwas anzusteigen und im weiteren Verlauf nahezu konstant zu bleiben.

Fazit Für Benzol wurde also im Falle von D und ρ_l ein zufriedenstellendes und im Falle von $\Delta_v H$ und ρ_l ein optimales Kraftfeld mit einer akzeptablen Anzahl an Iterationen und effektiven Anzahl an Simulationen erzielt. Alle drei Eigenschaften konnten mit dem vorliegenden Modell jedoch nicht gleichzeitig optimiert werden.

5.1.3 Phosgen: Sukzessive Optimierung von Zielgrößen

Das Modell für Phosgen beinhaltet drei Atomtypen: Kohlenstoff, Sauerstoff und Chlor. Bindungen und Winkel wurden mithilfe des LINCS-Algorithmus fixiert. Aufgrund des variablen

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	C, O, Cl	*All-Atom*-Modell
	Dipole/Quadrupole	–	
	Molare Masse	98.92 g/mol	m(C) = 12.01 g/mol, m(O) = 16.00 g/mol m(Cl) = 35.45 g/mol
	Molekülstruktur	verzerrbarer Tetraeder	polar und polarisierbar
	Intramolekulares Kraftfeld	3 starre Bindungen (LINCS)	Quelle: Guest u. a. (2005)
		3 starre Winkel (LINCS)	Quelle: Guest u. a. (2005)
		1 Diederwinkel	Quelle: Guest u. a. (2005)
	Ladungen	4 Punktladungen	Quelle: $WOLF_2PACK$
	Besonderheiten	1,4-Wechselwirkungen exkludiert	
Simulationsbox	Startkonfiguration	randomisiert	siehe Abbildung 5.12(a)
	Anzahl Moleküle	750	
	Kantenlänge	6.27 nm	
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung	10000	Schrittweite: 0.2 fs
	Prääquilibrierung	750000	Schrittweite: 2 fs
	Äquilibrierung	250000	Schrittweite: 2 fs
	Produktion	250000	Schrittweite: 2 fs
Optimierungsrelevantes	zu optimierende Parameter	σ(C), σ(O), σ(Cl) ε(C), ε(O), ε(Cl)	Quelle Startwerte: Gromos (van Gunsteren u. a., 1996)
	Temperaturen	235.65, 243.15, 250.65, 258.15, 265.65, 273.15, 280.65	in K
	Dampfdrücke	0.1203, 0.1830, 0.2704, 0.3889, 0.5461, 0.7504, 1.0109	in bar (aus Antoine-Gleichung (5.1))
Ausgaben:			
Zielgrößen	experimentell	$\Delta_v H$, ρ_l	Quellen: $\Delta_v H$: Literatur (Giauque u. Jones, 1948) ρ_l: Regressionsformel (5.2)
	simuliert	$\Delta_v H$: Gleichung (C.3) ρ_l: Gleichung (C.8)	Approximation (ideales Gas)
	Toleranzen	$\Delta_v H$: 1% ρ_l: 0.5%	

Tabelle 5.4: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Phosgen.

Diederwinkels handelt es sich bei Phosgen um einen verzerrbaren Tetraeder. Die intramolekularen Kraftfeldparameter stammen aus quantenmechanischen Rechnungen, die mit GÅMESS (Guest u. a., 2005) von Kirschner (2010) durchgeführt wurden. Weiterhin wurden vier Punktladungen angenommen, zwei gleich große negative Partialladungen für Chlor, eine negative für Sauerstoff und eine positive für Kohlenstoff. Die Werte dieser Partialladungen wurden, ebenfalls von Kirschner (2010), mithilfe des am Fraunhofer-Instituts SCAI entwickelten Softwarepakets *$WOLF_2PACK$* (Reith u. Kirschner, 2011) ermittelt, einem generischen Kraftfeldoptimierungsworkflow, welcher die Ebene der Quantenmechanik mit der atomistischen Ebene auf geeignete und akkurate Art und Weise miteinander verbindet. Sämtliche 1,4-Wechselwirkungen wurden exkludiert.

Simulations- und Optimierungseinstellungen Bei Phosgen wurden nur Verdampfungsenthalpie und Siededichte an der Phasenübergangskurve an experimentelle Werte der Flüssigkeit angepaßt. Diese Optimierung wird im folgenden mit $(\Delta_v H, \rho_l)$ markiert. Die anzupassenden Kraftfeldparameter waren die Lennard-Jones-Parameter aller drei Atomtypen. Die Startparameter stammen aus dem Gromos-43-A1-Kraftfeld (van Gunsteren u. a., 1996). Simuliert wurde stets an der Phasenübergangskurve zwischen Flüssigkeit und Gas zu den Temperaturen 235.65, 243.15, 250.65, 258.15, 265.65, 273.15 und 280.65 K. Die entsprechenden experimentellen Dampfdrücke (0.1203, 0.1830, 0.2704, 0.3889, 0.5461, 0.7504 und 1.0109 bar) sind aus der NIST-Datenbank (NIST, 2011) entnommen und wurden wieder mithilfe der Antoine-Gleichung (5.1) berechnet. Die Koeffizienten $A, B, C \in \mathbb{R}$ stammen dabei aus Giauque u. Jones (1948). Tabelle 5.4 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Phosgen. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen und Toleranzen für die simulierten Zielgrößen angegeben. Bei der experimentellen Siededichte gibt es bei Phosgen eine Besonderheit: Sie basiert auf der folgenden temperaturabhängigen Regressionsformel:

$$\rho(T) = a + bT + cT^2. \tag{5.2}$$

Die Regressionskoeffizienten $a, b, c \in \mathbb{R}$ wurden von Davies (1945) bestimmt. Gleichung (5.2) gilt allerdings nur bei Atmosphärendruck und nicht an der Phasenübergangskurve. Dies ist allerdings bei Phosgen nicht problematisch, da der Einfluß der isothermen Kompressibilität auf die Dichte vernachlässigbar gering ist.

Abbildung 5.12(a) zeigt die Startkonfiguration der Simulationsbox: In einem Würfel mit einer Kantenlänge von 6.27 nm wurden 750 Phosgenmoleküle per Zufall so positioniert, daß keine Überlappungen auftraten. Nach einer Energieminimierung und kurzen NVT-Simulation von 10000 Zeitschritten mit kleinerer Zeitschrittweite (Prä-Prä-Äquilibrierung) wurden weitere Läufe verwendet, um die Moleküle aus ihren originalen Positionen herauszutreiben (Präääquilibrierung). Das Äquilibrierungszeitfenster bestand dafür nur aus 250000 Zeitschritten, analog zu Benzol. Nach meistens nur einem Äquilibrierungszyklus waren die Äquilibrierungskriterien (3.70) für potentielle Energie, Siededichte und nichtbindende Energie erfüllt. Abbildung 5.12(b) zeigt eine äquilibrierte Konfiguration: Es ist zu erkennen, daß das System dichter ist, was auf die anziehenden Wechselwirkungen während des Simulationsverlaufs zurückzuführen ist. Abbildung 5.12(c) zeigt die Äquilibrierung der potentiellen Energie: Im Laufe der Präääquilibrierung sank die potentielle Energie drastisch ab, um dann um einen konstanten Mittelwert herum zu oszillieren, welcher auch in der Produktionsphase erhalten blieb.

Optimierungsergebnisse für $\Delta_v H$ und ρ_l Tabelle 5.5 und Abbildung 5.13 zeigen die Optimierungsergebnisse im Falle von $\Delta_v H$ und ρ_l. Verwendet wurde dabei die Methode des steilsten Abstiegs mit $h = 0.01$, $\zeta_A = 0.2$ und $\forall_{i,T}\ w_{i,T} = 1$. Auch hierbei bestand die Motivation darin, alle Eigenschaften zunächst als gleichwertig zu betrachten. Der zulässige Bereich war mit $\Omega = (30, 80)$ sehr groß, denn σ wurde um maximal 30% und ε um maximal 80% verändert. Im Laufe der Optimierung wurden die Energieparameter (ε) im Vergleich zu den Längenparametern (σ) nur unwesentlich verändert, was auch an den Gradienten in Tabelle 5.5 festzustellen ist, da vor allem die Siededichte verbessert wurde.

k	$x^{(k)}$	MAPE $\Delta_v H$	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\nabla F\left(x^{(k)}\right)\|$
	Steilster Abstieg, $h = 0.01$, $\Omega = (30, 80)$					
0	0.2473 0.2004 0.3973 0.3830 1.7155 1.2186	1.26%	17.20%	0.2083	-1.6982 0.0895 11.3474 -0.2978 -0.0246 -0.5636	11.49
1	0.2494 0.2003 0.3829 0.3834 1.7155 1.2193	0.15%	10.13%	0.0719	-0.8926 0.3762 7.3612 -0.2314 -0.0439 -0.3143	7.44
2	0.2508 0.1997 0.3718 0.3837 1.7156 1.218	0.74%	4.13%	0.0124	-0.1316 0.3749 3.2113 -0.0288 -0.0076 -0.0440	3.24
3	0.2511 0.1987 0.3630 0.3838 1.7156 1.2199	1.09%	0.97%	0.0016	0.4261 0.1672 -0.8928 0.0816 0.0225 0.1645	1.02
4	0.2507 0.1985 0.3638 0.3837 1.7156 1.2198	0.93%	0.48%	$8.3{\cdot}10^{-4}$	0.3291 0.1709 -0.4451 0.0629 0.0189 0.1262	0.60
5	0.2497 0.1980 0.3652 0.3835 1.7155 1.2194	0.47%	0.44%	$3.7{\cdot}10^{-4}$	0.1092 0.1201 0.3169 0.0161 0.0061 0.0395	0.36
6	0.2496 0.1978 0.3647 0.3835 1.7155 1.2193	0.41%	0.31%	$2.2{\cdot}10^{-4}$	0.1202 0.0934 0.0879 0.0216 0.0075 0.0456	0.18

Tabelle 5.5: Optimierungsergebnisse für Phosgen im Falle von $\Delta_v H$ und ρ_l: Innerhalb von sechs Iterationen mit der Methode des steilsten Abstiegs wurde die Fehlerfunktion um fast drei Größenordnungen verkleinert. Die Verdamfungsenthalpien waren von Anfang an bereits ziemlich nahe am Experiment, während die Fehler bezüglich der Siededichten um mehr als den Faktor 50 verbessert wurden. In der dritten Iteration wurden die Siededichten auf Kosten der Verdamfungsenthalpien verbessert. Von da an wurden beide Größen simultan verbessert. Das erhaltene Kraftfeld ist in Bezug auf $\Delta_v H$ und ρ_l optimal, da alle Zielgrößen zu allen Temperaturen in ihren spezifischen experimentellen Toleranzbereichen liegen. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{C})$, $\sigma(\mathrm{O})$ und $\sigma(\mathrm{Cl})$ sowie $\varepsilon(\mathrm{C})$, $\varepsilon(\mathrm{O})$ und $\varepsilon(\mathrm{Cl})$ das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{C})^{(k)}, \sigma(\mathrm{O})^{(k)}, \sigma(\mathrm{Cl})^{(k)}, \varepsilon(\mathrm{C})^{(k)}, \varepsilon(\mathrm{O})^{(k)}, \varepsilon(\mathrm{Cl})^{(k)}\right)$.

Innerhalb von nur sechs Iterationen konnte die Fehlerfunktion um fast drei Größenordnungen verkleinert werden, wobei der Fehler in Bezug auf ρ_l um mehr als den Faktor 50 verbessert wurde. Der Algorithmus war also insbesondere in der Lage, mit den potentiell großen Parameränderungen robust umzugehen. Die Verdampfungsenthalpie war für die Startparameter mit durchschnittlich 1.26% Abweichung bereits nahe am Experiment. In der dritten Iteration ging die drastische Verbesserung der Siededichte auf Kosten der Verdampfungsenthalpie. Allerdings wurde auch diese von da an stetig verbessert. Abbildung 5.14 zeigt die Entwicklung der Fehlerfunktion.

Für die Gradientenberechnung waren stets sechs Simulationen erforderlich. Die maximale Anzahl an Armijo-Schritten wurde wie bei Benzol auf 10 gesetzt. Insgesamt wurden sechs Iterationen durchgeführt, wobei bei $k = 0$, $k = 1$ und $k = 2$ jeweils eine, bei $k = 3$ und $k = 4$ jeweils drei und bei $k = 5$ sieben Armijo-Schritte notwendig waren. Inklusive der Gradientenberechnung für $k = 6$ und den neun anschließenden Armijo-Schritten waren insgesamt 56 Simulationen notwendig (effektiv 47). Mit einer zusätzlich zum Vergleich durchgeführten effizienten Gradientenberechnung (siehe Abschnitt 3.6.2) konnten bei $x^{(1)}$ sechs Simulationen eingespart werden. Dabei wurden $C = 1.1$ (Bedingung (3.98)) und $\epsilon_\kappa = 15$ (Bedingung (3.100)) gewählt. Die Optimierung verlief nahezu genauso wie zuvor: Die Richtung des Gradienten wurde korrekt wiedergegeben, und es war lediglich ein Armijo-Schritt zum Erhalt von $x^{(2)}$ notwendig. Die resultierenden Kraftfeldparameter unterschieden sich demnach ebenfalls nur geringfügig. Ab $x^{(2)}$ mußte die Normbedingung (3.98) jedoch verschärft werden ($C = 1.0$). Dies hatte zur Folge, daß letztere nicht mehr erfüllt war und der Gradient nur noch klassisch berechnet wurde. Allerdings zeigt der Vergleich, daß bei geeigneter Parametrisierung eine effiziente Gradientenberechnung

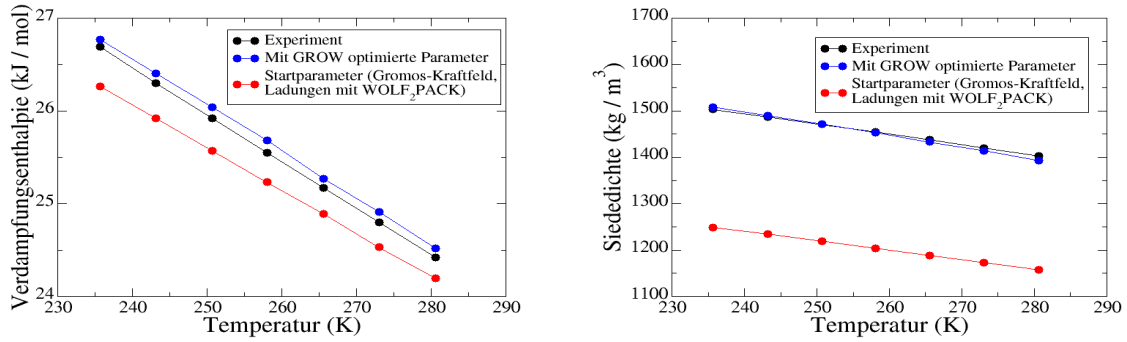


Abbildung 5.13: Optimierung von $\Delta_v H$ (a) und ρ_l (b) im Falle von Phosgen. Beide Eigenschaften konnten optimal reproduziert werden.

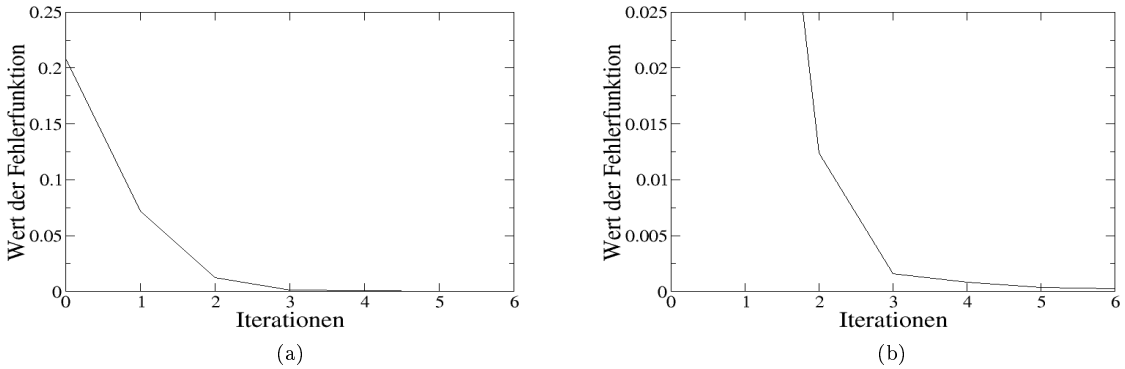
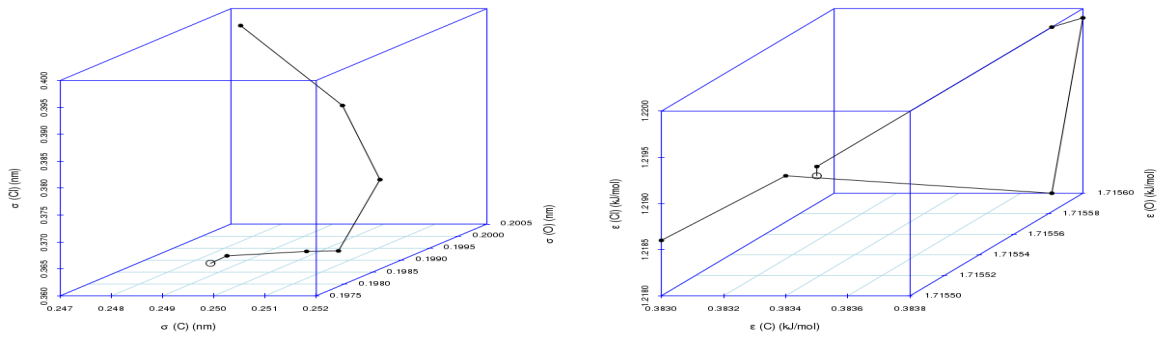


Abbildung 5.14: Entwicklung der Fehlerfunktion im Falle von Phosgen ($\Delta_v H$,ρ_l) unterhalb der Schwellenwerte $2.5{\cdot}10^{-1}$ (a) und $2.5{\cdot}10^{-2}$ (b). Diese konnte innerhalb von sechs Iterationen um fast drei Größenordnungen verkleinert werden.

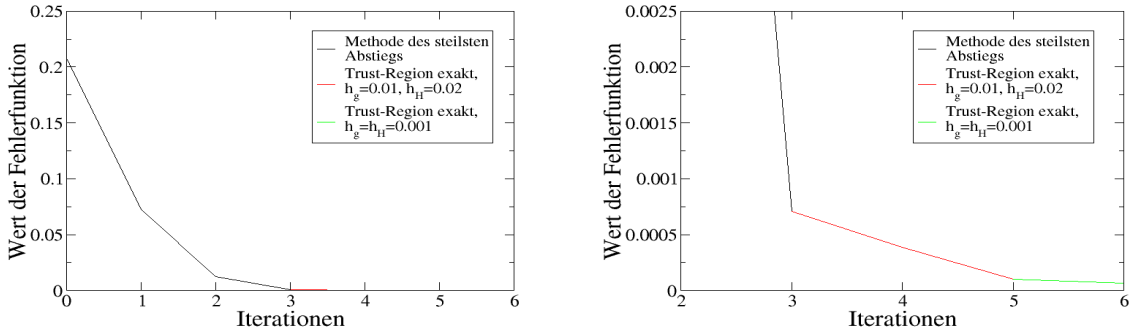
auf Molekulare Simulationen zumindest zu Beginn der Optimierung erfolgreich einsetzbar ist. Abbildung 5.13 zeigt, daß beide Eigenschaften nach der gradientenbasierten Optimierung fast mit den experimentellen Daten übereinstimmten. Auch der Trend der experimentellen Kurve wurde sehr gut wiedergegeben. Der Fehler in Bezug die Siededichte lag zu allen Temperaturen stets unter 0.5%, der in Bezug auf die Verdampfungsenthalpie ebenfalls.

Abbildung 5.15 zeigt die Entwicklung der Kraftfeldparameter im Laufe der Optimierung (Abbildung 5.15(a) bezieht sich auf σ und Abbildung 5.15(b) auf ε): ε wurde im Falle von C und Cl vergrößert und σ im Falle von O und Cl verkleinert. Es ist zu sehen, daß das Optimierungsverfahren zunächst in eine falsche Richtung läuft. Dies kann dadurch gedeutet werden, daß zu Beginn viel zu großen Wert auf die Verbesserung der Siededichte gelegt wurde, was ja auch auf Kosten der Verdampfungsenthalpie ging. Ab der dritten Iteration wurden beide Größen simultan verbessert, und das Verfahren konvergierte gerichtet gegen einen optimalen Parametervektor.



(a) (b)

Abbildung 5.15: Entwicklung der LJ-Parameter im Falle von Phosgen ($\Delta_v H$,ρ_l): σ(C), σ(O) und σ(Cl) (a) sowie ε(C), ε(O) und ε(Cl) (b). Die Kreise zeigen die optimalen Parameter an. ε wurde im Falle von C und Cl vergrößert und σ im Falle von O und Cl verkleinert. Das Optimierungsverfahren läuft zunächst in eine falsche Richtung und trifft erst später näherungsweise das Minimum.



(a) (b)

Abbildung 5.16: Entwicklung der Fehlerfunktion im Falle von Phosgen (ρ_l) unterhalb der Schwellenwerte $2.5{\cdot}10^{-1}$ (a) und $2.5{\cdot}10^{-2}$ (b). Diese konnte innerhalb von sechs Iterationen um fast vier Größenordnungen verkleinert werden.

Sukzessive Zielgrößenoptimierung: $\rho_l \rightarrow \Delta_v H$ In Abschnitt 5.1.2 wurde gezeigt, daß im Falle von Benzol ein Kraftfeld, welches für zwei Eigenschaften optimiert wurde, auf eine dritte nicht anwendbar ist. Da bei Phosgen keine anderen experimentellen Eigenschaften an der Phasenübergangskurve zu den gegebenen Temperaturen ermittelt werden konnten, wurden hier zwei verschiedene *sukzessive* Optimierungen durchgeführt, um ein resultierendes Kraftfeld zu bewerten. Gleichzeitig sollte dadurch überprüft werden, ob verschiedene Zielgrößen auch nacheinander anstatt simultan angepaßt werden können.
Tabelle 5.6 zeigt die Optimierungsergebnisse für die alleinige Anpassung der Siededichte, welche mit (ρ_l) markiert wird: Innerhalb der ersten drei Iterationen mit der Methode des steilsten

k	$x^{(k)}$	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
	Steilster Abstieg, $h = 0.01$, $\Omega = (30, 80)$				
0	0.2473 0.2004 0.3973 0.3830 1.7155 1.2186	17.22%	0.2076	-1.3596 0.3525 11.2082 -0.2961 -0.1368 -0.5332	11.3132
1	0.2490 0.1999 0.3830 0.3834 1.7156 1.2193	10.19%	0.0727	-0.8924 0.3292 7.3598 -0.2021 -0.0654 -0.3470	7.4321
2	0.2504 0.1995 0.3719 0.3837 1.7157 1.2198	4.19%	0.0124	-0.38342 0.1894 3.3058 -0.0993 -0.0398 -0.1444	3.3382
3	0.2514 0.1990 0.3630 0.3839 1.7158 1.2202	0.95%	$7.1{\cdot}10^{-4}$	0.0861 -0.0579 -0.7989 0.0216 0.0096 0.0376	0.8068
	Trust-Region exakt, $h_g = 0.01$, $h_H = 0.02$, $\Omega = (30, 80)$				
4	0.2513 0.1988 0.3635 0.3838 1.7157 1.2201	0.66%	$3.9{\cdot}10^{-4}$	0.0569 -0.0409 -0.5597 0.0156 0.0024 0.0230	0.5647
5	0.2542 0.1996 0.3653 0.3844 1.7160 1.2212	0.31%	$1.0{\cdot}10^{-4}$	-0.0223 0.0083 0.1619 -0.0061 -0.0022 -0.0091	0.1640
	Trust-Region exakt, $h_g = h_H = 0.001$, $\Omega = (30, 80)$				
6	0.2544 0.1998 0.3650 0.3845 1.7162 1.2211	0.26%	$6.8{\cdot}10^{-5}$	-0.0106 -0.0150 0.0140 -0.0161 0.0117 -0.0085	0.0316

Tabelle 5.6: Optimierungsergebnisse für Phosgen im Falle von ρ_l alleine: Innerhalb von drei Iterationen mit dem Verfahren des steilsten Abstiegs und drei Iterationen des exakten Trust-Region-Verfahrens wurde die Fehlerfunktion um fast vier Größenordnungen verkleinert. Das erhaltene Kraftfeld ist bezüglich ρ_l optimal. Obwohl nur eine Zielgröße betrachtet wurde, war das zugehörige Optimierungsproblem schwieriger zu lösen als im Falle von zwei Zielgrößen. Es stellte sich heraus, daß weder der negative Gradient noch eine Polak-Ribière-Richtung in der Nähe des Minimums als Abstiegsrichtung geeignet waren. Somit war ein anschließendes Trust-Region-Verfahren vonnöten.

Abstiegs verhielt sich der Optimierungsablauf ähnlich wie in Tabelle 5.5. Allerdings war er schneller, was anhand der Fehlerfunktionswerte und Gradienten zu erkennen und darauf zurückzuführen ist, daß keine andere Zielgröße vorhanden war, die zunächst verschlechtert wurde. Das Optimierungsproblem war jedoch schwieriger zu lösen als im Falle zweier Zielgrößen. Ab der dritten Iteration konnte weder die Methode des steilsten Abstiegs noch das Polak-Ribière-Verfahren kleinere Fehlerfunktionswerte finden. Nur mit dem exakten Trust-Region-Verfahren, welches auch Krümmungseigenschaften der Fehlerfunktion mitberücksichtigt, konnten Verbesserungen erzielt werden. Außerdem konnte durch die Modifizierung des jeweiligen Vertrauensgebiets und damit der lokalen Modellierung der Fehlerfunktion gezielter gesucht werden als mit einer vorgegebenen Abstiegsrichtung. Der Nachteil ist jedoch, daß in jeder Iteration eine Hesse-Matrix benötigt wurde, was den Rechenaufwand drastisch erhöhte. Gemäß den Überlegungen in Abschnitt 3.5.6 wurde zunächst $h_g = 0.01$ und $h_H = 0.02$ gewählt, was bereits zu einer deutlichen Verbesserung führte. Die Wahl $h_g = h_H = 0.001$ führte zu optimalen Siededichten zu jeder Temperatur. Die Gradientenrichtungen der Iterationen 4–6 in Tabelle 5.6 lassen vermuten, daß die Fehlerfunktion in der Nähe des Minimums einen anderen Verlauf hat als im Falle zweier Zielgrößen. Auf jeden Fall unterschieden sich die Optimierungsverläufe dort signifikant voneinander. Abbildung 5.16 zeigt den Verlauf der Fehlerfunktion bei der Optimierung mit der Methode des steilsten Abstiegs und dem exakten Trust-Region-Verfahren. Der Rechenaufwand teilte sich wie folgt auf: Das Verfahren des steilsten Abstiegs benötigte innerhalb der drei Iterationen jeweils nur einen Armijo-Schritt, insgesamt also 28 (effektiv 19) Simulationen. Für die Berechnung einer Hesse-Matrix waren jeweils 21 Simulationen erforderlich. Insgesamt wurden für die drei Trust-Region-Iterationen 88 Simulationen benötigt. Im letzten Trust-Region-Schritt

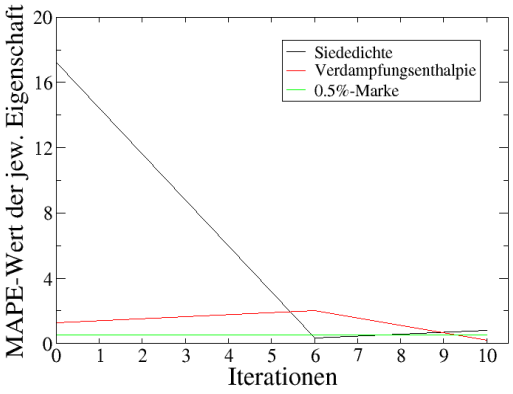
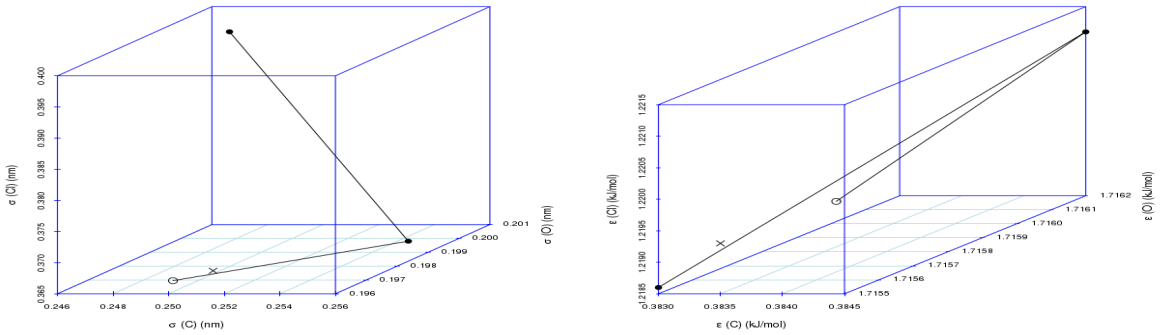


Abbildung 5.17: Sukzessive Optimierung von ρ_l und $\Delta_v H$ im Falle von Phosgen. Die Linien führen lediglich zu den jeweiligen Ergebnissen der beiden aufeinanderfolgenden Optimierungsläufen. Zwischenresultate sind der Übersichtlichkeit halber nicht angegeben. Zuerst wurde ρ_l und dann $\Delta_v H$ optimiert. Im Falle einer optimalen Siededichte wird die Verdampfungsenthalpie verschlechtert. Wird anschließend die Verdampfungsenthalpie optimiert, wird die Siededichte nur unwesentlich schlechter, ist jedoch nicht mehr als optimal einzustufen.

wurde lediglich noch ein Gradient, nicht jedoch eine weitere Hesse-Matrix berechnet. Zusammen ergab sich dadurch eine sehr hohe Gesamtanzahl an Simulationen von 116 (effektiv 107). Auch hier wurde zum Vergleich eine effiziente Gradientenberechnung durchgeführt, wobei ebenfalls bei $x^{(1)}$ sechs Simulationen eingespart werden konnten. Die Einstellung der entsprechenden Parameter für Norm- und Konditionszahlbedingung waren dieselben wie oben. Auch hier verlief die Optimierung nahezu genauso wie zuvor. Ab $x^{(2)}$ wurde der Gradient jedoch nur noch klassisch berechnet. Im Falle des Trust-Region-Verfahrens konnte keine effiziente Berechnung der Hesse-Matrix realisiert werden, da die Normbedingung (3.101) nie erfüllt war.

Abbildung 5.17 zeigt Siededichte und Verdampfungsenthalpie und Abbildung 5.18 die LJ-Parameter bei einer sukzessiven Optimierung (Abbildung 5.18(a) bezieht sich auf σ und Abbildung 5.18(b) auf ε), beginnend mit der Siededichte: Zunächst ging die Optimierung der Siededichte auf Kosten der Verdampfungsenthalpie, und die resultierenden LJ-Parameter unterschieden sich signifikant von denen, die aus der Optimierung beider Zielgrößen resultieren. Bei der nachfolgenden Optimierung der Verdampfungsenthalpie, welche vier Iterationen benötigte, wurde die Siededichte nur unwesentlich schlechter, blieb jedoch nicht im optimalen Bereich. Allerdings waren die Parameter denen aus der Optimierung beider Zielgrößen relativ ähnlich, abgesehen von denen des Wechselwirkungszentrums am Sauerstoffatom. Im Vergleich zur simultanen Optimierung von Siededichte und Verdampfungsenthalpie war der Rechenaufwand dieser sukzessiven Optimierung allerdings deutlich höher, was vor allem an der Notwendigkeit lag, das Trust-Region-Verfahrens einzusetzen.



(a) (b)

Abbildung 5.18: Entwicklung der LJ-Parameter bei der sukzessiven Optimierung von ρ_l und $\Delta_v H$ im Falle von Phosgen: σ(C), σ(O) und σ(Cl) (a) und ε(C), ε(O) und ε(Cl) (b). Zuerst wurde ρ_l und dann $\Delta_v H$ optimiert. Die Linien führen lediglich zu den jeweiligen Ergebnissen der beiden aufeinanderfolgenden Optimierungsläufen. Zwischenresultate sind der Übersichtlichkeit halber nicht angegeben. Die bei dieser Optimierung erhaltenen finalen Parameter sind mit Kreisen und die optimalen Parameter im Falle beider Zielgrößen mit Kreuzen markiert. Außer für das Wechselwirkungszentrum am Sauerstoff stimmen die Parameter relativ gut überein. Wird allerdings nur die Siededichte optimiert, unterscheiden sich alle Parameter signifikant voneinander.

k	$x^{(k)}$	MAPE $\Delta_v H$	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
	Steilster Abstieg, $h = 0.01$, $\Omega = (30, 80)$				
0	0.2473 0.2004 0.3973	1.26%	0.0012	-0.3884 -0.2999 0.0811	0.5240
	0.3830 1.7155 1.2186			-0.0585 -0.019 -0.1530	
1	0.2500 0.2025 0.3968	0.30%	$8.7{\cdot}10^{-5}$	-0.0583 -0.0469 0.0147	0.0806
	0.3834 1.7156 1.2197			-0.0096 -0.0045 -0.0240	

Tabelle 5.7: Optimierungsergebnisse für Phosgen im Falle von $\Delta_v H$ alleine: Innerhalb von nur einer Iteration mit der Methode des steilsten Abstiegs wurde die Fehlerfunktion um etwa zweieinhalb Größenordnungen verkleinert. Es ist dabei zu beachten, daß die Verdamfungsenthalpie von Anfang an bereits bereits nah am Experiment lag. Das erhaltene Kraftfeld ist in Bezug auf $\Delta_v H$ optimal.

Sukzessive Zielgrößenoptimierung: $\Delta_v H \rightarrow \rho_l$ Die sukzessive Optimierung der beiden Zielgrößen wurde auch umgekehrt durchgeführt: Tabelle 5.7 zeigt die Optimierungsergebnisse im Falle von $\Delta_v H$ alleine. Da die Verdampfungsenthalpien bereits zu Beginn bereits nah am Experiment lagen, war nur eine Iteration zum Erhalt einer optimalen Verdampfungsenthalpie nötig. Eine effiziente Gradientenberechnung machte daher in diesem Fall keinen Sinn.

Abbildung 5.19 zeigt Verdampfungsenthalpie und Siededichte und Abbildung 5.20 die LJ-Parameter bei einer sukzessiven Optimierung (Abbildung 5.20(a) bezieht sich auf σ und Abbildung 5.20(b) auf ε), beginnend mit der Verdampfungsenthalpie: Die optimale Verdampfungsenthalpie lieferte auch eine etwas bessere Siededichte, und die resultierenden LJ-Parameter unterscheiden sich signifikant von denen, die aus der Optimierung beider Zielgrößen resultierten. Nach der nachfolgenden Optimierung der Siededichte, welche fünf Iterationen benötigte, wurde die Verdampfungsenthalpie wesentlich schlechter. Alle Parameter, die aus den drei Optimierungsprozessen (beide Zielgrößen, nur Verdampfungsenthalpie und anschließend nur Siededichte) resultieren, unterschieden sich signifikant voneinander. Im Vergleich zur simultanen

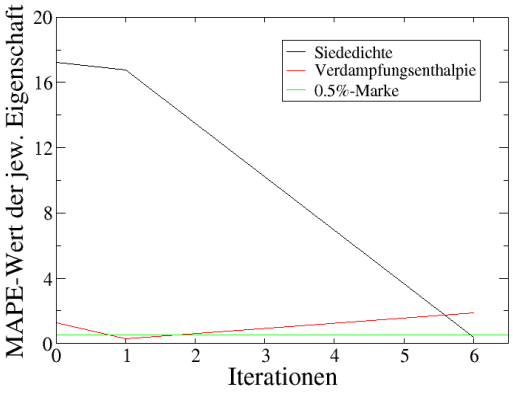
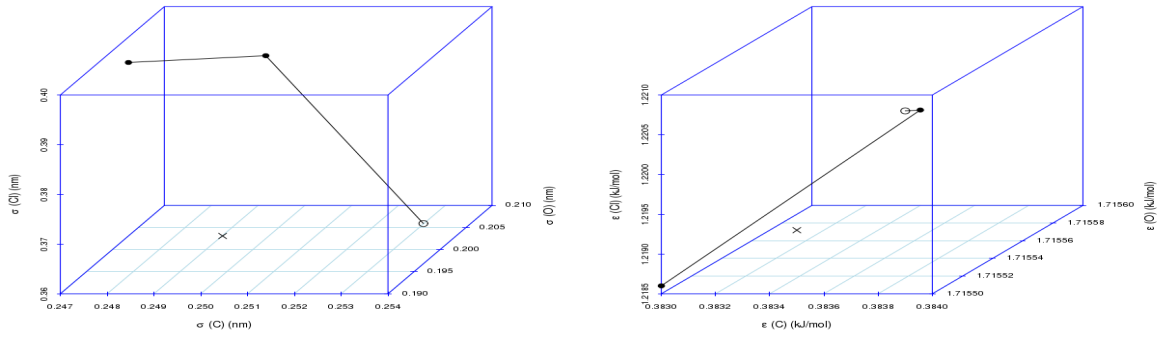


Abbildung 5.19: Sukzessive Optimierung von $\Delta_v H$ und ρ_l im Falle von Phosgen. Zuerst wurde $\Delta_v H$ und dann ρ_l optimiert. Im Falle einer optimalen Verdampfungsenthalpie wird die Siededichte ebenfalls geringfügig verbessert. Wird anschließend die Siededichte optimiert, wird die Verdampfungsenthalpie allerdings wesentlich schlechter.

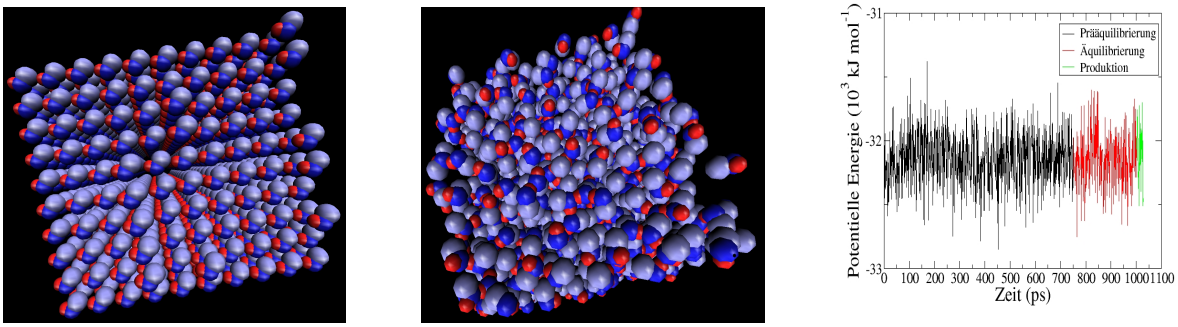


(a) (b)

Abbildung 5.20: Entwicklung der LJ-Parameter bei der sukzessiven Optimierung von $\Delta_v H$ und ρ_l im Falle von Phosgen: σ(C), σ(O) und σ(Cl) (a) und ε(C), ε(O) und ε(Cl) (b) Zuerst wurde $\Delta_v H$ und dann ρ_l optimiert. Die Linien führen lediglich zu den jeweiligen Ergebnissen der beiden aufeinanderfolgenden Optimierungsläufen. Zwischenresultate sind der Übersichtlichkeit halber nicht angegeben. Die bei dieser Optimierung erhaltenen finalen Parameter sind mit Kreisen und die optimalen Parameter im Falle beider Zielgrößen mit Kreuzen markiert. Alle Parameter, die aus den drei Optimierungsprozessen resultieren, unterschieden sich signifikant voneinander.

Optimierung von Siededichte und Verdampfungsenthalpie war der Rechenaufwand dieser sukzessiven Optimierung vergleichbar: Es wurden ebenfalls sechs Iterationen benötigt.

Fazit Für Phosgen wurde mit wenig Rechenaufwand ein optimales Kraftfeld für ρ_l und $\Delta_v H$ gefunden. Eine sukzessive Optimierung verschiedener Zielgrößen ist nicht zu empfehlen: Wird mit der Siededichte begonnen, so resultieren zwar ähnliche Kraftfeldparameter mit ähnlicher Güte, allerdings ist das Optimierungsproblem schwieriger zu lösen, was zu einer Erhöhung des Rechenaufwands führt. Wird mit der Verdampfungsenthalpie begonnen, so resultieren völlig verschiedene Kraftfeldparameter, die jedoch zum Schluß wieder eine schlechte Verdampfungs-



(a) (b) (c)

Abbildung 5.21: Startkonfiguration (a), äquilibrierte Konfiguration (b) und Äquilibrierung der potentiellen Energie (c) im Falle von Methanol. Die Kantenlänge der Box beträgt 40 nm.

enthalpie liefern. Bei einer sukzessiven Optimierung ist das Risiko viel größer, daß die Verbesserung einer Zielgröße auf Kosten einer anderen geht.

5.1.4 Methanol: Mitoptimierung von Partialladungen und temperaturbasierte Kreuzvalidierung

Das gewählte Modell für Methanol war vom *United-Atom*-Typ: Wasserstoff und Sauerstoff wurden als einzelne Atome betrachtet und CH_3 wurde zu einer Methylgruppe zusammengefaßt. Bindungen und Winkel wurden mithilfe des LINCS-Algorithmus fixiert, so daß ein starres, gewinkeltes Molekül simuliert wurde. Die intramolekularen Kraftfeldparameter und initialen LJ-Parameter stammen aus dem TraPPE-Kraftfeld für primäre, sekundäre und tertiäre Alkohole (Chen u. a., 2001a). Weiterhin wurden drei Punktladungen angenommen, eine hohe negative Partialladung für Sauerstoff und zwei positive für Wasserstoff und CH_3. Die Werte dieser Partialladungen wurden wieder von Kirschner (2010) mit $WOLF_2PACK$ ermittelt. Sämtliche 1,2-Wechselwirkungen wurden exkludiert.

Simulations- und Optimierungseinstellungen Bei Methanol wurden wie bei Phosgen Verdampfungsenthalpie und Siededichte an der Phasenübergangskurve an experimentelle Werte angepaßt. Die anzupassenden Kraftfeldparameter waren die LJ-Parameter von Sauerstoff und CH_3. Die des Wasserstoffs wurden gemäß dem TraPPE-Kraftfeld konstant auf 0 gesetzt. Simuliert wurde stets an der Phasenübergangskurve zwischen Flüssigkeit und Gas zu den Temperaturen 285, 290, 295, 300, 305, 310 und 315 K. Die entsprechenden experimentellen Dampfdrücke (0.08, 0.11, 0.14, 0.19, 0.24, 0.31 und 0.39 bar) sind aus der NIST-Datenbank (NIST, 2011) entnommen.
Tabelle 5.8 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Methanol. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen und Toleranzen für die simulierten Zielgrößen angegeben.

Abbildung 5.21(a) zeigt die Startkonfiguration der Simulationsbox: In einem Würfel mit einer

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	H, O, CH_3	*United-Atom*-Modell
	Dipole/Quadrupole	–	
	Molare Masse	32.041 g/mol	m(H) = 1.008 g/mol, m(O) = 16.00 g/mol $m(CH_3)$ = 15.034 g/mol
	Molekülstruktur	gewinkelt	polare OH-Gruppe
	Intramolekulares Kraftfeld	2 starre Bindungen (LINCS)	Quelle: TraPPE-Kraftfeld
		1 starrer Winkel (LINCS)	(Chen u. a., 2001a)
	Ladungen	3 Punktladungen	Quelle: $WOLF_2PACK$ oder werden mitoptimiert
	Besonderheiten	1,2-Wechselwirkungen exkludiert	
Simulationsbox	Startkonfiguration	kubisch	siehe Abbildung 5.21(a)
	Anzahl Moleküle	1000	
	Kantenlänge	40 nm	
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung	100000	Schrittweite: 0.2 fs
	Präääquilibrierung	750000	Schrittweite: 2 fs
	Äquilibrierung	250000	Schrittweite: 2 fs
	Produktion	25000	Schrittweite: 2 fs
Optimierungsrelevantes	zu optimierende Parameter	σ(O), $\sigma(CH_3)$,	Quelle Startwerte: TraPPE-Kraftfeld
		ε(O), $\varepsilon(CH_3)$	(Chen u. a., 2001a)
		(, q(O), $q(CH_3)$)	Quelle Startwerte: $WOLF_2PACK$
	Temperaturen	285, 290, 295, 300, 305, 310, 315	in K
	Dampfdrücke	0.08, 0.11, 0.14, 0.19, 0.24, 0.31, 0.39	in bar, Quelle: NIST-Datenbank (NIST, 2011)
Ausgaben:			
Zielgrößen	experimentell	$\Delta_v H$, ρ_l	Quelle: NIST-Datenbank (NIST, 2011)
	simuliert	$\Delta_v H$: Gleichung (C.3)	Approximation (ideales Gas)
		ρ_l: Gleichung (C.8)	
	Toleranzen	$\Delta_v H$: 1%	
		ρ_l: 0.5%	

Tabelle 5.8: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Methanol.

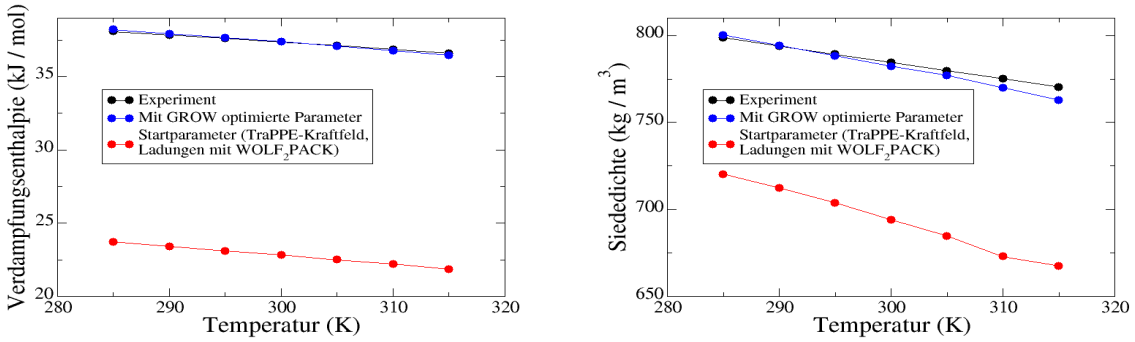
Kantenlänge von 40 nm wurden 1000 Methanolmoleküle auf einem regulären kubischen Gitter so positioniert, daß keine Überlappungen auftraten. Es stellte sich allerdings heraus, daß auch nach einer Energieminimierung eine sehr lange Prä-Prä-Äquilibrierung (100000 Zeitschritte à 0.2 fs) notwendig war, um zu vermeiden, daß das System in einen mechanisch instabilen Zustand überging. Das initiale Kraftfeld war äußerst schlecht, da die LJ-Parameter des TraPPE-Kraftfelds mit durch $WOLF_2PACK$ ermittelten Partialladungen kombiniert wurden. Diese Kombination war im Falle von Methanol ungeeignet, so daß die Simulation mit diesem Kraftfeld erschwert war. Die Präääquilibrierung war daher sicherheitshalber ebenfalls 750000 Zeitschritte lang, analog zu Phosgen. Das Äquilibrierungszeitfenster bestand dafür nur aus 250000 Zeitschritten, ebenfalls analog zu Phosgen. Nach meistens nur einem Äquilibrierungszyklus waren die Äquilibrierungskriterien (3.70) für potentielle Energie, Siededichte und nichtbindende Energie erfüllt. Abbildung 5.21(b) zeigt eine äquilibrierte Konfiguration: Das System ist dichter und ungeord-

k	$x^{(k)}$	MAPE $\Delta_v H$	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
Steilster Abstieg, $h = 0.02$, $\Omega = (50, 95)$						
0	0.30200 0.37500	39.05%	11.74%	1.1657	32.0705 19.1451	37.45
	0.77325 0.81482				-1.0353 -2.4970	
1	0.27548 0.35911	9.99%	9.63%	0.1351	8.4845 -7.3465	11.23
	0.77411 0.81699				0.2017 -0.1473	
2	0.27201 0.36202	5.51%	9.03%	0.0786	3.0832 -7.9204	8.50
	0.77403 0.81695				0.2889 0.0151	
3	0.26849 0.37131	1.71%	3.88%	0.0127	0.8849 -3.3892	3.50
	0.77369 0.81693				0.0589 0.1058	
4	0.26644 0.37879	0.25%	0.51%	$2.7{\cdot}10^{-4}$	0.6713 0.5610	0.88
	0.77356 0.81670				0.0040 -0.0221	
5	0.26644 0.37879	0.22%	0.41%	$2.5{\cdot}10^{-4}$	0.6245 0.5077	0.80
	0.77356 0.81670				0.0029 -0.0106	
Polak-Ribière, $h = 0.01$, $\Omega = (15, 40)$, längere Simulationen						
5	0.26644 0.37879	0.32%	0.49%	$3.5{\cdot}10^{-4}$	0.8359 0.5511	1.00
	0.77356 0.81670				0.0049 -0.0206	
6	0.26635 0.37873	0.25%	0.45%	$3.0{\cdot}10^{-4}$	0.5626 0.4939	0.75
	0.77356 0.81670				-0.0068 -0.0254	
7	0.26631 0.37870	0.19%	0.42%	$2.3{\cdot}10^{-4}$	0.3315 0.4001	0.52
	0.77356 0.81670				-0.0004 -0.0126	
8	0.26628 0.37867	0.20%	0.38%	$2.1{\cdot}10^{-4}$	0.1547 0.3203	0.36
	0.77356 0.81670				0.0005 -0.0076	

Tabelle 5.9: Optimierungsergebnisse für Methanol im Falle von $\Delta_v H$ und ρ_l: Innerhalb von acht Iterationen mit der Methode des steilsten Abstiegs wurde die Fehlerfunktion um etwa fünf Größenordnungen verkleinert. Die Verdamfungsenthalpie wurde um etwa den Faktor 200 und die Siededichte um etwa den Faktor 30 verbessert. Beide Größen wurden simultan optimiert, ohne daß eine Eigenschaft im Laufe der Optimierung verschlechtert wurde. Es ist zu beachten, daß bei den ersten fünf Schritten nur 250000 Äquilibrierungs- und 25000 Produktionsschritte verwendet wurden, was zu schlechten Statistiken geführt hat. Wegen $x^{(4)} = x^{(5)}$ wurde in diesem Schritt nur noch Rauschen reproduziert. Daher wurden von da an längere Simulationen durchgeführt und das anfangs sehr große zulässige Gebiet sowie h verkleinert, was noch Änderungen in der fünften Nachkommastelle bewirkte. Das erhaltene Kraftfeld ist in Bezug auf $\Delta_v H$ und ρ_l nahezu optimal, da alle Zielgrößen zu allen Temperaturen in ihren spezifischen Toleranzbereichen liegen, außer im Falle von ρ_l: Bei $T = 315$ K liegt der Fehler bei -0.99%. Die Steigung der Zielgrößen in Abhängigkeit von der Temperatur konnte nicht reproduziert werden. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{O})$ und $\sigma(\mathrm{CH_3})$ sowie $\varepsilon(\mathrm{O})$ und $\varepsilon(\mathrm{CH_3})$, das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{O})^{(k)}, \sigma(\mathrm{CH_3})^{(k)}, \varepsilon(\mathrm{O})^{(k)}, \varepsilon(\mathrm{CH_3})^{(k)}\right)$.

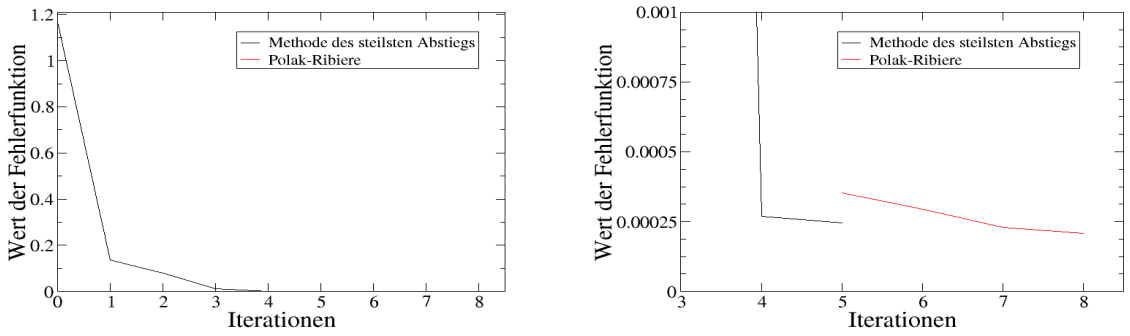
net. Abbildung 5.21(c) zeigt die Äquilibrierung der potentiellen Energie: Aufgrund der sehr langen Prä-Prä-Äquilibrierungsphase sank die potentielle Energie zu Beginn der Prääquilibrierung nur noch geringfügig ab und blieb dann weitestgehend konstant, auch während der viel kürzeren Äquilibrierungsphase. Aufgrund des sehr hohen Simulationsaufwands zum Erhalt einer äquilibrierten Konfiguration wurde die Produktionszeit um den Faktor 10^{-1} verringert. Dadurch verschlechtert sich zwar die Statistik, allerdings kann dadurch gleichzeitig überprüft werden, ob die numerischen Optimierungsverfahren zu Beginn der Optimierung auch Rauschen größeren Ausmaßes tolerieren können. Es handelte sich somit um einen Härtetest für die gradientenbasierten Algorithmen. In der Nähe des Minimums wurde die Produktionszeit um den Faktor 10 erhöht.

Optimierungsergebnisse für $\Delta_v H$ und ρ_l Tabelle 5.9 und Abbildung 5.22 zeigen die Optimierungsergebnisse im Falle von $\Delta_v H$ und ρ_l. Verwendet wurde zunächst die Methode des steilsten Abstiegs mit $h = 0.01$, $\zeta_A = 0.2$ und $\forall_{i,T}\ w_{i,T} = 1$. Auch hierbei bestand die Motivation darin, alle Eigenschaften als gleichwertig zu betrachten. Der zulässige Bereich war zunächst analog zu Phosgen $\Omega = (30, 80)$. Da mit diesen Einstellungen die Konvergenz zu langsam war, wurden



(a) (b)

Abbildung 5.22: Optimierung von $\Delta_v H$ (a) und ρ_l (b) im Falle von Methanol. Die Verdampfungsenthalpie konnte optimale reproduziert werden, wohingegen bei der Siededichte sowohl qualitativ als auch quantitativ noch Verbesserungsmöglichkeiten bestanden.



(a) (b)

Abbildung 5.23: Entwicklung der Fehlerfunktion im Falle von Methanol ($\Delta_v H$,ρ_l) unterhalb der Schwellenwerte 1.2 (a) und 10^{-3} (b). Innerhalb von acht Iterationen wurde diese um etwa fünf Größenordnungen verkleinert. Die Unstetigkeitsstelle ist darauf zurückzuführen, daß die Simulation bei $x^{(5)}$ neu durchgeführt wurde.

nach zwei Iterationen, die hier jedoch nicht angegeben sind, $\Omega = (50, 95)$ und $h = 0.02$ gesetzt, zumal den aufgrund der kürzeren Simulationszeit höheren statistischen Unsicherheiten mit einem größeren h entgegengewirkt werden konnte. Dies bewirkte eine drastische Verbesserung bereits im ersten Schritt. Im Laufe der Optimierung wurden die Energieparameter (ε) im Vergleich zu den Längenparametern (σ) nur unwesentlich verändert, was auch an den Gradienten in Tabelle 5.9 festzustellen ist. Am meisten änderte sich dabei σ(O).
Innerhalb von nur acht Iterationen konnte die Fehlerfunktion um etwa fünf Größenordnungen verkleinert werden, wobei der MAPE-Wert in Bezug auf $\Delta_v H$ um etwa den Faktor 200 und der in Bezug auf ρ_l um etwa den Faktor 30 verkleinert wurde. Beide Eigenschaften wurden simultan verbessert, das heißt, die MAPE-Werte waren für beide Eigenschaften monoton fallend. Abbildung 5.23 zeigt die Entwicklung der Fehlerfunktion. Zu Beginn der Optimierung wurden nur 250000 Äquilibrierungs- und 25000 Produktionsschritte verwendet. Bis zur vierten Iteration konnten dadurch trotz des hohen Ausmaßen an statistischem Rauschen signifikante Verbesse-

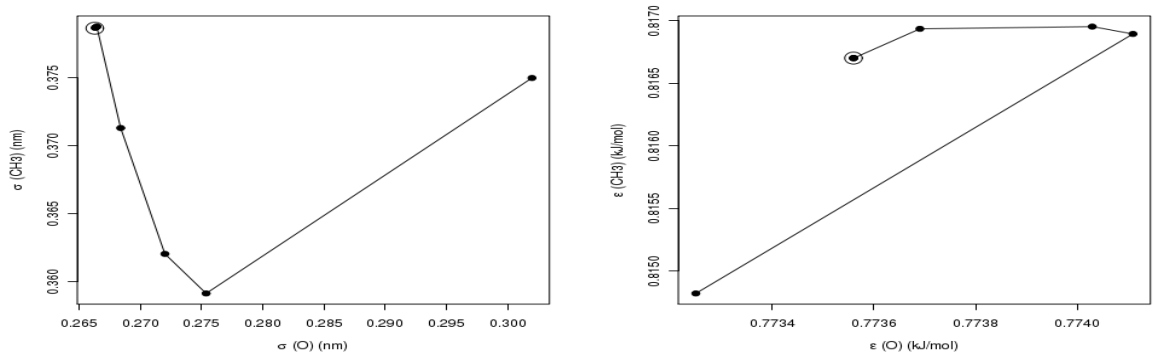
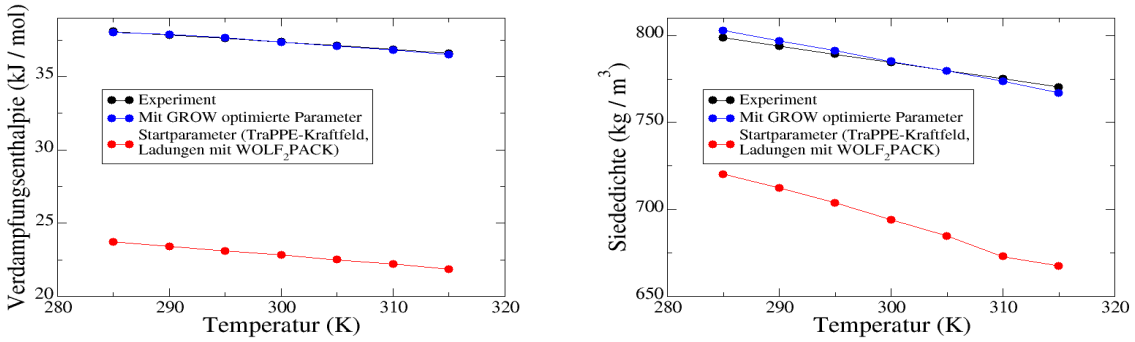


Abbildung 5.24: Entwicklung der LJ-Parameter im Falle von Methanol ($\Delta_v H$,ρ_l): σ(O) und $\sigma(CH_3)$ (a) sowie ε(O) und $\varepsilon(CH_3)$ (b). Die unausgefüllten Kreise zeigen die optimalen Parameter an. GROW läuft auch hier wieder zunächst in eine falsche Richtung und trifft erst später näherungsweise das Minimum.

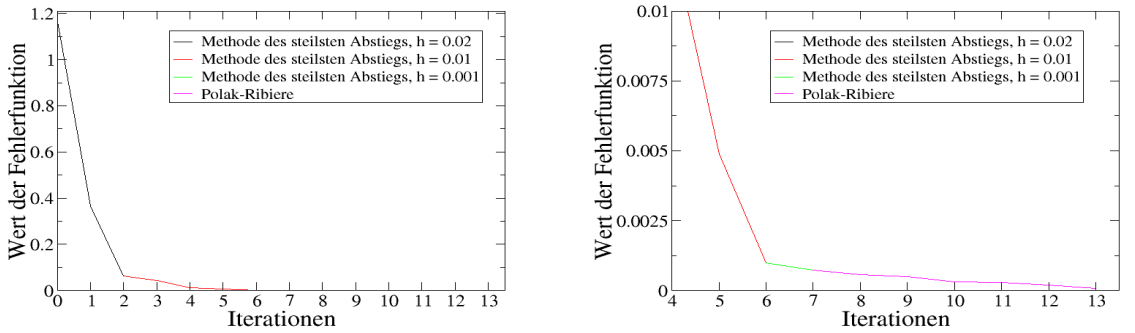
rungen erzielt werden. Danach wurden 500000 Äquilibrierungs- und 250000 Produktionsschritte eingestellt, um die Unsicherheiten zu reduzieren. Dadurch änderten sich σ(O) und $\sigma(CH_3)$ noch in der fünften Nachkommastelle, was jedoch noch signifikante Verbesserungen hervorrief. Die Änderung der Simulationszeit ist der Grund für die Unstetigkeitsstelle in Abbildung 5.23. Das vorliegende Optimierungsproblem ist somit ein Beispiel dafür, daß die Simulationslänge dynamisch gewählt, das heißt erst im Laufe der Optimierung erhöht werden kann.
Für die Gradientenberechnung waren stets vier Simulationen erforderlich. Die maximale Anzahl an Armijo-Schritten wurde wie bei Benzol und Phosgen auf 10 gesetzt. Insgesamt wurden acht Iterationen durchgeführt, wobei bei $k = 0$ eine, bei $k = 1, 2, 3$ zwei und $k = 4$ acht Armijo-Schritte notwendig waren. Die Verbesserung von $x^{(4)}$ nach $x^{(5)}$ war wie bereits erwähnt nur zufällig. Daher war auch eine derart hohe Anzahl an Armijo-Schritten erforderlich. Bis hierhin waren inklusive der Gradientenberechnung für $k = 5$ und den neun anschließenden Armijo-Schritten 50 (effektiv 41) Simulationen erforderlich. Im zweiten Teil der Optimierung waren für $k = 5$ zwei, für $k = 6$ drei und für $k = 7$ vier Armijo-Schritte vonnöten. Insgesamt ergaben sich 35 (effektiv 26) Simulationen, woraus eine Gesamtanzahl von 85 (effektiv 67) Simulationen resultierte. Abbildung 5.22 zeigt, daß beide Eigenschaften nach der gradientenbasierten Optimierung bis auf ihre Toleranzen mit den experimentellen Daten übereinstimmten, außer bei der Siededichte für $T = 315$ K, die um -0.99% vom Experiment abwich. Der Trend der experimentellen Kurven konnte für die Siededichte nicht reproduziert werden. Im folgenden wird gezeigt, daß bessere Zielgrößen erhalten werden können, wenn die Partialladungen mitoptimiert werden.

Abbildung 5.24 zeigt die Entwicklung der Kraftfeldparameter im Laufe der Optimierung (Abbildung 5.24(a) bezieht sich auf σ und Abbildung 5.24(b) auf ε): ε wurde in beiden Fällen zunächst vergrößert und dann verkleinert. Bei $\sigma(CH_3)$ war es genau umgekehrt, während σ(O) stetig verkleinert wurde. Die drastische Verbesserung im ersten Schritt ist darauf zurückzuführen, daß die Fehlerfunktion bei $x^{(0)}$ äußerst steil war. Somit konnte das Verfahren des steilsten Abstiegs schnell eine stark verbesserte Lösung finden. Allerdings konnte das Verfahren nicht erkennen, daß sich die Richtung des Gradienten in diesem Bereich verändert. Daher lief die



(a) (b)

Abbildung 5.25: Optimierung von $\Delta_v H$ (a) und ρ_l (b) im Falle von Methanol (mit Partialladungen). Beide Eigenschaften konnten optimal reproduziert werden.



(a) (b)

Abbildung 5.26: Entwicklung der Fehlerfunktion im Falle von Methanol ($\Delta_v H$,ρ_l) unterhalb der Schwellenwerte 1.2 (a) und 10^{-2} (b) (mit Partialladungen). Diese konnte innerhalb von sechs Iterationen um etwa den Faktor 15000 verkleinert werden.

Optimierung zunächst in eine falsche Richtung. Von da an war das Verfahren auf das Minimum gerichtet, wobei bei den letzten drei Iterationen nur noch minimale Änderungen zu verzeichnen waren.

Mitoptimierung der Partialladungen Tabelle 5.10 und Abbildung 5.25 zeigen die Optimierungsergebnisse im Falle von $\Delta_v H$ und ρ_l, wobei diesmal die Partialladungen mitoptimiert wurden. Verwendet wurde dabei die Methode des steilsten Abstiegs mit $h = 0.02$, $\zeta_A = 0.2$ und $\forall_{i,T}\ w_{i,T} = 1$. Der zulässige Bereich war wie oben zunächst $\Omega := (50, 95, 30)$ und damit wiederum recht großzügig gewählt. Im Laufe der Optimierung wurden die Energieparameter (ε) im Vergleich zu den Längenparametern (σ) auch bei diesem Optimierungsproblem nur unwesentlich verändert. Am meisten änderte sich dabei σ(O), aber auch die Partialladungen änderten

k	$x^{(k)}$	MAPE $\Delta_v H$	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
	Steilster Abstieg, $h = 0.02$, $\Omega = (50, 95, 30)$					
0	0.30200 0.37500 0.77325 0.81482 -0.55900 0.21210 0.34690	39.01%	11.75%	1.1635	32.28949 19.45199 -0.89093 -2.23397 21.03929 12.81604	45.10
1	0.27016 0.35582 0.77413 0.81702 -0.57975 0.19946 0.38029	17.74%	14.35%	0.3647	-45.57230 -18.94659 0.03275 1.71468 -17.11106 -10.41660	53.29
2	0.27851 0.35929 0.77412 0.81671 -0.57661 0.20137 0.37524	0.43%	9.34%	0.0613	-6.49257 -9.52190 0.14813 0.27983 -1.32428 -0.02479	11.60
	Steilster Abstieg, $h = 0.01$, $\Omega = (30, 40, 20)$					
3	0.27970 0.36110 0.77412 0.81666 -0.57640 0.20135 0.37505	1.74%	7.66%	0.0433	-0.91932 -6.84590 0.06170 0.11721 0.73089 1.33172	7.07
4	0.28126 0.37270 0.77402 0.81646 -0.57764 0.19909 0.37855	4.21%	0.38%	0.0127	7.89081 1.28756 0.10428 -0.21201 3.48852 2.43677	9.06
5	0.28041 0.37256 0.77401 0.81648 -0.57801 0.19883 0.37919	2.59%	0.35%	0.0049	4.78118 0.50447 0.07031 -0.12298 2.14639 1.56173	5.49
6	0.27959 0.37247 0.77400 0.81650 -0.57838 0.19856 0.37982	0.99%	0.57%	$9.8{\cdot}10^{-4}$	1.63620 -0.25565 0.01684 -0.03072 0.78632 0.60009	1.93
	Steilster Abstieg, $h = 0.001$, $\Omega = (15, 20, 10)$					
7	0.27949 0.37248 0.77399 0.81649 -0.57843 0.19852 0.37991	0.76%	0.58%	$7.2{\cdot}10^{-4}$	1.34880 -0.13612 0.11464 0.09428 0.66329 0.55715	1.62
	Polak-Ribière, $h = 0.001$, $\Omega = (15, 20, 10)$, längere Simulationen					
8	0.27938 0.37249 0.77398 0.81648 -0.57848 0.19848 0.38001	0.68%	0.50%	$5.6{\cdot}10^{-4}$	1.28709 -0.18541 0.14557 0.12209 0.52614 0.62400	1.55
9	0.27928 0.37250 0.77397 0.81647 -0.57852 0.19843 0.38009	0.44%	0.63%	$5.0{\cdot}10^{-4}$	0.68255 -0.44129 0.00467 -0.00110 0.31051 0.29943	0.92
10	0.27891 0.37284 0.77399 0.81649 -0.57870 0.19828 0.38042	0.24%	0.53%	$3.1{\cdot}10^{-4}$	-0.69893 -0.51970 0.00171 0.04509 -0.22224 -0.11686	0.91
11	0.27892 0.37299 0.77399 0.81649 -0.57871 0.19826 0.38045	0.23%	0.50%	$2.9{\cdot}10^{-4}$	-0.73387 -0.52997 -0.02789 -0.01440 -0.27036 -0.15074	0.96
12	0.27906 0.37310 0.77400 0.81649 -0.57866 0.19829 0.38037	0.13%	0.41%	$1.9{\cdot}10^{-4}$	-0.05107 -0.33831 -0.02120 -0.03008 -0.00095 0.01087	0.34
13	0.27901 0.37355 0.77403 0.81654 -0.57871 0.19825 0.38046	0.11%	0.26%	$8.2{\cdot}10^{-5}$	0.04158 -0.07637 0.00472 0.00432 0.03172 0.02605	0.10

Tabelle 5.10: Optimierungsergebnisse für Methanol im Falle von $\Delta_v H$ und ρ_l. Die Partialladungen wurden hierbei mitoptimiert: Innerhalb von 13 Iterationen mit der Methode des steilsten Abstiegs und dem Polak-Ribière-Verfahren wurde die Fehlerfunktion um etwa den Faktor 15000 verkleinert. Zunächst wurden beide Eigenschaften simultan optimiert, wobei die Verdampfungsenthalpie bei $x^(2)$ schon optimal war. Dann verbesserte sich die Siededichte auf Kosten der Verdampfungsenthalpie und später wieder die Verdampfungsenthalpie auf Kosten der Siededichte, bis sich beide Eigenschaften einpendelten. Das erhaltene Kraftfeld ist in Bezug auf $\Delta_v H$ und ρ_l optimal, da alle Zielgrößen zu allen Temperaturen in ihren spezifischen Toleranzbereichen liegen. Der Trend der experimentellen Kurve wird für beide Eigenschaften sehr gut wiedergegeben, diesmal auch für die Siededichte. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{O})$ und $\sigma(\mathrm{CH_3})$, $\varepsilon(\mathrm{O})$ und $\varepsilon(\mathrm{CH_3})$ sowie $q(\mathrm{O})$, das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{O})^{(k)}, \sigma(\mathrm{CH_3})^{(k)}, \varepsilon(\mathrm{O})^{(k)}, \varepsilon(\mathrm{CH_3})^{(k)}, q(\mathrm{O})^{(k)}, q(\mathrm{CH_3})^{(k)}\right)$. In der Tabelle ist der Vollständigkeit halber zusätzlich $q(\mathrm{H})^{(k)}$ angegeben.

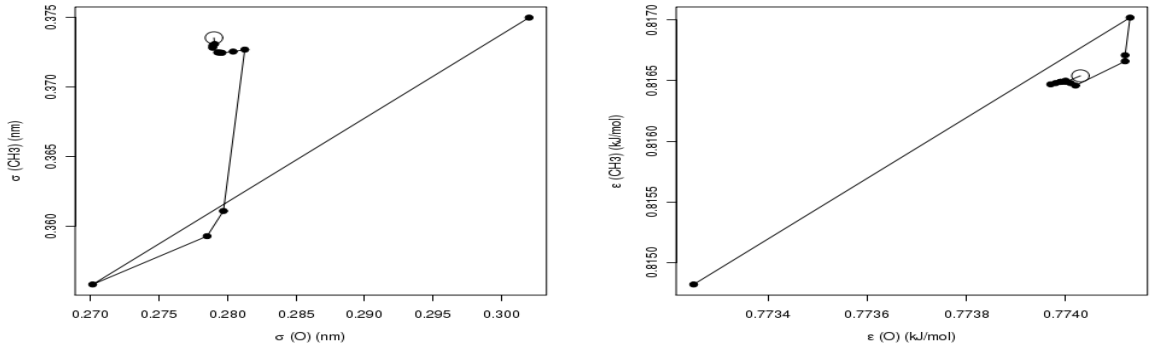


Abbildung 5.27: Entwicklung der LJ-Parameter im Falle von Methanol ($\Delta_v H$,ρ_l) bei Mitoptimierung der Partialladungen: σ(O) und $\sigma(CH_3)$ (a) sowie ε(O) und $\varepsilon(CH_3)$ (b). Die Kreise zeigen die optimalen Parameter an. GROW läuft auch hier wieder zunächst in eine falsche Richtung und trifft erst später näherungsweise das Minimum. Man sieht weiterhin, daß in der Nähe des Minimums die Parameter nur noch unwesentlich verändert wurden. Jede einzelne Iteration lieferte dennoch signifikante Verbesserungen in Bezug auf die Zielgrößen.

sich deutlich. Das zulässige Gebiet wurde bereits nach der zweiten Iteration verkleinert, da die Armijo-Schrittweitensteuerung zu langsam konvergierte. Nach der sechsten Iteration wurde $h = 0.001$ gesetzt, da der Gradient in die falsche Richtung zeigte und somit nur durch Zufall Verbesserungen erzielt werden konnten. Erst dann wurde ein Polak-Ribière-Verfahren nachgeschaltet.
Innerhalb von 13 Iterationen konnte die Fehlerfunktion etwa um den Faktor 15000 verkleinert werden. Zunächst wurden beide Eigenschaften simultan optimiert. Dabei gelangte die Verdampfungsenthalpie bei $x^{(2)}$ bereits in den optimalen Bereich. Dann verbesserte sich die Siededichte auf Kosten der Verdampfungsenthalpie, und später verhielt es sich umgekehrt. Abbildung 5.26 zeigt die Entwicklung der Fehlerfunktion. Bei dieser Optimierung wurden Äquilibrierungs- und Produktionszeit bei $x^{(7)}$ erhöht, das heißt zu Beginn des Polak-Ribière-Verfahrens. Hierbei waren jedoch nur Änderungen in der fünften Nachkommastelle zu verzeichnen.
Der Rechenaufwand wird hier nicht detailliert aufgeführt. Es waren allerdings weit über 100 Simulationen zum Erhalt des optimalen Kraftfelds erforderlich. Abbildung 5.25 zeigt, daß beide Eigenschaften nach der gradientenbasierten Optimierung bis auf ihre Toleranzen mit den experimentellen Daten übereinstimmten. Der Trend der experimentellen Kurven konnte diesmal ebenfalls für beide Eigenschaften reproduziert werden.

Abbildung 5.27 zeigt die Entwicklung der Kraftfeldparameter (Abbildung 5.27(a) bezieht sich auf σ und Abbildung 5.27(b) auf ε) und Abbildung 5.28 die von q(O) im Laufe der Optimierung: ε wurde in beiden Fällen zunächst vergrößert und dann verkleinert. Bei σ war es genau umgekehrt, und die (negative) Partialladung von q(O) wurde insgesamt verkleinert. Es zeigt sich auch hier wieder, daß die Fehlerfunktion zu Beginn aufgrund des großen h stark verbessert werden konnte, die Optimierung dadurch zunächst jedoch in die falsche Richtung ging. Es resultierten hierbei andere LJ-Parameter als vorher, was auch nicht verwunderlich ist, da die Partialladungen ja in diesem Fall nicht konstant blieben. Der Optimierungsverlauf war in diesem

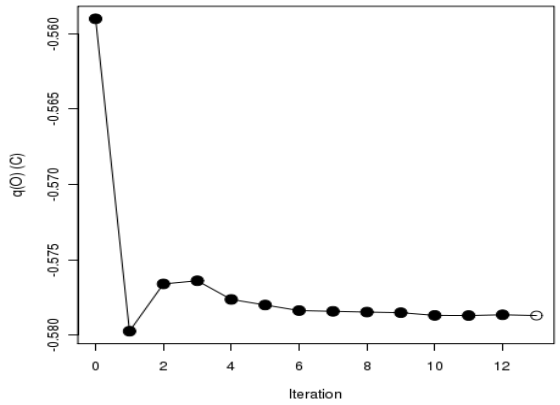


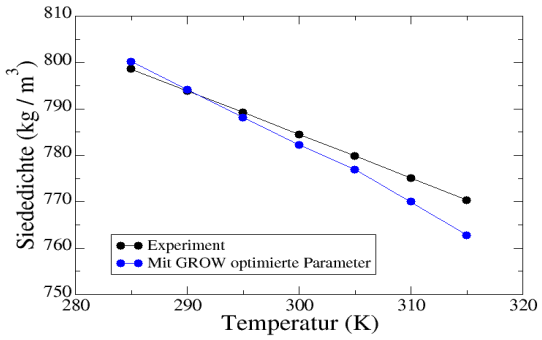
Abbildung 5.28: Partialladung des Sauerstoffs $q(\mathrm{O})$ im Falle von Methanol ($\Delta_v H$,ρ_l). Der Kreis zeigt die optimale Partialladung an. Trotz der nur sehr geringen Änderungen der Partialladungen in der Nähe des Minimums konnte nur durch durch deren Mitoptimierung ein optimales Kraftfeld erzielt werden.

Fall jedoch nicht so gerichtet wie vorher. Anscheinend irrten die gradientenbasierten Verfahren um das Minimum herum. Dies ist darauf zurückzuführen, daß hierbei zunächst versucht wurde, h zu verkleinern anstatt die Simulationszeit zu vergrößern. Zu Beginn der Optimierung können zwar kleinere Produktionszeiten verwendet werden, allerdings sollte dies geändert werden, sobald das Verfahren mit $h = 0.01$ keine Verbesserungen mehr liefert.

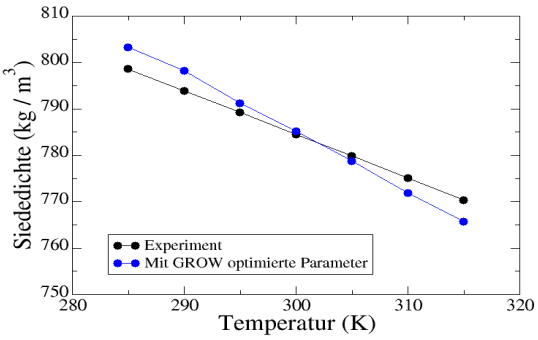
Die Mitoptimierung der Partialladungen führte zwar zu deutlich mehr Rechenaufwand, allerdings auch zu einem deutlich besseren Kraftfeld, was auch anhand von Abbildung 5.25(b) zu erkennen ist, wo der Trend der experimentellen Kurve deutlich besser wiedergegeben werden konnte als in Abbildung 5.22(b). LJ-Parameter und Partialladungen können nicht grundsätzlich als unabhängig betrachtet werden (vergleiche auch Hünenberger u. van Gunsteren (1997)). Wie anhand von Methanol zu belegen war, empfiehlt es sich, die Partialladungen in die Optimierung miteinzubeziehen, sofern der Rechenaufwand dadurch nicht inakzeptabel hoch wird. Die Hinzunahme der Partialladungen kann auch erst in der Nähe des Minimums erfolgen.

Kreuzvalidierung über betrachteten Temperaturbereich Die Studie bezüglich Methanol schließt mit einer Kreuzvalidierung über den betrachteten Temperaturbereich. Unterschieden wurden dabei die folgenden drei Fälle:

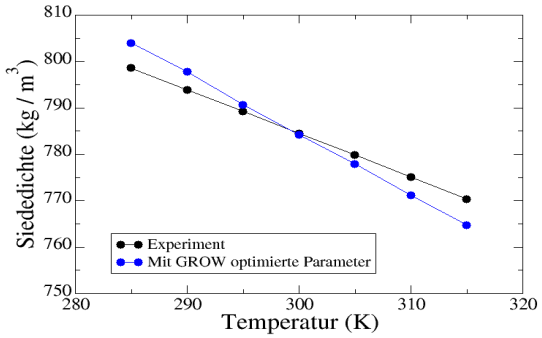
1. Trainingsmenge: T = 285, 295, 305, 315 K, Validierungsmenge: T = 290, 300, 310 K, das heißt, es wurde zu jeder zweiten Temperatur optimiert und das erhaltene Kraftfeld bezüglich der Zielgrößen zu den drei verbleibenden mittleren Temperaturen evaluiert,
2. Trainingsmenge: T = 285, 290, 295, 300 K, Validierungsmenge: T = 305, 310, 315 K, das heißt, es wurde zu den vier niedrigsten Temperaturen optimiert und das erhaltene



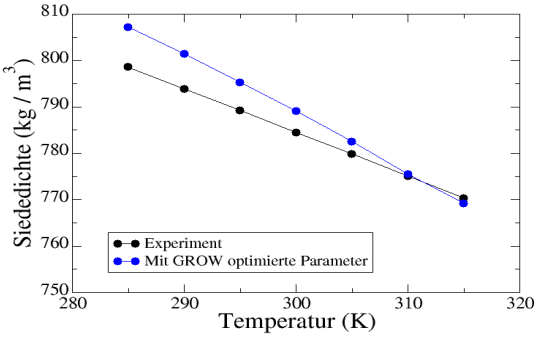
(a)



(b)



(c)



(d) Optimale Siededichte zu den vier höchsten Temperaturen.

Abbildung 5.29: Kreuzvalidierung bei Methanol über den betrachteten Temperaturbereich: Optimale Siededichte zu allen Temperaturen (a), zu jeder zweiten Temperatur (b), zu den vier niedrigsten Temperaturen (c) und zu den vier höchsten Temperaturen (d). Der Trend wurde bei der Betrachtung von nur vier Temperaturen stets deutlich schlechter wiedergegeben als bei der Betrachtung aller Temperaturen. Die Extrapolation in den niedrigen Temperaturbereich lieferte keine optimalen Siededichten.

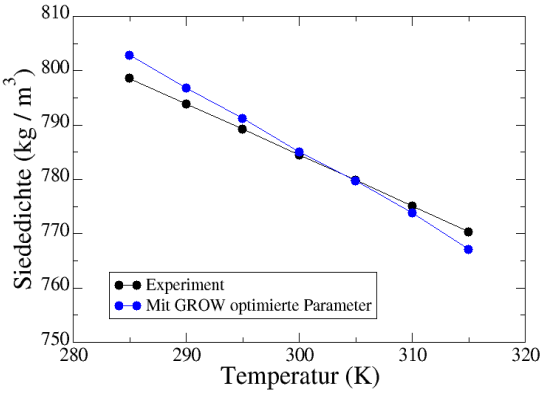


Abbildung 5.30: Optimale Siededichten zu allen Temperaturen. Die Partialladungen wurden hierbei mitoptimiert. Nur in diesem Fall konnten optimale Siededichte erhalten werden, und der Trend konnte am besten reproduziert werden.

KV	# It.	# Eval.	σ(O)	σ(CH_3)	ε(O)	ε(CH_3)
1	11	79	0.26634	0.37800	0.77345	0.81614
2	9	64	0.26639	0.37821	0.77348	0.81651
3	8	54	0.26634	0.37733	0.77369	0.81660
alle Temp.	8	67	0.26628	0.37867	0.77356	0.81670
alle Temp., Lad.	13	> 100	0.27901	0.37355	0.77403	0.81654

Tabelle 5.11: Finale Kraftfeldparameter für jeden der drei Bestandteile der Kreuzvalidierung (KV). Angegeben sind jeweils die Anzahl an Iterationen (# It.) und an effektiven Funktionsauswertungen (# Eval.) sowie die finalen Kraftfeldparameter für jeden Atomtyp (σ in nm und ε in kJ/mol). Obwohl diese sich bei den drei Bestandteilen kaum voneinander unterschieden, war der Einfluß der unterschiedlichen Parametervektoren auf die Zielgrößen signifikant. Zum Vergleich sind die Kraftfeldparameter, die aus der Optimierung zu allen sieben Temperaturen erhalten wurden, angegeben sowie die, die aus derjenigen Optimierung resultierten, welche auch die Partialladungen miteinbezogen hatte. Nur in diesem Fall konnte ein in Bezug auf $\Delta_v H$ und ρ_l optimales Kraftfeld erhalten werden. Die Kraftfeldparameter unterschieden sich deutlich stärker von denen in den anderen vier Fällen.

Kraftfeld bezüglich der Zielgrößen zu den drei höchsten Temperaturen evaluiert,

3. Trainingsmenge: $T = 300, 305, 310, 315$ K, Validierungsmenge: $T = 285, 290, 295$ K, das heißt, es wurde zu den vier höchsten Temperaturen optimiert und das erhaltene Kraftfeld bezüglich der Zielgrößen zu den drei niedrigsten Temperaturen evaluiert.

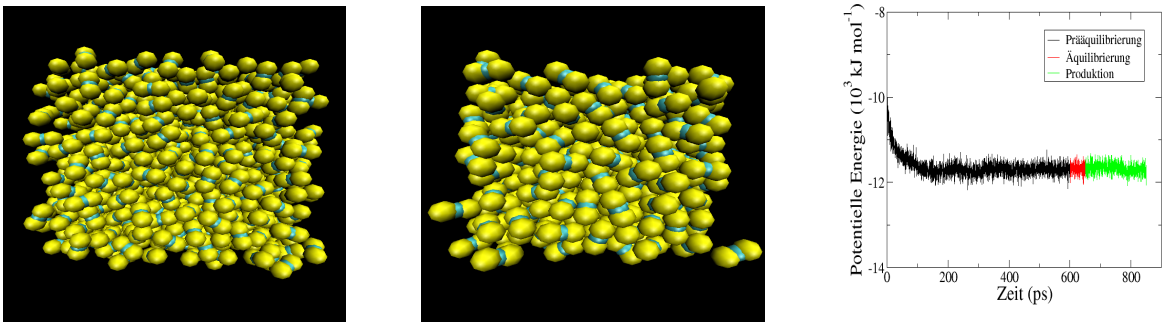
Die Optimierungen wurden wie oben mit der Methode des steilsten Abstiegs, verschiedenen Werten von h und verschiedenen Produktionszeiten durchgeführt. Die Simulationen mit den jeweils optimalen Kraftfeldparametern beinhalteten stets die langen Produktionszeiten, so daß Zufallsergebnisse ausgeschlossen werden konnten. Auf die einzelnen Optimierungsdetails wird im folgenden nicht eingegangen, entscheidend sind hierbei lediglich die Güte des erhaltenen Kraftfelds in Bezug auf die restlichen Temperaturen und der Rechenaufwand, sprich die Anzahl an durchzuführenden Iterationen und Funktionsauswertungen. In allen drei Fällen war die Verdampfungsenthalpie zu allen Temperaturen optimal. Es ergaben sich lediglich Unterschiede in Bezug auf die Siededichte: Abbildung 5.29 zeigt die durch die Optimierungen erhaltene Siededichte für alle drei Fälle im Vergleich zur Siededichte, die aus der Optimierung zu allen Temperaturen erhalten wurde (Abbildung 5.29(a) ist ein Ausschnitt von Abbildung 5.22(b)). Bei den drei Bestandteilen der Kreuzvalidierung ergab sich folgendes:

1. Abbildung 5.29(b): Die Siededichte war zwar durchgehend bis auf statistische Unsicherheiten optimal, allerdings wurde der temperaturabhängige Trend im Vergleich zur Optimierung zu allen Temperaturen etwas schlechter wiedergegeben.
2. Abbildung 5.29(c): Auch hier waren die Siededichte durchgehend bis auf ihre Unsicherheiten optimal. Der Trend ist jedoch auch hier deutlich schlechter.
3. Abbildung 5.29(d): Die Siededichte zu den höheren Temperaturen wich zum Teil weit über 1% vom Experiment ab. Das Kraftfeld war also nicht in Bezug auf alle Temperaturen optimal.

Bezüglich des Rechenaufwands ist zu sagen, daß sich die Anzahl an Simulationen pro Kraftfeldparametervektor in jedem Fall von sieben auf vier reduziert hat. Insgesamt ist daher bei gleicher Anzahl an Funktionswertungen die Anzahl an Simulationen geringer. Da sämtliche Simulationen zu verschiedenen Temperaturen jedoch parallel durchgeführt worden sind, wird auch hier zur Bewertung die effektive Anzahl an Funktionsauswertungen betrachtet. Tabelle 5.11 zeigt die

Anzahl an Iterationen und an effektiven Funktionsauswertungen sowie die erhaltenen Kraftfeldparameter für jeden der drei Bestandteile der Kreuzvalidierung. Die Unterschiede in den Kraftfeldparametern waren geringfügig, allerdings wurde die Güte des Gesamtkraftfelds zu allen Temperaturen signifikant beeinflußt. Allerdings führte jeder Optimierungslauf in denselben Parameterbereich. Bezüglich der Trainingsmenge lagen sämtliche zu optimierenden Zielgrößen stets innerhalb ihrer gewünschten Toleranzbereiche, ohne daß dazu eine sehr hohe Anzahl an Armijo-Schritten erforderlich war, so daß durch die Kreuzvalidierung Zufallsergebnisse bei allen Optimierungsläufen für Methanol ausgeschlossen werden konnten.
Tabelle 5.11 zeigt zum Vergleich ebenfalls diejenigen Kraftfeldparameter, die aus der Optimierung zu allen sieben Temperaturen resultierten. Hierzu waren 8 Iterationen und effektiv 67 Funktionsauswertungen erforderlich. Der Rechenaufwand war also bei vier Temperaturen von ähnlichem Ausmaß, falls die Anzahl an Funktionsauswertungen betrachtet wird, bei Fall 1 war er sogar deutlich höher. Wird jedoch die Anzahl an Simulationen für jede Temperatur einzeln betrachtet, so war der Rechenaufwand deutlich geringer, allerdings ging dies auf Kosten der Güte des Kraftfelds. Besonders war dies bei Fall 3 festzustellen, wo die vier höchsten Temperaturen als Trainingsmenge verwendet wurden. Eine Extrapolation ist grundsätzlich schwieriger als eine Interpolation, und anscheinend ist eine Extrapolation in den niedrigen Temperaturbereichen schwieriger als in den hohen. Eine mögliche physikalische Erklärung hierfür ist die folgende: Zu höheren Temperaturen sind die Wechselwirkungen zwischen den Teilchen geringer als bei niedrigen. Werden also nur schwächere Wechselwirkungen angepaßt, kann das resultierende Kraftfeld stärkere Wechselwirkungen nur schwer vorhersagen. Alles in allem ist es stets sinnvoll, so viele Informationen wie möglich in die Optimierung eines Kraftfelds miteinzubeziehen. Es wäre denkbar, zu Beginn der Optimierung weniger Temperaturen zu betrachten, um Rechenaufwand zu sparen. Die niedrigste und die höchste sollten jedoch auf jeden Fall miteinbezogen werden. Das in diesem Abschnitt beste Ergebnis bezüglich der Siededichte wurde durch die Mitoptimierung der Partialladungen zu allen Temperaturen erhalten, das in Abbildung 5.30 und Tabelle 5.11 nochmals zum Vergleich dargestellt ist. Zwar waren hierzu 13 Iterationen und mehr als einhundert Funktionsauswertungen notwendig, allerdings war dafür die Güte des resultierenden Kraftfelds deutlich besser. Die Parameter unterschieden sich signifikant von denen in den anderen vier Fällen.

Fazit Nur mithilfe der Mitoptimierung der Partialladungen und der Miteinbeziehung aller sieben Temperaturen konnte ein in Bezug auf $\Delta_v H$ und ρ_l optimales Kraftfeld erhalten werden. Die Entscheidung, wie viele und welche Zielgrößen sowie Kraftfeldparameter miteinbezogen werden, liegt stets im Ermessen des Benutzers. Zum Erhalt eines generischen Kraftfelds, welches möglichst viele Eigenschaften, das heißt zum Beispiel Eigenschaften der Flüssigkeit wie Dichte und Transporteigenschaften sowie VLE-Eigenschaften wie Siededichte, Dampfdruck und Verdampfungsenthalpie, reproduzieren soll, sind so viele Informationen wie möglich vonnöten. In diesem Fall ist ein hoher Rechenaufwand unabdingbar. Wird andererseits lediglich ein Kraftfeld benötigt, welches nur eine bestimmte Eigenschaft exakt beschreibt, und wird diese Eigenschaft zu einer Vielzahl an Temperaturen und Drücken mit dem so erhaltenen Kraftfeld berechnet, was im Labor experimentell nicht durchführbar wäre, so sind weniger Informationen und Rechenaufwand erforderlich. Die Art und Anzahl an betrachteten Größen und Zuständen sowie an Kraftfeldparametern bestimmt sich durch die gewünschte Genauigkeit des Kraftfelds. Um das Optimum möglichst exakt zu erreichen, das heißt, um lokale Verfeinerungen in der Nähe des



(a) (b) (c)

Abbildung 5.31: Startkonfiguration (a), äquilibrierte Konfiguration (b) und Äquilibrierung der potentiellen Energie (c) im Falle von CS_2. Die Kantenlänge der Simulationsbox beträgt 50 nm.

Minimums durchzuführen, ist die Anzahl an Temperaturen und Kraftfeldparametern sowie die Simulationszeit zu maximieren. Hier lohnt sich dann ein etwas höherer Rechenaufwand, welcher zu Beginn der Optimierung nicht erforderlich ist.

5.1.5 Kohlenstoffdisulfid: Grenzen der Kraftfeldoptimierung

Kohlenstoffdisulfid (CS_2) ist ein Beispiel dafür, daß nicht immer ein optimales Kraftfeld erhalten werden kann, und zeigt somit die Grenzen der Kraftfeldoptimierung auf. Zum einen war das hier verwendete molekulare Modell nicht geeignet, und zum anderen wurde die Optimierung wie bei Methanol mit dynamischer Simulationslänge durchgeführt, wobei diese im vorliegenden Fall zu spät erhöht wurde. Wie im folgenden zu sehen ist, ist dies gerade bei der Optimierung von Transportgrößen nicht empfehlenswert, da hierbei das Risiko sehr groß ist, daß ein Optimierungsverfahren aufgrund des zu hohen Rauschens keine Erfolge erzielen kann. Allerdings konnte das erhaltene zufriedenstellende Kraftfeld auch mit längerer Produktionszeit nicht mehr verbessert werden, was auf die im folgenden dargelegten Gegebenheiten zurückzuführen ist.

Simulations- und Optimierungseinstellungen Für CS_2 wurde ein sehr naives molekulares Modell angenommen, und zwar ein Stäbchenmodell ohne Berücksichtigung der Delokalisierung der π-Elektronen, die auf das Vorhandensein zweier Doppelbindungen zwischen Schwefel und Kohlenstoff zurückzuführen ist. Da jedoch ein quadrupolares Modell mit *Gromacs* nicht simulierbar ist, war dies die einzige Möglichkeit, CS_2 mit den vorhandenen Mitteln zu modellieren. Auch der Erhalt eines starren linearen Moleküls ist mit *Gromacs* problematisch: Die Bindungen konnten zwar mithilfe des LINCS-Algorithmus (siehe Anhang B) starr gehalten werden, allerdings konnte der Winkel nicht ohne weiteres auf 180° fixiert werden. Dazu mußten virtuelle Zentren zwischen Kohlenstoff und Schwefel eingeführt werden, deren Abstand ebenfalls konstant gehalten wurde. Die Bindungslängen und Kraftkonstanten stammen aus dem Amber-Kraftfeld von Wang u. a. (2004).
Die zu optimierenden Zielgrößen waren Diffusionskoeffizient und Flüssigkeitsdichte bei Atmosphärendruck zu vier verschiedenen Temperaturen: 268.2, 283.2, 298.2 und 313.2 K. Diese Optimierung wird im folgenden und in den Abbildungen mit (D,ρ) markiert. Bei der Suche

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	S, C	*All-Atom*-Modell
	Dipole/Quadrupole	–	zentraler Quadrupol denkbar
	Molare Masse	76.13 g/mol	m(S) = 32.06 g/mol, m(C) = 12.01 g/mol
	Molekülstruktur	linear	unpolar
	Intramolekulares Kraftfeld	2 starre Bindungen (LINCS)	Quelle: Amber-Kraftfeld
		1 starrer Winkel (LINCS)	(Wang u. a., 2004)
	Ladungen	3 Punktladungen	Quelle: $WOLF_2PACK$ oder werden mitoptimiert
	Besonderheiten	virtuelle Zentren	
Simulationsbox	Startkonfiguration	randomisiert	siehe Abbildung 5.31(a)
	Anzahl Moleküle	1000	
	Kantenlänge	50 nm	
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung	5000	Schrittweite: 0.2 fs
	Präääquilibrierung	300000	Schrittweite: 2 fs
	Äquilibrierung	25000	Schrittweite: 2 fs
	Produktion	100000	Schrittweite: 2 fs
Optimierungsrelevantes	zu optimierende Parameter	σ(S), σ(C),	Quelle Startwerte: Amber-Kraftfeld
		ε(S), ε(C)	(Wang u. a., 2004), per Hand nachjustiert
		(, q(S))	Quelle Startwerte: $WOLF_2PACK$
	Temperaturen	268.2, 283.2, 298.2, 313.2	in K
	Druck	1.0	in bar (Atmosphärendruck)
Ausgaben:			
Zielgrößen	experimentell	D, ρ	Quelle: Woolf (1982)
	simuliert	D: Gleichung (C.18)	Einstein-Darstellung
		ρ: Gleichung (C.8)	
	Toleranzen	D: unbekannt	
		ρ: 0.5%	

Tabelle 5.12: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von CS_2.

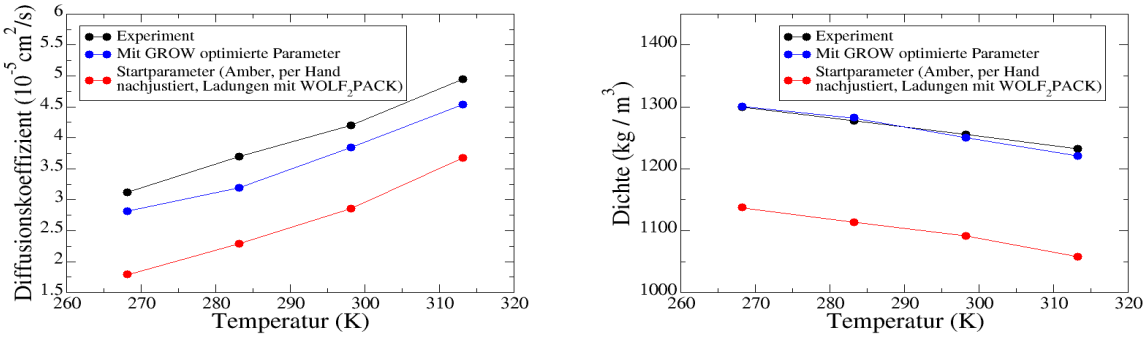
nach geeigneten initialen LJ-Parametern ergab sich ein weiteres Problem: CS_2 ist quadrupolar, was zu eher schwachen intermolekularen Wechselwirkungen führt. Die Kombination der mithilfe von $WOLF_2PACK$ erhaltenen Partialladungen und der LJ-Parameter aus Wang u. a. (2004) führte zu einem mechanisch instabilen System, was auch nicht durch eine verlängerte Prä-Prä-Äquilibrierung in den Griff zu bekommen war. Daher mußten die LJ-Parameter per Hand nachjustiert werden. Dabei wurden ε(S) und ε(C) stark vergrößert, um stärkere LJ-Wechselwirkungen zu realisieren.
Tabelle 5.12 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von CS_2. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen und Toleranzen für die simulierten Zielgrößen angegeben.
Abbildung 5.31(a) zeigt die Startkonfiguration der Simulationsbox: In einem Würfel mit einer Kantenlänge von 50 nm wurden 1000 CS_2-Moleküle so positioniert, daß keine Überlappungen auftraten. Eine Gitterstruktur wäre unphysikalisch, und sie wurde nicht gewählt, da ansonsten unnötiger Simulationsaufwand zu betreiben gewesen wäre, um die Gitterstruktur aufzulösen,

k	$x^{(k)}$	MAPE D	MAPE ρ_l	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
Steilster Abstieg, $h = 0.02$, $\Omega = (40, 80)$						
0	0.38000 0.26600	34.57%	13.16%	0.5636	68.5851 -11.5110	69.81
	1.60000 0.60000				5.9994 0.8796	
1	0.35648 0.26995	14.02%	1.89%	0.0887	8.0011 0.2065	8.58
	1.59794 0.59970				2.9599 -0.8931	
2	0.35404 0.26988	13.14%	0.64%	0.0744	8.3572 -0.8911	8.86
	1.59704 0.59997				2.7810 -0.2585	
3	0.35404 0.26988	13.01%	0.69%	0.0734	8.3370 -0.8431	8.83
	1.59704 0.59997				2.7748 -0.2421	
Mitoptimierung der Partialladungen (Startladung: $q(\mathrm{S})^{(0)} = -0.03310$), $h = 0.01$, $\Omega = (40, 80, 40)$ (Zufall)						
7	0.35371 0.26965	1.30%	1.56%	0.0032	0.6597 0.0296	0.96
	1.59658 0.59979				-0.3629 -0.1253	
	-0.03274 0.06548				-0.1253	
$x^{(7)}$ mit zehnfacher Produktionszeit						
7	0.35371 0.26965	12.29%	0.28%	0.0078	0.7594 0.2953	0.89
	1.59658 0.59979				0.3038 0.1904	
	-0.03274 0.06548				0.0824	

Tabelle 5.13: Optimierungsergebnisse für CS_2 im Falle von D und ρ: Innerhalb von sieben Iterationen mit der Methode des steilsten Abstiegs wurde die Fehlerfunktion um fast drei Größenordnungen verkleinert. Die Diffusionskoeffizienten und Flüssigkeitsdichten wurden zunächst simultan optimiert, ohne daß eine Eigenschaft im Laufe der Optimierung verschlechtert wurde. Die Fehler bezüglich beider Größen wurden um etwa eine Größenordnung verkleinert. Allerdings waren sämtliche Ergebnisse ab $x^{(3)}$ Zufallsergebnisse, da nur 25000 Äquilibrierungs- und 100000 Produktionsschritte verwendet wurden, was zu sehr schlechten Statistiken geführt hat. Es handelte sich somit nur noch um zufällige Verbesserungen. Daher wurde für $x^{(7)}$ die Produktionszeit um den Faktor 10 erhöht, um sichere Ergebnisse zu erhalten. Das erhaltene Kraftfeld war in Bezug auf ρ nahezu optimal, da die Dichte zu allen Temperaturen bis auf $T = 313.2$ K weniger als 0.5% vom Experiment abwich. Die Steigung der Dichte in Abhängigkeit von der Temperatur konnte jedoch nicht gut reproduziert werden. Der durchschnittliche Fehler in Bezug auf die Diffusion lag noch über 10%, was lediglich ein zufriedenstellendes Ergebnis ist. Daß der Fehlerfunktionswert von 0.0032 für $x^{(7)}$ nur ein Zufallsergebnis war, ist auch anhand von $\nabla F(x^{(7)})$ zu erkennen, welcher für eine längere Produktionszeit in eine völlig andere Richtung zeigte. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{S})$ und $\sigma(\mathrm{C})$ sowie $\varepsilon(\mathrm{S})$ und $\varepsilon(\mathrm{C})$. Ab $x^{(7)}$ kam $q((\mathrm{S}))$ hinzu, das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{S})^{(k)}, \sigma(\mathrm{C})^{(k)}, \varepsilon(\mathrm{S})^{(k)}, \varepsilon(\mathrm{C})^{(k)}(, q(\mathrm{S})^{(k)})\right)$. In der Tabelle ist der Vollständigkeit halber zusätzlich $q(\mathrm{C})^{(k)}$ angegeben.

zumal die intermolekularen Wechselwirkungen sowieso bereits sehr gering waren. Nach einer Energieminimierung und kurzen NVT-Simulation von 5000 Zeitschritten mit kleinerer Zeitschrittweite (Prä-Prä-Äquilibrierung) wurden 300000 Zeitschritte verwendet, um die Moleküle aus ihren originalen Positionen herauszutreiben (Präääquilibrierung). Die Äquilibrierung bestand nur aus 25000 Zeitschritten, da sich diese nur auf Dichte und potentielle Energie bezog, die nach einem derart kurzen Zeitraum bereits äquilibriert waren. Abbildung 5.31(b) zeigt eine äquilibrierte Konfiguration: Das System ist nur etwas dichter geworden als zu Beginn. Abbildung 5.31(c) zeigt die Äquilibrierung der potentiellen Energie: Im Laufe der Präääquilibrierung sank die potentielle Energie ab, um dann um einen konstanten Mittelwert herum zu oszillieren, welcher auch in der Produktionsphase erhalten blieb. Aufgrund der langen Präääquilibrierungsphase wurde die Produktionszeit mit 100000 Zeitschritten zu Beginn der Optimierung klein gewählt, allerdings wurde diese im Laufe der Optimierung zu spät erhöht, so daß vor allem für die Diffusion keine ausreichend gute Statistiken erhalten werden konnten.

Optimierungsergebnisse für D und ρ Tabelle 5.13 und Abbildung 5.32 zeigen die Optimierungsergebnisse im Falle von D und ρ. Verwendet wurde dabei die Methode des steilsten Ab-



(a) (b)

Abbildung 5.32: Optimierung von D (a) und ρ (b) im Falle von CS_2. Das resultierende Kraftfeld ist lediglich als zufriedenstellend zu bewerten. Die Siededichte wurde zwar nahezu optimal reproduziert, allerdings lag der MAPE-Wert bezüglich der Diffusion immer noch über 10%.

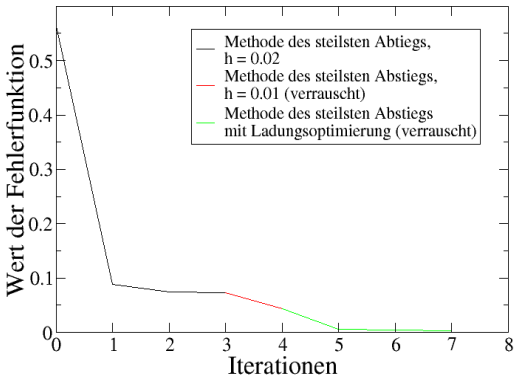
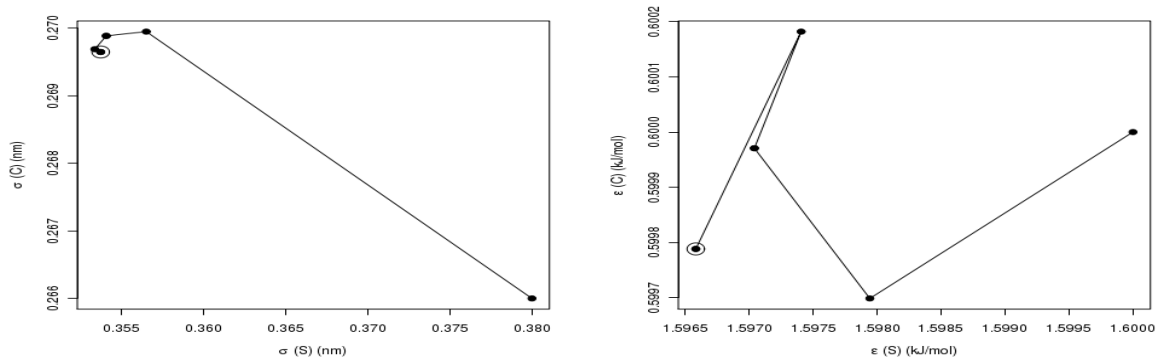


Abbildung 5.33: Entwicklung der Fehlerfunktion im Falle von CS_2. Diese wurde innerhalb von sieben Iterationen scheinbar um fast drei Größenordnungen verkleinert. Es ist dabei zu beachten, daß ab der dritten Iteration die Ergebnisse nur noch Zufall waren, so daß insgesamt nur drei Iterationen zu verzeichnen sind.

stiegs mit $h = 0.02$, $\zeta_A = 0.2$ und $\forall_{i,T}\ w_{i,T} = 1$. Der zulässige Bereich war $\Omega = (40, 80, 40)$, das heißt, σ wurde um maximal 40%, ε um maximal 80% und die Partialladungen um maximal 40% verändert. Die Motivation für die Wahl $h = 0.02$ lag wieder in der langsamen Konvergenz zu Beginn der Optimierung und darin, daß aufgrund der kurzen Simulationszeit zu Beginn der Optimierung die statistischen Unsicherheiten höher waren als im Normalfall.
Innerhalb von sieben Iterationen konnte die Fehlerfunktion scheinbar um fast drei Größenordnungen verkleinert werden, wobei beide Zielgrößen zunächst simultan verbessert wurden. Abbildung 5.33 zeigt die Entwicklung der Fehlerfunktion. Ab der dritten Iteration waren sämtliche Ergebnisse nur noch Zufallsergebnisse, so daß im Endeffekt nur von drei anstatt sieben Iterationen gesprochen werden kann. Die Produktionszeit war vor allem für die Diffusion viel zu kurz. Die Wahl $h = 0.01$, die ab $x^{(3)}$ erforderlich war, war somit ungeeignet. Die Produktionszeit hätte von da an bereits erhöht werden sollen. Auch die Hinzunahme der Partialladungen



(a) (b)

Abbildung 5.34: Entwicklung der LJ-Parameter im Falle von CS_2 (D,ρ): σ(S) und σ(C) (a) sowie ε(S) und ε(C) (b). Die Kreise zeigen die optimalen Parameter an. Weder σ noch ε haben sich ab der dritten Iteration signifikant geändert.

brachte keinen Erfolg, so daß das finale Kraftfeld in der siebten Iteration nur zufällig erhalten wurde. Um dieses Kraftfeld dennoch zu evaluieren und eventuell weiterzuoptimieren, wurde die Produktionszeit verzehnfacht. Dadurch ergaben sich statistische saubere Werte für Diffusion und Dichte. Die Diffusion wich im Durchschnitt um 12.29% und die Dichte um nur 0.28% vom Experiment ab. Die Abweichung der Dichte lag außer bei $T = 313.2$ K, wo sie -0.85% betrug, unterhalb der 0.5%-Marke. Das resultierende Kraftfeld war somit lediglich zufriedenstellend, da auch der Trend der experimentellen Kurve in Abhängigkeit von der Temperatur nicht allzu gut wiedergegeben werden konnte. Dies konnte auch mit der längeren Produktionszeit nicht verbessert werden. Die Resultate wurden mit zehn unabhängigen Replikaten abgesichert: Der durchschnittliche MAPE-Wert bezüglich der Diffusion lag bei 14.24% mit einer Standardabweichung von 0.69% und bezüglich der Dichte bei 0.44% mit einer Standardabweichung von 0.10%. Die Diffusionskoeffizienten sind somit auch bei zehnfacher Produktionszeit immer noch stark verrauscht.
Insgesamt waren mehr als 100 Simulationen zum Erhalt des Kraftfelds erforderlich. In den ersten drei Iterationen wurden jedoch nur 36 Simulationen durchgeführt. Von da an hätte die Produktionszeit bereits vergrößert werden müssen. Da sich allerdings $x^{(3)}$ von $x^{(7)}$ nicht wesentlich unterschied und das molekulare Modell wie oben motiviert eher ungeeignet war, wären auch auf diese Art und Weise keine signifikanten Verbesserungen mehr zu erwarten gewesen.

Abbildung 5.34 zeigt die Entwicklung der Kraftfeldparameter (Abbildung 5.34(a) bezieht sich auf σ und Abbildung 5.34(b) auf ε): Auch hier ist zu erkennen, daß das Ausmaß an Rauschen ab der dritten Iteration zu hoch war, da sich weder σ noch ε signifikant geändert haben.

Fazit Mit dem hier verwendeten molekularen Modell und *Gromacs* konnte kein in Bezug auf D und ρ optimales Kraftfeld erzielt werden. Für CS_2 ist ein zentraler Quadrupol unumgänglich, um ein möglichst realitätsnahes molekulares Modell zu erhalten, welches die intermolekularen Wechselwirkungen akkurat beschreibt (siehe zum Beispiel Vrabec u. a. (2001)). Der Quadrupolmoment sollte daher unbedingt in die Optimierung miteinbezogen werden.

5.2 Kombinationsalgorithmus mit CMA-ES und Selektion eines Verfahrens für die Optimierung in der Nähe des Minimums

Wie bereits in Abschnitt 3.6.6 motiviert, ist eine Vorschaltung von globalen Optimierungsverfahren in manchen Fällen unumgänglich, vor allem, wenn keine geeigneten initialen Kraftfeldparameter für ein spezielles Optimierungsproblem in der Literatur gefunden werden können. In anderen Fällen können zwar LJ-Parameter in der Literatur vorhanden sein, allerdings möchte man andere Partialladungen verwenden, die reproduzierbarer und über eine Vielzahl ähnlicher chemischer Umgebungen konsistent sind. Letzteres wird zum Beispiel am Fraunhofer-Institut SCAI mit $WOLF_2PACK$ realisiert. Mit $WOLF_2PACK$ ermittelte Partialladungen wurden bereits praktisch verwendet, und zwar für Phosgen, Methanol und Kohlenstoffdisulfid. Insbesondere bei Kohlenstoffdisulfid und Methanol konnte festgestellt werden, daß die Kombination dieser Partialladungen mit Standard-LJ-Parametern aus der Literatur problematisch sein kann, da das simulierte System oftmals mechanisch instabil wurde. In Abschnitt 5.1 wurde dem mit einer verbesserten Präääquilibrierung oder mit einer manuellen Nachjustierung der LJ-Parameter entgegengewirkt. Dies kann jedoch kein genereller Ansatz sein, um derartige Probleme zu lösen.

In dieser Arbeit wird empfohlen, einen globalen Optimierungsalgorithmus vorzuschalten. Die hier eingesetzten Verfahren sind der evolutionäre Algorithmus CMA-ES (vergleiche Abschnitt 3.3.2) und die am Fraunhofer-Institut SCAI entwickelte interpolationsbasierte Software DesParO (vergleiche Abschnitt 3.3.3). Der vorliegende Abschnitt behandelt lediglich CMA-ES und die Optimierung der LJ-Parameter. Es soll dabei zunächst gezeigt werden, daß eine Kombination von CMA-ES mit GROW, ohne einen zu hohen Rechenaufwand zu betreiben, bei Molekularen Simulationen sinnvoll ist. In Abschnitt 4.5.1 wurde anhand der Korrelationsfunktionen aus Abschnitt 3.5.2 praktisch evaluiert, daß bei einem geeigneten Abbruchkriterium für CMA-ES eine derartige Kombination grundsätzlich möglich ist. Abschnitt 5.2.1 kombiniert CMA-ES mit GROW am Beispiel von Phosgen.

Weiterhin ist abzuwägen, was in der Nähe des Minimums sinnvoll ist. Nicht in allen Fällen konnte GROW das Minimum bis auf statistische Unsicherheiten genau bestimmen, was jedoch bisher stets auf einen eher ungeeigneten Modellansatz zurückzuführen war. In Abschnitt 4.5.2 konnte mithilfe der Korrelationsfunktionen jedoch gezeigt werden, daß das modifizierte Verfahren nach Stoll (vergleiche Abschnitt 3.6.5) und das exakte Trust-Region-Verfahren (vergleiche Abschnitt 3.4.3) bei Verwendung der in Anhang H.1 angegebenen Temperaturfits näher an das Minimum gelangen können als GROW. Es wurde allerdings in Abschnitt 4.5.2 motiviert, daß eine Evaluation nur anhand der Korrelationsfunktionen nicht ausreicht und der Einsatz Molekularer Simulationen unumgänglich ist. Im Rahmen der *Industrial Fluid Property Simulation Challenge (IFPSC) 2010* (IFPSC, 2010) wurde eine Optimierung der Dichte eines Ethers, und zwar von *Dipropylen-Glykol-Dimethylether* ($C_8H_{18}O_3$), durchgeführt. Die Dichte konnte von GROW nicht bis auf ihre gewünschte Toleranz genau bestimmt werden. Abschnitt 5.2.2 zeigt, daß die in Abschnitt 4.5.2 eingesetzten Verfahren bei dieser Anwendung deutliche Verbesserungen liefern.

5.2.1 Anwendung von CMA-ES und Kombination mit GROW: Phosgen

Da sich die Kombination von CMA-ES mit GROW bereits für die Korrelationsfunktionen als erfolgreich herausgestellt hat, aber vor allem aufgrund der starken Abhängigkeit des Erfolgs von der Wahl des Abbruchkriteriums nicht gefolgert werden konnte, ob die Kombination tatsächlich auf Molekulare Simulationen anwendbar ist, wird dies in diesem Abschnitt diskutiert. Betrachtet wird die Optimierungsaufgabe aus Abschnitt 5.1.3, bei der die Siededichte von Phosgen zu sieben verschiedenen Temperaturen optimiert wurde. Bei GROW ist dabei eine Kombination der Methode des steilsten Abstiegs mit der exakten Trust-Region-Methode erforderlich (siehe Tabelle 5.6 und Abbildung 5.16).

Voroptimierung mit CMA-ES Die Startparameter für CMA-ES wurden wie folgt gewählt: $\sigma(\mathrm{C})^{(0)} := 0.5$ nm, $\sigma(\mathrm{O})^{(0)} := 0.5$ nm, $\sigma(\mathrm{Cl})^{(0)} := 0.5$ nm, $\varepsilon(\mathrm{C})^{(0)} := 0.5$ kJ/mol, $\varepsilon(\mathrm{O})^{(0)} := 1.0$ kJ/mol und $\varepsilon(\mathrm{Cl})^{(0)} := 1.0$ kJ/mol. Da erwartet werden kann, daß die LJ-Interaktionen bei Sauerstoff und Chlor größer sein werden als bei Kohlenstoff, da letzterer zentral im Molekül gelegen ist, wurden die Energieparameter dieser beiden Atome um den Faktor zwei größer gewählt als alle anderen Parameter. Diese chemische Vorkenntnis ist auch bei der Verwendung globaler Optimierungsverfahren vollkommen legitim, da dadurch die Optimierung beschleunigt werden kann.
Der zulässige Bereich bei der globalen CMA-ES-Optimierung wurde $\Omega = (80, 99)$ gesetzt. Als initiale Schrittweiten wurden die in Hansen (2011) empfohlenen Werte gewählt. Da in Abschnitt 4.4.1 keine generische Populationsgröße λ ermittelt werden konnte, jedoch gemäß Hansen u. Ostermeier (2001) stets $\lambda \geq N$ gelten sollte, wurde der Minimalwert für λ gewählt, also $\lambda = N = 6$, um zu testen, ob mit möglichst wenig Funktionsauswertungen pro Iteration eine Konvergenz mit akzeptablem Gesamtrechenaufwand erzielt werden kann.

Aufgrund der angesprochenen starken Abhängigkeit des Erfolgs des Kombinationsalgorithmus vom Abbruchkriterium $F(x) \leq \tau_{\mathrm{CMA-ES}}$ wurde darauf verzichtet. CMA-ES wurde dann abgebrochen, wenn die Standardabweichungen, also die CMA-ES-Schrittweiten, innerhalb von fünf Iterationen nahezu konstant blieben. Dies ist ein Zeichen dafür, daß der globale Optimierer anfängt zu stagnieren und lokale Verfeinerungen nur noch mit erheblichem Rechenaufwand möglich sind. Nach 26 Iterationen und 152 Funktionsauswertungen wurde CMA-ES abgebrochen. Im Verlauf der Optimierung ergab sich folgendes Problem: Die Kraftfeldparameter waren aufgrund der zufälligen Wahl gerade zu Beginn der Optimierung teilweise physikalisch gesehen unsinnig, so daß das System mechanisch instabil war. Dies wurde automatisch abgefangen und die Dichte künstlich auf einen beliebig hohen Wert, zum Beispiel 10^5 kg/m^3, gesetzt, so daß CMA-ES die resultierenden Fehlerfunktionswerte als Ausreißer erkennen konnte. Da CMA-ES äußerst robust in Bezug auf Ausreißer ist, war dies für die Gesamtoptimierung unproblematisch.

Feinoptimierung mit GROW Als Startwert für die nachfolgende Feinoptimierung mit GROW wurden diejenigen Kraftfeldparameter gewählt, die den kleinsten Fehlerfunktionswert besaßen. Es handelte sich dabei um ein Individuum der 18. Population (vergleiche Abschnitt 3.3.2).

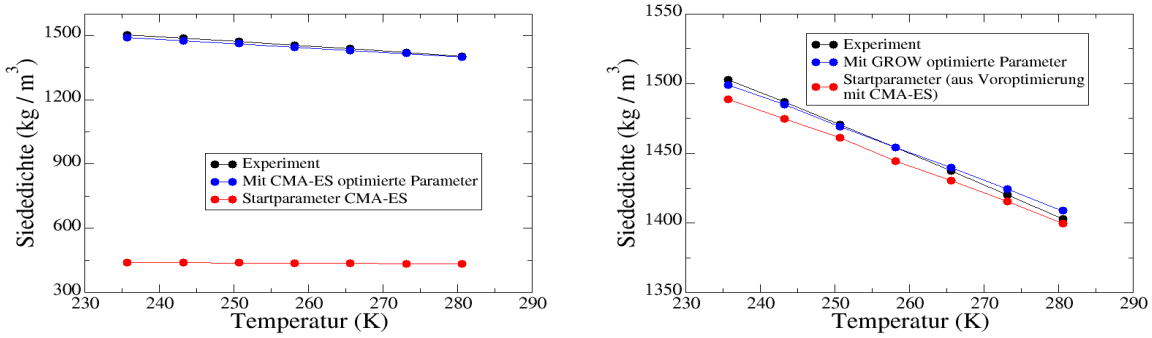


Abbildung 5.35: Voroptimierung mit CMA-ES von ρ_l (a) und anschließende Optimierung mit GROW (b) im Falle von Phosgen. GROW konnte die bereits sehr guten Ergebnisse, die aus der Voroptimierung resultierten, signifikant verbessern, so daß die Dichte optimal reproduziert werden konnte.

Da allerdings 26 Iterationen erforderlich waren, um sicherzustellen, daß die Voroptimierung abgebrochen werden kann, wird dieser hier mit $x^{(26)}$ bezeichnet. Es gilt:

$$x^{(26)} := (0.36827, 0.22741, 0.36288, 0.65066, 0.99893, 1.00618)^T.$$

Es ist deutlich zu sehen, daß $\varepsilon(\mathrm{O})$ und $\varepsilon(\mathrm{Cl})$ kaum verändert wurden. Der zugehörige Fehlerfunktionswert war $F(x^{(26)}) = 2.8 \cdot 10^{-4}$. Innerhalb von nur zwei Iterationen wurden mithilfe der Methode des steilsten Abstiegs optimale Ergebnisse erzielt. Die Einstellungen für diese Optimierung waren die gleichen wie in Abschnitt 5.1.3. Es gilt:

$$x^{(28)} = (0.36812, 0.22735, 0.36155, 0.65071, 0.99895, 1.00623)^T.$$

Dabei wurde $\sigma(\mathrm{Cl})$ in der dritten, $\varepsilon(\mathrm{O})$ in der fünften und alle anderen Parameter in der vierten Nachkommastelle verändert. Hierzu waren 41 (effektiv 32) Simulationen vonnöten. Die Gesamtanzahl an Simulationen beziehungsweise Funktionsauswertungen betrug somit 193 (effektiv 184), war also kleiner als 200. Für einen kompletten Kombinationsalgorithmus, bei dem die Startparameter weit vom Optimum entfernt lagen, ist dieser Rechenaufwand aus heutiger Sicht durchaus akzeptabel. Abbildung 5.35 zeigt die initialen und finalen Werte der Siededichte im Vergleich zum Experiment für die Voroptimierung mit CMA-ES (Abbildung 5.35(a)) sowie die Optimierung mit GROW (Abbildung 5.35(b)): Die Startwerte für CMA-ES lieferten eine Siededichte, die um etwa den Faktor 3 kleiner war als aus dem Experiment. Am Ende der CMA-ES-Voroptimierung konnten bereits sehr gute Ergebnisse für die Siededichte erzielt werden, die allerdings noch nicht optimal waren. Es ist zu beachten, daß es sich dabei um das Ergebnis eines einzelnen Individuums handelte. Die Ergebnisse der anderen Individuen in den letzten Iterationen von CMA-ES waren im Durchschnitt deutlich schlechter. Auch hier zeigt sich also erneut, daß CMA-ES als Vertreter eines globalen Optimierers alleine nicht eingesetzt werden sollte. Erst die Kombination mit GROW lieferte nach zwei Iterationen mit der Methode des steilsten Abstiegs eine optimale Siededichte.

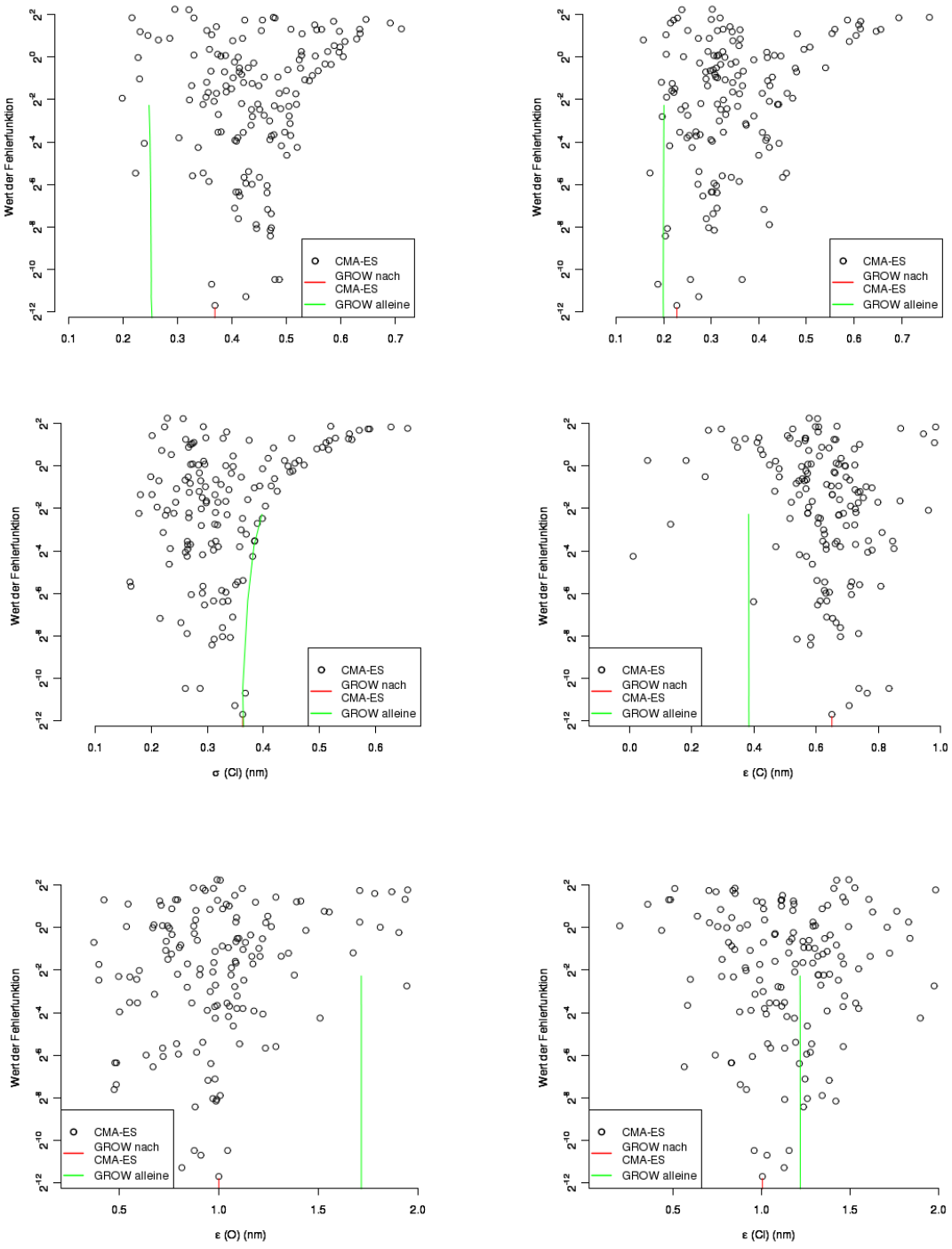
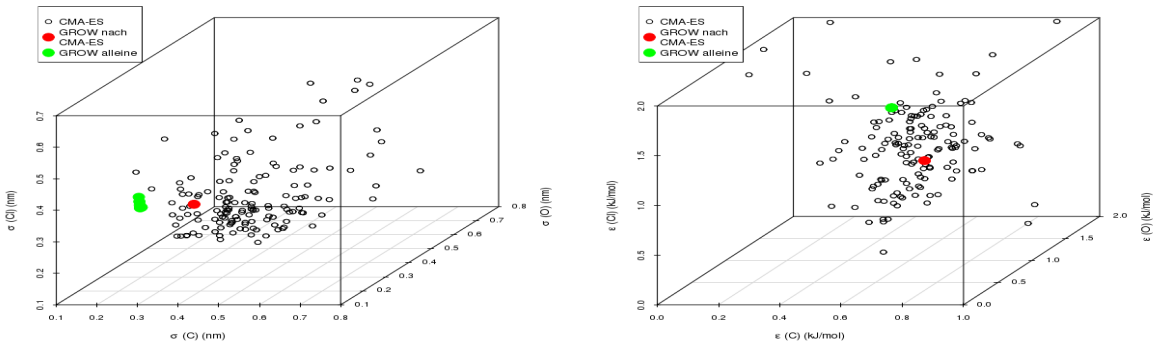


Abbildung 5.36: Fehlerfunktionswerte in Abhängigkeit von allen sechs zu optimierenden LJ-Parametern. Die schwarzen Kreise geben die innerhalb der Voroptimierung mit CMA-ES erhaltenen Funktionswerte wieder. Ausgehend vom kleinsten Funktionswert wurde eine Optimierung mit GROW gestartet, was durch die blauen Linien gekennzeichnet ist. Die grünen Linien zeigen die in Abschnitt 5.1.3 durchgeführte Optimierung mit GROW alleine. Außer für σ(Cl) wurden verschiedene Kraftfeldparameter aus den beiden verschiedenen Optimerungsläufen erhalten. Aufgrund der sechs Graphiken läßt sich ein parabolischer Verlauf der Fehlerfunktion, der einer Regenrinne gleicht, stark vermuten. Am Boden dieser Rinne gibt es eine Vielzahl an globalen Optima, vielleicht sogar unendlich viele. Je nach Startparameter werden durch verschiedene Optimierungsläufe somit auch verschiedene globale Minima gefunden.



(a) (b)

Abbildung 5.37: Entwicklung der LJ-Parameter im Falle von Phosgen (ρ_l) bei der Kombination von CMA-ES mit GROW: σ(C), σ(O) und σ(Cl) (a) sowie ε(H) und ε(C), ε(O) und ε(Cl) (b). Die schwarzen unausgefüllten Punkte geben die innerhalb der Voroptimierung mit CMA-ES erhaltenen Parameter wieder. Ausgehend von dem kleinsten Funktionswert wurde eine Optimierung mit GROW gestartet, deren Parameterentwicklung durch die blauen Punkten gekennzeichnet ist. Die grünen Punkte zeigen die Parameterentwicklungen innerhalb der in Abschnitt 5.1.3 durchgeführten Optimierung mit GROW alleine.

Diskussion: Kombination von CMA-ES mit GROW Im folgenden soll der Verlauf des Kombinationsworkflows noch etwas detaillierter dargestellt werden: Abbildung 5.36 zeigt die Fehlerfunktionswerte in Abhängigkeit von allen zu optimierenden LJ-Parametern. Anhand der Ergebnisse der Voroptimierung durch CMA-ES läßt sich analog zu den Beobachtungen in Abschnitt 3.5.3 (vergleiche auch Abbildung 3.11) auch hier ein parabolischer Verlauf der Fehlerfunktion vermuten, ähnlich einer Regenrinne, an deren Boden eine Vielzahl an lokalen und auch globalen Optima liegen kann, möglicherweise unendlich viele. Je nach Startwert kann ein anderes Minimum gefunden werden. Zwei verschiedene globale Minima wurden durch die hier durchgeführten Optimierungsläufe bereits detektiert: Die Graphiken zeigen den Vergleich zwischen der in Abschnitt 5.1.3 durchgeführten lokalen Optimierung mit GROW alleine (grüne Linien) und der hier durchgeführten GROW-Optimierung in Kombination mit CMA-ES (blaue Linien). Außer für σ(Cl) unterschieden sich sämtliche Kraftfeldparameter der beiden globalen Optima, woraus sich deuten läßt, daß σ(Cl) den größten Einfluß auf die Siededichte ausübt und zu einem großen Anteil die Optimalität des Kraftfeldes bestimmt. Dies läßt sich auch chemisch interpretieren: Da ε im allgemeinen nur sehr geringe Auswirkungen auf die Dichte hat, spielt σ bei dieser Optimierung von vornherein eine dominierende Rolle. Da Chlor zweimal im Molekül Phosgen auftaucht, ist dieses Atom an den meisten Wechselwirkungen beteiligt, was zu dem hohen Einfluß führt.
Abbildung 5.37 zeigt die Entwicklung der Kraftfeldparameter im Laufe der Optimierung (Abbildung 5.37(a) bezieht sich auf σ und Abbildung 5.37(b) auf ε): Die Kraftfeldparameter bei der Voroptimierung durch CMA-ES waren zunächst weit verstreut, fokussierten sich jedoch im Laufe der Optimierung auf die Mitte der zulässigen Gebiets. Die Änderungen bei der anschließenden GROW-Optimierung sind im Vergleich zu denen bei der Voroptimierung äußerst gering, da GROW lediglich lokale Verfeinerungen vornimmt. Weiterhin ist auch hier wieder festzustellen, daß im Falle von GROW die Änderungen von σ stärker ausgeprägt sind als die von ε.

Fazit In diesem Abschnitt konnte gezeigt werden, daß eine Kombination von CMA-ES mit GROW auf Molekulare Simulationen anwendbar ist, auch wenn CMA-ES mit Kraftfeldparametern startet, die sich sehr weit weg von einem globalen Minimum befinden. Es wurden andere Kraftfeldparameter als in Abschnitt 5.1.3 gefunden. Die Optimierung mit GROW war bei der Kombination zwar deutlich einfacher und weniger rechenintensiv als zuvor, da kein anschließendes Trust-Region-Verfahren verwendet werden mußte, allerdings kann dies nicht verallgemeinert werden. Es kann aufgrund der Untersuchungen in Abschnitt 4.5.1 nach wie vor nicht ausgeschlossen werden, daß die Wahl eines ungeeigneten Abbruchkriteriums für CMA-ES dazu führen kann, daß die anschließende Optimierung durch GROW nicht zum gewünschten Erfolg führt. Ebensowenig kann sichergestellt werden, daß stets das globale Minimum gefunden wird. Fällt die Fehlerfunktion irgendwo am Boden der Rinne nochmals ab, so kann das dadurch entstehende Minimum vermutlich höchstens durch einen enorm hohen Rechenaufwand detektiert werden (sogenanntes *Nadel-im-Heuhaufen-Problem*). Um dies dennoch mit akzeptablem Rechenaufwand zu ermöglichen, ist CMA-ES durch effizientere Voroptimierer zu ersetzen. Da derartige Studien allerdings den Rahmen dieser Arbeit sprengen würden, wird lediglich im Ausblick kurz darauf eingegangen.
Selbst wenn mehrere globale Minima vorliegen und die hier angewandte Kombination eines davon findet, so kann nicht automatisch davon ausgegangen werden, daß dieses Minimum das chemisch gesehen am besten geeignete Kraftfeld liefert. In Abschnitt 5.1.3 wurden zum Beispiel beim Erhalt optimaler Siededichten durch GROW alleine Verdampfungsenthalpien erzielt, die durchschnittlich etwa 3% von den experimentellen Daten abwichen (vergleiche Abbildung 5.17). Das bei der vorliegenden Kombination vom CMA-ES und GROW erhaltene Kraftfeld lieferte ebenfalls optimale Siededichten, allerdings lag der MAPE-Wert bezüglich der Verdampfungsenthalpie bei etwa 16%.
Zusammenfassend läßt sich sagen, daß eine derartige Kombination zwischen globaler und lokaler Optimierung in jedem Fall empfehlenswert und auch durchführbar ist, vor allem wenn keine geeigneten Startparameter vorliegen. Um jedoch robust und mit möglichst wenig Rechenaufwand das globale Minimum zu erreichen, sind zusätzliche Überlegungen zu treffen und CMA-ES gegebenenfalls zu ersetzen, da dieses Verfahren nicht dazu in der Lage ist, mit akzeptablem Rechenaufwand sicher und robust in einen für GROW geeigneten Einzugsbereich des globalen Minimums zu gelangen.

5.2.2 Bewertung der in der Nähe des Minimums eingesetzten gradientenbasierten Verfahren: Dipropylenglykoldimethylether

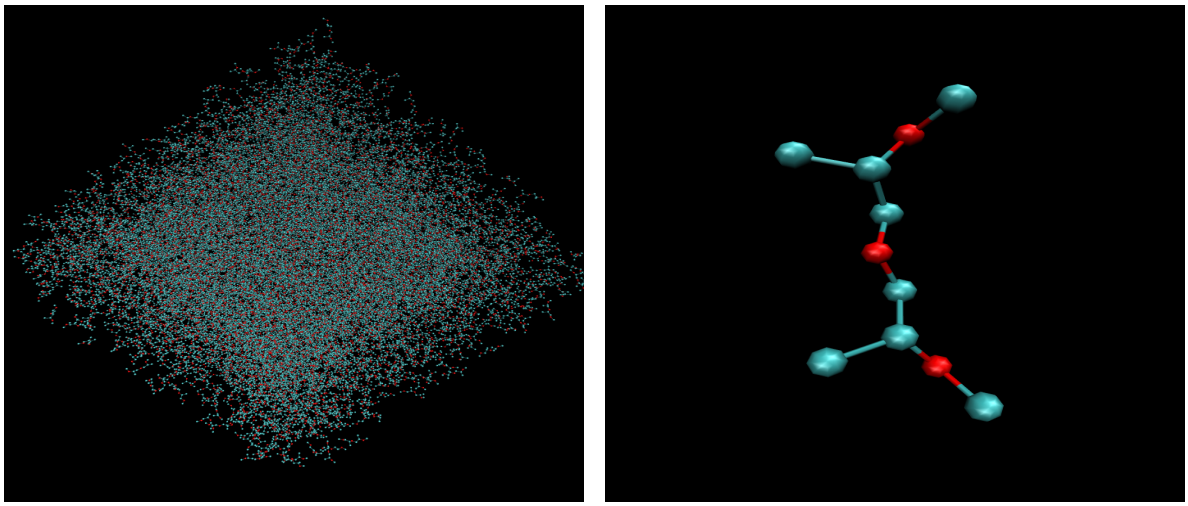
Im folgenden wird ein Optimierungsproblem betrachtet, das GROW alleine nicht zufriedenstellend lösen konnte. Wie bereits in Abschnitt 4.5.2 dargelegt, können in einem derartigen Fall in der Nähe des Minimums jedoch andere Verfahren eingesetzt werden. Es handelt sich dabei um die Variante des Verfahrens nach Stoll und das exakte Trust-Region-Verfahren, basierend auf Temperaturfits. In diesem Abschnitt wird nun eine endgültige Wahl zwischen diesen beiden Verfahren getroffen. Zunächst wird jedoch das Optimierungsproblem selbst vorgestellt.

Im Rahmen der *Industrial Fluid Property Simulation Challenge (IFPSC) 2010* (IFPSC, 2010) sollte ein Gleichgewichtszustand zwischen zwei flüssigen Phasen mithilfe von molekularen Modellen ermittelt werden. Eine der beiden Komponenten war Wasser und die andere Dipropylenglykoldimethylether, ein inertes, wasserabweisendes, ungiftiges und industriell weit eingesetztes Lösungsmittel. Es hat die Summenformel $C_8H_{18}O_3$, und es liegen die folgenden drei Isomere vor:

- Isomer I: CH_3-O-$CH(CH_3)$-CH_2-O-CH_2-$CH(CH_3)$-O-CH_3,
- Isomer II: CH_3-O-$CH(CH_3)$-CH_2-O-$CH(CH_3)$-CH_2-O-CH_3,
- Isomer III: CH_3-O-CH_2-$CH(CH_3)$-O-$CH(CH_3)$-CH_2-O-CH_3,

wobei typischerweise in Dipropylenglykoldimethylether Isomer I zu 50%, Isomer II zu 47% und Isomer III zu 3% vorhanden sind (DOW, 1995). Sämtliche Details zu den durchgeführten MD-Simulationen zum Erhalt des Phasengleichgewichts sind in Köddermann u. a. (2011) veröffentlicht.
Zunächst war ein geeignetes Kraftfeld für Dipropylenglykoldimethylether zu finden. Da lediglich experimentelle Flüssigkeitsdichten dieser Substanz bekannt waren, wurde GROW verwendet, um ein geeignetes Kraftfeld zur Reproduktion der Dichte aufzustellen. Die hier beschriebene GROW-Optimierung wurde bereits in Köddermann u. a. (2011) publiziert. Bei den durchzuführenden Simulationen handelte es sich wieder um MD-Simulationen im NPT-Ensemble, die mit *Gromacs* durchgeführt wurden. Weiterhin wurde analog zu Abschnitt 5.1 simuliert, allerdings wurde aufgrund der Größe des Moleküls stets $r_N = r_C = 1.2$ nm gewählt. Auf einem Rechencluster mit 215 zur Verfügung stehenden Knoten, auf dem jeder Knoten mit zwei Intel-Nehalem-EP-Quadcore-Prozessoren (Xeon X5550) ausgestattet ist mit jeweils 24 GB Arbeitsspeicher, die über einen schnellen QDR-Infiniband-Interconnect mit jeweils 40 Gb/s Double Date Rate (DDR) miteinander verbunden sind, benötigte eine Simulation, parallelisiert auf acht Prozessoren, etwa drei bis vier Stunden.

Simulations- und Optimierungseinstellungen Bei Dipropylenglykoldimethylether handelt es sich um ein relativ großes Molekül, bestehend aus biegsamen Ketten. Sämtliche Kohlenwasserstoffgruppen wurden zusammengefaßt, so daß es sich um ein *United-Atom-Modell* handelte. Es ist zu beachten, daß die inneren CH_3-Gruppen keine Partialladungen hatten, da die elektrostatischen Wechselwirkungen aufgrund ihrer Position zu vernachlässigen waren. Die Bindungen wurden mithilfe des LINCS-Algorithmus (siehe Anhang B) starr gehalten, Winkel und Diederwinkel waren flexibel. Intramolekulare Kraftfeldparameter wurden mit GÅMESS (Guest u. a., 2005) erhalten, abgesehen von den Rotationskonstanten für das Diederwinkelpotential, welche mithilfe von $WOLF_2PACK$ ermittelt wurden. 1,4-Wechselwirkungen wurden nicht exkludiert, sondern explizit als intermolekulare Wechselwirkungen betrachtet, da die Kette so biegsam ist, daß sich Atome desselben Moleküls so nahe kommen können, daß nichtbindende Wechselwirkungen zwischen ihnen entstehen. Es wurde ein NPT-Ensemble, bestehend aus allen drei Isomeren, mit der oben genannten Zusammensetzung simuliert. Die zu optimierende Zielgröße war lediglich die Flüssigdichte bei Atmosphärendruck zu vier verschiedenen Temperaturen: 273, 298, 303 und 353 K. Die initialen LJ-Parameter stammen aus einer Voroptimierung für den kleineren Ether 1,2-Dimethoxyethan. Für weitere Details siehe Köddermann u. a. (2011).
Tabelle 5.14 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Opti-



(a) (b)

Abbildung 5.38: Startkonfiguration bestehend aus allen drei Isomeren (a) und einzelnes Molekül von Isomer I (b) im Falle von Dipropylenglykoldimethylether. Die Kantenlänge der Box beträgt 140 nm.

mierung im Falle von Dipropylenglykoldimethylether. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen und Toleranzen für die simulierten Zielgrößen angegeben.
Abbildung 5.38(a) zeigt die Startkonfiguration der Simulationsbox und Abbildung 5.38(b) die Struktur eines einzelnen Moleküls. In einer 140 nm großen Box wurden 512 Dipropylenglykoldimethylether-Moleküle per Zufall so positioniert, daß keine Überlappungen auftraten. Eine Gitterstruktur wäre unphysikalisch, und sie wurde nicht gewählt, da ansonsten unnötiger Simulationsaufwand zu betreiben gewesen wäre, um die Gitterstruktur aufzulösen. Nach einer Energieminimierung und kurzen NVT-Simulation von 20000 Zeitschritten mit kleinerer Zeitschrittweite (Prä-Prä-Äquilibrierung) wurden 100000 Zeitschritte verwendet, um die Moleküle aus ihren originalen Positionen herauszutreiben (Präääquilibrierung). Die Äquilibrierung bestand nur aus 25000 Zeitschritten, da sich diese nur auf Dichte und potentielle Energie bezog, die nach einem derart kurzen Zeitraum bereits äquilibriert waren. Die Produktionsphase betrug 250000 Zeitschritte, um ausreichend gute Statistiken zu erhalten.

Optimierungsergebnisse für ρ Tabelle 5.15 und Abbildung 5.39 zeigen die Optimierungsergebnisse im Falle von Dipropylenglykoldimethylether. Verwendet wurde dabei zunächst die Methode des steilsten Abstiegs mit $h = 0.01$, $\zeta_A = 0.2$ und $\forall_{i,T}\ w_{i,T} = 1$ (Abbildung 5.39(a)). Sämtliche Dichten wurden selbstverständlich als gleichwertig betrachtet. Der zulässige Bereich war $\Omega = (30, 40)$. Alle Parameter wurden nach der ersten Iteration in der dritten Nachkommastelle geändert, außer $\sigma(\mathrm{CH})$, welches in der zweiten Nachkommastelle, und $\varepsilon(\mathrm{O})$, welches nur in der vierten Nachkommstelle geändert wurde. Weder andere Optimierungseinstellungen (kleineres h, kleineres zulässiges Gebiet) noch die Verwendung eines CG-Verfahrens konnten kleinere Fehlerfunktionswerte liefern, das heißt, mithilfe von GROW konnten keine Verbesse-

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	CH_3, CH_2, CH, O	*United-Atom*-Modell innere CH_3-Gruppen ohne Ladungen
	Dipole/Quadrupole	–	
	Molare Masse	162.22 g/mol	$m(CH_3)$ = 15.034 g/mol, $m(CH_2)$ = 14.026 g/mol $m(CH)$ = 13.018 g/mol $m(O)$ = 15.999 g/mol
	Molekülstruktur	biegsame Kette	3 Isomere
	Intramolekulares Kraftfeld	10 starre Bindungen (LINCS) 11 Winkel 30 Diederwinkel	Quelle: GÅMESS (Guest u. a., 2005) Quelle: $WOLF_2PACK$
	Ladungen	11 Punktladungen	Quelle: TraPPE-Kraftfeld (Stubbs u. a., 2004)
	Besonderheiten	1,4-Wechselwirkungen nicht exkludiert	Grund: biegsame Kette
Simulationsbox	Startkonfiguration Anzahl Moleküle Kantenlänge	randomisiert 512 140 nm	siehe Abbildung 5.38(a)
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung Präääquilibrierung Äquilibrierung Produktion	20000 100000 25000 250000	Schrittweite: 0.2 fs Schrittweite: 2 fs Schrittweite: 2 fs Schrittweite: 2 fs
Optimierungsrelevantes	zu optimierende Parameter	σ und ε für alle Atomtypen (8D)	Quelle Startwerte: Voroptimierung von 1,2-Dimethoxyethan (Köddermann u. a., 2011)
	Temperaturen	273, 298, 303, 353	in K
	Druck	1.0	in bar
Ausgaben:			
Zielgrößen	experimentell simuliert	ρ ρ: Gleichung (C.8)	Quelle: Esteve u. a. (2003)
	Toleranzen	ρ: 0.5%	

Tabelle 5.14: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Dipropylenglykoldimethylether.

rungen mehr erzielt werden. Aufgrund der Voroptimierung in Köddermann u. a. (2011) befanden sich die Startparameter bereits sehr nahe am Minimum. Die Richtung des Gradienten bei $x^{(1)}$ war völlig anders als die bei $x^{(0)}$. Abbildung 5.39(a) zeigt, daß bei $x^{(1)}$ im Gegensatz zu $x^{(0)}$ die Dichten größer waren als die experimentellen, was vermuten läßt, daß der Algorithmus über das Minimum hinausgesprungen ist, daß also $h = 0.01$ im Falle von $x^{(1)}$ zu groß und somit der berechnete negative Gradient keine Abstiegsrichtung mehr war. Dies war daran zu erkennen, daß die Folge der Fehlerfunktionswerte innerhalb der Armijo-Schrittweitensteuerung nicht monoton fallend war, sondern alternierend. Dies war auch noch bei $h = 0.001$ der Fall. Allerdings kann $h = 0.001$ auch schon zu klein gewesen sein, so daß lediglich statistisches Rauschen reproduziert wurde und der Algorithmus höchstens gegen ein durch Oszillationen hervorgerufenes Minimum hätte konvergieren können.
Wie in Abschnitt 3.5.5 motiviert, ist es nicht sinnvoll zu versuchen, ein optimales h zu fin-

k	$x^{(k)}$	MAPE ρ	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
	Steilster Abstieg, $h = 0.01$, $\Omega = (30, 40)$				
0	0.43300 0.39340 0.37064 0.28050	1.74%	0.0013	0.00374 0.00800 0.04550 -0.00779	0.05
	0.08315 0.38282 0.81480 0.45737			-0.00560 -0.00195 -0.00259 -0.00195	
1	0.43238 0.39207 0.36309 0.28179	1.33%	$8.1{\cdot}10^{-4}$	-0.00347 -0.00668 -0.03522 0.00480	0.04
	0.08408 0.38314 0.81523 0.45769			0.00410 0.00155 0.00184 0.00165	
	Stoll-Variante, $\Delta_0 = 0.1\Delta_{\max}$, $\Omega = (5, 10)$				
0	0.43300 0.39340 0.37064 0.28050	1.77%	0.0013	0.00358 0.00791 0.04593 -0.00834	0.05
	0.08315 0.38282 0.81480 0.45737			-0.00610 -0.00244 -0.00286 -0.00219	
1	0.43217 0.39257 0.36981 0.28133	1.23%	$7.0{\cdot}10^{-4}$	nicht ermittelt	
	0.08398 0.38365 0.81563 0.45820				
2	0.43084 0.39124 0.36848 0.28266	0.54%	$1.9{\cdot}10^{-4}$	nicht ermittelt	
	0.08531 0.38299 0.81497 0.45754				
3	0.43024 0.39243 0.36728 0.28386	0.40%	$8.4{\cdot}10^{-5}$	nicht ermittelt	
	0.08651 0.38179 0.81556 0.45634				

Tabelle 5.15: Optimierungsergebnisse für Dipropylenglykoldimethylether im Falle von ρ mit der Methode des steilsten Abstiegs und der Variante des Verfahrens nach Stoll: Die Startparameter befanden sich aufgrund der Voroptimierung in Köddermann u. a. (2011) bereits in der Nähe des Minimums. Die Methode des steilsten Abstiegs konnte die Fehlerfunktion innerhalb einer Iteration fast halbieren. Danach konnten mithilfe von GROW, das heißt durch andere Einstellungen oder Verwendung eines CG-Verfahrens, keine Verbesserungen mehr erzielt werden. Die Variante des Verfahrens nach Stoll hingegen konnte die Fehlerfunktion innerhalb von drei Iterationen um etwa anderhalb Größenordnungen verkleinern. Der MAPE-Wert bezüglich ρ wurde um mehr als den Faktor 4 verkleinert. Das erhaltene Kraftfeld ist in Bezug ρ optimal, da alle Zielgrößen zu allen Temperaturen in ihren spezifischen Toleranzbereichen liegen. Die Steigung der Zielgrößen in Abhängigkeit von der Temperatur konnte fast reproduziert werden. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{CH_3})$, $\sigma(\mathrm{CH_2})$, $\sigma(\mathrm{CH})$ und $\sigma(\mathrm{O})$ sowie $\varepsilon(\mathrm{CH_3})$, $\varepsilon(\mathrm{CH_2})$, $\varepsilon(\mathrm{CH})$ und $\varepsilon(\mathrm{O})$, das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{CH_3})^{(k)}, \sigma(\mathrm{CH_2})^{(k)}, , \sigma(\mathrm{CH})^{(k)}, \sigma(\mathrm{O})^{(k)}, \varepsilon(\mathrm{CH_3})^{(k)}, \varepsilon(\mathrm{CH_2})^{(k)}, \varepsilon(\mathrm{CH})^{(k)}, \varepsilon(\mathrm{O})^{(k)}\right)$. Es ist zu beachten, daß der Gradient bei der Variante des Verfahrens nach Stoll ab $x^{(1)}$ nicht mehr ermittelt wurde, da die Gradienten jeder der acht Dichten mithilfe der in Abschnitt 3.6.2 beschriebenen Methodik effizient berechnet wurden, die der Fehlerfunktion selbst jedoch nicht. Die Ergebnisse für die Startwerte unterschieden sich bei den beiden Optimierungsverfahren ein wenig, da die entsprechenden Simulationen wiederholt wurden, um ein hohes Ausmaß an Rauschen ausschließen zu können.

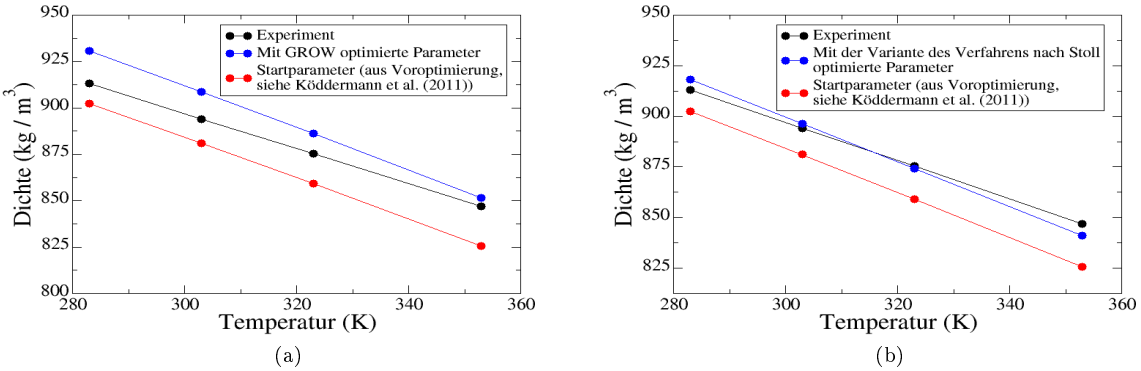


Abbildung 5.39: Optimierung von ρ im Falle von Dipropylenglykoldimethylether mit der Methode des steilsten Abstiegs (a) und der Variante des Verfahrens nach Stoll (b). Nur mit letzterer konnte die Dichte optimal reproduziert werden.

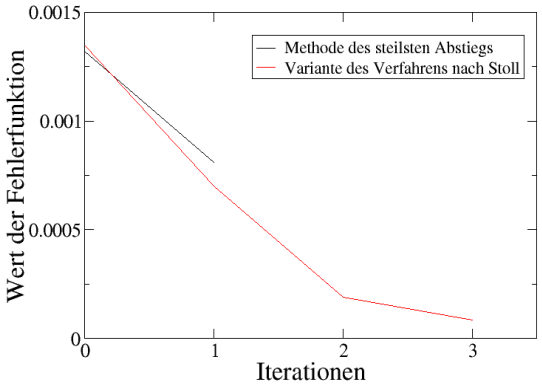


Abbildung 5.40: Entwicklung der Fehlerfunktion im Falle von Dipropylenglykoldimethylether bei der Methode des steilsten Abstiegs (schwarze Linie) und der Variante des Verfahrens nach Stoll (rote Linie). Die Ergebnisse für die Startwerte unterschieden sich bei den beiden Optimierungsverfahren ein wenig, da die entsprechenden Simulationen wiederholt wurden, um ein hohes Ausmaß an Rauschen ausschließen zu können. Die Fehlerfunktion konnte mithilfe der Methode des steilsten Abstiegs innerhalb von einer Iteration halbiert und mit der Variante des Verfahrens nach Stoll innerhalb von drei Iterationen um etwa anderhalb Größenordnungen verkleinert werden.

den. Das zulässige Gebiet hätte bereits bei $x^{(0)}$ verkleinert werden sollen, allerdings bietet sich in diesem Fall eher ein Trust-Region-Ansatz an, der von Anfang an ein kleines Vertrauensgebiet festsetzt. Daher wurde die Variante des Verfahrens nach Stoll aus Abschnitt 3.6.5 mit $\Delta_0 := 0.1\Delta_{\max}$ und $\Omega = (5, 10)$ verwendet. Dabei ist $\Delta_{\max}$ das größtmögliche Δ, so daß $B_\Delta(x^{(0)}) \subset \Omega$. Wie in Abschnitt 4.5.2 motiviert, macht eine quadratische Modellierung der Fehlerfunktion in der Nähe des Minimums mehr Sinn als die Berechnung eines Gradienten. Die Gradienten, die bei der Variante des Verfahrens nach Stoll berechnet werden, beziehen sich hier auf die Dichten, nicht auf die Fehlerfunktion selbst. Gemäß Abschnitt 3.5.5 ist das Ausmaß des Rauschens bei der Fehlerfunktion größer als das, mit welchem die Zielgrößen behaftet sind. Daher ist ein Gradient bezüglich der Zielgrößen zum einen leichter zu berechnen, und zum anderen werden alle nachfolgenden Gradienten mithilfe der in Abschnitt 3.6.2 dargestellten Methodik effizient berechnet, so daß nur auf vorherige Simulationen zurückgegriffen wird und somit die Wahrscheinlichkeit deutlich geringer ist, daß Punkte auf der anderen Seite des Minimums ausgewertet werden. Tatsächlich konnte die Variante des Verfahrens nach Stoll in diesem Fall im Gegensatz zu GROW optimale Kraftfeldparameter liefern: Innerhalb von nur drei Iterationen konnte die Fehlerfunktion um etwa anderthalb Größenordnungen verkleinert werden, wobei der Fehler in Bezug auf ρ um mehr als den Faktor 4 verbessert wurde. Abbildung 5.40 zeigt die Entwicklung der Fehlerfunktion im Falle der Methode des steilsten Abstiegs und der Variante des Verfahrens nach Stoll. Es ist zu beachten, daß sich die Funktionswerte sowie die Gradienten bei $x^{(0)}$ etwas unterschieden, da die Simulation wiederholt wurde, um ein hohes Ausmaß an statistischem Rauschen auszuschließen. Aufgrund der effizienten Gradientenberechnung dürfen die Dichten der Startwerte nicht mit großen statistischen Unsicherheiten behaftet sein.
Für eine klassische Gradientenberechnung waren stets acht Simulationen erforderlich. Die maximale Anzahl an Armijo-Schritten wurde wie wieder auf 10 gesetzt. Insgesamt wurden drei Iterationen durchgeführt, wobei stets nur ein Trust-Region Schritt notwendig war. Inklusive der Startwerte und der klassischen Gradientenberechnung für $k = 0$ waren insgesamt nur

$1 + 8 + 3 \cdot 1 = 12$ Simulationen notwendig. Abbildung 5.39 zeigt, daß sämtliche nach der Optimierung mit den experimentellen Daten bis auf Unsicherheiten übereinstimmten. Auch der Trend der experimentellen Kurve wurde ziemlich gut wiedergegeben.
Es ist zu beachten, daß es sich hierbei um ein achtdimensionales Optimierungsproblem handelt, wobei nur vier Zielgrößen angepaßt wurden. Das Problem war somit unterbestimmt, und das Verfahren nach Stoll aus Abschnitt 3.2.3 wäre aufgrund der singulären Matrizen der zu lösenden Gleichungssysteme nicht anwendbar gewesen. Die in dieser Arbeit entwickelte Variante aus Abschnitt 3.6.5 ist, wie dort motiviert, dagegen auch auf unterbestimmte Probleme anwendbar.

Selektion eines geeigneten Algorithmus in der Nähe des Minimums Bis hierher konnte also dargelegt werden, ob und inwieweit andere gradientenbasierte Verfahren eingesetzt werden können, um optimale Kraftfeldparameter zu liefern, wenn die Methoden aus Abschnitt 3.4 keine Verbesserungen mehr liefern. Die Variante des Verfahrens nach Stoll ist tatsächlich dazu in der Lage. Ein anderer Kandidat für diese Problemstellung ist das exakte Trust-Region-Verfahren in Kombination mit Temperaturfits und einer effizienten Hesse-Berechnung (siehe auch Abschnitt 4.5.2). Auch dieses Verfahren lieferte noch Verbesserungen, da es aufgrund der vorher eingestellten Größe des Vertrauensgebiets nicht über das Minimum hinaussprang, also nicht wie die Methode des steilsten Abstiegs zu einem Punkt gelangte, wo $h = 0.01$ zu groß und $h = 0.001$ zu klein war. Es wurde wieder $h_G = 0.01$ und $h_H = 0.02$ sowie $\Delta_0 = 0.6\Delta_{\max}$ und $\Omega = (5, 10)$ verwendet. Der Gradient wurde stets korrekt berechnet, und die Genauigkeit der Hesse-Matrix ist aufgrund von Bemerkung 3.5.5 nicht so ausschlaggebend. Allerdings konvergierte das Verfahren eben wegen eines derartig kleinen Vertrauensgebiets äußerst langsam. Nach sechs Iterationen und 315 Simulationen waren immer noch keine optimalen Kraftfeldparameter vorhanden, und die Hesse-Matrix wurde wie erwartet nie effizient berechnet. Somit wurde das Verfahren abgebrochen.

Fazit Mit äußerst geringem Rechenaufwand konnte die Variante des Verfahrens nach Stoll im Gegensatz zu allen anderen hier betrachteten gradientenbasierten Verfahren optimale Kraftfeldparameter liefern. Somit ist der untere Teil des Filters aus Abbildung 4.4 ergründet. Aufgrund der hier vorgenommenen vergeblichen Versuche, mit GROW noch Verbesserungen zu erzielen, ist es für die Zukunft ratsam, dieses Verfahren bereits früher zu verwenden, das heißt vor der Verkleinerung von h und des zulässigen Gebiets sowie der Vergrößerung der Äquilibrierungs- und Produktionszeiten. Vermutlich hätte dann auch beispielsweise im Falle von Methanol (siehe Abschnitt 5.1.4) Rechenzeit eingespart werden können. Daß die Methode auch bei anderen Applikationen sinnvoll ist, wird allerdings nicht nochmals anhand von Methanol dargelegt, da es dabei nur um die Anwendbarkeit gradientenbasierter Verfahren mit unterschiedlicher Anzahl an Zielgrößen und Kraftfeldparametern ging. Vielmehr wird dies anhand von Ethylenoxid in Abschnitt 5.3.3 gezeigt.

5.3 Kombinationsalgorithmus mit DesParO: Ethylenoxid

Im folgenden wird anstelle von CMA-ES die am Fraunhofer-Institut SCAI entwickelte Software *DesParO* (siehe Abschnitt 3.3.3) als globaler Voroptimierer getestet. Im Gegensatz zu CMA-

ES sind sämtliche molekularen Modelle *a priori* festzulegen, das heißt nicht nur ein einziges Startmodell. Die zu optmierenden Kraftfeldparameter sind in gewissen zulässigen Bereichen per Zufallsprinzip zu modifizieren. Wann eine ausreichende Anzahl an Zufallsmodellen vorliegt, wird hier durch eine manuelle Anpassung an die betrachteten Zielgrößen entschieden. Sind sämtliche Zielgrößen durch das von DesParO erstellte Interpolationsmodell zumindest annähernd auf ihre jeweiligen experimentellen Referenzdaten einstellbar, so wird das entsprechende Modell als Startmodell für die anschließende GROW-Optimierung verwendet. Falls nicht, sind weitere Zufallsmodelle hinzuzufügen und weniger gut geeignete Zufallsmodelle zu entfernen. Damit sind zum Beispiel solche gemeint, die einen um mehrere Größenordnungen höheren Fehlerfunktionswert liefern als andere und somit als statistische Ausreißer anzusehen sind.

Als Anwendungsbeispiel für den hier betrachteten Kombinationsalgorithmus wurde das hochgiftige Ethylenoxid (kurz: Epoxid) verwendet, da bereits in Maaß u. a. (2010) ein zufriedenstellendes Kraftfeld für diese Substanz mithilfe von DesParO erhalten werden konnte. Die anzupassenden Zielgrößen waren VLE-Daten, sprich Siededichte und Verdampfungsenthalpie zu verschiedenen Temperaturen. Allerdings wurden die Simulationen mithilfe der MC-Software *Towhee, Version 6.2.2,* (Towhee, 2011) durchgeführt. Dabei wurde die GEMC-Methode verwendet (siehe Anhang A.5), bei der Flüssigkeit und Gas simultan in separaten Boxen simuliert wurden, wobei ein Austausch von Teilchen und Energie zwischen den beiden Boxen möglich ist. Bei einer lokalen Verfeinerung durch GROW sind die Zielgrößen, wie bereits mehrfach motiviert, zu äquilibrieren, das heißt, es muß solange simuliert werden, bis das Äquilibrierungskriterium (3.70) für Siededichte, nichtbindende und potentielle Energie erfüllt ist. Es hat sich allerdings herausgestellt, daß eine derartige Äquilibrierung bei Verwendung der *Towhee*-Implementation der GEMC-Methode bis zu zwei Wochen dauern kann. Daher war *Towhee* für eine anschließende GROW-Optimierung nicht geeignet.
Stattdessen wurde das Simulationspaket *ms2* verwendet (siehe Anhang F.3) und der Dampfdruck als Zielgröße hinzugenommen, um ein Kraftfeld zu generieren, welches alle relevanten VLE-Daten reproduziert. Zum Erhalt von VLE-Daten benutzt *ms2* die sogenannte *Grand-Equilibrium-Methode* (siehe Anhang A.5), bei der Flüssigkeit und Gas separat simuliert werden. Zunächst wurde jeweils eine NPT-Simulation für die Flüssigkeit durchgeführt, aus der das chemische Potentiel ermittelt wurde. Betrachtet wurden sieben verschiedene Temperaturen. Es wurden stets MC-Simulationen zusammen mit der Methode der graduellen Einsetzung betrachtet, da die betrachteten Temperaturen für MD-Simulationen zusammen mit der Widom-Methode zu niedrig waren (vergleiche Abschnitt 2.5.2). Anschließend wurde jeweils eine pseudo-μVT-Simulation (vergleiche Anhang A.5) für die Gasphase durchgeführt, aus der Siededichte, Verdampfungsenthalpie und Dampfdruck ermittelt wurden. Die MC-Simulationen basierten auf dem Metropolis-Schema aus Abschnitt 2.4.2 mit einer Akzeptanzwahrscheinlichkeit für eine Versuchsbewegung von $P_{ij} = 0.5$. Bei den MD-Simulationen wurde die Prädiktor-Korrektor-Methode nach Gear benutzt (siehe Abschnitt 2.3.1). Bei *ms2* ist keine Äquilibrierung im Sinne dieser Arbeit möglich, da nicht mit einer bereits vorhandenen Trajektorie explizit neu gestartet werden kann. Die Anzahl an Zeitschritten für die entsprechenden Teilsimulationen sind im einzelnen anzugeben: Anstelle der Energieminimierung werden einige wenige MC-Schritte durchgeführt, um die Teilchen aus ihren originalen Positionen herauszubewegen. Die Prä-Prä-Äquilibrierung entspricht hier einer einfachen NVT-Simulation. Prääquilibrierung und Produk-

tion werden wie gewohnt durchgeführt, eine Äquilibrierung im Sinne von Abschnitt 3.5.4 gibt es bei *ms2* nicht. Allerdings gibt *ms2* für jeden thermodynamischen Durchschnitt statistische Fehler aus, welche die Abweichungen von den tatsächlichen Mittelwert schätzen. Diese sind im Laufe einer Optimierung stets zu beobachten.
Aufgrund der Tatsache, daß *ms2* nur für starre Moleküle verwendbar ist, wurden sämtliche Bindungen und Winkel automatisch starr gehalten. Da die Molekülstruktur von Epoxid ein zentrales geschlossenes Dreieck enthält (siehe Abbildung 5.41), ist ein starres Modell hierfür gut geeignet.
Für Epoxid werden im folgenden zwei verschiedene, gegenüberzustellende Kraftfeldmodelle betrachtet: Eines beinhaltet Dipole, deren Wechselwirkungen mithilfe des in Gleichung (2.31) angegebenen Dipolpotentials modelliert werden. Das andere enthält Punktladungen, deren Wechselwirkungen mithilfe des elektrostatischen Potentials aus Gleichung (2.24) beschrieben werden. Letzteres wird nicht mit der Ewald-Summation bestimmt, sondern durch Einführung eines Abschneideradius (siehe Abschnitt 2.6.2) für die Massenzentren der Teilchen und Verwendung der Reaktionsfeldmethode mit $\epsilon_{\mathrm{RF}} = 10^{10}$ (siehe Gleichung (2.35) und Anhang C.2). Weitere Details hierzu sind in Deublein u. a. (2011) nachzulesen. *ms2* beruht auf reduzierten Parametern und Zielgrößen (vergleiche Abschnitt 3.6.4). Die Größe des Abschneideradius betrug stets $r_C = 3.5\sigma_R$. Zum Erhalt konstanter Temperaturen wurde im Falle von MD-Simulationen die Methode der Zwangsbedingungen mit einfacher Reskalierung der Geschwindigkeiten (siehe Anhang A.1) und im Falle von MC-Simulationen die in Anhang A.3 beschriebene Methode verwendet. Zum Erhalt konstanter Drücke wurde für MD-Simulationen das Andersen-Barostat (Methode des erweiterten Systems mit einer Kolbenmasse von $m_V = 10^{-4}$ g, siehe Anhang A.2) und für MC-Simulationen ebenfalls die in Anhang A.3 beschriebene Methode verwendet.

Auf dem in Abschnitt 5.2.2 erwähnten Rechencluster benötigte eine VLE-Simulation, parallelisiert auf acht Prozessoren, etwa acht bis zehn Stunden. Da es sich um zwei aufeinanderfolgende Simulationen handelte, war die Gesamtzeit doppelt so hoch wie im Falle der bisherigen Simulationen.

Insgesamt werden im folgenden zwei verschiedene Kombinationsworkflows mit DesParO und GROW betrachtet:

1. Zunächst werden als Startparameter für GROW die in Maaß u. a. (2010) erhaltenen Modellparameter verwendet. Es ist allerdings zu beachten, daß in dieser Publikation die mit *Towhee* berechneten Verdampfungsenthalpien nicht ganz korrekt sind. Der Grund dafür lag in der Tatsache, daß zur Berechnung des intermolekularen Potentials ein relativ großer Abschneideradius zu wählen war. Eine Änderung des Abschneideradius von seinem Standardwert hatte in *Towhee* allerdings keine Auswirkungen, was auf einen Fehler in der *Towhee*-Implementation zurückzuführen war. Daher stimmten die hier mit *ms2* korrekt berechneten Verdampfungsenthalpien für die Startparameter mit denen aus Maaß u. a. (2010) nicht ganz überein. Für die Feinoptimierung werden sowohl ein Punktladungsmodell gemäß Maaß u. a. (2010) als auch ein Dipolmodell verwendet, welches auf Eckl u. a. (2008a) zurückgeht. Ziel dieser Studie ist es, mit einem der beiden Modelle ein möglichst

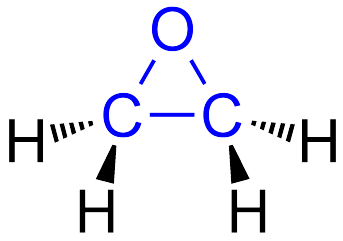


Abbildung 5.41: Strukturformel von Ethylenoxid (Wikipedia, 2011f). Schwere Atome sind in blau und leichte in schwarz dargestellt. Zur Modellierung wurden die CH_2-Gruppen zusammengefaßt.

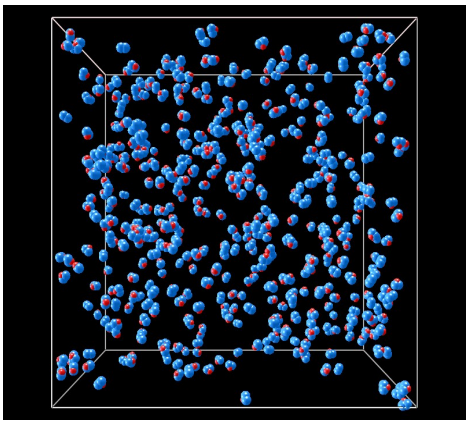
optimales Kraftfeld für Epoxid zu erhalten.

2. Um einen konsistenten Kombinationsworkflow zu gewährleisten, wird hier auch die Voroptimierung durch DesParO mithilfe von *ms2* durchgeführt. Dadurch kann auch die Verdampfungsenthalpie stets korrekt berechnet werden. Dabei wird lediglich das Punktladungsmodell aus Maaß u. a. (2010) verwendet. Ziel dieser Studie ist es, die Kombination von DesParO mit GROW bewerten und mit der Kombination von CMA-ES mit GROW vergleichen zu können.

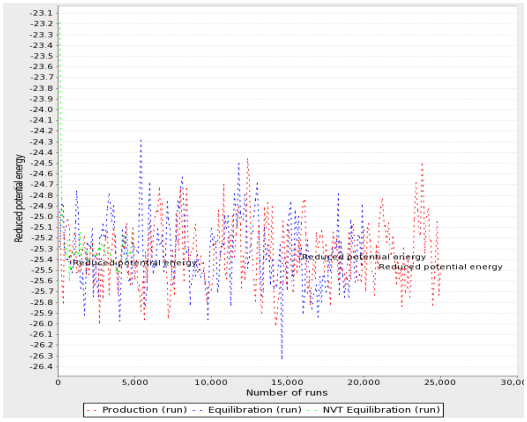
Nachdem in Abschnitt 5.3.1 kurz allgemeine Eigenschaften von Ethylenoxid vorgestellt werden, berichtet Abschnitt 5.3.2 über die in dieser Arbeit durchgeführte Voroptimierung mithilfe von DesParO. Sämtliche Feinoptimierungen mit GROW werden in Abschnitt 5.3.3 beschrieben.

5.3.1 Allgemeines zu Ethylenoxid

Epoxide, auch Oxacyclopropane genannt, sind sehr reaktionsfähige organische Verbindungen, bestehend aus einem dreiatomigen Ring, bei dem ein Kohlenstoffatom durch ein Sauerstoffatom ersetzt ist. Das einfachste und industriell relevanteste Epoxid ist Ethylenoxid (C_2H_4O), dessen Strukturformel in Abbildung 5.41 abgebildet ist. Ein Wasserstoffatom ist jeweils an der C–C–O-Ebene gespiegelt. Wird im folgenden von *Epoxid* gesprochen, so ist stets Ethylenoxid gemeint.
Ethylenoxid ist bei Raumtemperatur gasförmig, farblos, hochentzündlich und bildet mit Luft explosionsfähige Gemische. Der Schmelzpunkt liegt bei -112.55 °C und der Siedepunkt bei 10.45 °C. Es weist einen süßlichen Geruch auf, ist leicht wasserlöslich und hochgradig toxisch, was experimentelle Untersuchungen im Labor äußerst gefährlich und kostenspielig macht, da eine Vielzahl an Sicherheitsbestimmungen einzuhalten ist. Es ist beim Einatmen krebserregend und verursacht Kopfschmerzen, Schwindel und kann in hohen Dosen zum Koma führen. Außerdem kann sich die Lunge nach Einatmen von Ethylenoxid mit Flüssigkeit füllen.
Ursprünglich wurde Ethylenoxid durch Behandlung von 2-Chlorethanol mit einer Base erzeugt, wobei Salzsäure freiwurde. Heutzutage wird es industriell durch Oxidierung von Ethen bei 200–300°C an einem Silberkatalysator hergestellt. Aufgrund seiner Toxizität wird es oftmals zur Sterilisation eingesetzt. Weiterhin findet Ethylenoxid in der Automobilindustrie Verwendung, und zwar bei der Herstellung des Kühl- und Frostschutzmittels Ethylenglykol.
Da Experimente im Labor mit Ethylenoxid äußerst gefährlich sind, wird es sowohl wissenschaftlich als auch industriell für Molekulare Simulationen relevant. Außerdem eignet es sich sehr gut



(a) Startkonfiguration von Ethylenoxid, Gasbox. Die Graphik wurde mit *ms2molecules* erstellt (siehe Anhang F.3).



(b) Verlauf der reduzierten potentiellen Energie der Flüssigkeit im Falle von Ethylenoxid. Die Graphik wurde mit *ms2chart* erstellt (siehe Anhang F.3).

Abbildung 5.42: Startkonfiguration des Gases (a) und Verlauf der reduzierten potentiellen Energie der Flüssigkeit (b) im Falle von Ethylenoxid mit der Geometrie aus Abbildung 5.41. Die Boxgröße wurde so festgelegt, daß eine vorgegebene initiale Dichte erhalten wurde.

als Testsubstanz für den vorliegenden Kombinationsalgorithmus, da es sowohl experimentell (Buckles u. a., 1999; Olson u. Wilson, 2008) als auch durch Simulationen (Müller u. a., 2008; Eckl u. a., 2008a; Maaß u. a., 2010) bereits gut charakterisiert wurde.

5.3.2 Anwendung von DesParO zum Erhalt geeigneter Startwerte

Bei dem verwendeten Modell für Epoxid handelte es sich um ein starres *United-Atom-Modell*, das heißt, die dreiatomigen CH_2-Gruppen wurden zusammengefaßt. Die intramolekularen Kraftfeldparameter wurden aus Maaß u. a. (2010) genommen. Die Partialladungen wurden wieder mithilfe von $WOLF_2PACK$ ermittelt. Anstelle des Dipolmodells aus Eckl u. a. (2008a) wurde hier somit ein eigenes Punktladungsmodell verwendet. Der einzige Unterschied zu Maaß u. a. (2010) bestand darin, daß eine etwas andere Geometrie verwendet wurde: An einem Kohlenstoffatom wurde ein rechter Winkel angenommen. Dieses Modell besitzt ein Dipolmoment von 2.689 D. Bei dem Dipolmodell aus Eckl u. a. (2008a), welches sich bereits bewährt hat, lag das Dipolmoment bei 2.459 D. Nimmt man die Geometrie aus Abbildung 5.41, so ist das Dipolmoment mit 2.040 D deutlich weiter davon entfernt. Inwieweit mit diesem Modell die experimentellen Zielgrößen zu reproduzieren sind, wird bei der Feinoptimierung durch GROW zu analysieren sein.

Simulations- und Optimierungseinstellungen Tabelle 5.16 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Epoxid. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen für die simulierten Zielgrößen (Siededichte, Verdampfungsenthalpie und Dampfdruck) angegeben. Es wurden VLE-Simulationen zu den drei Temperaturen 230, 330 und 430 K durchgeführt. Details zu VLE-Simulationen sind in Anhang A.5 nachzulesen.
Abbildung 5.42(a) zeigt die Startkonfiguration des Gases, sprich eine kubische Simulationsbox

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	CH_2, O	*United-Atom*-Modell
	Dipole/Quadrupole	–	Dipolmodell möglich
	Molare Masse	44 g/mol	$m(CH_2)$ = 14 g/mol, $m(O)$ = 16 g/mol
	Molekülstruktur	planares Dreieck	polare CO-Bindung
	Intramolekulares Kraftfeld	3 starre Bindungen 3 starre Winkel	Quelle: Maaß u. a. (2010)
	Ladungen	3 Punktladungen	Quelle: Maaß u. a. (2010)
	Dipolmodell	1 Dipol im Massenzentrum	Quelle: Eckl u. a. (2008a)
Simulationsbox	Startkonfiguration	kubisch	siehe Abbildung 5.42(a)
	Anzahl Moleküle	500	
	Boxgröße	durch initiale Dichten festgesetzt	
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung	5000/10000	Schrittweite (MD): 2 fs
	Präääquilibrierung	20000/25000	Schrittweite (MD): 2 fs
	Äquilibrierung	–	–
	Produktion	200000/100000	Schrittweite (MD): 2 fs
Optimierungsrelevantes	zu optimierende Parameter	$\varepsilon(CH_2)$, $\varepsilon(O)$, $\sigma(CH_2)$, $\sigma(O)$	Quelle Startwerte: Zufallsmodelle
	Temperaturen	230, 330, 430	in K
Ausgaben:			
Zielgrößen	experimentell	ρ_l, $\Delta_v H$, p_σ	Quelle: Buckles u. a. (1999) Olson u. Wilson (2008)
	simuliert	ρ_l: Gleichung (C.8) $\Delta_v H$: Gleichung (C.4) p_σ: Gleichung (C.7) mit langreichweitigem Korrekturterm	Details: siehe Deublein u. a. (2011) und Anhang A.5

Tabelle 5.16: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von Epoxid.

bestehend aus 500 Teilchen mit der Molekülgeometrie aus Abbildung 5.41. Es ist zu beachten, daß *ms2* die Boxgröße so festlegt, daß eine vorgegebene initiale Dichte erhalten wird. Abbildung 5.42(b) zeigt den Verlauf der potentiellen Energie während der Simulation der flüssigen Phase. Nach einigen wenigen MC-Schritten, welche die Teilchen schnell aus ihrer unphysikalischen Startkonfiguration entfernten und einer kurzen NVT-Simulation (5000 Zeitschritte für die Flüssigkeit und 10000 Zeitschritte für das Gas) wurden 20000 Zeitschritte für die Flüssigkeit beziehungsweise 25000 für das Gas verwendet, um die Moleküle miteinander interagieren zu lassen (Präääquilibrierung). Die Produktionsphase betrug 200000 Zeitschritte im Falle der Flüssigkeit und 100000 im Falle des Gases.

Optimierung mit DesParO Die Optimierung mithilfe von DesParO hat im Vergleich zu den bisherigen Optimierungen den Unterschied, daß nicht die Fehlerfunktion aus Gleichung (4.1) minimiert wurde, sondern mittels multikriterieller Optimierung sämtliche Zielgrößen direkt an ihre experimentellen Referenzwerte angepaßt wurden. Im vorliegenden Fall lagen aufgrund der drei Temperaturen und der drei physikalischen Zielgrößen insgesamt neun Kriterien vor. Ein Kriterium hat dabei die Form

$$\min_{x\in\Omega} |f_i^{\mathrm{sim}}(x) - f_i^{\mathrm{exp}}|,\ i \in \{1, ..., n\}, \tag{5.3}$$

wobei Ω wieder das zulässige Gebiet ist und folgermaßen festgesetzt wurde:
$\varepsilon(CH_2)/K \in [84.1900, 144.3257]$, $\varepsilon(O)/K \in [72.1628, 132.2986]$, $\sigma(CH_2)/Å \in [3.4, 4.0]$ und $\sigma(O)/Å \in [2.7, 3.8]$. In Maaß u. a. (2010) wurde festgestellt, daß $\sigma(O)$ scheinbar keinen Einfluß auf die betrachteten Systemeigenschaften haben. Weitere Studien zeigten (Maaß, 2011), daß dies jedoch daran lag, daß dort das zulässige Gebiet für $\sigma(O)$ zu klein gewählt wurde. Bei größeren Werten für $\sigma(O)$ hingegen konnten Einflüsse insbesondere auf die Siededichte festgestellt werden. Daher wurden in der vorliegenden Analyse auch größere Werte in Betracht gezogen.
Es ist weiterhin zu beachten, daß die zu minimierenden Funktionen in Gleichung (5.3) nicht differenzierbar sind, was auch im Falle von DesParO, das heißt beim Erstellen eines Metamodells, zu Schwierigkeiten führen kann. Allerdings kann von DesParO nicht erwartet werden, daß mit akzeptablem Rechenaufwand optimale Kraftfeldparameter erhalten werden können, welche sämtliche Zielgrößen bis auf statistische Unsicherheiten genau wiedergeben. Dazu wird für eine Feinoptimierung GROW verwendet. Hierbei wurden sieben verschiedene Temperaturen, und zwar 230, 260, 300, 330, 375, 400 und 430 K, betrachtet. Um Rechenaufwand einzusparen, wurden bei der Voroptimierung durch DesParO die niedrigste, eine mittlere und die höchste Temperatur verwendet. Die Motivation hierfür bestand darin, daß bei einer groben Voroptimierung nicht alle Temperaturen in Betracht gezogen werden müssen. Zielgrößen zu den anderen Temperaturen sind gegebenenfalls auch mit Temperaturfits (siehe Anhang H.1) zu erhalten. Letzteres würde jedoch zu ungenau, wenn nur zwei Temperaturen in Betracht gezogen würden. Wird zwischen nur zwei Punkten approximiert, so besteht die Gefahr, daß die geglättete Funktion eine Gerade ist, was physikalisch unsinnig wäre. Daher wurde eine dritte Temperatur hinzugenommen. In Abschnitt 5.1.4 wurde bereits anhand von Methanol gezeigt, daß auch mit wenigen Temperaturen gute Kraftfeldparameter erhaltbar sind, sofern nicht extrapoliert werden muß. Somit wurden die niedrigste, die höchste und eine mittlere Temperatur gewählt.

Angestrebt wurde zunächst die Auswertung von 50 Zufallsmodellen. Waren die einzelnen Kraftfeldparameter derart ungünstig gewählt, daß das System mechanisch instabil war, wurde das entsprechende Modell verworfen. Ist bei der manuellen Justierung innerhalb von DesParO (vergleiche Abbildung 3.7) festgestellt worden, daß eine oder mehrere Zielgrößen nicht ausreichend gut wiedergegeben werden konnten oder ein oder mehrere Kraftfeldparameter am Rand ihres zulässigen Intervalls lagen, wurden zusätzliche Zufallsmodelle hinzugefügt. Dies war beispielsweise bei $\varepsilon(O)$ der Fall, dessen zulässiges Intervall nach unten hin vergrößert werden mußte: $\varepsilon(O)/K \in [66.1493, 132.2986]$. Bei der manuellen Einstellung der Zielgrößen wurde folgendermaßen vorgegangen: Zunächst wurden $\sigma(CH_2)$ und $\sigma(O)$ so justiert, daß die Siededichten annähernd mit ihren Referenzdaten übereinstimmten. Anschließend wurden $\varepsilon(CH_2)$ und $\varepsilon(O)$ entsprechend auf Verdampfungsenthalpie und Dampfdruck eingestellt. Abbildung 5.43 zeigt die Ergebnisse einer von insgesamt drei manuellen Justierungen sowie die von DesParO erzeugte Korrelationsmatrix: Sämtliche Zielgrößen stimmten annähernd gut mit ihren experimentellen Referenzdaten überein, so daß eine Feinoptimierung durch GROW ermöglicht werden konnte. Insgesamt waren dazu 57 Zufallsmodelle erforderlich. Die Korrelationsmatrix gibt die Relationen zwischen Kraftfeldparametern und Zielgrößen wieder: Es konnte verifiziert werden, daß σ vor allem Einfluß auf die Siededichten hat. Im Gegensatz dazu wirkte sich ε vor allem auf Verdampfungsenthalpien und Dampfdrücke aus. Bei σ und Siededichten handelte es sich wie erwartet um negative und bei ε und Verdampfungsenthalpien um positive Korrelationen. Alter-

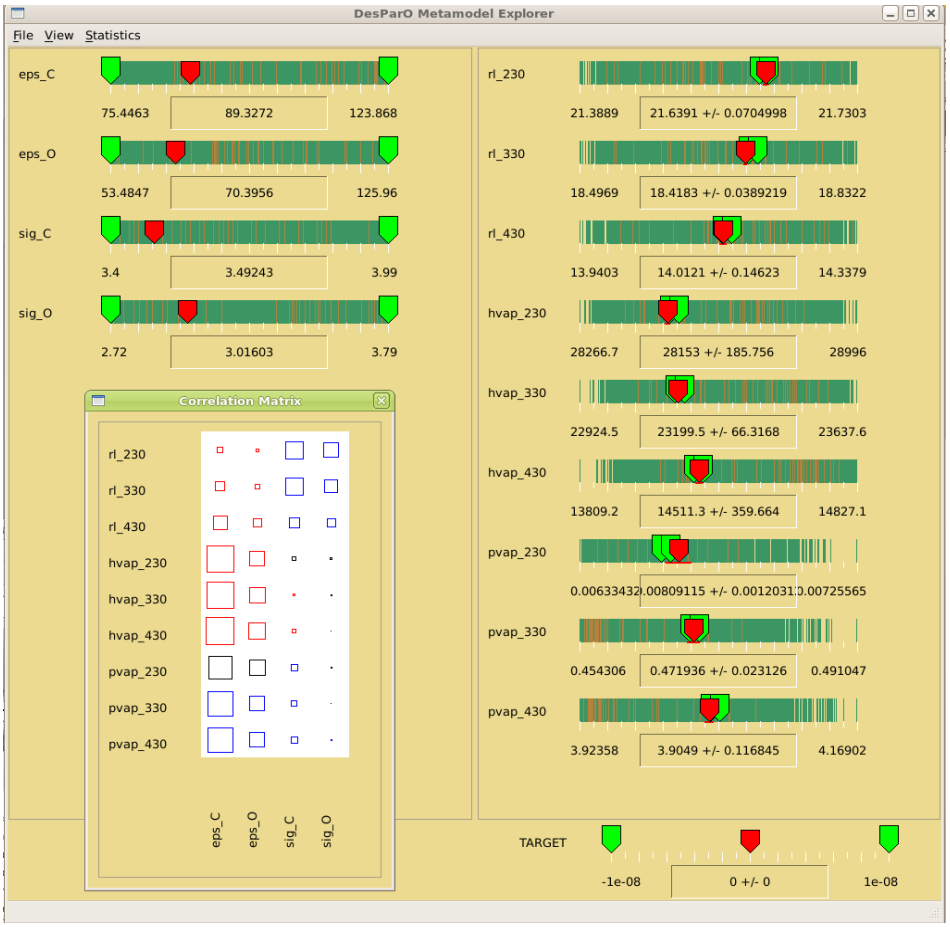


Abbildung 5.43: Manuelle Justierung der Kraftfeldparameter (linke Spalte) von DesParO und Korrelationsmatrix: Sämtliche Zielgrößen (rechte Spalte) stimmen annähernd gut mit ihren experimentellen Referenzwerten überein, so daß eine anschließende Feinoptimierung durch GROW ermöglicht wird. Die Kraftfeldparameter liegen weiterhin nicht am Rand ihrer zulässigen Intervalle. Die Korrelationsmatrix zeigt, daß σ vor allem Auswirkungen auf die Siededichten hat und ε auf Verdampfungsenthalpien sowie Dampfdrücke. Die Siededichte sinkt bei steigendem σ und die Verdampfungsenthalpie bei steigendem ε. Beim Dampfdruck sind bei $T = 230$ K alternierende Korrelationen festzustellen, allerdings ist auf der rechten Seite zu erkennen, daß dieser Dampfdruck ein großes von der Toleranzvorhersage von DesParO berechnetes Konfidenzintervall aufwies. Die Bezeichnungen in dieser Abbildung sind dieselben wie in Abbildung 3.7.

nierende Korrelationen konnten beim Dampfdruck im Falle von $T = 230$ K beobachtet werden, wobei das von der Toleranzvorhersage von DesParO berechnete Konfidenzintervall hierfür äußerst groß war. Es ist zu beachten, daß bei niedrigen Temperaturen und geringen Drücken der Erhalt korrekter physikalischer Größen generell schwierig ist. Dies gilt sowohl für experimentelle Messungen als auch für Simulationen.

In Tabelle 5.17 sind die aus der manuellen DesParO-Analyse resultierenden Startparametervektoren für die in Abschnitt 5.3.3 durchzuführende Feinoptimierung mit GROW angegeben. Zusätzlich wurde mittels der innerhalb von DesParO implementierten multikriteriellen Optimierung eine Pareto-Front extrahiert. Die daraus selektierten Kraftfeldparameter sind ebenfalls in Tabelle 5.17 dargestellt. Man sieht, daß auch mit DesParO unterschiedliche, ähnlich gute

Modell	$\varepsilon(CH_2)$	$\varepsilon(O)$	$\sigma(CH_2)$	$\sigma(CH_2)$
1	89.3272	70.3956	3.4924	3.0160
2	91.1026	65.8055	3.5927	2.7200
3	82.8710	94.3125	3.4275	3.1801
Pareto	82.7096	91.4134	3.4138	3.7579

Tabelle 5.17: Aus der DesParO-Analyse resultierende Kraftfeldparameter (ε in K und σ in Å) im Falle der drei manuellen Justierungen und des Pareto-Modells. Die Parametervektoren unterschieden sich sehr stark voneinander, das heißt, die Ergebnisse einer Voroptimierung mit DesParO können über den Parameterraum hinweg weit verstreut liegen.

Zielgröße	Fehler Modell 1	Fehler Modell 2	Fehler Modell 3	Fehler Pareto-Modell
$\rho_l,\ T = 230$ K	-0.002%	0.03%	0.15%	-0.32%
$\rho_l,\ T = 330$ K	-0.11%	0.04%	-0.35%	-0.46%
$\rho_l,\ T = 430$ K	-0.52%	-0.18%	-1.02%	-1.62%
$\Delta_v H,\ T = 230$ K	0.35%	-1.32%	0.67%	-1.03%
$\Delta_v H,\ T = 330$ K	-0.06%	-3.69%	-0.28%	-0.86%
$\Delta_v H,\ T = 430$ K	-1.93%	-1.39%	-2.79%	-3.12%
$p_\sigma,\ T = 230$ K	9.62%	1.03%	18.75%	3.90%
$p_\sigma,\ T = 330$ K	5.24%	3.83%	10.35%	4.88%
$p_\sigma,\ T = 430$ K	3.11%	2.83%	6.72%	5.26%

Tabelle 5.18: Prozentuale Fehler zwischen den Vorhersagen von DesParO und den mit *ms2* simulierten Zielgrößen für die drei manuellen Modelle und das Pareto-Modell. Die Fehler lagen im Bereich der statistischen Unsicherheiten, außer bei den Verdampfungsenthalpien für die beiden höchsten Temperaturen und dem Dampdruck für die höchste Temperatur. Im ersten Fall wurde von der Toleranzvorhersage von DesParO allerdings eine Modellunsicherheit von über 2% und im letzten Fall von über 10% ausgegeben.

Kraftfeldparameter erhalten werden können. Durch das Metamodell können lediglich Parameter detektiert werden, welche sich in der Nähe eines Minimums befinden. Ob es sich dabei um ein globales Minimum handelt, ist erst mit einer anschließenden Feinjustierung festzustellen.

Güte des Metamodells Ein weiteres Problem besteht in der Tatsache, daß DesParO lediglich dazu in der Lage ist, das Metamodell zu minimieren. Dabei müssen hohe Interpolationsfehler ausgeschlossen werden. Im folgenden werden daher die Vorhersagen von DesParO für die zu optimierenden Zielgrößen mit Simulationsergebnissen sowohl für die Kraftfeldparameter aus der manuellen Justierung als auch der Pareto-Front-Extraktion miteinander verglichen: Tabelle 5.18 zeigt die prozentualen Fehler zwischen den Modellvorhersagen von DesParO und den simulierten Zielgrößen. Lediglich im Falle der Verdampfungsenthalpien für $T = 330$ K und $T = 430$ K sowie im Falle des Dampfdrucks für $T = 430$ K lagen die Fehler nicht im Bereich der statistischen Unsicherheiten. Allerdings gab die Toleranzvorhersage von DesParO hierfür auch sehr hohe Unsicherheiten beim Interpolationsmodell aus: Im ersten Fall lagen diese bei über 2% und im letzten Fall sogar bei über 10%. Anhand der Fehler zwischen den simulierten Zielgrößen und ihren entsprechenden experimentellen Referenzdaten wurde jedoch festgestellt, daß alle vier Modelle für eine Feinoptimierung mit GROW geeignet waren.

5.3.3 Feinoptimierung mit GROW

In diesem Abschnitt werden Feinoptimierungen mithilfe von GROW beschrieben, basierend auf den in Maaß u. a. (2010) sowie den in Abschnitt 5.3.2 mittels DesParO erhaltenen Startparametern (siehe Tabelle 5.17). Es wurden die gleichen Simulationseinstellungen für Epoxid verwendet wie in Abschnitt 5.3.2, mit dem einzigen Unterschied, daß für die vorliegende Feinoptimierung zu sieben anstatt nur zu drei verschiedenen Temperaturen parallel simuliert wurde: $T/\mathrm{K} \in \{230, 260, 300, 330, 375, 400, 430\}$. Bei der Optimierung war folgendes zu beachten:

- Als erstes wurde stets die Methode des steilsten Abstiegs verwendet mit $h = 0.01$ und $\zeta_A = 0.2$. Im Gegensatz zu den mit *Gromacs* realisierten Optimierungsläufen wurde dann zunächst $h = 0.001$ gesetzt, bevor eine andere Optimierungsmethode eingesetzt wurde. Der Grund hierfür liegt in der Tatsache, daß aufgrund der Methode der graduellen Einsetzung zur Berechnung des chemischen Potentials das statistische Rauschen auf den zu optimerenden VLE-Daten zum einen besser abschätzbar und zum anderen deutlich geringer war als bei *Gromacs*. Besonders der Druck unterlag im Falle von *Gromacs* stets sehr hohen Schwankungen.
- Es wurde die folgende Gewichtung vorgenommen:

$$\forall_T \; w_{\rho_l,T} = w_{p_\sigma,T} = 10 w_{\Delta_v H,T}, \; \sum_{i,T} w_{i,T} = 1.$$

  Hierbei bestand die Motivation darin, daß der Dampfdruck zu Beginn am stärksten vom Experiment abwich und somit innerhalb des Optimierungsprozesses auf diesen zunächst mehr Wert gelegt werden muß. Daher wurde er genauso hoch wie die Siededichte gewichtet. Da die Verdampfungsenthalpie über die Clausius-Clapeyron-Gleichung (3.65) mit dem Dampfdruck in Beziehung steht, wurden hierfür kleinere Gewichte verwendet.
- Der zulässige Bereich war $\Omega = (20, 20)$, das heißt, ε und σ wurden jeweils um maximal 20% verändert.
- Die physikalische Einheit von ε war wieder kJ/mol und die von σ wieder nm. Die Einheiten K und Å haben zu Parametern unterschiedlicher Größenordnung geführt, was bei der Gradientenberechnung numerische Probleme verursacht hatte.

Kombinationsworkflow 1 Zunächst wurde eine Feinoptimierung ausgehend von den Modellparametern aus Maaß u. a. (2010) durchgeführt: Bei beiden Epoxid-Modellen (Punktladungs- und Dipolmodell) wurden als Einheiten von ε und σ zunächst fälschlicherweise K und Å verwendet. Eine Iteration mit der Methode des steilsten Abstiegs veränderte ε nur minimal, σ im Gegensatz dazu jedoch deutlich. Dadurch unterschieden sich die im folgenden angegebenen Startparameter bei den beiden Modellen.

Optimierungsergebnisse für das Ladungsmodell Zunächst werden die Optimierungsergebnisse für das Ladungsmodell angegeben: Tabelle 5.19 zeigt die entsprechenden Resultate. Im Laufe der Optimierung wurden sowohl die Energieparameter (ε) als auch die Längenparameter (σ) gleichmäßig variiert, da alle Eigenschaften verändert wurden, insbesondere die Dampfdrücke,

k	$x^{(k)}$	MAPE ρ_l	MAPE $\Delta_v H$	MAPE p_σ	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
Steilster Abstieg, $h = 0.01$, $\Omega = (20, 20)$							
0	0.74514 0.71444 0.35270 0.28297	4.07%	10.28%	31.35%	0.0403	1.21746 0.53659 0.78954 -0.01131	1.55
1	0.73010 0.70781 0.34295 0.28311	8.23%	7.22%	19.50%	0.0191	0.67622 0.18960 0.26955 -0.34523	0.83
2	0.71202 0.70274 0.33574 0.29234	8.74%	2.92%	4.00%	$3.7{\cdot}10^{-3}$	0.15289 0.03513 -0.14008 -0.14821	0.26
Steilster Abstieg, $h = 0.001$, $\Omega = (20, 20)$							
3	0.70650 0.69918 0.34996 0.30739	1.73%	2.12%	7.88%	$2.8{\cdot}10^{-3}$	-0.82833 -0.57212 -0.26785 -0.49546	1.15
4	0.70680 0.69939 0.35006 0.30757	1.77%	2.13%	7.10%	$2.6{\cdot}10^{-3}$	0.37826 -0.08265 0.36402 -0.02677	0.53
5	0.70410 0.69998 0.34746 0.30776	0.86%	1.67%	6.13%	$1.6{\cdot}10^{-3}$	-0.07128 -0.54705 -0.05752 0.02300	0.56
6	0.70418 0.70056 0.34752 0.30773	0.86%	1.61%	5.76%	$1.7{\cdot}10^{-3}$	0.10114 0.00448 0.27059 0.13572	
Variante des Verfahrens nach Stoll, $h = 0.001$, $\Omega = (10, 5)$							
7	0.70495 0.69979 0.34714 0.30697	0.57%	1.60%	5.37%	$1.4{\cdot}10^{-3}$	nicht ermittelt	
8	0.70505 0.69959 0.34702 0.30677	0.56%	1.67%	5.53%	$1.2{\cdot}10^{-3}$	nicht ermittelt	
9	0.70505 0.69958 0.34701 0.30676	0.55%	1.63%	5.23%	$1.1{\cdot}10^{-3}$	nicht ermittelt	

Tabelle 5.19: Optimierungsergebnisse von VLE-Daten für Epoxid, basierend auf einem Ladungsmodell: Innerhalb von sechs Iterationen mit der Methode des steilsten Abstiegs und drei Iterationen mit der Variante des Verfahrens nach Stoll wurde die Fehlerfunktion um mehr als eine Größenordnung verkleinert. Verdampfungsenthalpie und Dampfdruck wurden zunächst auf Kosten der Siededichte deutlich verbessert. Dann wurde die Siededichte auf Kosten des Dampfdrucks verbessert. Die zu optimierenden Kraftfeldparameter waren $\varepsilon(\mathrm{CH_2}), \varepsilon(\mathrm{O}), \sigma(\mathrm{CH_2})$ und $\sigma(\mathrm{O})$, das heißt, es gilt $x^{(k)} = \left(\varepsilon(\mathrm{CH_2})^{(k)}, \varepsilon(\mathrm{O})^{(k)}, \sigma(\mathrm{CH_2})^{(k)}, \sigma(\mathrm{O})^{(k)}\right)$.

auf die beide Kraftfeldparameter eine hohe Auswirkung haben. Am meisten verändert wurde $\varepsilon(\mathrm{O})$, welches um 5.38% verkleinert wurde. Weiterhin wurden $\varepsilon(\mathrm{CH_2})$ und $\sigma(\mathrm{O})$ verkleinert sowie $\sigma(\mathrm{CH_2})$ vergrößert.

Innerhalb von sechs Iterationen mit der Methode des steilsten Abstiegs und drei Iterationen mit der Variante des Verfahrens nach Stoll konnte die Fehlerfunktion um mehr als eine Größenordnung verkleinert werden. Ein CG-Verfahren konnte nach der Methode des steilsten Abstiegs keine Verbesserungen mehr erzielen. Zunächst wurden nur Verdampfungsenthalpie und Dampfdruck verbessert, und zwar auf Kosten der Siededichte. Nach der Verkleinerung von h in der dritten Iteration wurde die Siededichte auf Kosten des Dampfdrucks verbessert. Ab $x^{(5)}$ pendelten sich alle drei Eigenschaften ein.

Für die Gradientenberechnung waren stets vier Simulationen erforderlich. Die maximale Anzahl an Armijo-Schritten wurde wie in Abschnitt 5.1 auf 10 gesetzt. Insgesamt waren für die Optimierung mit der Methode des steilsten Abstiegs effektiv 41 und für die anschließende Optimierung mit der Variante des Verfahrens nach Stoll effektiv 9 Simulationen erforderlich, so daß eine Gesamtanzahl von effektiv 50 Simulationen resultierte. Der MAPE-Wert bezüglich der Siededichte lag bei über 0.5%, der bezüglich der Verdampfungsenthalpie bei über 1% und der bezüglich des Dampfdrucks bei über 5%. Das Ladungsmodell mit der etwas verzerrten Geometrie konnte also noch keine optimalen Kraftfeldparameter liefern.

Optimierungsergebnisse für das Dipolmodell Tabelle 5.20 und Abbildung 5.44 zeigen die Optimierungsergebnisse im Falle des Dipolmodells. Im Laufe der Optimierung wurden wieder sowohl die Energieparameter (ε) als auch die Längenparameter (σ) gleichmäßig verändert. Am meisten verändert wurde $\varepsilon(\mathrm{O})$, welches um 5.32% verkleinert wurde. Alle anderen Parameter wurden ebenfalls verkleinert. Der größte Unterschied zwischen dem resultierenden Modell aus Tabelle 5.19 und 5.20 bestand bei $\sigma(\mathrm{CH_2})$: Im Falle des Dipolmodells war $\sigma(\mathrm{CH_2})$ um 23.76% kleiner.

Innerhalb von 14 Iterationen mit der Methode des steilsten Abstiegs und zwei Iterationen mit der Variante des Verfahrens nach Stoll konnte die Fehlerfunktion um mehr als zwei Größenordnungen verkleinert werden. Ein CG-Verfahren konnte nach der Methode des steilsten Abstiegs keine Verbesserungen mehr erzielen. Zunächst wurden nur Verdampfungsenthalpie und Dampfdruck verbessert, und zwar auf Kosten einer leichten Verschlechterung der Siededichte. Erst ab $x^{(10)}$ wurde auch die Siededichte verbessert.
Insgesamt waren für die Optimierung mit der Methode des Steilsten Abstiegs effektiv 77 und für die anschließende Optimierung mit der Variante des Verfahrens nach Stoll effektiv 6 Simulationen erforderlich, so daß eine Gesamtanzahl von effektiv 83 Simulationen resultierte. Abbildung 5.44 zeigt, daß alle Eigenschaften nach der gradientenbasierten Optimierung mit den experimentellen Daten bis auf ihre gewünschten Toleranzen genau übereinstimmten. Auch der Trend der experimentellen Kurven wurde sehr gut wiedergegeben. Der Fehler in Bezug auf die Siededichte lag zu allen Temperaturen unter 0.5%, außer im Falle von $T = 375$ K (-0.63%) und $T = 400$ K (-0.65%), der in Bezug auf die Verdampfungsenthalpie unter 1% und der in Bezug auf den Dampfdruck unter 5%, sogar unter 3%. Weiterhin zeigt die Abbildung deutlich, daß das erhaltene Kraftfeld bezüglich Verdampfungsenthalpie und Dampfdruck deutlich besser war als in Eckl u. a. (2008a). Beim Dampfdruck konnten zusätzlich geringere statistische Unsicherheiten erzielt werden, was auf die Methode der graduellen Einsetzung zur Berechnung des chemischen Potentials zurückzuführen ist. In Eckl u. a. (2008a) wurde die Methode nach Widom verwendet, welche bei niedrigen Temperaturen zu hohen Unsicherheiten führt. Der Grund dafür liegt in der Tatsache, daß damals die Methode der graduellen Einsetzung noch nicht in *ms2* implementiert war.
Lediglich die Siededichten waren geringfügig schlechter, allerdings immer noch im Bereich der statistischen Unsicherheiten. Die finalen Kraftfeldparameter bei der vorliegenden Optimierung waren

$$x^{(\mathrm{opt})} := (0.70560, 0.70843, 0.36750, 0.23387)^T,$$

die in Eckl u. a. (2008a) erhaltenen hingegen

$$x^{(\mathrm{opt})}_{\mathrm{Eckl}} := (0.70456, 0.51655, 0.35266, 0.30929)^T.$$

Lediglich die LJ-Parameter der $\mathrm{CH_2}$-Gruppe waren ähnlich. Anscheinend sind in dem betrachteten zulässigen Bereich die Lösungen für Sauerstoff restringierter als für die $\mathrm{CH_2}$-Gruppe, was wieder für die Regenrinnengestalt der Fehlerfunktion spricht. Für die $\mathrm{CH_2}$-Gruppe ist der Lösungsbereich größer, das heißt, am Boden der Regenrinne befinden sich mehrere lokale Optima. Da hier sämtliche Eigenschaften bis auf statistische Unsicherheiten genau bestimmt worden sind, kann davon ausgegangen werden, daß ein globales Minimum gefunden wurde.

k	$x^{(k)}$	MAPE ρ_l	MAPE $\Delta_v H$	MAPE p_σ	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
Steilster Abstieg, $h = 0.01$, $\Omega = (20, 20)$							
0	0.74523 0.71447 0.37130 0.27060	2.41%	12.29%	38.26%	0.0607	1.40916 0.27407 -0.55772 1.24851	1.98
1	0.73998 0.71345 0.37338 0.26595	2.94%	10.21%	34.00%	0.0474	1.2449 0.16624 -0.50153 1.19312	1.80
2	0.73613 0.71293 0.37493 0.26225	3.55%	8.62%	30.06%	0.0360	1.0737 0.49751 -0.20862 1.23641	1.72
3	0.73359 0.71176 0.37542 0.25934	3.75%	7.59%	27.53%	0.0310	0.93452 0.24989 -0.76940 0.94760	1.56
4	0.73061 0.71096 0.37788 0.25631	5.40%	5.93%	23.48%	0.0226	1.06642 0.20603 -0.38721 0.84217	1.43
5	0.72721 0.71030 0.37911 0.25362	6.37%	4.40%	19.32%	0.0158	0.89735 0.38919 -0.17135 0.97966	1.39
6	0.72541 0.70952 0.37946 0.25166	6.74%	3.70%	17.72%	0.0137	0.69732 0.16328 -0.14769 0.75553	1.05
7	0.72382 0.70915 0.37979 0.24994	7.05%	2.97%	15.46%	0.0109	0.70245 0.11357 -0.19050 0.77924	1.07
8	0.72243 0.70893 0.38017 0.24840	7.45%	2.59%	14.15%	$9.6{\cdot}10^{-3}$	0.49204 0.13885 -0.13888 0.51079	0.74
9	0.72051 0.70839 0.38071 0.24640	7.97%	2.15%	10.83%	$7.6{\cdot}10^{-3}$	0.31589 -0.03144 0.04802 0.25987	0.41
10	0.71632 0.70880 0.38007 0.24296	7.84%	1.38%	6.87%	$4.4{\cdot}10^{-3}$	0.19728 0.05890 0.21726 0.21477	0.37
11	0.71332 0.70791 0.37677 0.23969	6.01%	0.98%	5.17%	$3.5{\cdot}10^{-3}$	0.03303 -0.03615 0.27645 0.04548	0.37
12	0.71253 0.70877 0.37014 0.23860	1.56%	1.67%	5.62%	$1.4{\cdot}10^{-3}$	0.16665 0.05678 -0.03999 0.20335	0.27
13	0.71056 0.70810 0.37061 0.23619	2.10%	0.98%	4.11%	$9.0{\cdot}10^{-4}$	0.07662 -0.01618 0.04971 0.05122	0.11
14	0.70619 0.70903 0.36778 0.23327	0.51%	0.52%	2.16%	$2.4{\cdot}10^{-4}$	-0.04312 -0.00936 0.08462 -0.03233	0.10
Variante des Verfahrens nach Stoll, $h = 0.001$, $\Omega = (10, 5)$							
15	0.70560 0.70844 0.36749 0.23386	0.45%	0.55%	1.75%	$1.8{\cdot}10^{-4}$	nicht ermittelt	
16	0.70560 0.70843 0.36750 0.23387	0.41%	0.49%	1.54%	$1.3{\cdot}10^{-4}$	nicht ermittelt	

Tabelle 5.20: Optimierungsergebnisse von VLE-Daten für Epoxid, basierend auf dem Dipolmodell aus Eckl u. a. (2008a): Innerhalb von 14 Iterationen mit der Methode des steilsten Abstiegs und zwei Iterationen mit der Variante des Verfahrens nach Stoll wurde die Fehlerfunktion um mehr als zwei Größenordnungen verkleinert. Verdampfungsenthalpie und Dampfdruck wurden zunächst auf Kosten der Siededichte deutlich verbessert. Erst ab $x^{(10)}$ wurden alle drei Größen gleichzeitig verbessert. Die zu optimierenden Kraftfeldparameter waren $\varepsilon(\mathrm{CH_2}), \varepsilon(\mathrm{O}), \sigma(\mathrm{CH_2})$ und $\sigma(\mathrm{O})$, das heißt, es gilt $x^{(k)} = \left(\varepsilon(\mathrm{CH_2})^{(k)}, \varepsilon(\mathrm{O})^{(k)}, \sigma(\mathrm{CH_2})^{(k)}, \sigma(\mathrm{O})^{(k)}\right)$.

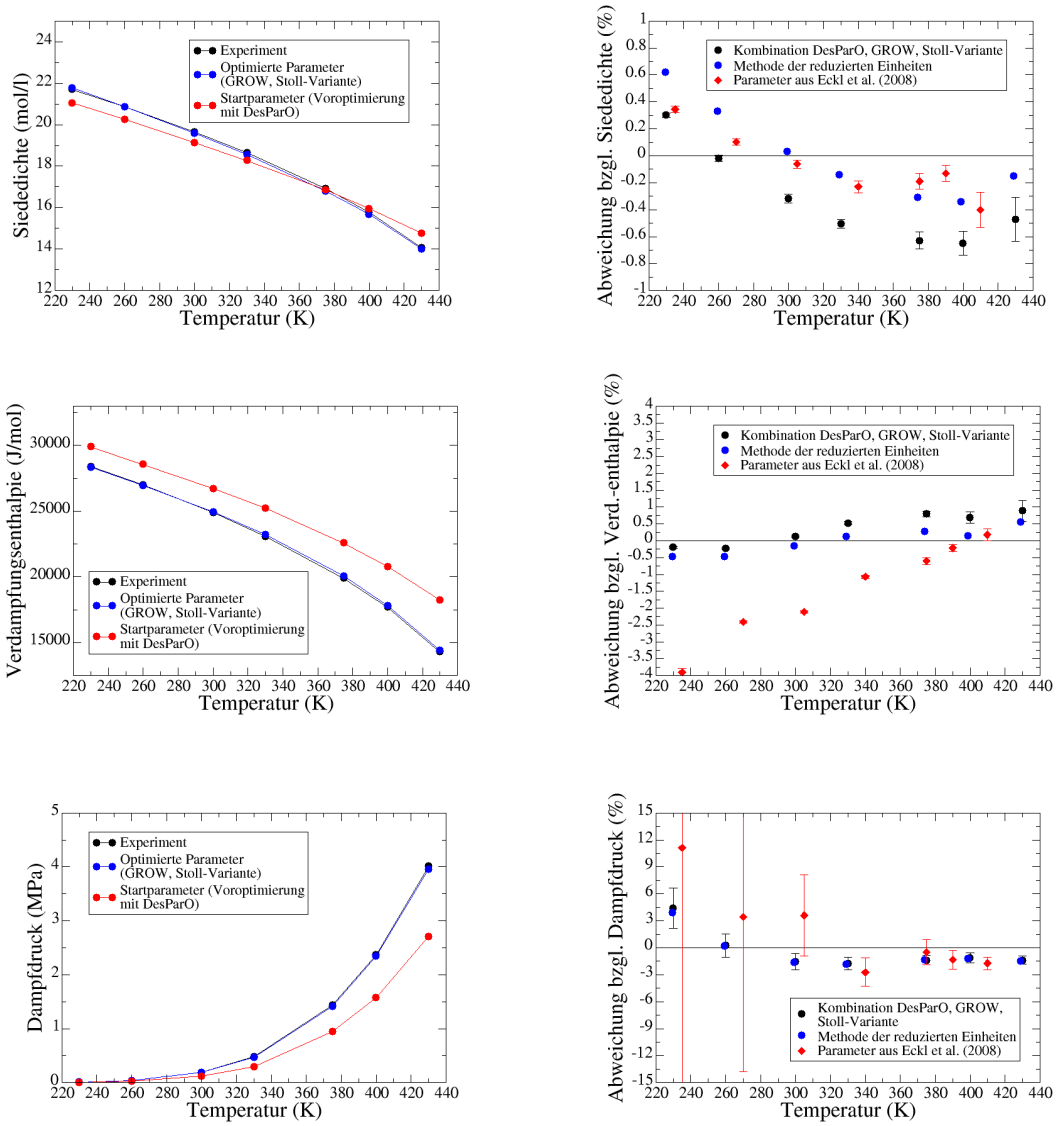


Abbildung 5.44: Optimierung der VLE-Daten (Siededichte, Verdampfungsenthalpie und Dampfdruck) für Epoxid, basierend auf dem Dipolmodell, im Vergleich zu Eckl u. a. (2008a). Links sind jeweils die Zielgrößen selbst und rechts die prozentualen Abweichungen von den experimentellen Referenzdaten angegeben. Sämtliche Zielgrößen wurden innerhalb ihrer gewünschten Toleranzwerte vorhergesagt. Weiterhin ist zu erkennen, daß mithilfe der Kombination von DesParO mit GROW und der Variante des Verfahrens nach Stoll bezüglich Verdampfungsenthalpie und Dampfdruck ein genaueres Kraftfeld mit einem geringeren Ausmaß an statistischen Unsicherheiten erhalten werden konnte als in Eckl u. a. (2008a), was auf die dort verwendete Methode nach Widom auch für niedrige Temperaturen zurückzuführen ist. Als Optimierungsmethode wurde in dieser Publikation das klassische Verfahren nach Stoll verwendet. Die hier erhaltenen Siededichten waren geringfügig schlechter. Mihilfe der nachgeschalteten Methode der reduzierten Einheiten konnte das Kraftfeld bezüglich Siededichte und Verdampfungsenthalpie nochmals signifikant verbessert werden.

Methode	T_1	T_2	T_3	T_4	T_5	T_6	T_7
Kombination	230.00	260.00	300.00	330.00	375.00	400.00	430.00
Red. Einheiten	229.36	259.28	299.17	329.09	373.96	398.89	428.81

Tabelle 5.21: Ursprüngliche Temperaturen für die Kombination von DesParO mit GROW und der Variante des Verfahrens nach Stoll sowie veränderte Temperaturen für die Methode der reduzierten Einheiten ($T_1, ..., T_7$ in K) im Falle von Epoxid.

Zielgröße	Methode	T_1	T_2	T_3	T_4	T_5	T_6	T_7	MAPE
ρ_l	Kombination	0.30%	-0.02%	-0.32%	-0.51%	-0.63%	-0.65%	-0.47%	0.41%
	Red. Einheiten	0.62%	0.33%	0.03%	-0.14%	-0.31%	-0.34%	-0.15%	0.27%
$\Delta_v H$	Kombination	-0.20%	-0.23%	0.14%	0.53%	0.79%	0.69%	0.89%	0.50%
	Red. Einheiten	-0.46%	-0.47%	-0.16%	0.12%	0.27%	0.14%	0.56%	0.31%
p_σ	Kombination	4.41%	0.28%	-1.55%	-1.74%	-1.38%	-1.13%	-1.38%	1.70%
	Red. Einheiten	3.89%	0.22%	-1.62%	-1.80%	-1.34%	-1.19%	-1.45%	1.64%

Tabelle 5.22: Relative Abweichungen vom Experiment zu allen Temperaturen $T_1, ..., T_7$ und MAPE-Werte für Siededichte, Verdampfungsenthalpie und Dampfdruck bei der Kombination von DesParO mit GROW und der Variante des Verfahrens nach Stoll sowie der Methode der reduzierten Einheiten im Falle von Epoxid: Mit letzterer konnten Siededichte und Verdampfungsenthalpie signifikant verbessert werden.

Der Rechenaufwand zum Erhalt dieses Kraftfeldes war allerdings mit 83 Simulationen noch relativ hoch. Im Falle des Ladungsmodells, für welches zwar keine optimalen Kraftfeldparameter erzielt werden konnten, waren nur 50 Simulationen erforderlich. Daher wurde zum Vergleich bei der Methode des steilsten Abstiegs eine effiziente Gradientenberechnung (siehe Abschnitt 3.6.2) durchgeführt. Es waren dabei 13 Iterationen und 39 (effektiv 30) Simulationen notwendig, um einen Parametervektor zu erhalten, der im Bereich von $x^{(10)}$ aus der originalen Optimierung lag. Dabei wurde der Gradient insgesamt zehnmal effizient berechnet. Danach waren noch 26 Simulationen mit klassischer Gradientenberechnung notwendig, um zu $x^{(14)}$ zu gelangen, also waren es insgesamt bei der Methode des steilsten Abstiegs 56. Ab $x^{(10)}$ befand sich der Algorithmus bereits in der Nähe des Minimums, so daß eine effiziente Gradientenberechnung nicht mehr möglich war. Zusammen mit der Variante des Verfahrens nach Stoll (effektiv 6 Simulationen) waren 62 Simulationen durchzuführen. Also konnten durch die effiziente Gradientenberechnung 21 Simulationen eingespart werden, also etwa ein Drittel.

Da das Ziel der ersten Untersuchung darin bestand, ein möglichst gutes Kraftfeld zu erhalten, wurde nach der Kombination von DesParO mit GROW und der Variante des Verfahrens nach Stoll noch die Methode der reduzierten Einheiten aus Abschnitt 3.6.4 nachgeschaltet. Die absoluten Abweichungen bezüglich der Siededichte vom Experiment zu den vier höchsten Temperaturen lagen noch etwa bei oder etwas über 0.5% und die bezüglich der Verdampfungsenthalpie deutlich über 0.5%. Die bezüglich des Dampfdrucks lagen weit unter 3%, außer im Falle der niedrigsten Temperatur, wo die Abweichung 4.41% betrug. Da zu dieser Temperatur allerdings, wie mehrfach motiviert, die Durchführung einer Simulation am schwierigsten ist und die Abweichung immerhin noch unter 5% lag, ist das Kraftfeld bezüglich des Dampfdrucks als optimal zu bewerten. Bezüglich Siededichte und Verdampfungsenthalpie hingegen sind jedoch noch Verbesserungen möglich.

Mit nur einer Newton-Iteration konnte die entsprechende Fehlerfunktion bei der Methode der reduzierten Einheiten (vergleiche Abschnitt 3.6.4) minimiert werden. Es wurde näherungsweise ein stationärer Punkt mit positiv definiter Hesse-Matrix gefunden. Die innerhalb der Fehlerfunktion verwendeten Zustandsgleichungen für die VLE-Eigenschaften stammen aus Buckles u. a. (1999). Mithilfe dieser Zustandsgleichungen wurden auch die experimentellen Zielgrößen zu den sieben Temperaturen $T_1, ..., T_7$ ermittelt. Es ist zu beachten, daß nur Siededichte und Verdampfungsenthalpie in die Fehlerfunktion miteinbezogen wurden, da der Dampfdruck bereits als optimal anzusehen war. Wie in Abschnitt 3.6.4 diskutiert, entsprechen die aus der Methode der reduzierten Einheiten resultierenden Zielgrößen etwas geringeren Temperaturen. Diese sind im Vergleich zu den ursprünglichen Temperaturen in Tabelle 5.21 angegeben. Tabelle 5.22 und die rechte Spalte von Abbildung 5.44 zeigen die Ergebnisse der Methode der reduzierten Einheiten im Vergleich zur Kombination von DesParO mit GROW und der Variante des Verfahrens nach Stoll: Siededichte und Verdampfungsenthalpie konnten signifikant verbessert werden, während der Dampfdruck innerhalb der statistischen Unsicherheiten gleich blieb. Sämtliche Abweichungen bezüglich der Siededichte lagen nun weit unter 0.5%, außer im Falle der niedrigsten Temperatur, was aus den oben erwähnten Gründen akzeptabel ist. Das Kraftfeld aus Eckl u. a. (2008a) war bezüglich der Siededichte allerdings immer noch leicht besser als das hier erzielte. Die Verdampfungsenthalpien zu den vier höchsten Temperaturen konnten deutlich verbessert werden. Dies ging jedoch auf Kosten der beiden niedrigsten Temperaturen, wobei die Abweichungen vom Experiment jedoch immer noch unter 0.5% lagen.

Das Ziel der ersten Untersuchung wurde somit mit großem Erfolg erreicht. Das in dieser Arbeit empfohlene VLE-Kraftfeld für Epoxid, welches insgesamt besser ist als das aus Eckl u. a. (2008a), besteht aus den folgenden Parametern:

$$\varepsilon(\mathrm{CH_2}) = 0.7036 \text{ kJ/mol},\ \varepsilon(\mathrm{O}) = 0.7065 \text{ kJ/mol},\ \sigma(\mathrm{CH_2}) = 0.3672 \text{ nm und } \sigma(\mathrm{O}) = 0.2337 \text{ nm}. \tag{5.4}$$

Zu diesem Kraftfeld gehört allerdings auch ein anderes Dipolmoment, welches sich von 2.459 D auf 2.447 D verkleinert hat.

Anhand der beiden Optimierungsbeispiele ist deutlich zu erkennen, daß für eine Feinjustierung das molekulare Modell entscheidend ist. Das Ladungsmodell mit testweise verzerrter Geometrie hat nicht zum Erfolg geführt, wohingegen das Dipolmodell ein optimales VLE-Kraftfeld lieferte.

Kombinationsworkflow 2 Im Falle der Startparameter aus Abschnitt 5.3.2 (zweite Untersuchung) wurde lediglich das Ladungsmodell betrachtet. Zum einen wurden die aus der manuellen Justierung und zum anderen die aus der Pareto-Front-Extraktion ermittelten Kraftfeldparameter (Tabelle 5.17) als Startparameter verwendet. Tabelle 5.23 gibt die entsprechenden Ergebnisse hierzu an, die im folgenden im Detail diskutiert werden:

- **Modell 1:** Alle Parameter wurden verkleinert. Der Trend innerhalb der Parameter wurde beibehalten, das heißt, es gilt nach wie vor $\varepsilon(\mathrm{CH_2}) > \varepsilon(\mathrm{O})$ und $\sigma(\mathrm{CH_2}) > \sigma(\mathrm{O})$. Der

Modell	# It. SA	# It. Stoll	$\varepsilon(CH_2)$	$\varepsilon(O)$	$\sigma(CH_2)$	$\sigma(CH_2)$
1	2	1	0.7408	0.5841	0.3463	0.3005
2	5	2	0.7412	0.5444	0.3515	0.2763
3	5	0	0.6832	0.7821	0.3367	0.3192
Pareto	5	1	0.6832	0.7821	0.3367	0.3192

Tabelle 5.23: Aus der Feinoptimierung mit GROW resultierende Kraftfeldparameter (ε in kJ/mol und σ in nm) im Falle der drei manuellen Justierungen und des Pareto-Modells. Die Startparameter zu den jeweiligen Optimierungsläufen sind in Tabelle 5.17 angegeben. Die Parametervektoren unterschieden sich sehr stark voneinander, das heißt, die Ergebnisse einer Voroptimierung mit DesParO können über den Parameterraum hinweg weit verstreut liegen.

MAPE-Wert für die Siededichte lag bei 0.78%, der für die Verdampfungsenthalpie bei 0.82% und der für den Dampfdruck bei 3.51%. Verdampfungsenthalpie und Dampfdruck wurden auf Kosten der Siededichte innerhalb der Optimierung signifikant verbessert.

- **Modell 2:** Die ersten drei Parameter wurden verkleinert und $\sigma(O)$ vergrößert. Der Trend innerhalb der Parameter wurde ebenfalls beibehalten. Der MAPE-Wert für die Siededichte lag bei 2.11%, der für die Verdampfungsenthalpie bei 0.81% und der für den Dampfdruck bei 2.87%. Verdampfungsenthalpie und Dampfdruck wurden auch hier auf Kosten der Siededichte innerhalb der Optimierung signifikant verbessert.

- **Modell 3:** Hierbei wurden $\varepsilon(CH_2)$ sowie $\sigma(O)$ vergrößert und $\varepsilon(O)$ sowie $\sigma(CH_2)$ verkleinert. Der Trend bezüglich ε wurde umgekehrt. Vor der Optimierung galt $\varepsilon(CH_2) < \varepsilon(O)$. Nach der Optimierung lag der gleiche Trend wie bei Modell 1 und 2 vor. Der MAPE-Wert für die Siededichte lag bei 1.90%, der für die Verdampfungsenthalpie bei 1.70% und der für den Dampfdruck bei 4.80%. Verdampfungsenthalpie und Dampfdruck wurden auch hier auf Kosten der Siededichte innerhalb der Optimierung signifikant verbessert.

- **Pareto-Modell:** Hierbei wurden $\varepsilon(CH_2)$ sowie $\varepsilon(O)$ vergrößert und $\sigma(O)$ sowie $\sigma(CH_2)$ verkleinert. Der Trend wurde beibehalten, allerdings gilt in diesem Fall $\varepsilon(CH_2) < \varepsilon(O)$ und $\sigma(CH_2) < \sigma(O)$. Der MAPE-Wert für die Siededichte lag bei 7.66%, der für die Verdampfungsenthalpie bei 2.10% und der für den Dampfdruck bei 7.29%. Alle drei Eigenschaften wurden verbessert. Es ist zu beachten, daß der MAPE-Wert für die Siededichte anfangs bei über 14% lag.

Eine Voroptimierung durch DesParO kann sehr unterschiedliche Kraftfeldparameter liefern, die sich in den Einzugsbereichen unterschiedlicher Optima befinden. Nur im Falle der Voroptimierung aus Maaß u. a. (2010), welche auch manuell geschah, konnte der Einzugsbereich eines globalen Optimums, wo die Fehlerfunktion verschwindet, erreicht werden. Dies war jedoch nur mit dem Dipolmodell möglich, welches anscheinend für Epoxid am besten geeignet ist. Die in dieser Arbeit durchgeführte Voroptimierung lieferte vier verschiedene Modelle, von denen aus GROW gegen drei unterschiedliche Optima konvergierte. GROW war jedoch in keinem Fall dazu in der Lage, alle drei Zielgrößen zu optimieren. Die Modelle 2 und 3 lagen im Einzugsbereich des gleichen Optimums, was anhand von Tabelle 5.23 deutlich zu erkennen ist. Im Falle von Modell 3 konnte GROW den Trend innerhalb der Parameter umkehren. Da das Ergebnis des Pareto-Modells im Gegensatz zu den anderen äußerst schlecht war, läßt sich vermuten, daß für ein gutes Modell $\varepsilon(CH_2) > \varepsilon(O)$ und $\sigma(CH_2) > \sigma(O)$ gelten muß.

Fazit und Selektion eines globalen Voroptimierers Die aus einer Voroptimierung mit DesParO resultierenden Kraftfeldparameter können über den Parameterraum hinweg weit verstreut liegen und sich in den Einzugsbereichen verschiedener Minima befinden. Auch CMA-ES kann je nach Restriktion gegen unterschliedliche Minima konvergieren, allerdings ist dabei die Wahl eines geeigneten Abbruchkriteriums einfacher, da sich die Modelle irgendwann auf einen bestimmten Bereich fokussieren. Die DesParO-Modelle hingegen waren hier viel weiter verstreut. Um dies zu vermeiden, kann selbstverständlich die Anzahl an Modellen für DesParO deutlich erhöht werden. Allerdings weiß man *a priori* nie, wie viel Rechenaufwand nötig ist, um ein Startmodell im Einzugsbereich eines globalen Minimums zu finden. Dies kann erst nach der Feinoptimierung durch GROW festgestellt werden. CMA-ES hingegen kam mit einem akzeptablen Rechenaufwand aus (vergleiche Abschnitte 4.5.1 und 5.2.1), und das Finden eines geeigneten Abbruchkriteriums ist weitaus einfacher. Daher ist CMA-ES für die vorliegende Problemstellung der Kraftfeldoptimierung Molekularer Simulationen vorzuziehen.

5.4 Wissenschaftlich und industriell relevante Applikation: Ionische Flüssigkeiten

Neben Dipropylenglykoldimethylether und Ethylenoxid wird noch eine dritte wissenschaftlich und industriell äußerst relevante Applikation betrachtet: Ionische Flüssigkeiten (**I***onic* **L***iquids (ILs)).* Diese sind in der Arbeitsgruppe *Computational Chemical Engineering (CoChE)* des Fraunhofer-Instituts SCAI von besonderer Bedeutung, wo spezifische relevante Eigenschaften mittels Molekularer Simulationen vorhergesagt werden. Aufgrund ihrer vielseitigen Einsetzbarkeit sind dabei die zugehörigen Kraftfelder an mehrere verschiedene physikalische Zielgrößen anzupassen wie zum Beispiel Dichte, Diffusion, Reorientierungszeit und Viskosität. Inwieweit dies mithilfe von GROW möglich ist, wird in diesem Abschnitt anhand von Beispielen diskutiert. Da es sich hier um Simulationen im flüssigen Zustand handelte, wurden wieder NPT-MD-Simulationen mit *Gromacs* verwendet. Auf dem in Abschnitt 5.2.2 erwähnten Rechencluster benötigte eine Simulation, parallelisiert auf zwölf Knoten, bis zu einem Tag, was darauf zurückzuführen ist, daß $4 \cdot 10^7$ Produktionsschritte für die akkurate Berechnung der oben genannten Größen erforderlich war.
In Abschnitt 5.4.1 werden spezifische physikalische und chemische Eigenschaften sowie Besonderheiten von ILs beschrieben. Dabei liegt der Fokus vor allem auf der Beantwortung der Frage, inwieweit ILs nicht nur aus industrieller, sondern auch aus wissenschaftlicher Sicht von enormer Bedeutung sein können. Abschnitt 5.4.2 behandelt dann die Kraftfeldoptimierung von ILs durch GROW anhand einiger Beispiele.

5.4.1 Allgemeines zu Ionischen Flüssigkeiten

Wie bereits erwähnt, zeichnen sich Ionische Flüssigkeiten durch besondere chemische und physikalische Eigenschaften aus. Die wichtigsten werden im vorliegenden Abschnitt beschrieben. Wie der Name schon sagt, bestehen derartige Substanzen aus Ionen. Das Kation ist dabei organisch und asymmetrisch aufgebaut. Das Anion ist hingegen anorganisch und oftmals, im Gegensatz zu Anionen, die man bei Salzen vorfindet, äußerst groß. Bei ILs handelt es sich um sogenannte

flüssige Salze, was jedoch nicht mit der Lösung eines Salzes in Wasser zu vergleichen ist.
Die wichtigsten Eigenschaften von ILs sind die folgenden:

- **Niedriger Dampfdruck:** ILs weisen einen sehr großen Flüssigkeitsbereich auf und zeichnen sich durch einen äußerst niedrigen Dampfdruck aus. Ihre geringe Flüchtigkeit macht sie interessant für viele Anwendungen, zum Beispiel als Schmiermittel. Weiterhin besitzen ILs eine hohe Energiedichte.

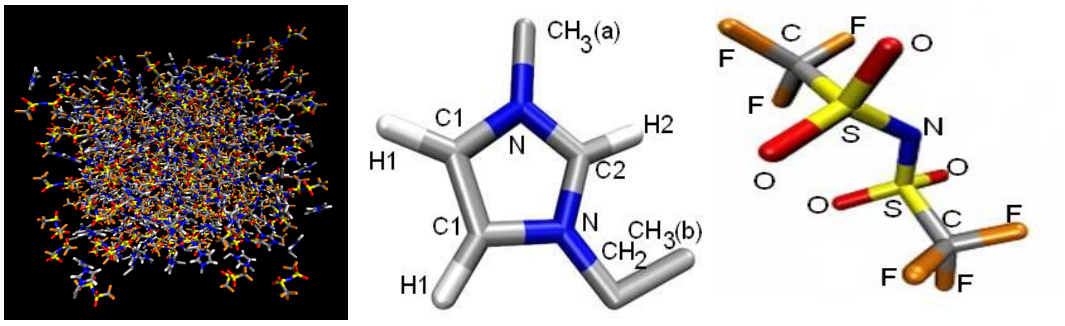
- **Niedriger Schmelzpunkt:** Die meisten ILs haben einen Schmelzpunkt von unter 100°C. Dies ist vor allem auf die starke Delokalisierung der Elektronen bei beiden Ionen zurückzuführen, wodurch die Ionischen Wechselwirkungen, wie sie beispielsweise in Salzen auftreten, deutlich geschwächt werden. Am interessantesten sind allerdings diejenigen ILs, die knapp unter, knapp über oder bei Raumtemperatur flüssig werden (sogenannte **R***oom* **T***emperature* **I***onic* **L***iquids (RTILs)*). Ethylammoniumnitrat war Anfang des 20. Jahrhunderts die erste in der Literatur erschienene IL, der ein Schmelzpunkt von 12°C nachgewiesen werden konnte. Die ersten Synthesen von RTILs, welche bei industriellen Prozessen eingesetzt wurden, gelangen hingegen erst Ende des 20. Jahrhunderts.

- **Außergewöhnliche Lösungseigenschaften:** ILs sind dazu in der Lage, eine Vielzahl von Substanzen zu lösen. Die Löslichkeit in Wasser oder organischen Lösungsmitteln ist dabei so gut wie frei wählbar. Je nach Seitenketten von Kation beziehungsweise Anion erhält man verschiedene Löslichkeiten, was auch auf Schmelzpunkt und Viskosität zutrifft.

Je nach Wahl von Kation beziehungsweise Anion ändern sich Reaktivität und physikochemische Eigenschaften von ILs. Dies führt zu gezielten industriellen Synthesen, das heißt, es ist möglich, ILs so herzustellen, daß sie bestimmte vorher festgelegte Eigenschaften aufweisen. Durch die große Anzahl an Kombinationsmöglichkeiten von Kationen und Anionen gibt es allerdings auch eine enorm große Anzahl an ILs. Dies macht ILs allerdings wissenschaftlich gesehen zu einem sehr breiten und interessanten Forschungsgebiet. Ein grundlegendes Strukturverständnis und die Detektion von Auswirkungen von Änderungen in Aufbau und Struktur einer IL auf physikochemische Eigenschaften können zu akkuraten Vorhersagen von Eigenschaften von ILs führen.

Zur Bestimmung eines geeigneten Kraftfeldes für ILs sind verschiedene physikalische Eigenschaften zu berücksichtigen, was auf die vielseitige industrielle Anwendbarkeit zurückzuführen ist: ILs werden zum Beispiel als Elektrolyte eingesetzt. In derartigen elektrochemischen Prozessen spielen vor allem Diffusion und Ladungstransportverhalten eine große Rolle. Bei ihrem Einsatz als Trennmittel in katalytischen Prozessen sind Grenzflächen- und Mischungsverhalten von besonderer Bedeutung. Weiterhin werden sie in der chemischen Verfahrenstechnik eingesetzt, das heißt, sie spielen bei chemischen Reaktionen zur Synthese bestimmter Stoffe eine große Rolle. Für die Reaktivität von ILs sind Transporteigenschaften wie Diffusion und Viskosität von herausragender Bedeutung. Näheres zu ILs, vor allem Details bezüglich ihrer physikalischer und chemischer Eigenschaften, sind zum Beispiel in Wasserscheid u. Welton (2003), Rogers u. Seddon (2003) und Endres u. El Abedin (2006) nachzulesen.

In dieser Arbeit wird für eine Kraftfeldoptimierung die folgende IL betrachtet: 1-Ethyl-3-methyl-imidazolium-bis(trifluormethylsulfonyl) ($[C_2MIM][NTf_2]$). Hierbei handelt es sich um



(a) (b) (c)

Abbildung 5.45: Startkonfiguration (a), Strukturformel des Kations (b) und des Anions (c) im Falle der Ionischen Flüssigkeit [C_2MIM][NTf_2]. Die Kantenlänge der Box beträgt 45.843 nm.

eine RTIL. Neben dem niedrigen Dampfdruck weist sie eine relativ geringe Viskosität auf sowie eine äußerst hohe Stabiltät gegenüber hohen Temperaturen und Wasser.

5.4.2 Kraftfeldoptimierung für ionische Flüssigkeiten

Das Kation von [C_2MIM][NTf_2] besteht aus einem fünfatomigen Ring, welcher aus drei C- und zwei N-Atomen aufgebaut ist. Zwei dieser C-Atome, welche als C1 bezeichnet werden, sind jeweils an ein Wasserstoffatom (H1) gebunden. Das C-Atom, das zwischen den beiden N-Atomen liegt, wird C2 genannt und ist ebenfalls an ein Wasserstoffatom (H2) gebunden. Eines der Stickstoffatome ist an eine CH_3-gruppe (CH_3(a)) gebunden und das andere an eine kleine Kette, die aus einer CH_2- und einer außenstehenden CH_3-Gruppe (CH_3(b)) besteht. Für das Kation wurde ein *United-Atom-Modell* verwendet, in dem beide außenstehenden CH_3-Gruppen und die CH_2-Gruppe zusammengefaßt wurden.
Das Anion ist symmetrisch. Im Zentrum befindet sich ein Stickstoffatom, gebunden an zwei Schwefelatome, welche jeweils an zwei Sauerstoffatome und einer fluorierten Methylgruppe gebunden sind. Für das Anion wurde ein *All-Atom*-Modell verwendet.

Simulations- und Optimierungseinstellungen Bindungen und Winkel wurden wieder mithilfe des LINCS-Algorithmus fixiert. Die intramolekularen Kraftfeldparameter und die Partialladungen wurden (Lopes u. Padua, 2006) entnommen, bei den initialen LJ-Parameter wurden zwei Quellen verwendet: Für CH_3- beziehungsweise CH_2-Gruppen wurde das TraPPE-Kraftfeld für Ether, Glykole, Ketone und Aldehyde (Chen u. a., 2001a) genommen, alle anderen LJ-Parameter stammen aus einer in Köddermann (2008) durchgeführten manuellen Justierung. Sämtliche 1,4-Wechselwirkungen wurden exkludiert.
Bei den experimentellen Zielgrößen handelte es sich um Flüssigdichte, Selbstdiffusion von Kation und Anion sowie Reorientierungszeit des Kations. Die anzupassenden Kraftfeldparameter waren die LJ-Parameter von CH_3(a) und CH_2 und der Energieparameter von H2, woraus ein fünfparametriges Optimierungsproblem resultierte. Der Unterschied zu der manuellen Justierung in Köddermann (2008) und dem vorliegenden Optimierungsproblem besteht lediglich in der Tatsache, daß hier für das Kation ein *United-Atom*-Modell angenommen wurde. Daher sind

Art der Eingabe	Name der Eingabe	Eingabe	Sonstiges/Bemerkungen
Eingaben:			
Topologie	Atome/Atomgruppen	Kation: C1, C2, H1, H2, CH_2, CH_3(a), CH_3(b), N	*United-Atom*-Modell
		Anion: C, O, S, F, N	*All-Atom*-Modell
	Dipole/Quadrupole	–	
	Molare Masse	391.31 g/mol	m(C1) = m(C2) = 12.01 g/mol m(H1) = m(H2) = 1.008 g/mol $m(CH_2)$ = 14.01 g/mol $m(CH_3)$ = 15.01 g/mol m(N) = 14.01 g/mol m(O) = 16.01 g/mol m(S) = 32.07 g/mol m(F) = 19.00 g/mol
	Molekülstruktur	Kation: Ringstruktur Anion: Kettenstruktur	
	Intramolekulares Kraftfeld	Kation: 11 starre Bindungen (LINCS) 16 Winkel 36 Diederwinkel Anion: 14 starre Bindungen (LINCS) 25 Winkel 42 Diederwinkel	Quelle: Literatur (Lopes u. Padua, 2006)
	Ladungen	11 Punktladungen	Quelle: Literatur (Lopes u. Padua, 2006)
	Besonderheiten	1,4-Wechselwirkungen exkludiert	
Simulationsbox	Startkonfiguration	randomisiert	siehe Abbildung 5.45(a)
	Anzahl Moleküle	$2 \cdot 216 = 432$	
	Kantenlänge	45.843 nm	
Anzahl Zeitschritte	Prä-Prä-Äquilibrierung	20000	Schrittweite: 0.2 fs
	Präääquilibrierung	100000	Schrittweite: 2 fs
	Äquilibrierung	25000	Schrittweite: 2 fs
	Produktion	$4{\cdot}10^7$	Schrittweite: 2 fs
Optimierungsrelevantes	zu optimierende Parameter	$\sigma(CH_3(a))$, $\sigma(CH_2)$, $\varepsilon(CH_3(a))$, $\varepsilon(CH_2)$, ε(H2)	Quelle Startwerte: TraPPE-Kraftfeld (Stubbs u. a., 2004) für Gruppierungen, sonst Köddermann (2008)
	Temperaturen	300, 360	in K
	Druck	1.0	in bar
Ausgaben:			
Zielgrößen	experimentell	ρ, D (Kation/Anion), τ (Kation)	Quelle: Tokuda u. a. (2005) Köddermann u. a. (2007)
	simuliert	ρ: Gleichung (C.8) D: Gleichung (C.18) τ: Gleichung (C.22)	Einstein-Darstellung
	Toleranzen	ρ: 0.5% D, τ: unbekannt	

Tabelle 5.24: Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle der Ionischen Flüssigkeit $[C_2MIM][NTf_2]$.

sämtliche LJ-Parameter für die Gruppierungen nachzujustieren. Dabei sollen die Parameter in der Kette möglichst unverändert bleiben, da die Kette austauschbar sein soll. Daher wurde CH_3(b) nicht in die Optimierung miteinbezogen. Ein weiteres Atom, welches für die intermolekularen Wechselwirkungen ein große Rolle spielt (vergleiche Köddermann (2008)), ist H2. Allerdings wurde der Längenparameter für dieses Atom festgehalten, da der Abstand von H2 zum Anion beibehalten werden sollte. Die LJ-Parameter des Anions geben die Wechselwirkungen des Anions mit Wasser bereits genauestens wieder, zumindestens im *All-Atom*-Modell aus Köddermann (2008). Daher wurden auch diese fixiert.

Simuliert wurde eine Flüssigbox bestehend aus $2 \cdot 216 = 432$ Teilchen zu den beiden Temperaturen 300 und 360 K sowie zu Atmosphärendruck. Tabelle 5.24 zeigt sämtliche Einstellungen für Ein- und Ausgaben von Simulation und Optimierung im Falle von $[C_2MIM][NTf_2]$. Dort sind auch die Quellen für die experimentellen Daten sowie die verwendeten Gleichungen und Toleranzen für die simulierten Zielgrößen angegeben. Abbildung 5.45(a) zeigt die Startkonfiguration der Simulationsbox mit einer Kantenlänge von 45.843 nm. In Abbildung 5.45(b) ist das Kation und in Abbildung 5.45(c) das Anion dargestellt. Sie stimmen farblich mit den Molekülen in der Box überein. Die Teilchen wurden per Zufallsprinzip so angeordnet, daß keine Überlappungen auftraten und ein gleichgewichtsähnlicher Zustand bereits von vornherein realisiert werden konnte. Nach einer kurzen Prä-Prä-Äquilibrierung (20000 Zeitschritte à 0.2 fs) wurde eine Präääquilibrierung von 100000 Zeitschritten à 2 fs durchgeführt. Die Äquilibrierungszeit war aufgrund der Startkonfiguration mit nur 25000 Zeitschritten à 2 fs sehr gering. Für die Produktion allerdings waren $4 \cdot 10^7$ Zeitschritte à 2 fs erforderlich, um die statistischen Unsicherheiten der schwierig zu berechnenden Größen Diffusion und Reorientierungszeit möglichst gering zu halten.

Optimierungsergebnisse Tabelle 5.25 und Abbildung 5.46 zeigen die entsprechenden Optimierungsergebnisse. Als zulässiger Bereich wurde $\Omega = (40, 40)$ gewählt. Da Diffusion und Reorientierungszeit mit einem großen Ausmaß an statistischem Rauschen behaftet sind, wurde $h = 0.08$ gesetzt. Im Laufe der Optimierung wurden die Längenparameter (σ) deutlich mehr verändert als die Energieparameter (ε). Alle Parameter wurden zunächst vergrößert. Die Energieparameter wurden ab $x^{(3)}$ verkleinert, was auch am Richtungswechsel des Gradienten zu erkennen ist. Beide Längenparameter wurden um etwa 12% vergrößert.
Innerhalb von fünf Iterationen mit der Methode des steilsten Abstiegs konnte die Fehlerfunktion um etwa den Faktor 4 verkleinert werden. Weder ein anschließendes CG-Verfahren noch die Variante des Verfahrens nach Stoll konnten Verbesserungen erzielen, auch nicht nach einer Verkleinerung von h und des zulässigen Gebiets. Die Reorientierungszeit wurde auf Kosten der Dichte stark verbessert. Bei der Diffusion waren insgesamt kaum Änderungen feststellbar, da die Diffusionskoeffizienten des Kations verbessert und die des Anions verschlechter wurden. Sehr wahrscheinlich sind die MAPE-Werte, im Bereich der statistischen Unsicherheiten, für Kation und Anion lediglich, zumindest näherungsweise, vertauscht worden. Es ist dabei zu beachten, daß weder im Falle der Diffusion noch im Falle der Reorientierungszeit keine genauen Schätzwerte für statistische Unsicherheiten vorliegen. Diese werden daher im Bereich von allgemeinen Erfahrungswerten (5–10%) angenommen.
Für die Gradientenberechnung waren stets fünf Simulationen erforderlich. Die maximale Anzahl an Armijo-Schritten wurde wieder auf 10 gesetzt. Bei $x^{(0)}$ und $x^{(2)}$ wurde jeweils ein Armijo-

k	$x^{(k)}$	MAPE ρ	MAPE D (Kation)	MAPE D (Anion)	MAPE τ	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\nabla F\left(x^{(k)}\right)\|$
				Steilster Abstieg, $h = 0.08$, $\Omega = (40, 40)$				
0	0.37500 0.39500 0.81482 0.38247 0.18828	3.29%	12.2%	6.0%	28.6%	0.2216	-3.25929 -4.28185 -0.30085 -0.51690 0.14437	5.42
1	0.39904 0.42658 0.81704 0.38628 0.18722	5.58%	6.0%	7.5%	17.4%	0.0958	-2.09056 -2.35493 -0.43567 -0.63382 -0.23210	3.25
2	0.40246 0.43043 0.81775 0.38732 0.18760	5.89%	0.8%	14.2%	13.2%	0.0874	-1.10617 -0.97851 -0.03340 -0.33816 -0.20626	1.53
3	0.42322 0.44880 0.81838 0.39367 0.19147	7.76%	8.2%	15.2%	3.3%	0.0751	0.54000 1.01481 0.40297 0.67446 -0.04927	1.39
4	0.42045 0.44359 0.81631 0.39021 0.19172	7.39%	5.6%	14.2%	6.1%	0.0691	0.09508 0.41606 0.29026 0.42330 0.01567	0.67
5	0.42017 0.44234 0.81544 0.38894 0.19167	7.35%	3.2%	12.5%	6.2%	0.0563	-0.20978 0.05985 0.31858 0.42736 0.16451	0.60

Tabelle 5.25: Optimierungsergebnisse für die Ionische Flüssigkeit [C_2MIM][NTf_2]. Die zu optimierenden Zielgrößen waren ρ, D (für Kation und Anion) und τ (für das Kation). Da für Diffusion und Reorientierungszeit höhere statistische Unsicherheiten angenommen werden müssen als für die Dichte, sind die entsprechenden MAPE-Werte nur bis zur ersten Nachkommastelle angegeben. Innerhalb von fünf Iterationen mit dem Verfahren des steilsten Abstiegs wurde die Fehlerfunktion um etwa den Faktor 4 verkleinert. Die Reorientierungszeiten wurden drastisch verbessert, wohingegen die Dichten immer stäker vom Experiment abwichen. Die Diffusionskoeffizienten des Kations wurden verbessert und die des Anions verschlechtert. Insgesamt waren kaum Änderungen bezüglich der Diffusion festzustellen. Die zu optimierenden Kraftfeldparameter waren $\sigma(CH_3(a))$, $\sigma(CH_2)$, $\varepsilon(CH_3(a))$, $\varepsilon(CH_2)$ und $\varepsilon(H2)$, das heißt, es gilt $x^{(k)} = \left(\sigma(CH_3(a))^{(k)}, \sigma(CH_2)^{(k)}, \varepsilon(CH_3(a))^{(k)}, \varepsilon(CH_2)^{(k)}, \varepsilon(H2)^{(k)}\right)$.

Schritt benötigt, bei $x^{(1)}$ zwei, bei $x^{(3)}$ drei und bei $x^{(4)}$ fünf. Insgesamt waren somit für die Optimierung 52 Simulationen (effektiv 43) erforderlich.
Es stellt sich nun die Frage, wie das erhaltene Kraftfeld zu bewerten ist. Im Falle von Ionischen Flüssigkeiten sind MAPE-Werte kleiner als 10% für die Diffusion aufgrund der statistischen Unsicherheiten als äußerst gut zu bewerten. Dies gilt auch für die Reorientierungszeit (vergleiche Köddermann (2008)). Bezüglich der Diffusion ist das resultierende Kraftfeld somit als zufriedenstellend und bezüglich der Reorientierungszeit als optimal zu bewerten, zumal im Falle von letzterer eine Verkleinerung des MAPE-Wertes von 28.6% auf 6.2% zu verzeichnen war. Allerdings ging dies zu stark auf Kosten der Dichte. Ein MAPE-Wert von 7.35% ist für diese Größe von Anwendern als unbefriedigend zu bewerten. In Summe kann man daher aus Anwendersicht daher mit diesem Ergebnis nicht zufrieden sein, obwohl die Fehlerfunktion wieder stark gefallen ist im Vergleich zu den Startparametern.

Wahl anderer Kraftfeldparameter Um eine Verschlechterung der Dichte im Laufe der Optimierung zu vermeiden, wurden in einem zweiten Optimierungsprozeß andere Kraftfeldparameter betrachtet und diejenigen, die einen starken Einfluß auf die Dichte haben, fixiert. Dabei handelte es sich um die Längenparameter in der Kette des Kations, also $\sigma(CH_3(a))$ und $\sigma(CH_2)$. Stattdessen wurden die LJ-Wechselwirkungszentren innerhalb des Rings in die Optimierung

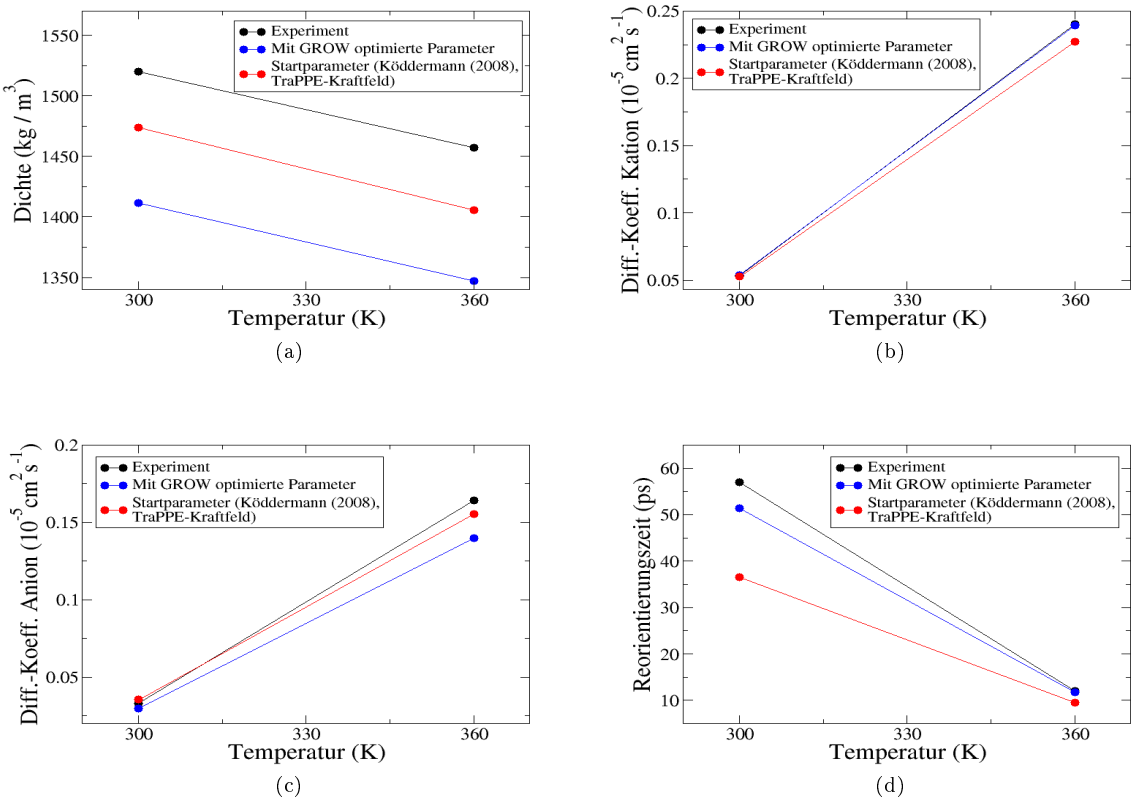


Abbildung 5.46: Optimierung von ρ (a), D (Kation) (b), D (Anion) und τ (Kation) (d) im Falle der Ionischen Flüssigkeit [C_2MIM][NTf_2]. Während bei der Diffusion insgesamt kaum Änderungen feststellbar waren, wurde die Reorientierungszeit auf Kosten der Dichte drastisch verbessert.

miteinbezogen, da deren Längenparameter erfahrungsgemäß einen nur geringen Einfluß auf die Dichte haben (vergleiche Köddermann (2008)). Auch die Wasserstoffatome, die an ein C-Atom des Rings gebunden sind, wurden neu parametrisiert. Es wurden wieder dieselben Simulations- und Optimierungseinstellungen wie in Tabelle 5.24 verwendet. Es handelte sich hierbei jedoch um ein achtdimensionales Optimierungsproblem. Die zu optimierenden Kraftfeldparameter waren σ(C1), σ(N), σ(C2), ε(C1), ε(N), ε(C2), ε(H2) und ε(H1).

Anhand der Ergebnisse für die in den Tabellen 5.25 und 5.26 Startparameter ist zunächst zu erkennen, daß im Falle der Diffusion ein MAPE-Wert von etwa 12% und etwa 14% keinen Unterschied macht, da es sich um dieselbe Simulation handelte, welche zweimal durchgeführt wurde. Die Reorientierungszeit hingegen ist deutlich weniger verrauscht. Weiterhin zeigen Tabelle 5.26 und Abbildung 5.47 die Optimierungsergebnisse im Falle der neuen Kraftfeldparameter. Die Längenparameter wurden zwar mehr verändert als die Energieparameter, allerdings nicht so deutlich wie in Tabelle 5.25. Am meisten verändert wurde σ(N). Dieser Parameter wurde um 16.44% verkleinert. Auch σ(C2) wurde verkleinert, während σ(C1) vergrößert wurde. Im Falle der Energieparameter war folgendes festzustellen: ε(N), ε(C1) sowie ε(H1) wurden vergrößert und ε(C2) und ε(H2) verkleinert.

k	$x^{(k)}$	MAPE ρ	MAPE D (Kation)	MAPE D (Anion)	MAPE τ	$F\left(x^{(k)}\right)$	$\nabla F\left(x^{(k)}\right)$	$\|\|\nabla F\left(x^{(k)}\right)\|\|$
				Steilster Abstieg, $h = 0.08$, $\Omega = (40, 40)$				
0	0.30175 0.32500 0.21300 0.20502 0.71128 0.43933 0.08787 0.18828	3.28%	14.8%	3.9%	28.7%	0.2273	0.00284 -4.05282 1.71706 -0.70669 -0.48175 -0.14796 -0.55461 0.16713	4.52
1	0.30174 0.34386 0.20501 0.20831 0.71352 0.44002 0.09045 0.18750	3.20%	6.0%	8.2%	26.5%	0.1717	0.56467 -1.48619 0.71411 -0.41254 -0.30051 0.06964 -0.17833 0.02866	1.83
2	0.29597 0.35904 0.19772 0.21252 0.71659 0.43931 0.09227 0.18721	3.16%	2.7%	8.9%	22.9%	0.1335	0.91691 -0.57088 0.60322 -0.11192 -0.25113 0.17082 -0.08868 0.14973	1.29
3	0.28007 0.36894 0.18726 0.21446 0.72094 0.43635 0.09381 0.18461	3.10%	3.7%	9.0%	20.0%	0.1126	0.94235 -0.67057 0.27529 -0.10459 -0.31373 0.10669 -0.13085 0.25571	1.27
4	0.26684 0.37835 0.18340 0.21593 0.72534 0.43485 0.09565 0.18102	3.12%	4.1%	10.2%	17.4%	0.0926	1.46752 -0.12292 0.36738 0.01975 -0.14391 0.14944 0.00306 0.06915	1.53
5	0.25915 0.37899 0.18147 0.21583 0.72609 0.43407 0.09563 0.18066	3.12%	1.3%	11.9%	14.8%	0.0816	1.16002 0.32787 0.15378 0.00737 0.16301 0.05356 -0.08861 0.19398	1.25

Tabelle 5.26: Optimierungsergebnisse für die Ionische Flüssigkeit [C_2MIM][NTf_2] für andere zu optimierende Kraftfeldparameter. Es handelt sich dabei um diejenigen, die nur geringen Einfluß auf die Dichte haben. Die zu optimierenden Zielgrößen waren ρ, D (für Kation und Anion) und τ (für das Kation). Da für Diffusion und Reorientierungszeit höhere statistische Unsicherheiten angenommen werden müssen als für die Dichte, sind die entsprechenden MAPE-Werte nur bis zur ersten Nachkommastelle angegeben. Innerhalb von fünf Iterationen mit dem Verfahren des steilsten Abstiegs wurde die Fehlerfunktion um etwa den Faktor 3 verkleinert. Die Reorientierungszeiten wurden nicht so deutlich verbessert wie bei der ersten Optimierung (Tabelle 5.25), die Dichten hingegen blieben nahezu konstant. Bezüglich der Diffusionskoeffizienten verlief die Optimierung wie zuvor. Die zu optimierenden Kraftfeldparameter waren σ(C1), σ(N), σ(C2), ε(C1), ε(N), ε(C2), ε(H2) und ε(H1), das heißt, es gilt $x^{(k)} = \left(\sigma(\mathrm{C1})^{(k)}, \sigma(\mathrm{N})^{(k)}, \sigma(\mathrm{C2})^{(k)}, \varepsilon(\mathrm{C1})^{(k)}, \varepsilon(\mathrm{N})^{(k)}, \varepsilon(\mathrm{C2})^{(k)}, \varepsilon(\mathrm{H2})^{(k)}, \varepsilon(\mathrm{H1})^{(k)}\right)$.

Innerhalb von fünf Iterationen mit der Methode des steilsten Abstiegs konnte die Fehlerfunktion um etwa den Faktor 3 verkleinert werden. Andere Verfahren und Einstellungen konnten wie oben keine Verbesserungen mehr erzielen. Die Reorientierungszeit wurde wieder stark verbessert. Dazu mußten also nicht die Längenparameter in der Kette des Kations vergrößert, sondern die im Ring (bis auf σ(C1)) verkleinert werden, denn die Dichten wurden wie erwartet nicht wesentlich verändert, allerdings auch nicht verbessert. Bei der Diffusion waren im Vergleich zur ersten Optimierung keine signifikanten Unterschiede feststellbar.
Für eine Gradientenberechnung waren acht Simulationen durchzuführen. Bei den ersten vier Iterationen wurde jeweils ein Armijo-Schritt benötigt. Insgesamt waren für die Optimierung somit 63 Simulationen (effektiv 54) erforderlich.
Das resultierende Kraftfeld ist zwar immer noch nicht optimal, aber als deutlich besser zu bewerten als im Falle der ersten Optimierung. Der MAPE-Wert in Bezug auf die Reorientierungszeit wurde zwar nur um etwa den Faktor 2 und nicht um etwa den Faktor 4.5 verkleinert, allerdings ist ein MAPE-Wert von etwa 14% als zufriedenstellend anzusehen. Was jedoch viel

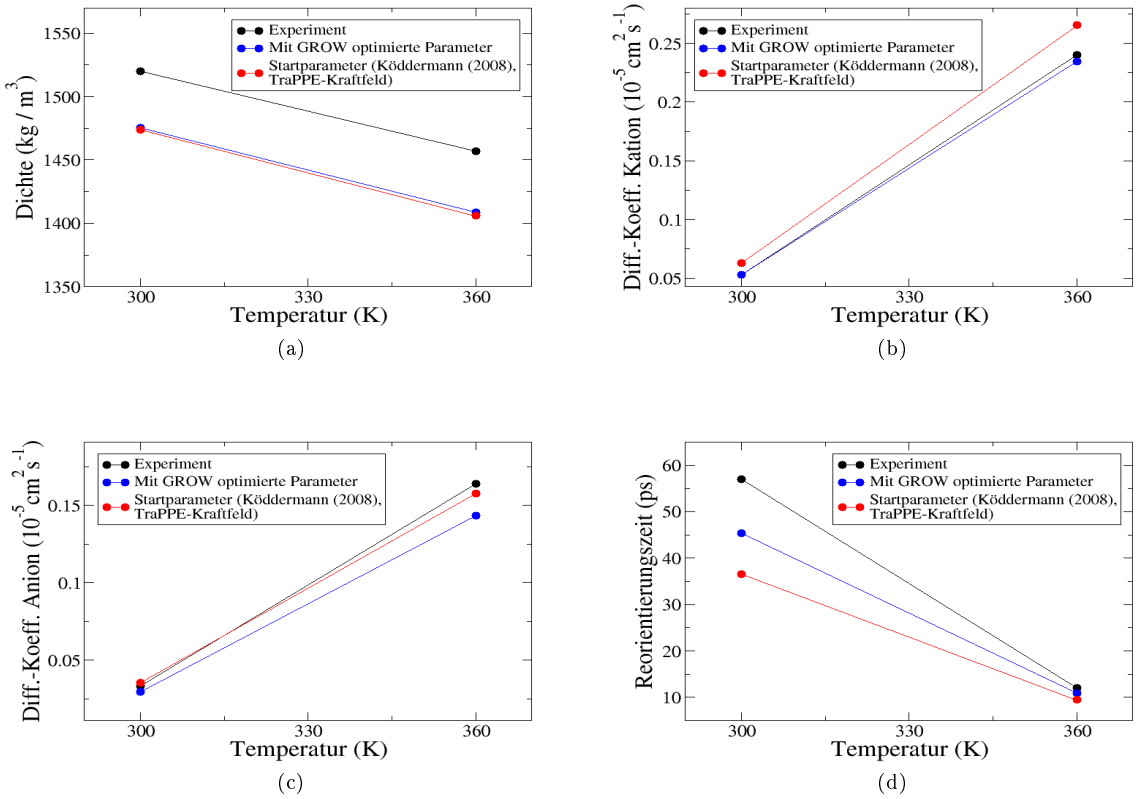


Abbildung 5.47: Optimierung von ρ (a), D (Kation) (b), D (Anion) und τ (Kation) (d) im Falle der Ionischen Flüssigkeit [C_2MIM][NTf_2] (andere Kraftfeldparameter). Während bei der Diffusion wieder insgesamt kaum Änderungen feststellbar waren, wurde die Reorientierungszeit erneut deutlich verbessert, allerdings nicht so drastisch wie im ersten Fall. Die Dichte wurde hier jedoch nicht verschlechtert, so daß das durch diese Optimierung erhaltene Kraftfeld in Summe als besser zu bewerten ist.

wichtiger ist, ist die Tatsache, daß der MAPE-Wert in Bezug auf die Dichte bei etwa 3% blieb und nicht auf mehr als 7% anstieg.

Fazit Die Optimierungsaufgaben für eine Ionische Flüssigkeit waren die in dieser Arbeit am schwierigsten zu lösenden. Dies ist nicht nur auf die rechenaufwendigen und komplexen Simulationen zurückzuführen, sondern auch auf die schwer zu reproduzierenden Zielgrößen. Die beiden Optimierungsprozesse zeigen auf, daß die Wahl der zu optimierenden Kraftfeldparameter ebenfalls äußerst ausschlaggebend ist, vor allem um zu vermeiden, daß Verbesserungen von Transporteigenschaften auf Kosten der Dichte gehen.
Zum Erhalt besserer Dichten könnten die Längenparameter der Kette wieder hinzugezogen werden. Es ist dann zwar zu erwarten, daß die MAPE-Werte in Bezug auf die Dichte zunächst wieder ansteigen, allerdings kann sich dieser Trend im Laufe der Optimierung aufgrund der zusätzlichen Präsenz der Längenparameter des Rings wieder umkehren. Dann wäre das Optimierungsproblem allerdings zehndimensional. Weitere Möglichkeiten bestehen in der Verwendung einer anderen Gewichtung der Zielgrößen, so daß mehr Wert auf die Optimierung der Dichte

Zielgröße	Einheit	T	Startparameter	Optimierte Parameter	Experiment	Quelle
ϵ	$\frac{\mathrm{As}}{\mathrm{Vm}}$	305	11.03	23.27	31.40	Franck u. Deul (1978)
η	mPas	293	0.155	0.430	0.544	Methanex Corporation (2006)
C_P	$\frac{\mathrm{J}}{\mathrm{mol\ K}}$	300	71.36	83.88	81.04	Carlson u. Westrum Jr (1971)

Tabelle 5.27: Anwendung des in dieser Arbeit erzielten Kraftfeldes für Methanol auf Dielektrizitätskonstante ϵ, Scherviskosität η und isobare Wärmekapazität C_P zu den jeweils angegebenen Temperaturen T in K. Startparameter und optimierte Parameter sind aus Tabelle 5.10 zu entnehmen. Details zum Optimierungsverlauf sind in Abschnitt 5.1.4 beschrieben. Alle drei Eigenschaften konnten durch die Optimierung verbessert werden.

im Gegensatz zu allen anderen Eigenschaften gelegt wird, und in der Veränderung des molekularen Modells. Aufgrund des hohen Rechenaufwands einer einzelnen IL-Simulation würde ein zusätzlicher Optimierungsprozeß jedoch den zeitlichen Rahmen dieser Dissertation sprengen.

5.5 Evaluation der hier erzielten Kraftfelder: Anwendung auf andere physikalische Eigenschaften

Als eine ihrer wichtigsten Eigenschaften sollen Molekulare Simulationen prädiktiv sein, das heißt, sie sollen dazu in der Lage sein, auf mikroskopischer Grundlage eine Vielzahl an makroskopischen Stoffdaten vorhersagen zu können. Es entspricht der intuitiven Erwartung eines Physikers beziehungsweise Chemikers, daß alle Stoffeigenschaften in irgendeiner Form voneinander abhängen. Falls somit im Laufe eines Optimierungsprozesses die Abweichung gewisser Systemeigenschaften vom Experiment sinkt, sollten die Abweichungen anderer Eigenschaften vom Experiment ebenfalls sinken. Nur dann kann von einem prädiktiven Kraftfeld gesprochen werden. Aus *physikalischer Sicht* muß hierzu ein geeignetes molekulares Modell verfügbar sein. Im vorliegenden Fall bedeutet dies, daß das Kraftfeld alle nötigen Terme, also die richtige funktionale Form, beinhaltet, um die Chemie des betrachteten Systems korrekt beschreiben zu können. Allein dies ist unter Umständen nicht einfach zu realisieren. Aus *mathematischer Sicht* sind jedoch weitere gewisse Grenzen gesetzt: Zum einen sind nicht zwischen allen physikalischen Zielgrößen analytische Abhängigkeiten bekannt, und somit ist die Existenz von Zusammenhängen zwischen allen Eigenschaften mathematisch nicht belegt. Zum anderen ist vor allem im Bereich des Maschinellen Lernens bekannt, daß die Trainingsmenge genügend groß sein muß, um akkurate Vorhersagen für unbekannte Daten zu treffen. In diesem Fall bedeutet dies, daß zum Erhalt eines prädiktiven Kraftfelds eine Vielzahl an Zielgrößen in die Optimierung miteinbezogen werden muß. Es kann beispielsweise von optimalen VLE-Kraftfeldern nicht erwartet werden, daß sie auch Transporteigenschaften korrekt reproduzieren. Dies ist in dieser Dissertation am Beispiel von Benzol (siehe Abschnitt 5.1.2) belegt worden, was hierbei allerdings auch auf ein schlecht geeignetes molekulares Modell zurückgeführt werden kann.

Es sollen dennoch im folgenden im Rahmen dieser Dissertation erzielte VLE-Kraftfelder validiert werden. Hierzu werden allerdings nur solche Optimierungsprozesse betrachtet, die auf einem gut geeigneten molekularen Modell und der Miteinbeziehung sämtlicher relevanter Kraftfeldparameter basierten. Es handelte sich dabei um Methanol (siehe Abschnitt 5.1.4) und Epoxid (siehe Abschnitt 5.3.3), wobei im ersten Fall die Partialladungen mitoptimiert wurden.

Zielgröße	Einheit	Startparameter	Optimierte Parameter	Experiment
κ_T	$\frac{1}{10^6 \mathrm{kPa}}$	2.32	3.58	2.60
λ_T	$\frac{\mathrm{W}}{\mathrm{m\ K}}$	0.12	0.13	0.12
η	mPas	0.237	0.226	0.151
C_P^{res}	$\frac{\mathrm{J}}{\mathrm{mol\ K}}$	39.75	46.26	41.69

Tabelle 5.28: Anwendung des in dieser Arbeit erzielten Kraftfeldes für Epoxid auf isotherme Kompressibilität κ_T, Wärmeleitfähigkeit λ_T, Scherviskosität η und isobare Wärmekapazität C_P. Startparameter und optimierte Parameter sind aus Tabelle 5.20 zu entnehmen. Details zum Optimierungsverlauf sind in Abschnitt 5.3.3 beschrieben. Die hier betrachtete Temperatur war $T = 375$ K. Alle experimentellen Daten stammen aus Case u. a. (2008), außer im Falle der Größe C_P^{res}, zu deren Berechnung der Idealanteil aus Hradetzky u. Lempe (2011) entnommen wurde. Nur die Scherviskosität konnte im Laufe der Optimierung verbessert werden. Alle anderen Eigenschaften wurden verschlechtert, wobei die Kraftfeldparameter auf die Wärmeleitfähigkeit einen nur sehr geringen Einfluß zu haben scheinen.

Tabelle 5.27 zeigt die Ergebnisse für Methanol: Berechnet wurden die Dielektrizitätskonstante ϵ gemäß Gleichung (C.24), die Scherviskosität gemäß Gleichung (C.20) und die isobare Wärmekapazität gemäß Gleichung (C.11). In diesem Fall konnten durch die Optimierung von Verdampfungsenthalpie und Siededichte auch diese drei Eigenschaften verbessert werden.
Tabelle 5.28 zeigt die Ergebnisse für Epoxid: Berechnet wurden in diesem Fall die isotherme Kompressibilität κ_T gemäß Gleichung (C.14), die Wärmeleitfähigkeit λ_T gemäß Gleichung (C.21), die Scherviskosität gemäß Gleichung (C.20) und der Residualanteil der isobaren Wärmekapazität. Hierbei konnte das hier erzielte VLE-Kraftfeld nur für die Scherviskosität eine Verbesserung liefern, wobei die Wärmeleitfähigkeit im betrachteten Parameterbereich keine sensitive Größe zu sein scheint. Insgesamt, das heißt angewandt auf die in Case u. a. (2008) angegebenen Eigenschaften, konnte das Kraftfeld von Eckl u. a. (2008a) bessere Ergebnisse erzielen. Bei der Wärmekapazität ist hier allerdings zu beachten, daß der experimentelle Wert nicht verläßlich ist, da er zwei unterschiedlichen Quellen entnommen wurde: Die isobare Wärmekapazität setzt sich additiv aus dem in der Tabelle angegebenen Residualanteil C_P^{res} und einem Idealanteil C_P^{id} zusammen. Ersterer ergibt sich aus den Wechselwirkungen eines Moleküls mit anderen Molekülen, und letzterer bezieht sich auf ein Einzelmolekül. Der Idealanteil berücksichtigt Translation und Rotation, der Residualanteil durch Interaktionen entstehende Schwingungen. Mit *ms2* konnte nur C_P^{res} berechnet werden. Daher mußte der Idealanteil von dem in Case u. a. (2008) angegebenen Wert für C_P abgezogen werden. Dazu war jedoch eine andere Quelle notwendig, und zwar die Merseburger Datenbank (Hradetzky u. Lempe, 2011).

Bei den meisten hier berechneten Zielgrößen handelt es sich um Transporteigenschaften. Es kann nicht erwartet werden, daß ein Kraftfeld, welches auf statische Größen wie Dichte und Druck sowie energetische Größen wie Verdampfungsenthalpie angepaßt ist, auch zwangsläufig andersartige Eigenschaften vorhersagen kann. Insbesondere in den Abschnitten 4.4, 4.5, 5.2 und 5.3 wurde festgestellt, daß unter Umständen eine Vielzahl lokaler und globaler Optima der zu minimierenden Zielfunktion existieren kann, möglicherweise sogar unendlich viele, die sich am Boden einer Regenrinne befinden. Der Erhalt eines bestimmten lokalen oder globalen Minimums ist für einen lokalen Optimierungsalgorithmus sehr stark von der Wahl der Startparameter abhängig. Unterschiedliche Startparameter können sich in den Einzugsbereichen unterschiedlicher Minima

befinden. Wird ein Minimum mithilfe eines Optimierungsverfahrens näherungsweise bestimmt, so ist es also nicht garantiert, daß die resultierenden Kraftfeldparameter an diesem Minimum auch optimal in Bezug auf andere, nicht in das Optimierungsproblem miteinbezogene Zielgrößen sind. Es gibt somit Optima unterschiedlicher Qualität, und im Falle von Methanol ist ein Optimum von guter und im Falle von Epoxid eines von eher schlechter Qualität gefunden worden. In Abschnitt 5.2.1 ist belegt worden, daß es auch globale Optima unterschiedlicher Qualität geben kann. Die einzige Möglichkeit, anderweitiges physikalisches beziehungsweise chemisches Wissen miteinzubeziehen besteht in einer geeigneteren Wahl des zulässigen Gebiets, welches in dieser Arbeit tendentiell sehr großzügig gewählt wurde, da es sich im Kern um eine Arbeit zur Methodenentwicklung und -bewertung handelt. Dann können andererseits wieder numerische Probleme für die Schrittweitensteuerung auftreten, was mehrfach diskutiert wurde. Aus den im vorliegenden Kapitel durchgeführten Untersuchungen und Analysen ergibt sich diesbezüglich das folgende Fazit: Die hier eingesetzten globalen und lokalen Optimierungsverfahren sind dazu in der Lage, schnell und robust die im jeweiligen Optimierungsproblem betrachteten Zielgrößen an ihre experimentellen Referenzdaten anzupassen. Zum Erhalt von Kraftfeldern von guter allgemeiner Qualität sind stets ein geeignetes molekulares Modell, eine geeignete Art und Anzahl an zu optimierenden Parametern sowie zu optimierenden Zielgrößen notwendig. Insbesondere sollten statische, energetische und dynamische Eigenschaften zu verschiedenen Temperaturen simultan angepaßt werden. Die Betrachtung einer Vielzahl an physikalischen Eigenschaften erfordert allerdings auch eine Erhöhung des Rechenaufwands für die Optimierung und den Zugang zu sämtlichen notwendigen experimentellen Daten. Daher hätte die sowohl physikalisch als auch mathematisch angemessene Realisierung eines prädiktiven Kraftfelds sehr schwierig. Dies hätte folglich den Rahmen dieser Dissertation gesprengt. Im folgenden Kapitel wird jedoch ein neuartiges, ableitungsfreies Verfahren entwickelt, welches die Effizienz eines lokalen Optimierungsprozesses nochmals deutlich erhöhen wird.

6 DGAFO – Ein neuartiges, ableitungsfreies Verfahren zur Parameteroptimierung, basierend auf Dünnen Gittern

Einige der in Abschnitt 3.4 beschriebenen gradientenbasierten Verfahren haben sich in dieser Arbeit und in Hülsmann u. a. (2010b) als für die vorliegende Problemstellung geeignet herausgestellt. Dabei handelt es sich aufgrund ihrer Konvergenzeigenschaften um effiziente und schnelle Optimierungsverfahren, vor allem im Vergleich zum Simplex-Algorithmus nach Nelder und Mead (siehe Abschitt 3.2). Allerdings weisen auch die gradientenbasierten Verfahren die folgenden Nachteile auf:

- In jedem Schritt ist ein Gradient zu berechnen und eine Schrittweitensteuerung durchzuführen. Manche Verfahren benötigen zusätzlich eine Hesse-Matrix, was von quadratischer Komplexität in der Anzahl an zu optimierenden Parameter ist. Um dies zu umgehen, können die in den Abschnitten 3.6.2 und 3.6.3 vorgestellten Verfahren zur Effizienzerhöhung eingesetzt werden. Diese sind allerdings nicht in der Nähe des Minimums anwendbar, und die Parametrisierung dieser Verfahren ist stets problemabhängig.
- Aufgrund des statistischen Rauschens ist die Genauigkeit eines Gradienten grundsätzlich begrenzt. Er kann und soll lediglich den Trend der zu minimierenden Fehlerfunktion repräsentieren. Dies wird jedoch, wie bereits motiviert, gerade in der Nähe des Minimums problematisch. Wie in Abschnitt 3.5.5 hergeleitet, ist es theoretisch durch geeignete Wahl von h möglich, diesem Problem gerecht zu werden, allerdings ist dies in der Praxis so gut wie unmöglich. Selbst wenn durch einen korrekten Gradienten, welcher in die richtige Richtung zeigt, noch Verbesserungen möglich sind, wird die Armijo-Schrittweitensteuerung trotz eines hohen Rechenaufwands nur noch zu unsignifikant kleineren Fehlerfunktionswerten führen, so daß sich der damit verbundene Aufwand nicht lohnt. Ein weiterer Nachteil von Abstiegsverfahren ist die Tatsache, daß bei einer falschen Abstiegsrichtung keine Schrittweitensteuerung Verbesserungen liefern kann. Bei Trust-Region-Verfahren hingegen kann die Schrittweite verkleinert werden, und es kann eine neue, geeignete Abstiegsrichtung gefunden werden.
- Die hier verwendeten gradientenbasierten Verfahren sind lokale Optimierer, das heißt, sie sind extrem startwertabhängig. Es ist daher notwendig, die Verfahren bereits mit einem guten Kraftfeld zu beginnen. In der Praxis liegen jedoch geeignete Kraftfelder nicht immer vor, so daß eine adäquate Startwertbestimmung vorgeschaltet werden muß. Eine Kombination der Verfahren, abhängig vom Abstand der aktuellen Iteration zum Minimum, ist stets zu empfehlen.

Diese Nachteile gradientenbasierter Verfahren motivieren die Suche nach einem Verfahren, welches mit deutlich weniger Simulationen auskommt und näher an ein lokales Minimum gelangen

kann. Das Verfahren sollte unabhängig von der Zerklüftung der Fehlerfunktion und vom statistischen Rauschen zum Ziel führen und möglichst robust sein. Es sollte nicht die Glattheit von F als Voraussetzung haben.
In diesem Kapitel wird ein neuartiger, ableitungsfreier Algorithmus vorgestellt, welcher auf Dünnen Gittern basiert und die Fehlerfunktion lokal glättet. Die Interpolation von Dünnen Gittern auf volle Gitter ermöglicht dabei eine deutliche Verringerung der Anzahl notwendiger Funktionsauswertungen, sprich Simulationen, ohne auf der anderen Seite eine deutliche Erhöhung des Interpolationsfehlers in Kauf nehmen zu müssen. Es wird sich dabei jedoch als sinnvoll erweisen, die Fehlerfunktion in jedem Iterationsschritt lokal mithilfe eines geeigneten Glättungsverfahrens zu approximieren. Die Güte der so erhaltenen Interpolation wird analog zu den Trust-Region-Verfahren (siehe Abschnitt 3.4.3) gemessen. Das aktuelle Minimum, welches diskret auf einem vollen Gitter auf dem vorher definierten Vertrauensgebiet bestimmt wird, wird dann entweder als neue Iteration akzeptiert, oder das Vertrauensgebiet wird verkleinert.
Von einer kontinuierlichen Minimierung innerhalb eines Vertrauensgebietes wird hier abgesehen, da ansonsten zusätzlich interne Optimierungsiterationen vonnöten wären. Die Fehlerfunktion selbst oder aber die geglättete Funktion müßten dabei zu oft ausgewertet werden. Es gibt auch Ansätze, die zu minimierende Funktion stückweise linear zu interpolieren: In Mangasarian u. a. (2004) wird ein beliebiges Funktional im $\mathbb{R}^N$ global stückweise-linear und konvex interpoliert und das globale Minimum mithilfe eines Liniensuchverfahrens, basierend auf einem Subgradienten, iterativ bestimmt. Allein für die Interpolation sind bereits mindestens 3^N Funktionsevaluationen notwendig. Hinzu kommt der Rechenaufwand für die Minimierung. Im vorliegenden Fall müßten aufgrund der Minimierung auf einem kompakten Vertrauensgebiet zusätzlich Nebenbedingungen in Betracht gezogen werden. Daher erscheint der Dünn-Gitter-Ansatz hier wesentlich sinnvoller.

In Abschnitt 6.1 werden zunächst Dünne Gitter eingeführt und die Vorteile in Bezug auf Effizienzerhöhung dargestellt. Geeignete Glättungs- beziehungsweise Approximationsmethoden werden in Abschnitt 6.2 vorgestellt. Das neuartige **D***ünn-***G***itter-basierte* **a***bleitungs***f***reie* **O***ptimierungsverfahren (DGAFO-Verfahren)* wird dann als Kombination von Interpolation auf Dünnen Gittern, Glättung und dem Trust-Region-Verfahren in Abschnitt 6.3 detailliert beschrieben. Die Konvergenz dieses Verfahrens behandelt schließlich Abschnitt 6.4.

6.1 Interpolation auf Dünnen Gittern

Die Interpolation auf Dünnen Gittern ist ein äußerst effizientes Diskretisierungs- und Interpolationsverfahren im Bereich der Finite-Elemente-Methodik. Dünne Gitter besitzen im Gegensatz zu vollbesetzten Produktgittern deutlich weniger Gitterpunkte, was sich vor allem im höherdimensionalen Raum bemerkbar macht. Beträgt die Anzahl an Freiheitsgraden bei Produktgittern $\mathcal{O}(h^{-N})$, so reduziert sich der Rechenaufwand bei Dünnen Gittern auf $\mathcal{O}(h^{-1} \cdot (\log h^{-1})^{N-1})$, wobei h die Maschenweite des Produktgitters ist. Im Gegensatz dazu erhöht sich der Interpolationsfehler in der L_∞-Norm lediglich von $\mathcal{O}(h^2)$ auf $\mathcal{O}(h^2 \cdot (\log h^{-1})^{N-1})$. Die ersten Interpolationsformeln auf Dünnen Gittern für die numerische Quadratur hochdimensionaler Funktionen wurden bereits von Smolyak (1963) angegeben. Später wurde von Zenger (1991) ein einfaches Diskretisierungs- und Interpolationsschema entwickelt, welches auf der hierarchischen Hutbasis

Level	$N = 1$	$N = 2$	$N = 3$	$N = 4$
1	$3 \to 3$	$9 \to 9$	$27 \to 27$	$81 \to 81$
2	$5 \to 5$	$25 \to 21$	$125 \to 81$	$625 \to 393$
3	$9 \to 9$	$81 \to 49$	$729 \to 225$	$6561 \to 1329$
4	$17 \to 17$	$289 \to 113$	$4913 \to 593$	$83521 \to 3921$
10	$1025 \to 1025$	$1.05 \cdot 10^6 \to 9217$	$1.07 \cdot 10^9 \to 47103$	$1.1 \cdot 10^{12} \to 1.78 \cdot 10^5$

Tabelle 6.1: Vergleich der Anzahl an Punkten auf vollen und Dünnen Gittern verschiedener Dimensionen und Level. Man sieht deutlich, daß gerade bei höheren Leveln und Dimensionen der Rechenaufwand auf Dünnen Gittern erheblich geringer ist als auf vollen Gittern.

und einem Tensorproduktansatz für höhere Dimensionen basiert. Zur Interpolation auf einem Dünnen Gitter werden dabei nicht alle, sondern nur bestimmte Basisfunktionen verwendet. Griebel u. a. (1990) setzten anstelle der hierarchischen Basis eine effiziente Kombinationsmethode ein, welche die entsprechende Funktion auf bestimmten regulären Teilgittern stückweise linear interpoliert. Die Kombination dieser Teilgitter ergibt ein Dünnes Gitter. Die Hauptanwendungsmethode der Dünnen Gitter ist ohne Frage die Lösung partieller Differentialgleichungen. Hierbei ist insbesondere die Arbeit von Bungartz (1992) zu nennen, wo ein adaptiver Löser der Poisson-Gleichung basierend auf hierarchischen Basen und Dünnen Gittern gefunden wurde.

In dieser Arbeit steht die Anwendung von Dünnen Gittern auf numerische Optimierung im Vordergrund. Es wird untersucht, wie die oben angesprochene effiziente Interpolationsmethode im Bereich der Kraftfeldparametrisierung verwendet werden kann. Der vorliegende Abschnitt behandelt zunächst die Interpolationsmethode selbst. Abschnitt 6.1.1 führt die Idee und die Grundvoraussetzungen der Dünn-Gitter-Methode ein. In Abschnitt 6.1.2 wird kurz die hierarchische Basis vorgestellt, und die im Rahmen dieser Arbeit im Vordergrund stehende Kombinationsmethode wird in Abschnitt 6.1.3 erläutert. Abschnitt 6.1.4 beschreibt schließlich die Umsetzung der multilinearen Interpolation, so daß das Minimum der Fehlerfunktion auf einem durch Interpolation erhaltenen Produktgitter bestimmt werden kann.

6.1.1 Idee und Definition der Dünnen Gitter

Die Hauptidee der Dünnen Gittern besteht darin, den Rechenaufwand weitestgehend zu reduzieren, ohne dabei eine nicht tolerierbare Erhöhung des Interpolationsfehlers zu erhalten. Gerade bei hohen Dimensionen macht sich die Reduktion des Rechenaufwandes bemerkbar: Tabelle 6.1 zeigt die Anzahl an Gitterpunkten von Produktgittern und Dünnen Gittern verschiedener Auflösung in Abhängigkeit von der Dimension.
Produktgitter und Dünne Gitter werden im folgenden mithilfe von Basisfunktionen für stückweise bilineare Funktionen eingeführt, das heißt, es wird zunächst der zweidimensionale Fall betrachtet:
Es sei $\Omega_{i,j}$ das äquidistante rechteckige Gitter auf dem Einheitsquadrat $\Omega := [0,1]^2$ mit Maschenweiten $h_i := 2^{-i}$ in x- und $h_j := 2^{-j}$ in y-Richtung. Der Vektor $\ell := (i,j) \in \mathbb{N}^2$ heißt *Level* des Gitters $\Omega_{i,j}$ mit 1-Norm $|\ell|_1 := i + j$. Weiterhin sei $S_{i,j}$ der Raum der stückweise bilinearen Funktionen auf dem Gitter $\Omega_{i,j}$. Der Einfachheit halber werden hier nur stückweise bilinea-

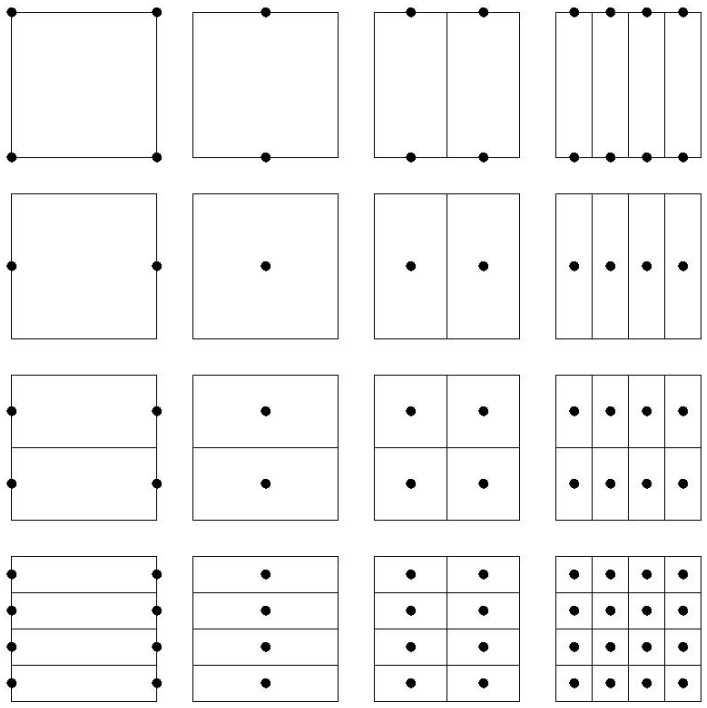


Abbildung 6.1: Dreiecksschema zur Kombination eines Dünnen Gitters vom Level 3 aus zweidimensionalen Teilgittern, die der Bedingung $|\ell|_1 \leq 3+2-1 = 4 \wedge \forall_{k \in \{1,2\}} \ell_k < 3$ genügen. Kombiniert man alle acht Teilgitter, so erhält man ein volles Gitter vom Level 3. Nimmt man nur die Gitter vom Level (0,0), (1,0), (2,0), (3,0), (0,1), (0,2), (0,3), (1,1), (2,1), (1,2), (2,2), (3,1) und (1,3), läßt man also das kleinere Dreieck bestehend aus drei Gittern unten rechts weg, so erhält man das entsprechende Dünne Gitter (siehe Abbildung 6.2 oben links).

re Funktionen betrachtet, welche auf $\Omega_{i,j}$ homogene Dirichlet-Randbedingungen erfüllen. Der entsprechende Raum sei mit $S^0_{i,j}$ bezeichnet, welcher sich als Tensorprodukt aus Unterräumen $T_{s,t},\ s = 1, ..., i,\ t = 1, ..., j$, ergibt, dessen Funktionen auf allen Gitterpunkten verschwinden, die den Räumen $S^0_{s-1,t}$ und $S^0_{s,t-1}$ entsprechen:

$$S^0_{i,j} = \bigotimes_{s=1}^{i} \bigotimes_{t=1}^{j} T_{s,t}. \tag{6.1}$$

Es werden eindeutig bestimmte stückweise bilineare Basisfunktionen in $T_{s,t}$ mit nichtüberlappenden rechteckigen Trägern der Größe $1/2^{s-1} \cdot 1/2^{t-1}$ eingeführt, die sogenannte *hierarchische Basis*, siehe Abschnitt 6.1.2. Jeder Gitterpunkt entspricht dabei einer spezifischen hierarchischen Basisfunktion. Er liegt dann im Zentrum des Trägers. Weiterhin gehört jeder Gitterpunkt zu einem Gitter eines bestimmten Levels ℓ. Die Kombination dieser Gitter ergibt das Produktgitter, und die direkten Summen der hierarchischen Basisfunktionen ergeben die Standardbasisfunktionen auf dem Produktgitter.
Jede Funktion $u \in S^0_{i,j}$ ist somit als Linearkombination aus hierarchischen Basisfunktionen darstellbar:

$$u = \sum_{s=1}^{i} \sum_{t=1}^{j} u_{s,t},\ u_{s,t} \in T_{s,t},\ s = 1, ..., i,\ t = 1, ..., j. \tag{6.2}$$

Die hierarchischen Basisfunktionen, die einem Gitterpunkt entsprechen, welcher verschiedenen hierarchischen Gittern angehört, sind gleich. Daher sind lediglich die Basiskoeffizienten zu addieren. In Zenger (1991) wurde die Ungleichung

$$\|u_{s,t}\|_\infty \leq 4^{-s-t-1}|u| \tag{6.3}$$

bewiesen, wobei die Seminorm $|u|$ durch

$$|u| := \left\| \frac{\partial^4 u}{\partial x^2 \partial y^2} \right\|_\infty \tag{6.4}$$

definiert ist. In den Beweis von Ungleichung (6.3) fließt ein, daß der Differenzenstern, der die Transformation der Koeffizienten vom Raum, der von den Standardbasisfunktionen aufgespannt wird, in den Raum, der von den hierarchischen Basisfunktionen aufgespannt wird, beschreibt, gleich dem Differenzenstern für die Finite-Differenzen-Approximation von $\frac{h_i^2 h_j^2}{4} \frac{\partial^4 u}{\partial x^2 \partial y^2}$ ist.

Um ein Dünnes Gitter zu erhalten, werden nur diejenigen Basisfunktionen gewählt, deren Koeffizienten größer oder gleich einem bestimmten Toleranzwert sind. In Zenger (1991) wird dieser Toleranzwert gleich $4^{-\hat{\ell}-1}|u|$ gewählt, wobei $\hat{\ell}$ das Level des Dünnen Gitters ist, welches nachfolgend definiert wird. Alle anderen Basisfunktionen werden vernachlässigt, und man erhält die folgende Definition eines Dünn-Gitter-Raumes von Funktionen:

Definition 6.1.1 (Dünn-Gitter-Raum von Funktionen)**.** *Der Raum $\hat{S}^0_{\hat{\ell}}$, der durch die Unterräume $T_{i,j}$ mit $i+j = \ell \leq \hat{\ell}+1$ aufgespannt wird, also*

$$\hat{S}^0_{\hat{\ell}} := \bigotimes_{s=1}^{\hat{\ell}} \bigotimes_{t=1}^{\hat{\ell}-s+1} T_{s,t} = \bigotimes_{s+t\leq\hat{\ell}+1} T_{s,t}, \tag{6.5}$$

heißt Dünn-Gitter-Raum von Funktionen*. Das aus der Kombination der Teilgitter, welche den Unterräumen $T_{s,t}$, $s+t \leq \hat{\ell}+1$, entsprechen, entstehende* Dünne Gitter *hat das Level $\hat{\ell}$.*

Die Bedingung $|\ell|_1 \leq \hat{\ell}+1$ führt zu einem Dreiecksschema von Unterräumen $T_{i,j}$, welches in Abbildung 6.1 veranschaulicht ist. Es ist zu beachten, daß $\ell_k = 0$, $k \in \{1,2\}$, grundsätzlich zugelassen ist, es muß jedoch $\forall_{k\in\{1,2\}}\ \ell_k < \hat{\ell}$ gelten. Der Übergang auf N-dimensionale Dünne Gitter vom Level $\hat{\ell}$ ist trivial: Man kombiniere alle Teilgitter vom Level ℓ mit $|\ell|_1 \leq \hat{\ell}+N-1 \ \wedge\ \forall_{k=1,\ldots,N}\ \ell_k < \hat{\ell}$. Dabei ist $|\ell|_1 = \sum_{i=1}^N \ell_i$.

Abbildung 6.2 zeigt einige Dünne Gitter der Level 3, 4 und 5 in 2D und 3D, welche mit einem Algorithmus erzeugt wurden, der Dünne Gitter eines vorgegebenen Levels aus den entsprechenden Teilgittern kombiniert.

Es läßt sich an dieser Stelle bereits sagen, daß eine Interpolation auf Dünnen Gittern im Vergleich zu Produktgittern ohne eine signifikante Erhöhung des Interpolationsfehlers durchführbar ist, da die Koeffizienten, die zu einem Produktgitter aber nicht zu einem Dünnen Gitter gehören, ab einem bestimmten Dünn-Gitter-Level keinen erheblichen Beitrag mehr zum Interpolanten

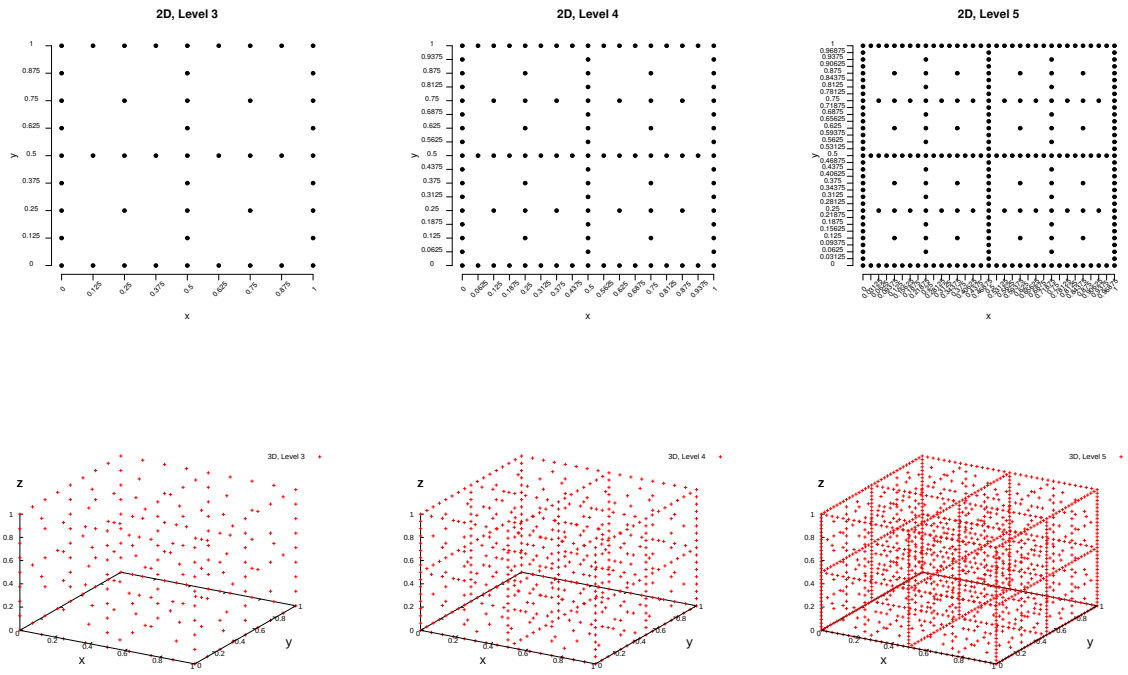


Abbildung 6.2: Dünne Gitter der Level 3, 4 und 5 in 2D und 3D.

leisten. Näheres zum Interpolationsfehler befindet sich in Abschnitt 6.4.1. Was sich im Rahmen dieser Arbeit jedoch zunächst als Nachteil erweist, ist die Tatsache, daß dies nur für Funktionen gilt, die genügend glatt sind. Aufgrund von Ungleichung (6.3) muß die zu interpolierende Funktion mindestens viermal stetig differenzierbar sein. Da dies bei der vorliegenden Problemstellung keinesfalls vorausgesetzt werden kann, wird zur Interpolation die Kombinationsmethode nach Griebel u. a. (1990) verwendet, welche in Abschnitt 6.1.3 beschrieben wird. Ein ableitungsfreies Optimierungsverfahren, welches auf Dünnen Gittern basiert, ist aufgrund des geringen Rechenaufwands für die vorliegende Problemstellung äußerst relevant. Problematisch sind allerdings die Randpunkte: Abbildung 6.2 zeigt deutlich, daß zweidimensionale Dünne Gitter am Rand voll besetzt sind und auch höherdimensionale Dünne Gitter sehr viele Randpunkte besitzen. Daher kann die Anzahl an Funktionsauswertungen für Punkte eines Dünnen Gitters erheblich reduziert werden, wenn homogene Dirichlet-Randbedingungen eingeführt werden und F dementsprechend transformiert wird, worauf in Abschnitt 6.3.3 näher eingegangen wird.

6.1.2 Hierarchische Basis

Es sei im folgenden mit $\ell = (\ell_1, ..., \ell_N)$ das Level eines Teilgitters bezeichnet. Das bedeutet, daß es in Richtung j für $j = 1, ..., N$ genau $2^{\ell_j} + 1$ Gitterpunkte gibt mit Maschenweite $h_j = 2^{-\ell_j}$, $j = 1, ..., N$. Dies ist in Abbildung 6.1 veranschaulicht: Das Gitter unten links hat das Level $(0, 3)$, weil es in y-Richtung $2^3 + 1 = 9$ Gitterpunkte und in x-Richtung nur einen Gitterpunkt gibt. Es ist zu beachten, daß in dieser Abbildung nur diejenigen Gitterpunkte dar-

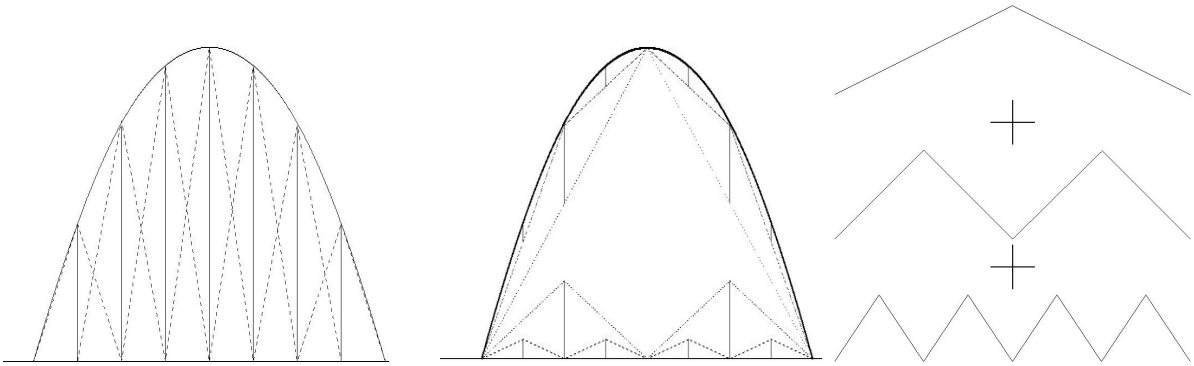


Abbildung 6.3: Nodale und hierarchische Basis zur Interpolation der Funktion $f(x) = 1 - x^2$ auf einem vollen Gitter vom Level $\bar{\ell} = 3$. Jedem inneren Gitterpunkt ist genau eine Basisfunktion zugeordnet. Bei der nodalen Basis ist die Größe des Trägers jeder Basisfunktion stets gleich $2 \cdot \frac{1}{8} = \frac{1}{4}$, bei der hierarchischen Basis ist sie gleich $2h_\ell$, also je kleiner, desto größer das Level ℓ des Teilgitters ist, zu dem der Gitterpunkt gehört. Die hierarchische Basis setzt sich aus den rechts dargestellten Basisfunktionen zusammen.

gestellt sind, die nicht bereits in einem Gitter mit betraglich kleinerem Level enthalten sind. Zu dem dargestellten Gitter gehören noch die Schnittpunkte der horizontalen und vertikalen Linien, daher sind es insgesamt neun. Weiterhin sei mit $\hat{\ell}$ das Level eines Dünnen Gitters bezeichnet. In diesem Abschnitt werden sowohl vollbesetzte als auch Dünne Gitter für den N-dimensionalen Fall analog zum 2D-Fall aus einer *hierarchischen* Sicht definiert.

Man geht zunächst von einem vollen Gitter mit Level $\bar{\ell} \in \mathbb{N}$ für $N = 1$ aus. Die Maschenweite beträgt dann $h_{\bar{\ell}} 2^{-\bar{\ell}}$. Die Gitterpunkte werden mit

$$x_{\bar{\ell},i} := i \cdot h_{\bar{\ell}},\ i = 0, ..., 2^{\bar{\ell}} + 1,$$

bezeichnet. Dabei heißt i der *Ort* von $x_{\bar{\ell},i}$. Besteht das Ziel darin, eine Funktion $u : [0,1]^N \to \mathbb{R}$ auf dem Gitter zu interpolieren, so entspricht jeder Gitterpunkt $x_{\ell,i}$ eines Teilgitters vom Level $\ell \leq \bar{\ell}$ einer stückweise linearen Basisfunktion $\phi_{\ell,i} \in T_\ell$. Diese Funktionen bilden die sogenannte *hierarchische Hutbasis*, welche aus der Translation und Stauchung der eindimensionalen Funktion $\max\{0, 1 - |x|\}$ entsteht. Die Basisfunktion $\phi_{\ell,i}$ nimmt an der Stelle $x_{\ell,i}$ den Wert 1 an und fällt zu den Nachbarn $x_{\ell,i\pm1}$ auf 0 ab. Ansonsten ist $\phi_{\ell,i}$ konstant gleich 0:

Definition 6.1.2 (Hierarchische Hutbasis)**.** *Die Basisfunktionen der Form*

$$\phi_{\ell,i} := \begin{cases} \frac{1}{h_\ell} \cdot x + 1 - \frac{x_{\ell,i}}{h_\ell}, & x \in [x_{\ell,i} - h_\ell, x_{\ell,i}) \\ -\frac{1}{h_\ell} \cdot x + 1 + \frac{x_{\ell,i}}{h_\ell}, & x \in [x_{\ell,i}, x_{\ell,i} + h_\ell] \\ 0, & \text{sonst}, \end{cases} \tag{6.6}$$

bilden für $\ell \leq \bar{\ell}$die sogenannte hierarchische Hutbasis *des Funktionenraums T_ℓ.*

Abbildung 6.3 zeigt den Unterschied zwischen der Interpolation mit sogenannten *nodalen* und hierarchischen Basisfunktionen. Bei letzteren handelt es sich um Hutfunktionen mit gleichgroßen Trägern. Falls h die Maschenweite des vollen Gitters ist, so ist die Größe der Träger

gleich $2h$. Die Funktionswerte der Punkte, die zwischen den Gitterpunkten liegen, werden durch die nodale Basis nicht ausreichend gut wiedergegeben, wie die linke Abbildung zeigt. Bei jeder Gitterverfeinerung ergibt sich eine neue Basis. Die mittlere Abbildung stellt die Interpolation mit einer hierarchischen Basis dar: Bei jeder Gitterverfeinerung kommen hier lediglich neue Basisfunktionen hinzu, deren Koeffizienten für $\ell \to \infty$ gegen 0 gehen. Da, wie bereits in Abschnitt 6.1.1 angesprochen wurde, die Interpolationsfehler durch die Basiskoeffizienten bestimmt werden, sind diese im Falle einer hierarchischen Basis bei gegebenem Gitterlevel deutlich kleiner als im Falle einer nodalen Basis.

Die Menge der zu einem Level ℓ gehörigen Indizes sei mit

$$V_\ell := \{(\ell, i) | i = 0, ..., 2^\ell + 1\}$$

bezeichnet. Falls es in V_ℓ Indizes (ℓ, i) gibt mit $x_{\ell,i} = x$, so sagt man, daß x durch V_ℓ beschrieben wird, und man schreibt $x \bar{\in} V_\ell$. Möchte man V_ℓ durch eine hierarchische Sichtweise beschreiben, so betrachtet man nur diejenigen Punkte $x_{\ell,i} \bar{\in} V_\ell$, die noch nicht durch $V_{\ell-1}$ beschrieben wurden. Diese sogenannten *hierarchischen Überschüsse* sind durch die Menge

$$W_\ell := \{(\ell, i) \in V_\ell | x_{\ell,i} \bar{\notin} V_{\ell-1}\}$$

gegeben. Wird weiterhin $W_1 := V_1$ gesetzt, so ist ein Gitter mit Level ℓ für $N = 1$ hierarchisch mittels der disjunkten Zerlegung

$$G_\ell^1 := \bigcup_{k=1}^{\ell} W_k$$

darstellbar.

Im N-dimensionalen sind die Indizes (ℓ, i) Vektoren der Länge $2N$. Die Definition der Indexmengen V_ℓ und W_ℓ, welche dem Funktionenraum T_ℓ entsprechen, lauten analog:

$$\begin{aligned} V_\ell &:= \{(\ell, i) |\ i := (i_1, ..., i_N), \forall_{j=1,...,N}\ i_j = 0, ..., 2^{\ell_j} + 1\} \\ W_\ell &:= \{(\ell, i) \in V_\ell |\ \forall_{j=1,...,N,\ \ell_j > 1}\ x_{\ell,i} \bar{\notin} V_{\ell - e_j}\} \\ &= \{(\ell, i) \in V_\ell |\ \forall_{j=1,...,N}\ i_j \text{ ungerade}\}. \end{aligned}$$

Dabei ist e_j der j-te Einheitsvektor. Sei $e := (1, ..., 1) \in \mathbb{R}^N$. Mit der Festlegung $W_e := V_e$ wird ein Gitter mit Level ℓ durch die disjunkte Zerlegung

$$G_\ell^N := \bigcup_{k_1=1}^{\ell_1} \cdots \bigcup_{k_N=1}^{\ell_N} W_{k_1,...,k_N}$$

definiert.
Ein volles Gitter mit Level $\bar{\ell}$ ist durch

$$G_{\bar{\ell}}^N := \bigcup_{1 \le |\ell|_\infty \le \bar{\ell}} W_\ell$$

hierarchisch darstellbar. Der zugehörige Funktionenraum $S^0_{\bar{\ell}}$ ergibt sich dann gemäß Gleichung (6.1) als Tensorprodukt aus den Unterräumen $T_\ell,\ 1 \leq |\ell|_\infty \leq \bar{\ell}$.
Ein Dünnes Gitter mit Level $\hat{\ell}$ ist durch

$$\hat{G}^N_{\hat{\ell}} := \bigcup_{|\ell|_1 \leq \hat{\ell}+N-1} W_\ell$$

hierarchisch darstellbar. Der zugehörige Funktionenraum $\hat{S}^0_{\hat{\ell}}$ ergibt sich dann gemäß Gleichung (6.5) als Tensorprodukt aus den Unterräumen $T_\ell,\ |\ell|_1 \leq \hat{\ell} + N - 1$.
Die verallgemeinerten Basisfunktionen $\phi_{\ell,i}$ ergeben sich als Produkt aus den eindimensionalen Basisfunktionen aus Definition 6.1.2:

$$\phi_{\ell,i}(x) = \prod_{j=1}^{N} \phi_{\ell_j,i_j}(x_j).$$

Die hierarchische Hutbasis auf dem Dünnen Gitter $G^N_{\hat{\ell}}$ ist gegeben durch

$$\Phi^N_{\hat{\ell}} := \{\phi_{\ell,i} | (\ell,i) \in G^N_{\hat{\ell}}\}.$$

Der Dünngitterinterpolant $I_{G^N_{\hat{\ell}}} u(x)$ einer Funktion u ist durch Linearkombination mit Koeffizienten $u_{\ell,i}$ berechenbar:

$$u(x) \approx I_{G^N_{\hat{\ell}}} u(x) = \sum_{(\ell,i) \in G^N_{\hat{\ell}}} u_{\ell,i} \phi_{\ell,i}(x). \tag{6.7}$$

6.1.3 Kombinationsmethode

Die auf Griebel u. a. (1990) zurückgehende *Kombinationsmethode* kombiniert effizient Probleme auf regulären Gittern vom Level ℓ mit verschiedenen Maschenweiten $\ell_1, ..., \ell_N$ zu einem Dünngitterproblem. Da sich diese Methode, wie in Griebel u. a. (1990) aufgeführt, in der Praxis bewährt hat und auch auf Funktionen anwendbar ist, die nicht notwendigerweise glatt sind, wird sie im Rahmen dieser Dissertation, das heißt innerhalb des DGAFO-Verfahrens, als effiziente Interpolationsmethode eingesetzt.

Zur Einführung der Kombinationsmethode wird zunächst der zweidimensionale Fall betrachtet: Es sei $u : \mathbb{R}^2 \to \mathbb{R}$ eine beliebige Funktion. Der Interpolant von u auf $\hat{S}^0_{\hat{\ell}}$, der in diesem Abschnitt mit $\hat{u}^I_{\hat{\ell}}$ bezeichnet wird, läßt sich aufgrund von Gleichung (6.5) als Linearkombination aus Funktionen $u_{s,t} \in T_{s,t}$ folgendermaßen darstellen:

$$\hat{u}^I_{\hat{\ell}} = \sum_{s=1}^{\hat{\ell}} \sum_{t=1}^{\hat{\ell}-s+1} u_{s,t},\ u_{s,t} \in T_{s,t}. \tag{6.8}$$

Es sei weiterhin $u^I_{i,j}$ der Interpolant von u auf $S^0_{i,j}$ mit $i+j \in \{\hat{\ell}, \hat{\ell}+1\}$. Der Raum $S^0_{i,j}$ bezieht sich gemäß den Definitionen aus Abschnitt 6.1.1 auf ein reguläres Gitter vom Level (i,j). Dann gilt der folgende Satz:

Satz 6.1.3 (Kombinationsmethode (2D)). *Für $u : \mathbb{R}^2 \to \mathbb{R}$ seien $\hat{u}^I_{\hat{\ell}}$ und $u^I_{i,j}$ die Interpolanten auf $\hat{S}^0_{\hat{\ell}}$ beziehungsweise $S^0_{i,j}$, wobei $i+j \in \{\hat{\ell}, \hat{\ell}+1\}$. Dann gilt die folgende Beziehung:*

$$\hat{u}^I_{\hat{\ell}} = \sum_{i+j=\hat{\ell}+1} u^I_{i,j} - \sum_{i+j=\hat{\ell}} u^I_{i,j}. \tag{6.9}$$

Beweis: Es gilt:

$$\begin{aligned}
\sum_{i+j=\hat{\ell}+1} u^I_{i,j} - \sum_{i+j=\hat{\ell}} u^I_{i,j} &\overset{(6.2)}{=} \sum_{i+j=\hat{\ell}+1} \sum_{s=1}^{i}\sum_{t=1}^{j} u_{s,t} - \sum_{i+j=\hat{\ell}} \sum_{s=1}^{i}\sum_{t=1}^{j} u_{s,t} \\
&= \sum_{k=1}^{\hat{\ell}} \sum_{s=1}^{k} \sum_{t=1}^{\hat{\ell}+1-k} u_{s,t} - \sum_{k=1}^{\hat{\ell}} \sum_{s=1}^{k} \sum_{t=1}^{\hat{\ell}-k} u_{s,t} \\
&= \sum_{k=1}^{\hat{\ell}} \sum_{s=1}^{k} u_{s,\hat{\ell}-k+1} = \sum_{k=1}^{\hat{\ell}} \sum_{s=1}^{\hat{\ell}-k+1} u_{k,s} \\
&\overset{(6.8)}{=} \hat{u}^I_{\hat{\ell}}.
\end{aligned}$$

□

Somit läßt sich jede Dünngitterfunktion aus ihren Interpolanten auf den regulären vollen Gittern $G^2_{i,j}$, $i+j \in \{\hat{\ell}, \hat{\ell}+1\}$, linear kombinieren.
Betrachtet man allgemein $u_{i,j} \in T_{i,j}$, so wird eine Kombinationsfunktion $u^c_{\hat{\ell}}$ auf einem Dünnen Gitter wie folgt definiert:

Definition 6.1.4 (Kombinationsfunktion (2D)). *Die* Kombinationsfunktion *von Lösungen $u_{i,j} \in T_{i,j}$ von Finite-Elemente-Problemen auf regulären 2D-Gittern vom Level (i,j), $i+j \in \{\hat{\ell}, \hat{\ell}+1\}$, ist gegeben durch*

$$u^c_{\hat{\ell}} := \sum_{i+j=\hat{\ell}+1} u_{i,j} - \sum_{i+j=\hat{\ell}} u_{i,j}. \tag{6.10}$$

Bei $u^c_{\hat{\ell}}$ handelt es sich zwar um eine Funktion auf dem Dünngitterraum $\hat{S}^0_{\hat{\ell}}$, allerdings ist sie im allgemeinen nicht gleich der Lösung $\hat{u}^I_{\hat{\ell}}$ des Finite-Elemente-Problems auf dem Dünnen Gitter. Dies ist bei der Abschätzung des Interpolationsfehlers $u - u^c_{\hat{\ell}}$ zu berücksichtigen (siehe Abschnitt 6.4.1). Es sei hier schon einmal erwähnt, daß der Fehler $u - u^c_{\hat{\ell}}$ in derselben Größenordnung liegt wie $u - u^I_{\hat{\ell}}$, nämlich in $\mathcal{O}\left(h^2_{\hat{\ell}} \log(h_{\hat{\ell}})^{-1}\right)$.

Analog zu Satz 6.1.3 und Definition 6.1.4 ergibt sich für den dreidimensionalen Fall die folgende Definition:

Definition 6.1.5 (Kombinationsfunktion (3D)). *Die* Kombinationsfunktion *von Lösungen $u_{i,j,k} \in T_{i,j,k}$ von Finite-Elemente-Problemen auf regulären 3D-Gittern vom Level (i,j,k), $i+j+k \in \{\hat{\ell}, \hat{\ell}+1, \hat{\ell}+2\}$, ist gegeben durch*

$$u_{\hat{\ell}}^c := \sum_{i+j+k=\hat{\ell}+2} u_{i,j,k} - 2 \sum_{i+j+k=\hat{\ell}+1} u_{i,j,k} + \sum_{i+j+k=\hat{\ell}} u_{i,j,k}. \tag{6.11}$$

Durch vollständige Induktion läßt sich Satz 6.1.3 direkt auf den N-dimensionalen Fall übertragen, und es ergibt sich die folgende Definition:

Definition 6.1.6 (Kombinationsfunktion (allgemein)). *Die* Kombinationsfunktion *von Lösungen $u_\ell \in T_\ell$ von Finite-Elemente-Problemen auf regulären N-dimensionalen Gittern vom Level ℓ mit $|\ell|_1 = \hat{\ell} + N - 1 - i,\ i = 0, ..., N-1$, ist gegeben durch*

$$u_{\hat{\ell}}^c := \sum_{i=0}^{N-1} (-1)^i \binom{N-1}{i} \sum_{|\ell|_1 = \hat{\ell}+N-1-i} u_\ell. \tag{6.12}$$

Bemerkung 6.1.7. *Aufgrund der Bedingung $|\ell|_1 = \hat{\ell} + N - 1 - i,\ i = 0, ..., N-1$, kann wie folgt auf ein vollbesetztes reguläres Gitter mit Maschenweite $1/2^{\hat{\ell}}$ und Level $\bar{\ell} = (\hat{\ell})$ multilinear interpoliert werden: Man berechnet sämtliche Funktionswerte auf einem Dünnen Gitter vom Level $(\hat{\ell}, ..., \hat{\ell}) \in \mathbb{N}^N$. Aufgrund der hierarchischen Struktur des Dünnen Gitters hat man dann automatisch alle Punkte der regulären Teilgitter vom Level ℓ mit $|\ell|_1 \leq \hat{\ell} + N - 1$, also sämtliche für die durch Gleichung (6.12) beschriebene Interpolation notwendige Gitterpunkte. Die multilineare Interpolation wird im nächsten Abschnitt näher erläutert.*

6.1.4 Multilineare Interpolation

Es sei $n \in \mathbb{N}$ so, daß $(n+1)^N$ die Anzahl an Gitterpunkten eines vollen Gitters $G_{\bar{\ell}}^N$ mit Level $\bar{\ell} = (\hat{\ell}, ..., \hat{\ell}) \in \mathbb{N}^N$ ist. Weiterhin sei $\ell = (\ell_1, ..., \ell_N) \in \mathbb{N}^N$ das Level eines Teilgitters mit $|\ell|_1 \leq \hat{\ell} + N - 1$. Nach Bemerkung 6.1.7 reichen alle Teilgitter aus, die kombiniert ein Dünnes Gitter vom Level $\hat{\ell} \in \mathbb{N}$ ergeben.
Es werden außerdem definiert: $h := \frac{1}{N},\ n_i := 2^{\ell_i}$ und $n_i^{\text{loc}} := \frac{n}{2^{\ell_i}},\ i = 1, ..., N$. Dann folgt für alle $i = 1, ..., N$:

$$\begin{aligned} n_i \cdot n_i^{\text{loc}} &= n \\ \ell_i + \log_2 n_i^{\text{loc}} &= \hat{\ell}. \end{aligned}$$

n_i^{loc} ist gleich der Anzahl an Intervallen auf dem vollen Gitter zwischen zwei benachbarten Gitterpunkten des Teilgitters in Richtung i.

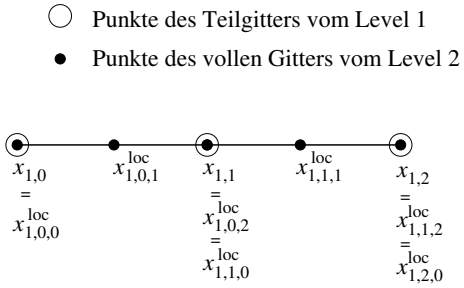


Abbildung 6.4: Multilineare Interpolation: Veranschaulichung von x_{1,j_1}, $j_1 = 0, ..., n_1$, und $x^{\text{loc}}_{1,j_1,\tilde{j}_1}$, $\tilde{j}_1 = 0, ..., n^{\text{loc}}_1$, auf einem 1D-Gitter ($i = 1$). Es gilt $n_1 = 2^{\ell_1} = 2^1 = 2$ und $n^{\text{loc}}_1 = \frac{n}{2^{\ell_1}} = \frac{4}{2^1} = 2$. Die Punkte x_{1,j_1} liegen auf einem Teilgitter vom Level 1 (3 Gitterpunkte) und die Punkte $x^{\text{loc}}_{1,j_1,\tilde{j}_1}$ auf einem vollen Gitter vom Level 2 (5 Gitterpunkte).

Um Redundanzen zu vermeiden, werden die folgenden Laufindizes eingeführt:

- $\tilde{j}_i = 0, ..., n^{\text{loc}}_i$,
- $j_i = 0, ..., n_i$,
- $i = 1, ..., N$.

Ein Punkt auf dem Teilgitter in Richtung i wird mit $x_{i,j_i} := j_i \cdot 2^{-\ell_i} = \frac{j_i}{n_i}$ und ein Punkt auf dem vollen Gitter in Richtung i wird mit $x^{\text{loc}}_{i,j_i,\tilde{j}_i} := x_{i,j_i} + \tilde{j}_i h$ bezeichnet. Abbildung 6.4 veranschaulicht x_{i,j_i} und $x^{\text{loc}}_{i,j_i,\tilde{j}_i}$ auf einem vollen 1D-Gitter mit fünf Gitterpunkten (Level 2) und einem Teilgitter vom Level 1.

Das Ziel besteht nun darin, für jeden Punkt $x^{\text{loc}}_{i,j_i,\tilde{j}_i}$ des vollen Gitters mittels multilinearer Interpolation einen Funktionswert $u_{\hat{\ell}}\left(x^{\text{loc}}_{i,j_i,\tilde{j}_i}\right)$ zu bestimmen (siehe Gleichung (6.12)). Hierzu sind sämtliche Funktionswerte von F für benachbarte Punkte notwendig, welche Teilgittern vom Level ℓ mit $|\ell|_1 \leq \hat{\ell} + N - 1$ angehören. Dafür sind die Gitterpunkte x_{i,j_i} des Teilgitters vom Level ℓ_i in Richtung i für alle i zu verwenden.

Dies geschieht mithilfe der folgenden Rekursion über die Dimensionen: Es sei zunächst $i = N$. Dann werden die interpolierten Funktionswerte $u^{(N)}_{\ell}\left(x_{1,j_1}, ..., x_{N-1,j_{N-1}}, x^{\text{loc}}_{N,j_N,\tilde{j}_N}\right)$ gemäß folgender Formel berechnet:

$$\begin{aligned} u^{(N)}_{\ell}\left(x_{1,j_1}, ..., x_{N-1,j_{N-1}}, x^{\text{loc}}_{N,j_N,\tilde{j}_N}\right) &= \frac{F\left(x_{1,j_1}, ..., x_{N-1,j_{N-1}}, x_{N,j_N+1}\right)\left(x^{\text{loc}}_{N,j_N,\tilde{j}_N} - x_{N,j_N}\right)}{x_{N,j_N+1} - x_{N,j_N}} \\ &+ \frac{F\left(x_{1,j_1}, ..., x_{N-1,j_{N-1}}, x_{N,j_N}\right)\left(x_{N,j_N+1} - x^{\text{loc}}_{N,j_N,\tilde{j}_N}\right)}{x_{N,j_N+1} - x_{N,j_N}}, \end{aligned}$$

für alle $j = 0, ..., n_N - 1$ und $\tilde{j} = 0, ..., n^{\text{loc}}_N$.

Für alle $i = 1, ..., N-1$ gilt dann analog:

$$
\begin{aligned}
& u_\ell^{(i)} \left(x_{1,j_1}, ..., x_{i-1,j_{i-1}}, x^{\mathrm{loc}}_{i,j_i,\tilde{j}_i}, x^{\mathrm{loc}}_{i+1,j_{i+1},\tilde{j}_{i+1}}, ..., x^{\mathrm{loc}}_{N,j_N,\tilde{j}_N} \right) \\
:= & \frac{F\left(x_{1,j_1}, ..., x_{i-1,j_{i-1}}, x_{i,j_i+1}, x^{\mathrm{loc}}_{i+1,j_{i+1},\tilde{j}_{i+1}}, ..., x^{\mathrm{loc}}_{N,j_N,\tilde{j}_N} \right) \left(x^{\mathrm{loc}}_{i,j_i,\tilde{j}_i} - x_{i,j_i} \right)}{x_{N,j_N+1} - x_{N,j_N}} \\
+ & \frac{F\left(x_{1,j_1}, ..., x_{i-1,j_{i-1}}, x_{i,j_i}, x^{\mathrm{loc}}_{i+1,j_{i+1},\tilde{j}_{i+1}}, ..., x^{\mathrm{loc}}_{N,j_N,\tilde{j}_N} \right) \left(x_{i,j_i+1} - x^{\mathrm{loc}}_{i,j_i,\tilde{j}_i} \right)}{x_{N,j_N+1} - x_{N,j_N}},
\end{aligned}
$$

für alle $j = 0, ..., n_i - 1$ und $\tilde{j} = 0, ..., n_i^{\mathrm{loc}}$.

6.2 Glättung

Wie in Abschnitt 6.1.3 bereits angesprochen, ist die Kombinationsmethode auch auf Funktionen anwendbar, die nicht notwendigerweise differenzierbar sind. Für ein in der Praxis anwendbares generisches Optimierungsverfahren sollte die zu interpolierende Funktion dennoch gewissen Glattheitsanforderungen genügen. Wie in Abschnitt 6.4.1 angegeben wird, ist für eine geeignete Abschätzung des Interpolationsfehlers eine asymptotische Fehlerentwicklung vonnöten, welche nicht stets als gegeben vorausgesetzt werden kann. Zumindest sollte die Funktion stetig sein, und es sollte auch eine gewisse Anforderung an die Güte der Stetigkeit vorausgesetzt werden. Statistisches Rauschen kann sich als äußerst unvorteilhaft erweisen, was in Abschnitt 6.2.1 motiviert wird. Daher ist ein vorgeschalteter Glättungsalgorithmus für das DGAFO-Verfahren unentbehrlich. Eine Möglichkeit, die Fehlerfunktion in Abhängigkeit von der Temperatur zu glätten, besteht in der Verwendung der in Anhang H.1 angesprochenen Temperaturfits. Allerdings wird die Fehlerfunktion in Abhängigkeit von den zu optimierenden Kraftfeldparametern dadurch nicht geglättet. Daher werden in Abschnitt 6.2.2 einige Glättungsverfahren vorgestellt, und zwar eine Regression mithilfe einer *Support-Vector-Machine* (SVM) und eine Approximation mit Radialen Basisfunktionen (RBFs). Im Falle von Molekularen Simulationen, bei denen statistische Unsicherheiten gemäß Gleichung (3.69) geschätzt werden können, kann letztere mithilfe der in Abschnitt 3.5.5 hergeleiteten Abschätzungen mit einer gewichteten Regression kombiniert werden. Da dies jedoch nicht für alle physikalischen Zielgrößen möglich ist, wie beispielsweise im Falle von Transporteigenschaften, muß eine generische Lösung gefunden werden, statistisches Rauschen geeignet herauszufiltern. Dies geschieht mithilfe sogenannter *Regularisationstechniken*. In Abschnitt 6.2.3 werden die wichtigsten kurz vorgestellt. Im allgemeinen handelt es sich dabei um *Elastische Netze*. Zwei sehr wichtige Spezialfälle sind dabei *Ridge Regression* und das *LASSO-Verfahren*. Die Auswahl von Glättungsverfahren wird aus theoretischer Sicht schließlich in Abschnitt 6.2.4 diskutiert. Regularisierungsverfahren sind so zu wählen, daß die geglättete Funktion für die Kombinationsmethode aus Abschnitt 6.1.3 geeignet ist. Dies kann nur praktisch evaluiert werden. Eine eingehende praktische Evaluation von Glättungs- und Regularisierungsverfahren erfolgt in Abschnitt 7.1.

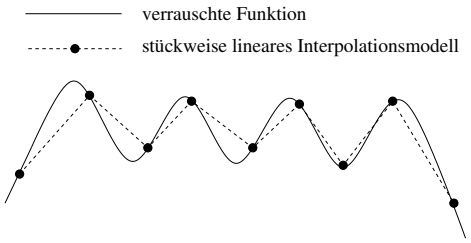


Abbildung 6.5: Problem bei stückweise linearer Interpolation einer verrauschten Funktion: Die Interpolation führt zu einer Zickzackfunktion, die das Rauschen wiedergibt.

6.2.1 Auswirkung von Rauschen auf Dünn-Gitter-Interpolation

Bei der Repräsentation einer zu interpolierenden Funktion u mithilfe der hierarchischen Basis aus Abschnitt 6.1.2 ist gemäß Abschnitt 6.1.1 stets $u \in C^4$ zu gewährleisten, da ansonsten die Seminorm aus Gleichung (6.4) nicht existiert und somit Abschätzung (6.3) nach Zenger (1991) nicht anwendbar ist. Das bedeutet, daß die Koeffizienten $u_{\ell,i}$ aus Gleichung (6.7) für $\hat{\ell} \to \infty$ nicht notwendigerweise gegen 0 konvergieren. Die Koeffizienten repräsentieren den hierarchischen Überschuß, was in Abbildung 6.3 veranschaulicht ist, welche die hierarchische Darstellung eines vollen Gitters darstellt. Man sieht deutlich, daß der Überschuß immer kleiner wird, je höher das Level eines in die Interpolation eingehenden Teilgitters ist. Im Falle von statistischem Rauschen bewegt er sich allerdings innerhalb einer Fehlerschranke in der Größenordnung des Rauschens, geht also nicht mehr notwendigerweise gegen 0. Aufgrund der hohen Glattheitsanforderung und der zusätzlichen Problematik bei statistischem Rauschen ist die hierarchische Methode somit für die vorliegende Aufgabenstellung nicht geeignet.

Bei der Kombinationsmethode aus Abschnitt 6.1.3 muß u zwar keiner expliziten Glattheitsanforderung genügen, allerdings führt auch hierbei statistisches Rauschen zu Schwierigkeiten: Angenommen, u ist bis auf das Rauschen linear. Dann müßte die in Abschnitt 6.1.4 beschriebene stückweise lineare Interpolation die Funktion genau wiedergeben. Rauschen kann jedoch bei der Interpolation zu einer Zickzackfunktion führen, was durch Abbildung 6.5 gezeigt wird. Da dies selbstverständlich nicht gewünscht ist, ist eine vorgeschaltete Glättung erforderlich, so daß das Rauschen so gut wie möglich eliminiert wird.

Daher beinhaltet das DGAFO-Verfahren bei statistischem Rauschen stets eine der in den folgenden Abschnitten vorgestellten Glättungsmethoden. Es ist allerdings zu beachten, daß auch im Falle einer Glättung die Gitterpunkte nicht zu nah beieinander liegen dürfen, da die Datenmenge nicht groß genug ist, um den korrekten Trend der Fehlerfunktion zu modellieren. Abbildung 6.6 zeigt, daß die Funktion, je nachdem wie das Rauschen sich auswirkt, komplett falsch wiedergegeben werden kann.

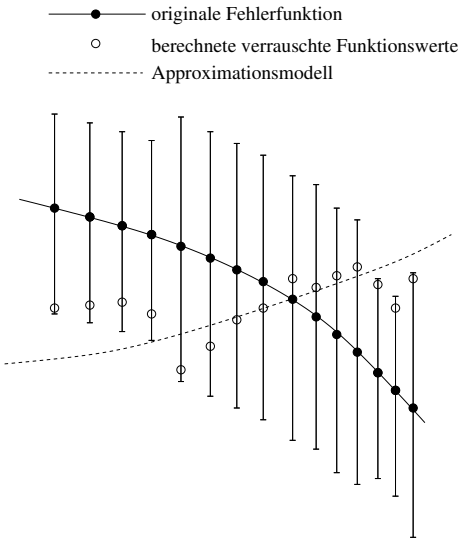


Abbildung 6.6: Problem bei einer Approximation der zu minimierenden verrauschten Funktion, wenn die Punkte zu nah beieinander liegen: Die Funktionswerte sind aufgrund des Rauschens nicht mehr unterscheidbar, und der Trend kann durch die Glättung komplett falsch wiedergegeben werden.

6.2.2 Glättungsverfahren

Im folgenden werden die eingesetzten Glättungsmethoden (SVM und RBFs) kurz vorgestellt. Die Auswahl der besten Glättungsmethode wird anschließend diskutiert.

Support-Vector-Regression Eine *Support-Vector-Machine (SVM)* eignet sich sehr gut zur robusten Modellierung von großen Datenmengen. Sie gehört zu den Methoden des überwachten maschinellen Lernens, das heißt, jedem Punkt ist ein sogenanntes *Label* zugeordnet, welches eine Klasse (*SVM-Klassifikation*) oder ein numerischer Wert (*SVM-Regression*) sein kann. Durch das von der SVM erstellte Modell können Vorhersagen für neue, unbekannte Daten getroffen werden. Robust bedeutet in diesem Falle, daß Ausreißern keine oder nur wenig Bedeutung beigemessen wird und kleine Änderungen in den Daten nur geringe Auswirkungen auf das Modell haben. Insofern kann eine SVM auch bei statistischem Rauschen angewandt werden.
Die mathematische Theorie einer SVM basiert auf dem *Perceptron-Algorithmus* nach Rosenblatt (1957). Die eigentliche Idee der SVM geht auf Vapnik (1995) zurück. Sowohl bei Klassifikation als auch Regression ist ein Optimierungsproblem zu lösen. Die Datenpunkte $x_i,\ i = 1, ..., m,\ m \in \mathbb{N}$, werden dabei durch eine Abbildung Φ in einen sogenannten *Merkmalsraum* abgebildet, welcher mathematisch beschreibbar ist und in dem ein Skalarprodukt $\langle \cdot, \cdot \rangle$ definiert ist. Die Dimension des Merkmalsraums wird so hoch angesetzt, daß die Punkte dort linear separabel, das heißt durch eine Hyperebene trennbar, sind. Das Ziel einer SVM besteht darin, die Bandbreite dieser Hyperebene zu maximieren und gleichzeitig den Modellfehler zu minimieren. Eine ausführliche Beschreibung der SVM-Klassifikation befindet sich in Schölkopf u. Smola (2002), Shawe-Taylor u. Christianini (2004) und Hülsmann (2006).

Bei der SVM-Regression unterscheidet man zwischen ϵ- und ν-Regression, welche im folgenden

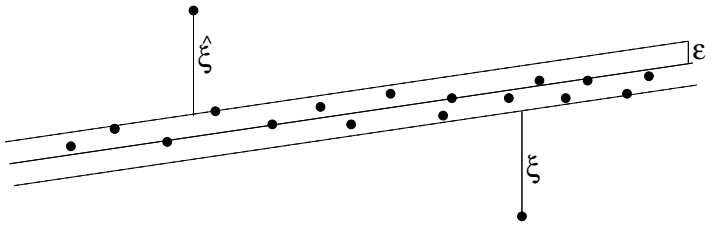


Abbildung 6.7: Trennende Hyperebene im Falle einer ϵ-SVM: Alle Datenpunkte befinden sich in einem möglichst flachen ϵ-Schlauch. Den Punkten, die außerhalb dieses Schlauchs liegen, werden Schlupfvariablen ξ und $\hat{\xi}$ zugeordnet. Es handelt sich dabei um Modellfehler, die bis zu einem gewissen Ausmaß toleriert werden, um den Testfehler zu minimieren.

kurz vorgestellt werden: Das Ziel der ϵ-Regression besteht darin, durch die Datenpunkte eine Hyperebene mit möglichst geringer Steigung zu legen, so daß sich möglichst viele Daten in einem ϵ-Schlauch (in der Praxis wird zumeist $\epsilon = 0.1$ gewählt) um diese Hyperebene befinden. Es dürfen nicht alle Daten in diesem Schlauch liegen, ansonsten ist das Modell überangepaßt, was eine SVM möglichst vermeiden soll. Es seien $y_i \in \mathbb{R}$ für $i = 1, ..., m$ die Label der Datenpunkte. Man führt nun in das Optimierungsproblem Schlupfvariablen $\xi_i,\ \hat{\xi}_i \in \mathbb{R}$ ein, je nachdem, ob sich ein Datenpunkt außerhalb des ϵ-Schlauchs unterhalb oder oberhalb der Hyperebene befindet (siehe Abbildung 6.7). Falls w der Normalenvektor der Hyperebene ist und b deren affine Verschiebung, so lautet das primäre Optimierungsproblem einer ϵ-SVM-Regression:

Definition 6.2.1 (Primäres ϵ-SVM-Optimierungsproblem)**.** *Das* primäre Optimierungsproblem *einer ϵ-SVM-Regression lautet:*

$$\min_{\xi,w,b} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\left(\xi_i + \hat{\xi}_i\right), \tag{6.13}$$

wobei
$$(\langle w, x_i\rangle + b) - y_i \geq \epsilon + \xi_i,\ \ i = 1, ..., m, \tag{6.14}$$
$$y_i - (\langle w, x_i\rangle + b) \geq \epsilon + \hat{\xi}_i,\ \ i = 1, ..., m, \tag{6.15}$$
$$\xi_i,\ \hat{\xi}_i \geq 0,\ \ i = 1, ..., m. \tag{6.16}$$

Dabei ist $C > 0$ eine Regularitätskonstante, die für den Kontrast steht, die Bandbreite zu maximieren und gleichzeitig den Modellfehler zu minimieren.
Es gilt:

- $(\langle w, x_i\rangle + b) - y_i > \epsilon \Rightarrow \xi_i > 0,\ \ i = 1, ..., m,$
- $y_i - (\langle w, x_i\rangle + b) > \epsilon \Rightarrow \hat{\xi}_i > 0,\ \ i = 1, ..., m.$

Dieses nichtlineare Optimierungsproblem mit Nebenbedingungen wird mithilfe der Lagrange-Theorie (siehe zum Beispiel Shawe-Taylor u. Christianini (2004)) gelöst. Dabei werden Lagrange-Multiplikatoren $\alpha_i,\ i = 1, ..., m$, eingeführt. Sind die Datenpunkte im Eingaberaum nicht linear separabel, so wird anstelle des Skalarprodukts eine sogenannte *Kernfunktion* $k(x_i, x_j) := \langle \Phi(x_i), \Phi(x_j)\rangle,\ i, j = 1, ..., m$, verwendet, und man erhält das folgende duale Optimierungsproblem einer ϵ-SVM:

Definition 6.2.2 (Duales ϵ-SVM-Optimierungsproblem)**.** *Das* duale Optimierungsproblem einer ϵ-SVM-Regression *lautet:*

$$\max_{\alpha} \sum_{i=1}^{m} y_i \alpha_i - \epsilon \sum_{i=1}^{m} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j), \tag{6.17}$$

$$\text{wobei} \quad \sum_{i=1}^{m} \alpha_i = 0, \ -C \leq \alpha_i \leq C, \ i = 1, ..., m. \tag{6.18}$$

Dieses Optimierungsproblem wird mithilfe eines Newton-Lagrange-Verfahrens gelöst.
Die Vektoren, für die $0 < |\alpha_i| < C$ gilt, heißen *Stützvektoren* (englisch *Support Vectors*). Es sind diejenigen Vektoren, die auf dem ϵ-Schlauch um die Hyperebene liegen. Für diejenigen Vektoren, die außerhalb des ϵ-Schlauchs liegen, gilt $|\alpha_i| = C$ und für diejenigen, die innerhalb liegen, gilt $\alpha_i = 0$. Um Überangepaßtheit des Modells zu vermeiden, sollte die SVM mit möglichst wenig Stützvektoren auskommen. Ein unbekannter Datenpunkt $\bar{x}$ (sogenannter *Testpunkt*) kann über die folgende Regressionsfunktion f mithilfe der Lösung α^* des Optimierungsproblems aus Definition 6.2.2 bestimmt werden:

$$f(\bar{x}) = \sum_{i,j=1}^{m} \alpha_i^* k(x_i, x_j) + b^*, \tag{6.19}$$

wobei b^* so gewählt ist, daß $\forall_{i=1,...,m, \ 0<\alpha_i^*<C} \ f(x_i) - y_i = -\epsilon$. Als Kernfunktionen stehen lineare, sigmoide und polynomielle Kerne sowie Radiale Basisfunktionen zur Verfügung. In dieser Arbeit wird der Gauß-Kern $k(x_1, x_2) = \exp\left(-\gamma \|x_1 - x_2\|^2\right)$ verwendet. Der Funktionsparameter γ und die Regularitätskonstante C werden im allgemeinen durch ein sogenanntes *Grid Search* so bestimmt, daß der Testfehler möglichst klein ist. In Hülsmann (2006) hat sich der Gauß-Kern als sehr gut geeignet erwiesen. Dort ist auch das hier verwendete *Bootstrapping* zur Bestimmung von C und γ detailliert beschrieben. Da die SVM in dieser Arbeit zur Glättung eingesetzt werden soll, müssen die Parameter so bestimmt werden, daß das Rauschen möglichst aus der Fehlerfunktion eliminiert wird. Inwieweit eine SVM für eine Funktionsglättung geeignet ist, wird in Abschnitt 7.1 diskutiert. In Brühl u. a. (2009) wurde eine ϵ-SVM für Zeitreihenanalysen für die Automobilindustrie eingesetzt. Dabei konnte jedoch bereits festgestellt werden, daß die SVM stets die Saisonkomponente in das Modell miteinbezogen hat und somit ein multivariater Trend, welcher die Zeitreihe glättet, erst nach einer vorgeschalteten Saisonbereinigung errechnet werden konnte. Die Datenmenge war in diesem Fall und ist auch im Falle des DGAFO-Verfahrens nicht groß genug, um eine Überangepaßtheit des Modells auszuschließen, so daß Oszillationen im Modell mitberücksichtigt werden können.

Desweiteren wird in diesem Abschnitt eine ν-SVM-Regression vorgestellt, welche aufgrund des mathematisch besser handhabbaren Parameters ν eher zur Glättung geeignet sein könnte:

Definition 6.2.3 (Primäres ν-SVM-Optimierungsproblem)**.** *Das* primäre Optimierungsproblem *einer ν-SVM-Regression lautet:*

$$\min_{\xi,w,b,\epsilon} \ \frac{1}{2}\|w\|^2 + C\left(\nu\epsilon + \frac{1}{m}\sum_{i=1}^{m}\left(\xi_i + \hat{\xi}_i\right)\right), \tag{6.20}$$

wobei
$$(\langle w, x_i\rangle + b) - y_i \geq \epsilon + \xi_i, \ \ i = 1, ..., m, \tag{6.21}$$
$$y_i - (\langle w, x_i\rangle + b) \geq \epsilon + \hat{\xi}_i, \ \ i = 1, ..., m, \tag{6.22}$$
$$\xi_i, \ \hat{\xi}_i \geq 0, \ \ i = 1, ..., m. \tag{6.23}$$

Dabei sind $\xi, \hat{\xi}$ und C analog zur ϵ-SVM. Der Parameter ϵ, der die Modellkomplexität steuert, wird nun in die Optimierung miteinbezogen. Die Größe von ϵ wird direkt von der Konstanten $\nu \geq 0$ beeinflußt. Ein duales Optimierungsproblem ergibt sich analog zu Definition 6.2.2.
Die Bedeutung von ν ergibt sich aus dem folgenden Satz:

Satz 6.2.4. *Bei einer ν-SVM-Regression mit $\nu \geq 0$ und resultierendem $\epsilon \neq 0$ gilt:*

- *ν ist obere Schranke für den Anteil an Bandbreitenfehlern.*
- *ν ist untere Schranke für den Anteil an Stützvektoren.*

Beweis: Der Beweis ist in Schölkopf u. Smola (2002) zu finden. □

Der Anteil an Bandbreitenfehlern ist dabei wie folgt definiert:

Definition 6.2.5 (Bandbreitenfehler)**.** *Sei μ die Bandbreite der Hyperebene und g so, daß $f = \text{sign} \circ g$. Dann ist durch*

$$R^{\mu}_{\text{emp}}[g] := \frac{1}{m}\left|\{i | y_i g(x_i) < \mu\}\right| \tag{6.24}$$

der Anteil an Bandbreitenfehlern *definiert.*

Ein Bandbreitenfehler tritt genau dann auf, wenn $\exists_i \ \xi_i > 0$ oder $\exists_i \ \hat{\xi}_i > 0$. Falls die SVM also als Glättungsverfahren eingesetzt werden soll, besteht die Möglichkeit, dies über den Parameter ν zu steuern: Je größer ν ist, desto weniger Datenpunkte liegen im ϵ-Schlauch um die Hyperebene. Die restlichen Punkte werden als Ausreißer angesehen. Um statistisches Rauschen zu eliminieren, sollten möglichst viele Punkte als Ausreißer angesehen werden, damit lediglich der Trend der Fehlerfunktion wiedergegeben wird. Daher ist die ν-SVM eine denkbare Glättungsmethode für das DGAFO-Verfahren und möglicherweise besser geeignet als die ϵ-SVM. Es ist allerdings zu beachten, daß zum einen die Parametrisierung einer SVM durch *Grid Search* relativ rechenaufwendig ist, und zum anderen, daß es nur sehr wenige innere Punkte auf einem Dünnen Gitter gibt. Eine SVM neigt in diesem Fall dazu, zu viele Punkte als Ausreißer zu erkennen, so daß ein viel zu simples Modell resultiert. Letzteres ist selbstverständlich für die vorliegende Problemstellung nicht gewünscht.

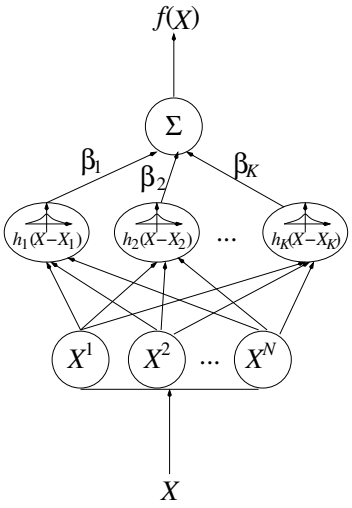


Abbildung 6.8: Darstellung eines RBF-Netzes: Das Netz verläuft für einen Testpunkt X von den Eingabeneuronen X^j, $j = 1, ..., N$, (Komponenten von X) über eine Schicht versteckter Neuronen, die für die RBFs $h_i(X - X_i)$, $i = 1, ..., K$, stehen und mit Koeffizienten β_i, $i = 1, ..., K$, in Richtung des Ausgabeneurons, wo die Aufsummierung stattfindet. Das Resultat ist ein dem Testpunkt X zugeordneter Funktionswert $f(X)$.

Approximation mit Radialen Basisfunktionen Im folgenden wird die Approximation höherdimensionaler Funktionen mittels *Radialer Basisfunktionen (RBFs)* und *Verallgemeinerter Radialer Basisfunktionen (GRBFs)* kurz vorgestellt.
RBFs sind eine effiziente sowie weit verbreitete Methode zur Interpolation und Approximation von Funktionen und sind beispielsweise in Powell (1987) beschrieben. *Radial* bedeutet dabei, daß sie in Abhängigkeit vom Abstand zweier benachbarter Punkte dargestellt werden. Punkte, die auf einer Hyperkugel vom Radius r um einen Bezugspunkt liegen, haben somit denselben Funktionswert. Beispiele für RBFs sind:

$$\phi(r) = r \text{ (linear)} \tag{6.25}$$
$$\phi(r) = \exp\left(-cr^2\right),\ c \in \mathbb{R}^+ \text{ (Gauß-Funktion)} \tag{6.26}$$
$$\phi(r) = \sqrt{r^2 + c^2},\ c \in \mathbb{R}^{\neq 0} \text{ (multiquadrisch)} \tag{6.27}$$
$$\phi(r) = \frac{1}{\sqrt{r^2 + c^2}},\ c \in \mathbb{R}^{\neq 0} \text{ (invers multiquadrisch)} \tag{6.28}$$
$$\phi(r) = r^3 \text{ (kubisch)} \tag{6.29}$$
$$\phi(r) = r^2 \log r \text{ (Thin-Plate-Splines)} \tag{6.30}$$

Eine Funktion $f : \mathbb{R}^N \to \mathbb{R}$ wird durch

$$f(X) \approx \sum_{i=1}^{K} \beta_i h_i \left(\|X - X_i\|\right) \tag{6.31}$$

approximiert, wobei $X \in \mathbb{R}^N$ ein beliebiger Testpunkt ist und $K \leq m$. Die Punkte $X_i \in \mathbb{R}^N$, $i = 1, ..., K$, stehen für K repräsentative Zentren aus der Trainingsmenge, auf der die

Koeffizienten $\beta_i \in \mathbb{R},\ i = 1, ..., K$, bestimmt werden. Die Funktionen $h_i \in \mathcal{H},\ i = 1, ..., K$, sind aus der Menge $\mathcal{H}$ der RBFs.
Die Bestimmung der Koeffizienten $\beta_i,\ i = 1, ..., K$, erfolgt über sogenannte *RBF-Netze* (siehe Abbildung 6.8), welche gerichtete neuronale Netze sind (Bishop, 2007) und von den Eingabe- in Richtung der Ausgabeneuronen verlaufen. Sie besitzen lediglich eine Schicht von verdeckten Neuronen, welche der Koeffizientenbestimmung entspricht.
Bei einer Approximation gilt $m \leq K$, und es ist stets das folgende überbestimmte LGS zu lösen:

$$H\beta = Y, \tag{6.32}$$

wobei

$$H = \begin{pmatrix} h_1(\|X_1 - X_1\|) & h_2(\|X_1 - X_2\|) & \cdots & h_K(\|X_1 - X_K\|) \\ h_1(\|X_2 - X_1\|) & h_2(\|X_2 - X_2\|) & \cdots & h_K(\|X_2 - X_K\|) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(\|X_m - X_1\|) & h_2(\|X_m - X_2\|) & \cdots & h_K(\|X_m - X_K\|) \end{pmatrix} \in \mathbb{R}^{m \times K}$$

und

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} f(X_1) \\ f(X_2) \\ \vdots \\ f(X_m) \end{pmatrix} \in \mathbb{R}^m.$$

Der Lösungsvektor $\beta := (\beta_1, ..., \beta_K)^T \in \mathbb{R}^K$ kann zum Beispiel mithilfe der Methode der kleinsten Quadrate bestimmt werden.
Um K repräsentative Zentren aus den m Trainingsdaten effizient auszuwählen, wird ein unüberwachtes Lernverfahren angewandt, und zwar eine automatische Klassifikation (sogenanntes *Clustering*). Hierzu wird das klassische *k-means-Verfahren* nach MacQueen (1967) verwendet. Die K Zentren sind gleichzeitig die Zentroide der aus dem Clustering ermittelten Datenklassen.
Aufgrund der vorhandenen statistischen Unsicherheiten kann auch eine gewichtete lineare Regression durchgeführt werden. Die Gewichte für einen Datenpunkt x können dabei wie folgt bestimmt werden: In Abschnitt 3.5.5 wurde die Auswirkung des Rauschens auf die Gesamtfehlerfunktion F theoretisch analysiert. Die Abschätzungen (3.76) und (3.93) definieren eine Fehlerschranke für den Funktionswert $F(x)$. Es gilt die Abschätzung

$$\begin{aligned} \frac{\tilde{F}(x) - D(x)}{1 + 2C} &= F(x) - \frac{2CF(x) + 2C + \bar{C}(x)}{1 + 2C} && (6.33) \\ &\leq F(x) && (6.34) \\ &\leq F(x) + \frac{2CF(x) + 2C + \bar{C}(x)}{1 - 2C} = \frac{\tilde{F}(x) + D(x)}{1 - 2C}. && (6.35) \end{aligned}$$

Für einen Punkt x kann somit als Gewicht $w(x)$ die inverse halbe Länge des Konfidenzintervalls, also

$$w(x) := \left[\frac{1}{2} \left(\frac{\tilde{F}(x) + D(x)}{1 - 2C} - \frac{\tilde{F}(x) - D(x)}{1 + 2C} \right) \right]^{-1}, \tag{6.36}$$

gewählt werden. Die Schranke $C = \sum_{i=1}^{n} C^i$ (siehe Gleichung (3.89)) kann aus den in der Simulation geschätzten Fehlern $\mathcal{F}^i(f^i(x), M) := \frac{\sigma(f^i(x))}{\sqrt{M}}$ (siehe Gleichung (3.69)) für die $f^i(x),\ i =$

$1, ..., n$, ermittelt werden. Wegen $\mathcal{F}^i(f^i(x), M) \approx c^i f^i(X)$ wird dabei

$$\forall_{i=1,...,n}\ C^i = C^i(x) := \frac{\mathcal{F}^i(f^i(x), M)}{\tilde{f}^i(x)} \tag{6.37}$$

gesetzt. Die Schranke $\bar{C}(x) = \sum_{i=1}^{n} \bar{C}^i(x)$ (siehe Gleichung (3.89)) kann gemäß Ungleichung (3.75) ebenfalls aufgrund von $c^i f^i(x) \approx \mathcal{F}^i(f^i(x), M)$ bestimmt werden.
Wegen Gleichung (6.36) erfolgt die Gewichtung antiproportional zum Ausmaß des statistischen Rauschens. Somit kann eine Überangepaßtheit der glättenden RBF vermieden werden, da das Rauschen *a priori* quantitativ in die Regression miteinbezogen wird. Diese Methodik macht jedoch nur dann Sinn, wenn sich die statistischen Unsicherheiten, mit denen die zu minimierenden Funktion F behaftet ist, für verschiedene Punkte auf einem Dünnen Gitter signifikant voneinander unterscheiden.

Eine Verallgemeinerung der RBFs wurde von Sturm u. Pietruschka (1997) eingeführt: Eine *Generalisierung von Radialen zu Ellipsoiden Basisfunktionen (GRBF)* erwies sich insbesondere dann als geeignet, wenn die Form der verwendeten Radialen Basisfunktionen sich für die Approximation als ungeeignet erwies oder wenn starke Variationen der zu approximierenden Funktion nicht nur von einzelnen Komponenten des Eingaberaums, sondern auch von beliebigen Linearkombinationen, abhingen. Die Generalisierungsidee besteht darin, durch die Einführung *wachsender Ellipsoide* den Approximationsfehler lokal zu minimieren. Die Ausdehnung der Ellipsiode geht dabei stets in Richtung des Ortes, an dem der Approximationsfehler am größten ist. An dieser Stelle wird dann eine neue Radiale Basisfunktion gemäß Gleichung (6.31) hinzugefügt, wobei das neue Gewicht c_{k+1} und die ellipsoide Ausdehnung um das neue Zentrum X_{k+1} so eingestellt werden, daß der Approximationsfehler am Zentrum und am dem Zentrum nächstgelegenen Datenpunkt verschwindet. So erhält man ein adaptives Verfahren, welches in Abhängigkeit vom Approximationsfehler die Form der Approximationsfunktion verändert. Es ermöglicht eine sehr genaue Modellierung von äußerst komplexen Funktionen, was bei der vorliegenden Problematik jedoch nicht wünschenswert ist, da hier das statistische Rauschen durch die Glättung eliminiert werden muß. Bei der Generalisierung der RBFs besteht die Gefahr, daß durch die Adaptivität dem Rauschen eine viel zu hohe Bedeutung beigemessen wird.

6.2.3 Regularisationstechniken

Wie in Abschnitt 6.2.2 bereits motiviert, sollte jede vorgeschaltete Glättungsprozedur dazu in der Lage sein, statistisches Rauschen in geeigneter Art und Weise herauszufiltern, um Überangepaßtheit zu vermeiden. Bei der Bestimmung der Regressionskoeffizienten (Lösung des dualen Optimierungsproblems (6.18) im Falle einer SVM und Lösung des LGS aus Gleichung (6.32) im Falle von RBFs) dürfen diese nicht überschätzt werden, damit kein zu hohes Gewicht auf verrauschte Datenpunkte gelegt wird. Dies kann zum einen durch eine *gewichtete Regression* erfolgen, indem jedem Datenpunkt ein Gewicht zugeordnet wird, das umso kleiner ist, je größer die statistische Unsicherheit bezüglich seines Funktionswertes ist. Zum anderen können Nebenbedingungen für die Regressionskoeffizienten festgesetzt werden, um zu vermeiden, daß diese betraglich zu groß werden. Bei einer SVM geschieht dies durch die Definition des primären Optimierungsproblems (6.16), bei dem durch die Einführung der Schlupfvariablen $\xi_i,\ i = 1, ..., m$, gewisse Trainingsfehler toleriert werden, um Überangepaßtheit zu vermeiden.

Dies hat zur Folge, daß die Regressionskoeffizienten $\alpha_i^*,\ i = 1, ..., m$, die das duale Optimierungsproblems lösen, außer im Falle der Stützvektoren gleich 0 sind. Außerdem gilt die Nebenbedingung $\forall_{i=1,...,m}\ 0 \leq |\alpha_i^*| \leq C$. Eine Regularisierung ist in einer SVM also bereits implizit enthalten. Wie verschiedene Anwendungen gezeigt haben (Hülsmann, 2006; Hülsmann u. Friedrich, 2007; Brühl u. a., 2009; Hülsmann u. a., 2011a), ist eine SVM zusammen mit einer Gaußschen Kernfunktion eine geeignete nichtlineare Regressionsmethode, die gleichzeitig robust in Bezug auf Rauschen und vor allem Ausreißer ist, falls C und γ optimal gewählt sind. Das bedeutet, eine SVM hat bereits Regularisierungseigenschaften intern implementiert. Ob sie allerdings bei einer derart kleinen Trainingsmenge wie im vorliegenden Fall geeignet ist (es ist zu beachten, daß es sich stets um ein dünnes Gitter handelt) und dabei im Falle von Rauschen nicht zur Überangepaßtheit neigt, wird im Rahmen einer Evaluation in Abschnitt 7.1 diskutiert.

Im Falle der RBFs sind verschiedene Regularisierungsverfahren zu evaluieren, welche das LGS (6.32) auf eine andere Art und Weise lösen. Die einfachste Möglichkeit, das überbestimmte LGS zu lösen, liefert die *Methode der Kleinsten Quadrate*. Der sogenannte *KQ-Schätzer* β_{KQ} wird dann durch das Optimierungsproblem

$$\beta_{\text{KQ}} = \arg\min_{\beta} ||Y - H\beta||_2^2 \tag{6.38}$$

bestimmt. Der KQ-Schätzer hat jedoch die folgenden Nachteile:

1. Er ist zwar erwartungstreu, weist jedoch eine ziemlich hohe Varianz auf, was zur Überangepaßtheit führen kann.
2. Die Koeffizienten des KQ-Schätzers werden oftmals zu groß geschätzt, das heißt, die euklische Norm $||\beta_{\text{KQ}}||_2$ ist im allgemeinen recht hoch.
3. Die Methode der Kleinsten Quadrate führt keine Variablenselektion durch. Das bedeutet, daß sämtliche Koeffizienten von β_{KQ} ungleich 0 sind. Somit werden sogar miteinander korrelierte Variablen im Modell mitberücksichtigt, auch wenn sie keinen oder nur einen geringen Einfluß auf die Zielgrößen haben.
4. Ist die Dimension echt größer als die Anzahl an Datenpunkten, was in diesem Fall allerdings wegen $K << m$ ausgeschlossen ist, liefert die Methode der Kleinsten Quadrate keine Lösung.

Um diesen Nachteilen entgegenzuwirken, sind die Regressionskoeffizienten zu *schrumpfen* oder teilweise auf 0 zu setzen, was einer sogenannten *Variablenselektion* gleichkommt. Dies bewirkt, daß gewisse Variablen, besonders korrelierte, innerhalb des Modells keinen oder nur noch einen geringen Einfluß besitzen und Ausreißern keine oder nur wenig Bedeutung beigemessen wird. Weiterhin wird dadurch auch die Varianz des Schätzers reduziert, so daß auch die Wahrscheinlichkeit, ein überangepaßtes Modell zu erhalten, sinkt. Um dies mathematisch zu realisieren, wird für eine Konvexkombination aus euklidischer und 1-Norm eine Nebenbedingung in das Optimierungsproblem (6.38) eingeführt. Es wird dann das folgende Optimierungsproblem gelöst:

$$\beta_{\text{NEN}} = \arg\min_{\beta} ||Y - H\beta||_2^2,\ \text{wobei}\ \alpha||\beta||_2^2 + (1-\alpha)|\beta|_1 \leq t,\ \alpha \in [0,1],\ t > 0. \tag{6.39}$$

Dieses Verfahren heißt *Naives Elastisches Netz* und geht auf Zou u. Hastie (2005) zurück. Es enthält zwei zusätzliche Modellparameter $\alpha \in [0,1]$ und $t \in \mathbb{R}^{>0}$, die zum Beispiel durch zehnfache Kreuzvalidierung bestimmt werden können (für weitere Details, siehe Zou u. Hastie (2005)). Ein Naives Elastisches Netz besitzt einen sogenannten *Gruppierungseffekt*, das heißt, korrelierte Modellvariablen haben ähnliche Regressionskoeffizienten. Aus der Lagrange-Theorie ergibt sich eine äquivalente penalisierte Schreibweise durch Einführung von Lagrange-Multiplikatoren $\lambda_1, \lambda_2 \geq 0$:

$$\beta_{\text{NEN}} = \arg\min_{\beta} \mathcal{L}(\beta, \lambda_1, \lambda_2),\ \mathcal{L}(\beta, \lambda_1, \lambda_2) := ||Y - H\beta||_2^2 + \lambda_2||\beta||_2^2 + \lambda_1|\beta|_1. \tag{6.40}$$

Es gilt $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2}$. Im Falle von $\alpha \in \{0,1\}$ erhält man die zwei bekanntesten Verfahren im Bereich verzerrter Regression: Bei $\alpha = 1$ handelt es sich um eine sogenannte *Ridge Regression*, welche auf Hoerl u. Kennard (1970) zurückgeht, und bei $\alpha = 0$ um den sogenannten **L***east* **A***bsolute* **S***hrinkage and* **S***election* **O***perator (LASSO)*, zurückzuführen auf Tibshirani (1996). Ridge-Regression führt zu dem Schätzer β_R und das LASSO-Verfahren zu dem Schätzer β_L. Beide Verfahren führen eine Variablenschrumpfung durch, wobei nur im Falle des LASSO-Verfahrens eine Variablenselektion möglich ist. Ein Naives Elastisches Netz mit $\alpha \notin \{0,1\}$ funktioniert gemäß Zou u. Hastie (2005) folgendermaßen: Für jedes feste λ_2 werden zunächst die Ridge-Regression-Koeffizienten ermittelt, und anschließend wird eine Variablenschrumpfung vom LASSO-Typ durchgeführt. Dies führt also zu einer doppelten Schrumpfung, was wiederum den Testfehler erhöht und die Vorhersagegüte senkt. Es ist also stets nach einem Kompromiß zu suchen zwischen Maximierung der Varianz mittels Schrumpfung und Vermeiden eines zu naiven Prädiktors: Sogenannte *Elastische Netze*, welche ebenfalls in Zou u. Hastie (2005) beschrieben werden, multiplizieren β_{NEN} mit dem Faktor $\sqrt{1+\lambda_2}$, was eine doppelte Schrumpfung vermeidet und gleichzeitig die Möglichkeit der Variablenselektion erhält. Allerdings ist die Parametrisierung eines Elastischen Netzes viel rechenaufwendiger als die des LASSO-Verfahren und der Ridge-Regression, da mit λ_1 und λ_2 zwei Variablen zu bestimmen sind.

Im folgenden werden einige wichtige Eigenschaften sowohl der Ridge-Regression als auch des LASSO-Verfahrens näher erläutert: Wie bereits erwähnt, ist das LASSO-Verfahren im Gegensatz zur Ridge-Regression dazu in der Lage, eine Variablenselektion durchzuführen, was in Abbildung 6.9 verdeutlicht wird: Aufgrund der $|\cdot|_1$-Norm handelt es sich bei den Gebieten, in denen die Nebenbedingungen erfüllt sind, um Rauten mit Diagonalen der Länge $2t$, wohingegen im Falle der Ridge-Regression Kreise mit Radien der Länge $\sqrt{c}$ vorliegen. Die Schnittpunkte der Konturlinien der zu minimierenden Residuenquadratsumme, deren globales Minimum bei β_{KQ} liegt, mit den Kreisen können nicht auf einer der Koordinatenachsen liegen. Die Schnittpunkte mit den Rauten hingegen liegen ab einem gewissen kleiner werdenden t alle auf einer der Koordinatenachsen. Das bedeutet, daß der entsprechende Koeffizient gleich 0 ist. Im Falle des LASSO-Verfahrens haben die so entstehenden *Pfade*, die durch die Schnittpunkte der Rauten mit den Konturlinien führen, in 2D einen charakteristischen Verlauf: Zunächst sind sie konstant 0, bis sie an einem bestimmten Punkt in Abhängigkeit von t in eine bestimmte Richtung abknicken.
Sowohl Ridge-Regression als auch das LASSO-Verfahren liefern nicht erwartungstreue, also verzerrte Schätzer, die jedoch eine geringere Varianz als der KQ-Schätzer aufweisen und so-

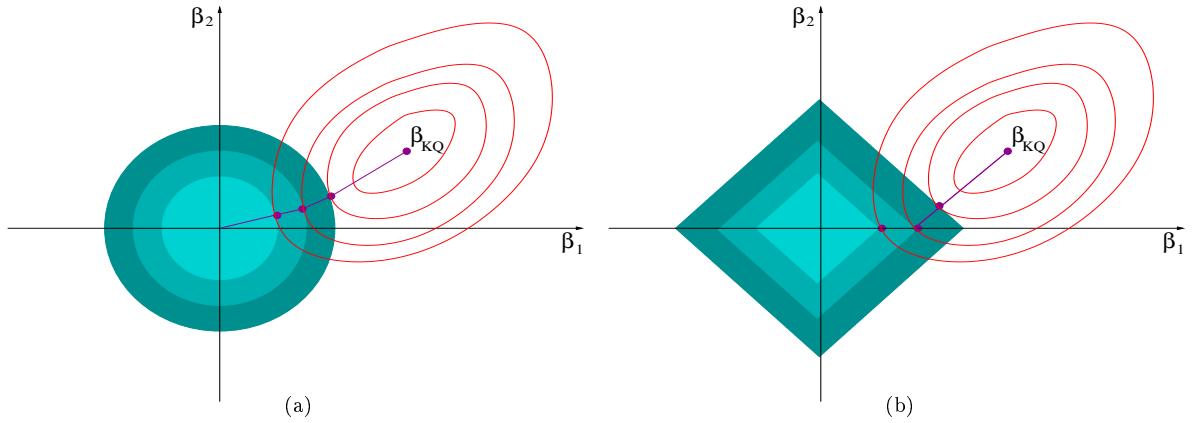


Abbildung 6.9: Geometrie der *Ridge-Regression* (a) und des *LASSO-Verfahrens* (b) in 2D: Im ersten Fall liefert die Nebenbedingung $||\beta||_2^2 \leq c$ Kreise mit Radien der Länge $\sqrt{c}$ und im zweiten Fall die Nebenbedingung $|\beta|_1 \leq t$ Rauten mit Diagonalen der Länge $2t$. Die roten Konturlinien, die die Residuenquadratsumme mit Minimum β_{KQ} beschreiben, wobei β_{KQ} der Kleinste-Quadrate-Schätzer ist, berühren die Kreise beziehungsweise Rauten in den violetten Punkten. Nur im Falle des LASSO-Verfahrens kann eine Koordinate dieser Schnittpunkte (β_1) zu 0 werden. Die violetten Linien zeigen die jeweiligen Pfade, welche durch die Schnittpunkte verlaufen.

mit weniger zu überangepaßten Modellen neigen. Allerdings kann die Variablenselektion des LASSO-Verfahrens dazu führen, daß bestimmten Merkmalen überhaupt keine Bedeutung beigemessen wird. Dadurch kann das resultierende Modell insbesondere auf einer kleinen Trainingsmenge zu simpel sein und wiederum den Testfehler erhöhen.
Die Bestimmung der Modellparameter c und t erfolgt über eine sogenannte *Generalisierte Kreuzvalidierung (GKV)*. Im Gegensatz zu einer gewöhnlichen Kreuzvalidierung, welche selbstverständlich auch zur Bestimmung der Parameter angewandt werden kann, wird bei der GKV das folgende Maß in Abhängigkeit von c beziehungsweise t minimiert:

$$\Gamma(v) := \frac{1}{K} \frac{||Y - H\beta(v)||_2^2}{\left(1 - \frac{\mathrm{eff}(v)}{K}\right)^2}, \;\; v \in \{c, t\}, \;\; \beta \in \{\beta_R, \beta_L\}. \tag{6.41}$$

Es handelt sich dabei um eine Gewichtung der Residuenquadratsumme $||Y - H\beta(t)||_2^2$, die umso größer ist, je höher die Anzahl an effektiven Modellparametern $\mathrm{eff}(v)$ ist. Letztere kann durch die Spur der Matrix $X(X^TX + \lambda W^-)^{-1}X^T$ geschätzt werden, wobei $\lambda := \lambda(v) \in \{\lambda_1, \lambda_2\}$ und $W := (\mathrm{diag}(|\beta_j(v)|)_{j=1,\ldots,K})^{-1}$. Durch die Minimierung von Γ wird der Trainingsfehler unter Berücksichtigung der Flexibilität des Modells minimiert.

Eine weitere Methode zur Regularisierung ist die Verwendung sogenannter **M***ultivariate* **A***daptive* **R***egression* **S***plines (MARS)*, siehe Friedman (1991), da diese auch dazu in der Lage sind, statistisches Rausches herauszufiltern. Die dabei zugrundeliegenden Basisfunktionen sind Produkte sogenannter *Gelenkfunktionen* der Ordnung $q \in \mathbb{N}$:

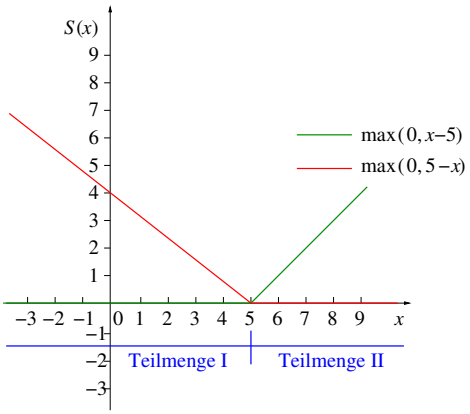


Abbildung 6.10: Einteilung des Eingaberaums in Richtung einer Dimension durch zwei lineare gespiegelte Gelenkfunktionen $\max(0, x-5)$ (grün) und $\max(0, 5-x)$ (rot), allgemein für einen Eingabepunkt x mit $S(x)$ gekennzeichnet: Bei $x = 5$ findet die Einteilung in zwei Teilmengen I und II statt. Durch Aufsummierung entsteht eine stückweise lineare Funktion.

$$S_q(x) = S_q(x-\nu) := [\pm(x-\nu)]_+^q, \ c \in \mathbb{R}, \tag{6.42}$$

wobei ν ein sogenannter *Knoten* ist. Das '+' bedeutet, daß die Funktion im Falle von negativen Argumenten zu 0 wird. Abbildung 6.10 zeigt zwei gespiegelte Gelenkfunktionen in 2D für $q = 1$ und $\nu = 5$, und zwar $[+(x-5)]_+ = \max(0, x-5)$ und $[-(x-5)]_+ = \max(0, 5-x)$. Diese teilen den Eingaberaum in x-Richtung in zwei disjunkte Teilmengen ein. Durch die rekursive Selektion von insgesamt p Knoten wird der Eingaberaum in $p+1$ disjunkte Teilräume aufgeteilt. Jedem dieser Teilräume wird eine Gelenkfunktion oder ein Produkt aus mehreren Gelenkfunktionen, welche den Teilraum weiter aufteilen, zugeordnet, was die Interaktion der Eingabevariablen modellieren soll. Insgesamt wird ein Funktional $f : \mathbb{R}^N \to \mathbb{R}$ im Falle von MARS folgendermaßen approximiert:

$$f(x) \approx \sum_{i=1}^{p} (\beta_M)_i \prod_{j=1}^{s_i} S_q(x_{N(i,j)} - \nu_{i,j}). \tag{6.43}$$

Dabei sind die s_i die Anzahlen an betrachteten Gelenkfunktionen bei der Partitionierung $i \in \{1, ..., p\}$, $\nu_{i,j}$, $i = 1, ..., p$, $j = 1, ..., s_i$, sämtliche Knoten aus der rekursiven Unterteilung und $N(i,j)$ die Dimension des Eingabevektors x, auf welche sich die durch den Knoten $\nu_{i,j}$ gegebene Unterteilung bezieht. Insgesamt wird f durch einen Regressionsspline approximiert, im Falle von $q = 1$ durch eine stückweise lineare Funktion. Das Ergebnis ist der MARS-Schätzer $\beta_M = ((\beta_M)_i)_{i=1,...,p}$.
Die Selektion der Gelenkfunktionen und somit der Knoten erfolgt zunächst durch Hinzunahme sämtlicher gespiegelter Gelenkfunktionen, welche die Residuenquadratsumme reduzieren. Da ein derartiges Modell allerdings zu Überangepaßtheit führen würde, werden anschließend diejenigen Basisfunktionen herausgenommen, die nur einen geringfügigen Beitrag zur Effizienz des Modells leisten (sogenanntes *Pruning*). Auch hierbei wird wieder Γ aus Gleichung (6.41) minimiert, allerdings in Abhängigkeit von der sich im aktuellen Modell μ befindenden Ba-

sisfunktionen. Dabei ist $\text{eff}(\mu) := \sum_{i=1}^{p} s_i + \chi \frac{\sum_{i=1}^{p} s_i - 1}{2}$. Es ist zu beachten, daß $\sum_{i=1}^{p} s_i$ die Anzahl an Basisfunktionen ist, bestehend aus $\sum_{i=1}^{p} s_i - 1$ gespiegelten Gelenkfunktionen und der konstanten Funktion $S \equiv 1$. Der Faktor $\chi \in [2, 3]$ ist ein Strafterm für die Anzahl an Gelenkfunktionspaaren, das heißt, die Hinzunahme von Knoten wird durch die Formel bestraft.
Die durch das *Pruning* realisierte Vermeidung von Überangepaßtheit ermöglicht die Anwendung von MARS als Regularisationsverfahren, hat allerdings zum Nachteil, daß dies bei höherdimensionalen Optimierungsproblemen auch mit einem entsprechend hohem Rechenaufwand verbunden ist.
Als stückweise lineare Approximation kann der MARS-Algorithmus auch als reines Glättungsverfahren eingesetzt werden.

Die Frage, ob Regularisationsverfahren wie Elastische Netze und MARS in Kombination mit einer RBF-Approximation auf die vorliegende Problemstellung anwendbar sind, wird in den Abschnitten 7.2.1 und 7.3.1 anhand von praktischen Evaluationen beantwortet. Es sei jedoch jetzt schon einmal erwähnt, daß die Verfahren bei höheren Dimensionen aufgrund der Selektion der Modellparameter mit zu hohem Rechenaufwand verbunden und damit für praktische Anwendungen ungeeignet sind.

6.2.4 Auswahl von Glättungsverfahren

Die Auswahl des besten Glättungsverfahrens erfolgt in diesem Abschnitt lediglich theoretisch. Ob und inwieweit welche Verfahren für praktische Anwendungen geeignet sind, wird in Abschnitt 7.1 eingehend diskutiert. Theoretisch kann die Selektion eines Glättungsverfahrens über eine geeignete Abschätzung des Approximationsfehlers auf einem Gebiet $\Omega \subset \mathbb{R}^N$ erfolgen. Eine derartige Abschätzung existiert für positiv definite RBFs und geht auf Wendland (2005) zurück. Positiv definite RBFs sind zum Beispiel die Gaußsche RBF, die inverse Multiquadrik (siehe Abschnitt 6.2.2) und sogenannte *Wendland-Funktionen* (Wendland, 1996). Bei letzteren handelt es sich um RBFs mit kompaktem Träger, die stückweise polynomiell sind und minimalen Grad in Abhängigkeit von Glätte und Dimension haben.
Eine Abschätzung des Glättungsfehlers kann über die Einführung sogenannter *Native Spaces* erfolgen:

Definition 6.2.6 (Native Spaces)**.** *Der* Native Space $\mathcal{N}_{\phi(\Omega)}$ *für eine gegebene positiv definite RBF* ϕ *ist auf dem Gebiet* Ω *gegeben durch:*

$$\mathcal{N}_{\phi(\Omega)} := \overline{\{\phi(||\cdot - x||), x \in \Omega\}},$$

wobei für $f_1 = \sum_{k=1}^{n_1} \alpha_k \phi(||\cdot - x_k||) \in \mathcal{N}_{\phi(\Omega)}$ *und* $f_2 = \sum_{j=1}^{n_2} \beta_j \phi(||\cdot - x_j||) \in \mathcal{N}_{\phi(\Omega)}$ *gilt:*

$$\langle f_1, f_2 \rangle_{\phi(\Omega)} := \sum_{k=1}^{n_1} \sum_{j=1}^{n_2} \alpha_k \beta_j \phi(||x_k - x_j||)$$

Dabei sind $n_1, n_2 \in \mathbb{N}$, $\alpha_k, \beta_j \in \mathbb{R}$, $k = 1, ..., n_1$, $j = 1, ..., n_2$, *und* $x_k, x_j \in \Omega$, $k = 1, ..., n_1$, $j = 1, ..., n_2$.

Der Hilbertraum $\mathcal{N}_{\phi(\Omega)}$ ist die Vervollständigung des Prä-Hilbertraums $\{\phi(||\cdot - x||), x \in \Omega\}$. Da der Approximationsfehler in Abhängigkeit von der Größe des Gebiets Ω abzuschätzen ist, wird als nächstes die sogenannte *Fülldistanz* definiert:

Definition 6.2.7 (Fülldistanz)**.** *Für eine gegebene diskrete Trainingsmenge* $\mathcal{X} \subset \Omega$ *mit Trainingsdaten* x_j, $j = 1, ..., m$, $m \in \mathbb{N}$, *ist die* Fülldistanz $\Delta_{\Omega,\mathcal{X}}$ *definiert durch*

$$\Delta_{\Omega,\mathcal{X}} := \sup_{x \in \Omega} \min_{j=1,...,m} ||x - x_j||.$$

Aus den Definitionen 6.2.6 und 6.2.7 kann der folgende Satz zur Abschätzung des Approximationsfehlers formuliert werden:

Satz 6.2.8 (Approximationsfehler im Falle positiv definiter RBFs)**.** *Es seien* Ω, $\mathcal{X}$, ϕ *und* f *wie oben. Es sei weiterhin* g *die mithilfe der RBF* ϕ *erhaltene Approximationsfunktion für* $f \in \mathcal{N}_{\phi(\Omega)}$. *Dann gilt die folgende Abschätzung für den Approximationsfehler:*

$$||f - g||_{L_\infty(\Omega)} \leq h(\Delta_{\Omega,\mathcal{X}})||f||_{\mathcal{N}_{\phi(\Omega)}},$$

wobei $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} h(\Delta_{\Omega,\mathcal{X}}) = 0$.

Beweis: Der Beweis ist in Wendland (2005) nachzulesen. □

Dieser Satz ist von entscheidender Bedeutung für die Konvergenz des DGAFO-Verfahrens, was in Abschnitt 6.4.2 deutlich wird. Für das DGAFO-Verfahren muß allerdings zusätzlich zu der Bedingung $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} h(\Delta_{\Omega,\mathcal{X}}) = 0$ noch eine weitere Voraussetzung erfüllt sein. Als nächstes wird der in dieser Arbeit entwickelte Begriff der *Gutartigkeit* einer Glättung Glättung im Sinne des DGAFO-Verfahrens definiert:

Definition 6.2.9 (Gutartige Glättung)**.** *Es seien* Ω *und* $\mathcal{X}$ *wie oben. Weiterhin sei* $f \in \mathcal{H}$, *wobei* $\mathcal{H}$ *ein Hilbertraum von Funktionen auf* Ω *ist mit Skalarprodukt* $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ *und* $||f||_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ *für* $f \in \mathcal{H}$. *Eine Glättung auf dem Gebiet* Ω *durch eine Funktion* $g \in \mathcal{P}$, *wobei* $\mathcal{P}$ *ein Prä-Hilbertraum mit demselben Skalaprodukt und* $\bar{\mathcal{P}} = \mathcal{H}$ *ist, heißt gutartig, falls*

$$||f - g||_{\mathcal{H}} \leq \kappa h(\Delta_{\Omega,\mathcal{X}}), \tag{6.44}$$

wobei $\kappa > 0$, $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} h(\Delta_{\Omega,\mathcal{X}}) = 0$ *und* $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} \frac{h(\Delta_{\Omega,\mathcal{X}})}{\Delta_{\Omega,\mathcal{X}}} = 0$.

Es muß also zusätzlich noch die Bedingung $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} \frac{h(\Delta_{\Omega,\mathcal{X}})}{\Delta_{\Omega,\mathcal{X}}} = 0$ erfüllt sein. Nach dem folgenden Korollar ist die Glättung basierend auf einer Gaußsche RBF gutartig:

Korollar 6.2.10 (Gutartigkeit der Glättung durch eine Gaußsche RBF). *Es seien $\Omega, \mathcal{X}$ und f wie in Satz 6.2.8. Es sei weiterhin $\phi(||\cdot - x||) = \exp\left(-c||\cdot - x||^2\right)$ für $x \in \Omega$ und $c \in \mathbb{R}^+$ eine Gaußsche RBF. Dann ist die Glättung von f durch $g(x) := \sum_{k=1}^{\nu} \alpha_k \phi(||x_k - x||),\ \nu \in \mathbb{N},\ \alpha_k \in \mathbb{R},\ x_k \in \Omega,\ k = 1, ..., \nu$, gutartig.*

Beweis: Gemäß Wendland (2005) gilt die Abschätzung

$$||f - g||_{L_\infty(\Omega)} \leq \exp\left(-d\left(\frac{\log(\Delta_{\Omega,\mathcal{X}})}{\Delta_{\Omega,\mathcal{X}}}\right)\right) ||f||_{\mathcal{N}_{\phi(\Omega)}},\ d > 0.$$

Mit $h(\Delta_{\Omega,\mathcal{X}}) := \exp\left(-d\left(\frac{\log(\Delta_{\Omega,\mathcal{X}})}{\Delta_{\Omega,\mathcal{X}}}\right)\right)$ gilt $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} h(\Delta_{\Omega,\mathcal{X}}) = 0$ und $\lim_{\Delta_{\Omega,\mathcal{X}} \to 0} \frac{h(\Delta_{\Omega,\mathcal{X}})}{\Delta_{\Omega,\mathcal{X}}} = 0$. Setzt man $\kappa := ||f||_{\mathcal{N}_{\phi(\Omega)}}$ und berücksichtigt man noch, daß $\mathcal{N}_{\phi(\Omega)}$ ein Hilbertraum und g Element des Prä-Hilbertraums $\{\phi(||\cdot - x||), x \in \Omega\}$ mit $\overline{\{\phi(||\cdot - x||), x \in \Omega\}} = \mathcal{N}_{\phi(\Omega)}$ ist, so resultiert, daß die Glättung basierend auf der Gaußsche RBF gemäß Definition 6.2.9 gutartig ist.

□

Aufgrund von Korollar 6.2.10 wird die Gaußsche RBF zur Glättung für das DGAFO-Verfahren aus theoretischen Gründen selektiert. Allerdings sind ähnliche Abschätzungen auch für andere positiv definite RBFs möglich. Daß die Gaußsche RBF für die vorliegende Problemstellung, auf die das DGAFO-Verfahren angewandt wird, geeignet ist, wird anhand praktischer Evaluationen in Abschnitt 7.1 gezeigt. Für eine SVM, die in dieser Arbeit auch als Glättungsmethoden in Frage kommt, sind derartige Abschätzungen nicht bekannt. Allerdings wird die endgültige Wahl des besten Glättungsverfahrens in Abschnitt 7.1 getroffen.

Das Problem bei RBFs ist jedoch der Rechenaufwand alleine zur Erstellung der Matrix H aus Gleichung (6.32): Für $N = 5$ sind bereits mehr als tausend RBF-Auswertungen und für $N = 6$ sogar über 72 Millionen durchzuführen. Daher sollte eine RBF im Falle von $N > 4$ nicht verwendet werden. Auch der Rechenaufwand im Falle einer SVM steigt stark mit der Dimension an, da die Dimension des Merkmalsraums, in dem die Daten linear separabel sind, noch stärker ansteigt, da dessen Dimensionalität, natürlich in Abhängigkeit von der verwendeten Kernfunktion, stets signifikant größer ist als die des originalen Eingaberaums. Näheres zu Merkmalsräumen und Kernfunktionen ist zum Beispiel in Hülsmann (2006) nachzulesen.
Für höherdimensionale Optimierungsprobleme wird in dieser Arbeit daher ein einfacheres Glättungsverfahren verwendet, dessen Aufwand nur quadratisch mit der Dimension steigt: Die Idee besteht darin, auf dem Dünnen Gitter nicht die Fehlerfunktion, sondern sämtliche zu optimierenden physikalischen Zielgrößen linear zu approximieren. Dieses Verfahren wird mit **Li***near* **pr***operty* **a***pproximation (Lipra)* bezeichnet. Um den Rechenaufwand klein zu halten, wird lediglich die KQ-Methode als Regularisierungsverfahren in Betracht gezogen. Dann ist auf jedem Dünnen Gitter lediglich ein LGS der Größe N zu lösen. Die zugehörigen Fehlerfunktionswerte auf dem Dünnen Gitter erhält man dann einfach durch Einsetzen der approximierten physikalischen Eigenschaften. Selbstverständlich steigt der Rechenaufwand dieser Methode auch mit

der Anzahl an betrachteten Eigenschaften, auch zu verschiedenen Temperaturen. Der Gesamtaufwand des Verfahrens liegt somit in $\mathcal{O}(n|\mathcal{T}|N^2)$.

6.3 DGAFO-Verfahren

Das DGAFO-Verfahren beschränkt sich bei der Bestimmung der Fehlerfunktionswerte auf ein Dünnes Gitter, was den Simulationsaufwand erheblich reduzieren sollte. Nach der Anwendung eines der in Abschnitt 6.2 beschriebenen Glättungsverfahren wird gemäß Abschnitt 6.1 die Fehlerfunktion mithilfe der Kombinationsmethode auf ein vollbesetztes Gitter interpoliert. Der entstehende Interpolationsfehler ist nach Abschnitt 6.4.1 tolerierbar, wohingegen der Rechenaufwand deutlich sinkt.
Die Minimierung erfolgt diskret auf dem vollen Gitter, was den Vergleich der Funktionswerte erfordert. Letzteres ist ein $\mathcal{O}\left((n+1)^{2N}\right)$-Problem, wobei $n \in \mathbb{N}$ wie in Abschnitt 6.1.4 definiert ist. Es stellt sich zunächst die Frage, ob das DGAFO-Verfahren lokal oder global zu verwenden ist. Global bedeutet in diesem Fall, daß auf dem gesamten zulässigen Gebiet für die Kraftfeldparameter sowohl geglättet als auch interpoliert und minimiert wird. In Abschnitt 6.3.1 wird die Schlußfolgerung getroffen, daß sich Glättung und Interpolation bei der vorliegenden Problemstellung nur lokal eignen, das heißt in einer kleinen Umgebung um die aktuelle Iteration. Dadurch wird das DGAFO-Verfahren zu einem iterativen Optimierungsverfahren. Der Übergang zur nächsten Iteration erfolgt dabei mittels Kombination mit der Trust-Region-Idee aus Abschnitt 3.4.3: Abschnitt 6.3.2 beschreibt den iterativen Ablauf des DGAFO-Verfahrens. Wie in Abschnitt 6.1.1 bereits motiviert, sollte die zu minimierende Funktion so modifiziert werden, daß auf dem betrachteten Gitter homogene Dirichlet-Randbedingungen erfüllt sind. Wie dies zu realisieren ist, wird in Abschnitt 6.3.3 dargestellt. Der Gesamtalgorithmus des DGAFO-Verfahrens wird in Abschnitt 6.3.4 beschrieben. Abschließend wird in Abschnitt 6.3.5 auf Komplexität und lokale Verfeinerung eingegangen. Als erster Test des DGAFO-Verfahrens wird in Abschnitt 6.3.6 die Expontentialfunktion $H(x_1, x_2) := -\exp\left(-(x_1-2)^2 - (x_2+1)^2\right)$ minimiert. Das globale Minimum dieser Funktion ist bekannt und liegt bei $(x_1^*, x_2^*) = (2, -1)$. Daher kann anhand der Testanalysen praktisch dargelegt werden, daß eine Glättung und eine lokale Betrachtung beim DGAFO-Verfahren notwendig sind.

6.3.1 Minimierung mithilfe von Dünnen Gittern: lokal oder global?

Die Kombinationsmethode aus Abschnitt 6.1.3 liefert eine stückweise multilineare Funktion $q : \mathbb{R}^N \to \mathbb{R}$, welche entweder die Fehlerfunktion F selbst oder eine geglättete Funktion G von einem Dünnen Gitter vom Level $\hat{\ell} \in \mathbb{N}$ auf ein volles Gitter $G_{\bar{\ell}}^N$ mit Level $\bar{\ell} = (\hat{\ell}, ..., \hat{\ell}) \in \mathbb{N}^N$ interpoliert. Der Gesamtfehler, der sich aus Glättungs- und Interpolationsfehler ergibt, kann durch den L_2- oder L_∞-Abstand zwischen F und dem Modell q auf dem Einheitsquadrat $\Omega := [0,1]^N$ gemessen werden:

$$||e||_{L_2,[0,1]^N} := ||F-q||_{L_2,[0,1]^N} = \left(\int_{[0,1]^N} (F(x)-q(x))^2\, dx\right)^{\frac{1}{2}} \tag{6.45}$$

$$||e||_{L_\infty,[0,1]^N} := ||F-q||_{L_\infty,[0,1]^N} = \max_{x\in[0,1]^N} |F(x)-q(x)|. \tag{6.46}$$

Sind die Funktionswerte $F(x)$ für $x \in G_\ell^N$, bekannt, so können die Fehlerterme (6.45) und (6.46) approximativ bestimmt werden.
Der Gesamtfehler ist wichtig für die Beurteilung, ob das DGAFO-Verfahren global auf das gesamte zulässige Gebiet für die Kraftfeldparameter angewandt werden kann. Dabei muß zum einen der Gesamtfehler auf einem vollen Gitter sowohl lokal als auch global ermittelt werden. Andererseits ist das Konvergenzverhalten des DGAFO-Verfahrens in beiden Fällen zu analysieren. Folgende Gründe sprechen dafür, eine lokale Betrachtung durchzuführen: Gemäß den Überlegungen in Abschnitt 6.2 ist es bei mit Rauschen behafteten Funktionswerten stets vonnöten, eine Glättung vorzuschalten. Diese Glättung würde im globalen Fall auf einem Dünnen Gitter auf dem gesamten zulässigen Gebiet erfolgen. Da die zu minimierende Fehlerfunktion beliebig komplex und zerklüftet sein kann, wird es unmöglich sein, sie auf einem großen Gebiet mit wenig Datenpunkten durch eine Glättung akkurat wiederzugeben. Außerdem wird der Diskretisierungsfehler auf dem großen Gebiet zu hoch sein, um das globale Minimum exakt bestimmen zu können. Man könnte zwar eine globale Betrachtung vorschalten, um möglichst schnell in die Nähe des globalen Minimums zu gelangen, und anschließend lokale Verfeinerungen durchführen, allerdings kann eine inakkurate Glättung dazu führen, daß man sich unter Umständen noch weiter vom Minimum entfernt. Somit sollten, wie bei den gradientenbasierten Verfahren, geeignete Startparameter definiert und eine lokale Betrachtung verwendet werden, bei der sowohl akkurat geglättet als auch der Diskretisierungsfehler klein gehalten werden kann. In Abschnitt 6.3.6 sind die Ergebnisse einer Testanwendung des DGAFO-Verfahrens angegeben, bei der zum einen lokale und globale Gesamtfehler verglichen werden und zum anderen eine Konvergenzanalyse in beiden Fällen durchgeführt wird. In den folgenden Abschnitten wird das lokale (iterative) DGAFO-Verfahren zunächst theoretisch eingeführt. Die Ergebnisse werden anschließend vorgestellt. Es sei hier jedoch bereits eine Zusammenfassung der Ergebnisse aus Abschnitt 6.3.6 in Bezug auf die Entscheidung, ob das DGAFO-Verfahren lokal oder global angewandt werden sollte, gegeben:

- **Vergleich der Gesamtfehler:**
  - Die Gesamtfehler sind sowohl in der L_2- als auch in der L_∞-Norm im globalen Fall deutlich höher als im lokalen Fall.
  - Der Glättungsfehler ist im globalen Fall wesentlich größer als im lokalen Fall, das heißt, die Funktion kann durch die Glättung nicht akkurat wiedergegeben werden. Dies ist auch der Fall, wenn die Funktion nicht mit künstlichem Rauschen behaftet ist und trotzdem eine Glättung durchgeführt wird.
- **Konvergenzanalyse:**
  - Ohne Einführen von künstlichem Rauschen führt das globale DGAFO-Verfahren nur dann zum Minimum, wenn das Minimum in einem Gitterpunkt liegt. Ansonsten kann das Minimum nur im Bereich des Diskretisierungsfehlers bestimmt werden.
  - Bei mit Rauschen behafteten Funktionswerten haben sowohl der Diskretisierungsfehler als auch der Glättungsfehler einen negativen Effekt auf die Konvergenz. Das resultierende Minimum kann weit vom tatsächlichen Minimum entfernt liegen.

Die hier getroffenen theoretischen Überlegungen deuten bereits darauf hin, daß ein lokales, iteratives DGAFO-Verfahren besser geeignet sein wird als ein globales. Ersteres wird daher in den folgenden Abschnitten vorgestellt.

6.3.2 Kombination mit dem Trust-Region-Ansatz

Innerhalb des DGAFO-Verfahrens wird für jede Iteration eine kompakte Umgebung bestimmt, auf der das Minimierungsproblem mithilfe einer Dünn-Gitter-Interpolation diskret gelöst wird. In den meisten Fällen wird das Minimum am Rand des kompakten Gebiets liegen. Falls bestimmte Voraussetzungen erfüllt sind, wird das am Rand liegende Minimum zur neuen Iteration und zum Zentrum einer neuen kompakten Umgebung.
Analog zur Trust-Region-Idee aus Abschnitt 3.4.3 ist die kompakte Umgebung ein Vertrauensgebiet der Größe $\Delta_k > 0$; aufgrund der Dünn-Gitter-Interpolation handelt es sich hierbei jedoch nicht um eine Kugel $B_{\Delta_k}(x^k)$, sondern um einen Hyperwürfel $\mathcal{W}^k$ der Form

$$\mathcal{W}^k := \bigotimes_{i=1}^{N} \left[x_i^k - \Delta_k, x_i^k + \Delta_k \right], \tag{6.47}$$

wobei x^k die k-te Iteration des DGAFO-Verfahrens ist. Die Größe Δ_k muß so klein gewählt werden, daß $\mathcal{W}^k$ zum einen innerhalb des zulässigen Gebiets für die Kraftfeldparameter liegt und zum anderen das durch die Interpolation erhaltene Modell möglichst gut mit der Originalfunktion F übereinstimmt. Letzteres wurde bereits in Abschnitt 6.3.1 diskutiert. Andererseits muß Δ_k so groß gewählt werden, daß das Verfahren möglichst schnell gegen das globale Minimum des zulässigen Gebiets konvergiert. Das heißt, daß die einzelnen nicht-disjunkten Hyperwürfel, die nach und nach das zulässige Gebiet abdecken, nicht zu klein sind, damit die Unterteilung nicht zu fein ist.
Analog zu Gleichung (3.38) wird die Güte des Modells q mithilfe des folgenden Verhältnisses geschätzt:

$$r^k := \frac{F(x^k) - F(x^*)}{q(x^k) - q(x^*)} = \frac{F(x^k) - F(x^*)}{G(x^k) - q(x^*)}, \tag{6.48}$$

wobei x^* das diskrete Minimum auf einem vollen Gitter auf dem Hyperwürfel $\mathcal{W}^k$ ist. Die Funktion G sei die auf dem Dünnen Gitter geglättete Funktion gemäß Abschnitt 6.2. Da x^k ein Punkt des Dünnen Gitters ist und das Modell q die geglättete Funktion G von dem Dünnen Gitter auf das volle Gitter interpoliert, gilt $q(x^k) = G(x^k)$.
Es werden Schwellenwerte $0 < \eta_1 < \eta_2$ und Größenparameter $0 < \gamma_1 < 1 < \gamma_2$ eingeführt und die folgenden vier Fälle unterschieden:

- $r^k > \eta_1$: Dann stimmt das Modell gut mit F überein, und das Minimum x^* wird als neue Iteration gewählt: $x^{k+1} := x^*$.

- $r^k > \eta_2 > \eta_1$: Dann wird ebenfalls $x^{k+1} := x^*$ gesetzt, und gleichzeitig wird das Vertrauensgebiet vergrößert, um die Konvergenz zu beschleunigen: $\Delta^{k+1} := \gamma_2 \Delta_k$.

- $r^k < \eta_1$: Dann stimmt das Modell nicht gut mit F überein, und es muß ein besseres Modell bestimmt werden, indem das Vertrauensgebiet verkleinert wird: $\Delta^{k+1} := \gamma_1 \Delta_k$.

Analog zum Trust-Region-Verfahren wird *a priori* ein minimales $\Delta^{\min} > 0$ festgesetzt. Das Verfahren wird abgebrochen, sobald $\exists_k \; \Delta_k < \Delta^{\min}$.

6.3.3 Behandlung von Randwerten

Nach einer Transformation $\xi : \mathcal{W}^k \to [0,1]^N$ in den Einheitswürfel, welche in Abschnitt 6.3.4 definiert wird, werden die Fehlerfunktionswerte am Rand des Einheitswürfels durch eine geeignete Modifikation gleich 0 gesetzt. Der Vorteil der Einhaltung von homogenen Dirichlet-Randbedinungen liegt in der Tatsache, daß auch Dünne Gitter am Rand relativ dicht besetzt sind. Um einen Großteil an Simulationen einzusparen, wird die Originalfunktion F mit einem Produkt aus Sinusfunktionen multipliziert, um homogene Dirichlet-Randbedingungen zu erhalten: Es wird also die modifizierte Funktion

$$\begin{aligned} \bar{F} : [0,1]^N &\to \mathbb{R}_0^+ \\ y &\mapsto \left(\prod_{i=1}^{N} \sin \pi y_i \right) F(y) \end{aligned} \tag{6.49}$$

anstelle von F betrachtet. Dabei ist $y = \xi(x)$ mit $x \in \mathcal{W}^k$. Glättung und Interpolation werden somit auf $\bar{F} \circ \xi : \mathcal{W}^k \to \mathbb{R}_0^+$ angewandt. Aufgrund von Gleichung (6.49) gilt $\bar{F}|_{\partial[0,1]^N} = 0$. Da jedoch F und nicht $\bar{F}$ minimiert werden soll und F und $\bar{F}$ nicht dasselbe Minimum besitzen, ist vor der diskreten Minimierung die Rücktransformation

$$F(x) = F(\xi^{-1}(y)) = \frac{\bar{F}(y)}{\prod_{i=1}^{N} \sin \pi y_i} \tag{6.50}$$

anzuwenden. Es ist also das Minimum von $F \circ \xi^{-1}$ zu bestimmen. Dies ist natürlich auf dem Rand von $\mathcal{W}^k$ nicht möglich, allerdings ist stets zu erwarten, daß das Minimum am Rand angenommen wird. Daher wird für die Minimierung lediglich das Gitter

$$\tilde{G}_{\hat{\ell}}^N := G_{\hat{\ell}}^N \setminus G_{\hat{\ell}}^N \cap \{0,1\}^N \tag{6.51}$$

betrachtet, welches keinen Randpunkt mehr enthält. Dies reduziert zwar die Größe des Vertrauensgebiets, allerdings ist der Verlust an Konvergenzgeschwindigkeit im Gegensatz zu dem Gewinn, den man durch das Einsparen von Simulationen erzielt, verschwindend gering: Für $N = 4$ und $\hat{\ell} = 2$ sind nur noch 9 anstelle von 393 Simulationen (vergleiche Tabelle 6.1) pro Iteration durchzuführen.

6.3.4 DGAFO-Algorithmus

Der Algorithmus innerhalb des DGAFO-Verfahrens hat die folgende Struktur:

1. **Initialisierung:** Wähle Startvektor x^0 und Startschrittweite $D^0 > 0$, so daß

   $$\forall_{i=1,\ldots,N}\ c_i^0 \le x_i^0 \le C_i^0,\ \Delta_0 < C_i^0 - x_i^0,\ \Delta_0 < x_i^0 - c_i^0.$$

   Dabei ist $[c_i^0, C_i^0]$ das zulässige Intervall für den i-ten Kraftfeldparameter. Die maximal zulässige Schrittweite $\Delta^{\max}$ wird zu Beginn errechnet und Δ_0 sowie eine minimale Schrittweite $\Delta^{\min}$ relativ dazu gesetzt. Es ist zu beachten, daß Δ_0 einerseits nicht zu klein gewählt werden darf, damit die Daten zum einen nicht durch das Rauschen ununterscheidbar werden und zum anderen die Konvergenz nicht zu langsam sein darf. Andererseits darf es

auch nicht zu groß gewählt, da man ansonsten auf die in Abschnitt 6.3.1 angesprochenen Probleme stößt.
Setze $k := 0$.

2. **Transformation:** Durch die Transformation

$$\begin{aligned} \xi : \mathcal{W}^k &\rightarrow [0,1]^N \\ x &\mapsto \frac{x - \left(\min_{i=1,...,N} x_i^k\right) \cdot e}{\left(\max_{i=1,...,N} x_i^k\right) \cdot e - \left(\min_{i=1,...,N} x_i^k\right) \cdot e} \\ &= \frac{1}{2\Delta_k}\left(x - x^k + \Delta_k \cdot e\right) \end{aligned} \tag{6.52}$$

wird der Startvektor x^0 vom Hyperwürfel der Größe Δ_0 in das Einheitsquadrat abgebildet. Dabei ist $e = (1, ..., 1)^T \in \mathbb{R}^N$.
Es ist zu beachten, daß nur die Rücktransformation über die Umkehrfunktion

$$\begin{aligned} \xi^{-1} : [0,1]^N &\rightarrow \mathcal{W}^k \\ y &\mapsto 2\Delta_k y + x^k - \Delta_k \cdot e. \end{aligned} \tag{6.53}$$

eine Rolle spielt, denn es wird zunächst ein Dünnes Gitter in $[0,1]^N$ bestimmt. Anschließend werden die Gitterpunkte in den Raum der Kraftfeldparameter durch ξ^{-1} rücktransformiert, um Molekulare Simulationen ausführen zu können.

3. **Dünnes Gitter:** Bestimme ein Dünnes Gitter $\hat{G}^N_{\hat{\ell}}$ auf $[0,1]^N = \xi\left(\mathcal{W}^k\right)$. Es ist zu beachten, daß die Transformation ξ dafür nicht explizit anzuwenden ist. Da die Kombinationsmethode aus Abschnitt 6.1.3 verwendet wird, werden die Gitterpunkte des Dünnen Gitters hierarchisch aus den für die Kombinationsmethode notwendigen Teilgittern (siehe Gleichung (6.12)) bestimmt. Es sei $y_{\ell,j}$ ein Punkt des Teilgitters vom Level $\ell = (\ell_1, ..., \ell_N)$ und Position $j = (j_1, ..., j_N)$, das heißt gleichzeitig ein Punkt des Dünnen Gitters, der jedoch kein Randpunkt ist. Für jeden dieser Punkte ist der Fehlerfunktionswert $F(y_{\ell,j})$ zu bestimmen.

4. **Randwerte:** Durch die Multiplikation von F mit einem Sinusterm,

$$\begin{aligned} \bar{F} : [0,1]^N &\rightarrow \mathbb{R}_0^+ \\ y &\mapsto \left(\prod_{i=1}^{N} \sin \pi y_i\right) F(y), \end{aligned}$$

werden homogene Dirichlet-Randbedingungen realisiert (siehe Abschnitt 6.3.3), um den Rechenaufwand maßgeblich zu reduzieren. Die Funktion $\bar{F}$ wird auf jeden Punkt $y_{\ell,j}$ des Dünnen Gitters angewandt.

5. **Glättung:** Falls die zu minimierende Funktion mit statistischem Rauschen behaftet ist, so ist aus den in Abschnitt 6.2.1 genannten Gründen die Funktion $\bar{F}$ auf geeignete Weise zu glätten. Dazu stehen die Glättungsmethoden aus Abschnitt 6.2.2 und Regularisierungsverfahren aus Abschnitt 6.2.3 zur Verfügung. Man erhält somit für jeden Punkt $y_{\ell,j}$ des Dünnen Gitters einen geglätteten Funktionswert $G(y_{\ell,j})$. Ist kein Rauschen auf den Daten vorhanden, so gilt $G \equiv \bar{F}$.

6. **Interpolation:** Mithilfe der Kombinationsmethode aus Abschnitt 6.1.3 und der in Abschnitt 6.1.4 beschriebenen multilinearen Interpolation wird die Funktion G von dem Dünnen Gitter $\hat{G}_{\hat{\ell}}^N$ auf das volle Gitter $G_{\bar{\ell}}^N$ interpoliert. Man erhält somit ein Interpolationsmodell $\bar{q}$ für jeden Punkt $y \in G_{\bar{\ell}}^N$.
Da die Interpolation auf $\bar{F}$ angewandt wurde, ist für jeden Punkt, der nicht Randpunkt des vollen Gitters ist, die folgende Division durchzuführen:

$$\forall_{y \in \tilde{G}_{\bar{\ell}}^N} \; q(y) = \frac{\bar{q}(y)}{\prod_{i=1}^N \sin \pi y_i}.$$

Das Interpolationsmodell q ist somit nur für alle $y \in \tilde{G}_{\bar{\ell}}^N$ gültig.

7. **Diskrete Minimierung:** Bestimme

$$y^* := \arg \min_{y \in \tilde{G}_{\bar{\ell}}^N} \; q(y).$$

8. **Iterationsschritt $x^k \to x^{k+1}$:** Bestimme das Verhältnis

$$r^k := \frac{F(y^k) - F(y^*)}{q(y^k) - q(y^*)} = \frac{F(y^k) - F(y^*)}{G(y^k) - q(y^*)}, \; y^k = \xi(x^k),$$

und unterscheide die folgenden Fälle:

- $r^k \geq \eta_1 \Rightarrow x^{k+1} := \xi^{-1}(y^*) \wedge \Delta_{k+1} := \Delta_k$.
- $r^k \geq \eta_2 > \eta_1 \Rightarrow x^{k+1} := \xi^{-1}(y^*) \wedge \Delta_{k+1} := \gamma_2 \Delta_k$
- $r^k < \eta_1 \Rightarrow x^{k+1} := x^k \wedge \Delta_{k+1} := \gamma_1 \Delta_k$.

Dabei sind $\eta_2 > \eta_1 > 0$ als Schwellenwerte und $\gamma_2 > 1 > \gamma_1 > 0$ als multiplikative Faktoren globale Parameter.
Setze $k := k + 1$, und gehe zu Schritt 2.

9. **Abbruchkriterien:** Mit derselben Argumentation wie in Abschnitt 3.5.3 wird auch hier als Abbruchkriterium

$$F(x^*) \leq \tau$$

für ein $\tau > 0$ festgelegt. Allerdings ist es beim DGAFO-Verfahren zusätzlich notwendig, daß das Minimum im Innern des Hyperwürfels liegt. Insgesamt werden die folgenden drei Abbruchkriterien unterschieden:

(i) $F(x^*) \leq \tau \wedge \xi(x^*) \notin U_0 \cup U_1$
(ii) $F(x^*) \leq \tau \wedge \xi(x^*) \notin U_0 \cup U_1 \wedge \Delta^* < \Delta^{\min}$
(iii) $\exists_{k \in \mathbb{N}} \; r^k < \eta_1 \wedge \Delta_k < \Delta^{\min}$.

Dabei gilt:

$$\begin{aligned} U_0 &:= \{y \in [0,1]^N | \; \exists_{i \in 1,\ldots,N} \; y_i \in \{0,1\}\}, \\ U_1 &:= \{y \in [0,1]^N | \; \exists_{i \in 1,\ldots,N} \; y_i \in \{2^{-\hat{\ell}}, 1 - 2^{-\hat{\ell}}\}\}. \end{aligned}$$

Weiterhin ist $\Delta^* := \Delta_k$, wobei $x^* = x^k$.
Die Abbruchkriterien (i) und (ii) stehen für eine erfolgreiche Konvergenz des DGAFO-Verfahrens: Das Hauptkriterium ist erfüllt, und das Minimum wird im Innern des Hyperwürfels angenommen. Dadurch, daß $\xi(x^*) \notin U_1$, wird ebenfalls ausgeschlossen, daß es am Rand des Gitters angenommen wird, auf dem das Interpolationsmodell q gültig ist. Abbruchkriterium (ii) enthält die Zusatzbedingung $\Delta^* < \Delta^{\min}$, welche ausschließt, daß durch lokale Verfeinerungen noch Verbesserungen erzielt werden können, worauf in Abschnitt 6.3.5 noch näher eingegangen wird. Aufgrund dessen ist dieses Abbruchkriterium als ideal anzusehen.
Abbruchkriterium (iii) besagt, daß das DGAFO-Verfahren nicht zum Erfolg geführt hat. Es bedeutet, daß selbst durch Verkleinerung des Vertrauensgebiets kein Modell q mehr gefunden werden kann, welches F akkurat wiedergibt. Dies ist insbesondere dann der Fall, wenn die Anforderungen für die Anwendung der Kombinationsmethode (siehe Abschnitt 6.4.1) nicht erfüllt sind, was beispielsweise auf eine inakkurate Glättung zurückzuführen ist, bei der das Rauschen nicht ausreichend gut herausgefiltert worden ist.

6.3.5 Komplexität und lokale Verfeinerungen

In diesem Abschnitt wird die Komplexität des DGAFO-Verfahrens angesprochen, auch im Hinblick auf lokale Verfeinerungen, die zum Erhalt eines möglichst exakten Minimums führen sollen. Sämtliche in Abschnitt 6.3.4 angesprochenen Teilschritte des DGAFO-Algorithmus werden hier nochmals bezüglich Komplexität diskutiert:

1. **Initialisierung:** Der Aufwand hierbei besteht lediglich in der Allokation der initialen Variablen wie Startparameter, obere und untere Schranken sowie maximale, initiale und minimale Schrittweite.

2. **Transformation:** Die Transformation geschieht an dieser Stelle nur imaginär. Man beginnt direkt mit der Konstruktion eines Dünnen Gitters in Punkt 3. Lediglich die Ausführung der Rücktransformation ξ^{-1} liefert einen geringen Rechenaufwand.

3. **Dünnes Gitter:** Der Rechenaufwand zum Erhalt eines Dünnen Gitters beträgt gemäß Abschnitt 6.1 $\mathcal{O}\left(2^{\hat{\ell}} \cdot \hat{\ell}^{N-1}\right)$, wobei $\hat{\ell}$ das Level des Dünnen Gitters ist. Dieser Aufwand ist zwar immer noch exponentiell in der Dimension, allerdings stellt er gerade bei höheren Dimensionen einen erheblichen Vorteil gegenüber dem Aufwand auf vollen Gitter dar (vergleiche Tabelle 6.1). Dieser Aufwand betrifft nicht nur die Anzahl an zu berechnenden Gitterpunkten, sondern vor allem die Anzahl an Funktionsauswertungen, sprich Simulationen. Eine Minimierung dieses Aufwands ist, wie bereits mehrfach diskutiert, von enormer Wichtigkeit.

4. **Randwerte:** Die Funktionswerte am Rand eines Dünnen Gitters werden einfach auf 0 gesetzt, was somit keinen Rechenaufwand darstellt. Für alle anderen Gitterpunkte sind gemäß Gleichung (6.49) $2N + 1$, also $\mathcal{O}(N)$ Multiplikationen durchzuführen. Hinzu kommen N Sinusauswertungen. Die Anzahl an Randpunkten eines N-dimensionalen Dünnen Gitters von Level $\hat{\ell}$ beträgt $\mathcal{O}\left(N2^{N-1}2^{\hat{\ell}}\right)$. Dabei ist $N2^{N-1}$ die Anzahl Kanten eines N-dimensionalen Hyperwürfels. Die Anzahl an durchzuführenden Simulationen wird somit

auf

$$\mathcal{O}\left[2^{\hat{\ell}}\left(\hat{\ell}^{N-1} - N2^{N-1}\right)\right]$$

reduziert. Diese Anzahl multipliziert mit $2N+1$ ist die Anzahl an notwendigen Multiplikationen und multipliziert mit N die Anzahl an auszuführenden Sinusauswertungen. Die Reduktion von Molekularen Simulationen geht somit auf Kosten von

$$\mathcal{O}\left[N2^{\hat{\ell}}\left(\hat{\ell}^{N-1} - N2^{N-1}\right)\right]$$

Multiplikationen und Sinusauswertungen, ein Rechenaufwand, der in jedem Fall vernachlässigbar ist, wenn man die enorme Rechenzeit einer Simulation betrachtet.

5. **Glättung:** Bei den meisten Glättungsverfahren ist eine multivariate lineare Regression durchzuführen, was einen Rechenaufwand von $\mathcal{O}(mM^2 + M^3)$ bedeutet, wobei m die Anzahl an Datenpunkten und M die Anzahl an zugrundegelegten Basisfunktionen ist. Bei RBFs beispielsweise gilt $M = K$ (siehe Abschnitt 6.2.2). Dieser Aufwand kann oftmals durch geschickte numerische Rechenwege, zum Beispiel im Falle positiver Definitheit durch eine Cholesky-Zerlegung, auf $\mathcal{O}(mM^2 + M^2)$ oder in manchen Fällen sogar auf $\mathcal{O}(1)$ gebracht werden. Der für die Glättung benötigte Aufwand stellt somit im Vergleich zu den auszuführenden Simulationen kein Problem dar, allerdings ist zu beachten, daß m und M einerseits groß genug sein müssen, um möglichst genau glätten zu können. Andererseits müssen sie klein genug sein, um den Rechenaufwand zu reduzieren und Überangepaßtheit des Modells zu vermeiden. Allerdings tritt bei jeder der in Abschnitt 6.2 angegebenen Glättungsmethoden zusätzlicher Rechenaufwand auf:

   - Bei einer Support-Vector-Machine muß stets die Lagrange-Theorie angewandt werden, um ein nichtlineares Optimierungsproblem mit Nebenbedingungen zu lösen. Dies macht eine SVM äußerst aufwendig, was jedoch dadurch ausgeglichen wird, daß die Auswertung der resultierenden Modellfunktion nur durch eine Summation über die Stützvektoren erfolgt.
   - Bei Radialen Basisfunktionen ist für die Wahl der Zentroide ein *k-means*-Algorithmus durchzuführen, dessen Konvergenzgeschwindigkeit stets von der zufälligen Wahl der Startzentroide abhängig ist. Die Auswertung der Modellfunktion erfolgt jedoch lediglich durch eine Summation über die Zentroide.
   - Beim MARS-Verfahren führt die spezifische Wahl der Knoten und Schwellenwerte sowie das anschließende *Pruning* zu erheblichem Mehraufwand.

   Zusätzlicher Rechenaufwand entsteht durch den Einsatz von Regularisierungsverfahren, insbesondere aufgrund der Verwendung von Newton-Lagrange-Verfahren zum Erhalt der Nebenbedingungen sowie der Parametrisierung.

6. **Interpolation:** Bei der in Abschnitt 6.1.4 beschriebenen multilinearen Interpolation gehen ebenfalls für jeden Gitterpunkt seine benachbarten Gitterpunkte mit ein. Somit ist die Interpolation auch ein $\mathcal{O}\left[(2N+1)\cdot 2^{\hat{\ell}}\cdot\hat{\ell}^{N-1}\right]$-Problem. Bei jeder einzelnen linearen Interpolation sind jeweils zwei Multiplikationen, zwei Subtraktionen, eine Addition und eine Division durchzuführen.
   Weiterhin ist nach der multilinearen Interpolation für jeden Punkt im Innern des Einheitsquadrats eine Division durch den Sinusterm aus Punkt 4 durchzuführen, welcher jedoch

bereits für jeden Punkt des Dünnen Gitters berechnet worden ist. Somit sind nur noch

$$\mathcal{O}\left[(2N+1)\left((2^{\hat{\ell}}-1)^N - 2^{\hat{\ell}}\hat{\ell}^{N-1}\right)\right]$$

Multiplikationen und

$$\mathcal{O}\left[N\left((2^{\hat{\ell}}-1)^N - 2^{\hat{\ell}}\hat{\ell}^{N-1}\right)\right]$$

Sinusauswertungen durchzuführen. Die Anzahl an durchzuführenden Divisionen beträgt $(2^{\hat{\ell}}-1)^N$. Es ist zu beachten, daß die Divisionen auch für jeden inneren Punkt des Dünnen Gitters durchzuführen sind, da dort nur die geglättete Funktion G mit dem Interpolationsmodell übereinstimmen, nicht jedoch $\bar{F}$.

7. **Diskrete Minimierung:** Für die Minimierung von q wird ein maximaler Funktionswert von 10^6 angenommen. Für jeden weiteren Funktionswert auf dem vollen Gitter ohne Rand ist ein Größenvergleich durchzuführen, also insgesamt $(2^{\hat{\ell}}-1)^N$ Größenvergleiche.
8. **Iterationsschritt:** Zur Realisierung eines Iterationsschritts sind nur einige geringe Operationen notwendig. Es handelt sich lediglich um einfache Subtraktionen, Divisionen, Multiplikationen und Größenvergleiche.
9. **Abbruchkriterien:** Bei der Feststellung der Abbruchkriterien handelt es sich lediglich um Größenvergleiche. Bei der Bestimmung, ob ein Vektor in $U_0 \cup U_1$ liegt oder nicht, werden die Komponenten von y mit den Werten 0, 1, h und $1-h$ verglichen. Stimmt eine Komponente mit einem dieser Werte überein, wird abgebrochen und $y \in U_0 \cup U_1$ festgestellt.

Auch beim DGAFO-Verfahren gilt es, möglichst viele Molekulare Simulationen einzusparen, was vor allem bei hohen Dimensionen und kleinem Level durch die Verwendung Dünner Gitter erfolgt. Eine Erhöhung von $\hat{\ell}$ macht nicht nur aufgrund der damit verbundenen Erhöhung des Rechenaufwands keinen Sinn. Durch das in den Simulationsdaten vorhandene statistische Rauschen dürfen die Gitterpunkte nicht zu nah beieinanderliegen, da ihre Funktionswerte ansonsten nicht mehr unterscheidbar sind. Dies führt dazu, daß das DGAFO-Verfahren ab einem bestimmten Level nicht mehr zum Ziel führt: Abbildung 6.11 zeigt, zu welcher Minimalstelle das DGAFO-Verfahren bei Verwendung der Funktion $H(x_1, x_2) := -\exp\left(-(x_1-2)^2 - (x_2+1)^2\right)$ aus Abschnitt 6.3.6 mit künstlichem Rauschen bei verschiedenem Level im Durchschnitt über 10 Zufallsreplikate gelangt. Man sieht einen deutlichen Anstieg der Kurve ab $\hat{\ell} = 4$. Daher wird $\hat{\ell} \in \{2, 3\}$ empfohlen (siehe auch Abschnitt 6.3.6).
Die Abbruchkriterien (i) und (ii) zeigen eine erfolgreiche Konvergenz des DGAFO-Verfahrens an. Allerdings schließt Abbruchkriterium (i) keine Fehler aufgrund zu grober Diskretisierung aus, denn ein weiteres Ziel des DGAFO-Verfahrens besteht darin, näher ans Minimum zu gelangen als die hier betrachteten gradientenbasierten Verfahren. Um dies zu erreichen, sind *lokale Verfeinerungen* vonnöten. Befindet sich das Minimum innerhalb des Vertrauensgebiets, so soll es möglich sein, es mithilfe des Dünn-Gitter-Interpolationsmodells möglichst genau zu bestimmen. Da es aufgrund des statistischen Rauschens niemals exakt bestimmt werden kann und die Gitterpunkte nicht zu nah beieinanderliegen dürfen, sollte das DGAFO-Verfahren enden, falls $\Delta^k < \Delta^{\min}$. Somit ist die Erfüllung von

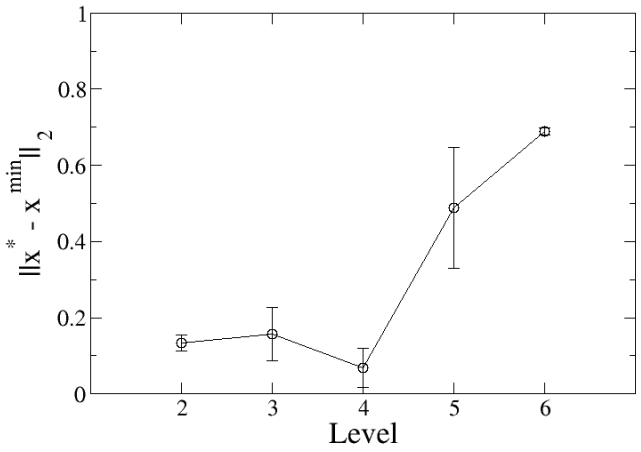


Abbildung 6.11: Abhängigkeit der Konvergenz des DGAFO-Verfahrens vom Dünn-Gitter-Level im Falle von statistischem Rauschen. Aufgetragen sind die euklidischen Abstände zwischen der vom DGAFO-Verfahren erreichten Minimalstelle x^* und der tatsächlichen Minimalstelle $x^{\min}$, gemittelt über 10 Zufallsreplikate. Die zugehörigen Standardabweichungen werden durch die Fehlerbalken dargestellt. Man sieht, daß das Verfahren umso schlechter konvergieren kann, je höher das Level ist, das heißt je näher die Punkte beieinanderliegen. Minimiert wurde die Funktion $H(x_1, x_2) := -\exp\left(-(x_1-2)^2-(x_2+1)^2\right)$ aus Abschnitt 6.3.6. Geglättet wurde mithilfe einer SVM.

Abbruchkriterium (ii) das oberste Ziel des Verfahrens. Allerdings ist hierbei zu beachten, daß die Parameter $\Delta^{\min}, \gamma_1$ und η_1 so eingestellt werden müssen, daß Abbruchkriterium (ii) ohne einen zusätzlichen hohen Rechenaufwand erfüllt werden kann. Dies hängt auch von der Wahl des Glättungsverfahrens ab. Daher wird die Konvergenz des DGAFO-Verfahrens in Abhängigkeit von verschiedenen Einstellungen in Abschnitt 7.2 anhand der Korrelationsfunktionen systematisch evaluiert.

6.3.6 Testanwendung: Praktische Rechtfertigung der Glättungs- und Iterationsnotwendigkeit

Als erster Test des DGAFO-Verfahrens wird in diesem Abschnitt die Funktion

$$H(x_1, x_2) := -\exp\left(-(x_1-2)^2-(x_2+1)^2\right)$$

betrachtet. Diese Funktion besitzt ein globales Minimum bei $(x_1^*, x_2^*) := (2, -1)$ mit Funktionswert $H(2,-1) = -1$. Tabelle 6.2 zeigt den Verlauf des DGAFO-Verfahrens bei verschiedenen Dünn-Gitter-Leveln $\hat{\ell}$: Je höher das Level, desto weniger Iterationen sind zur Konvergenz notwendig. Allerdings steigt mit dem Level auch die Anzahl an Funktionsauswertungen, welche für $\hat{\ell} = 2$ gleich 54, für $\hat{\ell} = 3$ gleich 72 und für $\hat{\ell} = 4$ gleich 100 waren. Das globale Minimum wurde stets genau erreicht, da $(x_1^*, x_2^*) = (2, -1)$ ein Punkt des Dünnen Gitters im letzten Vertrauensgebiet war. Auch die Minima in den anderen Vertrauensgebieten lagen an Dünn-Gitter-Punkten, so daß $\forall_k\ r^k = 1$ gilt und Δ im Laufe des Algorithmus nicht verändert wird. Wurden auf die Funktionswerte 3% künstliches Rauschen eingeführt, so waren im Falle von

Level	Iteration	(x_1^k, x_2^k)	$H(x_1^k, x_2^k)$	r^k	Δ^k
$\hat{\ell} = 2$	0	(2.5,-1.5)	-0.6065	1.0	0.25
	1	(2.375,-1.5)	-0.6766	1.0	0.25
	2	(2.375,-1.375)	-0.7548	1.0	0.25
	3	(2.25,-1.375)	-0.8162	1.0	0.25
	4	(2.25,-1.25)	-0.8829	1.0	0.25
	5	(2.125,-1.25)	-0.9248	1.0	0.25
	6	(2.125,-1.125)	-0.9692	1.0	0.25
	7	(2,-1.125)	-0.9845	1.0	0.25
	8	(2,-1)	-1	1.0	0.25
$\hat{\ell} = 3$	0	(2.5,-1.5)	-0.6065	1.0	0.25
	1	(2.375,-1.375)	-0.7548	1.0	0.25
	2	(2.25,-1.25)	-0.8825	1.0	0.25
	3	(2.125,-1.125)	-0.9692	1.0	0.25
	4	(2,-1)	-1	1.0	0.25
$\hat{\ell} = 4$	0	(2.5,-1.5)	-0.6065	1.0	0.25
	1	(2.28125,-1.28125)	-0.8537	1.0	0.25
	2	(2,-1)	-1	1.0	0.25

Tabelle 6.2: Einfache Anwendung des DGAFO-Verfahrens: Es wurde mit verschiedenen Dünn-Gitter-Leveln die Funktion $H(x_1, x_2) = -\exp\left(-(x_1 - 2)^2 - (x_2 + 1)^2\right)$ minimiert. Das globale Minimum liegt bei $\left(x_1^*, x_2^*\right) = (2, -1)$ mit Funktionswert $H(2, -1) = -1$. Der Startvektor war stets $\left(x_1^0, x_2^0\right) = (2.5, -1.5)$. Das DGAFO-Verfahren konvergierte umso schneller, je höher das Level des Dünnen Gitters war. Es galt stets $r^k = 1$, da das Minimum von q im Vertrauensgebiet stets an einem Punkt des Dünnen Gitters angenommen wurde. Daher wurde auch Δ nicht verändert. Bei den drei angegebenen Leveln waren jeweils 54, 72 beziehungsweise 100 Funktionsauswertungen erforderlich.

$\hat{\ell} = 2$ und $\hat{\ell} = 3$ weniger Iterationen und Funktionsauswertungen notwendig. In allen Fällen stieg jedoch auch der Modellierungsfehler, so daß das Minimum nur angenähert werden konnte. Bei $\hat{\ell} = 2$ kam hinzu, daß oftmals auch Abbruchkriterium (iii) erfüllt war. Letzteres änderte sich, nachdem eine Glättung mithilfe einer SVM vorgeschaltet wurde. Die Verwendung einer SVM machte das DGAFO-Verfahren deutlich robuster, und falls $x^{\min} := (2, -1)$ und x^* das vom DGAFO-Verfahren berechnete approximative Minimum ist, so wurde $||x^* - x^{\min}||_2$ kleiner als ohne Glättung, das heißt, die Approximation und somit auch die Konvergenz wurden durch die Glättung verbessert.
Auch wenn $\hat{\ell} = 3$ im Falle von Rauschen bessere Resultate als $\hat{\ell} = 2$ lieferte, empfiehlt es sich dennoch, insbesondere zu Beginn des Verfahrens $\hat{\ell} = 2$ zu verwenden, um mit möglichst wenig Funktionsauswertungen in die Nähe des Minimums zu gelangen. Gerade bei höheren Dimensionen ist dann der Rechenaufwand deutlich geringer. Weitere Analysen (siehe Abschnitt 7.2) haben ergeben, daß beispielsweise für $N = 4$ die Wahl $\hat{\ell} = 2$ gut geeignet ist. In der Nähe des Minimums kann gegebenenfalls $\hat{\ell} = 3$ gewählt oder Δ verkleinert werden. Höhere Level sollten nicht nur aufgrund des steigenden Rechenaufwands nicht verwendet werden: Aus Abbildung 6.11 geht deutlich hervor, daß bei Vorhandensein von statistischem Rauschen mit steigendem Level die Konvergenz des DGAFO-Verfahrens immer schlechter wird. Liegen die Gitterpunkte zu nahe beieinander, so können die zugehörigen Funktionswerte aufgrund des Rauschens nicht mehr unterschieden werden, und das durch die SVM beziehungsweise durch die Dünn-Gitter-Interpolation erstellte Modell kann nur noch durch Zufall bessere Resultate liefern. Interessanterweise war der Durchmesser des Fehlerbalkens im Falle von $\hat{\ell} = 6$ sehr klein. Der Grund hierfür ist die Tatsache, daß das DGAFO-Verfahren keine Iteration durchgeführt hat. Das Ergebnis der Minimierung war zumeist x^0 selbst.

Uns.	ε ohne SVM	ε mit SVM	μ	ϑ
nein	$\|\varepsilon\|_{L_2} = 0.0480$ $\|\varepsilon\|_{L_\infty} = 0.0190$	$\|\varepsilon\|_{L_2} = 0.1156$ $\|\varepsilon\|_{L_\infty} = 0.0325$	$\|\mu\|_{L_2} = 0.1165$ $\|\mu\|_{L_\infty} = 0.0207$	$\|\vartheta\|_{L_2} = 0.1165$ $\|\vartheta\|_{L_\infty} = 0.0207$
ja	$\|\varepsilon\|_{L_2} = 0.3921$ (0.092) $\|\varepsilon\|_{L_\infty} = 0.2030$ (0.065)	$\|\varepsilon\|_{L_2} = 0.2259$ (0.054) $\|\varepsilon\|_{L_\infty} = 0.0870$ (0.027)	$\|\mu\|_{L_2} = 0.2057$ (0.039) $\|\mu\|_{L_\infty} = 0.0766$ (0.018)	$\|\vartheta\|_{L_2} = 0.2756$ (0.048) $\|\vartheta\|_{L_\infty} = 0.1756$ (0.045)

Tabelle 6.3: Rechtfertigung der Notwendigkeit der vorgeschalteten Glättung beim DGAFO-Verfahren durch den Vergleich des Interpolationsfehlers ε ohne und mit Glättung sowie des Glättungsfehlers μ und des Trainingsfehlers ϑ. Geglättet wurde auf einem Dünnen Gitter vom Level $\hat{\ell} = 3$ und einer SVM mit $(C, \gamma) = (16, 0.25)$. Es wurden für alle x gleichverteilte statistische Unsicherheiten (Uns.) aus dem Intervall $[-0.3H(x), 0.3H(x)]$ angenommen, um signifikante Unterschiede ohne detailliertere statistische Analysen feststellen zu können. Im Falle von Rauschen wurde wieder über zehn Zufallsreplikate gemittelt. Die zugehörigen Standardabweichungen sind in Klammern angegeben. Die zu erwartenden Tendenzen werden durch die Werte wiedergegeben.

Neben der Wahl eines geeigneten Dünn-Gitter-Levels soll ein weiteres Ziel dieses Abschnitts sein, im Falle von statistischem Rauschen die Notwendigkeit der vorgeschalteten Glättung zu rechtfertigen. Wie bereits erwähnt, führt die Glättung zu einer höheren Robustheit und sorgt für eine bessere Approximation des Minimums. Der Grund dafür liegt in der Verringerung des Interpolationsfehlers beim resultierenden Dünn-Gitter-Modell.

Für die Rechtfertigung der Notwendigkeit der Glättung werden Interpolationsfehler ε ohne und mit SVM sowie Glättungsfehler μ und Trainingsfehler ϑ sowohl bezüglich der $\|\cdot\|_{L_2}$- als auch bezüglich der $\|\cdot\|_{L_\infty}$-Norm gegenübergestellt. Der Glättungsfehler ergibt sich aus dem Vergleich der eigentlichen glatten Funktion ohne Rauschen mit dem SVM-Modell und der Trainingsfehler aus dem Vergleich der zu modellierenden, also verrauschten Funktion mit dem SVM-Modell. Tabelle 6.3 zeigt die entsprechenden Fehlergrößen bei glatten und verrauschten Funktionen. Die Glättung erfolgte dabei durch eine SVM mit $(C, \gamma) = (16, 0.25)$ auf einem Dünnen Gitter vom Level $\hat{\ell} = 3$. Um signifikante Unterschiede feststellen zu können, ohne daß statistische Signifikanzanalysen notwendig waren, wurden für alle x gleichverteilte statistische Unsicherheiten aus dem Intervall $[-0.3H(x), 0.3H(x)]$ gewählt. Die Tabelle zeigt die Mittelwerte und Standardabweichungen der jeweiligen Fehlergrößen über zehn Zufallsreplikate. Die folgenden zu erwartenden Tendenzen können durch die erhaltenen Resultate bestätigt werden:

- Im glatten Fall empfiehlt es sich nicht, eine SVM zu verwenden, da der entstehende Glättungsfehler negative Auswirkungen auf den Interpolationsfehler hat, im verrauschten Fall hat eine Glättung jedoch einen geringeren Interpolationsfehler zur Folge.
- Im Falle von Rauschen ist der Interpolationsfehler stets größer als im glatten Fall. Bei einer glatten Funktion ohne Glättung ist er am niedrigsten.
- Ist kein statistisches Rauschen auf den Daten vorhanden, so gilt $\mu = \vartheta$, da die SVM ein Modell auf den originalen Funktionswerten erstellt.
- Der Interpolationsfehler liegt im Falle einer SVM-Glättung in derselben Größenordnung wie der Glättungsfehler, da die geglättete Funktion interpoliert wird.
- Im verrauschten Fall gilt $\mu < \vartheta$, was ein Indiz dafür ist, daß das Rauschen teilweise herausgeglättet worden ist. Allerdings sind ε und μ bei Verwendung einer SVM sowohl im glatten als auch im verrauschten Fall in Relation zu ε im glatten Fall ohne SVM vergleichsweise

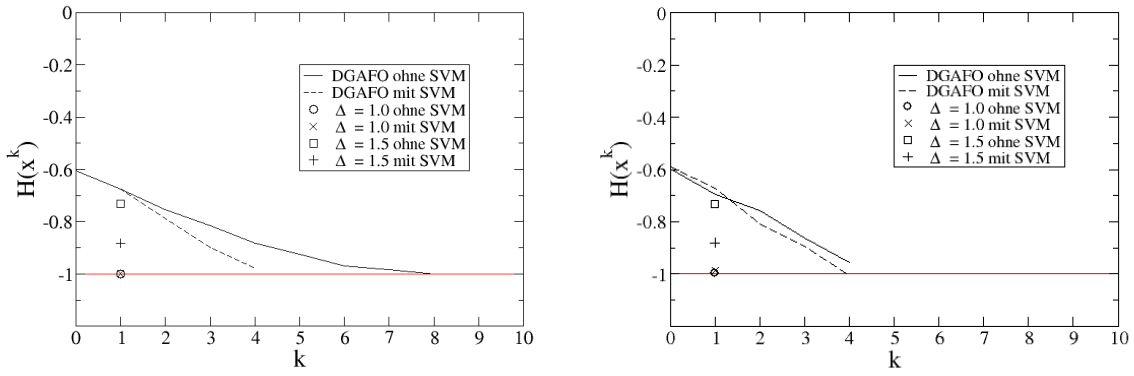


Abbildung 6.12: Globale und lokale Anwendung des DGAFO-Verfahrens mit und ohne SVM. Die rote Linie zeigt jeweils das globale Minimum -1. Bei der globalen Anwendung wurden $\Delta = 1.0$ und $\Delta = 1.5$ verwendet, und es wurde eine DGAFO-Iteration durchgeführt. Die linke Graphik zeigt den glatten und die rechte den verrauschten Fall. Je größer das zulässige Gebiet ist, desto schlechter ist die durch das DGAFO-Verfahren darin erzielte globale Lösung. Im verrauschten Fall sind die Mittelwerte über zehn Replikate angegeben. Die Fehlerbalken wurden weggelassen, da sie genauso groß sind wie die Symbole in den Graphiken.

hoch. Das bedeutet, daß die SVM die Originalfunktion nicht optimal wiedergibt und das zugehörige Modell vermutlich überangepaßt ist.

Der Abschnitt schließt mit der praktischen Diskussion der in Abschnitt 6.3.1 theoretisch erörterten Frage, ob das DGAFO-Verfahren global oder lokal angewandt werden sollte. Abbildung 6.12 zeigt den Verlauf des DGAFO-Verfahrens in beiden Fällen sowohl mit als auch ohne SVM-Glättung. Im globalen Fall wurde ein großes Δ angenommen. Es wurde $\Delta = 1.0$ und $\Delta = 1.5$ gesetzt, so daß das globale Minimum im zulässigen Bereich lag, und dann wurde eine DGAFO-Iteration durchgeführt. Lag kein statistisches Rauschen vor, so konvergierte das globale DGAFO-Verfahren mit und ohne Glättung für $\Delta = 1.0$ mit nur sechs Funktionsauswertungen exakt gegen das globale Minimum, wohingegen die lokale Version ohne Glättung acht Iterationen und 54 Funktionsauswertungen benötigte. Mit Glättung brach das Verfahren nach der vierten Iteration ab, da das Modell zu schlecht war ($r^4 = -479.62$). Das bedeutet, daß die SVM eine Minimalstelle innerhalb eines Vertrauensbereichs vorhersagt, welches jedoch einen viel größeren originalen Funktionswert hat als die aktuelle Iteration. Das kann daran liegen, daß die SVM diese Minimalstelle als Ausreißer erkennt und somit einen viel kleineren Funktionswert vorhersagt. Mit SVM war die Konvergenz zwar schneller, da die Modellfunktion schneller abfiel als die Originalfunktion, allerdings ist auch der Interpolationsfehler größer, was auch Tabelle 6.3 zu entnehmen ist. Somit ist auch Abbildung 6.12 ein praktischer Beleg dafür, daß im glatten Fall keine SVM verwendet werden sollte.
Im globalen Fall ergaben sich für $\Delta = 1.5$ schlechtere Funktionswerte als bei den jeweiligen lokalen Anwendungen. Die Interpolationsfehler waren um eine Größenordnung höher als im lokalen Fall. Im verrauschten Fall ergaben sich die gleichen Beobachtungen. Diesmal wurden nur 3% Rauschen angenommen. Hierbei führte das lokale DGAFO-Verfahren mit SVM-Glättung nach vier Iterationen bis auf Rauschen zum globalen Minimum.

Zusammenfassend läßt sich somit folgendes sagen: Das DGAFO-Verfahren ist für die vorliegende Problemstellung einsetzbar, allerdings ist die ε-SVM nicht das bestmögliche Glättungsverfahren. Bei glatten Funktionen sollte keine zusätzliche Glättung durchgeführt werden, im verrauschten Fall ist eine Glättung notwendig. Außerdem sollte das DGAFO-Verfahren bei verrauschten Daten lokal angewandt werden, da je nach Größe des zulässigen Gebiets sowohl Glättungs- als auch Diskretisierungs- und Interpolationsfehler intolerabel hoch werden.

Bemerkung 6.3.1. *Es erscheint zunächst ziemlich unbefriedigend, daß das lokale DGAFO-Verfahren für derart einfache Funktionen mehr als 50 Funktionsauswertungen benötigt, um das Abbruchkriterium zu erfüllen, wohingegen im globalen Fall lediglich sechs Funktionsauswertungen nötig sind. Es sei hier jedoch angemerkt, daß für das Abbruchkriterium im glatten Fall* $\tau = 0$ *und im verrauschten Fall* $\tau = 0.03$ *gewählt wurde. Damit das Verfahren exakt zum globalen Minimum gelangt, muß die Minimalstelle ein Gitterpunkt sein, was im globalen Fall nur durch Zufall der Fall war. Es ist a priori nicht klar, wie groß* Δ *maximal sein darf, so daß die globale Version noch gute Approximationen liefert. In den Abschnitten 7.2 und 7.3 wird gezeigt, daß das DGAFO-Verfahren bei geeigneter Wahl von* Δ *stets mit deutlich weniger Funktionsauswertungen auskommt als das jeweils beste gradientenbasierte Verfahren.*

6.4 Konvergenz des DGAFO-Verfahrens

Dieser Abschnitt befaßt sich mit der Konvergenz des DGAFO-Verfahrens. Es soll gezeigt werden, unter welchen Voraussetzungen das DGAFO-Verfahren konvergiert. Dabei wird zwischen glatten und verrauschten zu minimierenden Funktionen unterschieden. Im glatten Fall spielt der Interpolationsfehler auf dem vollen Gitter eine große Rolle, im verrauschten Fall muß zusätzlich der Glättungsfehler bezüglich der originalen Funktion, sprich der Funktion ohne Rauschen, in Betracht gezogen werden. Weiterhin soll in diesem Abschnitt ergründet werden, inwieweit das DGAFO-Verfahren schneller zum Ziel führt, das heißt mit deutlich weniger, oftmals mit nur halb so vielen Funktionsauswertungen, sprich Simulationen, auskommt als die gradientenbasierten Verfahren. Bei letzteren sind neben der aktuellen Iteration die Gradientenkomponenten, die Komponenten der Hesse-Matrix und die Armijo-Schritte auszuwerten, beim DGAFO-Verfahren sind es die Funktionsauswertungen auf den Punkten des Dünnen Gitters und die Trust-Region-Schritte. Die Methode des steilsten Abstiegs und die CG-Verfahren benötigen zur Berechnung des Gradienten deutlich weniger Simulationen als das DGAFO-Verfahren zur Evaluierung des Dünnen Gitters. Für $N = 4$ werden für den Gradienten vier und für ein Dünnes Gitter vom Level 2 neun Simulationen gebraucht. Bezüglich der Schrittweitensteuerung läßt sich sagen, daß sowohl die gradientenbasierten Methoden als auch das DGAFO-Verfahren zu Beginn der Optimierung zumeist einen Armijo- beziehungsweise Trust-Region-Schritt benötigen, in der Nähe des Minimums deutlich mehr. Das bedeutet, daß beim DGAFO-Verfahren weniger Iterationen durchgeführt werden müssen, die Konvergenzgeschwindigkeit also höher sein muß. Dies ist so zu erklären, daß die Schrittweite in jeder Iteration vorgegeben wird. Somit sind gerade zu Beginn des Verfahrens große Schritte möglich. Die in Abschnitt 6.1.3 beschriebene Kombinationsmethode liefert bei glatten Funktionen zumeist auch bei großen Schrittweiten geringe Interpolationsfehler, allerdings stets eine Abstiegsrichtung, welche meistens am Rand des aktuellen Vertrauensbereichs liegt.

Beide Ansätze haben zu einem bestimmten Zeitpunkt der Optimierung das Problem, daß nach einer Vielzahl von Schrittweitensteuerungsiterationen, also nach einem hohen Rechenaufwand, nur äußerst geringfügige Verbesserungen in der Fehlerfunktion erzielt werden. Wie jedoch bereits erwähnt, kann aufgrund des statistischen Rauschens das Minimum sowieso niemals exakt vorhersagt werden. Wird ein Ergebnis erzielt, welches sich innerhalb der Unsicherheiten bewegt, so ist die Optimierung abzubrechen. Ziel des DGAFO-Verfahrens ist es allerdings, näher ans Mininmum zu gelangen als die gradientenbasierten Verfahren, da es aufgrund des Dünnen Gitters in verschiedene Richtungen nach signifikant kleineren Funktionswerten sucht. Die Probleme der Gradientenverfahren in der Nähe des Minimums sind die folgenden:

- Der Gradient beziehungsweise die Hesse-Matrix sind ab einem bestimmten Punkt nicht mehr korrekt berechenbar. Dies wirkt sich insbesondere im Falle von Rauschen aus.
- Die Armijo-Schritte konvergieren nur noch langsam, da die Abstiegsrichtung oft weit über das Minimum hinaus führt. Da die Armijo-Schrittweite sehr schnell gegen 0 konvergiert, erhält man nach einem hohen Rechenaufwand nur irrelevante Verbesserungen.
- Das in Abschnitt 3.5.3 angesprochene *Regenrinnenproblem* kann zusätzlich zu langsamer Konvergenz führen.

Das DGAFO-Verfahren hingegen modelliert die Fehlerfunktion ohne Gradienten- und Hesse-Matrix-Information stückweise multilinear. Die Schrittweite wird jeweils vorgegeben, und man kann sie so einstellen, daß sie nicht zu schnell zu klein wird. Dies führt dazu, daß sie am Anfang groß ist und umso kleiner wird, je näher das Verfahren an das Minimum gelangt. Dieser Sachverhalt erklärt auch, warum das Trust-Region-Verfahren oftmals näher an das Minimum gelangt, da es ein quadratisches Modell innerhalb eines Vertrauensgebiets kontinuierlich mit vorgegebener Schrittweite minimiert. Allerdings ist der Rechenaufwand hierbei um ein Vielfaches höher als beim DGAFO-Verfahren. Gemeinsam haben die beiden Verfahren jedoch die Tatsache, daß sie mit deutlich weniger Iterationen auskommen als die anderen Methoden. Dies zeigt sich insbesondere zu Beginn der Optimierung.

Ziel dieses Abschnittes ist es, die Konvergenzgüte des DGAFO-Verfahrens zu erschließen, das heißt, es soll theoretisch ergründet werden,

- unter welchen Voraussetzungen die Konvergenz des Verfahrens garantiert ist,
- mit welcher Geschwindigkeit, das heißt sowohl in Bezug auf Iterationen als auch auf Funktionsevaluationen, das Verfahren im Vergleich zu den anderen konvergiert, und
- wie es sich in der Nähe des Minimums verhält.

Da der Fehler der Dünn-Gitter-Interpolation für die Konvergenz eine große Rolle spielt, wird dieser zunächst in Abschnitt 6.4.1 gemäß Griebel u. a. (1990) hergeleitet und diskutiert. Abschnitt 6.4.2 befaßt sich mit dem Glättungsfehler, der bei verrauschten Fehlerfunktionen sehr wichtig ist. Aus Interpolations- und Glättungsfehler ergibt sich ein Gesamtfehler zwischen der originalen Funktion ohne Rauschen und dem Interpolationsmodell. Dieser wird dann für den Konvergenzbeweis des DGAFO-Verfahrens verwendet. Er wird in Abschnitt 6.4.3 ermittelt, in dem anschließend der Konvergenzbeweis durchgeführt wird. Aspekte in Bezug auf die Konvergenzgeschwindigkeit zu Beginn der Optimierung und das Verhalten in der Nähe des Minimums

werden schließlich in Abschnitt 6.4.4 erörtert.

6.4.1 Interpolationsfehler

Wie bereits in Abschnitt 6.1.3 dargelegt, liegt der Interpolationsfehler bei Verwendung eines Dünnen Gitters vom Level $\hat{\ell}$ unter bestimmten Voraussetzungen in $\mathcal{O}\left(h_{\hat{\ell}}^2 \left(\log(h_{\hat{\ell}})^{-1}\right)^{\hat{\ell}-1}\right)$. Falls die Differenz $u - u_{i,j}$, wobei $u_{i,j} \in T_{i,j}$, $i+j \in \{\hat{\ell}, \hat{\ell}+1\}$, einer asymptotischen Fehlerentwicklung genügt, so ist im 2D-Fall $u \in S_{\bar{\ell}}^0$ die exakte Lösung des Interpolationsproblems auf einem vollen Gitter vom Level $\bar{\ell}$ ($\bar{\ell} = (\hat{\ell}, \hat{\ell}) \in \mathbb{N}^2$). Der Beweis hierzu geht auf Griebel u. a. (1990) zurück. Für den 2D-Fall wird er im folgenden bewiesen. Für den 3D-Fall sei auf Griebel u. a. (1990) verwiesen, und für den ND-Fall wird hier nur eine entsprechende Formel angegeben.

Satz 6.4.1 (Dünn-Gitter-Interpolationsfehler in 2D). *Es seien $u \in S_{\bar{\ell}}^0$ die exakte Lösung des Interpolationsproblems auf einem vollen Gitter vom Level $\bar{\ell}$ und $\bar{\ell} = (\hat{\ell}, \hat{\ell}) \in \mathbb{N}^2$. Die Interpolation erfolge von einem Dünnen Gitter vom Level $\hat{\ell}$ auf das volle Gitter. Weiterhin existiere für alle $u_{i,j} \in T_{i,j}$ mit $i + j \in \{\hat{\ell}, \hat{\ell} + 1\}$ punktweise eine asymptotische Fehlerentwicklung vom Typ*

$$u - u_{i,j} = C_1(h_i)h_i^2 + C_2(h_j)h_j^2 + D(h_i, h_j)h_i^2 h_j^2, \tag{6.54}$$

wobei $\forall_i\ |C_1(h_i)| \leq \kappa$, $\forall_j\ |C_2(h_j)| \leq \kappa$ und $\forall_{i,j}\ |D(h_i, h_j)| \leq \kappa,\ \kappa > 0$. Dann gilt

$$|u - \hat{u}_{\hat{\ell}}^c| \leq \left(1 + \frac{5}{4}\log\left(h_{\hat{\ell}}^{-1}\right)\right)\kappa h_{\hat{\ell}}^2 = \mathcal{O}\left(h_{\hat{\ell}}^2 \log(h_{\hat{\ell}})^{-1}\right). \tag{6.55}$$

Beweis: Der Beweis ist in Anhang H.3 angegeben. □

Analoges gilt für den 3D-Fall:

Satz 6.4.2 (Dünn-Gitter-Interpolationsfehler in 3D). *Es seien $u \in S_{\ell}^0$ die exakte Lösung des Interpolationsproblems auf einem vollen Gitter vom Level $\bar{\ell}$ und $\bar{\ell} = (\hat{\ell}, \hat{\ell}, \hat{\ell}) \in \mathbb{N}^3$. Die Interpolation erfolge von einem Dünnen Gitter vom Level $\hat{\ell}$ auf das volle Gitter. Weiterhin existiere für alle $u_{i,j,k} \in T_{i,j,k}$ mit $i+j+k \in \{\hat{\ell}, \hat{\ell}+1, \hat{\ell}+2\}$ punktweise eine asymptotische Fehlerentwicklung vom Typ*

$$\begin{aligned} u - u_{i,j,k} &= C_1(h_i)h_i^2 + C_2(h_j)h_j^2 + C_3(h_k)h_k^2 && (6.56)\\ &+ D_1(h_i, h_j)h_i^2 h_j^2 + D_2(h_i, h_k)h_i^2 h_k^2 + D_3(h_j, h_k)h_j^2 h_k^2 && (6.57)\\ &+ E(h_i, h_j, h_k)h_i^2 h_j^2 h_k^2, && (6.58) \end{aligned}$$

wobei $\forall_{p=1,2,3}\ \forall_i\ |C_p(h_i)| \leq \kappa$, $\forall_{p=1,2,3}\ \forall_{i,j}\ |D_p(h_i, h_j)| \leq \kappa$ und $\forall_{i,j,k}\ |E(h_i, h_j, h_k)| \leq \kappa,\ \kappa > 0$. Dann gilt

$$|u - \hat{u}_{\hat{\ell}}^c| \leq \left(1 + \frac{65}{32}\log\left(h_{\hat{\ell}}^{-1}\right) + \frac{25}{32}\left(\log\left(h_{\hat{\ell}}^{-1}\right)\right)^2\right)\kappa h_{\hat{\ell}}^2 = \mathcal{O}\left(h_{\hat{\ell}}^2 \left(\log(h_{\hat{\ell}})^{-1}\right)^2\right). \tag{6.59}$$

Beweis: Der Beweis ist in Anhang H.3 angegeben. □

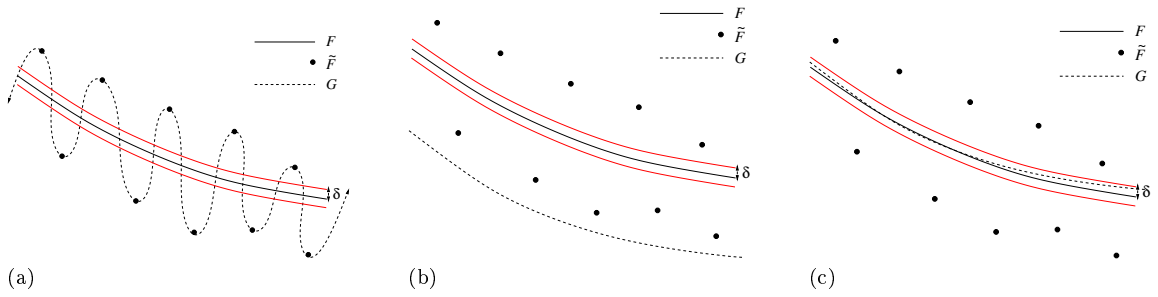


Abbildung 6.13: Glättungsmodelle mit verschiedenen Trainingsfehlern, wobei die Glättungsfunktion G die verrauschte Funktion $\tilde{F} = F + \Delta F$ approximiert: Überangepaßtes Modell mit $\vartheta = 0$ (a), Modell mit zu großem Trainingsfehler ϑ (b) und zulässiges Modell mit $|\Delta F - \vartheta| \leq \delta$ (c). Im Idealfall gilt $\vartheta(x) = \Delta F_x$.

Für den allgemeinen Fall ergibt sich mit einer entsprechenden asymptotischen Fehlerentwicklung die Formel

$$|u - \hat{u}^c_{\hat{\ell}}| \leq \left(\sum_{l=0}^{N-1} \chi_l \left(\log \left(h_{\hat{\ell}}^{-1} \right) \right)^l \right) \kappa h^2_{\hat{\ell}} = \mathcal{O}\left(h^2_{\hat{\ell}} \left(\log(h_{\hat{\ell}})^{-1} \right)^{\hat{\ell}-1} \right). \tag{6.60}$$

Dabei sind $u \in S^0_{\bar{\ell}}$, $\bar{\ell} = (\hat{\ell}, ..., \hat{\ell}) \in \mathbb{N}^N$ und $\chi_l \in \mathbb{N}$, $l = 0, ..., \hat{\ell} - 1$.
Es ist zu beachten, daß für die Existenz derartiger asymptotischer Fehlerentwicklungen die exakte Lösung u gewissen Stetigkeits- und Glattheitsanforderungen genügen muß. Da u die Fehlerfunktion F auf dem Dünnen Gitter exakt und auch sonst möglichst gut wiedergeben soll, sind diese Anforderungen auf F zu übertragen. Sind die Voraussetzungen zum Beispiel aufgrund von statistischem Rauschen nicht erfüllt, so ist, wie bereits motiviert, eine Glättung vorzuschalten. Bei praktischen Anwendungen kann in den meisten Fällen *a priori* nicht geklärt werden, ob eine asymptotische Fehlerentwicklung existiert oder nicht. Nach Griebel u. a. (1990) liefert die Kombinationsmethode in der Praxis jedoch sehr gute Ergebnisse.

6.4.2 Glättungsfehler

Es sei $||\cdot||$ eine der beiden Normen $||\cdot||_{L_2}$ oder $||\cdot||_{L_\infty}$. Der Glättungsfehler μ ist der Fehler bezüglich $||\cdot||$ zwischen der Originalfunktion $F = \tilde{F} - \Delta F$ ohne Rauschen und der geglätteten Funktion G. Dabei ist $\tilde{F}$ wieder die verrauschte Funktion. Es gilt:

$$\mu := ||F - G|| = ||(F + \Delta F) - G - \Delta F|| = ||\tilde{F} - G - \Delta F|| \leq |\vartheta| + ||\Delta F||, \tag{6.61}$$

wobei ϑ den Trainingsfehler bezeichnet. Falls Ω ein Vertrauensgebiet ist, so sei $\forall_{x \in \Omega}\ \vartheta(x) := \tilde{F}(x) - G(x)$. Im Idealfall $\vartheta(x) = \Delta F_x$ gilt $\mu(x) = 0$. Ansonsten muß für $0 < \delta << |\Delta F_x|$ gelten:

$$|\mu(x)| = |\Delta F_x - \vartheta(x)| \leq \delta. \tag{6.62}$$

Das bedeutet, daß der Trainingsfehler ϑ nicht zu klein sein darf: Abbildung 6.13(a) zeigt ein überangepaßtes Modell G, welches die durch das Rauschen hervorgerufenen Oszillationen exakt wiedergibt. In diesem Fall gilt zwar $\vartheta = 0$, allerdings $|\mu(x)| = |\Delta F_x| >> \delta$ für gewisse $x \in \Omega$.

Auf der anderen Seite darf der Trainingsfehler auch nicht zu groß sein: In Abbildung 6.13(b) gilt $\forall_{x\in\Omega}\ |\mu(x)| \approx |\Delta F_x| >> \delta$. Nur Abbildung 6.13(c) zeigt einen zulässigen Fall: Hier liegt der Trainingsfehler im Bereich des Rauschens, und es gilt $|\mu| \leq \delta$.

Um den Glättungsfehler gering zu halten, ist also ein Glättungsverfahren zu verwenden, welches das Rauschen herausfiltert und zumindest auf dem Dünnen Gitter die Fehlerfunktion möglichst gut repräsentiert. Da ein Glättungsverfahren im Falle von Simulationen aufgrund des Rechenaufwands nur auf einem Dünnen Gitter evaluiert werden kann, wird die Bedingung $|\mu| \leq \delta$ nur auf dem Dünnen Gitter betrachtet. Es ist nicht von Belang, wie G zwischen den Gitterpunkten aussieht, da die stückweise multilineare Dünn-Gitter-Interpolation mögliche Oszillationen zwischen den Gitterpunkten sowieso nicht modellieren kann und natürlich auch nicht soll.

Der folgende Satz ist entscheidend für den Konvergenzbeweis in Abschnitt 6.4.3 und liefert eine Abschätzung für den Glättungsfehler im Falle von positiv definiten RBFs:

Satz 6.4.3 (Abschätzung des Glättungsfehlers)**.** *Es sei $\Delta > 0$ die Größe des Hyperwürfels $\mathcal{W}^\Delta(x)$ mit dem Zentrum $x \in \mathbb{R}^N$, auf dem ein Dünnes Gitter vom Level $\hat{\ell}$ definiert wird. Die Fehlerfunktion $F : \mathbb{R}^N \to \mathbb{R}_0^+$ sei auf diesem Dünnen Gitter gegeben und werde mit der Funktion $G : \mathbb{R}^N \to \mathbb{R}$ approximiert, wobei die Glättung gutartig sei. Dann gilt für den Glättungsfehler μ aus Gleichung* (6.61)*:*

$$\exists_{\kappa_\mu>0}\ \forall_{y\in\mathcal{W}^\Delta(x)}\ |\mu(y)| \leq \kappa_\mu f^\mu(\Delta), \tag{6.63}$$

wobei $f^\mu : \mathbb{R}^+ \to \mathbb{R}^+$ stetig ist mit $\lim_{\Delta\to 0} f^\mu(\Delta) = 0$. Weiterhin ist $\tilde{f}(\mu) := \frac{f^\mu}{\Delta} : \mathbb{R}^+ \to \mathbb{R}^+$ ebenfalls stetig mit $\lim_{\Delta\to 0} \tilde{f}^\mu(\Delta) = 0$.

Beweis: Aufgrund der Gutartigkeit der Glättung gilt Abschätzung (6.44), wobei $\Omega = \mathcal{W}^\Delta(x)$ und $\mathcal{X}$ das Dünne Gitter ist. Die Fülldistanz auf dem Vertrauensgebiet und Δ unterscheiden sich lediglich um eine Konstante $\omega > 0$, das heißt, es gilt $\Delta_{\Omega,\mathcal{X}} = \omega\Delta$. Setzt man $\kappa_\mu := \kappa$ und $f^\mu(\Delta) := h(\omega\Delta) = h(\Delta_{\Omega,\mathcal{X}})$, so folgt Abschätzung (6.63). □

Bemerkung 6.4.4. *Aufgrund von Korollar 6.2.10 ist Abschätzung* (6.63) *im Falle der Glättung basierend auf einer Gaußschen RBF gegeben.*

Bemerkung 6.4.5. *Für Satz 6.4.3 und den gesamten Konvergenzbeweis in Abschnitt 6.4.3 ist es unerheblich, ob die originale Fehlerfunktion $F : \mathbb{R}^N \to \mathbb{R}_0^+$ oder die transformierte Funktion $\bar{F} : [0,1]^N \to \mathbb{R}_0^+$ aus dem DGAFO-Algorithmus (siehe Abschnitt 6.3.4) durch $G : \mathbb{R}^N \to \mathbb{R}$ approximiert wird. Die zu glättende Funktion muß lediglich innerhalb des Vertrauensgebietes stetig sein. Es ist zu beachten, daß gerade bei der Approximation von $\bar{F}$ die Funktion G auch negative Werte annehmen kann, da $\bar{F}$ auf dem Rand des Vertrauensgebietes gleich 0 ist. Bei der originalen Fehlerfunktion kann im Falle von $F(x) > 0$ jedoch, für Δ klein genug, aufgrund von Satz 6.4.3 $G(x) \geq 0$ vorausgesetzt werden. Im Falle von $\bar{F}$ kann dies ebenfalls oBdA angenommen werden. Ansonsten kann eine Verschiebung betrachtet werden, was weder an der Approximation noch an der Minimierung irgendetwas ändert.*

6.4.3 Gesamtfehler und Konvergenzbeweis

Der in diesem Abschnitt durchgeführte Konvergenzbeweis für das DGAFO-Verfahren ist angelehnt an einen allgemeinen Konvergenzbeweis für ableitungsfreie Trust-Region-Verfahren aus Conn u. a. (1997). Das dort verwendete Trust-Region-Verfahren basiert jedoch auf der Interpolation mit Newtonschen Fundamentalpolynomen. Daher sind die meisten Teilbeweise nicht direkt übertragbar, sondern müssen neu entwickelt werden. Ein weiterer wesentlicher Unterschied besteht darin, daß in Conn u. a. (1997) als Voraussetzung formuliert wird, daß die zu minimierende Funktion mindestens zweimal differenzierbar ist mit beschränkter Hesse-Matrix-Norm. Diese Voraussetzungen können und sollen hier nicht getroffen werden.

Es seien im folgenden zunächst ein paar grundlegende Definitionen gegeben: Es seien $c_k := \frac{\sqrt{N}\Delta_k}{\Delta_k} = \sqrt{N}$, wobei $\sqrt{N}\Delta_k = \frac{1}{2}\sqrt{N}2\Delta_k$ die Hälfte der Raumdiagonalen des Vertrauensgebiets $\Omega_k \subset \Omega$ der Größe Δ_k und Zentrum x^k (Hyperwürfel) ist und $G_k : B_{c_k\Delta_k}(x^k) \to \mathbb{R}$ die durch Approximation von $F : \Omega \to \mathbb{R}_0^+$ auf Ω_k erhaltene geglättete Funktion. Dabei sei $k \in \mathbb{N}$. Es gilt $\Omega_k \subset B_{c_k\Delta_k}(x^k)$. Da G_k im Falle einer Glättung basierend auf positiv definiten RBFs mindestens zweimal stetig differenzierbar ist, können die folgenden Definitionen aufgestellt werden: Es seien $g_k := \nabla G_k(x^k)$ und $H_k := D^2 G_k(x^k)$. Dann sei die Funktion $\Psi_k : B_{c_k\Delta_k}(0) \to \mathbb{R}$ gegeben durch:

$$\begin{aligned} \Psi_k(d) &:= G_k(x^k) + \langle g^k, d\rangle + \frac{1}{2}\langle H_k d, d\rangle \\ &= G_k(x^k + d) + \mathcal{O}\left(||d||^3\right) \\ &= F(x^k + d) - \mu(x^k + d) + \mathcal{O}\left(||d||^3\right),\ d \in B_{c_k\Delta_k}(0). \end{aligned} \tag{6.64}$$

Weiterhin sei $d^\star := \min_{d \in B_{c_k\Delta_k}(0)} \Psi_k(d)$. Das Minimum existiert, da Ψ_k stetig und $B_{c_k\Delta_k}(0)$ kompakt ist. Außerdem sei Δ_k so klein gewählt, daß $\forall_{y \in B_{\Delta_k}(x^k)}\ G_k(y) \geq 0$ und $\forall_{d \in B_{c_k\Delta_k}(0)}\ \Psi_k(d) \geq 0$. Ersteres kann aufgrund von Bemerkung 6.4.5 und letzteres aus Stetigkeitsgründen wegen $||d|| \leq c_k\Delta_k$ erzielt werden.

Abbildung 6.14 zeigt die Struktur des Konvergenzbeweises für das DGAFO-Verfahren: Gemäß Satz 6.4.3 konvergiert der Glättungsfehler für $\Delta \to 0$ gegen 0. Das gleiche wird in Satz 6.4.6 ebenfalls für den Interpolationsfehler gezeigt. Aus diesen beiden Tatsachen wird in Korollar 6.4.7 gefolgert, daß auch der Gesamtfehler für $\Delta \to 0$ gegen 0 konvergiert. In jedem Iterationsschritt des DGAFO-Verfahrens werden sowohl Ψ als auch das Dünn-Gitter-Modell q innerhalb des aktuellen Vertrauensgebiets verkleinert. Ersteres folgt aus der Standard-Trust-Region-Literatur (siehe Lemma 6.4.8), und letzteres wird in Lemma 6.4.9 bewiesen. Es gilt jedoch nur unter gewissen Voraussetzungen, die allerdings in der Regel nicht gewährleistet werden können und daher das DGAFO-Verfahren zu einem Verfahren machen, welches in der Praxis getestet werden muß. Ähnliches gilt für die in Abschnitt 6.1.3 eingeführte Kombinationsmethode nach Griebel u. a. (1990): Die Existenz der asymptotischen Fehlerentwicklungen (Gleichungen (6.54) und (6.58)) ist ebenfalls in der Regel nicht beweisbar. Lemma 6.4.10 folgert mithilfe von Lemma 6.4.9, daß in steilen Bereichen von F beziehungsweise G die Iterationen stets erfolgreich sind. Dabei wird eine Iteration *erfolgreich* genannt, falls die Fehlerfunktion

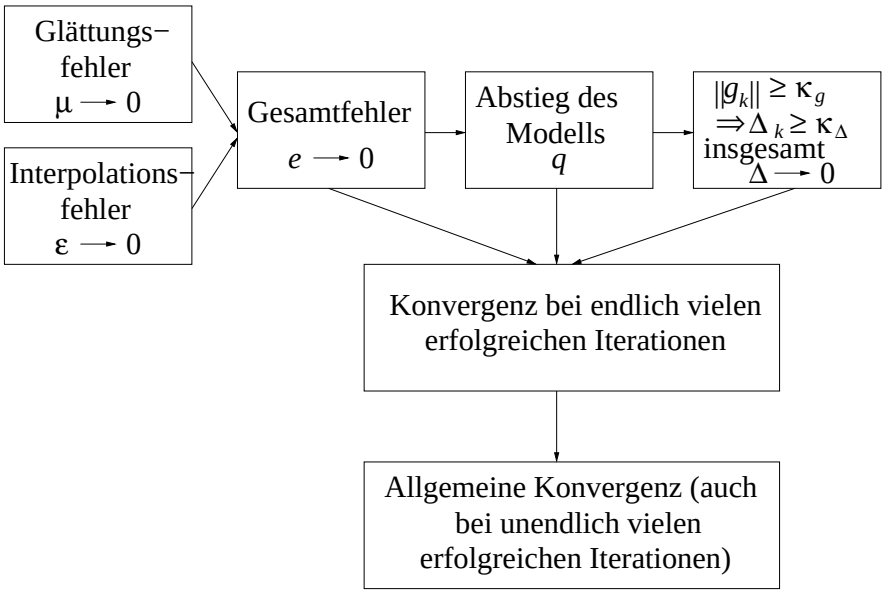


Abbildung 6.14: Struktur des Konvergenzbeweises für das DGAFO-Verfahren: Zunächst wird gezeigt, daß sowohl Glättungs- als auch Interpolationsfehler und damit der Gesamtfehler für $\Delta \to 0$ gegen 0 konvergieren. Daraus kann gefolgert werden, daß das durch die Dünn-Gitter-Interpolation erhaltene Modell q unter gewissen Voraussetzungen in jedem Schritt innerhalb eines Vertrauensgebietes verkleinert werden kann. Damit kann wiederum bewiesen werden, daß auch F verkleinert werden kann, das heißt, daß unter diesen Voraussetzungen die jeweilige Iteration erfolgreich ist. Dies gilt in jedem Fall, solange $||g^k|| \geq \kappa_g$. Dann gibt es eine untere Schranke für Δ_k. Asymptotisch konvergiert Δ gegen 0 und damit auch g^k. Letzteres wird zunächst im Falle von endlich vielen erfolgreichen Iterationen bewiesen und dann schließlich für den allgemeinen Fall. Da G^k die zu minimierende Funktion F für $\Delta \to 0$ genau approximiert, kann letztendlich geschlossen werden, daß das DGAFO-Verfahren gegen das Minimum von F konvergiert.

durch eine neue Iteration signifikant verkleinert wird, also falls $r^k \geq \eta_1$, wobei r^k in Gleichung (6.48) definiert wurde. Dies kann nur dann der Fall sein, wenn Δ_k nicht unendlich oft verkleinert wird. Lemma 6.4.10 findet eine untere Schranke $\kappa_\Delta > 0$ für Δ_k, falls $||g^k|| \geq \kappa_g > 0$. Solange dies der Fall ist, wird es erfolgreiche Iterationen nach endlich vielen Trust-Region-Schritten geben. Asymptotisch konvergiert allerdings auch Δ gegen 0 und somit auch g^k, was Satz 6.4.11 im Falle von endlich vielen erfolgreichen Iterationen beweist. Theorem 6.4.12 beweist schließlich die allgemeine Konvergenz des DGAFO-Verfahrens: Auch im Falle unendlich vieler erfolgreicher Iterationen konvergiert g^k gegen 0, und da für $\Delta \to 0$ die zu minimierende Fehlerfunktion F durch G^k exakt wiedergegeben wird, da der Glättungsfehler ebenfalls gegen 0 geht, konvergiert das DGAFO-Verfahren gegen das lokale Minimum von F, in dessen Einzugsbereich sich der Startvektor x^0 befindet.

Satz 6.4.6 (Abschätzung des Interpolationsfehlers). *Es sei $\Delta > 0$ die Größe des Hyperwürfels $\mathcal{W}^\Delta(x)$ mit dem Zentrum $x \in \mathbb{R}^N$, auf dem ein Dünnes Gitter vom Level $\hat{\ell}$ definiert wird. Die geglättete Funktion $G : \mathbb{R}^N \to \mathbb{R}_0^+$ sei auf diesem Dünnen Gitter gegeben und werde durch das Modell $q : B_\Delta(0) \to \mathbb{R}$ auf ein volles Gitter vom Level $\ell := (\hat{\ell}, ..., \hat{\ell}) \in \mathbb{N}^N$ mithilfe der Kombinationsmethode aus Abschnitt 6.1.3 interpoliert. Dann gilt für den Interpolationsfehler $\varepsilon = |u - \hat{u}^c_{\hat{\ell}}|$ aus Ungleichung (6.60) mit $u := G$ und $\hat{u}^c_{\hat{\ell}} := q$:*

$$\forall_{y \in \mathcal{W}^\Delta(x)} \; |\varepsilon(y)| \leq \kappa_\varepsilon(\hat{\ell}) f^\varepsilon_{\hat{\ell}}(\Delta), \tag{6.65}$$

wobei $\kappa_\varepsilon : \mathbb{N} \to \mathbb{R}^+$ *und* $f_{\hat{\ell}}^\varepsilon : \mathbb{R}^+ \to \mathbb{R}^+$ *mit* $\lim_{\hat{\ell}\to\infty} \kappa_\varepsilon(\hat{\ell}) = 0$ *und* $\lim_{\Delta\to 0} f_{\hat{\ell}}^\varepsilon(\Delta) = 0$. *Die Funktion* $f_{\hat{\ell}}^\varepsilon$ *ist stetig. Weiterhin ist* $\tilde{f}_{\hat{\ell}}^\varepsilon := \frac{f_{\hat{\ell}}^\varepsilon}{\Delta} : \mathbb{R}^+ \to \mathbb{R}^+$ *ebenfalls stetig mit* $\lim_{\Delta\to 0} \tilde{f}_{\hat{\ell}}^\varepsilon(\Delta) = 0$.

Beweis: Sei $y \in \mathcal{W}^\Delta(x)$ beliebig. Es gilt $h_{\hat{\ell}} = 2^{1-\hat{\ell}}\Delta$. Nach Ungleichung (6.60) existieren Konstanten $\chi_l > 0,\ l = 0, ..., N-1$, und ein $\kappa > 0$, so daß

$$\begin{aligned}
|\varepsilon(y)| &\leq \left(\sum_{l=0}^{N-1} \chi_l \left(\log\left(2^{\hat{\ell}-1}\Delta^{-1}\right)\right)^l\right) \kappa 2^{2-2\hat{\ell}}\Delta^2 \\
&= \left(\sum_{l=0}^{N-1} \chi_l \left(\hat{\ell} - 1 - \log\Delta\right)^l\right) \kappa 2^{2-2\hat{\ell}}\Delta^2 \\
&\leq \underbrace{N \cdot \max_{l=0}^{N-1} \chi_l \cdot \kappa 2^{2-2\hat{\ell}}}_{:=\kappa_\varepsilon(\hat{\ell})} \cdot \underbrace{\max_{l=0}^{N-1} \left(\hat{\ell} - 1 - \log\Delta\right)^l \cdot \Delta^2}_{:=f_{\hat{\ell}}^\varepsilon(\Delta)} \\
&= \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta).
\end{aligned}$$

Es gilt $\lim_{\hat{\ell}\to\infty} \kappa_\varepsilon(\hat{\ell}) = 0$ und aufgrund der Regel von de l'Hospital auch $\lim_{\Delta\to 0} f_{\hat{\ell}}^\varepsilon(\Delta) = 0$. Wegen $\tilde{f}_{\hat{\ell}}^\varepsilon(\Delta) = \max_{l=0}^{N-1} \left(\hat{\ell} - 1 - \log\Delta\right)^l \cdot \Delta$ folgt ebenfalls nach der Regel von de l'Hospital, daß $\lim_{\Delta\to 0} \tilde{f}_{\hat{\ell}}^\varepsilon(\Delta) = 0$. □

Korollar 6.4.7 (Abschätzung des Gesamtfehlers)**.** *Unter den Voraussetzungen von Satz 6.4.3 und Satz 6.4.6 gilt für den Gesamtfehler e:*

$$\forall_{y\in\mathcal{W}^\Delta(x)}\ |e(y)| \leq \kappa_e(\hat{\ell}) f_{\hat{\ell}}^e(\Delta), \tag{6.66}$$

wobei $\kappa_e : \mathbb{N} \to \mathbb{R}^+$ *und* $f_{\hat{\ell}}^e : \mathbb{R}^+ \to \mathbb{R}^+$ *mit* $\lim_{\hat{\ell}\to\infty} \kappa_e(\hat{\ell}) = \kappa_\mu$ *und* $\lim_{\Delta\to 0} f_{\hat{\ell}}^e(\Delta) = 0$. *Die Funktion* $f_{\hat{\ell}}^e$ *ist stetig. Weiterhin ist* $\tilde{f}_{\hat{\ell}}^e := \frac{f_{\hat{\ell}}^e}{\Delta} : \mathbb{R}^+ \to \mathbb{R}^+$ *ebenfalls stetig mit* $\lim_{\Delta\to 0} \tilde{f}_{\hat{\ell}}^e(\Delta) = 0$.

Beweis: Sei $y \in \mathcal{W}^\Delta(x)$ beliebig. Es gilt nach den Sätzen 6.4.3 und 6.4.6:

$$\begin{aligned}
|e(y)| &= |F(y) - q_k(y-x)| \leq |F(y) - G_k(y)| + |G_k(y) - q_k(y-x)| = |\mu(y)| + |\varepsilon(y)| \\
&\leq \kappa_\mu f^\mu(\Delta) + \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta).
\end{aligned}$$

Es sei $\kappa_e(\hat{\ell}) := \kappa_\mu + \kappa_\varepsilon(\hat{\ell})$. Dann gilt:

$$\begin{aligned}
|e(y)| &\leq \kappa_e(\hat{\ell}) \underbrace{\max\left(f^\mu(\Delta), f_{\hat{\ell}}^\varepsilon(\Delta)\right)}_{:=f_{\hat{\ell}}^e(\Delta)} \\
&= \kappa_e(\hat{\ell}) f_{\hat{\ell}}^e(\Delta).
\end{aligned}$$

Es gilt $\lim_{\hat{\ell}\to\infty} \kappa_e(\hat{\ell}) = \kappa_\mu$ und $\lim_{\Delta\to 0} f_{\hat{\ell}}^e(\Delta) = 0$. Die Funktion $f_{\hat{\ell}}^e$ ist als Maximumsfunktion zweier stetiger Funktionen stetig. Trivialerweise ist auch $\tilde{f}_{\hat{\ell}}^e$ stetig, und es gilt $\lim_{\Delta\to 0} \tilde{f}_{\hat{\ell}}^e(\Delta) = 0$. □

Lemma 6.4.8 (Abstieg von Ψ). *In jeder Iteration k gilt*

$$\exists_{\kappa_\Psi \in (0,1)} \; \Psi_k(0) - \Psi_k(d^\star) \geq \kappa_\Psi ||g^k|| \min\left(\Delta_k, \frac{||g^k||}{||H_k||}\right). \tag{6.67}$$

Beweis: Die Funktion Ψ_k ist nichts anderes als die allgemeine quadratische Form aus Gleichung (3.36). Der Beweis der obigen Abschätzung ist daher in der Standard-Trust-Region-Literatur zu finden, beispielsweise in Moré (1983). □

Lemma 6.4.9 (Abstieg des Modells q). *Es seien $k \in \mathbb{N}$ und $\Delta_k > 0$ die Größe des Vertrauensgebiets Ω_k mit Zentrum x^k. Es sei weiterhin $\alpha^k \in \mathbb{R}$ so, daß*

$$\forall_{d \in B_{c_k \Delta_k}(0)} \; G_k(x^k + d) \leq \Psi_k(d) + \alpha^k ||d||^3. \tag{6.68}$$

Außerdem sei $\beta^k \in \mathbb{R}$ so, daß

$$\forall_{d \in B_{c_k \Delta_k}(0)} \; \Psi_k(d) \leq \Psi_k(d^\star) + \beta^k ||d - d^\star||. \tag{6.69}$$

Es sei $\beta^k \leq 0$ oder

$$\beta^k > 0 \wedge ||g^k|| > \frac{c_k \beta^k}{\kappa_\Psi \tilde{\kappa}_q} \tag{6.70}$$

mit $\tilde{\kappa}_q \in (0,1)$ und Δ_k so klein, daß

(i)

$$\Delta_k \leq \frac{||g^k||}{||H_k||} \tag{6.71}$$

und

(ii)

$$c_k^3 \alpha^k \Delta_k^2 + \kappa_\varepsilon(\hat{\ell}) \tilde{f}_{\hat{\ell}}^{\varepsilon}(\Delta) \leq \underbrace{\kappa_\Psi \tilde{\kappa}_q ||g^k|| - c_k \beta^k}_{>0, \text{ falls } \beta^k \leq 0, \text{ oder im Falle von (6.70)}}. \tag{6.72}$$

Außerdem seien die Voraussetzungen von Satz 6.4.6 erfüllt. Dann gilt:

$$\exists_{\kappa_q \in (0,1)} \; q_k(0) - q_k(d^k) \geq \kappa_q ||g^k|| \Delta_k > 0, \tag{6.73}$$

das heißt, das Modell q wird innerhalb von Ω_k verkleinert.

Beweis: Es sei $G_{\hat{\ell}}^N \subset \Omega_k$ das auf dem Vertrauensgebiet Ω_k festgelegte volle Gitter vom Level

$\bar{\ell} := (\hat{\ell}, ..., \hat{\ell}) \in \mathbb{N}^N$ mit Maschenweite $2^{1-\hat{\ell}}\Delta_k$. Dann gilt:

$$
\begin{aligned}
q_k(0) - q_k(d^k) &= q_k(0) - \min_{d \in G_{\bar{\ell}}^N} q_k(d) \\
&= q_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Big(G_k(x^k + d) - \varepsilon(x^k + d) \Big) \\
&\geq q_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Big(|G_k(x^k + d)| + |\varepsilon(x^k + d)| \Big) \\
&\overset{\text{Satz 6.4.6}}{\geq} \underbrace{q_k(0)}_{= G_k(x^k) = \Psi_k(0)} - \min_{d \in G_{\bar{\ell}}^N} \Big(|G_k(x^k + d)| + \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \Big) \\
&\overset{G_k(x^k+d) \geq 0}{=} \Psi_k(0) - \min_{d \in G_{\bar{\ell}}^N} G_k(x^k + d) - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&\overset{(6.68)}{\geq} \Psi_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Big(\Psi_k(d) + \alpha^k ||d||^3 \Big) - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&\geq \Psi_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Big(\Psi_k(d) + c_k^3 \alpha^k \Delta_k^3 \Big) - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&= \Psi_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Psi_k(d) - c_k^3 \alpha^k \Delta_k^3 - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&\overset{(6.69)}{\geq} \Psi_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Big(\Psi_k(d^\star) + \beta^k ||d - d^\star|| \Big) - c_k^3 \alpha^k \Delta_k^3 - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&\geq \Psi_k(0) - \min_{d \in G_{\bar{\ell}}^N} \Big(\Psi_k(d^\star) + c_k \beta^k \Delta_k \Big) - c_k^3 \alpha^k \Delta_k^3 - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&= \Psi_k(0) - \Psi_k(d^\star) - c_k \beta^k \Delta_k - c_k^3 \alpha^k \Delta_k^3 - \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \\
&\overset{\text{Lemma 6.4.8}}{\geq} \kappa_\Psi ||g^k|| \underbrace{\min\left(\Delta_k, \frac{||g^k||}{||H_k||} \right)}_{= \Delta_k \text{ wg. (6.71)}} - \Big(c_k \beta^k \Delta_k + c_k^3 \alpha^k \Delta_k^3 + \kappa_\varepsilon(\hat{\ell}) f_{\hat{\ell}}^\varepsilon(\Delta_k) \Big) \\
&\overset{(6.72)}{\geq} \kappa_\Psi ||g^k|| \Delta_k - \kappa_\Psi \tilde{\kappa}_q ||g^k|| \Delta_k \\
&= \underbrace{\kappa_\Psi (1 - \tilde{\kappa_q})}_{:= \kappa_q \in (0,1)} ||g^k|| \Delta_k \\
&= \kappa_q ||g^k|| \Delta_k .
\end{aligned}
$$

□

Lemma 6.4.10 (Erfolgreiche Iterationen: Untere Schranke für Δ_k und asymptotisches Verhalten)**.** *Es sei*

$$\Psi_{k,l}(d) := G_{k,l}(x^k) + \langle g^{k,l}, d \rangle + \frac{1}{2} \langle H_{k,l} d, d \rangle, \;\; d \in B_{c_k \Delta_{k,l}}(0),$$

wobei $\Delta_{k,l} := \gamma_1^l \Delta_k > 0$, $l = 0, 1, 2, ...$, *und* $G_{k,l}$ *die Fehlerfunktion* F *in* $B_{c_k \Delta_{k,l}}(x^k)$ *approximiert. Weiterhin gelte* $\forall_{k,l} \; ||g^{k,l}|| > \kappa_g$, *wobei* $\kappa_g > 0$. *Außerdem seien alle Voraussetzungen*

von Korollar 6.4.7 und Lemma 6.4.9 für alle $k, l \in \mathbb{N}$ erfüllt. Dann gilt:

$$\exists_{\kappa_\Delta > 0} \forall_k \exists_l \ \Delta_{k,l} > \kappa_\Delta, \tag{6.74}$$

das heißt, die k-te DGAFO-Iteration ist erfolgreich.

Beweis: Es sei $k \in \mathbb{N}$ gegeben. Weiterhin sei $\bar{l} \in \mathbb{N}$ das erste l, so daß $\Delta_{k,\bar{l}} > 0$ so klein ist, daß

$$\frac{\kappa_e(\hat{\ell}) f_{\hat{\ell}}^e(\Delta_{k,\bar{l}})}{\Delta_{k,\bar{l}}} \leq \frac{(1-\eta_1)\kappa_g \kappa_q}{2}. \tag{6.75}$$

Die Existenz eines derartigen $\Delta_{k,\bar{l}}$ ist aufgrund von Korollar 6.4.7 gewährleistet.
Es sei $d^{k,\bar{l}}$ das Minimum des Dünn-Gitter-Interpolationsmodells q auf dem vollen Gitter mit Maschenweite $2^{1-\hat{\ell}}\Delta_{k,\bar{l}}$. Nach Lemma 6.4.9 gilt:

$$q_k(0) - q_k(d^{k,\bar{l}}) > \kappa_q ||g^{k,\bar{l}}|| \Delta_{k,\bar{l}}.$$

Es sei weiterhin:

$$r^{k,l} := \frac{F(x^k + d^{k,l}) - F(x^k)}{q_k(d^{k,l}) - q_k(0)}.$$

Dann gilt:

$$\begin{aligned}
|r^{k,\bar{l}} - 1| &= \left| \frac{F(x^k + d^{k,\bar{l}}) - F(x^k)}{q_k(d^{k,\bar{l}}) - q_k(0)} - \frac{q_k(d^{k,\bar{l}}) - q_k(0)}{q_k(d^{k,\bar{l}}) - q_k(0)} \right| \\
&\leq \left| \frac{F(x^k + d^{k,\bar{l}}) - q_k(d^{k,\bar{l}})}{q_k(d^{k,\bar{l}}) - q_k(0)} \right| + \left| \frac{F(x^k) - q_k(0)}{q_k(d^{k,\bar{l}}) - q_k(0)} \right| \\
&\leq \frac{2\kappa_e(\hat{\ell}) f_{\hat{\ell}}^e(\Delta_{k,\bar{l}})}{\kappa_q ||g^{k,\bar{l}}|| \Delta_{k,\bar{l}}} \\
&\overset{(6.75)}{\leq} 1 - \eta_1. \\
\Rightarrow r^{k,\bar{l}} &\geq \eta_1.
\end{aligned}$$

Damit ist gezeigt, daß ein endliches $\bar{l} \in \mathbb{N}$ existiert, so daß die k-te DGAFO-Iteration erfolgreich ist. Es bleibt zu zeigen, daß die untere Schranke für $\Delta_{k,l}$, κ_Δ, unabhängig von k und l ist. Betrachte hierzu die Funktion $f^\Delta : \mathbb{R}^+ \to \mathbb{R}^+$ mit

$$f^\Delta(\Delta) := \frac{\kappa_e(\hat{\ell}) f_{\hat{\ell}}^e(\Delta)}{\Delta}.$$

Da f^Δ stetig ist und $\lim_{\Delta \to 0} f^\Delta(\Delta) = 0$ gilt, existiert ein $\Delta^\star \in \mathbb{R}^+$ mit

$$f^\Delta(\Delta^\star) \leq \frac{(1-\eta_1)\kappa_g \kappa_q}{2}.$$

Für $\kappa_\Delta := \gamma_1 \Delta^\star$ gilt $\forall_k \exists_l \ \Delta_{k,l} > \kappa_\Delta$. □

Nun kann die eigentliche Konvergenz des DGAFO-Verfahrens bewiesen werden:

Satz 6.4.11 (Konvergenz für endlich viele erfolgreiche Iterationen)**.** *Es seien die Voraussetzungen von Korollar 6.4.7 und Lemma 6.4.9 für alle $k, l \in \mathbb{N}$ erfüllt. Falls es nur endlich viele erfolgreiche Iterationen $k^\star$ gibt, so folgt $g^{k^\star} = 0$.*

Beweis: (durch Widerspruch). Annahme: $\exists_{\kappa_g > 0}\ ||g^{k^\star}|| > \kappa_g$. Dann gilt nach Lemma 6.4.10: $\exists_{\bar{l}}\ \Delta_{k^\star,\bar{l}} > \kappa_\Delta$. Da es allerdings nur endlich viele erfolgreiche Iterationen gibt, gilt $\forall_{l\in\mathbb{N}}\ x^{k^\star+l} = x^{k^\star}$ und somit $\lim_{l\to\infty} \Delta_{k^\star,l} = 0$ ↯.
Damit ist die Annahme falsch, und es gilt $g^{k^\star} = 0$. □

Theorem 6.4.12 (Konvergenz des DGAFO-Verfahrens)**.** *Es seien die Voraussetzungen von Korollar 6.4.7 und Lemma 6.4.9 für alle $k, l \in \mathbb{N}$ erfüllt, und es gebe unendlich viele erfolgreiche Iterationen. Dann gilt:*

$$\liminf_{k\to\infty} ||g^k|| = 0. \tag{6.76}$$

Beweis: (durch Widerspruch). Annahme: $\exists_{\kappa_g>0}\ \forall_{k\in\mathbb{N}}\ ||g^k|| > \kappa_g$. Dann gilt:

$$\begin{aligned} F(x^k) - F(x^{k+1}) &\geq \eta_1 \left(q_k(0) - q_k(d^k) \right) \\ &\overset{\text{Lemma 6.4.9}}{\geq} \eta_1 \kappa_q ||g^k|| \Delta_k \\ &\overset{\text{Lemma 6.4.10, Annahme}}{>} \eta_1 \kappa_q \kappa_g \kappa_\Delta. \end{aligned}$$

Es werden nun alle bisherigen Iterationen $x^0, ..., x^{k+1}$ betrachtet. Es folgt sofort:

$$\begin{aligned} F(x^0) - F(x^{k+1}) &= \sum_{j=0}^{k} F(x^j) - F(x^{j+1}) \\ &> (k+1)\eta_1 \kappa_q \kappa_g \kappa_\Delta. \end{aligned}$$

Da es unendlich viele erfolgreiche Iterationen gibt, kann der Grenzwert für $k \to \infty$ betrachtet werden: Die rechte Seite konvergiert gegen $+\infty$. Damit konvergiert auch die linke Seite gegen $+\infty$, woraus folgt, daß F nach unten unbeschränkt ist. Es gilt jedoch $\forall_{x\in\mathbb{R}^N}\ F(x) \geq 0$ ↯.
Damit ist die Annahme falsch, und es gilt $\liminf_{k\to\infty} ||g^k|| = 0$. □

Bemerkung 6.4.13. *(i) Die Voraussetzung $F \in C^1$, das heißt insbesondere, daß ∇F existiert, ist für die Konvergenz des DGAFO-Verfahrens nicht notwendig. Die Existenz eines lokalen Minimums bedeutet, daß es in einem Vertrauensgebiet um dieses Minimum keinen Punkt mit kleinerem Funktionswert gibt. Satz 6.4.3 besagt, daß F mittels einer glatten Funktion G beliebig genau approximiert werden kann. Gemäß Theorem 6.4.12 konvergiert das DGAFO-Verfahren gegen ein lokales Minimum x^{opt} von G. Gäbe es in einem Vertrauensgebiet um dieses Minimum einen Punkt y mit einem kleineren Funktionswert $F(y) < F(x^{\text{opt}})$, so wäre $\nabla G_k(x^{\text{opt}}) \neq 0$ für Δ klein genug, und x^{opt} wäre kein lokales Minimum von G. Somit ist durch $\liminf_{k\to\infty} ||g^k|| = 0$ die Konvergenz des DGAFO-Verfahrens bewiesen. Der Beweis der Konvergenz gegen ein x^{opt} mit der Eigenschaft $\nabla F(x^{\text{opt}}) = 0$ für den Fall $F \in C^2$ und einigen weiteren Voraussetzungen ist in Anhang H.4 angegeben.*

(ii) Die Beschränktheit von Δ_k aus Lemma 6.4.10 gilt nur, solange $||g^k|| > \kappa_g$. Asymptotisch gesehen gibt es jedoch eine Teilfolge $(k_i)_{i\in\mathbb{N}}$ mit $\lim_{i\to\infty} g^{k_i} = 0$. Wegen

$$0 \leq 2^{1-\hat{\ell}}\Delta_{k_i} \leq ||x^{k_i} - x^{k_i+1}||$$

und $\lim_{i\to\infty} ||x^{k_i} - x^{k_i+1}|| = 0$, folgt auch $\lim_{i\to\infty} \Delta_{k_i} = 0$. Das bedeutet, daß mit g^{k_i} auch Δ_{k_i} gegen 0 konvergiert, so daß Δ so klein gemacht werden kann, daß der Glättungsfehler so gering ist, daß das lokale Minimum von G, gegen das das DGAFO-Verfahren konvergiert, bis auf eine gewisse Toleranz auch das lokale Minimum von F ist.

6.4.4 Konvergenz in der Praxis und Aspekte zur Konvergenzgeschwindigkeit

Beim DGAFO-Verfahren handelt es sich um ein mathematisches Optimierungsverfahren, dessen Konvergenz unter bestimmten Voraussetzungen beweisbar ist. Ob diese Voraussetzungen allerdings in der Praxis erfüllt sind, ist in der Regel nicht gewährleistet. Somit können Konvergenz und auch Konvergenzgeschwindigkeit nur numerisch bewiesen werden, das heißt, das Verfahren muß praktisch evaluiert werden. Auch das Polak-Ribière-Verfahren (siehe Abschnitt 3.4.1) gehört zu dieser Klasse von Methoden: Eine Konvergenz kann nur im Falle der Existenz einer in der Praxis nur äußerst aufwendig oder gar nicht zu bestimmenden Schrittweite bewiesen werden, und Aussagen über die Konvergenzgeschwindigkeit können gar nicht getroffen werden. Nach einer kurzen Diskussion über die Voraussetzungen für die Konvergenz des DGAFO-Verfahrens wird in diesem Abschnitt erörtert, ob und inwieweit Aussagen über dessen Konvergenzgeschwindigkeit getroffen werden können.

Das DGAFO-Verfahren konvergiert gemäß Abschnitt 6.4.3 unter den folgenden Voraussetzungen:

1. Es muß eine Abschätzung für den Glättungsfehler μ existieren. Für $\Delta \to 0$ muß auch der Glättungsfehler gegen 0 konvergieren. Diese Abschätzung ist durch Ungleichung (6.63) gegeben und für Glättungen, welche auf positiv definiten RBFs basieren, erfüllt. Die in Abschnitt 6.2.3 angesprochenen Regularisationsverfahren müssen jedoch dazu in der Lage sein, das Rauschen geeignet herauszufiltern. Letzteres kann allerdings nur praktisch evaluiert werden, was in Abschnitt 7.1 erfolgt.
2. Auch die Existenz der asymptotischen Fehlerentwicklung aus Griebel u. a. (1990) ist in der Praxis nicht garantiert und hängt auch von den Glattheitseigenschaften der zu interpolierenden Funktion ab. Bislang konnte sie theoretisch lediglich für die Lösung der Poisson-Gleichung bewiesen werden. Eine detaillierte Evaluation in Griebel u. a. (1990) hat jedoch gezeigt, daß die Dünn-Gitter-Interpolation in der Praxis in vielen Fällen zu guten Ergebnissen führt. Aufgrund der hier vorgeschalteten Glättung ist die Anwendbarkeit allerdings wahrscheinlicher.
3. Die Voraussetzungen von Lemma 6.4.9 müssen erfüllt sein:
   (i) Voraussetzung (6.68): Die Funktion G sollte nach der Glättung und Regularisierung möglichst wenig oder gar nicht oszillierend sein. Im Falle von $G_k(x+d) \leq \Psi_k(d)$ für $x \in \Omega_k$, $d \in B_\Delta(0)$ und μ_k klein genug stimmt G_k in Ω_k gut mit F überein, und Voraussetzung (6.68) ist für jedes beliebige $\alpha^k \geq 0$ erfüllt. Das heißt, daß in steilen

Bereichen der Fehlerfunktion die Voraussetzung nicht problematisch ist. Ist allerdings $G_k(x+d) \geq \Psi_k(d)$ für $x \in \Omega_k$ und $d \in B_\Delta(0)$, so muß G_k nahezu quadratisch (wie Ψ_k) sein. Ansonsten kann kein Abstieg von q_k bewiesen werden. Das bedeutet aber, daß auch F nahezu quadratisch sein muß, was höchstens in einer kleinen Umgebung eines lokalen Minimums vorausgesetzt werden kann.

(ii) Voraussetzungen (6.69) und (6.70): Die Existenz von β^k kann aufgrund einer Taylorentwicklung von Ψ_k vorausgesetzt werden. Allerdings besagt Vorausssetzung (6.70) nur, daß das DGAFO-Verfahren im Falle von $\beta^k > 0$ in steilen Bereichen der Fehlerfunktion zum Erfolg führt. Je kleiner der Diskretisierungsfehler ist, also im Falle von $\Psi_k(d^\star) \approx \min_{d \in G_{\tilde{\ell}}^N} \Psi_k(d)$, desto kleiner kann β^k vorausgesetzt werden, und Voraussetzung (6.69) kann durch

$$\Psi_k(d_{G_{\tilde{\ell}}^N}) \leq \Psi_k(d^\star) + \beta^k ||d_{G_{\tilde{\ell}}^N} - d^\star|| \tag{6.77}$$

ersetzt werden, wobei $d_{G_{\tilde{\ell}}^N} := \min_{d \in G_{\tilde{\ell}}^N} \Psi_k(d)$. Dies zeigt der Beweis von Lemma 6.4.9. Bei kleinem β^k ist Voraussetzung (6.70) nicht problematisch.

(iii) Voraussetzung (6.71): Es gilt: $\forall_k \ \lim_{l \to \infty} \Delta_{k_l} = 0$ für eine Teilfolge $(k_i)_{i \in \mathbb{N}}$. Allerdings ändert sich mit $\Delta_{k,l}$ innerhalb einer Iteration auch $g^{k,l}$. Die Norm $||g^{k,l}||$ konvergiert jedoch für $l \to \infty$ gegen $||\nabla F(x^k)||$, falls $\nabla F(x^k)$ existent, allerdings niemals gegen 0, es sei denn, x^k ist bereits das gesuchte lokale Minimum. Somit ist Voraussetzung (6.71) unproblematisch.

(iv) Voraussetzung (6.72): Diese ist aufgrund von Satz 6.4.6 erfüllbar.

Sowohl Konvergenz und Konvergenzgeschwindigkeit werden auch in Bezug auf das vorliegende statistische Rauschen im folgenden diskutiert. Analog zum Parameter h bei gradientenbasierten Optimierungsverfahren darf auch Δ im Falle des DGAFO-Verfahrens nicht so klein gewählt werden, daß die Funktionswerte zweier benachbarter Gitterpunkte aufgrund des Rauschens nicht mehr unterscheidbar sind. Allerdings basiert der Konvergenzbeweis aus Abschnitt 6.4.3 auf der Wahl eines möglichst kleinen Δ. Es existiert hierbei also auch das besagte Dilemma, daß Δ einerseits nicht zu groß gewählt werden darf (aufgrund der Konvergenz) und andererseits auch nicht zu klein (aufgrund des Rauschens). Analog zu den gradientenbasierten Verfahren kann daher auch im Falle des DGAFO-Verfahrens nur praktisch evaluiert werden, ob und mit welchen Verfahrensparametern es zu optimalen Kraftfeldparametern führt. Evaluierung und Anwendungen des DGAFO-Verfahrens werden zusammen mit einem Vergleich zu gradientenbasierten Verfahren in Kapitel 7 aufgeführt.

Auch zum Erhalt einer hohen Konvergenzgeschwindigkeit, vor allem zu Beginn der Optimierung, das heißt in steilen Bereichen von F, ist die Wahl eines größeren Δ vonnöten, ohne daß dabei eine der oben genannten Voraussetzungen für die Konvergenz des DGAFO-Verfahrens verletzt wird. Es werden im folgenden einige heuristische Überlegungen zur Konvergenzgeschwindigkeit getroffen. Der Index $_g$ beziehe sich auf gradientenbasierte Abstiegsverfahren, der Index $_H$ auf Abstiegsverfahren, welche eine Hesse-Matrix verwenden, und der Index $_D$ auf das DGAFO-Verfahren. Es seien weiterhin $\bar{M}$ die durchschnittliche Anzahl an Funktionsauswertungen pro Iteration sowie $\bar{l}$ die durchschnittliche Anzahl an Armijo- beziehungsweise Trust-

Region-Schritten. Dann gilt:

$$\begin{aligned} \bar{M}_g &= N + \bar{l}_g, \\ \bar{M}_H &= N + \frac{N(N+1)}{2} + \bar{l}_H, \\ \bar{M}_D &= 2N \cdot \bar{l}_D. \end{aligned} \tag{6.78}$$

Der Nachteil des DGAFO-Verfahrens besteht in der multiplikativen Abhängigkeit von $\bar{M}_D$ von $\bar{l}_D$, da in jedem Trust-Region-Schritt ein neues Dünnes Gitter verwendet wird. Zu Beginn der Optimierung werde allerdings $\bar{l}_g = \bar{l}_H = \bar{l}_D = 1$ vorausgesetzt. Es gilt dann $\bar{M}_g < \bar{M}_D < \bar{M}_H$. Das DGAFO-Verfahren ist somit günstiger als ein Verfahren, welches eine Hesse-Matrix benötigt, also auch günstiger als das exakte Trust-Region-Verfahren aus Abschnitt 3.4.3. Insgesamt muß es jedoch mit weniger Iterationen auskommen als ein gradientenbasiertes Verfahren, damit es insgesamt weniger Funktionsauswertungen braucht: Falls k im allgemeinen die Anzahl an Iterationen bezeichnet, muß also gelten: $k_D < k_g$. Falls

$$k_D < \frac{N+1}{2N} k_g, \tag{6.79}$$

so benötigt das DGAFO-Verfahren weniger Iterationen und Funktionsauswertungen als ein gradientenbasiertes Verfahren. Es stellt sich nun die Frage, wie dies gesteuert werden kann. Durch die Wahl eines genügend großen Δ_0 (und auch γ_1) zu Beginn der Optimierung kann eine schnelle Konvergenz erzielt werden, denn nur dann sind große DGAFO-Schritte realisierbar. Allerdings sind die Voraussetzungen des Konvergenzsatzes dann erneut zu diskutieren: Angenommen, F hat eine ähnliche Gestalt wie in Abbildung 3.11, das heißt, F sei annähernd linear mit großer Steigung und kleiner Krümmung. Dann gilt $\forall_{d \in B_{\Delta_k}(0)}\ G_k(x^k + d) \approx \Psi_k(d)$, und Voraussetzung (6.68) ist für $\alpha^k \approx 0$ erfüllt. Falls F nicht annähernd linear ist, jedoch mehr als quadratisch abfällt, so gilt bei geringem Glättungsfehler $\forall_{x \in \Omega_k,\ d \in B_\Delta(0)}\ G_k(x + d) \leq \Psi_k(d)$, und die Voraussetzung ist, wie oben bereits diskutiert, für alle $\alpha^k \geq 0$ erfüllt. In derart steilen Bereichen von F gelten auch die Voraussetzungen bezüglich β^k ((6.69) und (6.70)). Auch Voraussetzung (6.71) ist dann unproblematisch, vor allem bei kleiner Krümmung. Bezüglich Voraussetzung (6.72) ist zu sagen, daß für $\alpha^k \approx 0$ auch $c_k^3 \alpha^k \Delta_k^2 \approx 0$ gilt. Sämtliche Voraussetzungen sind somit unproblematisch, auch bei der Wahl eines großen Δ_k. Letztere könnte sich höchstens negativ auf Interpolations- und Glättungsfehler auswirken, was jedoch auch in steilen Bereichen kein Problem darstellt, solange der abfallende Trend von F analog zu den gradientenbasierten Verfahren korrekt wiedergegeben wird.

Im folgenden wird der Anfang des Optimierungsprozesses betrachtet und ein kurzer Vergleich zwischen dem DGAFO- und einem gradientenbasierten Abstiegsverfahren gegeben, das heißt, es wird erörtert, bei welchen Gegebenheiten die Geschwindigkeit des DGAFO-Verfahrens deutlich höher ist. 'Zu Beginn des Optimierungsprozesses' bedeutet hierbei, daß bei jeder Iteration die Anzahl an Trust-Region- beziehungsweise Armijo-Schritten gleich 1 ist, und daß k_g so gewählt ist, daß die Anzahl an Armijo-Schritten für x^{k_g} noch gleich 1 und für x^{k_g+1} größer als 1 ist. Weiterhin sei k_D mit $||x^{k_D} - x^0|| \leq ||x^{k_g} - x^0||$ so, daß die Größe des Vertrauensgebiets für alle $k \leq k_D$ gleich Δ_0 ist und daß $x^{k_g} \in \Omega_k$. Somit gilt auch $x^{k_g} \notin \Omega_{k-1}$. Das bedeutet, daß

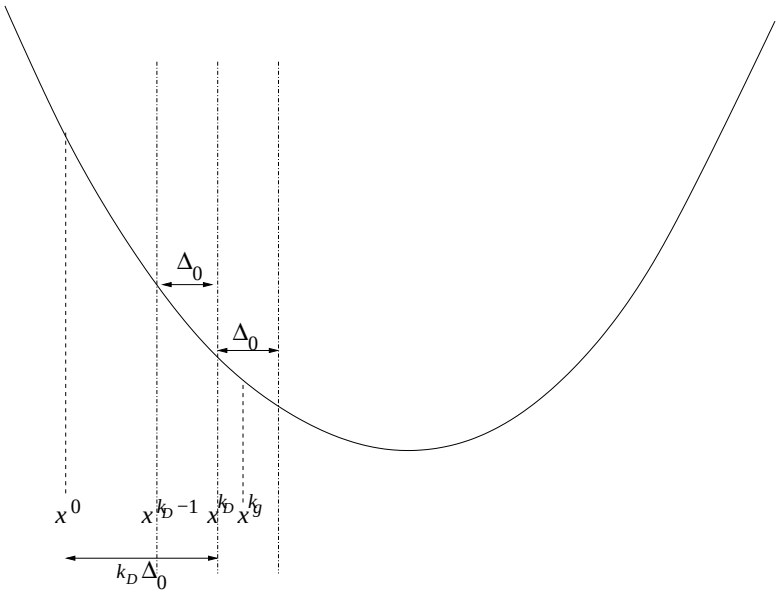


Abbildung 6.15: Konvergenzgeschwindigkeit des DGAFO-Verfahrens zu Beginn des Optimierungsprozesses: Bei geeigneter Wahl der Größe des initialen Vertrauensgebiets Δ_0 kann erreicht werden, daß die Anzahl an Iterationen des DGAFO-Verfahrens (k_D) in diesem Bereich deutlich kleiner ist als im Falle eines gradientenbasierten Verfahrens (k_g). Realistisch ist, daß die Anzahl an Iterationen und Funktionsauswertungen bei Verwendung des DGAFO-Verfahrens um den Faktor 2 reduziert werden kann.

sowohl x^{k_D} als auch x^{k_g} vom DGAFO-Verfahren mit k_D Schritten der Größe Δ_0 erreicht wird. Abbildung 6.15 veranschaulicht dies. Der Abstand zwischen x^0 und x^{k_D} beträgt $k_D\Delta_0$. Es gilt:

$$k_D\Delta_0 \leq ||x^{k_g} - x^0||.$$

Falls

$$\Delta_0 > \frac{||x^{k_g} - x^0||}{\underbrace{\frac{N+1}{2N}k_g}_{>1,\text{ falls } k_g \geq 2}},$$

das heißt, falls

$$\Delta_0 > \zeta||x^{k_g} - x^0||$$

für ein möglichst großes $\zeta \in (0,1)$, so folgt Ungleichung (6.79). Ein realistischer Fall ist $k_g = 4$. Wählt man

$$\zeta := \frac{4N}{(N+1)k_g} = \frac{N}{N+1} < 1,$$

so benötigt das DGAFO-Verfahren weniger als halb so viele Iterationen und Funktionsauswertungen wie das gradientenbasierte Abstiegsverfahren. In der Praxis ist somit eine Reduktion des Rechenaufwandes um den Faktor 2 zu Beginn des Optimierungsprozesses äußerst plausibel.

Es bleibt zu erörtern, wie gut und wie schnell das DGAFO-Verfahren in der Nähe des Minimums konvergiert. Die quadratische Konvergenz des exakten Trust-Region-Verfahrens ist lediglich auf die r-überlineare beziehungsweise q-quadratische Konvergenz des Newton-Verfahrens zurückzuführen, falls sich das Minimum im Innern des Vertrauensgebiets befindet (Satz 3.39). Zu Beginn

der Optimierung, das heißt in steilen Bereichen der Fehlerfunktion, wo $||g^k|| \geq \kappa_g$, beträgt die Länge der DGAFO-Schhritte gemäß Lemma 6.4.10 mindestens κ_Δ, bei geeigneter Wahl von Δ_0 und η_1 sogar Δ_0 (maximal $c_0\Delta_0$). Dabei sei hier der Einfachheit halber außer acht gelassen, daß das Minimum nicht auf dem Rand des Vertrauensbereichs angenommen werden kann, sondern an einem Punkt, der um h vom Rand entfernt liegt. Von einem gewissen Punkt an ist Δ_k zu verkleinern, und asymptotisch gilt $\Delta_k \overset{k\to\infty}{\to} 0$. Daher gibt es nur endlich viele Schritte der Länge Δ_0. In der Nähe des Minimums, wo nur noch lokale Verfeinerungen erforderlich sind, muß Δ_k so klein gewählt werden, daß auch der Interpolationsfehler $f_{\hat{\ell}}^{\varepsilon}(\Delta_k)$ eine gewisse obere Schranke unterschreitet, die zum Beispiel durch Ungleichung (6.72) definiert ist. Zum einen verhält sich dieser wie $\mathcal{O}\left(\Delta_k^2\left(\log\Delta_k^{-1}\right)^{N-1}\right)$, und $\Delta_k^2\left(\log\Delta_k^{-1}\right)^{N-1}$ konvergiert wesentlich langsamer gegen 0 als Δ_k^2. Zum anderen ist in der Nähe des Minimums die Abschätzung von $||g^k||$ nach unten (Ungleichung (6.70)) nicht mehr gewährleistet, so daß auch Ungleichung (6.72) für jedes $\Delta_k > 0$ verletzt ist. Diese beiden Aspekte führen dazu, daß Δ_k in der Praxis irgendwann zu klein ist, so daß keine signifikanten Verbesserungen mehr erzielt werden können. Das ist dasselbe Argument, welches auch auf die Armijo-Schrittweite bei gradientenbasierten Abstiegsverfahren zutrifft. Hinzu kommt das statistische Rauschen, welches dazu führt, daß zu nah beieinanderliegende Punkte bezüglich ihres Fehlerfunktionswertes nicht mehr unterscheidbar sind. In der Praxis ist es daher stets ratsam, das Abbruchkriterium $\Delta_k < \Delta_{\min}$ zu verwenden. Das bedeutet, daß sowohl Konvergenz als auch Konvergenzgeschwindigkeit nur in der Praxis evaluierbar sind. Es kann keinen theoretischen Beweis in Bezug auf die Konvergenzgeschwindigkeit geben, und zwar aus folgendem Grund: Die jeweilige Größe von Δ_k ist geschwindigkeitsbestimmend, Δ_k wird jedoch unabschätzbar verkleinert. Die neue Iteration kann sowohl am Rand als auch im Innern des aktuellen Vertrauensgebiets liegen. Daher ist keine Konvergenzgeschwindigkeitsbetrachtung innerhalb des letzten fixen Vertrauensgebiets wie beim exakten Trust-Region-Verfahren möglich.

7 Bewertung und Anwendung des DGAFO-Verfahrens

In diesem Kapitel wird das DGAFO-Verfahren praktisch evaluiert und auf Molekulare Simulationen angewandt. Dazu werden zunächst für die vorliegende Problemstellung geeignete Glättungs- und Regularisationsverfahren (siehe Abschnitt 6.2) selektiert, was in Abschnitt 7.2 erfolgt. Weiterhin befaßt sich dieser Abschnitt mit der Bewertung des DGAFO-Verfahrens anhand der Korrelationsfunktionen aus Abschnitt 3.5.2 bezüglich Rechenaufwand im Vergleich zu gradientenbasierten Verfahren (GROW, siehe Anhang G.1), sowohl für den glatten als auch den verrauschten Fall. Außerdem wird analysiert, ob und inwieweit das DGAFO-Verfahren näher an das Minimum herangelangt als GROW beziehungsweise die Variante des Verfahrens nach Stoll. Abschnitt 7.3 befaßt sich dann mit Anwendungen der ableitungsfreien Methode auf Molekulare Simulationen. Es wird dabei die Fragestellung untersucht, ob und inwieweit das DGAFO-Verfahren schneller konvergiert als GROW und näher an das Minimum herangelangt als GROW beziehungsweise die Variante des Verfahrens nach Stoll. Schließlich wird in Abschnitt 7.4 eine finale Evaluation des DGAFO-Verfahrens gegeben.

7.1 Praktische Auswahl von Glättungs- und Regularisationsverfahren

Aufgrund theoretischer Überlegungen bestand bereits die Tendenz, als Glättungsverfahren eine positiv definite RBF, insbesondere die Gaußsche RBF, zu wählen. In diesem Abschnitt wird gezeigt, daß diese Wahl auch aus praktischen Gründen zu favorisieren ist.
Eine detaillierte praktische Evaluation von Glättungs- und Regularisierungsverfahren, angewandt auf das DGAFO-Verfahren, wird in Hülsmann u. a. (2012) gegeben. Diese Arbeiten basieren auf einer Diplomarbeit (Hemmersbach, 2011), die am Fraunhofer-Institut SCAI angefertigt wurde. Die wichtigsten Ergebnisse dieser Diplomarbeit werden im folgenden zusammengefaßt. Um zu bewerten, ob sich ein Glättungs- beziehungsweise Regularisierungsverfahren für das DGAFO-Verfahren eignet, wird das Verhalten des DGAFO-Verfahrens zusammen mit dem jeweiligen Verfahren analysiert. Dabei werden neben der Konvergenz, das heißt, wie nahe das Verfahren an das Minimum gelangt, auch die Robustheit und die Anzahl an benötigten Funktionsauswertungen innerhalb des DGAFO-Optimierungsprozesses berücksichtigt. Tabelle 7.1 zeigt die in dieser Arbeit zur Auswahl stehenden Glättungs- und Regularisationsverfahren und deren Abkürzungen im Überblick.

Glättungsverfahren	Regularisierungsverfahren
Support Vector Machine (SVM)	Methode der Kleinsten Quadrate (KQ)
Radiale Basisfunktionen (RBFs)	Naive Elastische Netze (NEN)
	$\alpha = 0$: Least Absolute Shrinkage and Selection Operator (LASSO)
	$\alpha = 1$: Ridge Regression
Multivariate Adaptive Regression Splines (MARS)	Multivariate Adaptive Regression Splines (MARS)
Linear propery approximation (Lipra)	

Tabelle 7.1: In dieser Arbeit zur Auswahl stehenden Glättungs- und Regularisierungsverfahren. Angegeben sind der Klarheit halber auch nochmals sämtliche hier verwendeten Abkürzungen.

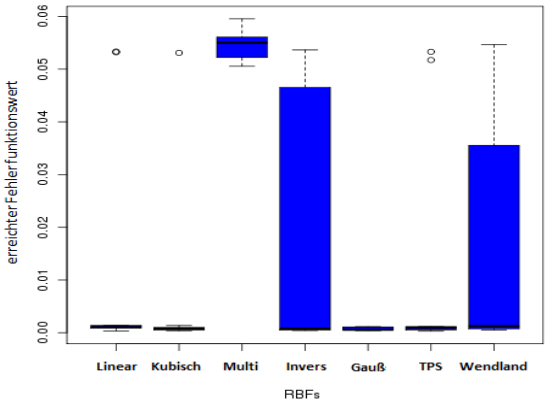
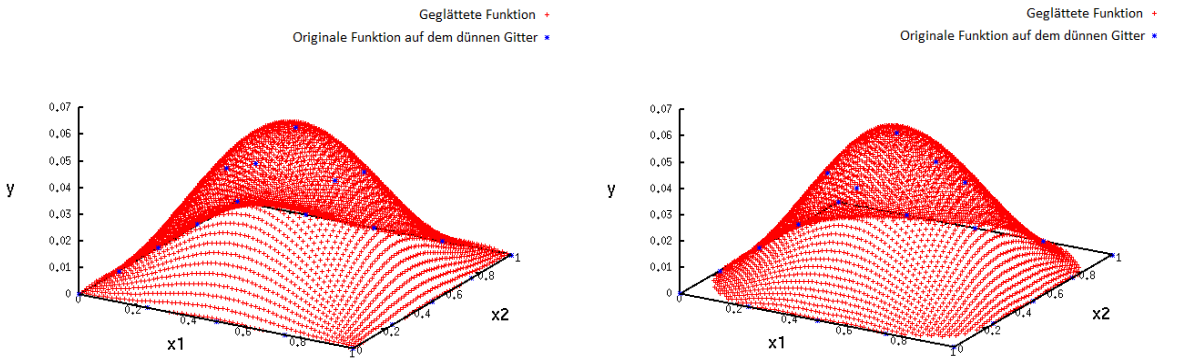


Abbildung 7.1: Box-Plots der Fehlerfunktionswerte, die durch das DGAFO-Verfahren, kombiniert mit auf RBFs basierenden Glättungsverfahren, erreicht wurden. Die eingesetzten RBFs waren die lineare, kubische, Multiquadrik (Multi), Inverse Multiquadrik (Invers), Gaußsche, Thin-Plate-Spline-RBF (TPS) und eine Wendland-Funktion. Als geeignet konnten lediglich die lineare, kubische, Gaußsche und Thin-Plate-Splite-RBF eingestuft werden.

Wahl des besten Glättungsverfahrens Daß eine Glättung im Falle verrauschter Fehlerfunktionswerte absolut notwendig ist, wurde bereits in Abschnitt 6.3.6 anhand eines Testbeispiels praktisch dargelegt. Dabei wurde eine SVM als Glättungsmethode eingesetzt. Diese ist jedoch nicht die finale Wahl. Die Analyse in Hemmersbach (2011) hat ergeben, daß bestimmte RBFs weitaus bessere Ergebnisse liefern. Bei diesen RBFs handelt es sich um die lineare, kubische, Gaußsche und Thin-Plate-Spline-RBF (vergleiche Abschnitt 6.2.2). Als nicht geeignet konnten hingegen die Multiquadrik und die inverse Multiquadrik eingestuft werden. Weiterhin wurden in Hemmersbach (2011) zum Vergleich Wendlandsche RBFs betrachtet (vergleiche Abschnitt 6.2.4). Abbildung 7.1 zeigt Box-Plots für die jeweils erreichten Fehlerfunktionswerte, was ein Kriterium für die Konvergenzgüte ist, aus zehn statistisch unabhängigen Zufallsreplikaten (analog zu den Verfahrensevaluationen in Kapitel 4): Je kleiner der erreichte Fehlerfunktionswert ist, desto näher gelangt das Verfahren an das Minimum heran. Die kleinsten Fehlerfunktionswerte erreichten äußerst robust die vier oben genannten RBFs. Selektiert wurde allerdings die Gaußsche RBF, und zwar aus den folgenden Gründen:

1. Es waren keine Ausreißer im entsprechenden Box-Plot vorhanden.
2. Eine genauere Analyse des Approximationsfehlers hat gezeigt, daß die Glättungen basierend auf kubischen und Thin-Plate-Spline-RBFs die Funktion $\bar{F}$ (vergleiche Abschnitt



(a) (b)

Abbildung 7.2: Glättungen von $\bar{F}$ auf dem Einheitsquadrat $[0,1]^2$ basierend auf Gaußschen RBFs (a) und Thin-Plate-Spline-RBFs (b). Die blauen Punkte markieren die originalen (verrauschten) Funktionswerte von $\bar{F}$ auf dem Dünnen Gitter. Es gilt $x1 = \xi(Q^2)$, $x2 = \xi(L)$ und $y = \bar{F}(x1, x2)$. Die Glättung basierend auf Thin-Plate-Spline-RBFs gibt $\bar{F}$ auf dem Rand des Einheitsquadrats nur schlecht wieder.

6.3.4) auf dem Einheitsquadrat weniger gut wiedergeben als diejenige, die auf Gaußschen RBFs basiert. Abbildung 7.2 zeigt dies am Beispiel der Thin-Plate-Spline-RBFs für den zweidimensionalen Fall: Abgebildet sind die Funktionswerte der mithilfe von Gaußschen (Abbildung 7.2(a)) und Thin-Plate-Spline-RBFs (Abbildung 7.2(b)) geglätten Funktion sowie die Originalwerte von $\bar{F}$ aus Gleichung (6.49) auf dem zugehörigen Dünnen Gitter vom Level 2 gegen $\xi(Q^2)$ und $\xi(L)$ (siehe Gleichung (6.52)). Die LJ-Parameter σ und ε wurden festgehalten. Es ist deutlich zu erkennen, daß eine Glättung basierend auf Thin-Plate-Spline-RBFs $\bar{F}$ am Rand des Einheitsquadrats nur sehr schlecht wiedergab, wohingegen Gaußsche RBFs eine gute Approximation auf dem gesamten Einheitsquadrat lieferten.

3. Die Gaußsche RBF ist die einzige dieser vier RBFs, die positiv definit ist, das heißt, die Selektion von Gaußschen RBFs ist gemäß den Überlegungen in Abschnitt 6.2.4 auch theoretisch fundiert.

Es ist zu beachten, daß lineare RBFs zu Beginn der Optimierung ebenfalls gute Approximationen lieferten, das heißt in Bereichen, wo die Fehlerfunktion relativ steil ist und in denen auch die Methode des steilsten Abstiegs äußerst erfolgreich anwendbar ist. Als Alternative wird zu Beginn der Optimierung daher neben der Gaußschen RBF auch die lineare RBF als für das DGAFO-Verfahren geeignet eingestuft. Der MARS-Algorithmus (siehe Abschnitt 6.2.3), also eine stückweise lineare Approximation, hingegen ist als Glättungsmethode nicht verwendbar. Allerdings konnte die Lipra-Methode (siehe Abschnitt 6.2.4) überraschenderweise überzeugen. Neben der durch die Verwendung dieses Glättungsverfahrens resultierenden Konvergenzgüte, die vergleichbar mit der der RBFs war, waren im Durchschnitt auch etwas weniger Funktionsauswertungen zu verzeichnen. Ein detaillierterer Vergleich zwischen Lipra und der Glättung basierend auf Gaußschen RBFs wird in Abschnitt 7.2.1 angegeben.

Wahl des besten Regularisierungsverfahrens Wie bereits in Abschnitt 6.2 erwähnt, ist die Wahl des besten Regularisierungsverfahrens nur praktisch möglich. Zur Auswahl stehen die Ge-

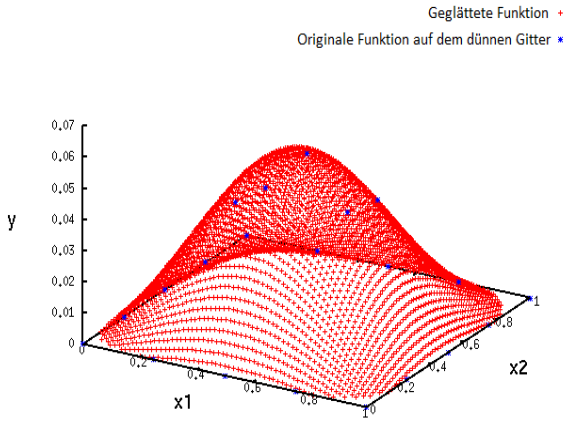


Abbildung 7.3: Gättungen von $\bar{F}$ auf dem Einheitsquadrat $[0,1]^2$ basierend auf Gaußschen RBFs, kombiniert mit einer LASSO-Regularisierung. Die blauen Punkte markieren die originalen (verrauschten) Funktionswerte von $\bar{F}$ auf dem Dünnen Gitter. Es gilt $x1 = \xi(Q^2)$, $x2 = \xi(L)$ und $y = \bar{F}(x1, x2)$. Die zu approximierende Funktion wird auf dem Rand des Einheitsquadrats nur schlecht wiedergegeben.

wichtung gemäß der statistischen Unsicherheiten (Gleichung (6.36)) sowie die in Abschnitt 6.2.3 dargestellten Verfahren, also die KQ-Methode, Naive Elastische Netze (das LASSO-Verfahren für $\alpha = 0$ und die Ridge-Regression für $\alpha = 1$) sowie der MARS-Algorithmus. Bei einer SVM ist eine Regularisierung bereits implizit enthalten.
Die Regularisierungsverfahren wurden in Kombination mit der selektierten Glättungsmethode basierend auf Gaußschen RBFs evaluiert. Die Verwendung der KQ-Methode kommt damit dem Vorgehen gleich, keine Regularisierung durchzuführen. Die statistischen Unsicherheiten bezüglich der zu optimierenden Zielgrößen unterschieden sich zwar je nach Zielgröße und Temperatur signifikant, nicht jedoch bei unterschiedlichen Parametervektoren auf einem Dünnen Gitter. Somit machte eine Gewichtung gemäß der Unsicherheiten keinen Sinn. Sämtliche anderen Regularisierungsverfahren konnten jedoch die Ergebnisse der DGAFO-Optimierung im Gegensatz zur KQ-Regularisierung deutlich verbessern. Das LASSO-Verfahren führt eine Variablenselektion durch, das heißt, es neigt dazu, Ausreißer zu detektieren, die gar keine Ausreißer sind. Somit war das mittels des LASSO-Verfahrens erstellte Modell oftmals unterangepaßt. Demgegenüber stellte sich der Ridge-Regression-Schätzer als besser geeignet heraus, da er einen Kompromiß zwischen dem zur Überangepaßtheit neigenden KQ-Schätzer und zur Unterangepaßtheit neigenden LASSO-Schätzer darstellt. Abbildung 7.3 zeigt die Glättung basierend auf Gaußschen RBFs, kombiniert mit einer LASSO-Regularisierung: Man sieht, daß auch hier die zu glättende Funktion am Rand nur schlecht wiedergegeben wurde.
Ähnliches gilt auch für eine SVM: Sowohl die ϵ- als auch die ν-SVM neigten aufgrund der geringen Anzahl an Datenpunkten auf einem Dünnen Gitter dazu, zu viele Punkte als Ausreißer zu detektieren. Die Ridge-Regression ist aufgrund des oben erwähnten Kompromisses besser geeignet und zusätzlich einfacher zu parameterisieren als eine SVM. Somit ist eine SVM als Glättungs- und Regularisierungsverfahren weniger zu empfehlen.
Ein Naives Elastisches Netz mit $\alpha = 0.7$ lieferte zwar eine noch bessere Konvergenzgüte des DGAFO-Verfahrens, allerdings war der Rechenaufwand zur Bestimmung des optimalen α im Vergleich zu dem erhaltenen Gewinn viel zu hoch. Die Verwendung eines Naiven Elastischen Netzes mit $\alpha \notin \{0, 1\}$ ist somit im Falle der vorliegenden Problemstellung nicht lohnenswert.

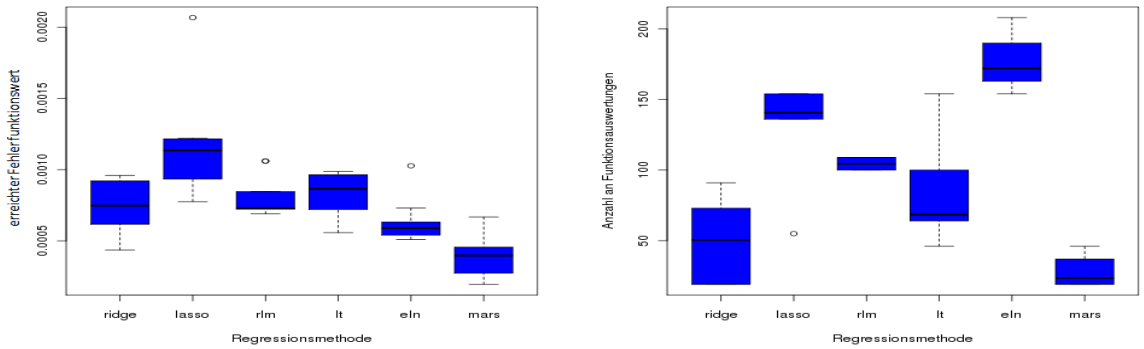


Abbildung 7.4: Box-Plots der Fehlerfunktionswerte (a) und der Anzahl an Funktionsauswertungen (b), die aus der Verwendung des DGAFO-Verfahrens, kombiniert mit einer Glättung basierend auf Gaußschen RBFs und verschiedenen Regressionsmethoden als Regularisierungsverfahren, resultierten. Die eingesetzten Regressionsmethoden waren Ridge-Regression, das Lasso-Verfahren, eine gewichtete lineare Regression (rlm), die Erweiterung der RBF-Approximation um einen linearen Term (lt), ein Naives Elastisches Netz (eln) mit $\alpha = 0.7$ und der MARS-Algorithmus. Aus den beiden Abbildungen wird deutlich, daß der MARS-Algorithmus als Regularisierungsverfahren zu selektieren ist. Als Alternative steht auch die Ridge-Regression zur Verfügung.

Als am besten geeignet hat sich der MARS-Algorithmus als Regularisierungsmethode herausgestellt, und dies nicht nur bezüglich Robustheit und Konvergenzgüte, sondern auch bezüglich der Anzahl an Funktionsauswertungen: Abbildung 7.4 zeigt Box-Plots sämtlicher in Hemmersbach (2011) verwendeter Regularisationsmethoden. Abbildung 7.4(a) zeigt die erreichten Fehlerfunktionswerte und Abbildung 7.4(b) die Anzahl an Funktionsauswertungen. Neben den hier betrachteten Verfahren wurden dort zusätzlich eine gewichtete lineare Regression, basierend auf M-Schätzern, und die Erweiterung der RBF-Approximation um einen linearen Term betrachtet. Man erkennt deutlich, daß der MARS-Algorithmus die besten Ergebnisse lieferte. Somit wird dieser als beste Regularisierungsmethode selektiert. Ridge-Regression wird aus den oben genannten Gründen als Alternative vorgeschlagen. Das eingesetzte Naive Elastische Netz mit $\alpha = 0.7$ erreichte zwar äußerst robust sehr kleine Fehlerfunktionswerte, benötigte aber stets weit über hundert Funktionsauswertungen.
Für das Lipra-Verfahren war der KQ-Schätzer der beste Regularisierer. Alle anderen Verfahren verzerrten die quadratische Approximation auf äußerst ungünstige Art und Weise.

Fazit Gaußsche RBFs in Kombination mit dem MARS-Algorithmus (alternativ auch mit Ridge-Regression) und das Lipra-Verfahren zusammen mit dem KQ-Schätzer (insbesondere für $N \geq 5$) haben sich also als am besten geeignet herausgestellt und werden daher in diesem Kapitel zur Glättung innerhalb des DGAFO-Verfahrens eingesetzt.

Optimierungsaufgabe	Verfahren	# It.	# Eval.	Spezielle Einstellungen
1, $\tau = 10^{-5}$	DGAFO	5	53	
	PR	6	41	
3, $\tau = 10^{-2}$	DGAFO	5	55	$\eta_1 = 0.2$ $\eta_2 = 0.5$, $\Delta_0 = 0.2 \cdot \Delta_{\max}$
	PR	33	224	
5, $\tau = 10^{-4}$	DGAFO	8	91	$\Delta_0 = 0.1 \cdot \Delta_{\max}$
	PSB	14	214	$\zeta_A = 0.1$
7, $\tau = 1.2 \cdot 10^{-2}$	DGAFO	6	91	$\eta_1 = 0.2, \Delta_0 = 0.1 \cdot \Delta_{\max}$
	SA/FR/PR	20	>140	

Tabelle 7.2: Vergleich des DGAFO-Verfahrens mit GROW bezüglich Rechenaufwand im Falle der glatten Korrelationsfunktionen: Angegeben sind der Schwellenwert τ des Abbruchkriteriums sowie die Anzahl an Iterationen (# It.) und Funktionsauswertungen (# Eval.) bis zum Erhalt des Abbruchkriteriums für DGAFO und das jeweils schnellste Verfahren innerhalb von GROW für die Optimierungsaufgaben 1, 3, 5 und 7 (siehe Abschnitt 3.5.3). Es handelte sich hierbei um das Polak-Ribière-Verfahren (PR) und das PSB-Verfahren (PSB). Im Falle von Optimierungsaufgabe 7 lieferten mehrere Verfahren ähnliche Konvergenzgeschwindigkeiten. Allerdings benötigten die Methode des steilsten Abstiegs (SA), das Fletcher-Reeves- (FR) und das Polak-Ribière-Verfahren 20 Iterationen und mehr als 140 Funktionsauswertungen. Es wurde die Gewichtung $\forall_{i,T}\ w_{i,T} = 1$ verwendet. Weiterhin sind spezielle Einstellungen bezüglich der Verfahrensparameter angegeben. Die Standardeinstellungen waren $\zeta_A = 0.2$, $\eta_1 = 0.4$, $\eta_2 = 0.7$, $\gamma_1 = 0.5$, $\gamma_2 = 1.5$ und $\Delta_0 = 0.6 \cdot \Delta_{\max}$. In den meisten Fällen benötigte das DGAFO-Verfahren deutlich weniger Iterationen und Funktionsevaluationen als das jeweils schnellste gradientenbasierte Verfahren, oftmals war es mindestens um den Faktor 2 schneller als GROW.

7.2 Bewertung des DGAFO-Verfahrens anhand von simulierten Simulationen

In diesem Abschnitt wird das DGAFO-Verfahren anhand der Korrelationsfunktionen evaluiert. Dazu werden zunächst in Abschnitt 7.1 geeignete Glättungs- und Regularisationsverfahren praktisch selektiert. Anschließend wird das DGAFO-Verfahren bezüglich zwei verschiedenen Gesichtspunkten bewertet: Abschnitt 7.2.1 behandelt die Fragestellung, ob und inwieweit das Verfahren tatsächlich durchschnittlich mit deutlich weniger Funktionsevaluationen auskommt als die hier betrachteten gradientenbasierten Verfahren, und Abschnitt 7.2.2, ob und inwieweit es näher an das Minimum gelangt.

7.2.1 Vergleich mit gradientenbasierten Verfahren bezüglich Rechenaufwand

In diesem Abschnitt wird das DGAFO-Verfahren bezüglich Rechenaufwand anhand der Korrelationsfunktionen aus Abschnitt 3.5.2 evaluiert. Unterschieden wird dabei im folgenden zwischen glatten und verrauschten Korrelationsfunktionen.

Glatte Korrelationsfunktionen Tabelle 7.2 zeigt die Ergebnisse des Verfahrens im Vergleich zu dem jeweils besten gradientenbasierten Algorithmus innerhalb von GROW im Falle der glatten Korrelationsfunktionen, sprich für die Optimierungsaufgaben 1, 3, 5 und 7, allerdings zunächst mit der Gewichtung $\forall_{i,T}\ w_{i,T} = 1$. Die Ergebnisse bezüglich GROW wurden Hülsmann u. a. (2010b) entnommen. Außer bei Optimierungsaufgabe 1 kam das DGAFO-Verfahren

mit deutlich weniger Iterationen und Funktionsauswertungen bis zum Erhalt des jeweiligen Abbruchkriteriums aus, oftmals war es sogar etwa um den Faktor 2 schneller als GROW. Spezielle Änderungen bestimmter Verfahrensparameter sind ebenfalls in der Tabelle angegeben. Bei GROW war in einigen Fällen lediglich der Armijo-Akzeptanzparameter ζ_A in einem Fall zu verkleinern, was allerdings keinen Nutzen mit sich brachte, falls die Abstiegsrichtung inkorrekt war. Analog dazu wurde beim DGAFO-Verfahren der Akzeptanzparameter η_1 in zwei Fällen verkleinert. Wie bereits in Abschnitt 6.4.4 dargelegt, ist die Größe des Vertrauensgebiets der entscheidende Parameter bezüglich der Konvergenzgeschwindigkeit des DGAFO-Verfahrens. Die Standardeinstellung $\Delta_0 = 0.6 \cdot \Delta_{\max}$ war in den meisten Fällen zu groß, so daß ein kleineres Vertrauensgebiet zum Erhalt einer guten Modellierung vonnöten war. Allerdings führte nur die Wahl eines nicht zu kleinen Δ zu einer Erhöhung der Konvergenzgeschwindigkeit im Vergleich zu GROW. Bei Optimierungsaufgabe 1 konnte weder mit einem kleineren noch mit einem größeren Δ eine geringere Anzahl an Funktionsauswertungen erzielt werden, allerdings führte die Standardeinstellung dennoch zu einer kleineren Anzahl an Iterationen. Es ist zu beachten, daß ζ_A und η_1 in gewisser Weise auch geschwindigkeitsbestimmende Parameter sind. Es hat sich allerdings herausgestellt, daß eine Verkleinerung dieser Parameter lediglich dazu führt, daß das Abbruchkriterium überhaupt erreicht wird, allerdings nicht zu einer sichtlichen Erhöhung der Konvergenzgeschwindigkeit. Das DGAFO-Verfahren hat allerdings den Vorteil, daß es dort den zusätzlichen, viel entscheidenderen Parameter Δ_0 gibt. Diesen Vorteil weist auch das exakte Trust-Region-Verfahren auf, allerdings ist dort in jedem Iterationsschritt eine Hesse-Matrix zu berechnen, was den Rechenaufwand im Vergleich drastisch erhöht.

Für den verrauschten Fall wurde zunächst eine SVM als Glättungsmethode verwendet, die jedoch bei keiner der vier Optimierungsaufgaben überzeugen konnte: Entweder das jeweilige Abbruchkriterium wurde gar nicht erreicht, oder aber mit deutlich mehr Rechenaufwand, sprich Funktionsevaluationen, als im Falle des jeweils besten gradientenbasierten Verfahrens. Da sich jedoch im Falle der Korrelationsfunktionen in Abschnitt 4.2 eine Gewichtung der Zielgrößen gemäß des Ausmaßes an statistischen Unsicherheiten als besser geeignet herausgestellt hat, wurde diese für den verrauschten Fall hier ebenfalls verwendet. Dadurch kann ausgeschlossen werden, daß die Tatsache, daß eine SVM zu keinen zufriedenstellenden Resultaten geführt hat, nur auf eine ungeeignete Gewichtung zurückzuführen ist. In Hemmersbach (2011) hat sich herausgestellt, daß eine SVM mit oben erwähnter Gewichtung bessere Resultate liefern kann. Wie bereits erwähnt, führten jedoch eine Glättung basierend auf Gaußschen RBFs kombiniert mit dem MARS-Algorithmus oder Ridge-Regression als Regularisierer sowie die Lipra-Methode kombiniert mit dem KQ-Schätzer im Durchschnitt zu kleineren Fehlerfunktionswerten. Dies ist in Hemmersbach (2011) auch anhand eines studentischen t-Tests überprüft worden. Es handelt sich dabei um einen statistischen Test, der überprüft, ob sich zwei Stichproben voneinander signifikant unterscheiden oder nicht. Die Kombination des DGAFO-Verfahrens mit einer SVM wird daher im folgenden beim Vergleich mit gradientenbasierten Verfahren nicht mehr betrachtet.

Verrauschte Korrelationsfunktionen Tabelle 7.3 zeigt den Vergleich zwischen dem DGAFO-Verfahren, kombiniert mit den am besten geeigneten Glättungs- und Regularisierungsverfahren, und GROW bezüglich Rechenaufwand im Falle der verrauschten Korrelationsfunktionen: Die durchschnittliche Anzahl an Funktionsevaluationen war beim DGAFO-Verfahren im Falle der

Algorithmus	Glättung/Regularisierung	# Repl.	# It.	# Eval.
PR	–	9	4–5	28–29
DGAFO	Gaußsche RBF/MARS	10	1–2	28–29
DGAFO	Lipra/KQ	10	1	21

Tabelle 7.3: Vergleich des DGAFO-Verfahrens mit GROW bezüglich Rechenaufwand im Falle der verrauschten Korrelationsfunktionen: Angegeben sind das regularisierte Glättungsverfahren, die Anzahl an erfolgreichen Replikaten (# Repl.) sowie die Anzahl an Iterationen (# It.) und Funktionsauswertungen (# Eval.) bis zum Erreichen des Abbruchkriteriums für DGAFO und das schnellste Verfahren innerhalb von GROW für Optimierungsaufgabe 8 (siehe Abschnitt 3.5.3). Bei letzterem handelte es sich gemäß Tabelle 4.3 um das Polak-Ribière-Verfahren (PR). Es wurde die gleiche Gewichtung wie in Abschnitt 4.2 verwendet. Das DGAFO-Verfahren war im Durchschnitt mindestens genauso schnell wie GROW, im Falle einer Glättung mit dem Lipra-Verfahren sogar schneller. Es ist allerdings zu beachten, daß das DGAFO-Verfahren in vielen Fällen mit nur einer Iteration und 19 Funktionsauswertungen auskam. Es kann also zu deutlich weniger Rechenaufwand führen als das jeweils schnellste gradientenbasierte Verfahren.

Glättung basierend auf Gaußschen RBFs und dem MARS-Verfahren genauso hoch wie bei GROW, im Falle der Glättung mit der Lipra-Methode allerdings geringer. Dies täuscht jedoch über die Tatsache hinweg, daß das DGAFO-Verfahren im ersten Fall bei fünf von zehn Replikaten mit nur einer Iteration und 19 Funktionsauswertungen auskam. Da es zwei Replikate gab, bei denen 46 Evaluationen benötigt wurden, waren die Durchschnittswerte bei DGAFO und GROW gleich. Bei GROW lag die Anzahl an Evaluationen jedoch bei allen neun Replikaten sehr nahe am angegebenen Durchschnittswert. Im Falle der Lipra-Glättung waren neun von zehn Replikaten mit nur 19 und nur eines mit 37 Funktionsauswertungen zu verzeichnen. In vielen Fällen ist das DGAFO-Verfahren somit deutlich schneller als GROW. Ob das DGAFO-Verfahren auch bei Molekularen Simulationen schneller ist als GROW, wird in Abschnitt 7.3 diskutiert. Dort wird das DGAFO-Verfahren auch mit einer GROW-Optimierung, die effiziente Gradientenberechnungen gemäß Abschnitt 3.6.2 enthält, verglichen. Hier sei nur hypothetisch erwähnt, daß GROW bei einer Reduktion der durchschnittlichen Anzahl an Funktionsevaluationen von 28–29 um etwa den Faktor 2 auf 14–15 Simulationen bei einer effizienten Gradientenberechnung geringfügig schneller wäre als das DGAFO-Verfahren in beiden Fällen.

7.2.2 Vergleich mit gradientenbasierten Verfahren in der Nähe des Minimums

Im folgenden wird untersucht, ob das DGAFO-Verfahren näher an das Minimum gelangen kann als GROW. Auch hier wird wieder zwischen glatten und verrauschten Korrelationsfunktionen unterschieden. Für den verrauschten Fall wird ebenfalls ein Vergleich zu der Variante des Verfahrens nach Stoll angegeben. Analog zu Abschnitt 4.5.2 sei auch hier von vornherein erwähnt, daß anhand der Korrelationsfunktionen in diesem Abschnitt nicht eindeutig entschieden werden kann, welches Verfahren in der Nähe des Minimums besser geeignet ist. Dies wird am Beispiel einer Anwendung auf Molekulare Simulationen feststellbar sein, was in Abschnitt 7.3.2 nachgeholt wird.

Glatte Korrelationsfunktionen Tabelle 7.4 zeigt den Vergleich des DGAFO-Verfahrens mit GROW in der Nähe des Minimums im Falle der glatten Korrelationsfunktionen: Bei Optimierungsaufgabe 1 war der erreichte Fehlerfunktionswert im Falle des exakten Trust-Region-

Optimierungsaufgabe	Verfahren	# It.	# Eval.	$F(x^{\text{opt}})$	Spezielle Einstellungen
1	DGAFO	13	208	$7.4{\cdot}10^{-11}$	
	TR	51	791	$2{\cdot}10^{-12}$	
3	DGAFO	7	118	$9.5{\cdot}10^{-3}$	$\eta_1 = 0.2\ \eta_2 = 0.5,\ \Delta_0 = 0.2 \cdot \Delta_{\max}$
	PR	46	> 300	$9.5{\cdot}10^{-3}$	
5	DGAFO	13	172	$3.8{\cdot}10^{-7}$	$\Delta_0 = 0.1 \cdot \Delta_{\max}$
	TR	45	701	$2{\cdot}10^{-12}$	
7	DGAFO	25	> 200	$1.2{\cdot}10^{-2}$	$\eta_1 = 0.0, \eta_2 = 0.7, \gamma_1 = 0.1$
	PR	60	> 400	$1.2{\cdot}10^{-2}$	

Tabelle 7.4: Vergleich des DGAFO-Verfahrens mit GROW in der Nähe des Minimums im Falle der glatten Korrelationsfunktionen: Angegeben sind die Anzahl an Iterationen (# It.) und Funktionsauswertungen (# Eval.) bis zum Erhalt des erreichten Funktionswertes $F(x^{\text{opt}})$ für DGAFO und das jeweils beste Verfahren innerhalb von GROW für die Optimierungsaufgaben 1, 3, 5 und 7 (siehe Abschnitt 3.5.3). Es handelte sich hierbei um das Polak-Ribière-Verfahren (PR) und das Trust-Region-Verfahren mit exakter Lösung des Teilproblems (TR). Es wurde die Gewichtung $\forall_{i,T}\ w_{i,T} = 1$ verwendet. Weiterhin sind spezielle Einstellungen bezüglich der Verfahrensparameter angegeben. Die Standardeinstellungen waren $\zeta_A = 0.2,\ \eta_1 = 0.4,\ \eta_2 = 0.7,\ \gamma_1 = 0.5,\ \gamma_2 = 1.5$ und $\Delta_0 = 0.6 \cdot \Delta_{\max}$. GROW gelangte bei den Optimierungsaufgaben 1 und 5 näher an das Minimum heran als das DGAFO-Verfahren, letzteres benötigte jedoch stets weniger Iterationen und Funktionsauswertungen, um einen Fehlerfunktionswert in derselben Größenordnung zu erhalten. Die zu optimierenden Zielgrößen waren allerdings zum einen bereits in beiden Fällen innerhalb ihrer entsprechenden Toleranzwerte vorhergesagt worden, und zum anderen gelangte das DGAFO-Verfahren stets näher an das Minimum heran als die weniger rechenaufwendigen Verfahren SA, FR und PR. Bei den Optimierungsaufgaben 3 und 7 konnten von keinem Verfahren signifikante Verbesserungen erzielt werden. Das DGAFO-Verfahren war jedoch zum Erhalt minimaler Verbesserungen deutlich schneller als GROW.

Verfahrens um mehr als eine Größenordnung kleiner als beim DGAFO-Verfahren. Allerdings benötigte es auch viermal so viele Funktionsauswertungen. Um in den Bereich von $F(x) \approx 7 \cdot 10^{-11}$ zu gelangen, brauchte es ebenfalls deutlich mehr Evaluationen. Bei Optimierungsaufgabe 3 konnten keine signifikanten Verbesserungen erzielt werden. Das Abbruchkriterium war $F(x) \leq 10^{-2}$, und die erreichten Fehlerfunktionswerte waren in beiden Fällen etwa $9.5{\cdot}10^{-3}$. Allerdings benötigte das PR-Verfahren auch hier mehr als doppelt so viele Funktionsevaluationen wie das DGAFO-Verfahren. Der Parameter Δ_0 mußte beim DGAFO-Verfahren jedoch drastisch verkleinert werden. Im Falle von Optimierungsaufgabe 5 konnte GROW mit dem exakten TR-Verfahren einen um mehr als fünf Größenordnungen kleineren Fehlerfunktionswert erzielen, allerdings mit mehr als viermal soviel Rechenaufwand. Der Parameter Δ_0 mußte hierbei ebenfalls verkleinert werden. Bei Optimierungsaufgabe 7 ergaben sich ähnliche Resultate wie bei Optimierungsaufgabe 5: Es konnten weder mit GROW noch mit dem DGAFO-Verfahren signifikant kleinere Fehlerfunktionswerte nach Erhalt des Abbruchkriteriums erreicht werden. Das PR-Verfahren benötigte jedoch auch hier wieder etwa zweimal so viele Funktionsauswertungen. Das DGAFO-Verfahren erreichte die minimalen Verbesserungen allerdings nur, weil $\eta_1 = 0.0$ gesetzt wurde. Dies bedeutet, daß jede noch so geringfügige Verkleinerung des Fehlerfunktionswertes für eine neue Iteration akzeptiert wird. Aufgrund der Verkleinerung von γ_1 wurde auch das Vertrauensgebiet bei jedem TR-Schritt drastisch verkleinert, was ebenfalls nur zu sukzessiven minimalen Verbesserungen führen konnte.
Das TR-Verfahren mit exakter Lösung des TR-Teilproblems scheint aufgrund der vorliegenden Ergebnisse, zumindest für die Optimierungsaufgaben 1 und 5, näher an das Minimum zu gelangen als das DGAFO-Verfahren. Der Fehlerfunktionswert war allerdings in beiden Fällen so klein, daß alle zu optimierenden Zielgrößen innerhalb ihrer entsprechenden Toleranzwerte vorhergesagt wurden. Das DGAFO-Verfahren konnte jedoch stets mit erheblich weniger Rechenaufwand

	# It.	# Eval.	x^{opt}	ρ_l	p_σ	# It.	# Eval.	x^{opt}	ρ_l	p_σ
	Variante des Verfahrens nach Stoll					DGAFO-Verfahren, Gaußsche RBF/MARS				
$\bar{R}$	2–3	20–21	0.3713 0.3130 0.0229 0.1871	0.69% (0.09%)	1.89% (0.47%)	3–4	112-113	0.3700 0.3131 0.0241 0.1873	0.84% (0.17%)	1.83% (0.51%)
$R_{\min}$	4	23	0.3733 0.3124 0.0205 0.1845	0.51%	1.93%	1	73	0.3697 0.3122 0.0236 0.1853	0.62%	1.06%

Tabelle 7.5: Vergleich der Variante des Verfahrens nach Stoll mit dem DGAFO-Verfahren, kombiniert mit einer Glättung basierend auf Gaußschen RBFs und dem MARS-Verfahren, in der Nähe des Minimums. Angegeben sind Durchschnittswerte über zehn statistisch unabhängige Replikate ($\bar{R}$) und die Ergebnisse für den jeweils kleinsten MAPE-Wert bezüglich ρ_l ($R_{\min}$). Standardabweichungen sind in Klammern angegeben. Bezüglich Siedichte (ρ_l) und Dampfdruck (p_σ) sind jeweils die erreichten MAPE-Werte angezeigt. Die Ergebnisse der Variante des Verfahrens nach Stoll stammen aus Tabelle 4.12. Nur die Variante des Verfahrens nach Stoll gelangte äußerst robust näher an das Minimum heran als GROW (vergleiche Abschnitt 4.5.2). Anhand der hohen Anzahl an Funktionsevaluationen beim DGAFO-Verfahren ist zu erkennen, daß die erlangten minimalen Verbesserungen lediglich Zufallsergebnisse waren: Es waren in jeder Iteration so viele Trust-Region-Schritte erforderlich, daß die Größe des Vertrauensgebiets so gering wurde, daß nur noch Rauschen reproduziert werden konnte. Außerdem wurden die Kraftfeldparameter bei der Variante des Verfahrens nach Stoll bei allen Replikaten signifikant verändert, was beim DGAFO-Verfahren nicht der Fall war.

als das TR-Verfahren einen Fehlerfunktionswert in einer bestimmten Größenordnung erreichen, und es gelangte näher an das Minimum heran als die Methode des steilsten Abstiegs und die CG-Verfahren, welche weniger rechenaufwendig sind als das TR-Verfahren.

Verrauschte Korrelationsfunktionen Für den verrauschten Fall wurde wieder die andere Gewichtung verwendet. Das DGAFO-Verfahren wurde ausgehend von $x^{(0)}_{\mathrm{Stoll,TR}}$ aus Abschnitt 4.5.2 gestartet. Die Variante des Verfahrens nach Stoll und das exakte Trust-Region-Verfahren kombiniert mit Temperaturfits konnten von dort aus näher an das Minimum gelangen als das beste Verfahren innerhalb von GROW. Letzteres war in diesem Fall das Polak-Ribière-Verfahren, und $x^{(0)}_{\mathrm{Stoll,TR}}$ war die von diesem Verfahren erreichte Iteration. Da sich das Trust-Region-Verfahren sowohl in Abschnitt 4.5.2 als auch in Abschnitt 5.2.2 als zu rechenaufwendig herausgestellt hat, wird das DGAFO-Verfahren hier lediglich mit GROW und der Variante des Verfahrens nach Stoll verglichen. Tabelle 7.5 zeigt diesen Vergleich für die Kombination des DGAFO-Verfahrens mit einer Glättung basierend auf Gaußschen RBFs und dem MARS-Verfahren: Lediglich die Variante des Verfahrens nach Stoll konnte näher an das Minimum gelangen als GROW. Die hohe Anzahl an Funktionsauswertungen beim DGAFO-Verfahren, welche auf die hohe Anzahl an Trust-Region-Schritten pro Iteration zurückzuführen sind, deutet darauf hin, daß das DGAFO-Verfahren hier lediglich Zufallsergebnisse geliefert hat. Das Vertrauensgebiet wurde stets so klein, daß nur noch Rauschen reproduziert werden konnte, was auch daran zu erkennen ist, daß sich die Kraftfeldparameter im Laufe der Optimierung so gut wie gar nicht veränderten. Dieser Tatsache konnte auch nicht durch die Verwendung von Lipra als Glättungsmethode oder durch Erhöhung der Modellgenauigkeit innerhalb des initialen Vertrauensgebiets mittels Verwendung

eines größeren Dünn-Gitter-Levels ($\hat{\ell} = 3$) entgegengewirkt werden.
Es zeigt sich hier die in Abschnitt 6.4.4 diskutierte Problematik in Bezug auf den Verfahrensparameter Δ: Wird dieser zu groß gewählt, ist die Modellierung der Fehlerfunktion schlecht, und wird dieser zu klein gewählt, sind die Funktionswerte aufgrund der statistischen Unsicherheiten nicht mehr unterscheidbar. Die Existenz eines optimalen Δ kann nicht bewiesen werden, da sie von der Gestalt der Fehlerfunktion und dem Ausmaß an Rauschen abhängig ist. Gerade in der Nähe des Minimums, wo die Fehlerfunktion keinen klaren abfallenden Trend aufweist, kann sich dies als gravierend erweisen, was anhand des vorliegenden Beispiels zu erkennen ist. Eine schlechte Wahl des Parameters Δ wirkt sich also nicht nur negativ auf die Robustheit des DGAFO-Verfahrens aus, sondern kann in der Nähe des Minimums sogar dazu führen, daß der Algorithmus keine signifikanten Verbesserungen mehr erzielt. Es ist jedoch zu beachten, daß im vorliegenden Fall auch GROW mit einem Zufallsreplikat schon sehr nahe an das Minimum gelangt ist.

Fazit Wie bereits erwähnt, können hier keine endgültigen Schlußfolgerungen gezogen werden. Dazu ist eine Anwendung auf Molekulare Simulationen unentbehrlich (siehe Abschnitt 7.3.2). Es konnte lediglich folgendes festgestellt werden: Die Variante des Verfahrens nach Stoll war bei diesem Beispiel in der Nähe des Minimums deutlich robuster als das DGAFO-Verfahren.

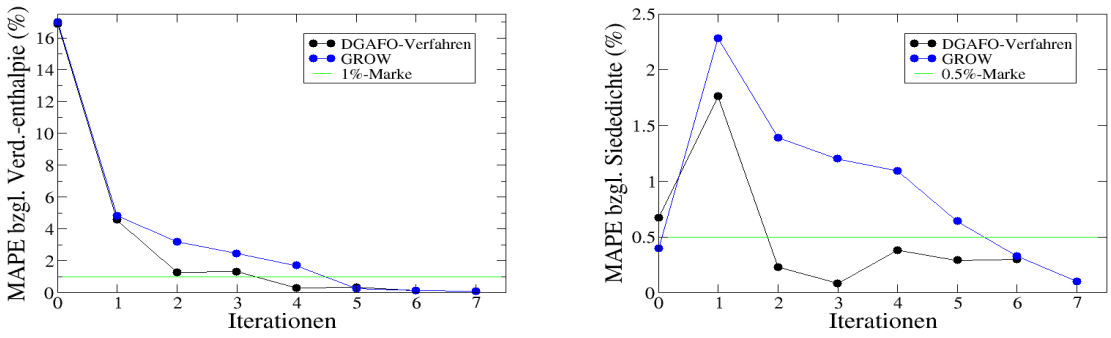
7.3 Anwendungen des DGAFO-Verfahrens auf Molekulare Simulationen

Abschließend wird das DGAFO-Verfahren anhand Molekularer Simulationen bewertet, und zwar unter denselben beiden Aspekten wie in Abschnitt 7.2: Abschnitt 7.3.1 vergleicht das Verfahren mit GROW in Bezug auf Rechenaufwand, und Abschnitt 7.3.2 analysiert, wie weit es im Gegensatz zu anderen Verfahren an das Minimum gelangt. Anhand dieser Evaluationen werden sich konkretere Aussagen ableiten lassen als im Falle der Korrelationsfunktionen in Abschnitt 7.2.

7.3.1 Vergleich mit gradientenbasierten Verfahren bezüglich Rechenaufwand: Benzol und Ethylenoxid

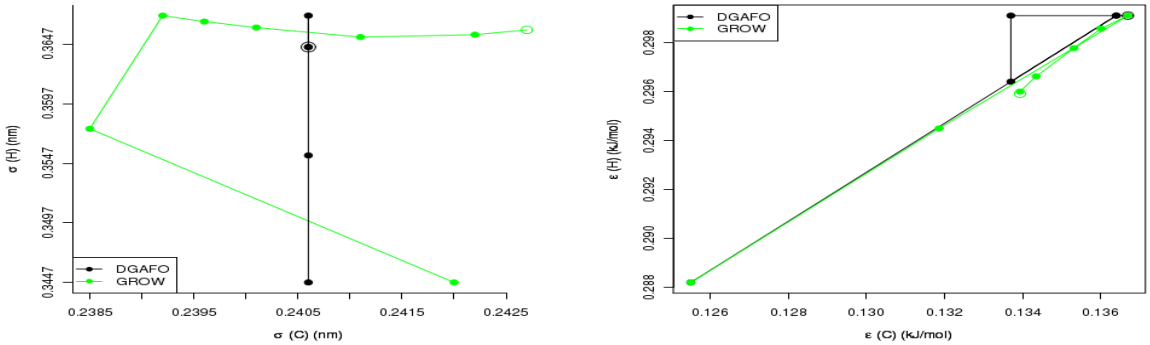
In diesem Abschnitt wird das DGAFO-Verfahren mit GROW bezüglich Rechenaufwand anhand von zwei Applikationen verglichen: Benzol und Ethylenoxid.

Evaluation anhand von Benzol Abbildung 7.5 zeigt den Vergleich im Falle von Benzol. Die GROW-Ergebnisse sind Abschnitt 5.1.2 entnommen, wobei die zu optmierenden Zielgrößen Verdampfungsenthalpie (Abbildung 7.5(a)) und Siededichte (Abbildung 7.5(b)) waren. Sie stammen somit aus Tabelle 5.3. Für das DGAFO-Verfahren wurden die folgenden Einstellungen



(a) (b)

Abbildung 7.5: MAPE-Werte bezüglich $\Delta_v H$ (a) und ρ_l (b) im Falle von Benzol im Laufe der Optimierung mithilfe des DGAFO-Verfahrens im Vergleich zu GROW. Die Glättung basierte auf Gaußschen RBFs kombiniert mit dem MARS-Algorithmus. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{H}), \sigma(\mathrm{C}), \varepsilon(\mathrm{H})$ und $\varepsilon(\mathrm{C})$. Die schnellere Konvergenz des DGAFO-Verfahrens konnte bestätigt werden.



(a) (b)

Abbildung 7.6: Entwicklung der LJ-Parameter im Falle von Benzol ($\Delta_v H$,ρ_l) für GROW und DGAFO: $\sigma(\mathrm{H})$ und $\sigma(\mathrm{C})$ (a) sowie $\varepsilon(\mathrm{H})$ und $\varepsilon(\mathrm{C})$ (b). Die unausgefüllten Kreise zeigen die optimalen Parameter an. Das DGAFO-Verfahren führte auf direkterem Wege zum Minimum als GROW. Nur im Falle von ε sind aufgrund des Dreiecks einige Umwege zu erkennen.

gewählt: $\eta_1 = 0.2$, $\eta_2 = 0.7$, $\gamma_1 = 0.5$, $\gamma_2 = 1.1$ und $\Delta_0 = 0.3 \cdot \Delta_{\max}$. Die zu optimierenden Kraftfeldparameter waren $\sigma(\mathrm{H}), \sigma(\mathrm{C}), \varepsilon(\mathrm{H})$ und $\varepsilon(\mathrm{C})$, und als zulässiger Bereich wurde $\Omega = (30, 80)$ gewählt. Da es sich hierbei um ein vierdimensionales Optimierungsproblem handelte, wurden Gaußsche RBFs für die Glättung und der MARS-Algorithmus für die Regularisierung verwendet.

GROW benötigte für die Optimierung insgesamt sieben Iterationen. Dazu wurden fünf Iterationen mit der Methode des steilsten Abstiegs und zwei Iterationen mit dem Polak-Ribière-Verfahren durchgeführt. Das DGAFO-Verfahren kam hingegen mit nur sechs Iterationen aus, wobei hier zu beachten ist, daß nach nur vier Iterationen bereits ein in Bezug auf $\Delta_v H$ und ρ_l optimales Kraftfeld vorlag, was bei GROW erst nach sieben Iterationen der Fall war. Hierzu wa-

ren beim DGAFO-Verfahren 37 und bei GROW 62 Simulationen erforderlich. Bei letzterem sind allerdings sieben Simulationen abzuziehen, da in Abschnitt 5.1.2 auch die Partialladungen mitoptimiert wurden. Es handelte sich dabei also um ein fünfdimensionales Optimierungsproblem, und es war in jeder Iteration eine Simulation mehr für die Gradientenberechnung erforderlich. Das DGAFO-Verfahren war mit vier Iterationen und 37 Simulationen im Gegensatz zu GROW, welches sieben Iterationen und 55 Simulationen benötigte, deutlich schneller.
Abbildung 7.6 gibt die Entwicklung der LJ-Parameter (Abbildung 7.6(a) bezieht sich auf σ und Abbildung 7.6(b) auf ε) im Vergleich zu GROW an. Es waren die gleichen Tendenzen festzustellen wie in Tabelle 5.3: $\sigma(\mathrm{C})$ blieb konstant, und alle anderen Kraftfeldparameter wurden vergrößert. In der vierten Iteration des DGAFO-Verfahrens waren die Parameter denen in der siebten Iteration von GROW sehr ähnlich. Es ist allerdings die folgende interessante Feststellung zu verzeichnen: Der Parameter $\sigma(\mathrm{C})$ blieb beim DGAFO-Verfahren sogar während der gesamten Optimierung konstant. Bei der GROW-Optimierung hingegen wurde er zunächst verkleinert, um dann wieder fast denselben Wert wie zu Beginn zu erreichen. Aufgrund der Gradienteninformation neigt GROW dazu, Umwege zu gehen. Dazu wurde in Abschnitt 5.1 eine Vielzahl an Beispielen gefunden. Dadurch, daß das DGAFO-Verfahren gitterbasiert ist, können ein oder mehrere Parameter während der gesamten Optimierung konstant bleiben, denn das Verfahren kann entlang einer bestimmten Gitterlinie oder -hyperebene konvergieren. Dadurch wurden Umwege vermieden, was als weiterer Grund für die schnellere Konvergenz interpretiert werden kann. Nur im Falle von ε sind beim DGAFO-Verfahren einige kleine Umwege zu erkennen. Das Verfahren ist komplett durch das Dreieck in Abbildung 7.6(b) gelaufen. Das DGAFO-Verfahren lieferte etwas andere Kraftfeldparameter als GROW, gelangte also in einen anderen Bereich in der Nähe des Minimums. Der Vergleich zu einer effizienten Gradientenberechnung liegt im Falle von Benzol nicht vor. Dieser wird im folgenden anhand von Epoxid angegeben.

Evaluation anhand von Ethylenoxid Abbildung 7.7 zeigt die entsprechenden MAPE-Werte für Epoxid im Laufe der beiden Optimierungsprozesse. Die GROW-Ergebnisse sind Abschnitt 5.3.3 entnommen, wobei die zu optmierenden Zielgrößen Siededichte (Abbildung 7.7(a)), Verdampfungsenthalpie (Abbildung 7.7(b)) und Dampfdruck (Abbildung 7.7(c)) waren. Es handelte sich um VLE-Simulationen (vergleiche Anhang A.5), basierend auf dem Dipolmodell nach Eckl u. a. (2008a). Die Ergebnisse stammen aus Tabelle 5.20. Auch die Ergebnisse des DGAFO-Verfahrens basierten auf dem Dipolmodell. Dabei wurden dieselben Verfahrensparameter wie bei Benzol gewählt. Die zu optimierenden Kraftfeldparameter waren $\varepsilon(\mathrm{CH_2}), \varepsilon(\mathrm{O}), \sigma(\mathrm{CH_2})$ und $\sigma(\mathrm{O})$, und als zulässiger Bereich wurde $\Omega = (20, 20)$ gewählt. Da es sich hierbei um ein vierdimensionales Optimierungsproblem handelte, wurden erneut Gaußsche RBFs für die Glättung und der MARS-Algorithmus für die Regularisierung verwendet.

GROW benötigte für die Optimierung insgesamt 14 Iterationen mit der Methode des steilsten Abstiegs. Die anschließende Optimierung mit der Variante des Verfahrens nach Stoll (vergleiche Abschnitt 5.3.3) wurde für den Vergleich bezüglich Konvergenzgeschwindigkeit nicht betrachtet. Ein Vergleich des DGAFO-Verfahrens mit der Variante des Verfahrens nach Stoll bezüglich des Verhaltens in der Nähe des Minimums wird in Abschnitt 7.3.2 angegeben. Das DGAFO-Verfahren kam mit nur fünf Iterationen aus, um einen vergleichbaren Fehlerfunktionswert in derselben Größenordnung wie GROW zu erzielen: Im Falle von GROW galt $F(x^{(14)}) = 2.4 \cdot 10^{-4}$

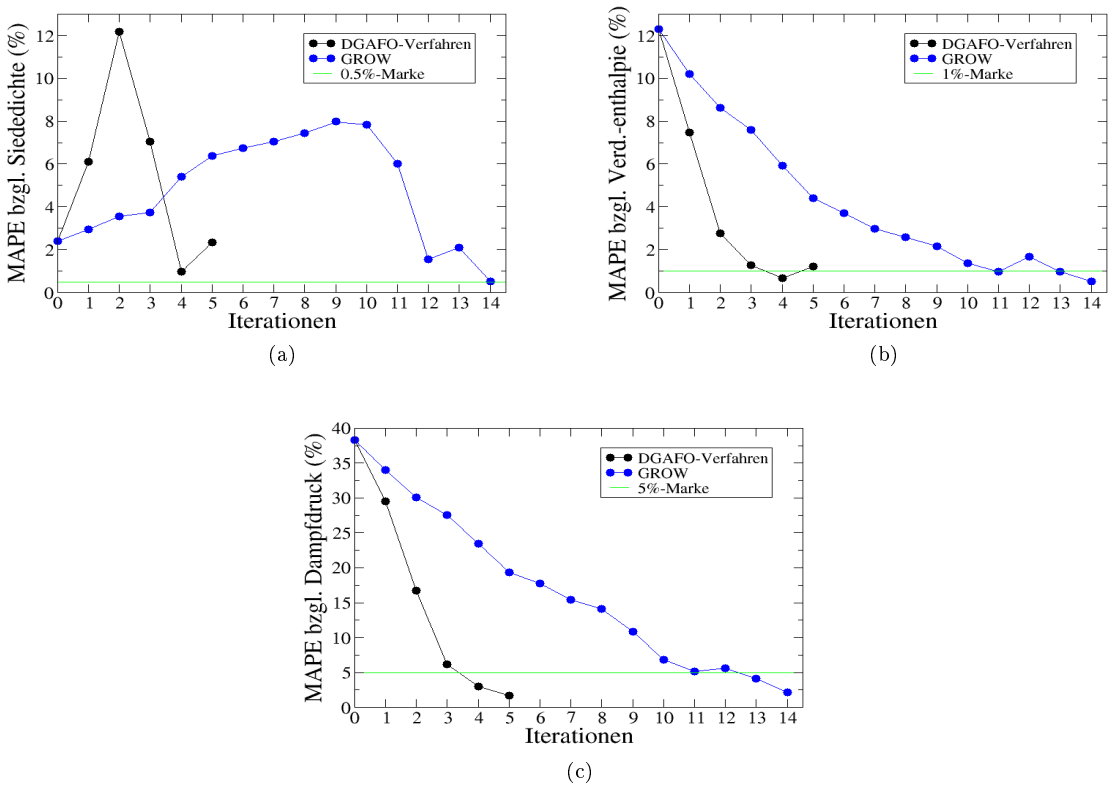
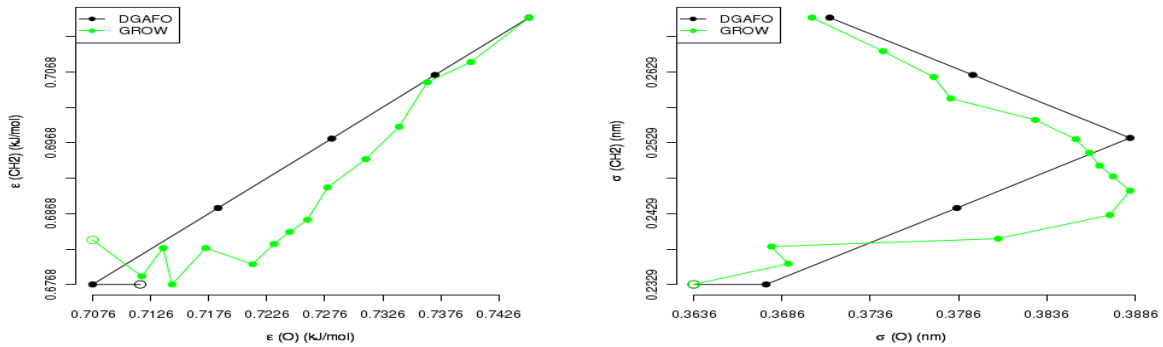


Abbildung 7.7: MAPE-Werte bezüglich ρ_l (a), $\Delta_v H$ (b) und p_σ (c) im Falle von Epoxid im Laufe der VLE-Optimierung mithilfe des DGAFO-Verfahrens im Vergleich zu GROW. Die Glättung basierte auf Gaußschen RBFs kombiniert mit dem MARS-Algorithmus. Die zu optimierenden Kraftfeldparameter waren $\varepsilon(\mathrm{CH_2}), \varepsilon(\mathrm{O}), \sigma(\mathrm{CH_2})$ und $\sigma(\mathrm{O})$. Die schnellere Konvergenz des DGAFO-Verfahrens konnte erneut bestätigt werden.

und im Falle des DGAFO-Verfahrens $F(x^{(5)}) = 3.9 \cdot 10^{-4}$. Um die entsprechenden Fehlerfunktionswerte zu erreichen, waren beim DGAFO-Verfahren 54 und bei GROW 77 Simulationen erforderlich. Das DGAFO-Verfahren war also mit nur fünf Iterationen und 54 Funktionsauswertungen erneut deutlich schneller als GROW.
Abbildung 7.8 gibt die Entwicklung der LJ-Parameter (Abbildung 7.8(a) bezieht sich auf ε und Abbildung 7.8(b) auf σ) im Vergleich zu GROW an. Es waren die gleichen Tendenzen festzustellen wie in Tabelle 5.20: Alle Kraftfeldparameter wurden verkleinert. Das DGAFO-Verfahren lieferte nahezu dieselben Kraftfeldparameter wie GROW, gelangte also in denselben Bereich in der Nähe des Minimums. Erneut sind durch das DGAFO-Verfahren unnötige Umwege vermieden worden. Anhand der vom DGAFO-Verfahren gelieferten Kurven in Abbildung 7.8 ist deutlich zu erkennen, daß das Verfahren dazu in der Lage ist, die Wege, die GROW benötigt, zu linearisieren. Allerdings machte das DGAFO-Verfahren im Falle von σ denselben Umweg wie GROW, was jedoch auch auf die Gestalt der Fehlerfunktion zurückgeführt werden kann.
Die in dieser Arbeit ebenfalls entwickelte effiziente Gradientenberechnung konnte den Rechenaufwand von GROW auf 13 Iterationen und 30 Simulationen reduzieren (vergleiche Abschnitt



(a) (b)

Abbildung 7.8: Entwicklung der LJ-Parameter im Falle von Epoxid (VLE) für GROW und DGAFO: $\varepsilon(CH_2)$ und $\varepsilon(O)$ (a) sowie $\sigma(CH_2)$ und $\sigma(O)$ (b). Die unausgefüllten Kreise zeigen die finalen Parameter an. Das DGAFO-Verfahren führte erneut auf direkterem Wege zum Minimum als GROW.

5.3.3). Das DGAFO-Verfahren benötigte zwar deutlich mehr Simulationen, allerdings ist zu beachten, daß zum einen die von GROW gemachten Umwege auch bei einer effizienten Gradientenberechnung nicht vermieden werden können und zum anderen die Einstellung der Verfahrensparameter, die für eine effiziente Gradientenberechnung notwendig sind, von der Gestalt der Fehlerfunktion abhängig und somit viel schwieriger zu realisieren ist.

Fazit Zusammenfassend läßt sich sagen, daß das DGAFO-Verfahren eine deutlich höhere Konvergenzgeschwindigkeit aufweist als GROW. Eine effiziente Gradientenberechnung kann zwar eine geringere Anzahl an Simulationen liefern als das DGAFO-Verfahren, allerdings nur, wenn sie robust anwendbar und geeignet parametrisiert ist.

7.3.2 Vergleich mit gradientenbasierten Verfahren in der Nähe des Minimums: Dipropylenglykoldimethylether

In Abschnitt 7.2.2 konnte anhand der Korrelationsfunktionen aus den dort angegebenen Gründen nicht eindeutig festgestellt werden, ob das DGAFO-Verfahren tatsächlich dazu in der Lage ist, näher an das Minimum heranzugelangen als GROW beziehungsweise die Variante des Verfahrens nach Stoll. Dies soll im folgenden anhand von Molekularen Simulationen exemplarisch untersucht werden: Wie in Abschnitt 5.2.2 dargestellt, konnte GROW im Falle von Dipropylenglykoldimethylether kein optimales Kraftfeld für die Flüssigdichte ρ erhalten. Die Variante des Verfahrens nach Stoll hingegen konnte nach drei Iterationen optimale Flüssigdichten erzielen. Abbildung 7.9 zeigt, daß das DGAFO-Verfahren mit $\Delta_0 = 0.3 \cdot \Delta^{\max}$ hierzu nur zwei Iterationen benötigte. Da es sich im vorliegenden Fall um ein achtdimensionales Optimierungsproblem handelt, wurde die Lipra-Methode zur Glättung eingesetzt. Für das DGAFO-Verfahren war der Rechenaufwand allerdings mit 38 Simulationen mehr als dreimal so hoch wie für die Variante des Verfahrens nach Stoll, für das nur 12 Simulationen erforderlich waren. Bei optimaler Selektion von Δ_0 hätte der Rechenaufwand im besten Fall auf eine Iteration und 19 Simulationen reduziert werden können. Aufgrund der effizienten Gradientenberechnung für die Zielgrößen, welche

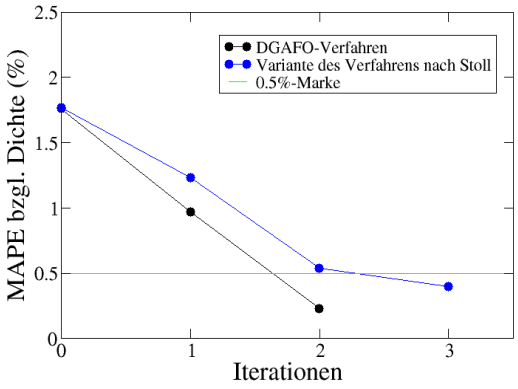


Abbildung 7.9: MAPE-Werte bezüglich ρ im Falle von Dipropylenglykoldimethylether im Laufe der Optimierung mithilfe des DGAFO-Verfahrens im Vergleich zu der Variante des Verfahrens nach Stoll. Die Glättung wurde mithilfe der Lipra-Methode realisiert. Die zu optimierenden Kraftfeldparameter waren $\sigma(CH_3)$, $\sigma(CH_2)$, $\sigma(CH)$ und $\sigma(O)$ sowie $\varepsilon(CH_3)$, $\varepsilon(CH_2)$, $\varepsilon(CH)$ und $\varepsilon(O)$. Das DGAFO-Verfahren benötigte lediglich zwei Iterationen aber deutlich mehr Simulationen als die Variante des Verfahrens nach Stoll.

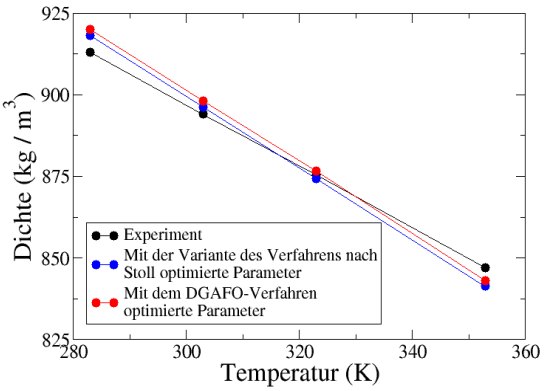


Abbildung 7.10: Optimierung von ρ im Falle von Dipropylenglykoldimethylether mit der Variante des Verfahrens nach Stoll und dem DGAFO-Verfahren. Mit dem DGAFO-Verfahren konnte ein geringfügig besseres Kraftfeld erzielt werden, allerdings mit deutlich mehr Rechenaufwand.

auch noch in der Nähe des Minimums angewandt werden kann, ist die Variante des Verfahrens nach Stoll die schnellste Methode, um das Minimum möglichst exakt zu detektieren.

Allerdings ist diese auch nur in der Nähe des Minimums verwendbar, wohingegen das DGAFO-Verfahren für Startparameter eingesetzt werden kann, die weiter vom Minimum entfernt liegen, das heißt in deren Umgebung die Fehlerfunktion nicht notwendigerweise quadratisch approximierbar ist. Ob die effiziente Gradientenberechnung immer funktioniert, ist in der Praxis im allgemeinen nicht beweisbar, allerdings auch nicht die Existenz eines optimalen Δ für das DGAFO-Verfahren. Wird der Gradient jedoch falsch bestimmt, so wird die Variante des Verfahrens nach Stoll auch nach einer Verkleinerung des Vertrauensgebiets keine Verbesserungen mehr liefern. Beim DGAFO-Verfahren hingegen besteht diese Möglichkeit noch.

Abbildung 7.10 zeigt die optimalen Flüssigdichten in Abhängigkeit von der Temperatur für die Variante des Verfahrens nach Stoll und das DGAGO-Verfahren: Erstere lieferte keine optimalen Dichten für die höchste und die niedrigste Temperatur. Beim DGAFO-Verfahren hingegen wich die Dichte nur im Falle der niedrigsten Temperatur, bei der eine Molekulare Simulation am schwierigsten durchzuführen ist, um mehr als 0.5% vom Experiment ab. Das bedeutet, daß das DGAFO-Verfahren für Dipropylenglykoldimethylether ein etwas besseres Kraftfeld erzielen konnte als die Variante des Verfahrens nach Stoll.
Zusammenfassend läßt sich sagen, daß das DGAFO-Verfahren mit geeigneter Glättung und geeigneter Wahl von Δ_0 näher an das Minimum gelangen kann als GROW. Aus den Ergebnissen des vorliegenden Abschnitts und denen aus Abschnitt 7.2.2 ist allerdings zu schließen, daß die Variante des Verfahrens nach Stoll in der Nähe des Minimums in Bezug auf Geschwindigkeit und Robustheit besser geeignet ist, sofern sie einsetzbar ist.

7.4 Finale Evaluation des DGAFO-Verfahrens

Abschließend kann das DGAFO-Verfahren inklusive Glättung und Regularisierung bezüglich der drei Kriterien Geschwindigkeit, lokale Verfeinerungen und Robustheit wie folgt bewertet werden:

1. **Geschwindigkeit:** Bei geeigneter Wahl von Δ_0 konnte das DGAFO-Verfahren stets mit deutlich weniger, oftmals nur halb so vielen Funktionsauswertungen auskommen als das jeweils beste gradientenbasierte Verfahren innerhalb von GROW, um in die Nähe eines lokalen Minimums zu gelangen. Desweiteren waren stets weniger Iterationen erforderlich, was auf die Wahl von Δ_0 und die Möglichkeit zur Änderung von nur einem oder wenigen Parametern zurückzuführen ist. Die effiziente Gradientenberechnung aus Abschnitt 3.6.2 konnte den Rechenaufwand in einigen Fällen zwar weiterhin reduzieren, allerdings ist diese nicht immer problemlos verwendbar. Die Wahl der Konstanten innerhalb dieser Methodik ist abhängig von der Problemstellung und der Gestalt der zu minimierenden Funktion.
   Die Wahl $\Delta_0 = 0.3 \cdot \Delta^{\max}$ hat sich beim DGAFO-Verfahren meistens als geeignet erwiesen, das heißt, die Parametrisierung des DGAFO-Verfahrens ist zu Beginn der Optimierung wesentlich unkritischer.
2. **Lokale Verfeinerungen:** Auch hier konnte das DGAFO-Verfahren überzeugen: Wurde die Größe des Vertrauensbereichs geeignet gewählt, so konnte der Algorithmus näher an das Minimum gelangen als GROW. Allerdings wird die Wahl von Δ in diesem Bereich äußerst kritisch, was das Verfahren in der Nähe des Minimums unrobust macht: Wird Δ zu groß gewählt, kann durch Glättung und Interpolation kein geeignetes Modell für die Fehlerfunktion gefunden werden, und wird es zu klein gewählt, wird nur noch Rauschen reproduziert. Die Variante des Verfahrens nach Stoll ist an dieser Stelle besser geeignet, zumal sie auch mit sehr wenigen Funktionsevaluationen auskommt. Allerdings besteht auch hier wieder die Problematik der effizienten Gradientenberechnung. Wird der Gradient zu ungenau berechnet, so kann das Verfahren keine Verbesserungen mehr erzielen, wohingegen beim DGAFO-Verfahren hierzu noch Möglichkeiten bestehen. Die Problematik der Regenrinnengestalt bleibt bei allen Verfahren bestehen.

3. **Robustheit:** Das DGAFO-Verfahren weist eine etwas geringere Robustheit auf als die in dieser Arbeit eingesetzten gradientenbasierten Verfahren, was ebenfalls an der Wahl der Größe des Vertrauensgebiets liegt: Bei ungeeignetem Δ kann aufgrund des Rauschens das Minimum des Modells unter Umständen signifikant verschoben werden, wodurch sich der Verlauf des Verfahrens ebenfalls verändert. Der Verfahrensparameter muß so gewählt werden, daß ein klarer abfallender Trend der Fehlerfunktion innerhalb des Vertrauensgebiets besteht, welcher durch das Modell wiedergegeben werden kann. Dies ist jedoch bei der vorliegenden Problemstellung keineswegs trivial, da die Gestalt der Fehlerfunktion in der Regel nicht bekannt ist. Andererseits konnte das DGAFO-Verfahren erfolgreich auf Fehlerfunktionen angewandt werden, deren Glattheit nicht vorausgesetzt und deren Gestalt vor der Optimierung nicht abgeschätzt werden kann.

Es ist äußerst schwierig, ein Verfahren zu finden, welches sowohl effizient als auch robust ist. Oftmals stehen sich diese beiden Eigenschaften gegenüber: Stochastische globale Optimierungsverfahren sind sehr robust in Bezug auf Rauschen, benötigen aber einen enormen Rechenaufwand, um das globale Minimum nahezu exakt zu bestimmen. Gradientenbasierte Verfahren zeichnen sich durch gutes Konvergenzverhalten aus, sind allerdings weniger robust und auf die Differenzierbarkeit der zu minimierenden Funktion angewiesen. Das DGAFO-Verfahren steht zwischen diesen Extremen: Die Effizienzerhöhung ging etwas auf Kosten der Robustheit. Sämtliche Anwendungen haben jedoch gezeigt, daß es für die vorliegende Problemstellung absolut geeignet ist. Daher ist es gradientenbasierten Verfahren vor allem zu Beginn der Optimierung vorzuziehen, denn die Reduktion des Rechenaufwands ist bei Molekularen Simulationen von entscheidender Bedeutung, und das DGAFO-Verfahren ist nicht derart unrobust, als daß es nur selten zum Ziel führen würde. Es kann unter gewissen Voraussetzungen sogar dann noch eingesetzt werden, wenn GROW keine Verbesserungen mehr liefert. Aus all diesen Gründen läßt sich abschließend sagen, daß mit dem DGAFO-Verfahren ein für die Problemstellung in dieser Arbeit äußerst geeignetes ableitungsfreies Verfahren gefunden werden konnte.

8 Diskussion und Ausblick

8.1 Zusammenfassung und Diskussion

In dieser Dissertation wurden mithilfe von klassischen und neuartigen numerischen Optimierungsverfahren Kraftfelder für Molekulare Simulationen auf effiziente und robuste Art und Weise parametrisiert. Abbildung 1.1 veranschaulicht die hier realisierte Verknpüfung zwischen Simulation und Optimierung sowie die hier eingesetzten und entwickelten Verfahren. Die Notwendigkeit von automatisierten Optimierungsprozessen, welche auf der Lösung eines mathematischen Minimierungsproblems basieren, wurde dabei eingehend motiviert. Bei der Formulierung des Minimierungsproblems wurde eine gewichtete quadratische Fehlerfunktion zwischen simulierten und experimentellen physikalischen Zielgrößen betrachtet, welche nicht analytisch darstellbar ist. Da die Parametrisierung von Kraftfeldern ein eigenständiges Forschungsfeld bildet und nur wenige Autoren Ansätze zur Lösung dieses Minimierungsproblems aufgestellt haben, bestand die Aufgabenstellung der Dissertation darin, die noch vorhandene große Lücke teilweise zu füllen. Da Molekulare Simulationen äußerst rechenaufwendig und die daraus resultierenden physikalischen Zielgrößen stets mit statistischem Rauschen behaftet sind, war insbesondere auf die Effizienz und Robustheit der einzusetzenden numerischen Verfahren Wert zu legen. Effizienz bedeutet dabei, daß die Verfahren mit möglichst wenig Simulationen auskommen und Robustheit, daß sie mit einem gewissen Ausmaß an statistischem Rauschen umgehen können müssen. In dieser Dissertation sind einige neuartige Verfahren entwickelt worden, welche beide Eigenschaften im Vergleich zu von anderen Autoren eingesetzten Verfahren deutlich verbessern konnten. Da aus Komplexitätsgründen nur eine geringe Anzahl an physikalischen Zielgrößen simultan zu verschiedenen Temperaturen und Drücken angepaßt werden konnten, waren die aus der Optimierung resultierenden Kraftfelder in Bezug auf andere Eigenschaften zu evaluieren. Die Diskussion der Optimierungsprozesse bezüglich ihrer generischen Anwendbarkeit spielte somit in dieser Dissertation ebenfalls eine sehr wichtige Rolle.

Methodische und mathematische Aspekte Ein Ansatz bestand darin, gradientenbasierte Verfahren zu verwenden, da diese im allgemeinen gute Konvergenzeigenschaften aufweisen. Diese sind in zwei Kategorien einzuteilen: Zum einen wurden Abstiegsmethoden betrachtet, darunter die Methode des steilsten Abstiegs, (Quasi-)Newton-Verfahren und Konjugierte Gradienten. Zum anderen wurden Trust-Region-Verfahren eingesetzt, bei denen die zu minimierende Fehlerfunktion in einem Vertrauensgebiet gewisser Größe quadratisch approximiert wurde. Die Minimierung des quadratischen Modells innerhalb des Vertrauensgebiets wurde dabei auf zwei verschiedene Arten gelöst, zum einen mit dem Verfahren des Doppelten Hundebeins und zum anderen nahezu exakt durch eine Eigenwertzerlegung der Hesse-Matrix.
Bei allen Verfahren wurden Gradient und Hesse-Matrix durch finite Differenzen erster Ordnung approximiert. Auf Diskretisierungen höherer Ordnung wurde aus Komplexitätsgründen verzich-

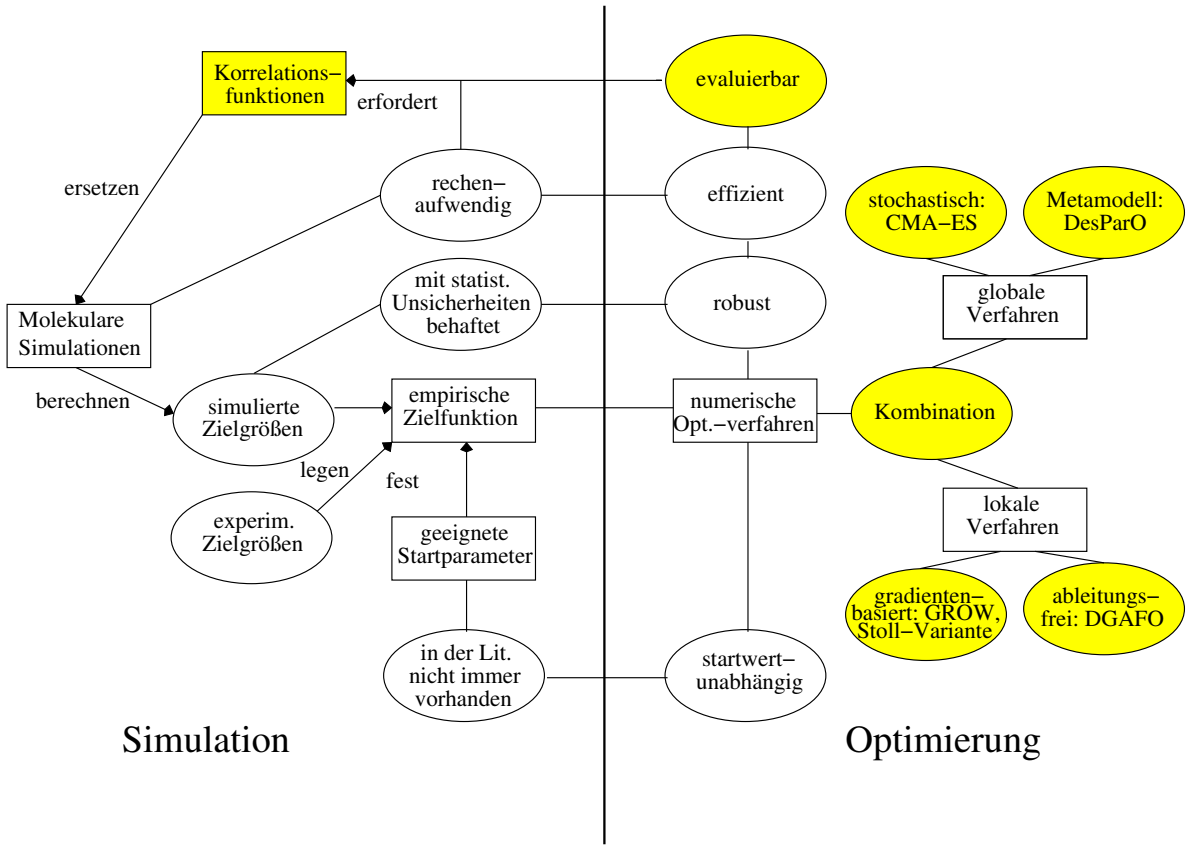


Abbildung 8.1: In dieser Arbeit realisierte Verknüpfung zwischen Simulation und Optimierung. Die gelb markierten Felder zeigen die gewonnen Erkenntnisse sowie die hier eingesetzten und entwickelten Verfahren zur verbesserten Parametrisierung von Kraftfeldern.

tet. Die Diskretisierungsschrittweite mußte aufgrund des statistischen Rauschens relativ hoch gewählt werden, damit die Verfahren nicht gegen durch Oszillationen hervorgerufene intermediäre lokale Minima konvergierten. Dadurch und mithilfe geeigneter Äquilibrierungskriterien für die Simulationen konnte die robuste Anwendbarkeit von gradientenbasierten Verfahren realisiert werden. Eine detaillierte theoretische Analyse, wie sich das statistische Rauschen auf die Fehlerfunktion und deren Gradienten auswirkt, wurde ebenfalls durchgeführt. Dabei ergab sich, daß das Rauschen in geringem Ausmaß und unter gewissen Voraussetzungen an die Fehlerfunktion tolerabel ist. Die einzige problematische Voraussetzung für die Konvergenz der Verfahren war die positive Definitheit der Hesse-Matrix, welche bei den hier interessierenden Anwendungen nicht garantiert werden kann. Somit erwiesen sich lediglich die Methode des steilsten Abstiegs, Konjugierte Gradienten und das Trust-Region-Verfahren mit exakter Lösung des Teilproblems als für die vorliegende Problemstellung geeignet. Alle hier betrachteten gradientenbasierten Verfahren wurden im Rahmen dieser Dissertation in einer am Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen (SCAI) vom Autor entwickelten Software namens **GR***adient-based* **O***ptimization* **W***orkflow (GROW)* implementiert.
Da für jede Iteration innerhalb eines Optimierungsprozesses ein Gradient und beim Trust-Region-Verfahren auch eine Hesse-Matrix zu bestimmen ist, wurden diese zur Erhöhung der

Effizienz mithilfe von Richtungsableitungen berechnet. Dabei wurde auf Fehlerfunktionswerte bereits durchgeführter Simulationen zurückgegriffen. Unter gewissen Voraussetzungen konnte dabei im Falle einer derartigen, in dieser Arbeit entwickelten effizienten Gradientenberechnung die Anzahl an durchzuführenden Simulationen um den Faktor 2–4 und im Falle einer effizienten Hesse-Matrix-Berechnung um den Faktor 1.5 reduziert werden. Dies ist als enormer Erfolg anzusehen, da die Durchführung einer Molekularen Simulation auch auf hochmodernen Rechenclustern im Bereich von Stunden oder sogar Tagen liegen kann. In der Nähe des Minimums sind die entsprechenden Voraussetzungen allerdings nicht mehr erfüllt, so daß diese Methodik lediglich zu Beginn des Optimierungsprozesses, das heißt in steilen Bereichen der Fehlerfunktion, anwendbar ist.

Bei sämtlichen gradientenbasierten Verfahren handelt es sich um lokale Optimierer. Da jedoch nicht für jedes Optimierungsproblem geeignete initiale Kraftfeldparameter aus der Literatur zu finden sind, waren auch globale Optimierer in Betracht zu ziehen. Auch diese sind in zwei Kategorien einzuteilen: Zum einen gibt es stochastisch basierte Verfahren, welche auf Zufallssuchen basieren, und zum anderen Algorithmen, die Metamodelle der zu minimierenden Funktion erstellen. Aus der ersten Kategorie wurde ein äußerst robuster evolutionärer Algorithmus namens **C***ovariance* **M***atrix* **A***daptation* **E***volution* **S***trategy (CMA-ES)* gewählt, und aus der zweiten Kategorie die am Fraunhofer-Institut SCAI entwickelte **Des***ign* **Par***ameter* **O***ptimisation Toolbox (DesParO)*, ein Interpolationsverfahren, basierend auf multikriterieller Optimierung. Bei keinem der beiden Verfahren konnte mit akzeptablem Rechenaufwand die Konvergenz gegen ein globales Minimum garantiert werden. Es ist dagegen stets möglich, daß der Einzugsbereich eines lokalen Minimums, auch bei großen statistischen Unsicherheiten, erreicht wird. Hierbei ist für CMA-ES die Wahl des Abbruchkriteriums einfacher als für DesParO, insofern ist es DesParO vorzuziehen.
Weiterhin wurde in der vorliegenden Dissertation ein Kombinationsalgorithmus vorgeschlagen. Dieser wird durch einen Filterungsprozeß beschrieben, bei dem zunächst globale Optimierungsverfahren eingesetzt werden, die den Parameterraum auf effiziente und robuste Art abtasten. Diese sollen mit möglichst geringem Rechenaufwand in den Einzugsbereich eines globalen Minimums gelangen, wo lokale Optimierer wie gradientenbasierte Verfahren effizienter sind. Die globalen Optimierungsverfahren sind also geeignet abzubrechen, so daß die erhaltenen Parameter als Startparameter für lokale Verfahren verwendet werden können. Globale Optimierungsverfahren sind jedoch aufgrund des höheren Rechenaufwands nur dann vorzuschalten, wenn keine geeigneten Startparameter zum Beispiel aus der Literatur oder durch chemische Intuition erzielt werden können. Ein Nachteil von gradientenbasierten Verfahren ist jedoch ihr Verhalten in der Nähe des Minimums: Einerseits können die Richtung des Gradienten oder die Hesse-Matrix aufgrund des statistischen Rauschens dort komplett falsch berechnet werden, und andererseits führt die hier eingesetzte funkionsangepaßte Armijo-Schrittweitensteuerung auch bei korrekt berechneter Abstiegsrichtung äußerst langsam zu nur sehr kleinen Verbesserungen. Auch eine effiziente Gradienten- und Hesse-Matrix-Berechnung ist in der Nähe des Minimums nicht mehr möglich. Daher ist die Fehlerfunktion in diesem Bereich geeignet zu modellieren, was in einem ersten Schritt durch Temperaturfits erfolgte. Dabei wurden die betrachteten physikalischen Zielgrößen in Abhängigkeit von der Temperatur durch glatte Funktionen beschrieben. Da das exakte Trust-Region-Vefahren trotz verrauschter Hesse-Matrix oftmals näher an das Minimum gelangte als alle anderen gradientenbasierten Verfahren, wurde dieses in Kombination mit Temperaturfits weiterhin in Betracht gezogen. Zum Vergleich wurde das Verfahren nach Stoll

verwendet, ebenfalls in Kombination mit Temperaturfits, welches keinen Gradienten der Fehlerfunktion, sondern lediglich Richtungsableitungen der physikalischen Eigenschaften betrachtet und auf einem Gauß-Newton-Verfahren basiert. Dadurch wird ein quadratisches Modell der Fehlerfunktion erstellt. Das Verfahren nach Stoll wurde jedoch in einer in dieser Dissertation entwickelten Variante betrachtet: Anstatt der globalen Minimierung des quadratischen Modells wurde die Methode mit dem Trust-Region-Ansatz kombiniert, das heißt, das Modell wurde lokal auf einem kleinen Vertrauensbereich minimiert, dessen Größe innerhalb des Algorithmus modifiziert wird. Da die Variante des Verfahrens nach Stoll ebenfalls bei der Gradientenberechnung auf bereits durchgeführte Simulationen zurückgreift, zeichnet es sich durch einen äußerst geringen Rechenaufwand aus. Das Trust-Region-Verfahren hingegen benötigt in jeder Iteration eine Hesse-Matrix, welche in der Nähe des Minimums allerdings nie effizient berechnet werden kann. Weiterhin ist die Variante des Verfahrens nach Stoll robuster. Falls für jede zu optimierende Zielgröße eine experimentell bestimmte temperatur- und druckabhängige Funktion bekannt ist, kann als letzter Schritt die Methode der reduzierten Einheiten verwendet werden.

Abstiegsverfahren haben den Nachteil, daß nach falsch berechneter Abstiegsrichtung auch mit einer geeigneten Schrittweitensteuerung nicht mehr die Möglichkeit besteht, einen kleineren Fehlerfunktionswert zu finden. Verfahren, die mit dem Trust-Region-Ansatz kombiniert werden, sind daher gerade in der Nähe des Minimums besser geeignet, da hierbei zunächst die Schrittweite festgesetzt und in Abhängigkeit davon eine Abstiegsrichtung bestimmt wird. Eine falsche Abstiegsrichtung entsteht vor allem durch die aufgrund des statistischen Rauschens hervorgerufene Berechnung einer falschen Gradientenrichtung. In derartigen Fällen sind ableitungsfreie Methoden gradientenbasierten Verfahren vorzuziehen. Aus diesem Grund wurde in dieser Dissertation das **D***ünn-***G***itter-basierte* **a***bleitungs***f***reie* **O***ptimierungsverfahren (DGAFO-Verfahren)*, eine Kombination von Interpolation auf Dünnen Gittern, Glättung und dem Trust-Region-Verfahren, entwickelt. Das übergeordnete Ziel bestand darin, gradientenbasierte Verfahren in den Punkten Effizienz, Robustheit und exakter Erhalt eines lokalen Minimums zu übertreffen. Tatsächlich konnten diese Eigenschaften zumindest bedingt realisiert werden: Das Verfahren kam bei geeigneter Wahl der Größe des initialen Vertrauensgebiets mit nur etwa halb so vielen Funktionsevaluationen aus wie das jeweils beste gradientenbasierte Verfahren. Dies kann dadurch erklärt werden, daß aufgrund der Verwendung eines Gitters bei einem Iterationsschritt nur ein oder mehrere Kraftfeldparameter verändert werden konnten. Somit konnte vermieden werden, daß es wie gradientenbasierte Verfahren zunächst in eine falsche Richtung lief. Auch gelangte es in einem sehr wichtigen Anwendungsfall näher an das Minimum heran als GROW. Die Variante des Verfahrens nach Stoll benötigte dafür jedoch deutlich weniger Simulationen. Bezüglich Robustheit konnte das DGAFO-Verfahren nicht so überzeugen wie die gradientenbasierten Verfahren, was auf die schwierig einzustellenden Größen der Vertrauensgebiete zurückzuführen war.
Da die Interpolation auf Dünnen Gittern gewisse Glattheitsanforderungen an die zu minimierende Funktion stellt, wurde das DGAFO-Verfahren mit geeigneten Glättungs- und Regularisierungsverfahren kombiniert, welche das statistische Rauschen herausfiltern sollten. Hierbei stellte sich eine Glättung basierend auf positiv definiten, insbesondere Gaußschen, Radialen Basisfunktionen und für höherdimensionale Probleme eine lineare Approximation der physikalischen Zielgrößen als am besten geeignet heraus. Als beste Regularisierungsverfahren stellten sich der MARS-Algorithmus, welcher auf stückweise linearen Basisfunktionen basiert, und die Ridge-Regression, welche verzerrte Regressionskoeffizienten liefert, heraus.

Praktische Aspekte Für eine detaillierte Verfahrensevaluation wurden zunächst Molekulare Simulationen durch sogenannte *Korrelationsfunktionen* ersetzt, welche eine analytische Abhängigkeit zwischen bestimmten Zielgrößen von bestimmten Kraftfeldparametern beschreiben. Um Simulationen zu imitieren, wurde auf die Zielgrößen statistisches Rauschen künstlich addiert. Da die Korrelationsfunktionen nur für Zweizentren-Lennard-Jones-Teilchen mit einem dipolaren oder quadrupolaren Moment gültig sind, waren die aus der Verfahrensevaluation selektierten Verfahren in einem nachfolgenden Schritt auf Molekulare Simulationen anzuwenden. MD-Simulationen von Flüssigkeiten wurden dabei mit der Software *Gromacs* und MC-Simulationen an der Phasenübergangskurve mit der Software *ms2* durchgeführt. Für erste Verfahrenstests für die gradientenbasierten Verfahren wurden dabei kleine Substanzen betrachtet, welche allerdings nicht auf einfache Art und Weise zu simulieren waren. Es handelte sich dabei um Benzol, Phosgen, Methanol und Kohlenstoffdisulfid. In der Regel änderten sich die Kraftfeldparameter (σ in nm und ε in kJ/mol) im Laufe der Optimierung in der zweiten oder dritten Nachkommastelle, in einigen wenigen Fällen nur in der vierten oder fünften, was trotzdem oftmals noch zu signifikanten Verkleinerungen der Fehlerfunktion führte. Einige der erhaltenen Kraftfelder wurden für andere physikalische Zielgrößen ausgewertet, was jedoch nur in manchen Fällen zu guten beziehungsweise zufriedenstellenden Ergebnissen führte. Es kann allerdings prinzipiell nicht davon ausgegangen werden, daß ein durch eine Optimierung erhaltenes Kraftfeld generisch anwendbar, das heißt transferierbar auf andere Observablen, ist. Ob ein Kraftfeld für weiterführende Simulationen verwendet werden kann, ist stets von einem Physiker oder Chemiker zu beurteilen. Allerdings konnte in dieser Dissertation die Aussage getroffen werden, daß gerade für eine Feinjustierung akkurate molekulare Modelle und möglichst viele verschiedenartige Zielgrößen zu verschiedenen Temperaturen in Betracht gezogen werden sollten. Außerdem müssen Art und Anzahl der zu optimierenden Kraftfeldparameter geeignet gewählt werden. Nur dann ist der Erhalt eines generischen Kraftfelds garantiert.

Eine Kombination von CMA-ES mit GROW wurde anhand von Phosgen evaluiert und eine Kombination von GROW mit der Variante des Verfahrens nach Stoll anhand von Dipropylenglykoldimethylether, einem wissenschaftlich und industriell interessanten Lösungsmittel. Ein Kraftfeld an der Phasenübergangskurve wurde für Ethylenoxid mit einer Kombination von DesParO und GROW erzielt. Auch anhand dieser Untersuchung konnte festgestellt werden, daß das molekulare Modell für eine Feinoptimierung äußerst wichtig ist. In diesem Fall war ein auf einem Dipol basierendes einem auf Partialladungen basierenden Modell vorzuziehen. Eine aktuell sowohl für die Wissenschaft als auch für die Industrie sehr interessante Applikation ist weiterhin die Substanzklasse der Ionischen Flüssigkeiten. Am Beispiel einer konkreten Ionischen Flüssigkeit, $[C_2MIM][NTf_2]$, wurde versucht, ein Kraftfeld zu generieren, welches Dichte und Transporteigenschaften reproduziert. Dies konnte aufgrund der hohen Schwierigkeit im Falle der Simulation Ionischer Flüssigkeiten allerdings nur bedingt erzielt werden. Diese Applikation zeigte jedoch deutlich, wie wichtig die Wahl der zu optimierenden Kraftfeldparameter ist.
Das DGAFO-Verfahren wurde sowohl anhand der Korrelationsfunktionen als auch anhand von Molekularen Simulationen für Benzol, Ethylenoxid und Dipropylenglycoldimethylether evaluiert. In den ersten beiden Fällen konnte die schnellere Konvergenz praktisch belegt, und im zweiten Fall konnte gezeigt werden, daß das Verfahren näher an das Minimum gelangen kann als GROW.

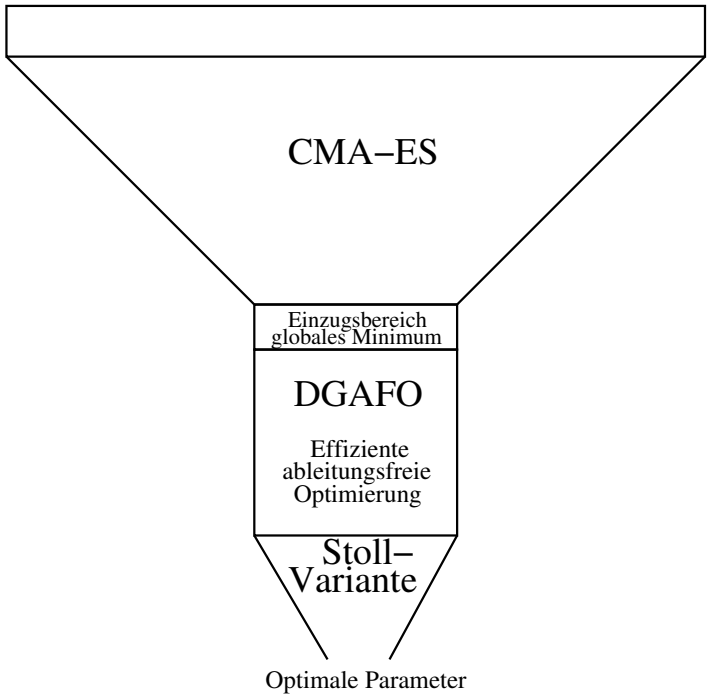


Abbildung 8.2: In dieser Arbeit resultierender Filterungsprozeß: Zu Beginn wird, falls erforderlich, der Parameterraum von CMA-ES effizient und robust abgetastet, um in den Einzugsbereich eines globalen Minimums zu gelangen. Von da an wird das DGAFO-Verfahren verwendet, da es mit deutlich weniger Funktionsauswertungen auskommt als GROW. In der Nähe des Minimums hat sich die Variante des Verfahrens nach Stoll als am besten geeignet herausgestellt. Zum Schluß erhält man die optimalen Kraftfeldparameter.

Fazit Aufgrund sowohl theoretischer Überlegungen als auch praktischer Erfahrungen entstand der in Abbildung 8.2 dargestellte in dieser Arbeit empfohlene Kombinationsworkflow. Zusammenfassend läßt sich sagen, daß mit dieser Dissertation ein entscheidender Beitrag für den Bereich Kraftfeldparametrisierung für Molekulare Simulationen geleistet werden konnte.

8.2 Ausblick

Die vorliegende Dissertation konnte auf dem Gebiet der Kraftfeldoptimierung Molekularer Simulationen deutliche Fortschritte erzielen, allerdings ist noch eine Vielzahl an weiterführenden Untersuchungen unerläßlich.

Verbesserung und Erweiterung der Verfahren Die hier getroffene Verfahrensauswahl ist noch lange nicht als final anzusehen. Es sind diesbezüglich allerdings einige Einschränkungen anzugeben: Aufgrund des statistischen Rauschens und des hohen Rechenaufwands Molekularer Simulationen empfiehlt es sich nicht, komplexere Verfahren als die hier verwendeten anzuwenden. Damit sind solche gemeint, welche eine höhere Konvergenzgeschwindigkeit aufweisen, jedoch auch mehr Funktionsauswertungen benötigen. Auch von Verfahren, die mit zusätzlichem Rechenaufwand geeignet parametrisiert werden müssen, ist abzusehen.

Bezüglich der in dieser Arbeit eingesetzten Verfahren sind die folgenden Verbesserungsmöglichkeiten denkbar: Zunächst kann die zu minimierende Fehlerfunktion selbst modifiziert werden. Es ist nicht unbedingt nowendig, hierzu die euklidische Norm zu verwenden. Es können auch zum Beispiel die 1-Norm, die Maximumsnorm oder andere Normen betrachtet werden. Dabei ist jedoch zu beachten, daß dadurch die Differenzierbarkeit nicht mehr gewährleistet ist, und es sind entsprechende ableitungsfreie Algorithmen einzusetzen wie beispielsweise das in dieser Dissertation entwickelte DGAFO-Verfahren oder stochastische globale Optimierungsverfahren. Wichtig ist, daß stets gewichtete Normen verwendet werden. Dabei ist selbstverständlich auch eine andere Wahl der Gewichtung denkbar. Bei Betrachtung der euklidischen Norm haben sich hier insbesondere gradientenbasierte Verfahren als sehr gut geeignet herausgestellt. Im Bereich der gradientenbasierten Optimierung gibt es eine Vielzahl an Algorithmen, zum Beispiel verbesserte Quasi-Newton-Verfahren. Die Voraussetzung der positiven Definitheit der Hesse-Matrix sollte dabei jedoch nicht notwendig sein. Es ist möglich, ein geeignetes $\lambda > 0$ zu finden, so daß die approximative Hesse-Matrix $H + \lambda I$ positiv definit ist. Dann ist es jedoch eher ratsam, den Levenberg-Marquardt-Algorithmus vor allem in der Nähe des Minimums einzusetzen, und zwar als Verallgemeinerung der Variante des Verfahrens nach Stoll. Die Fehlerfunktion wird dabei als vektorwertige Funktion $\hat{F} : \mathbb{R}^N \to \mathbb{R}^n$ formuliert und die Hesse-Matrix mithilfe der Jacobi-Matrix J von $\hat{F}$ angenähert. Anstatt $H + \lambda I$ wird dann $J^T J + \lambda I$ betrachtet. Der Dämpfungsparameter λ wird zu Beginn der Optimierung so eingestellt, daß der Algorithmus der Methode des steilsten Abstiegs gleichkommt. In der Nähe des Minimums sollen jedoch die Krümmungseigenschaften der Fehlerfunktion ausgenutzt werden. Nach Marquardt (1963) ist mit dieser Methode auch das Regenrinnenproblem besser handhabbar, die Problematik der akkuraten Gradientenberechnung in der Nähe des Minimums aufgrund des statistischen Rauschens bleibt jedoch weiterhin bestehen. Die Justierung des Verfahrensparameters λ erhöht zusätzlich den Rechenaufwand.
Auch die numerische Approximation der partiellen Ableitungen kann mit Diskretisierungen höherer Ordnung erfolgen, beispielsweise mithilfe zentraler Differenzen. Bei letzteren sind doppelt so viele Simulationen wie im Falle der hier ausschließlich verwendeten Vorwärtsdiskretisierung notwendig. Es kann jedoch bewiesen werden, daß die Produktionszeit bei jeder Simulation halbiert und dennoch die gewünschte Erhöhung der Genauigkeit erzielt werden kann.
Bezüglich der in dieser Arbeit entwickelten effizienten Gradientenberechnung ist die folgende Verbesserung denkbar: Es können anstatt der Gradienten der Fehlerfunktion die Gradienten der zu optimierenden Zielgrößen effizient berechnet werden. Dadurch konnte die Variante des Verfahrens nach Stoll insbesondere in der Nähe des Minimums eingesetzt werden. Es ist also möglich, daß eine effiziente Gradientenberechnung auch bei allen anderen Verfahren dort verwendbar ist.

Äußerst lohnender Forschungsbedarf besteht weiterhin im Bereich der globalen Optimierung: Die hier eingesetzten Verfahren CMA-ES und DesParO waren nicht dazu in der Lage, mit akzeptablem Rechenaufwand robust in den Einzugsbereich eines globalen Minimums zu gelangen, wo lokale Optimierer effizienter sind. Weiterhin konnte kein geeignetes Abbruchkriterium für die vorgeschaltete globale Optimierung gefunden werden.

Globale Optimierungsverfahren können grob in zwei Kategorien eingeteilt werden, und zwar in stochastische Verfahren und solche, die eine Metamodellierung der zu minimierenden Funktion durchführen. Neben klassischen Evolutionären Algorithmen (siehe zum Beispiel Schwefel (1995)) kann zum Beispiel die Methode der *Differential Evolution* mit CMA-ES verglichen werden. Dieses Verfahren findet bevorzugte Richtungen durch Differenzenbildung innerhalb einer Population und ist somit gerichteter als andere stochastische Optimierer. Bezüglich der Metamodellierung haben sich Interpolationen und Approximationen, welche auf Radialen Basisfunktion basieren, in vielen Fällen als günstig erwiesen. Hier kann eine Verbesserung in der Verknüpfung derartiger Modellierungsansätze mit effizienten stochastischen Suchverfahren bestehen. Bei den hier verwendeten Optimierungsprozessen mit DesParO wurde in dieser Arbeit eine sogenannte *Reine Zufallssuche* durchgeführt, das heißt, die auszuwertenden Datenpunkte wurden ausschließlich per Zufallsprinzip gewählt. Eine *Rekursive Zufallssuche* würde bereits eine wesentliche Verbesserung darstellen. Diese basiert auf Intervallschachtelungen. Die Intervalle innerhalb des Parameterraums werden je nach Güte des Modells verkleinert oder vergrößert, analog zu den Trust-Region-Verfahren. Dadurch werden auf effiziente Art und Weise lokale Verfeinerungen ermöglicht, und das Finden eines geeigneten Abbruchkriteriums kann dadurch erleichtert werden. Eine weitere Effizienzerhöhung kann prinzipiell durch eine sogenannte *Tabu-Suche* erreicht werden: Diese sieht vor, bereits vom Verfahren frequentierte Suchbereiche zu verwerfen, so daß diese nicht mehr in Betracht gezogen werden. Die klassische Tabu-Suche besteht darin, lokale Optimierungsmethoden bis zum Erhalt eines lokalen Minimums einzusetzen und dieses dann durch stochastische Operationen wieder zu verlassen. Dies ist jedoch aus den folgenden Gründen für die vorliegende Problemstellung nicht empfehlenswert: Zum einen ist die Verwendung lokaler Optimierer für eine globale Optimierung viel zu rechenaufwendig, da sämtliche lokale Minima bis zum Erhalt eines globalen Minimums gefunden werden müssen. Zum anderen ist aufgrund des statistischen Rauschens das Verlassen eines lokalen Minimums problematisch, da das Finden eines geeigneten Abbruchkriteriums hierzu keineswegs trivial ist. Allerdings kann die Idee der Tabu-Suche insofern miteinbezogen werden, als daß Fehlerfunktionswerte von Punkten, die in bereits besuchten Bereichen liegen, innerhalb der Metamodellierung mit einem Strafterm versehen werden. Umsetzungen dieser Ideen konnten lediglich aus Zeitgründen bisher nicht angegangen werden, sind jedoch für die nahe Zukunft geplant.

Das in dieser Dissertation entwickelte DGAFO-Verfahren konnte insbesondere in Bezug auf Effizienz überzeugen, allerdings ist die Einstellung der Größen der Vertrauensgebiete problematisch, in der Nähe des Minimums sogar kritisch. Diesem Problem könnte folgendermaßen entgegengewirkt werden: Zu Beginn der Optimierung könnte eine Metamodellierung der Fehlerfunktion auf einem Dünnen Gitter über dem Innern des gesamten zulässigen Gebiets durchgeführt werden. Dadurch kann eine Umgebung des Startpunkts detektiert werden, in der das Metamodell einen klaren abfallenden Trend aufweist, und die Größe des Vertrauensgebiets kann entsprechend gesetzt werden. Letztere sollte dann aufgrund des Modellierungsfehlers etwas kleiner als maximal möglich gewählt werden. Wie dies am effizientesten zu realisieren ist, bedarf allerdings noch eingehender Untersuchungen. In der Nähe des Minimums kann, wie mehrfach motiviert, die Existenz einer optimalen Größe des Vertrauensgebiets nicht bewiesen werden. Das statistische Rauschen ist die Hauptursache für dieses Problem, das durch die Regenrinnengestalt der Fehlerfunktion noch erschwert wird. Es ist daher naheliegend, in der Nähe des Minimums anstelle

des DGAFO-Verfahrens die ebenfalls in dieser Arbeit entwickelte Variante des Verfahrens nach Stoll oder den oben erwähnten Levenberg-Marquardt-Algorithmus einzusetzen.

Verbesserungen in Bezug auf praktische Anwendungen Wie bereits erwähnt, ist die Methodik bereits heute so weit ausgereift, daß sie routinemäßig für beliebige chemische Substanzen eingesetzt werden kann. Weitere Anwendungen der hier betrachteten Optimierungsverfahren auf Molekulare Simulationen werden die folgenden sein: Zunächst soll ein Kraftfeld für Azetonitril (CH_3CN), ein toxisches Lösungsmittel, gefunden werden, welches sowohl VLE-Daten als auch Transporteigenschaften optimal reproduziert. Dazu ist jedoch zunächst ein geeignetes molekulares Modell zu definieren, was bei dieser Substanz keineswegs trivial ist. Das Finden geeigneter Startparameter hat sich, bei Verwendung eines einfachen molekularen Modells, bereits als äußerst schwierig erwiesen. Hierzu sind geeignete globale Voroptimierer einzusetzen, allerdings effizientere als die in dieser Arbeit betrachteten. Dann werden das DGAFO-Verfahren und, falls notwendig, die Variante des Verfahrens nach Stoll verwendet werden.
Weiterhin sollen in naher Zukunft verbesserte Kraftfelder für Ionische Flüssigkeiten gefunden werden, sowohl für die in dieser Arbeit betrachtete als auch für weitere. Auch hierbei werden die entsprechenden Kraftfeldparameter an verschiedenartige Zielgrößen angepaßt werden. Insbesondere soll auch die Viskosität in Betracht gezogen werden, eine äußerst verrauschte Größe, für deren Berechnung lange Simulationszeiten erforderlich sind.
Zum Erhalt optimaler Kraftfelder werden die in dieser Dissertation empfohlenen Optimierungsverfahren mit dem am Fraunhofer-Institut SCAI entwickelten Softwarepaket $WOLF_2PACK$ zu kombinieren sein. Dabei werden durch empirische Optimierungsprozesse erhaltene intermolekulare Kraftfeldparameter zur Nachjustierung von intramolekularen Kraftfeldparametern verwendet. Letztere können dann wieder für eine empirische Nachjustierung von intermolekularen Parametern eingesetzt werden. Dadurch kann der Problematik entgegengewirkt werden, daß intra- und intermolekulare Parameter nicht als vollständig unabhängig voneinander betrachtet werden können. Diese Problematik zeigt sich insbesondere bei 1,4- und 1,5-Lennard-Jones-Wechselwirkungen. Ob der avisierte Kreislauf tatsächlich zur Konvergenz führt, wird eingehend zu untersuchen sein. Die resultierenden Kraftfelder sind dazu in Bezug auf ihre Transferierbarkeit hin zu evaluieren. Diese bezieht sich nicht nur auf andere Observablen, sondern auch auf andere chemische Substanzen. Ein übergeordnetes Ziel besteht darin, für jeden Atomtyp Kraftfeldparameter zu detektieren, die für alle chemischen Substanzen verwendbar sind.

In mittelfristiger Zukunft sollte es möglich sein, eine Vielzahl von Kraftfeldern, welche auf verschiedenartige chemische und physikalische Problemstellungen anwendbar sind, auf äußerst effiziente Art zu entwickeln. Diese sollen dann bei Molekularen Simulationen zum Einsatz kommen, welche teure und aufwendige Experimente im Labor tatsächlich ersetzen können, was im Idealfall zur computergestützten Entwicklung neuer Materialen und zur Ermöglichung, *in silico* Einblick in eine Vielzahl an mikroskopischen Prozessen zu erhalten, führen kann. Bis dahin ist es allerdings noch ein weiter Weg. Molekulare Simulationen werden erst nach der Realisierung deutlich geringerer Rechenzeiten, welche auch mit der stetig ansteigenden Kapazität moderner Rechenressourcen ermöglicht werden wird, regelmäßig zum Einsatz kommen und Experimente ersetzen können.

A Ensembles

Dieser Anhang behandelt die Realisierung der im Rahmen dieser Arbeit eingesetzten Ensembles. Die vier wichtigen Ensembles aus der Klassischen Mechanik wurden bereits in Definition 2.1.1 eingeführt.
Oftmals zur Vorbereitung einer Prääquilibrierung und teilweise zur Berechnung von Transporteigenschaften werden NVT-Simulationen verwendet. Das Volumen konstant zu halten, geschieht über räumliche periodische Randbedingungen: Verläßt ein Teilchen die Simulationsbox, so wird ein anderes auf der gegenüberliegenden Seite einfach wieder hineingesetzt. Schwieriger ist es, die Temperatur konstant zu halten. Wie dies bei MD-Simulationen zu realisieren ist, wird in Anhang A.1 dargestellt.
Die in dieser Arbeit am häufigsten durchgeführten Simulationen sind NPT-Simulationen, da meistens die Dichte als zu optimierende Eigenschaft verwendet wird. Auch bei Simulationen an der Phasenübergangskurve sind NPT-Simulationen vorzuschalten, um ein chemisches Potential zu berechnen, welches in einer anschließenden μVT-Simulation konstant gehalten wird. Anhang A.2 befaßt sich mit isobarisch-isothermen Ensembles bei MD-Simulationen, und Anhang A.3 behandelt die Realisierung von NPT-Ensembles bei MC-Simulationen. Großkanonische Ensembles werden in Anhang A.4 beschrieben, und in Anhang A.5 wird die technische Umsetzung von Simulationen an der Phasenübergangskurve dargestellt, welche aus der Kombination einer NPT-Simulation und der Flüssigkeit und einer anschließenden Pseudo-μVT-Simulation des Gases besteht.

In Kapitel 5 ist stets angegeben, welches Ensemble eingesetzt und ob dieses Ensembles mit MD oder MC simuliert wurde.

A.1 Molekulardynamik bei konstanter Temperatur

In diesem Abschnitt geht es um kanonische Ensembles, das heißt um MD-Simulationen bei konstanter Temperatur. Das Ziel dabei ist es, die Systemtemperatur zu regulieren, also ein Thermostat zu simulieren, welches das System auf eine konstante Temperatur T_0 bringt. Die allgemeine Vorgehensweise dafür besteht aus den folgenden drei Schritten:

1. Finden einer Beziehung zwischen Temperatur und Geschwindigkeiten, entweder direkt oder durch neue Bewegungsgleichungen,
2. Reskalierung der Geschwindigkeiten, so daß $T \to T_0$,
3. Berechnung der neuen Temperatur.

Die wichtigsten vier Methoden, MD-Simulationen bei konstanter Temperatur durchzuführen, werden im folgenden vorgestellt:

- Bei der *Methode der Zwangsbedingungen* wird die Zwangsbedingung

$$T \equiv const. \Leftrightarrow \dot{T} \equiv 0$$

als Nebenbedingung zur zugrundeliegenden Bewegungsgleichung eingeführt. Eine direkte Beziehung zwischen Temperatur und Geschwindigkeiten ist dabei über die kinetische Energie gegeben. Es gilt:

$$\frac{3N}{2} k_B T = \frac{1}{2} \sum_{i=1}^{N} m_i v_i^2 \overset{p_i = m_i v_i}{=} \frac{1}{2} \sum_{i=1}^{N} \frac{p_i^2}{m_i}. \tag{A.1}$$

Dabei ist p_i der Impuls des Teilchens i und $k_B = 1.38 \cdot 10^{-23}$ J/K die Boltzmann-Konstante. Es gilt also:

$$T \propto \sum_{i=1}^{N} m_i v_i^2 = \sum_{i=1}^{N} \frac{p_i^2}{m_i}. \tag{A.2}$$

Die Reskalierung der Geschwindigkeiten erfolgt gemäß

$$v_i' = \left(\sqrt{\frac{T}{T_0}} \right)^{-1} v_i, \;\; i = 1, ..., N,$$

das heißt, es werden neue Geschwindigkeiten $v_i', \;\; i = 1, ..., N$, berechnet, die wiederum zu einer neuen Temperatur $\tilde{T}$ führen. Dies wird solange wiederholt, bis T_0 erreicht ist. Diese sehr einfache Methode hat jedoch den Nachteil, daß sie zu einer falschen Geschwindigkeitsverteilung führt, so daß man gar nicht zu einem physikalisch korrekten kanonischen Ensemble gelangt. Hoover u. a. (1982) und Evans (1983) haben daher ein generelles Verfahren eingesetzt, welches einen Reibungsterm $\xi(r, p)$ in die Bewegungsgleichung einführt und diese dann unter der Nebenbedingung $\dot{T} = 0$ löst. Neben der allgemeinen Gleichung aus der Newtonschen Mechanik

$$\dot{r}_i = \frac{p_i}{m_i}, \;\; i = 1, ..., N, \tag{A.3}$$

wird die zusätzliche Bewegungsgleichung mit Reibung als Lagrange-Mulitplikator

$$\dot{p}_i = f_i - \xi(r, p) p_i, \;\; i = 1, ..., N, \tag{A.4}$$

betrachtet. Aufgrund von Gleichung (A.2) ergibt sich:

$$\dot{T} \propto \frac{d}{dt} \left(\sum_{i=1}^{N} p_i^2 \right) = \sum_{i=1}^{N} \dot{p}_i p_i \overset{!}{=} 0.$$

Aus dieser Nebenbedingung ergibt sich aus Gleichung (A.4) für den Reibungsterm:

$$\xi(r, p) = \frac{\sum_{i=1}^{N} p_i f_i}{\sum_{i=1}^{N} p_i^2}.$$

Als Variante des Bocksprung-Verlet wird die folgende Iterationsvorschrift zur Bestimmung der Geschwindigkeiten verwendet:

$$\dot{r}(t+\frac{\Delta t}{2}) = \dot{r}(t-\frac{\Delta t}{2}) + \left(\underbrace{\frac{f(t)}{m} - \xi \dot{r}(t)}_{=\dot{p}(t)} \right) \Delta t.$$

Ein Nachteil dieses Verfahrens ist, daß bei kleinen Systemen ($N \lesssim 1$ mol) der Wärmeaustausch mit der Umgebung verloren geht, was physikalisch unsinnig ist.

- Bei *stochastischen* Verfahren, welche auf Andersen (1980) zurückgehen, besteht das Thermostat aus stochastischen Kollisionen zwischen dem System und einem außen angelegten Bad mit konstanter Temperatur $T = T_0$. Dabei wird zwischen den folgenden zwei Arten von Kollisionen unterschieden:

  1. *Zufallskollisionen*: Nach einer bestimmten Anzahl von Zeitschritten werden Zufallsgeschwindigkeiten aus einer Maxwell-Boltzmann-Verteilung ermittelt. Dies darf nicht zu oft geschehen, da ansonsten Unstetigkeiten in der Trajektorie entstehen.
  2. *Massive Kollisionen*: Alle 1000 Zeitschritte wird eine drastische Änderung der Geschwindigkeiten gemäß einer Maxwell-Boltzmann-Verteilung vorgenommen.

  Falls derartige stochastische Kollisionen zu selten geschehen, verlangsamt dies die Äquilibrierung zu T_0, geschehen sie jedoch zu oft, wird die Trajektorie unstetig, was dazu führt, daß keine Berechnung dynamischer Eigenschaften mehr möglich ist. Nach Andersen (1980) sollte für die Kollisionsrate μ_{coll} gelten:

$$\mu_{\text{coll}} \propto \frac{\lambda_T}{\sqrt[3]{\rho}\sqrt[3]{N^2}},$$

  wobei ρ die Dichte und λ_T die Wärmeleitfähigkeit des Systems ist.
  Eine Reskalierung der Geschwindigkeiten für stochastische Verfahren wurde von Heyes (1983) eingeführt:

$$v^n = (1+\zeta)v^a.$$

  Dabei bezieht sich der Index n auf den neuen und a auf den alten Zustand, und $\zeta \in [-0.05, 0.05]$ ist eine gleichverteilte Zufallszahl. Für die Wahrscheinlichkeit des neuen Zustands im Verhältnis zum alten gilt dann:

$$\frac{P_n}{P_m} = \exp(-A)$$

  mit

$$A := \frac{1}{2} m \beta_T \sum_{i=1}^{N} (v^m)^2 \left((1+\zeta)^2 - 1 \right) - 3N \log(1+\zeta),$$

  wobei $\beta_T := k_B T$. Eine Versuchsbewegung wird mit der Wahrscheinlichkeit $\min(1, \exp(-A))$ akzeptiert.

- Bei der *Methode des erweiterten Systems* wird eine Zusatzkoordinate s eingeführt, und es werden dadurch neue, fiktive Kräfte, Potentiale und Bewegungsgleichungen eingeführt. Eine Reskalierung der Geschwindigkeiten erfolgt durch

$$v_i = s\dot{r}_i, \;\; i = 1, ..., N,$$

und die Masse der Zusatzkoordinate m_s bestimmt die Äquilibrierung des Systems zur gewünschten Temperatur.
Nosé (1984) hat die folgenden fiktiven Größen eingeführt:

  - Fiktives Potential: $U_s := (1 + df)k_BT \log s$
  - Fiktive Kraft: $f_s := -\frac{dU_s}{ds} = -(1 + df)\frac{k_BT}{s}$
  - Fiktive kinetische Energie: $K_s := \frac{1}{2}m_s\dot{s}^2$
  - Totaler Hamilton-Operator: $\hat{H}_s := \hat{K} + \hat{U} + \hat{K}_s + \hat{U}_s = \hat{H} + \hat{K}_s + \hat{U}_s \equiv const.$

Dabei sind df die Anzahl der Freiheitsgrade und $\hat{H}$ der Hamilton-Operator des ursprünglichen kanonischen Ensembles. Das neue, fiktive Ensemble ist aufgrund der Bedingung $\hat{H}_s \equiv const.$ mikrokanonisch. Die neuen Bewegungsgleichungen lauten

$$\begin{aligned} \ddot{r}_i &= \frac{f_i}{m_s} - 2\dot{s}\frac{r_i}{s} \\ m_s\ddot{s} &= \sum_{i=1}^{N} m_i\dot{r}_i^2 - (1 + df)\frac{k_BT}{s}. \end{aligned}$$

Bei numerischer Lösung dieser Bewegungsgleichung erhält man die Koordinaten r eines kanonischen Ensembles. Falls m_s zu groß ist, ist die Äquilibrierung langsam, und falls m_s zu klein gewählt wird, entstehen unphysikalische Oszillationen in der Trajektorie des mikrokanonischen Ensembles.
Hoover (1985) kombinierte die Methode von Nosé mit der Methode der Zwangsbedingungen. Zusammen mit den Bewegungsgleichungen (A.4) ergab sich dabei für den Reibungsterm eine Differentialgleichung erster Ordnung:

$$\dot{\xi} = \frac{df}{m_s}(k_BT_0 - k_BT).$$

Diese Kombinationsmethode zur Durchführung von MD-Simulationen bei konstanter Temperatur wurde bekannt unter dem Namen *Nosé-Hoover-Thermostat*. Anstatt m_s kann vom Benutzer auch die Periode τ_T der Oszillationen der kinetischen Energie zwischen System und Reservoir angegeben werden. Diese steht direkt zu m_s in Beziehung. Es gilt:

$$m_s = \frac{\tau_T^2 T_0}{4\pi^2}.$$

- Die *Methode der schwachen Kopplung* geht auf Berendsen u. a. (1984) zurück und ist auch unter dem Namen *Berendsen-Thermostat* bekannt. Auch hierbei handelt es sich um die Temperaturkopplung mit der eines außen angelegten Bades. Das Berendsen-Thermostat hat die Form

$$\frac{dT}{dt} = \frac{1}{\tau}(T_0 - T), \tag{A.5}$$

wobei τ eine vorgegebene Zeitkonstante ist. In der Literatur wird τ =0.4 ps empfohlen. Die Reskalierung der Geschwindigkeiten geschieht durch

$$v_i' = \lambda v_i, \;\; i = 1, ..., N,$$

wobei λ folgendermaßen hergeleitet wird: Für den Unterschied an kinetischer Energie gilt zunächst

$$\begin{aligned} \Delta K &= \frac{1}{2}\sum_{i=1}^{N} m_i v_i'^2 - \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 = (\lambda^2 - 1)\frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 \\ &= (\lambda^2 - 1)K = (\lambda^2 - 1)\frac{3Nk_BT}{2}. \end{aligned} \tag{A.6}$$

Die Wärmekapazität bei konstantem Volumen V ist definiert durch

$$C_V = \left(\frac{\partial K}{\partial T}\right)_V \approx \frac{\Delta K}{\Delta T} \tag{A.7}$$

Wegen Gleichung (A.6) gilt somit

$$\begin{aligned} (\lambda^2 - 1)\frac{3Nk_BT}{2C_V} &= \Delta T \\ \Rightarrow (\lambda^2 - 1)\frac{3Nk_BT}{2C_V\Delta t} &= \frac{\Delta T}{\Delta t} \approx \frac{dT}{dt} \overset{(A.5)}{=} \frac{1}{\tau}(T_0 - T) \\ \Rightarrow \lambda^2 - 1 &= \frac{\Delta t}{\Delta T}\frac{2C_V}{3Nk_B}\frac{T_0 - T}{T}. \end{aligned} \tag{A.8}$$

Das Berendsen-Thermostat kann auch als Nosé-Hoover-Thermostat mit

$$\xi = \frac{1}{2\tau k_B T}(k_B T_0 - k_B T)$$

aufgefaßt werden, allerdings wird hierbei die kinetische Energie nicht konstant gesetzt.

A.2 Molekulardynamik bei konstantem Druck

Analog zu Anhang A.1 werden nun MD-Simulationen bei konstantem Druck betrachtet. Dabei wird ein Barostat simuliert, so daß der Druck auf einen konstanten Wert P_0 gebracht wird. Stochastische Verfahren werden bei isotherm-isobarischen Ensembles nur selten angewandt. Ansonsten sind die hier vorgestellten Methoden ähnlich derer, die bei kanonischen Ensembles Verwendung finden:

- Die *Methode der Zwangsbedingungen* führt die Nebenbedingung $\dot{P} = 0$ in die Lösung der Bewegungsgleichung ein und wurde von Evans u. Morriss (1983) und Evans u. Morriss (1984) entwickelt. Anstelle des Reibungsterms ξ wird hier ein anderer Lagrange-Multiplikator verwendet. Analog zu (A.4) ergibt sich die folgende Bewegungsgleichung:

$$\begin{aligned} \dot{r} &= \frac{p}{m} + \chi(r,p)r \\ \dot{p} &= f - \chi(r,p)p \\ \dot{V} &= 3V\chi(r,p). \end{aligned} \tag{A.9}$$

Dabei ist V das Volumen der Simulationsbox, welches verändert wird, um einen bestimmten Druck P_0 zu simulieren. Der Lagrange-Multiplikator χ wird auch als *Dilationsrate* des Systems bezeichnet. Nach der Cheung-Formel für den Druck aus der Klassischen Mechanik gilt

$$P = \rho k_B T + \frac{W}{V}, \tag{A.10}$$

wobei $W = \frac{1}{3}\sum_{i=1}^{N} r_i f_i$ das *Virial* ist. Mit der Produktregel folgt sofort

$$3\dot{P}V + 3P\dot{V} = \sum_{i=1}^{N} \frac{2}{m} p_i \dot{p}_i + \dot{r}_i f_i + \dot{f}_i r_i. \tag{A.11}$$

Setzt man $P = P_0$ beziehungsweise $\dot{P} = 0$, so erhält man aus den Gleichungen (A.10) und (A.11)

$$\chi(r,p) = \frac{\frac{2}{m}\sum_{i=1}^{N} p_i f_i - \frac{1}{m}\sum_{i=1}^{N}\sum_{j>i}^{N}(r_{ij} p_{ij} \chi(r_{ij})/r_{ij}^2)}{\frac{2}{m}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N}\sum_{j>i}^{N}\chi(r_{ij}) + 9PV},$$

wobei $\chi(r_{ij}) := r_{ij}\frac{\partial W(r_{ij})}{\partial r_{ij}}$ und $p_{ij} := p_i p_j$. Für die Berechnung des Nenners sind die folgenden langreichweitigen Korrekturterme (WRK) miteinzubeziehen:

$$\begin{aligned}\left(\sum_{i=1}^{N}\sum_{j>i}\chi(r_{ij})\right)_{\text{WRK}} &:= 2\pi\rho N \int_{r_C}^{\infty} \chi(r_{ij}) r_{ij}^2 d_{r_{ij}} \\ (PV)_{\text{WRK}} &:= -\frac{2}{3}\pi N \rho \int_{r_C}^{\infty} r^2 W(r) dr.\end{aligned}$$

Die Lösung der Bewegungsgleichungen (A.9) erhält man mit einem Standard-Prädiktor-Korrektor-Verfahren aus Abschnitt 2.3.1. Dazu benötigt man die Ableitung des Lagrange-Multiplikators χ nach der Zeit. Aufgrund der dritten Bewegungsgleichung erhält man mit der Produktregel

$$\ddot{V} = 3V\dot{\chi} + 3\dot{V}\chi$$

und somit

$$\dot{\chi} = \frac{\ddot{V} - 3\dot{V}\chi}{3V} = \frac{\ddot{V}V - 3\dot{V}V\chi}{3V^2} \overset{\chi=\frac{\dot{V}}{3V}}{=} \frac{\ddot{V}V - \dot{V}^2}{3V^2}.$$

Bei isotherm-isobarischen Ensembles ist es wie bei kanonischen möglich, einen Reibungsterm ξ in die Bewegungsgleichung einzuführen. Hierbei erhält man

$$\dot{p} = f - (\chi + \xi)(r,p)p.$$

Dann gilt

$$\chi = -\frac{\frac{1}{m}\sum_{i=1}^{N}\sum_{j>i}^{N}(r_{ij} p_{ij} \chi(r_{ij})/r_{ij}^2)}{\sum_{i=1}^{N}\sum_{j>i}^{N}\chi(r_{ij}) + 9PV}$$

und

$$\chi + \xi = \frac{\sum_{i=1}^{N} p_i f_i}{\sum_{i=1}^{N} p_i^2}.$$

- Bei der *Methode des erweiterten Systems*, die auf Andersen (1980) zurückgeht und daher unter dem Namen *Andersen-Methode* bekannt ist, wird die Aktion eines Kolbens auf das System simuliert. Dabei wird wieder eine Zusatzvariable s eingeführt, um das System an eine externe Variable V zu koppeln, die für das Volumen der Simulationsbox steht. Es seien m_V die Masse des Kolbens und P_0 der gewünschte Druck. Dann werden die folgenden fiktiven Größen eingeführt:
  - Fiktives Potential: $U_V := P_0 V$
  - Fiktive kinetische Energie: $K_V := \frac{1}{2} m_V \dot{V}^2$.

  Eine Reskalierung der Koordinaten und Geschwindigkeiten erfolgt gemäß

$$\begin{aligned} r &= \sqrt[3]{V} s \\ v &= \sqrt[3]{V} \dot{s}. \end{aligned}$$

  Es gilt dann für die ursprüngliche potentielle und kinetische Energie:
  - Potentielle Energie: $U = U(\sqrt[3]{V} s)$
  - Kinetische Energie: $K = \frac{1}{2} m \sum_{i=1}^{N} v_i^2 = \frac{1}{2} \sqrt[3]{V^2} \sum_{i=1}^{N} \dot{s}_i^2$.

  Das erweiterte System obliegt den folgenden neuen Bewegungsgleichungen:

$$\begin{aligned} \ddot{s} &= \frac{f}{m \sqrt[3]{V}} - \frac{2}{3} \dot{s} \frac{\dot{V}}{V} \\ \ddot{V} &= \frac{P_0 - P}{m_V}. \end{aligned}$$

  Die Kraft f und der aktuelle Druck P werden dabei mit unskalierten Koordinaten berechnet. Der Druck kann dabei zum Beispiel durch die Cheung-Formel (A.10) berechnet werden. Unskalierte Geschwindigkeiten erhält man durch

$$\dot{r} = \sqrt[3]{V} \dot{s} + \frac{1}{3} \left(\sqrt[3]{V^2} \right)^{-1} \dot{V} s.$$

  Die Lösung der neuen Bewegungsgleichung ergibt ein isotherm-isobarisches Ensemble. Das sogenannte *Parrinello-Rahman-Barostat* (Parrinello u. Rahman, 1981) kombiniert das Andersen-Barostat mit der Ermöglichung von Volumenbewegungen.
  Hoover (1985) kombinierte die Andersen-Methode wieder mit der Methode der Zwangsbedingungen. Dabei entstanden die folgenden Bewegungsgleichungen:

$$\begin{aligned} \dot{s} &= \frac{p}{m} \sqrt[3]{V} \\ \dot{p} &= f - (\chi + \xi)(r, p) p \\ \chi &= \frac{\dot{V}}{3V}, \end{aligned}$$

  und für die Lagrange-Multiplikatoren ξ und χ gelten die folgenden Differentialgleichungen erster Ordnung:

$$\begin{aligned} \dot{\xi} &= \left(\sum_{i=1}^{N} \frac{p_i^2}{m} - f k_B T \right) / m_V \\ \dot{\chi} &= \left((P_0 - P) V \right) / t_P^2 k_B T. \end{aligned}$$

  Dabei ist t_P die Kopplungszeit für das Barostat.

- Analog zur *Methode der schwachen Kopplung* für kanonische Systeme wurde von Berendsen u. a. (1984) die Kopplung des Systems mit einem *Druckbad* simuliert, was durch folgende Differentialgleichung erster Ordnung beschrieben wird:

$$\frac{dP}{dt} = \frac{1}{t_P}(P_0 - P).$$

Eine Reskalierung der Koordinaten erfolgt gemäß

$$r' = \sqrt[3]{\chi} r,$$

wobei

$$\chi = 1 - \kappa_T \frac{\Delta t}{t_P}(P_0 - P).$$

Dabei ist κ_T die isotherme Kompressibilität. In der Literatur wird $t_P = 0.01$ ps empfohlen.

A.3 Monte-Carlo bei konstanter Temperatur und konstantem Druck

Es wird nun dargestellt, wie ein NPT-Ensemble bei MC-Simulationen realisiert wird. Die theoretischen Grundlagen hierzu, welche sich nur auf harte Kugeln beziehen, stammen von Wood (1968). Die Methode wurde später von McDonald (1969) weiterentwickelt und ist dadurch auch auf Lennard-Jones-Teilchen anwendbar. Sie stellt eine Erweiterung des in Abschnitt 2.4.2 eingeführten Metropolis-Schemas dar. Dabei wird eine MC-Bewegung mit einer bestimmten Wahrscheinlichkeit, welche vom Potential und der Temperatur abhängig ist, akzeptiert oder verworfen. Hält man diese Temperatur und das Boxvolumen konstant, so erhält man ein kanonisches Ensemble. Für ein isotherm-isobarisches Ensemble wird eine temperatur- und druckabhängige Wahrscheinlichkeit benötigt. Entscheidend bei dieser Methode ist die Realisierung von zufälligen Volumenbewegungen, die genau wie die Teilchenbewegungen mit einer gewissen Wahrscheinlichkeit akzeptiert oder verworfen werden. Der Ensemble-Durchschnitt in einem NPT-Ensemble ist gegeben durch ein Integral über dV (abhängig von P) und dq^N:

Definition A.3.1 (Ensemble-Durchschnitt (isotherm-isobarisch))**.** *Der* Ensemble-Durchschnitt einer Eigenschaft A in einem isotherm-isobarischen Ensemble *ist gegeben durch*

$$\langle A \rangle_{NPT} := \frac{1}{Z_{NPT}} \int_0^\infty \exp\left(-\beta_T P V\right) V^N dV \int A(s^N) \exp\left(-\beta_T U(s^N)\right) ds^N.$$

Dabei ist Z_{NPT} wieder ein Normalisierungsterm. Die skalierten Koordinaten s^N ergeben sich aus $s^N = L^{-1} q^N$ mit $L := \sqrt[3]{V}$.

Durch die Verwendung skalierter Koordinaten wird das obige Integral über den Einheitswürfel gebildet.
Die entsprechende Akzeptanzwahrscheinlichkeit im Metropolis-Schema ist nicht $\exp\left(-\beta_T U(s^N)\right)$, sondern

$$\exp\left(-\beta_T (PV + U(s^N)) + N \ln V\right).$$

Die Entscheidung, ob eine Zufallsbewegung $q_i \rightarrow q_j$ akzeptiert wird, geschieht mittels der Enthalpieänderung

$$\Delta H_{ij} = U_j - U_i + P(V_j - V_i) - N\beta_T^{-1} \ln \frac{V_j}{V_i}.$$

Eine MC-Bewegung wird dann mit Wahrscheinlichkeit $\min\left(1, \exp\left(-\beta_T \Delta H_{ij}\right)\right)$ akzeptiert. Bei dieser kann es sich entweder um eine Teilchenbewegung $s_i \rightarrow s_j$, eine Volumenbewegung $V_i \rightarrow V_j$ oder um eine Kombination aus Teilchen- und Volumenbewegung handeln. Es ist dabei zu beachten, daß die Akzeptanz einer Volumenbewegung weitaus rechenaufwendiger als die einer Teilchenbewegung ist, da für erstere sämtliche $\frac{N(N-1)}{2}$ Interaktionen zu berechnen sind. Für das Lennard-Jones-Potential läßt sich durch einen geschickten Separationsansatz der Rechenaufwand erheblich reduzieren, bei komplexeren Potentialen ist dies jedoch oftmals nicht möglich. Details hierzu sind in Allen u. Tildesley (1987) zu finden.

A.4 Monte-Carlo bei konstantem chemischem Potential

Dieser Abschnitt befaßt sich mit der Frage, wie ein großkanonisches Ensemble zu simulieren ist. Bei diesem Ensemble werden das chemische Potential μ (siehe Abschnitt 2.5.2), das Volumen V und die Temperatur T konstant gehalten. Die Grundlagen zu der hier vorgestellten Methodik gehen auf Norman u. Filinov (1969) zurück. Auch bei diesem Verfahren handelt es sich um eine Erweiterung des Metropolis-Schemas aus Abschnitt 2.4.2: Teilchenbewegungen werden mit einer bestimmten temperaturabhängigen Wahrscheinlichkeit akzeptiert oder verworfen, allerdings spielen bei einem μVTEnsemble zusätzlich Konstruktion und Destruktion eines Teilchens eine große Rolle, da $\mu \equiv const.$ bedeutet, daß die Anzahl an Systemteilchen N variiert. Der Ensemble-Durchschnitt in einem μVT-Ensemble ist gegeben durch eine von μ abhängige Summe über N und ein Integral über dq^N:

Definition A.4.1 (Ensemble-Durchschnitt (großkanonisch))**.** *Der* Ensemble-Durchschnitt einer Eigenschaft A in einem großkanonischen Ensemble *ist gegeben durch*

$$\langle A \rangle_{\mu VT} := \frac{1}{Z_{\mu VT}} \sum_{N=0}^{\infty} \frac{1}{N!} V^N E_A^N \int A(s^N) \exp\left(-\beta_T U(s^N)\right) ds^N.$$

Dabei ist $Z_{\mu VT}$ wieder ein Normalisierungsterm. Die skalierten Koordinaten s^N ergeben sich aus $s^N = L^{-1} q^N$ mit $L := \sqrt[3]{V}$. Die Aktivierungsenergie E_A ist abhängig vom chemischen Potential μ und gegeben durch $E_A = \exp\left(-\beta_T \mu\right) / \Lambda^3$, wobei $\Lambda := \sqrt{h^2/(2\pi m k_B T)}$ die de-Broglie-Wellenlänge ist. Dabei bezeichnet m die Masse des Systems und $h \approx 6.63 \times 10^{-34}$ Js das Plancksche Wirkungsquantum.

Das Einfügen oder Entfernen eines Teilchens kann nur durch Zufallsentscheidungen erfolgen und nicht durch das Lösen einer Bewegungsgleichung. Daher kann ein μVT-Ensemble nur mit MC simuliert werden. Das Verhältnis der Wahrscheinlichkeiten des neuen und alten Zustandes P_n und P_m ist im Falle einer Destruktion gegeben durch

$$\frac{P_n}{P_m} = \exp\left(-\beta_T (U_n - U_m) + \ln \frac{N}{E_A V}\right) \tag{A.12}$$

und im Falle einer Konstruktion durch

$$\frac{P_n}{P_m} = \exp\left(-\beta_T(U_n - U_m) + \ln\frac{E_A V}{N+1}\right). \tag{A.13}$$

Das Problem bei dieser Methode ist, daß die zugrundeliegende stochastische Matrix bezüglich Konstruktion und Destruktion nicht symmetrisch ist. Es sei α^K die Versuchswahrscheinlichkeit einer Konstruktion und α^D die einer Destruktion. Um die mikroskopische Umkehrbarkeit aus Lemma 2.4.1 zu gewährleisten, muß $\alpha^K = \alpha^D$ gelten. Nach Norman u. Filinov (1969) gilt weiterhin, daß die Wahl $\alpha^K = \alpha^D = \alpha^B = \frac{1}{3}$ die schnellste Konvergenz liefert, wobei α^B die Wahrscheinlichkeit einer einfachen Teilchenbewegung ist.
Ein Vorteil dieser Methode besteht darin, daß die freie Energie sofort gemäß

$$\frac{A}{N} = \mu - \frac{\langle P\rangle_{\mu VT} V}{\langle N\rangle_{\mu VT}} \tag{A.14}$$

berechnet werden kann. Ein Nachteil ist jedoch, daß Korrekturterme für den Druck nicht erst nach der Simulation verwendet werden können, was den Rechenaufwand während der Simulation erhöht. Dasselbe gilt für die Dichte: Da N variiert, ist zwar das Volumen, aber nicht die Dichte konstant. Auch das führt dazu, daß Korrekturterme für die potentielle Energie für langreichweitige Interaktionen nicht *a posteriori* verwendet werden dürfen.

Großkanonische Ensembles werden insbesondere für Adsorptionssimulationen an einer Oberfläche oder für Simulationen an der Phasenübergangskurve verwendet. Bei ersteren wird durch Teilchendestruktion eine metastabile dichte Phase in der Nähe der Oberfläche vermieden. Simulationen an der Phasenübergangskurve sind in Anhang A.5 beschrieben.

A.5 Simulationen an der Phasenübergangskurve

Simulationen an der Phasenübergangskurve sind im Rahmen dieser Arbeit von großem Interesse, da zum einen experimentelle Daten für Dichte und Verdampfungsenthalpie oftmals nur auf der Phasenübergangskurve erhältich sind und es zum anderen von großem Interesse ist, ein Kraftfeld zu finden, welches gleichzeitig Eigenschaften einer Substanz im flüssigen Zustand und im Zweiphasengleichgewicht wiedergibt. Phasengleichgewichte sind gerade im Bereich der Thermodynamik von sehr großer Bedeutung, da viele VLE-Eigenschaften miteinander in Beziehung stehen und auch experimentell bereits weitgehend erfaßt werden konnten (van Ness, 1995).
Es gibt sowohl MD- als auch MC-Ansätze zur simulativen Bestimmung von Phasengleichgewichten. Die hier angewandten Verfahren, welche bei den verwendeten Simulationsprogrammen (siehe Anhang F) eine Rolle spielen, werden im folgenden erläutert.

Sind Temperatur und Druck für die zu berechnenden physikalischen Eigenschaften bekannt, so besteht die einfachste Möglichkeit darin, eine MD- oder eine MC-Simulation der Flüssigkeit bei dem vorgegebenen Druck und der vorgegebenen Temperatur durchzuführen, also eine NPT-Simulation ohne explizite Gasphase. Der damit verbundene Nachteil ergibt sich sofort: Durch die Abwesenheit der Gasphase gibt es keine klare Trennung der beiden Phasen. Der Erhalt eines Zweiphasengleichgewichts ist somit praktisch unmöglich, und die Systemeigenschaften, die sich

aus dem molekularen Modell berechnen lassen, sind keinesfalls diejenigen, die im Gleichgewicht auf der Phasenübergangskurve vorliegen. Da die Flüssigkeit jedoch zu demselben Druck und derselben Temperatur simuliert wird wie es im Zweiphasengleichgewicht der Fall ist, sind die so ermittelten Systemeigenschaften jedoch zumeist gute Näherungen für die Gleichgewichtseigenschaften, denn die Simulation der Flüssigkeit ist nahezu gleich. Was dadurch nicht erzielt werden kann, ist, daß ein Teilchen aus der flüssigen Phase in die Gasphase übergehen kann und umgekehrt. Beim Gas wird angenommen, daß es sich um ein ideales Gas handelt. Dies ist allerdings insbesondere in der Nähe der kritischen Temperatur problematisch. Durch eine nachgeschaltete großkanonische Gassimulation können die Systemeigenschaften jedoch korrigiert werden, worauf später noch eingegangen wird. Da diese Korrekturen oftmals klein sind, können in vielen Fällen die aus der NPT-Simulation der Flüssigkeit erhaltenen Systemeigenschaften verwendet werden. Die auf Panagiotopoulos (1987) und Panagiotopoulos u. a. (1988) zurückgehende *Gibbs-Ensemble-MC-(GEMC)-Methode* betrachtet innerhalb einer Box die explizite Trennung der flüssigen und gasförmigen Phase, welche gleichzeitig simuliert werden. Fluktuationen innerhalb einer Phase können jedoch Auswirkungen auf die andere Phase haben, so daß insbesondere in der Nähe des Tripelpunkts jede Phase in die jeweils andere übergehen kann, was selbstverständlich unerwünscht ist. Ein weiterer Nachteil ist die Tatsache, daß besonders bei hohen Dichten der im Rahmen dieser Methode verwendete Teilchenaustausch zwischen den beiden Phasen zu extrem hohen statistischen Unsicherheiten in der flüssigen Phase führen kann. Ähnliches wurde bereits in Abschnitt 2.5.2 bei der Darstellung der Widom-Methode zur Berechnung des chemischen Potentials diskutiert. Hinzu kommt, daß bei der GEMC-Methode die Einstellung des Phasengleichgewichts oftmals mit einem sehr hohen Rechenaufwand verbunden ist. Beide der hier vorgestellten Methoden wurden im Rahmen dieser Arbeit verwendet. Um jedoch exakte Phasengleichgewichte zu bestimmen, erscheint das im folgenden beschriebene Verfahren als äußerst sinnvoll: Die sogenannte *Grand-Canonical-MC-(GCMC)-Methode* wurde erstmals von Vrabec u. Hasse (2002) eingesetzt und betrachtet die beiden Phasen sukzessive. Zunächst wird eine NPT-Simulation der Flüssigkeit durchgeführt, aus der das chemische Potential μ in Abhängigkeit vom Druck und dessen Ableitung nach dem Druck, ebenfalls in Abhängigkeit vom Druck, berechnet werden. Die resultierende Kurve $\mu(P)$ wird anschließend für eine μVT-Simulation des Gases angewendet. Durch die Bedingung $\mu \equiv const.$ wird das Phasengleichgewicht realisiert, und der in diesem Gleichgewicht herrschende Druck wird, ähnlich wie bei der GEMC-Methode, aus der Simulation berechnet. Eine direkte Kopplung zwischen den Phasen wird durch die GCMC-Methode vermieden, so daß die für die GEMC-Methode angesprochenen Nachteile umgangen werden können. Das Verfahren besteht aus den folgenden Teilschritten:

1. **NPT-Simulation der Flüssigkeit:** Zu vorgegebener Temperatur T und einem Druck P_0, der nicht allzu weit vom Dampfdruck p_σ entfernt liegen sollte, wird für die Flüssigkeit eine NPT-Simulation durchgeführt, bei der das chemische Potential μ berechnet wird. Bei hohen Temperaturen, also bei geringeren Dichten, wird hierzu die Widom-Methode zusammen mit MD verwendet, da dies das schnellste Verfahren ist. Die Realisierung einer NPT-Simulation mit MD ist in Anhang A.2 dargestellt. Bei niedrigen Temperaturen, also bei hohen Dichten, wird die Methode der graduellen Einsetzung angewandt (vergleiche Abschnitt 2.5.2). Die Realisierung einer NPT-Simulation mit MC ist in Anhang A.3 dargestellt. In der Praxis wird für die Temperatur ein Schwellenwert ϑ festgelegt. Für $T \leq \vartheta$ wird das chemische Potential mittels gradueller Einsetzung und für $T > \vartheta$ mittels der Widom-Methode berechnet.

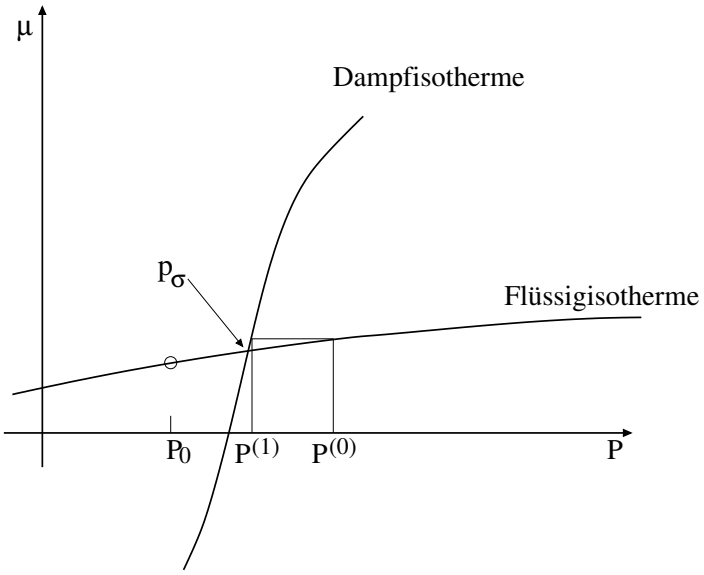


Abbildung A.1: Schematische Darstellung zum Erhalt des Dampfdrucks p_σ im Zweiphasengleichgewicht bei einer Pseudo-μVT-Simulation des Dampfes: Der Druck P_0 aus der NPT-Simulation der Flüssigkeit legt zusammen mit dem chemischen Potential $\mu(P)$ und dem partiellen Molvolumen V_m die Flüssigisotherme fest. Startet die Gassimulation bei einem Druck $P^{(0)}$, so ist das initiale chemische Potential $\mu(P^{(0)})$ durch diese Isotherme ebenfalls festgelegt, welches dem chemischen Potential auf der Dampfisotherme zu einem anderen Druck $P^{(1)}$ entspricht. Dieser Druck liegt näher an p_σ als $P^{(0)}$, und man kann sich die Simulation als einen iterativen Prozeß vorstellen, welcher zwangsläufig gegen p_σ beziehungsweise $\mu(p_\sigma)$ konvergiert.

2. **Berechnung von** $\mu(P)$: Aus der NPT-Simulation der Flüssigkeit wird $\mu_0 := \mu(P_0)$ ermittelt. Eine Taylorentwicklung erster Ordnung liefert

$$\mu(P) \doteq \mu_0 + \left(\frac{\partial \mu}{\partial P}\right)_T (P)\,(P - P_0) = \mu_0 + V_m(P)\,(P - P_0)\,, \tag{A.15}$$

wobei $V_m(P)$ das partielle Molvolumen ist, welches innerhalb der Widom-Methode oder des Verfahrens der graduellen Einsetzung gemäß

$$V_m(P) = \frac{\left\langle V^2 \exp\left(-\beta_T^{-1} U_{\text{pot}}\right)\right\rangle_{NPT}}{\left\langle V \exp\left(-\beta_T^{-1} U_{\text{pot}}\right)\right\rangle_{NPT}} - \langle V \rangle \tag{A.16}$$

errechnet werden kann.

3. **μVT-Simulation des Gases:**

Für die Gasphase wird eine μVT-Simulation gemäß Anhang A.4 durchgeführt. Im Grunde genommen handelt es sich dabei um eine *Pseudo-μVT-Simulation*, da nicht μ, sondern $\mu(P)$ konstant gehalten wird. Aus dieser Simulation ergibt sich der Dampfdruck p_σ und das chemische Potential $\mu(p_\sigma)$. Somit ist der Erhalt des Phasengleichgewichts gewährleistet. Abbildung A.1 zeigt eine schematische Darstellung, wie der Dampfdruck aus der Pseudo-μVT-Simulation erhalten wird. Die Flüssigisotherme und die Dampfisotherme sind während der Simulation konstant, das heißt, deren Verlauf ändert sich nicht. Zu einem Startdruck $P^{(0)}$ wird das durch die Flüssigisotherme festgelegte chemische Potential

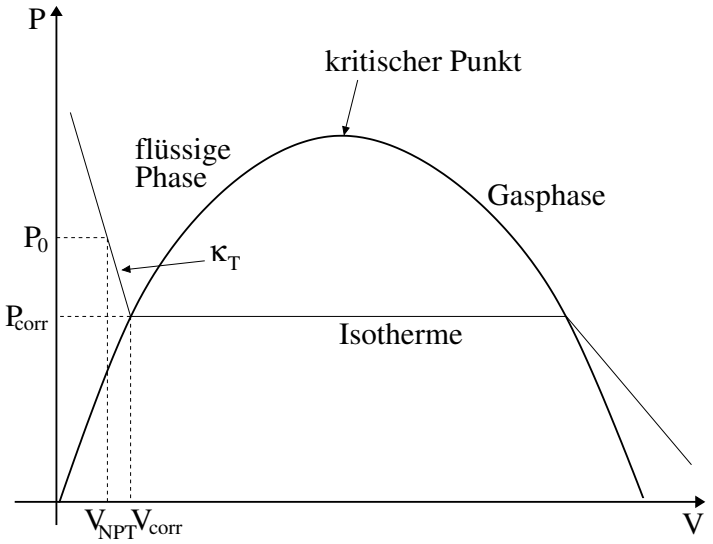


Abbildung A.2: Korrektur der Systemeigenschaften am Beispiel der Dichte: Die Kurve zeigt die Abhängigkeit des Drucks vom Volumen an der Phasenübergangskurve, das heißt, die Temperatur ist auf der Kurve nicht konstant. Die schwarze Linie zeigt eine Isotherme zum Druck P_0 und zum Volumen V_{NPT} aus der NPT-Simulation der Flüssigkeit. Sie schneidet die Kurve im Punkt $(V_{\mathrm{corr}}, P_{\mathrm{corr}})$, wobei $P_{\mathrm{corr}} = p_\sigma$ der korrekte Dampfdruck und V_{corr} das der korrekten Siededichte ρ_l entsprechende Volumen ist. Die Steigung der Isotherme ist gleich der isothermen Kompressibilität κ_T aus Gleichung C.14, die innerhalb der Flüssigsimulation zu bestimmen ist.

$\mu(P^{(0)})$ bestimmt, welches dem des Gases zu einem anderen Druck $P^{(1)}$ entspricht. Dieser Druck kann als Ensemble-Durchschnitt mithilfe von $\mu(P^{(0)})$ berechnet werden und liefert ein anderes chemisches Potential $\mu(P^{(1)})$ auf der Flüssigisotherme. Der so entstehende iterative Prozeß konvergiert im allgemeinen sehr schnell gegen p_σ beziehungsweise $\mu(p_\sigma)$, so daß das Zweiphasengleichgewicht erhalten wird.

4. **Korrektur der Systemeigenschaften:**

   Zum Schluß der großkanonischen Gassimulation werden die Systemeigenschaften der Flüssigkeit korrigiert: Bei der NPT-Simulation der Flüssigkeit soll eine Systemeigenschaft X berechnet werden. Die Korrektur beruht auf einer Taylorentwicklung erster Ordnung um den Flüssigdruck P_0:

$$X(p_\sigma) = X(P_0) + \left(\frac{\partial X}{\partial P}\right)_T (P_0)\,(p_\sigma - P_0)\,, \tag{A.17}$$

   wobei $\left(\frac{\partial X}{\partial P}\right)_T (P_0)$ innerhalb der Flüssigsimulation zu bestimmen ist. Abbildung A.2 zeigt diesen Sachverhalt anhand der Dichte: Die Isotherme, welche durch P_0 und $\left(\frac{\partial X}{\partial P}\right)_T (P_0)$ definiert ist, schneidet die Druck-Volumen-Kurve des Phasengleichgewichts im Punkt $(V_{\mathrm{corr}}, P_{\mathrm{corr}})$, wobei $P_{\mathrm{corr}} = p_\sigma$ der korrekte Dampfdruck und V_{corr} das der korrekten Siededichte ρ_l entsprechende Volumen ist. Die Steigung der Isotherme ist in diesem Fall gleich der isothermen Kompressibilität κ_T aus Gleichung C.14, welche aus der Flüssigsimulation bekannt ist. Diese Korrektur ist für eine Zweiphasensimulation unentbehrlich, ist jedoch in manchen Fällen ausreichend klein, so daß eine NPT-Simulation der flüssigen Phase das Zweiphasengleichgewicht zumindest approximativ beschreiben kann.

B Molekulardynamik mit Nebenbedingungen

Um starre Systeme zu simulieren, fügt man zur betrachteten Bewegungsgleichung Nebenbedingungen hinzu. Die mathematische Idee dabei ist, das System durch Einführung von Lagrange-Multiplikatoren auf ein System ohne Nebenbedingungen zurückzuführen. Die zugehörige Theorie sowie die Entwicklung des daraus resultierenden SHAKE-Algorithmus gehen auf Ryckaert u. a. (1977) zurück.
Man betrachte wieder ein System aus N Teilchen und n Nebenbedingungen $\sigma_k,\ k = 1, ..., n$, die sich auf die Bindungslängen zwischen jeweils zwei Teilchen ki und kj beziehen, mit entsprechenden Ortskoordinaten r_{ki} und r_{kj}. Diese sollen konstant gleich d_k sein, das heißt, insgesamt muß gelten:

$$\begin{aligned} m_i \ddot{r}_i &= -\frac{\partial}{\partial r_i} U(r_i) = F_i,\ i = 1, ..., N, \\ \text{wobei} \qquad & \sigma_k := \|r_{ki} - r_{kj}\|^2 - d_k^2 = 0. \end{aligned} \tag{B.1}$$

Nach Einführung von Lagrange-Multiplikatoren λ_k lautet die Bewegungsgleichung:

$$m_i \ddot{r}_i = \frac{\partial}{\partial r_i} \left(U(r_i) + \sum_{k=1}^{n} \lambda_k \sigma_k \right). \tag{B.2}$$

Es sei $\hat{r}_i$ die Lösung der Newtonschen Bewegungsgleichung ohne Nebenbedingungen. Dann wird zur Erfüllung der Nebenbedingungen im Verlet-Algorithmus die folgende Korrektur vorgenommen:

$$r_i(t + \Delta t) = \hat{r}_i(t + \Delta t) + \sum_{k=1}^{n} \frac{\lambda_k}{m_i} \frac{\partial \sigma_k(t)}{\partial r_i(t)} (\Delta t)^2. \tag{B.3}$$

Dabei ist $\sigma_k(t) := \sigma_k(r(t))$. Das Ziel besteht nun darin, daß die Nebenbedingungen auch zum Zeitpunkt $t + \Delta t$ erfüllt sein sollen. Es soll gelten:

$$\begin{aligned} \forall_{k=1,...,n}\ \sigma_k(t + \Delta t) &= \|r_{ki}(t + \Delta t) - r_{kj}(t + \Delta t)\|^2 - d_k^2 \\ &= \left\| \hat{r}_{ki}(t + \Delta t) - \hat{r}_{kj}(t + \Delta t) + \sum_{k=1}^{n} \lambda_k \left(\frac{\partial \sigma_k(t)}{\partial r_{ki}} m_{ki}^{-1} - \frac{\partial \sigma_k(t)}{\partial r_{kj}} m_{kj}^{-1} \right) \right\|^2 \\ &- d_k^2 \stackrel{!}{=} 0. \end{aligned}$$

Dies ist ein nichtlineares Gleichungssystem für die Lagrange-Multiplikatoren $\lambda_k,\ k = 1, ..., n$, das sich zum Beispiel mit dem Newton-Verfahren lösen läßt. Es seien $\lambda := (\lambda_k)_{k=1,...,n}$ und

$\sigma := (\sigma_k)_{k=1,\ldots,n}$. Dann betrachtet man die folgenden Newton-Iterationen:

$$\begin{aligned} \lambda^{(\ell+1)} &= \lambda^{(\ell)} - (J^{(\ell)})^{-1}\sigma, \\ \text{wobei} \qquad & J^{(\ell)} = (J_{kl}^{(\ell)})_{k,l=1,\ldots,n} = \left(\frac{\partial \sigma_k}{\partial \lambda_l^{(\ell)}}\right)_{k,l=1,\ldots,n}. \end{aligned}$$

Die Jacobi-Matrix $J^{(\ell)}$ ist blockdiagonal, da sich die Nebenbedingungen auf Bindungen in jedem Molekül beziehen. Bezieht sich eine Nebenbedingung nicht auf eine bestimmte Bindung innerhalb eines Moleküls, so ist der entsprechende Matrixeintrag gleich Null, ansonsten wiederholen sich die Einträge von Molekül zu Molekül. Nach jeder Iteration löst man die modifizierte Bewegungsgleichung (B.2) gemäß Gleichung (B.3) mit dem entsprechenden $\lambda^{(\ell)}$, bis sämtliche Nebenbedingungen für alle Zeitschritte erfüllt sind. Dies bedeutet zusätzlichen Rechenaufwand, da für jede Iteration $\lambda^{(\ell)}$ die entsprechende Bewegungsgleichung gelöst werden muß.
Zum Erhalt von Bindungswinkeln $\phi = \phi_0$ zwischen drei Teilchen a, b und c hält man die Strecke d_{ac} gemäß dem Kosinussatz konstant:

$$d_{ac}^2 = d_{ab}^2 + d_{bc}^2 - 2d_{ab}d_{bc}\cos\phi_0. \tag{B.4}$$

Zum Erhalt von Diederwinkel $\omega = \omega_0$ zwischen vier Teilchen a, b, c und d hält man die Winkel ϕ_{abc} und ϕ_{bcd} konstant:

$$\omega_0 = \arccos\left(\frac{(d_{ab}\times d_{cb})(d_{bc}\times d_{dc})}{\sin\phi_{abc}\sin\phi_{bcd}}\right). \tag{B.5}$$

Der auf Ryckaert u. a. (1977) zurückgehende SHAKE-Algorithmus war die erste Methode zur Miteinbeziehung von Nebenbedingungen. Wenn ohne Nebenbedingungen im nächsten Schritt die Koordinaten r_i', $i = 1, ..., N$, auftreten, so betrachtet man die sogenannten SHAKE-Korrekturen $\Delta r_i = r_i - r_i'$. Die Lösung des LGS $J\lambda = \sigma$ erfolgt durch eine Standard-Gauß-Seidel-Relaxation. In dieser Form ist das Verfahren jedoch nicht für Geschwindigkeiten geeignet. Daher entwickelte Andersen (1983) den RATTLE-Algorithmus, welcher zusätzlich einen Geschwindigkeits-Verlet-Algorithmus betrachtet. Dabei sind zusätzlich zu den λ_k Lagrange-Multiplikatoren μ_k für die Geschwindigkeiten zu bestimmen. Tabelle B.1 zeigt die wichtigsten bisher entwickelten Methoden zu MD-Simulationen mit Nebenbedingungen. Es gibt verschiedene Versionen von SHAKE, die sich zumeist in der Lösungsmethode des nichtlinearen Gleichungssystems für die Lagrange-Multiplikatoren unterscheiden. Die sogenannte EEM-Methode, welche auf Edberg u. a. (1986) zurückgeht, betrachtet $\frac{d^2\sigma}{dt} = 0$ anstatt $\sigma = 0$. Die zweite Ableitung nach der Zeit wird jedoch über eine Diskretisierung berechnet. Daher entstehen Fehler, und eine entsprechende Korrektur ist stets vonnöten.

Eine weitere wichtige Klasse von Algorithmen für MD-Simulationen mit Nebenbedingungen sind die sogenannten *Projektionsmethoden*, welche beispielsweise in Mazur (1999) beschrieben sind. Es hierzu die Newtonsche Bewegungsgleichung in Matrixform betrachtet:

$$\frac{d^2 r}{dt^2} = M^{-1} f. \tag{B.6}$$

Dabei beschreiben die Vektoren $r \in \mathbb{R}^{3N}$ und $f \in \mathbb{R}^{3N}$ die Koordinaten beziehungsweise Kräfte, bezogen auf N Systemteilchen. Die Matrix $M \in \mathbb{R}^{3N\times 3N}$ ist die sogenannte *Massenmatrix*,

Algorithmus	Autor(en)	Beschreibung
SHAKE	Ryckaert u. a. (1977)	Erster Algorithmus für Nebenbedingungen; nur für Koordinaten; Lösung des LGS durch Standard-Gauß-Seidel-Verfahren
RATTLE	Andersen (1983)	SHAKE + Geschwindigkeits-Verlet
SETTLE	Miyamoto u. Kollman (1992)	$n = 3$: analytische Lösung des nichtlinearen Gleichungssystems
Matrixmethode	Yoneya u. a. (1994)	Einmalige Invertierung der Jacobi-Matrix, dann vereinfachtes Newton
MSHAKE	Lambrakos u. a. (1989)	Verbesserte Konvergenz bzgl. Koordinaten; Nebenbedingungen bzgl. Kräfte
WIGGLE	Lee u. a. (2005)	Verbesserte Konvergenz bzgl. Koordinaten und Geschwindigkeiten; Nebenbedingungen bzgl. Beschleunigungen
P-SHAKE	Gonnet (2007)	Präkonditionierung der Iterationsmatrix
QSHAKE	Forester u. Smith (1998)	Erweiterung von SHAKE auf aromatische Ringsysteme
M-SHAKE	Kräutler u. a. (2001)	LR-Zerlegung der Iterationsmatrix; Variante: direkte Lösung mit CG-Verfahren
EEM	Edberg u. a. (1986)	2. Zeitableitung der Nebenbedingungen = 0
Projektionsmethoden	z.B. Mazur (1999)	Projektion auf ursprüngliche Bindungen
LINCS	Hess u. a. (1997)	Approximation $(I - A_n)^{-1} = \sum_{i=1}^{\infty} A_n^i$, wobei A_n eine Konnektivitätsmatrix ist
P-LINCS	Hess (2008)	Parallelisierte Version von LINCS: jedes Reihenglied wird parallel berechnet

Tabelle B.1: Überblick über MD-Verfahren mit Nebenbedingungen.

deren Einträge $(m_{ij})_{i,j=1,...,3N}$ gegeben sind durch $m_{ij} = \partial^2 K/(\partial \dot{r}_i \partial \dot{r}_j)$, wobei K die kinetische Energie ist. Die Massenmatrix ist symmetrisch positiv definit. Weiterhin wird die Nebenbedingungen $g_i(r) = 0,\ i = 1, ..., k$, betrachtet. Durch Einführung von Lagrange-Multiplikatoren λ erhält man:

$$-M\frac{d^2 r}{dt} = \frac{\partial}{\partial r}(U - \lambda g).$$

Es sei $B = (b_{hi})_{h=1,...,k;i=1,...,3N} := \frac{\partial g_h}{\partial r_i} \in \mathbb{R}^{k \times 3N}$. Dann gilt:

$$-M\frac{d^2 r}{dt} + B^T \lambda + f = 0. \tag{B.7}$$

Wegen $g = 0$ folgt für die Zeitableitung von g:

$$\frac{dg}{dt} = \frac{\partial g}{\partial r}\frac{dr}{dt} = B\frac{dr}{dt} = 0, \tag{B.8}$$

und für die zweite Ableitung gilt:

$$\frac{d^2 g}{dt^2} = B\frac{d^2 r}{dt^2} + \frac{dB}{dt}\frac{dr}{dt} = 0. \tag{B.9}$$

Multipliziert man Gleichung (B.7) von links mit BM^{-1}, so erhält man:

$$\begin{aligned} -B\frac{d^2r}{dt^2} + BM^{-1}B^T\lambda + BM^{-1} &= 0 \\ \overset{\text{(B.9)}}{\Rightarrow} \frac{dB}{dt}\frac{dr}{dt} + BM^{-1}B^T\lambda + BM^{-1} &= 0 \\ \Rightarrow B^T\lambda = -B^T(BM^{-1}B^T)^{-1}BM^{-1}f &- B^T(BM^{-1}B^T)^{-1}\frac{dB}{dt}\frac{dr}{dt}. \end{aligned}$$

Die Matrix $T := M^{-1}B^T(BM^{-1}B^T)^{-1} \in \mathbb{R}^{3N\times k}$ bildet die bedingten Koordinaten in die kartesischen Koordinaten ab. Setzt man dies für $B^T\lambda$ in Gleichung (B.7) ein, so ergibt sich:

$$\frac{d^2r}{dt^2} = (I - TB)M^{-1}f - T\frac{dB}{dt}\frac{dr}{dt}. \tag{B.10}$$

Dabei ist $I - TB$ die sogenannte *Projektionsmatrix*, welche die ursprünglichen Koordinaten, für die die Nebenbedingungen erfüllt sind, auf 0 setzt.
Die Bewegungsgleichung (B.10) läßt sich mit einem Bocksprung-Verlet-Algorithmus lösen:

$$\begin{aligned} \frac{v_{n+\frac{1}{2}} - v_{n-\frac{1}{2}}}{\Delta t} &= (I - T_nB_n)M^{-1}f_n - T_n\frac{B_n - B_{n-1}}{\Delta t}v_{n-\frac{1}{2}} \\ \frac{r_{n+1} - r_n}{\Delta t} &= v_{n+\frac{1}{2}}. \end{aligned} \tag{B.11}$$

Diese Methode projiziert allerdings nur die neuen auf die alten Bindungen. Die Länge ℓ_i der neuen Bindungen ist jedoch nicht gleich der der alten Bindungen, die hier mit d_i bezeichnet wird. Daher wird ein Korrekturterm p_i benötigt. Die korrigierten Koordinaten r'_{n+1} werden auf folgende Weise ermittelt:

$$\begin{aligned} r'_{n+1} &= (I - T_nB_n)r_{n+1} + T_np = (I - T_nB_n)r^{\text{oN}}_{n+1} + T_nd \\ &= r^{\text{oN}}_{n+1} - M^{-1}B_n(B_nM^{-1}B_n^T)^{-1}(B_nr^{\text{oN}}_{n+1} - d). \end{aligned} \tag{B.12}$$

Dabei sind r^{oN} die Koordinaten ohne Nebenbedingungen.
Ein Problem besteht noch in der Inversion der Matrix $B_nM^{-1}B_n^T$. Das sogenannte LINCS-Verfahren, entwickelt von Hess u.a. (1997), stellt die Matrix in der Form $D(I - A_n)^{-1}D$ dar, wobei D eine Diagonalmatrix ist. Die symmetrische Matrix A_n ist dünnbesetzt. Ihre Diagonalelemente sind gleich 0. Die echten Einträge von A_n beziehen sich auf die Konnektivität innerhalb der Moleküle. Daher ist die Methode nur auf Moleküle mit geringer Konnektivität anwendbar, denn nur dann gilt für den Spektralradius $\rho(A_n) < 1$, und es kann die geometrische Reihe $(I - A_n)^{-1} = \sum_{i=1}^{\infty} A_n^i$ verwendet werden.
Die Methode kann folgendermaßen parallelisiert werden: Die Reihenglieder A_n^i beziehen sich auf Bindungen, zwischen denen $i - 1$ weitere Bindungen liegen. Die sogenannte P-LINCS-Methode wurde von Hess (2008) eingeführt und betrachtet die Potenzen von A_n parallel.

C Berechnung relevanter Systemeigenschaften

In diesem Anhang werden sämtliche Formeln angegeben, mithilfe derer die in dieser Arbeit relevanten Systemeigenschaften berechnet werden. Anhang C.1 behandelt dabei statische und Anhang C.2 dynamische Eigenschaften.

C.1 Statische Eigenschaften

Es seien $N_A := 6.022 \cdot 10^{23}$ mol^{-1} die Avogadro-Zahl, $R := 8.314$ J mol^{-1} K^{-1} die ideale Gaskonstante und M die molare Masse in g mol^{-1}. Im folgenden bezeichnet P keine Wahrscheinlichkeiten, sondern Drücke.

- **Energie:** Die potentielle Energie des äquilibrierten Systems U_{tot} wird aus dem intramolekularen Potential U_{intra} aus Abschnitt 2.2.1 und dem intermolekularen U_{nb} aus den Abschnitten 2.2.2 bis 2.2.3 berechnet:

$$\langle U_{\text{pot}} \rangle = \langle U_{\text{intra}} \rangle + \langle U_{\text{nb}} \rangle. \tag{C.1}$$

  Die Äquilibriumkonformation ist dabei diejenige Konformation mit niedrigster interner Energie für $T \to 0$. Diese wird von der MD-Simulation berechnet.
  Die kinetische Energie ist gegeben durch

$$\langle U_{\text{kin}} \rangle = \frac{1}{2} \sum_{i=1}^{N} m_i \langle v_i^2 \rangle = \frac{3N}{2} k_B \langle T \rangle. \tag{C.2}$$

  Es ist zu beachten, daß die Temperatur in kanonischen Ensembles durch die in Abschnitt A.1 beschriebenen Methoden konstant gesetzt wird, jedoch auch einem Äquilibrierungsprozeß unterliegt. Somit ist auch hier ein Mittelwert $\langle T \rangle$ zu berechnen.

- **Verdampfungsenthalpie:** In sehr engem Bezug zum intermolekularem Potential U_{nb} steht die Verdampfungsenthalpie $\Delta_v H$ gemäß der folgenden Näherungsformel:

$$\langle \Delta_v H \rangle \approx -\frac{\langle U_{\text{nb}} \rangle N_A}{N} + R \langle T \rangle. \tag{C.3}$$

  Die Verdampfungsenthalpie ist die Differenz aus der Enthalpie des Gases und der Enthalpie der Flüssigkeit, welche miteinander im Phasengleichgewicht stehen. Die Enthalpie ist definiert als $H = U + PV$, wobei U die innere Energie und PV die Volumenarbeit ist. Die Volumenarbeit der Flüssigkeit und die innere Energie des Gases werden in Gleichung

(C.3) vernachlässigt. Die innere Energie der Flüssigkeit ist gegeben durch $-\frac{\langle U_{\mathrm{nb}}\rangle N_A}{N}$ und die Volumenarbeit des Gases durch $R\langle T\rangle$.
Mithilfe des Dampfdrucks p_σ und den potentiellen Energien für Dampf U_v und Flüssigkeit U_l läßt sich die Verdampfungsenthalpie auch folgendermaßen berechnen:

$$\langle \Delta_v H\rangle = N_A \left(\left\langle \frac{U_v}{N_v} \right\rangle - \left\langle \frac{U_l}{N_l} \right\rangle + \langle p_\sigma \rangle \left(\left\langle \frac{V_v}{N_v} \right\rangle - \left\langle \frac{V_l}{N_l} \right\rangle \right) \right). \tag{C.4}$$

- **Temperatur:** Die Temperatur läßt sich aus der Verteilung Ω der inneren Energie U berechnen:

$$T = \frac{1}{k_B} \left(\frac{\partial \log \Omega}{\partial U} \right)^{-1}. \tag{C.5}$$

Dabei ist $k_B := 1.3806504 \cdot 10^{-23}$ J/K die sogenannte *Boltzmann-Konstante*. Als zeitlicher Durchschnitt ist sie auch durch die zeitlichen Durchschnitte der Geschwindigkeiten erhältlich:

$$\langle T\rangle = \frac{M\langle v^2\rangle}{3R}. \tag{C.6}$$

- **Druck:** Der Druck ist direkt aus der Cheung-Formel (A.10) und aus der Beziehung zwischen Temperatur und kinetischer Energie (A.1) erhältlich:

$$\langle P\rangle = \frac{1}{3\langle V\rangle} \left(\sum_{i=1}^{N} m_i \langle v_i^2\rangle + \left\langle \sum_{i=1}^{N} \langle r_i, f_i\rangle \right\rangle \right). \tag{C.7}$$

Weitere Drücke sind durch die folgenden Formeln gegeben:

  - Hydrodynamischer Druck: $\langle P_{\mathrm{hydro}}\rangle = \frac{1}{2}\langle \rho v^2\rangle$. Dabei ist ρ die Dichte des Systems.
  - Gasdruck: $\langle P_v\rangle \cdot \langle V\rangle = N \cdot R \cdot \langle T\rangle$ (ideale Gasgleichung).

- **Dichte:** Mit dem zeitlichen Durchschnitt des Volumens erhält man direkt den für die Dichte des Systems in einem NPT-Ensemble:

$$\langle \rho\rangle = \frac{MN}{\langle V\rangle N_A}. \tag{C.8}$$

Die Lösung der Newtonschen Bewegungsgleichung gibt die Teilchenkoordinaten zu jedem Zeitpunkt an, woraus sich auch das Volumen des Systems berechnen läßt. Der Durchschnittswert über alle Zeitschritte ergibt dann $\langle V\rangle$.
In einem Gibbs-Ensemble ist Flüssigkeitsdichte ρ_l durch

$$\langle \rho_l\rangle = \frac{\langle N_l\rangle M}{\langle V_l\rangle N_A} \tag{C.9}$$

oder approximativ durch

$$\langle \rho_l\rangle = \left\langle \frac{N_l}{V_l} \right\rangle \frac{M}{N_A} \tag{C.10}$$

bestimmbar.

Fluktuationen lassen sich mithilfe der Varianz

$$\sigma^2(A) := \left\langle (A - \langle A \rangle)^2 \right\rangle ,$$

welche physikalisch allgemein als Aufnahmefähigkeit beziehungsweise Suszeptibilität interpretiert wird, berechnen:

- **Wärmekapazität:** Für die Wärmekapazität in einem NPT-Ensemble gilt:

$$C_P = \left(\frac{\partial H}{\partial T} \right)_P . \tag{C.11}$$

  Sie gibt an, in welchem Maße ein System dazu in der Lage ist, Wärme aufzunehmen, das heißt, wie sich die Enthalpie in Abhängigkeit von der Temperatur ändert beziehungsweise wie viel Energie notwendig ist, um die Systemtemperatur um einen bestimmten Betrag zu erhöhen. Die rechte Seite von Gleichung (C.11) wird über finite Differenzen ermittelt.

- **Isochorer Spannungskoeffizient:** Es handelt sich hierbei um die Veränderung des Drucks in Abhängigkeit von der Temperatur bei gegebenem Volumen:

$$\gamma_V = \left(\frac{\partial P}{\partial T} \right)_V . \tag{C.12}$$

  Er wird berechnet nach der Formel

$$\langle (U_{\text{pot}} - \langle U_{\text{pot}} \rangle)(W - \langle W \rangle) \rangle = k_B T^2 (V \gamma_V - N k_B), \tag{C.13}$$

  wobei W das Virial ist: $W := \sum_{i=1}^{N} \langle r_i, f_i \rangle$.

- **Isotherme Kompressibilität:** Sie gibt an, wie sich das Volumen eines Systems bei Änderung des äußeren Drucks und konstanter Temperatur verändert:

$$\kappa_T = -\frac{1}{V} \left(\frac{\partial V}{\partial P} \right)_T = \frac{1}{k_B T V} \sigma^2(V). \tag{C.14}$$

Wie zwei bestimmte Systemeigenschaften A und B miteinander korreliert sind, gibt der sogenannte **Korrelationskoeffizient** an:

$$c_{AB} = \frac{\langle (A - \langle A \rangle)(B - \langle B \rangle) \rangle}{\sigma(A)\sigma(B)}. \tag{C.15}$$

Der Korrelationskoeffizient ist direkt aus der Mathematischen Statistik entnommen. Dabei wird die Kovarianz von A und B durch deren Standardabweichungen dividiert. Werden A und B in Abhängigkeit von der Zeit t betrachtet, so entsteht aus (C.15) die sogenannte *Zeitkorrelationsfunktion*. Dabei sind die Eigenschaften zu zwei aufeinanderfolgenden Zeitschritten oft korreliert. Starke Unkorreliertheiten und statistische Unabhängigkeiten treten erst zu späteren Zeitpunkten auf. Analog zum Korrelationskoeffizient aus der Statistik gilt:

- $c_{AB} > 0 \Rightarrow$ Korrelation,
- $c_{AB} = 1 \Rightarrow$ perfekte Korrelation,
- $c_{AB} < 0 \Rightarrow$ Antikorrelation,
- $c_{AB} = 0 \Rightarrow$ keine Korrelation.

C.2 Dynamische Eigenschaften

Durch instantane Fluktuationen wird ein System aus dem Gleichgewicht gebracht. Die Stärke derartiger Fluktuationen sind ein Maß für dynamische Eigenschaften, welche im folgenden auch Transporteigenschaften genannt werden. Transporteigenschaften werden durch die zeitlichen Veränderungen innerhalb des Systems berechnet. Dies kann nur mit MD-Simulationen realisiert werden und geschieht zumeist über die Autokovarianz- oder Autokorrelationsfunktion bestimmter Systemeigenschaften. Für die Autokovarianzfunktion der Geschwindigkeit gilt beispielsweise:

$$c_v(t) := \langle v(t)v(0)\rangle. \tag{C.16}$$

Dadurch wird also die zeitliche Veränderung in Bezug auf den Startpunkt $t = 0$ beschrieben. Transporteigenschaften C werden im allgemeinen durch gemittelte quadratische Abweichungen thermodynamischer Eigenschaften vom Startwert (Einstein-Darstellung) oder durch Integration über die Autokovarianzfunktion ihrer zeitlichen Veränderungen (Green-Kubo-Darstellung) ermittelt:

$$C = \frac{c}{t}\left\langle (A(t) - A(0))^2 \right\rangle = \tilde{c}\int_0^\infty \langle \dot{A}(t)\dot{A}(0)\rangle dt. \tag{C.17}$$

Dabei sind c und $\tilde{c}$ spezifische Konstanten. Oftmals ist $\dot{A}$ ein Fluß (zum Beispiel Massen-, Impuls- oder Energiefluß), der proportional zu einer treibenden Kraft oder einem Feld ist.
In dieser Arbeit sind die folgenden Transporteigenschaften relevant:

- **Selbstdiffusionskoeffizient:**

$$D = \frac{1}{6t}\left\langle (r(t) - r(0))^2 \right\rangle = \frac{1}{3}\int_0^\infty \langle v(t)v(0)\rangle dt. \tag{C.18}$$

- **Scherviskosität:** Die Scherviskosität η spielt vor allem bei Makromolekülen eine große Rolle. Eine Scherung ist geometrisch als die Deformation eines Quaders in einen Spat mit Winkel $90° - \gamma$ vorstellbar. Dazu ist eine bestimmte Kraft τ erforderlich. Ein Zusammenhang zwischen dieser *Scherkraft* und der *Scherrate* $\dot{\gamma}$ besteht durch die Beziehung

$$\eta = \frac{\tau}{\dot{\gamma}}. \tag{C.19}$$

  Eine Green-Kubo-Darstellung der Scherviskosität ist gegeben durch die Drucktensorelemente $P_{\alpha\beta}$, die mithilfe von Gleichung (C.7) bestimmt werden:

$$\eta = \frac{V}{3k_BT}\int_0^\infty \langle P_{\alpha\beta}(t)P_{\alpha\beta}(0)\rangle dt. \tag{C.20}$$

  Dabei steht $P_{\alpha\beta}$ für den Druck, der orthogonal zur $\alpha\beta$-Ebene ausgeübt wird.

- **Wärmeleitfähigkeit:** Mithilfe der Autokovarianzfunktion des Wärmeflusses j_W läßt sich die Wärmeleitfähigkeit berechnen:

$$\lambda_T = \frac{V}{k_BT^2}\int_0^\infty \langle j_W(t)j_W(0)\rangle dt. \tag{C.21}$$

- **Molekulare Reorientierung der Dipole:** Diese ist insbesondere in der physikalischen Chemie relevant, da sie einfach zu messen ist. Ein Einheitsvektor u wird am betrachteten Molekül fixiert, der in Richtung des Dipols zeigt. Die Autokovarianzfunktion ist dann gegeben durch

  $$\langle u(t)u(0)\rangle = \langle \Phi(t)\rangle,$$

  wobei $\Phi(t)$ der Winkel zwischen $u(t)$ und $u(0)$ ist. Nach dem Debye-Modell gilt:

  $$\langle u(t)u(0)\rangle = \exp\left(-\frac{2t}{\tau}\right), \tag{C.22}$$

  wobei τ die sogenannte *Rotationsrelaxationszeit* ist, nach der der Dipolvektor genau in die entgegengesetzte Richtung zeigt. Dies wird in dieser Arbeit auch einfach nur *Reorientierungszeit* genannt.

- **Dielektriztätskonstante:** Dabei handelt es sich um eine substanzspezifische Eigenschaft eines sogenannten *Dielektrikums*. Dabei handelt es sich nach der *Clausius-Mosottischen Theorie* um ein System, in dem die Moleküle als elektrische Leiter angesehen werden und durch induktive Kräfte eine Trennung der Ladungen stattfindet. Dadurch werden Anziehungen geladener Körper, die sich in einem derartigen System befinden, erschwert. Die Theorie der Dielektrizität beruht auf dem Dipolmoment

  $$\mu = \sum_{i=1}^{N} q_i r_i^q, \tag{C.23}$$

  welcher sich aus den Ladungen q_i und den Koordinaten r_i^q relativ zu den Ladungszentren errechnet. Die Dielektrizitätskonstante ϵ ist durch die *Clausius-Mosotti-Gleichung* berechenbar:

  $$\frac{\langle \mu^2\rangle - \langle \mu\rangle^2}{3\epsilon_0 V k_B T} = \frac{(2\epsilon_{\mathrm{RF}}+1)(\epsilon-1)}{2\epsilon_{\mathrm{RF}}+\epsilon}. \tag{C.24}$$

  Dabei sind ϵ_0 analog zu Abschnitt 2.2.4 die elektrische Feldkonstante und ϵ_{RF} die Dielektrizitätskonstante des Reaktionsfelds, welche durch quantenmechanische Methoden, zum Beispiel mit *Gaussian* (Frisch u. a., 2004), berechnet werden kann.

D Praktische Umsetzung Molekularer Simulationen

Im folgenden werden die in Abschnitt 2.6.1 angesprochenen Bestandteile der initialen Konfiguration einer Molekularen Simulation und die Zweiteilung des Simulationsverlaufs detailliert erläutert.

Wahl des Ensembles Zunächst muß der Benutzer entscheiden, in welchem Ensemble er simulieren möchte. Gemäß der Statistischen Physik sind zwar alle thermodynamischen Systeme äquivalent, allerdings nur für $N \to \infty$. Simulationen der in dieser Arbeit relevanten Ensembles sind in Anhang A beschrieben.
Entsprechend der Wahl des Ensembles muß ebenfalls festgelegt werden, welches Thermostat beziehungsweise Barostat bei der Simulation verwendet werden soll. Weiterhin muß entschieden werden, nach welchen Gesetzmäßigkeiten die Teilchen interagieren sollen, das heißt, die Wahl des Potentials ist ausschlaggebend. Beispiele für intermolekulare Potentialterme wurden bereits in den Abschnitten 2.2.2 bis 2.2.3 angegeben. Die meisten Simulationsprogramme verwenden die in Abschnitt 2.2.4 eingeführten Gesamtpotentiale. Auch die Größe der in Anhang E.1 angesprochenen Verlet-Nachbarliste bedarf einiger durch den Benutzer vordefinierter, initialer Parameter.

Anfangsbedingungen Nachdem die Wahl des Ensembles und bei MD die Wahl des Integrators der Bewegungsgleichung (siehe Abschnitte 2.3.1 und 2.3.2) getroffen wurden, müssen Anfangsbedingungen bezüglich Koordinaten und im Falle von MD Geschwindigkeiten festgelegt werden:

1. **Anfangsbedingungen für Koordinaten:** Die einfachste Möglichkeit, eine initiale Anordnung von Systemteilchen zu definieren, ist, sie per Zufallsprinzip in der Simulationsbox zu verteilen. Dies kann jedoch zu physikalisch sinnlosen Überlappungen der Teilchen führen. Die potentielle Energie und die daraus resultierenden Kräfte sind dann sehr hoch, was auch zu numerischen Problemen führen würde. Eine bessere Lösung, die in der Praxis meistens verwendet wird, ist die Plazierung der Teilchen auf ein regelmäßiges, dreidimensionales Gitter. Dabei hat sich praktisch jedes Gitter als geeignet erwiesen, aus historischen Gründen wird zumeist jedoch ein kubisch-flächenzentriertes Gitter (*fcc-Packung*) verwendet. Der Gitterzellenabstand wird so gewählt, daß eine initiale Dichte erreicht wird. Die Anordnung von Teilchen auf einem Gitter steht für einen kristallinen Zustand. Im Laufe der Simulation verlassen die Teilchen den kristallinen Zustand, um in einen amorphen, flüssigen und später eventuell auch gasförmigen Zustand überzugehen. Dies ist vor allem von Systemtemperatur und -druck abhängig. Allerdings sind kristalline Strukturen

für Flüssigkeiten unphysikalisch, so daß oftmals auch randomisierte Strukturen verwendet werden, so daß ein Gleichgewichtszustand schneller erreicht werden kann.
Weiterhin ist die initiale Orientierung der Moleküle von Bedeutung. Lineare Moleküle wie CO_2 und N_2 werden dabei entlang der vier Raumdiagonalen einer Gitterzelle orientiert. Bei nichtlinearen Molekülen kann irgendeine geeignete bekannte kristalline Struktur verwendet werden. Geeignet bedeutet hierbei vor allem, daß Überlappungen zweier oder sogar mehrerer Atome ausgeschlossen sind.

2. **Anfangsbedingungen für Geschwindigkeiten:** Es ist bei MD-Simulationen üblich, für die Anfangsgeschwindigkeiten zufällige Werte zu verwenden, allerdings müssen sie einigen physikalisch sinnvollen Bedingungen genügen. Ihre Größenordnung muß beispielsweise der gewünschten Temperatur entsprechen. Außerdem muß der Gesamtimpuls des Systems p_N verschwinden, das heißt es muß gelten:

$$p_N := \sum_{i=1}^{N} m_i v_i = 0. \tag{D.1}$$

Dies kann durch eine Maxwell-Boltzmann-Verteilung mit Wahrscheinlichkeitsdichte

$$f(v_{ix}) := \sqrt{\frac{m_i}{2\pi k_B T}} \exp\left(-\frac{1}{2} m_i \frac{v_{ix}^2}{k_B T}\right) \tag{D.2}$$

erreicht werden. Dabei ist v_{ix} eine Komponente des Geschwindigkeitsvektors v_i.
Eine Alternative besteht darin, gleichverteilte Zufallsgeschwindigkeiten aus einem Intervall $(-v_{\max}, v_{\max})$ zu wählen.

Räumliche Randbedingungen Bezüglich räumlicher Randbedingungen unterscheidet man hauptsächlich zwischen folgenden beiden Arten:

1. **Periodische Randbedingungen:** Eine Simulationszelle wird bei diesen Randbedingungen periodisch wiederholt. Tritt ein Teilchen aus einer Zelle aus, so tritt ein anderes auf der gegenüberliegenden Seite wieder ein. Die Anzahl an Systemteilchen pro Zelle ist somit im Laufe der Simulation konstant. Periodische Simulationszellen können kubisch, orthorhombisch oder abgeschnittene Oktaeder sein.
2. **Semiperiodische Randbedingungen:** Dabei handelt es sich um periodische Randbedingungen in nur einer oder zwei Dimensionen. Ein Teilchen, welches an den Rand der Simulationszelle gelangt, bleibt in der Zelle und wird wieder von der Wand abgestoßen, ähnlich einem Ball, der in einem bestimmten Winkel auf den Boden geworfen wird.
3. **Stochastische Randbedingungen:** Dabei wird eine Einteilung des Raumes in drei Gebiete vorgenommen:
   a) Zentrale Simulationsbox: Hier wird die volle Teilchenbewegung simuliert.
   b) Oberflächenabschnitt fixer Teilchen: Hier sind sämtliche Teilchenpositionen fixiert. Dem Oberflächenabschnitt liegen stochastische Kräfte zugrunde.
   c) Umgebung: Die Wechselwirkungen werden am Rand des Systems stetig approximiert.

Topologie Die Topologie des Systems enthält die Anzahl N an Teilchen, die Größe der initialen Simulationsbox und den räumlichen Aufbau der einzelnen Moleküle. Sie wird also durch die folgenden Systembestandteile definiert:

- **Atome:** Für jedes Atom müssen Name, Nummer, Molekülzugehörigkeit, molare Masse und Ladung angegeben werden. Weiterhin sind sämtliche Potentialparameter zu definieren, also beispielsweise ε und σ im Falle des Lennard-Jones-Potentials.
- **Bindungen:** Die Atombindungen innerhalb der Moleküle werden durch die Angabe von Atomnummern festgelegt. Weiterhin sind Bindungslängen r_0 und Kraftkonstanten k_r (siehe Gleichung (2.14)) anzugeben. Die Bindungslängen werden bei MD in der Praxis oft durch geeignete Techniken konstant gehalten. Derartige Verfahren, die als *MD mit Nebenbedingungen* bekannt sind, werden in Anhang B angesprochen.
- **Winkel:** Auch die Größe der Bindungswinkel ϕ_0 zwischen jeweils drei Molekülatomen ist in der Topologie festzulegen, zusammen mit den Kraftkonstanten k_ϕ (siehe Gleichung (2.15)). Auch Winkel können im Laufe der Simulation konstant gehalten werden.
- **Diederwinkel:** Diederwinkel und uneigentliche Torsionen müssen ebenfalls in der Topologie angegeben werden, zusammen mit den Rotationskonstanten V_n (siehe Gleichung (2.17)) beziehungsweise den Kraftfeldkonstanten k_τ (siehe Gleichung (2.18)). Auch Diederwinkel können konstant gehalten werden.

Äquilibrierung und Produktionslauf Die initiale Konfiguration beinhaltet Anfangsbedingungen für Koordinaten (und Geschwindigkeiten), wobei die initialen Koordinaten im allgemeinen auf einem kubisch-flächenzentrierten Gitter liegen. Diese kristalline Struktur wird im Laufe der Simulation verlassen, und es stellt sich eine für das gewählte Ensemble spezifische Struktur ein. Die Endstruktur befindet sich dann in einem thermodynamischen Gleichgewicht. Dieser Prozeß wird als *Relaxierung* oder *Äquilibrierung* des Systems bezeichnet. Um geeignete physikalische Eigenschaften berechnen zu können, muß sich das System im Gleichgewicht befinden. Die Berechnung geschieht nach Abschnitt 2.5 über thermodynamische Durchschnitte, das heißt, die stichprobenartig ermittelten Eigenschaften müssen im Mittel konstant sein mit möglichst wenig Oszillationen. Da jede Simulation aufgrund von Zufallsalgorithmen (Bestimmung der Geschwindigkeiten, Thermostat, Barostat, siehe Anhang A) mit statistischem Rauschen behaftet ist, sind geringfügige Oszillationen niemals eliminierbar, was auch physikalisch korrekt ist. Die Oszillationen sind von Eigenschaft zu Eigenschaft mehr oder minder stark ausgeprägt. Die Entscheidung, ob ein System äquilibriert ist oder nicht, ist keineswegs trivial. Im folgenden seien ein paar allgemeine Richtlinien angesprochen. Die konkrete praktische Umsetzung eines Äquilibrierungstests wird in Abschnitt 3.5.5 erläutert.
Es werden Äquilibrierungsperioden festgesetzt, nach denen jeweils entschieden wird, ob sich das System im thermodynamischen Gleichgewicht befindet oder nicht. Es ist jeweils ein bestimmter Prozentsatz der bisherigen Gesamtsimulation verwendbar oder auch aufeinanderfolgende Perioden. Im zweiten Fall müssen diese jedoch ziemlich groß gewählt werden, um zu garantieren, daß kein Trend mehr in der betrachteten Eigenschaft vorhanden ist. Die letzte Konfiguration der letzten Äquilibrierungsphase wird dann als Startkonfiguration für die nächste Phase verwendet. In der Praxis werden oftmals die potentielle Energie und der Druck als betrachtete Eigenschaften gewählt, wobei die Oszillationen beim Druck im allgemeinen deutlich größer sind als bei der potentiellen Energie.

Es ist grundsätzlich nicht möglich, eine globale Simulationszeit festzusetzen, nach der jedes beliebige System äquilibriert ist.
Kann letztendlich geschlossen werden, daß sich das System im thermodynamischen Gleichgewicht befindet, wird die letzte Konfiguration als Startkonfiguration für einen sogenannten *Produktionslauf* verwendet. Die in dieser Phase berechneten einzelnen physikalischen Eigenschaften werden gemittelt, und es resultieren geeignete, physikalisch sinnvolle thermodynamische Durchschnitte. Es ist zu beachten, daß sowohl bei der Äquilibrierung als auch bei der Produktion über statistisch unabhängige Werte gemittelt werden muß. Das bedeutet, daß auf keinen Fall alle Werte in die Berechnung miteinbezogen werden sollten, sondern nur diejenigen, die nicht, oder zumindest kaum noch, mit vorher ermittelten Werten korreliert sind. Im allgemeinen ist es sinnvoll, nur etwa jeden tausendsten Wert miteinzubeziehen.

E Effizienzerhöhung bei Molekularen Simulationen

Im folgenden wird beschrieben, wie die Effizienz bei Molekularen Simulationen erhöht werden kann. Da bei MD-Simulationen eingesetzte numerische Methoden auf einer Taylorentwicklung basieren und nach dem zweiten Newtonschen Gesetz, Gleichung (2.10), die Beschleunigung, also die zweite Ableitung des Ortes nach der Zeit, proportional zur Kraft und somit zum negativen Ortsgradienten der potentiellen Energie ist, gehen die Ortsableitungen der in Abschnitt 2.2 besprochenen Kraftfeldparameter in den Algorithmus mit ein. Da diese sich jedoch auf ein System von N Teilchen beziehen, stellt sich die Frage nach einer effizienten Berechnung der Kräfte. Anhang E.1 behandelt die sogenannten *nächsten Nachbarn* in einer Umgebung des jeweils betrachteten Teilchens. Es werden dann lediglich die Wechselwirkungen mit diesen Nachbarn berücksichtigt. In Anhang E.2 wird schließlich dargestellt, wie Kräfte und Potentiale gemäß den in Abschnitt 2.6.2 beschriebenen Ideen effizient berechnet werden können.

E.1 Nächste Nachbarn

Nachdem die Koordinaten zu einem bestimmten Zeitpunkt t beispielsweise mit dem Verlet-Algorithmus aus Abschnitt 2.3.2 berechnet worden sind, müssen sie zum Erhalt neuer Koordinaten zum nachfolgenden Zeitpunkt $t+\Delta t$ in die entsprechenden Potentialterme aus Abschnitt 2.2 eingesetzt werden. Gerade bei den intermolekularen Potentialtermen, welche in den Abschnitten 2.2.2 bis 2.2.3 eingeführt wurden, ist dies bei großen Systemen sehr aufwendig, da sämtliche Wechselwirkungen zwischen den Teilchen berücksichtigt werden müssen. Dies führt zu einer Komplexität von $\mathcal{O}(N^2)$, denn es existieren insgesamt $\binom{N}{2} = \frac{N(N-1)}{2}$ verschiedene Paare von Wechselwirkungszentren.
Um die Komplexität auf $\mathcal{O}(N)$ zu reduzieren, werden für jedes Teilchen nur die Wechselwirkungen mit denjenigen Teilchen betrachtet, die sich in einer Kugel vom Radius $r_N > 0$ des Teilchens befinden, den sogenannten *nächsten Nachbarn*. Die Wechselwirkungen mit Teilchen, die weiter entfernt sind, werden außer acht gelassen. Die Erstellung einer derartigen *Nachbarschaftsliste* mit *Nachbarschaftsradius* r_N ist jedoch nur bei kurzreichweitigen Wechselwirkungen möglich, die zum Beispiel durch das Lennard-Jones-Potential aus Abschnitt 2.2.2 beschrieben werden. Es wird dabei ein Abschneideradius $0 < r_C < r_N$ eingeführt, bei dem das Potential stetig auf Null gesetzt wird. Eine Erfahrungsregel besagt, daß beim Lennard-Jones-Potential $r_C \in [2.5\sigma, 5\sigma]$ gelten soll. Ist die Kantenlänge der Simulationsbox deutlich größer als dieser Abschneideradius, so kann die Komplexität mittels Betrachtung der nächsten Nachbarn von $\mathcal{O}(N^2)$ auf $\mathcal{O}(N)$ reduziert werden.
Es wird zwischen zwei verschiedenen Möglichkeiten unterschieden, um die nächsten Nachbarn zu bestimmen:

1. die *Verlet-Nachbarschaftsliste* bei Systemen mit $N \lesssim 10000$ oder
2. die *Zellenindexmethode* bei Systemen mit $N \gtrsim 10000$ Teilchen.

Diese beiden Methoden werden im folgenden kurz erläutert:

1. Bei der von Verlet (1967) entworfenen Verlet-Nachbarschaftsliste handelt es sich um eine Liste, welche die nächsten Nachbarn eines Systemteilchens enthält und die alle n Zeitschritte aktualisiert wird. Es entsteht dabei eine Kugelschale der Breite $\Delta r := r_N - r_C$ um jedes Teilchen.
   Man betrachte ein Systemteilchen i. Falls für ein weiteres Teilchen j die Bedingung

   $$r_{ij} < r_C + \Delta r = r_N$$

   erfüllt ist, so sind i und j Nachbarn. Der Aufwand, alle Teilchen zu finden, die dieser Ungleichung genügen, beträgt $\mathcal{O}(N^2)$. Zur Reduktion der Komplexität werden nur diejenigen Paare behandelt, welche bereits in der Verlet-Nachbarschaftsliste vorhanden sind. Der resultierende Aufwand liegt dann nur noch bei $\mathcal{O}(N)$ Operationen. Die Bestimmung der Verlet-Nachbarschaftsliste geschieht über eine weitere Liste, die für jedes Teilchen den *ersten* Nachbarn enthält und hier als *point* bezeichnet wird. Dann ist $point(i+1)-1$ der *letzte* Nachbar des Teilchens i, wenn $point(i)$ der erste Nachbar von i ist. Diese und die Verlet-Nachbarschaftsliste werden zum Zeitpunkt $t = 0$ initialisiert und alle n Schritte aktualisiert. Die Kugelschale muß breit genug gewählt werden, so daß weiter voneinander entfernte Moleküle nicht in den Kreis vom Radius r_C um das betrachtete Teilchen eindringen können. Ansonsten müßten diese mitberücksichtigt werden, was wiederum die Komplexität erhöht. Der Nachbarschaftsradius sorgt somit zwar auf der einen Seite für einen etwas höheren Rechenaufwand, stellt aber auch sicher, daß innerhalb von ν Zeitschritten keine Moleküle fälschlicherweise außer acht gelassen werden. Die Variablen r_N und ν sind somit voneinander abhängig und von der atomistischen Mobilität, sprich Diffusion. In der Literatur wird oft $\nu \in \{10, ..., 20\}$ empfohlen.

2. Bei größeren Systemen und kleinem r_C wird die auf Hockney u. Eastwood (1981) zurückgehende *Zellenindexmethode* verwendet, bei der das System in ein reguläres Gitter mit M^3 Zellen der Länge $\frac{L}{M} > r_C$ eingeteilt wird und als nächste Nachbarn diejenigen Teilchen definiert werden, die sich in derselben oder einer angrenzenden Zelle befinden. In einem zweidimensionalen homogenen System enthält jede Zelle dann etwa $N_C = \frac{N}{M^2}$ Moleküle, also insgesamt $9NN_C/2 = 4.5NN_C$ Interaktionen zwischen nächsten Nachbarn. Am Rand des Systems werden periodische Zellen betrachtet. Analog gibt es im dreidimensionalen Fall $27NN_C/2 = 13.5NN_C$ Interaktionen. Der Aufwand, sämtliche nächste Nachbarn zu bestimmen, beträgt somit $\mathcal{O}(N)$. Zur Komplexitätsreduktion bedient man sich sogenannter *Zellketten*, welche Ketten von Molekülen innerhalb einer Zelle darstellen, sowie zweier Listen: Die erste Liste enthält alle Kettenanfangsmoleküle und die zweite beschreibt die Kette selbst, besteht also aus allen weiteren Molekülen einer Kette. Durch die periodischen Randbedingungen wird die Komplexität ebenfalls reduziert. Es bleibt letztendlich nur die Entscheidung übrig, wie viele Moleküle in einer Zelle enthalten sind, was auf das Finden der Zelle für jedes Atom zurückzuführen ist und somit ein $\mathcal{O}(N)$-Problem ist.
   Alternativ kann man das Gitter so klein wählen, daß höchstens ein Molekül in einer Zelle vorhanden ist. Dann werden alle Zellen betrachtet, die höchstens r_C von einem Teilchen

entfernt sind. Dann ist es nur noch notwendig, zu entscheiden, ob eine Zelle besetzt ist oder nicht, was ebenfalls ein $\mathcal{O}(N)$-Problem ist.

Weitere Details zu den einzelnen Algorithmen sind in Allen u. Tildesley (1987) aufgeführt.

E.2 Methoden zur effizienten Kräfteberechnung

Intramolekulare Wechselwirkungen Für die Berechnung intramolekularer Kräfte wird zunächst das Bindungslängenpotential (2.14) betrachtet. Es sei r_{ij} der Verbindgungsvektor zwischen zwei Teilchen i und j, r_{ij}^0 der entprechende Verbindungsvektor im Gleichgewichtszustand, r_i der Ortsvektor des Teilchens i und r_j der Ortsvektor des Teilchens j. Es gilt dann: $r_{ij} = r_i - r_j$. Man ist nun an den Kräften F_i und F_j interessiert. Es gilt:

$$\begin{aligned} F_i &= -\frac{\partial U_b(||r_{ij}||)}{\partial ||r_{ij}||} \cdot \frac{\partial ||r_{ij}||}{\partial r_i} \\ &= -k_r \left(||r_{ij}|| - ||r_{ij}^0|| \right) \frac{r_{ij}}{||r_{ij}||}. \end{aligned} \tag{E.1}$$

Nach dem dritten Newtonschen Gesetz, dem sogenannten *Wechselwirkungsprinzip*, treten Kräfte immer paarweise auf, das heißt, wenn ein Körper A auf einen anderen Körper B eine Kraft ausübt, so wirkt eine gleichgroße, entgegengerichtete Kraft von B auf A. Somit gilt $F_j = -F_i$. Für das Bindugswinkelpotential (2.15) ergibt sich folgendes: Es sei ϕ der Bindungswinkel zwischen drei Teilchen i, j und k sowie ϕ^0 der entsprechende Bindungswinkel im Gleichgewichtszustand. Es gilt:

$$\cos\phi = \frac{\langle r_{ij}, r_{kj} \rangle}{||r_{ij}|| \cdot ||r_{kj}||}.$$

Somit folgt für die Kraft, die auf Teilchen i wirkt:

$$\begin{aligned} F_i &= -\frac{\partial U_a(\phi)}{\partial \phi} \cdot \frac{\partial \phi}{\partial \cos\phi} \cdot \frac{\partial \cos\phi}{\partial r_i} \\ &= -k_a \left(\phi - \phi^0 \right) \cdot \left(\frac{\partial \cos\phi}{\partial \phi} \right)^{-1} \cdot \frac{\partial}{\partial r_i} \frac{\langle r_{ij}, r_{kj} \rangle}{||r_{ij}|| \cdot ||r_{kj}||} \\ &= -k_a \frac{\phi - \phi^0}{\sin\phi} \cdot \frac{1}{||r_{kj}||} \left(\frac{r_{kj}}{||r_{ij}||} - \frac{\langle r_{ij}, r_{kj} \rangle}{||r_{ij}||^3} r_i \right). \end{aligned} \tag{E.2}$$

Aus geometrischen Überlegungen folgt $F_k = \exp(\imath\phi) F_i$ und $F_j = -F_i - F_k$.
Zur Berechnung des Diederwinkels ω zwischen vier Teilchen i, j, k und l wird das Vektorprodukt verwendet:

$$\cos\omega = \frac{\langle r_{jk} \times r_{ji}, r_{kj} \times r_{kl} \rangle}{||r_{jk} \times r_{ji}|| \cdot ||r_{kj} \times r_{kl}||}. \tag{E.3}$$

Die Berechnung der Kräfte erfolgt dann analog zum Binrdungswinkelpotential.

Intermolekulare Wechselwirkungen Die Berechnung kurzreichweitiger intermolekularer Interaktionen ist ein $\mathcal{O}(N^2)$-Problem und somit äußerst rechenaufwendig. Effizientere Berechnungsweisen werden für dispersive Interaktionen hier am Beispiel des LJ-Potentials beschrieben. Wie in Anhang E.1 diskutiert wird, bestimmt man einen Abschneideradius $r_C \in [2.5\sigma, 5\sigma]$

und berücksichtigt für jedes Teilchen lediglich seine nächsten Nachbarn, also alle Teilchen, die in einer Kugel vom Radius r_C um das betrachtete Teilchen liegen. Außerhalb dieser Kugel wird das Potential auf 0 gesetzt. Somit wird die Komplexität auf $\mathcal{O}(N)$ reduziert.
Die Kraft, welche durch die Interaktion zweier Teilchen i und j entsteht, ergibt sich nach Gleichung (2.21) gemäß:

$$F_{ij} = \frac{24\epsilon}{||r_{ij}||^2}\left[2\left(\frac{\sigma}{||r_{ij}||}\right)^{12} - \left(\frac{\sigma}{||r_{ij}||}\right)^{6}\right] r_{ij}. \tag{E.4}$$

Im folgenden sei der Einfachheit halber $r := ||r_{ij}||$.
Es ist zu beachten, daß sich durch das Abschneiden des Potentials im Falle $||r_{ij}|| = r_C$ eine Unstetigkeit ergibt. Sobald ein Molekülpaar diese Grenze überschreitet, führt dies dazu, daß die Gesamtenergie des Systems nicht erhalten bleibt, was ein Widerspruch zum Energieerhaltungssatz ist. Daher wird in vielen Fällen das Potential nach oben verschoben, so daß $U_{\mathrm{LJ}}(r_C) = 0$. Man verwendet dann das *verschobene Potential*

$$U^S_{\mathrm{LJ}}(r) := \begin{cases} U_{\mathrm{LJ}}(r) - U_{\mathrm{LJ}}(r_C), & r \le r_C \\ 0, & r > r_C \end{cases} \tag{E.5}$$

Das so verschobene LJ-Potential ist jedoch bei r_C immer noch nicht differenzierbar, was dazu führt, daß die entsprechende Kraft an dieser Stelle unstetig ist. Damit auch die Ableitung bei r_C verschwindet, verwendet man statt (E.5) das sogenannte *Shifted-Force-Potential*

$$U^{SF}_{\mathrm{LJ}}(r) := \begin{cases} U_{\mathrm{LJ}}(r) - U_{\mathrm{LJ}}(r_C) - \left(\frac{dU_{\mathrm{LJ}}(r)}{dr}\right)_{r=r_C}(r - r_C), & r \le r_C \\ 0, & r > r_C \end{cases} \tag{E.6}$$

Es ist jedoch zu beachten, daß die Einführung eines Abschneideradius Auswirkungen auf die zu berechnenden physikalischen Eigenschaften hat. Es werden daher langreichweitige Korrekturterme eingeführt, die nachträglich entweder auf das Potential oder auf die zu berechnenden Eigenschaften selbst addiert werden.
Auch bei langreichweitigen Interaktionen werden Abschneideradien verwendet. Beim Coulomb-Potential beispielsweise werden kubische Splines $S(r)$ ab einem bestimmten r_0 eingesetzt, welche das Potential stetig und glatt bei r_C auf 0 setzen. Ein Spline soll dabei die Eigenschaften $S(r_0) = 1$, $S(r_C) = 0$, $S'(r_0) = 1$ und $S'(r_C) = 0$ haben. Es gilt dann:

$$U_{\mathrm{El}}(r) \propto \begin{cases} \frac{1}{r}, & r < r_0 \\ \frac{1}{r}S(r), & r_0 < r < r_C \\ 0, & r > r_C. \end{cases} \tag{E.7}$$

Für $r_0 < r < r_C$ gilt dann für die resultierende Kraft:

$$F(r) = \frac{1}{r^2}S(r) - \frac{1}{r}S'(r). \tag{E.8}$$

Eine andere Möglichkeit besteht in der Verwendung der *Heavyside-Funktion*

$$\Theta(x) = \begin{cases} 1, & x \ge 0 \\ 0, & x < 0. \end{cases} \tag{E.9}$$

Für das Coulomb-Potential ergibt sich dann:

$$U_{\mathrm{El}}(r) = \left(\frac{q_i q_j}{4\pi\epsilon_0}\right)\frac{1}{r}\Theta(r_C - r). \tag{E.10}$$

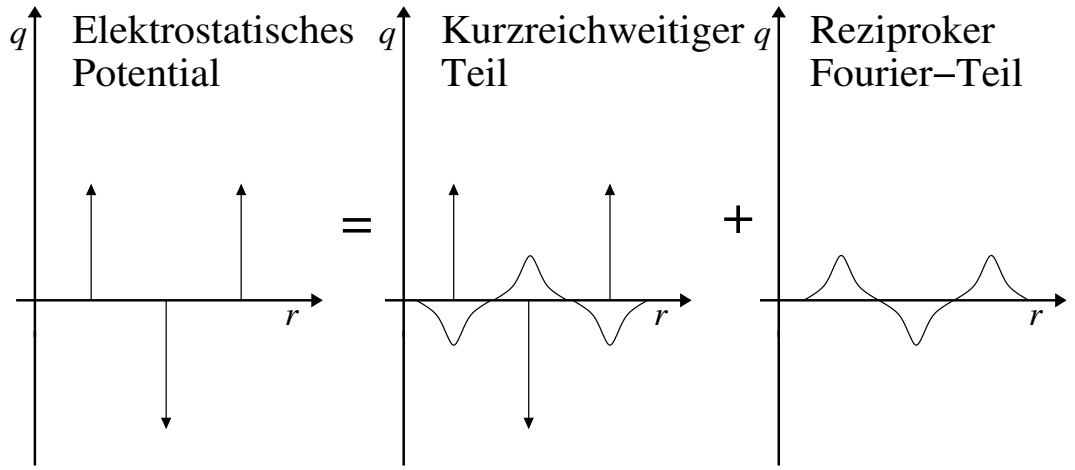


Abbildung E.1: Veranschaulichung der Ewald-Summation: Zu einem kurzreichweitigen Term, der die Umgebung der Punktladungen $q(r)$ durch eine Ladungsverteilung gleichen Ausmaßes und mit entgegengesetztem Vorzeichen beschreibt, wird ein langreichweitiger reziproker Fourier-Term addiert. Der Rechenaufwand sinkt dadurch von $\mathcal{O}(N^2)$ auf $\mathcal{O}(N^{\frac{3}{2}})$.

Die resultierende Kraft läßt sich dann gemäß

$$F(r) = \left(\frac{q_i q_j}{4\pi\epsilon_0}\right)\left[\frac{1}{r^2}\Theta(r_C - r) - \frac{1}{r}\delta(r_C - r)\right] \tag{E.11}$$

berechnen. Der Nachteil besteht darin, daß die mathematischen Unstetigkeiten zu Fluktuationen im System führen.
Eine sehr populäre Methode zur effizienten Berechnung langreichweitiger Kräfte ist die sogenannte *Ewald-Summation*, die auf Ewald (1921) zurückgeht und einen Rechenaufwand von $\mathcal{O}(N^{\frac{3}{2}})$ hat. Eine Grundvoraussetzung hierfür sind periodische Randbedingungen. Es werden dabei Vektoren $n = (n_x L, n_y L, n_z L)^T$ auf einem kubischen Gitter der Größe L mit $n_x, n_y, n_z \in \mathbb{Z}$ betrachtet. Dann läßt sich das elektrostatische Potential schreiben als Summe über alle kubischen Gitterpunkte:

$$U_{\mathrm{El}}(R) = \frac{1}{2}\sum_n{}' \left(\sum_{i,j=1}^{N} \frac{q_i q_j}{4\pi\epsilon_0}\frac{1}{||r_{ij} + n||}\right), \tag{E.12}$$

wobei R der Gesamtvektor aller Teilchenpositionen im Raum darstellt. $\sum'$ bedeutet: $i \neq j$, $n = 0$. Die Summation geschieht in der folgenden Reihenfolge: $||n|| = 0$ $(n = (0,0,0)^T)$, $||n|| = L$ $(n = (\pm L, 0, 0)^T, (0, \pm L, 0)^T, (0, 0, \pm L)^T)$ usw. Insgesamt baut sich das System hierarchisch aus annähernd sphärischen Elementen auf. Es ist zu beachten, daß sich die potentielle Energie in einem guten Leiter wie zum Beispiel einem Metall ($\epsilon_{\mathrm{RF}} = \infty$) und im Vakuum ($\epsilon_{\mathrm{RF}} = 1$) folgendermaßen unterscheiden:

$$U_{\mathrm{El}}(R, \epsilon_{\mathrm{RF}} = \infty) = U_{\mathrm{El}}(R, \epsilon_{\mathrm{RF}} = 1) - \frac{1}{6\epsilon_0 L^3}\left\|\sum_{i=1}^{N} q_i r_i\right\|^2. \tag{E.13}$$

Die Idee der Ewald-Summation besteht nun darin, daß jede Punktladung von einer Ladungsverteilung gleichen Ausmaßes und mit entgegengesetztem Vorzeichen umgeben wird. Die Ver-

teilungsdichte ρ^q ist üblicherweise Gaußsch:

$$\rho^q(r) = \frac{q_i \alpha^3}{\pi^{\frac{3}{2}}} \exp(-\alpha^2 r^2).$$

Dabei sind α die Breite der Verteilung und r die zum Verteilungszentrum relative Position. Diese Verteilung kommt einer ionischen Atmosphäre gleich und dient dazu, Interaktionen zwischen benachbarten Ladungen darzustellen. Diese Interaktionen sind kurzreichweitig. Insgesamt besteht die Ewald-Summe aus einem kurzreichweitigen Term und einem langreichweitigen Korrekturterm, der gemäß Abbildung E.1 hinzuaddiert wird und das gleiche Vorzeichen wie der Originalterm haben muß. Dieser langreichweitige Korrekturterm ist die Summe aller Interakionsenergien zwischen den Ladungen einer Zentralbox und sämtlichen Ladungen des Gitters. Er wird als Fourier-Transformierte der Ladungsverteilungen berechnet, da er periodisch ist. Der kurzreichweitige Teil wird in kartesischen Koordinaten berechnet und summiert die Integrale über die Ladungsverteilungsdichten innerhalb der einzelnen Gitterzellen. Die Effizienz der Ewald-Summe liegt darin, daß der langreichweitige Term einfach und schnell im Fourier-Raum berechnet und der Rest als wenig aufwendig zu berechnender kurzreichweitiger Term angegeben werden kann. Da die Fourier-Transformierte auch die Interaktion eines Ladungszentrums mit sich selbst berücksichtigt, muß dies durch einen entsprechenden Korrekturterm ausgeglichen werden. Mit $\mathrm{erfc}(x) := (2/\sqrt{\pi}) \int_x^\infty \exp(-t^2) dt$ und Gleichung (E.13) ergibt sich die Ewald-Summe zu:

$$\begin{aligned} 4\pi\epsilon_0 U_{\mathrm{El}}(R, \epsilon_{\mathrm{RF}} = 1) \;=\; & \underbrace{\frac{1}{2} \sum_{i,j=1}^{N} {\Big[\sum_{n}}' q_i q_j \frac{\mathrm{erfc}(\alpha ||r_{ij} + n||)}{||r_{ij} + n||}}_{\text{kurzreichweitiger Teil}} \\ & + \underbrace{\frac{1}{\pi L^3} \sum_{k \neq 0} q_i q_j \frac{4\pi^2}{||k||^2} \exp(-\frac{||k||^2}{4\alpha^2}) \cos\langle k, r_{ij} \rangle \Big]}_{\text{langreichweitiger Fourier-Teil}} \\ & - \underbrace{\frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^{N} q_i^2}_{\text{Korrekturterm}} + \underbrace{\frac{2\pi}{3L^3} \left\| \sum_{i=1}^{N} q_i r_i \right\|^2}_{\text{(E.13)}}, \end{aligned} \tag{E.14}$$

mit $k = 2\pi n / L^2$.
Wegen $\lim_{x \to \infty} \mathrm{erfc}(x) = 0$, leistet, falls α groß genug ist, nur der Summand mit $n = 0$ zum kurzreichweitigen Term einen positiven Beitrag. Daher bezieht sich dieser Term im Endeffekt nur auf eine Gitterzelle. Andererseits bedeutet ein großes α eine schmale Ladungsverteilung, so daß im Fourier-Raum viele Terme mit in die Summation eingehen. Die akkurate Wahl von α entscheidet somit über die Effizienz der Ewald-Summe. Auch die Kraft als negativer Gradient läßt sich mithilfe der Ewald-Summe effizient berechnen.

Eine weitere Methode zur Behandlung von langreichweitigen Kräften ist die *Particle-Particle and Particle-Mesh-* beziehungsweise *P^3M-Methode* für ionische Systeme, welche von Eastwood u.a. (1980) entwickelt wurde. Auch diese Methode teilt das elektrostatische Potential in einen kurzreichweitigen und einen langreichweitigen Teil ein. Der kurzreichweitige Teil wird dabei wie

das LJ-Potential behandelt (*Particle-Particle-Methode*). Die Behandlung des langreichweitigen Teils erfolgt nach der *Particle-Mesh-Methode*:

1. Die Ladungsdichte wird auf einem feinen Gitter in der Simulationsbox approximiert.
2. Auf diesem Gitter wird die Poisson-Gleichung mithilfe der Fast-Fourier-Methode gelöst. Als Lösung erhält man das elektrostatische Potential auf jedem Gitterpunkt.
3. Die Kraft wird durch finite Differenzen aus dem Potential auf dem Gitter bestimmt. Die Kraft, die auf ein bestimmtes Teilchen wirkt, wird dann durch Interpolation berechnet.

Der Vorteil der P^3M-Methode gegenüber der Ewald-Summation liegt im kleineren Rechenaufwand: Die Berechnung der Ewald-Summe ist asymptotisch ein $\mathcal{O}(N^{\frac{3}{2}})$-Problem, während die P^3M-Methode in $\mathcal{O}(N \log N)$ Rechenzeit erfolgt.

F Simulationsprogramme

F.1 Gromacs

Die Software **GRO***ningen* **MA***chine for* **C***hemical* **S***imulations (Gromacs)* ist ein freies, unter der *GNU Public License* zur Verfügung stehendes Softwarepaket zur Durchführung von MD-Simulationen. Es löst die Newtonsche Bewegungsgleichung (2.11) für Systeme von Millionen von Teilchen. Zunächst wurde *Gromacs* für biochemische Moleküle kreiert (siehe zum Beispiel Berendsen u. van Gunsteren (1987) und van Gunsteren u. a. (1996)), und es ist äußerst schnell bei der Berechnung nichtbindender Interaktionen. Die in dieser Arbeit verwendeten Versionen reichen von 3.3 über 4.0 bis 4.5.4. Die neueste Version 4.5.4 (van der Spoel u. a., 2010) beinhaltet sämtliche Standardverfahren zur Realisierung von MD-Simulationen (siehe Abschnitt 2.3), basiert auf dem Gromos-Kraftfeld aus Abschnitt 2.2.4 und hat insbesondere die folgenden hervorzuhebenden Eigenschaften:

- *Gromacs* ist in der Programmiersprache *C++* geschrieben, was es zu einer extrem hochleistungsfähigen Software macht.
- Es ist benutzerfreundlich in dem Sinne, daß Topologien und sonstige Eingabedateien in menschenlesbarem Textformat erwartet werden. Einige Beispieldateien sind unten angegeben.
- Es verfügt über eine aufschlußreiche Dokumentation des aktuellen Simulationslaufs, inklusive Datum und Uhrzeit einzelner Teilschritte.
- Es existiert eine große Anzahl an Analysetools zur Berechnung physikalischer Eigenschaften aus den berechneten Trajektorien.
- Es ist parallelisierbar und benutzt eine Standard-MPI-Kommunikation (siehe Anhang G.8).

Im folgenden werden für den Beispielfall Methanol (siehe Abschnitt 5.1.4) eine Topologiedatei (`.top`) und eine Datei, die sämtliche notwendigen Eingaben für den Verlauf einer Simulation festlegt (`.mdp`), angegeben:

Beispiel einer `.top`-Datei:

```
[ defaults ]     ; nbfunc = 1: LJ-Parameter (sigma/epsilon)
                 ; comb-rule = 2: Lorentz-Berthelot-Kombination
```

```
                ; gen-pairs = yes: automatische Berechnung der Wechselwirkungen
                ; fudgeLJ = 1.0: Faktor fuer LJ-Potentialterm
                ;                (nur 1,4-Wechselwirkungen)
                ; fudgeQQ = 1.0: Faktor fuer Couloumb-Potentialterm
; nbfunc   comb-rule   gen-pairs  fudgeLJ  fudgeQQ
    1       2          yes        1.0     1.0

[ atomtypes ]   ; Atomtypen mit Masse, Ladung, Typ (A: Atom), sigma und epsilon
;name   mass        charge   ptype   sigma   epsilon
 ho     1.00800     0.3469   A       0.0     0.0
 oh     15.99900    -0.5590  A       0.30200 0.37500
 ch3    15.03400     0.2121  A       0.77325 0.81482

[ moleculetype ] ; Name des Molekuels, Ausschaltung der 1,3-Wechselwirkungen
                 ; und weniger
; Name            nrexcl
  meoh           3

[ atoms ]        ; Atome mit Nummer, Typ, Molekuelzugehoerigkeit, Ladungsnummer,
                 ; Ladung und Masse
;   nr   type  resnr residue    atom  cgnr   charge      mass
    1    ch3    1    meoh       ch3   1      0.2121      15.0340
    2    oh     1    meoh       oh    2     -0.5590      15.9990
    3    ho     1    meoh       ho    3      0.3469      1.0080

[ constraints ]  ; Nebenbedingungen fuer Bindungen mit Angabe der Bindungslaengen
;  ai    aj
   1     2     1    0.143000E+00
   2     3     1    0.945000E-01
   1     3     1    0.193226E+00

[ exclusions ]   ; exkludierte Wechselwirkungen
;  ai    aj
   1     2
   1     3
   2     3

[ system ]       ; Name des Systems
; Name
  MeOH

[ molecules ]    ; Anzahl Molekuele
; Compound       #mols
  meoh           1000
```

Beispiel einer `.mdp`-Datei:

```
title               =  Yo           ; Name der Simulation
cpp                 =  /lib/cpp     ; C++-Library-Pfad
dt                  =  0.002        ; Zeitschrittweite in ps
tinit               =  0.0          ; Startzeit
nsteps              =  250000       ; Anzahl Zeitschritte
nstcomm             =  1            ; Zuruecksetzen der Massenzentrenbewegung
                                    ; zu jedem Zeitschritt
nstxout             =  0            ; Herausschreiben der Koordinaten zu
                                    ; keinem Zeitschritt
nstvout             =  0            ; Herausschreiben der Geschwindigkeiten zu
                                    ; keinem Zeitschritt
nstxtcout           =  0            ; Herausschreiben der Koordinaten in
                                    ; Trajektorie zu keinem Zeitschritt
nstlog              =  1000         ; Aktualisierung der .log-Datei alle
                                    ; 1000 Zeitschritte
nstenergy           =  1000         ; Herausschreiben der physikalischen
                                    ; Eigenschaften alle 1000 Zeitschritte
nstlist             =  6            ; Aktualisierung der Nachbarschaftsliste
                                    ; alle 6 Zeitschritte
ns_type             =  grid         ; Erstellung der Nachbarschaftsliste mit
                                    ; Zellenindexmethode
coulombtype         =  pme          ; Particle-Mesh-Ewald
fourierspacing      =  0.12         ; Maschenweite des Gitters im
                                    ; Fourierraum fuer Ladungsdichte
pmeorder            =  4            ; Ordnung des PME-Algorithmus
optimize_fft        =  yes          ; Fast-Fourier-Transformation
ewald_rtol          =  1.0e-5       ; Relative Staerke der elektrostatischen
                                    ; Interaktion am Cutoff
DispCorr            =  EnerPres     ; Langreichweitige Korrekturterme fuer
                                    ; Energie und Druck
constraint_algorithm=  lincs        ; MD mit Nebenbedingungen (LINCS-Algorithmus)
lincs_order         =  4            ; Ordnung des LINCS-Algorithmus
lincs_iter          =  3            ; Anzahl an LINCS-Iterationen
lincs-warnangle     = 30            ; Ausgabe einer Warnung bei Rotation einer
                                    ; Bindung um mehr als 30 Grad
rlist               =  0.9          ; Cutoff fuer Nachbarschaftsliste
rvdw                =  0.9          ; Cutoff fuer van-der-Waals-Interaktionen (LJ)
rcoulomb            =  0.9          ; Cutoff fuer Coulomb-Interaktionen
Tcoupl              =  nose-hoover  ; Nose-Hoover-Thermostat
tc-grps             =  system       ; Thermostatkopplung an das gesamte System
ref_t               =  303          ; Temperatur
tau_t               =  0.5          ; Kopplungsparameter fuer Thermostat
```

```
Pcoupl              =  parrinello-rahman   ; Parrinello-Rahman-Barostat
Pcoupltype          =  isotropic     ; Isotropische Druckgeometrie
tau_p               =  2.0           ; Kopplungszeit fuer Barostat
compressibility     =  33.0e-6       ; Kompressibilitaet
ref_p               =  1.0           ; Druck
gen_vel             =  no            ; Keine automatische Erzeugung initialer
                                     ; Geschwindigkeiten bei Thermostat
```

Weiterhin sind Koordinaten und Geschwindigkeiten in einer separaten Datei festzulegen, entweder in dem von vielen Simulationsprogrammen akzeptierten `.pdb`- oder im *Gromacs*-spezifischen `.gro`-Format. Die Erstellung der Simulationsbox, in der sich sämtliche Teilchen befinden, geschieht über das Präparationstool `grompp`, die eigentliche Erstellung der Trajektorie geschieht mittels des Haupttools `mdrun`, und die Berechnung von Systemeigenschaften erfolgt durch ein Tool namens `g_energy`, welches die Eigenschaften zu jedem gewünschten Zeitschritt in eine Datei im Listenformat speichert.

F.2 Moscito

Moscito ist wie *Gromacs* ein freies, unter der *GNU Public License* zur Verfügung stehendes Softwarepaket zur Durchführung von MD-Simulationen. In dieser Arbeit wurde lediglich Version 4 (siehe Paschek u. Geiger (2003)) verwendet. *Moscito* ist dazu in der Lage, Flüssigkeiten und Gase zu simulieren und beinhaltet Standard-Kraftfelder wie *Amber* (Wang u. a., 2004), OPLS (Jorgensen u. a., 1996), CHARMM (Brooks u. a., 1983) und Gromos (Berendsen u. van Gunsteren, 1987). Die Software zeichnet sich durch eine relativ hohe Geschwindigkeit auf Intel-Architekturen aus, da die wichtigsten Teile des Quellcodes in Assembler geschrieben sind. Weiterhin ist *Moscito* wie *Gromacs* durch eine Standard-MPI-Kommunikation (siehe Anhang G.8) parallelisierbar.
In dieser Arbeit wird *Moscito* lediglich zur Berechnung bestimmter physikalischer Zielgrößen aus einer bereits bestehenden, durch *Gromacs* erstellten, Trajektorie im `.xtc`-Format verwendet. Dabei werden neben dem Haupttool `msdmol` verschiedene in *Moscito* implementierte Tools verwendet: `trajenergy` für das intermolekulare Potential, welches zur Berechnung der Verdampfungsenthalpie notwendig ist, `fitpoly` für den Diffusionskoeffizienten, `vectorcor` für die Reorientierungszeit und `viscosity_`*ab* für die Scherviskosität in Richtung *ab* ($a, b \in \{x, y, z\}$). Als Eingabe für *Moscito* sind neben der Trajektorie eine `.sys`-Datei und eine `.str`-Datei nötig. Erstere legt analog zur `.top`-Datei in *Gromacs* (siehe Anhang F.1) die Topologie fest und erhält sämtliche Informationen über das Kraftfeld. Letztere ist eine Koordinatendatei, analog zur `.pdb`-Datei beziehungsweise `.gro`-Datei in *Gromacs*. Die Ausgabe von `trajenergy` besteht aus einer Datei, die zu jedem Zeitschritt sämtliche Terme der potentiellen Energie enthält. Aufsummieren von Lennard-Jones- und Coulomb-Potential ergibt das intermolekulare Potential. Alle anderen Größen werden mithilfe der *Moscito*-Funktion `average` als Mittelwerte über alle betrachteten Zeitschritte in einzelne Dateien geschrieben.

F.3 ms2

Das Simulationspaket *ms2 (Version 1.0)*, siehe Deublein u. a. (2011), wurde für die Berechnung thermodynamischer Eigenschaften von Flüssigkeiten, bestehend aus elektroneutralen Teilchen, im Gleichgewichtszustand entwickelt. Es ist dazu in der Lage, sowohl MD- als auch MC-Simulationen durchzuführen, und zeichnet sich dadurch aus, daß sowohl VLE-Daten als auch Transporteigenschaften für Reinstoffe und Gemische berechnet werden können (siehe auch Guevara-Carrion u. a. (2012)). Für die Berechnung von VLE-Daten wird die sogenannte *Grand-Canonical-Monte-Carlo-(GEMC-)*Methode verwendet (siehe Anhang A.5). Zur Berechnung des chemischen Potentials stehen die Methode der graduellen Einsetzung und die Methode nach Widom (vergleiche Abschnitt 2.5.2) zur Verfügung. Transporteigenschaften wie Diffusionskoeffizienten, Viskositäten und Wärmeleitfähigkeit werden mithilfe der Green-Kubo-Darstellung (Gleichung (C.17)) ermittelt. Eine Vielzahl an Anwendungen des Softwarepakets wurde bereits veröffentlicht, zum Beispiel VLE-Simulationen für kleine Moleküle wie Benzol und Phosgen in Huang u. a. (2011) und Berechnung von Transporteigenschaften in Guevara-Carrion u. a. (2012). Kraftfeldparameter wurden dabei stets mithilfe des Verfahrens nach Stoll aus Abschnitt 3.2.3 optimiert.
ms2 ist in *Fortran90* geschrieben und für eine schnelle Ausführung auf verschiedenen Rechenarchitekturen optimiert. Parallelisierungen werden wie bei *Gromacs* (siehe Anhang F.1) mithilfe von MPI (siehe Anhang G.8) realisiert. Weiterhin existieren drei Hilfstools zur Erstellung von Eingabedateien über ein GUI (`ms2par`), zur Visualisierung der Entwicklung physikalischer Eigenschaften im Laufe einer Simulation `ms2chart` sowie zur Visualisierung von Simulationsboxen (`ms2molecules`).

Einfache Ein- und Ausgabedateien machen *ms2* äußerst benutzerfreundlich: Die Eingabe besteht aus lediglich zwei Dateien, einer `.par`-Datei und einer `.pm`-Datei. Erstere definiert sämtliche für die Simulation relevante Eingaben, ähnlich einer `.mdp`-Datei bei *Gromacs*, und letztere legt die Topologie fest, ähnlich einer `.top`-Datei bei *Gromacs*. Die Bestimmung der Koordinaten sämtlicher Atome innerhalb einer Simulationsbox erfolgt innerhalb von *ms2* automatisch. Im folgenden werden für den Beispielfall Epoxid (siehe Abschnitt 5.3) eine Topologiedatei (`.pm`) und eine Datei, die sämtliche notwendigen Eingaben für den Verlauf einer Simulation festlegt (`.par`), angegeben:

Beispiel einer `.pm`-Datei:

```
NSiteTypes =   2            # Anzahl an Zentren

SiteType   =   LJ126        # LJ-Zentrum
NSites     =   3            # Anzahl an LJ-Zentren
                            # 2xCH2, O

# CH2
x          =     0.7800000
```

```
y          =     0.000000
z          =    -0.48431
sigma      =     3.71300
epsilon    =     89.62986
mass       =    14.0
# CH2
x          =    -0.780000
y          =     0.000000
z          =    -0.48431
sigma      =     3.71300
epsilon    =     89.62986
mass       =    14.0
# O
x          =     0.000000
y          =     0.000000
z          =     0.735469
sigma      =     2.70600
epsilon    =     85.93032
mass       =     16.0

SiteType   =   Charge        # Punktladungen
NSites     =   3             # Anzahl an Punktladungen

# CH2
x          =     0.000000
y          =     0.000000
z          =     1.466000
charge     =     0.1741
mass       =     0.0
shielding  =     0.0         # Abschirmung:
                             # ab dieser Distanz unendliches
                             # elektrostatisches Potential
# CH2
x          =     0.000000
y          =     0.000000
z          =     0.000000
charge     =     0.1741
mass       =     0.0
shielding  =     0.0
# O
x          =     0.733000
y          =     1.229000
z          =     0.000000
charge     =    -0.3482
mass       =     0.0
shielding  =     0.0
```

```
NRotAxes = auto              # Setzen des Koordinaten-
                             # ursprungs auf Massenzentrum

NFluct     =    5            # Graduelle Einsetzung:
1.000  1.000  0.800  0.750   # 5 Zustaende mit Skalierungsfaktoren
1.000  0.800  0.600  0.500   # fuer 4 versch. Parameter
0.900  0.600  0.400  0.250
0.600  0.400  0.200  0.000
0.000  0.200  0.000  0.000
```

Beispiel einer .par-Datei:

```
Units         =      SI      # Zielgroessen in SI-Einheiten
LengthUnit    =       3.5    # Referenzwert fuer sigma
EnergyUnit    =     100.0    # Referenzwert fuer epsilon
MassUnit      =      50.0    # Referenzwert fuer Masse
Simulation    =      MC      # Simulationsmethode (MD/MC)
Acceptance = 0.5             # Akzeptanzrate (Versuchsbewegung)
Ensemble      =     NPT      # Ensemble
MCORSteps     =       0      # MC-Schritte zum Verlassen der
                             # initialen Box
NVTSteps      =    5000      # NVT-Schritte
NPTSteps      =   20000      # NPT-Schritte
RunSteps      =  200000      # Produktionsschritte
ResultFreq    =     100      # Aufdatierung der .run-Datei alle
                             # 100 Zeitschritte
ErrorsFreq    =    5000      # Aufdatierung der .res-Datei alle
                             # 5000 Zeitschritte
VisualFreq    =       0      # Aufdatierung der Trajektorien
CutoffMode    =     COM      # Abschneideradien anhand von
                             # Massenzentren
NEnsembles    =       1      # Anzahl an Ensembles

Temperature   = 230.0        # Temperatur
Pressure      = 0.005        # Druck
Density       = 21.0         # initiale Fluessigdichte
PistonMass    = 0.0002       # Masse der Zusatzkoordinate
                             # im Andersen-Barostat
NParticles    = 500          # Anzahl an Teilchen
NComponents   = 1            # Anzahl an Systemkomponenten
                             # (hier Reinstoff)
PotModel      =   pm_file.pm # Name der .pm-Datei
MolarFract    =       1.0    # Molbruch der Systemkomponente
ChemPotMethod =   GradIns    # Berechnung des chem. Potentials
```

```
                              # (Widom/Graduelle Einsetzung)
WeightFactors = Guess         # Benutzerdefinierte Gewichte fuer
1.00                          # graduelle Einsetzung,
8                             # werden waehrend der Simulation
76                            # veraendert
407
1393
3066
4443
4507
3585

Cutoff        =       3.5     # Abschneideradius, wird mit
                              # Referenzwert fuer sigma
                              # mulitpliziert
Epsilon       =       1.0E10 # dielektrische Konstante
```

ms2 verfügt über ein ausführbares Tool, welches alle Präparations-, Simulationsroutinen und Berechnungsroutinen steuert. Im seriellen Fall ist dies `ms2.linux` und im parallelen Fall `ms2mpi`. Zur Berechnung von Transporteigenschaften ist `ms2trans` zu verwenden.
Die Ausgabe der Systemeigenschaften erfolgt einerseits wie bei *Gromacs* im Listenformat (`.run`-Dateien), andererseits werden auch sämtliche thermodynamischen Durchschnitte zusammen mit ihren statistischen Fehlern in eine benutzerlesbare Datei geschrieben. Ein Beispiel einer derartigen `.res`-Datei ist im folgenden angegeben:

Beispiel einer `.res`-Datei:

```
================================================================================

                        SIMULATION RESULT FILE
                        ----------------------

Simulation type                   :         Monte-Carlo
Ensemble type                     :         NPT

Number of NVT equilibration steps  :      5000
Number of NPT equilibration steps  :     20000
Number of production steps         :    200000

Acceptance rate                 :         0.500000000

Number of particles               :       500
```

```
Mole fraction of current.pm        :         1.000000000
Chemical potential calculated by gradual insertion

Initial pressure            reduced:         0.000155270
                             in MPa:         0.005000000

Initial density             reduced:         0.542218160
                           in mol/l:        21.000000000

Initial temperature         reduced:         2.300000000
                               in K:       230.000000000

Unit of length                     :         3.500000000 A
Unit of energy                     :       100.000000000 K
Unit of mass                       :        50.000000000 a.u.

Lennard-Jones cutoff radius        :         3.500000000 sigma


================================================================================

VALUE                         UNITS          AVERAGE             ERROR
-----                         -----          -------             -----

Pressure                    reduced:         0.294038354         0.005362129
                             in MPa:         9.468604230         0.172670920

Density                     reduced:         0.550545746         0.000102561
                           in mol/l:        21.322525736         0.003972161

Temperature                 reduced:         2.300000000         0.000000000
                               in K:       230.000000000         0.000000000

Potential energy            reduced:       -31.683852074         0.005941638
                           in J/mol:    -26343.575643361         4.940182233

Enthalpy                    reduced:       -33.983570029         0.005941688
                           in J/mol:    -28255.678810485         4.940223253

Chemical potential:        -9.445959062         0.019350767

Partial molar volume:       reduced:         -0.957054341         0.098998423
                           in l/mol:         -0.024711059         0.002556131

Isothermal compressibility  reduced:         0.021954733         0.000492548
                           in 1/MPa:         0.000681783         0.000015296
```

```
dH/dP                        reduced:          1.268743456          0.014587932
                            in l/mol:          0.032758845          0.000376659

Isobaric heat capacity       reduced:          4.679412276          0.116212427
                       in J/(mol K):         38.907027771          0.966249573

Volume expansivity           reduced:          0.131107800          0.003488855
                              in 1/K:          0.001311078          0.000034889

Speed of sound               reduced:         12.282957307          0.090438412
                              in m/s:       1583.931196698         11.662356165

================================================================================

Cutoff radius is         0 times (  0.00%) too large

================================================================================

Statistics
----------

Acceptance rate volume changes in %:         50.002000000
Maximum displacement volume     r'd:          0.006887746

Component pm_file.pm
Acceptance rate trans.          in %:         45.885886981
                rotates         in %:         52.875026813
                biased trans.   in %:         43.972297813
                biased rotates  in %:         53.017059736
Maximum displacement trans.     r'd:          0.007547072
                     rotational r'd:          0.139298130

Acceptance rate gradual insertion change fluctuating particle moves:
  up        down (%)
  --------  --------
    0.0000    4.8893
    9.8571    7.8646
    7.8152    9.0372
    8.5396    9.6779
    9.2452   90.4434
   45.2579    0.0000
```

G Verwendete Software und Programmiersprachen

G.1 GROW – ein automatisierter Optimierungsworkflow

Die in den Abschnitten 3.1–3.3 vorgestellten theoretischen Überlegungen zum Erhalt optimaler Kraftfeldparameter wurden im Rahmen dieser Arbeit mithilfe eines automatisierten Optimierungsablaufs praktisch umgesetzt: In diesem Abschnitt werden das Konzept, Implementationsdetails und Dateiformate des am Fraunhofer-Institut SCAI hauptsächlich durch den Autor entwickelten Optimierungstools **GR***adient-based* **O***ptimization* **W***orkflow for the Automated Development of Molecular Models (GROW)* dargestellt. GROW (Version 1.0) wurde in Hülsmann u. a. (2010a) veröffentlicht. Die Software ermöglicht es dem Benutzer, Kraftfeldparameter mithilfe der in Abschnitt 3.4 eingeführten gradientenbasierten numerischen Optimierungsalgorithmen automatisch durch einen iterativen Prozeß zu optimieren. GROW ist modular konzipiert und besteht aus einem Hauptkontrollskript, sekundären Kontrollskripten für jeden einzelnen Algorithmus sowie Analyseskripts. Ein weiteres wichtiges Merkmal ist die generische Implementierung: GROW ist auf einfachste Art und Weise durch Entwickler erweiterbar, so daß ohne großen Aufwand weitere Algorithmen und Simulationstools angeschlossen werden können.

In den letzten Jahren ist bereits vereinzelte automatische Parameterisierungssoftware für Kraftfelder entwickelt worden, um spezielle Optimierungsaufgaben auf verschiedenen Granularitätsebenen zu lösen: Auf der quantenmechanischen Ebene beispielsweise wurde von Wang u. Kollman (2001) und Wang u. a. (2004) ein Tool namens *Parmscan* veröffentlicht, welches nur intramolekulare Kraftfeldparameter optimiert.
Für vergröberte Modelle (sogenanntes *Coarse Graining*) wurde von Reith u. a. (2002) ein Softwarepaket für ein hochwertiges automatisiertes Kraftfelddesign entwickelt: Das Tool *CG-OPT* bildet ein volldetailliertes atomistisches Modell eines Polymersystems auf ein vergröbertes mesoskaliges Modell ab und parametrisiert das damit verbundene Kraftfeld automatisch mithilfe des Simplex-Algorithmus aus Abschnitt 3.2.
GROW hat nicht nur den Vorteil, daß die Implementation schneller konvergentere und effizientere numerische Optimierungsverfahren enthält, sondern auch, daß es auf eine Vielzahl an Optimierungsproblemen anwendbar ist. Es ist sowohl auf über- als auch auf unterbestimmte Probleme anwendbar und nicht von der Art und Anzahl der zu optimierenden Kraftfeldparameter und Zielgrößen abhängig. Die Software betrachtet zunächst nur atomistische Modelle, eine Erweiterung auf Mesoskalen ist jedoch ebenfalls denkbar. Durch die generische Implementierung ist GROW ebenfalls nicht auf ein bestimmtes Simulationstool fixiert.

GROW beinhaltet verschiedene Programme und Skripte, um die Verwendung gradientenba-

sierter Verfahren zu erleichtern. Die Hauptbestandteile der Software sind:

- Gradientenbasierte Optimierungsverfahren,
- Simulationsprogramme,
- Analyseskripte,
- Bearbeitungsroutinen für Ein- und Ausgabe und
- Allgemeine Berechnungsroutinen.

Die meisten Skripte sind in *python* (Version 2.4.3, siehe Anhang G.9) und einige Hilfsskripte in *S+* (Version 2.7.1), einer zum Statistikpaket *R* (siehe Anhang G.10) zugehörigen Skriptsprache, geschrieben. Parallelrechnungen auf Clustern sowie die Steuerung der Simulationstools werden durch *shell*-Skripte (*bash* und *tcsh*) realisiert. Der Grund für diese Wahl besteht in der Tatsache, daß *python* objektorientiert ist und daß es auf einfachste Art und Weise als Schnittstelle zu anderen Tools verwendet werden kann, die durch *shell*-Kommandos gesteuert werden. Allgemeine Berechnungsroutinen wie die Lösung eines LGS oder Matrix-Vektor-Multiplikationen können sehr leicht mit eingebauten *R*-Bibliotheken umgesetzt werden. Daher wird hierzu die Skriptsprache *S+* verwendet.

Mithilfe von GROW wurde eine detaillierte Bewertung der eingesetzten gradientenbasierten Verfahren durchgeführt. Hierzu wurden Molekulare Simulationen durch die in Abschnitt 3.5.2 angesprochenen Korrelationsfunktionen ersetzt und künstliches Rauschen auf die berechneten physikalischen Eigenschaften addiert. Als Beispiel wurden Phasenübergangsdaten von Stickstoff (N_2) verwendet. Die Verfahrensbewertung ist in Hülsmann u. a. (2010b) veröffentlicht.

Nach der Darstellung in Anhang G.1.1, wie GROW zu konfigurieren ist, wird in Anhang G.1.2 der Programmaufbau im Detail vorgestellt. Die wichtigste Ausgabedatei, welche den gesamten Optimierungsablauf zusammenfaßt, wird in Anhang G.1.3 angegeben. Erweiterungsmöglichkeiten für Entwickler im Rahmen der generischen Implementation von GROW werden in Anhang G.1.4 erläutert.

G.1.1 Konfiguration

Die Konfigurationsdatei von GROW umfaßt sämtliche Eingabedateien, welche für den Optimierungsablauf notwendig sind, sowie alle zu definierenden Optionen bezüglich System, Simulation und Optimierung. Diese Dreiteilung ist im Aufbau der Datei wiederzufinden, und zwar durch die drei Sektionen 'SYS', 'SIM' und 'OPT': Die Sektion 'SYS' enthält sämtliche Spezifizierungen bezüglich des Systems, zum Beispiel generische Pfade und Einstellungen bezüglich der parallelen Umgebungen, also alle Optionen, die für die Verwendung eines Clusters notwendig sind. Die Sektion 'SIM' enthält sämtliche Eingabedateien für das Simulationstool, die initialen Kraftfeldparameter, experimentellen Zielgrößen und alle Variablen, die zur Definition des Ensembles ausschlaggebend sind (beispielsweise Temperatur, Druck, Anzahl an Systemteilchen und ähnliches). Die Sektion 'OPT' legt das zu verwendende gradientenbasierte Verfahren fest und definiert sämtliche für das jeweilige Verfahren einzustellenden Parameter. Weiterhin wird die Schrittweitensteuerung durch diese Sektion geregelt, und das zulässige Gebiet sowie das

Abbruchkriterium werden hier spezifiziert.
Eine typische Konfigurationsdatei hat die folgende Gestalt:

```
[SYS]
distribution: y                                 # Simulationen parallelisieren?
root_path: /home/mhuelsma/tags/MeOH             # Ausgabepfad fuer Simulation
batch: y                                        # batch-System via qsub?
qsub_path: /usr/local/bin/                      # qsub-Pfad
qsub_options: -pe mvapichl 4                    # qsub-Optionen
cluster_options: --machinefile $TMPDIR/machines # Weitere qsub-Optionen
qsub_queue: queue1                              # Name der Queue
proc_option: -np                                # Prozessoroption (Anzahl an Prozessoren)
nof_procs: 4                                    # Anzahl an Prozessoren
overwrite: n                                    # Alte Verzeichnisse loeschen, falls vorhanden
outpath: /scratch/mhuelsma/grow/output/         # Wurzelpfad fuer Ausgabe
mpi_run: /home/user/local/mvapich_1.2.7/bin/    # Pfad fuer ausfuehrbare MPI-Skripte
delete: n                                       # tmppath/outpath am Ende loeschen?

[SIM]
program: gromacs                                # Simulationstool
bin_path: /home/user/local/gromacs-4.0.2/bin/   # Pfad fuer ausfuehrbare Skripte des Simulationstools
moscito_bin_path: /home/user/moscito-4/bin/     # Pfad fuer ausfuehrbare Skripte des Analysetools
topology: ../input/topology.top                 # Topologiedatei Simulationstool
coordinates: /home/user/input/start.pdb         # Koordinatendatei Simulationstool
sim_in: /home/user/input/sim.mdp                # Eingabedatei Simulationstool
target: ../input/exp_properties.target          # Datei fuer experimentelle Zielgroessen
parameters: ../input/initial.para               # Datei fuer initiale Kraftfeldparameter
par_file_name: ../input/parameters.para         # Kopie der Kraftfeldparameterdatei (wird von GROW stets
                                                # aktualisiert)
substance: Benzol                               # Textfeld: Name des Ensembles
VLE: y                                          # VLE?
properties: diff sld                            # Zu optimierende Zielgroessen (Diffusion/Dichte)
s_rho: 0.005                                    # 0.5%-Toleranz fuer Dichte
s_nb: 0.01                                      # 1%-Toleranz fuer intermolekulare Wechselwirkungen
s_pot: 0.01                                     # 1%-Toleranz fuer potentielle Energie
fits: n                                         # Temperaturfits?
equilib: y                                      # Aequilibrierung?
nof_steps: 250000                               # Groesse des Zeitfensters fuer Aequilibrierung
time_window: 70                                 # verwende 70% des Zeitfensters fuer Mittelwerte
timestep: 0.002                                 # Zeitschrittweite in ps
xtcout: 1000                                    # Aktualisieren der Trajektorie alle 1000 Zeitschritte
preequi_steps_fac: 3                            # Praeaequilibrierung: 3x nof_steps
preequi_timestep_fac: 1.0                       # Zeitschrittweite fuer Praeaequilibrierung: 1x timestep
prod_steps_fac: 0.1                             # Produktionslauf: 0.1x nof_steps
temperature: 285 290 295 300 305 310 315        # Temperaturen
VLE_pressure: 0.08 0.11 0.14 0.19 0.24 0.31 0.4 # Druecke
mass: 32.041                                    # Molare Masse
molecules: 1000                                 # Anzahl Molekuele
emin_in: /home/user/input/e-min.mdp             # .mdp-Datei fuer Energieminimierung
prepreequi_in: /home/user/input/ppequi.mdp      # .mdp-Datei fuer Prae-Prae-Aequilibrierung
equi_in: /home/user/input/MeOH.mdp              # .mdp-Datei fuer Aequilibrierung
diff_structure: /home/user/input/MeOH.str       # .str-Datei fuer Moscito
diff_system: /home/user/input/MeOH.sys          # .sys-Datei fuer Moscito
moscito_par_file: /home/input/moscito.par       # Weitere Eingabedatei fuer Moscito
optoutpath: steepest_descent/armijo/final_out   # Ausgabepfad Optimierungsablauf
tmppath: steepest_descent/armijo/tmp            # Temporaeres Ausgabeverzeichnis fuer Zwischenablagen
outpath: steepest_descent/armijo/out            # Zusaetzlicher Ausgabepfad

[OPT]
method: conjugate_gradient                      # Numerisches Optimierungsverfahren
cg: fletcher_reeves                             # CG-Verfahren
gradient: numerical                             # Art der Gradientenberechnung
                                                # (Version 1.0: nur numerisch)
sl_method: armijo                               # Schrittweitensteuerung (heuristic/armijo)
```

```
max_armijo: 10                          # Max. Anzahl Armijo-Schritte
zeta: 0.5                               # Regularisierungsparameter (Armijo)
h: 0.01                                 # Diskretisierungsparameter fuer Gradient
limit: 0.0001                           # Abbruchkriterium
weights: 1.0 1.0                        # Gewichte fuer die Zielgroessen
weight_mode: compute                    # Summe Gewichte = 1
param_order: sigma epsilon              # Reihenfolge der Parameter in .para-Datei
param_number: 2 2                       # Anzahl sigmas und epsilons in .para-Datei
boundary: 30 40                         # Zul. Gebiet fuer sigma und epsilon
restart: 0                              # Neustart-Option
efficiency: n                           # Effiziente Gradientenberechnung?
```

In diesem Beispiel wird *Gromacs* (Version 4.0.2, siehe Abschnitt F.1) als Simulationstool und zur Berechnung des Diffusionskoeffizienten *Moscito* (Version 4) verwendet. Die Simulationen werden zu jeder Temperatur mithilfe eines *batch-Systems* parallelisiert. Das *shell*-Kommando hierzu ist `qsub`. Als Optimierungsverfahren wird das CG-Verfahren nach Fletcher und Reeves (siehe Abschnitt 3.4) benutzt.

G.1.2 Programmaufbau

Abbildung G.1 zeigt die Programmstruktur von GROW: Das Hauptkontrollskript `grow.py` bearbeitet sämtliche Eingaben und liest die Konfigurationsdatei ein. Danach werden alle Pfade, Dateien und für den Optimierungsablauf notwendigen Skripte auf Existenz und Konsistenz hin geprüft. Weiterhin werden alle nötigen Umgebungsvariablen gesetzt.
Das Kontrollskript des Optimierungsalgorithmus, das allgemein als `<Algorithmus>.py` bezeichnet wird und beispielsweise im Falle der CG-Verfahren `conjugate_gradient.py` heißt, steuert die eigentliche Optimierung. Es kreiert ein Objekt der *python*-Klasse `Optimization`, welche in einer Datei namens `opt_class.py` enthalten ist. Diese Klasse besteht aus sämtlichen für die Optimierung relevanten Methoden. Sie ändert beispielsweise die Kraftfeldparameter für die Berechnung des Gradienten und gegebenenfalls der Hesse-Matrix, ruft das *shell*-Skript auf, welches die Simulation startet, liest die berechneten physikalischen Eigenschaften ein und ermittelt den Wert der Fehlerfunktion. Zusätzlich enthält diese Klasse alle notwendigen Rechenroutinen wie beispielsweise geometrische und statistische Berechnungen sowie Schreib- und Einleseroutinen.
Die Gradientenberechnung erfolgt durch ein Skript names `gradient.py`, die der Hesse-Matrix durch ein Skript namens `Hessian.py`. Solange $||\nabla F(x^k)|| \geq 1$, wird die Abstiegsrichtung gemäß Gleichung (3.30) normalisiert, außer beim PSB-Verfahren, da sich diese Normalisierung in diesem Falle als nicht geeignet herausgestellt hat, was in Abschnitt 4.2 näher erläutert wurde. Die Evaluierung der aktuellen Iteration wird ebenfalls durch das Kontrollskript des verwendeten Algorithmus durchgeführt, das heißt, es wird überprüft, ob die Norm des Gradienten kleiner als ein sinnvoller Schwellenwert ist (gewöhnlich 0.001) und ob das Abbruchkriterium (3.67) erfüllt ist. Falls keine dieser Bedingungen erfüllt ist, wird die Abstiegsrichtung mithilfe von algorithmusspezifischen Subskripten berechnet, und die Schrittweitensteuerung wird durch ein Skript namens `step_length_class.py` ausgeführt. Dieses enthält wiederum eine *python*-Klasse namens `Step_Length_Control`, welche alle notwendigen Methoden für die Schrittweitensteuerung enthält, wie zum Beispiel die Ermittlung der zulässigen Schrittweite und eine Methode, die für den Algorithmus der Schrittweitensteuerung selbst zuständig ist. Im Falle der Armijo-Schrittweite werden zwei Subskripte aufgerufen: Zum einen das Sktipt `armijo.py`, welches eine Armijo-Schrittweite β_A^ℓ berechnet, und zum anderen das Skript `armijo_evaluate.py`, wel-

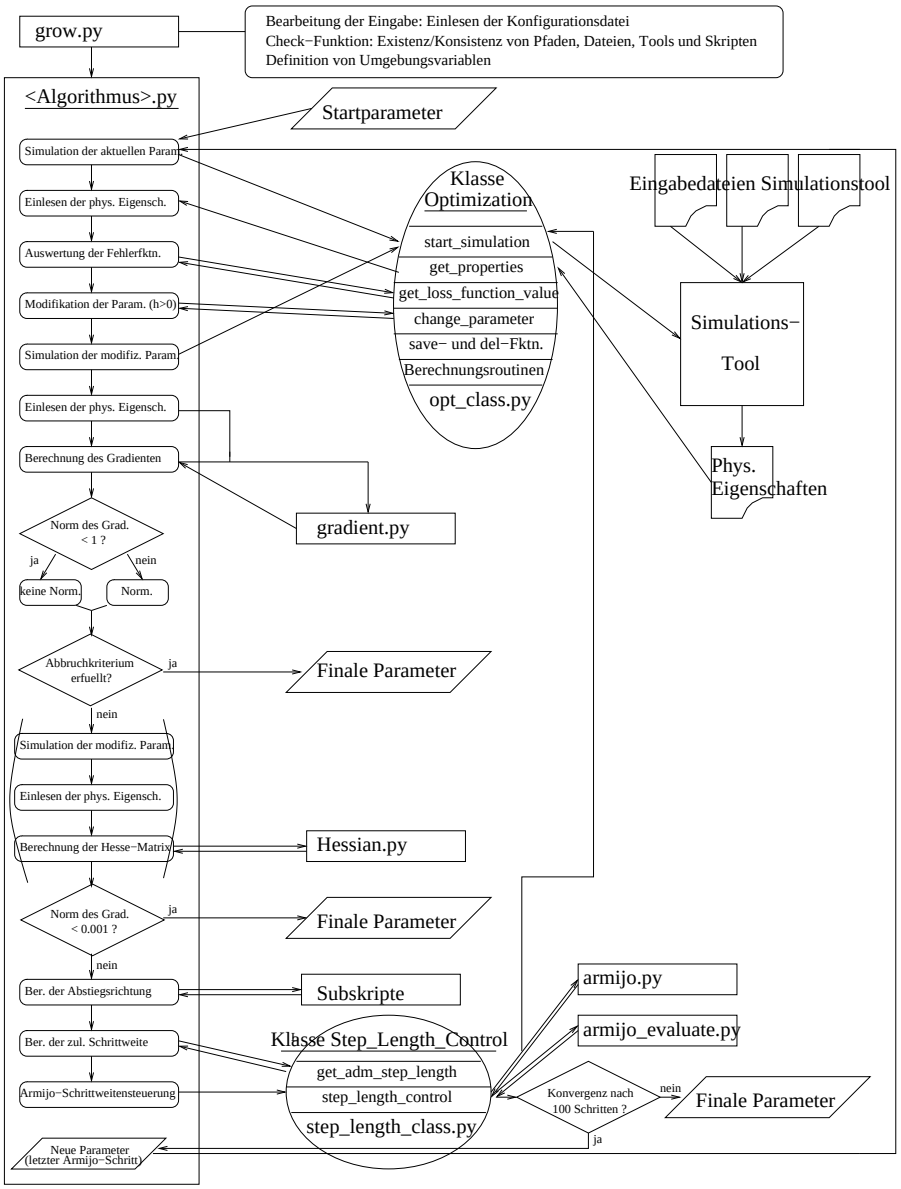


Abbildung G.1: Programmstruktur von GROW.

ches überprüft, ob diese Schrittweite gemäß der Armijo-Bedingung (3.33) akzeptiert wird oder nicht. Die letzte Iteration der Armijo-Schrittweitensteuerung ist die neue Iteration innerhalb des gesamten Optimierungsablaufs, und der Gesamtalgorithmus läuft weiter. Falls die Armijo-Schrittweitensteuerung nicht innerhalb einer sinnvollen Anzahl an Iterationen (gewöhnlich 100) konvergiert, so wird der Optimierungsablauf unterbrochen, und die aktuellen Kraftfeldparameter sind die finalen Parameter.
Der Algorithmus endet, falls das Abbruchkriterium für die Fehlerfunktion (3.67) oder aber, was jedoch gemäß Abschnitt 3.5.3 nicht erwartet werden kann, das Abbruchkriterium (3.66) für die Norm des Gradienten erfüllt ist.

G.1.3 Verwaltung der Ergebnisse

Während der Iterationen des Optimierungsablaufs erstellt GROW eine für den Benutzer lesbare Ergebnisdatei. Diese Datei wird in den in der Konfigurationsdatei angegebenen `optoutpath` geschrieben. Sie enthält den aktuellen Kraftfeldparametervektor, den Gradienten und seine Norm, die durch die Simulation berechneten physikalischen Eigenschaften für jede Temperatur, den MAPE-Wert für jede Eigenschaft sowie den Wert der Fehlerfunktion.
Zum Schluß werden der finale Parametervektor, der finale Fehlerfunktionswert, die Anzahl an Iterationen und Funktionsauswertungen sowie der finale Gradient und seine Norm herausgeschrieben.
Als Beispiel sei hier eine Ergebnisdatei gegeben, welche mithilfe des Verfahrens des steilsten Abstiegs generiert wurde. Anstelle von Simulationen wurden die Korrelationsfunktionen aus Abschnitt 3.5.2 verwendet. Optimiert wurden die Verdampfungsenthalpie (`Vapor`) und die Siededichte (`Density`) von Stickstoff an der Phasenübergangskurve, und zwar zu den Temperaturen 65, 75, 85, 95, 105 und 115 K. Das Abbruchkriterium für die Fehlerfunktion war $F(x) \leq 0.01$.

```
GROW Optimization Workflow 2009/3/23 10:52:38
Program: fits, Method: steepest_descent
Configuration file: /home/user/nitrogen/steepest_descent/armijo/T_range/vap/grow.cfg
Substance: Nitrogen

Values for x0:
x0 = 0.3101 0.331 0.02073 0.10464
gradient = 28.17 4.56 15.38 -47.90
norm of gradient = 57.84
T = 65.0: calc vapor = 6.3367, exp vapor = 5.9800, abs error = 0.3567, rel error = 0.0596
T = 65.0: calc density = 886.2005, exp density = 859.6000, abs error = 26.6005, rel error = 0.0309
T = 75.0: calc vapor = 6.2757, exp vapor = 5.6600, abs error = 0.6157, rel error = 0.1088
T = 75.0: calc density = 845.1734, exp density = 816.6700, abs error = 28.5034, rel error = 0.0349
T = 85.0: calc vapor = 6.2172, exp vapor = 5.2800, abs error = 0.9372, rel error = 0.1775
T = 85.0: calc density = 801.4362, exp density = 770.1300, abs error = 31.3062, rel error = 0.0407
T = 95.0: calc vapor = 6.2888, exp vapor = 4.8000, abs error = 1.4888, rel error = 0.3102
T = 95.0: calc density = 754.0496, exp density = 718.2600, abs error = 35.7896, rel error = 0.0498
T = 105.0: calc vapor = 5.3340, exp vapor = 4.1700, abs error = 1.1640, rel error = 0.2791
T = 105.0: calc density = 701.3797, exp density = 657.5200, abs error = 43.8597, rel error = 0.0667
T = 115.0: calc vapor = 4.4874, exp vapor = 3.2600, abs error = 1.2274, rel error = 0.3765
T = 115.0: calc density = 640.0994, exp density = 578.7000, abs error = 61.3994, rel error = 0.1061
MAPE on Density = 5.49
MAPE on Vapor = 21.86
loss = 0.384781

Values for x1:
x1 = 0.308706 0.330774 0.019966 0.107011
gradient = 21.70 4.15 11.56 -35.97
norm of gradient = 43.76
T = 65.0: calc vapor = 6.2347, exp vapor = 5.9800, abs error = 0.2547, rel error = 0.0426
T = 65.0: calc density = 876.4636, exp density = 859.6000, abs error = 16.8636, rel error = 0.0196
T = 75.0: calc vapor = 6.1731, exp vapor = 5.6600, abs error = 0.5131, rel error = 0.0907
T = 75.0: calc density = 834.7286, exp density = 816.6700, abs error = 18.0586, rel error = 0.0221
T = 85.0: calc vapor = 6.1118, exp vapor = 5.2800, abs error = 0.8318, rel error = 0.1575
T = 85.0: calc density = 790.1106, exp density = 770.1300, abs error = 19.9806, rel error = 0.0259
T = 95.0: calc vapor = 5.9631, exp vapor = 4.8000, abs error = 1.1631, rel error = 0.2423
T = 95.0: calc density = 741.5581, exp density = 718.2600, abs error = 23.2981, rel error = 0.0324
T = 105.0: calc vapor = 5.0845, exp vapor = 4.1700, abs error = 0.9145, rel error = 0.2193
T = 105.0: calc density = 687.1901, exp density = 657.5200, abs error = 29.6701, rel error = 0.0451
T = 115.0: calc vapor = 4.2219, exp vapor = 3.2600, abs error = 0.9619, rel error = 0.2951
T = 115.0: calc density = 622.9822, exp density = 578.7000, abs error = 44.2822, rel error = 0.0765
MAPE on Density = 3.70
```

```
MAPE on Vapor = 17.46
loss = 0.239212

...

Values for x50:
x50 = 0.304200 0.326315 0.011931 0.114235
gradient = -0.29 0.12 0.06 0.42
norm of gradient = 0.53
T = 65.0: calc vapor = 5.8284, exp vapor = 5.9800, abs error = -0.1516, rel error = -0.0254
T = 65.0: calc density = 866.5656, exp density = 859.6000, abs error = 6.9656, rel error = 0.0081
T = 75.0: calc vapor = 5.7651, exp vapor = 5.6600, abs error = 0.1051, rel error = 0.0186
T = 75.0: calc density = 820.6438, exp density = 816.6700, abs error = 3.9738, rel error = 0.0049
T = 85.0: calc vapor = 5.6910, exp vapor = 5.2800, abs error = 0.4110, rel error = 0.0778
T = 85.0: calc density = 770.9424, exp density = 770.1300, abs error = 0.8124, rel error = 0.0011
T = 95.0: calc vapor = 4.9639, exp vapor = 4.8000, abs error = 0.1639, rel error = 0.0342
T = 95.0: calc density = 715.7594, exp density = 718.2600, abs error = -2.5006, rel error = -0.0035
T = 105.0: calc vapor = 4.1979, exp vapor = 4.1700, abs error = 0.0279, rel error = 0.0067
T = 105.0: calc density = 651.6152, exp density = 657.5200, abs error = -5.9048, rel error = -0.0090
T = 115.0: calc vapor = 3.1471, exp vapor = 3.2600, abs error = -0.1129, rel error = -0.0346
T = 115.0: calc density = 568.8182, exp density = 578.7000, abs error = -9.8818, rel error = -0.0171
MAPE on Density = 0.73
MAPE on Vapor = 3.29
loss = 0.009933

Optimal set of parameters: 0.304200 0.326315 0.011931 0.114235
Value of loss function: 0.009933
Number of iterations: 50
Number of function evaluations: 346
Gradient: -0.29 0.12 0.06 0.42
Norm of gradient: 0.53
```

G.1.4 Erweiterungsmöglichkeiten

GROW kann durch seine generische Implementierung auf einfachste Art und Weise in Bezug auf verschiedene Aspekte erweitert werden: Eine Schnittstelle zu einem neuen Simulationstool kann innerhalb der Funktion `start_simulation` aus der Klasse `Optimization` realisiert werden. Diese Methode muß ein neues Skript aufrufen, welches Äquilibrierung und Produktion innerhalb des neuen Simulationstools steuert, ähnlich wie `mkjobequigromacs.sh`. Die zu optimierenden physikalischen Eigenschaften müssen in ASCII-Dateien geschrieben werden, welche aus zweispaltigen Tabellen bestehen: Die erste Spalte enthält die Zeitschritte und die zweite Spalte die jeweilige physikalische Eigenschaft zu jedem Zeitschritt. Die Dateien, deren Namen durch die Eigenschaft und die Endung `.list` festgelegt ist, werden von GROW automatisch mithilfe der Methode `get_properties` eingelesen, welche sich ebenfalls in der Klasse `Optimization` befindet (vergleiche Abbildung G.1).
Es ist ebenfalls ohne erheblichen Aufwand möglich, ein neues Optimierungsverfahren einzubauen. Es ist nicht notwendig, daß es sich dabei um ein gradientenbasiertes Verfahren handelt. Hat das Verfahren den Namen `XYZ`, so ist ein neues Steuerskript gemäß dem Konzept aus Abbildung G.1 namens `XYZ.py` zu programmieren, welches sämtliche für den Algorithmus notwendige Subskripte aufruft. Handelt es sich um ein gradientenbasiertes Verfahren, so kann es zur Gradientenberechnung das bereits vorhandene Skript `gradient.py` aufrufen. Im Falle einer Hesse-Matrix-Berechnung steht das Skript `Hessian.py` zur Verfügung. Eine Schrittweitensteuerung zum Erhalt der Randbedingungen ist mithilfe der Klasse `Step_Length_Class` möglich. Selbstverständlich sind alle bereits vorhandenen Rechenroutinen aus der Klasse `Optimization` ver-

wendbar. Neue Rechenroutinen können als neue Methoden in die Klasse implementiert werden. Sämtliche für das neue Optimierungsverfahren notwendigen Variablen sind in der Konfigurationsdatei anzuzeigen und durch das Hauptkontrollskript `grow.py` einzulesen.
Zur Optimierung einer weiteren physikalischen Eigenschaft ist das Steuerskript für das Simulationstool (zum Beispiel `mkjobequigromacs.sh`) zur verändern. Es kann entweder eine Tabelle erzeugen oder aber auch selbst den thermodynamischen Durchschnitt über verschiedene Zeitschritte berechnen und diesen dann als einzelne Zahl in eine separate Datei schreiben. In der GROW-Methode `get_properties` aus der Klasse `Optimization` muß die neue Datei korrekt eingelesen werden. Es ist zu beachten, daß für jede zu optimierende physikalische Eigenschaft in `grow.py` eine Umgebungsvariable definiert werden muß, die an das Steuerskript für das Simulationstool weiterzuleiten ist, damit dieses die gewünschten Zielgrößen in die korrekten Dateien schreibt.
Bei allen Erweiterungen von GROW ist zu berücksichtigen, daß sämtliche in der Konfigurationsdatei neu zu definierenden Dateien und Parameter in `grow.py` auf Existenz und Konsistenz hin überprüft werden müssen, da dies die Programmstruktur von GROW vorschreibt. Der Vorteil besteht darin, daß falsche Benutzereingaben möglichst früh erkannt werden, so daß GROW nicht erst nach einer Simulation mit einer Fehlermeldung aussteigt. Das Hauptkontrollskript enthält hierzu eine vorgegebene Struktur, die in jedem Fall eingehalten werden muß. Das bedeutet, daß jede Variable aus der Konfigurationsdatei an der richtigen Stelle in `grow.py` eingelesen und überprüft werden muß.
Obwohl die Hauptaufgabe von GROW darin besteht, Kraftfelder zu optimieren und somit als Schnittstelle zwischen numerischen Optimierungsverfahren und Simulationstools fungiert, ist ebenfalls ohne weiteres die Minimierung oder Maximierung einer beliebigen mathematischen Funktion möglich. Hierzu muß in `start_simulation` anstatt eines Steuerskripts für ein Simulationstool lediglich eine Methode aufgerufen werden, welche die zu optimierende Funktion auswertet. Die Methoden `get_properties` und `loss_function` müssen dabei umgangen werden, da es keine Dateien mit physikalischen Eigenschaften gibt und keine Fehlerfunktion zwischen simulierten und experimentellen Zielgrößen notwendig ist.
Alles in allem ist die Implementierung von GROW äußerst generisch. Die Software kann somit auf eine Vielzahl von Optimierungsproblemen angewandt werden. Wie GROW in der Praxis zum Einsatz kommt, wurde in Kapitel 5 anhand verschiedener Beispiele dargestellt.

G.2 CMA-ES

Quellcode zum globalen Optimierungsalgorithmus **C***ovariance* **M***atrix* **A***daptation* **E***volution* **S***trategy (CMA-ES)* aus Abschnitt 3.3.2 ist unter der *GNU Public License* im Internet frei verfügbar. Die in dieser Arbeit verwendete objektorientierte Java-Implementierung basiert auf einer Hauptklasse namens `CMAEvolutionStrategy`. Details bezüglich der Implementierung sind in einem entsprechenden Tutorial (Hansen, 2011) und in einer Java-Dokumentation (CMA-ES Javadoc, 2011) nachzulesen. CMA-ES ist zum größten Teil selbstparametrisierend, das heißt, es sind nur einige wenige Variablen in einer einzigen Eingabedatei zu definieren. Das folgende Beispiel zeigt eine Eingabedatei für CMA-ES im Falle von Phosgen (siehe Abschnitt 5.1.3):

```
# Pflichtvariablen
dimension = 6                          # Dimension
initialX = 0.5 0.5 0.5 0.5 1.0 1.0  # Startparameter
initialStandardDeviations = 0.23 0.23 0.23 0.33 0.66 0.66
                                       # Initiale Standaradabweichungen

# Optional, aber emfohlen
populationSize = 6                     # Populationsgroesse
stopFitness = 0.0001                   # Abbruchkriterium
outputFileNamesPrefix = /home/user/output/final_out_phosgene/outcmaes
                                       # Ausgabedatei

# Optional
randomSeed = 100                       # Zufallszahlengeneration
numberOfRestarts = 0                   # Wie oft groessere Population?
incPopSizeFactor = 1.0                 # Faktor zur Erhoehung der Population
```

Pflichtvariablen sind lediglich Dimension, Startparameter und initiale Standardabweichungen beziehungsweise Schrittweiten. Für alle anderen Variablen existieren *Default*-Werte (vergleiche Hansen (2011)). Es emfiehlt sich jedoch, die Populationsgröße und das Abbruchkriterium problemspezifisch zu wählen. Die Variable `randomSeed` legt die Art der Zufallszahlengeneration fest. `randomSeed`= 100 bedeutet, daß bei einem Neustart von CMA-ES dieselben Individuen innerhalb einer Population gewählt werden. Dies ermöglicht es beispielsweise, die CMA-ES-Optimierung abzubrechen, bestimmte Simulationseinstellungen zu ändern und die Optimierung bis zu der Stelle, an der sie abgebrochen wurde, zu wiederholen.
Die letzten beiden Parameter geben an, wie oft innerhalb der Optimierung die Populationsgröße zu erhöhen ist und um welchen Faktor. Es handelt sich dabei um ein erweitertes *Feature* von CMA-ES (siehe Auger u. Hansen (2005) für weitere Details).

G.3 VMD

V*isual* **M***olecular* **D***ynamics (VMD)* (Humphrey u. a., 1996) ist eine 3D-Visualisierungssoftware für chemische und biologische Systeme. Moleküle können dabei auf unterschiedliche Art und Weise koloriert repräsentiert werden, zum Beispiel als durch Linien verbundene Punkte, van-der-Waals-Kugeln, Zylinder, Röhren oder Bänder. Weiterhin können Trajektorien, die aus einer MD-Simulation resultieren, sowohl animiert, das heißt als Film dargestellt, als auch analysiert werden. VMD erlaubt Darstellungen auf verschiedenen Skalen. Neben der molekularen Darstellung sind insbesondere auch Visualisierungen von Enzymen, Proteinen und anderen biologischen Systemen möglich. Außerdem können Potentialhyperflächen berechnet und eingezeichnet werden. Die Software steht unter der *University of Illinois Open Source License* und ist online frei verfügbar. Sämtliche mit VMD erstellten Graphiken sind ohne Einschränkung publizierbar. In dieser Arbeit wurde VMD (Version 1.8.7) zur Visualisierung von mithilfe von *Gromacs* (siehe Anhang F.1) erstellten Trajektorien verwendet, insbesondere zur kolorierten Darstellung von Simulationsboxen (siehe zum Beispiel Abbildung 5.4(a)).

G.4 Open Babel

Open Babel (O'Boyle u. a., 2011) ist eine Toolbox für die Suche, Konvertierung, Analyse und das Abspeichern molekularer Daten in den Bereichen Chemie, Festkörpermaterie, Biochemie und ähnliches. Die Software steht unter der *GNU Public License* und ist online frei verfügbar. In dieser Arbeit wurde Open Babel (Version 1.6) zur Konvertierung von Koordinatendateien verwendet. Aus dem sehr simplen `.xyz`-Format, welches lediglich den jeweiligen Atomnamen inklusive seiner dreidimensionalen Koordinaten beinhaltet, wurde ein `.pdb`-Format erstellt, welches oftmals als Eingabe für *Gromacs* (siehe Anhang F.1) benötigt wurde.

G.5 Xfig

Xfig ist ein interaktives Zeichenprogramm, welches Zeichenobjekte wie zum Beispiel Kreise, Boxen, Linien, Spline-Kurven und ähnliches verwendet. Der Benutzer erstellt eine `.fig`-Datei und kann diese in viele auch plattformunabhängige Bildformate exportieren. Hier wurden ausschließlich `.pdf`-Dateien verwendet.
Die Software und ein Benutzerhandbuch (Xfig, 2007) sind online frei vergügbar. Bei *Xfig* handelt es sich um eine *Free-and-Open-Source-(FOSS-)Software*. Der Quellcode darf frei verwendet, kopiert und modifiziert werden. Verwendet wurde die Version 3.2.5a zur Erstellung sämtlicher 2D-Graphiken, in denen kein Koordinatensystem mit beschrifteten Skalen enthalten ist. Insbesondere wurde es zur Erstellung skizzenhafter Darstellungen benutzt (zum Beispiel Abbildung 2.3).

G.6 Xmgrace

Xmgrace ist ein 2D-Visualisierungstool, insbesondere zum Erstellen von kolorierten Graphiken in einem Koordinatensystem. Neben einem benutzerfreundlichen Kontroll-GUI ist es äußerst flexibel in Bezug auf Anzahl an gleichzeitig erstellten Graphiken und Anzahl an Kurven pro Graphik. Weiterhin verfügt es über Regressions- und Analysetools und ist dazu in der Lage, verschiedene auch plattformunabhängige Graphikformate zu erstellen. Hier wurden ausschließlich `.png`-Dateien verwendet.
Beim Erstellen einer Graphik mithilfe des Kontroll-GUIs wird automatisch eine sogenannte `.agr`-Datei erstellt, welche lesbar und durch den Benutzer modifizierbar ist.
Xmgrace steht unter der *GNU Public License*. Die Software und ein Benutzerhandbuch (Xmgrace, 2008) sind online frei vergügbar. Verwendet wurde die Version 5.1.21 zur Erstellung sämtlicher 2D-Graphiken, in denen ein Koordinatensystem mit beschrifteten Skalen enthalten ist, insbesondere zum Vergleich der durch einen Optimierungsprozeß erhaltenen Zielgrößen mit ihren experimentellen Referenzdaten (zum Beispiel Abbildung 5.5).

G.7 Gnuplot

Gnuplot ist ein 2D- und 3D-Visualisierungstool, insbesondere zum Erstellen von kolorierten Graphiken in einem Koordinatensystem. Es ist kommandozeilenbasiert, das heißt, es können Skripte geschrieben werden, welche spezifische Befehle zum Erstellen von *Gnuplot*-Graphiken enthalten. Mithilfe dieser Dateien können viele auch plattformunabhängige Bildformate erstellt werden. Hier wurden ausschließlich `.png`-Dateien verwendet.
Die Software und ein Benutzerhandbuch (Williams u. Kelley, 2007) sind online frei vergügbar. Die spezielle *Gnuplot*-Lizenz sieht vor, daß die Software kostenlos verwendet, kopiert und verbreitet werden kann, allerdings keine Verbreitung von modifiziertem Quellcode. Details bezüglich dieser Lizenz sind in (Williams u. Kelley, 2007) zu finden. Verwendet wurde die Version 4.2 zur Erstellung von 3D-Graphiken, in denen ein Koordinatensystem mit beschrifteten Skalen enthalten ist. Insbesondere wurde es zur Visualisierung von Glättungen auf dem Einheitsquadrat benutzt (zum Beispiel Abbildung 7.2).

G.8 MPI

Das *Message Passing Interface (MPI)* dient zum Nachrichtenaustausch für die Parallelrechnung auf verteilten Systemen. Eine MPI-Anwendung besteht zumeist aus mehreren Prozessen, die miteinander kommunizieren, und dies geschieht teilweise knotenübergreifend. Allgemeine Informationen zu MPI sind unter MPI (1997) zu finden. In dieser Arbeit wurden zwei verschiedene MPI-Implementierungen, abhängig vom jeweiligen Rechencluster, verwendet:

1. *MVAPICH*, Version 1.2.7: *MVAPICH* (MVAPICH, 2011) ist online frei verfügbar und ermöglicht mit einigen Restriktionen die Verwendung, Modifizierung und Verteilung des Quellcodes, welcher unter der sogenannten *Berkeley-Software-Distribution-(BSD-)Lizenz* steht.

2. *IntelMPI*, Version 3.2.2: *IntelMPI* (IntelMPI, 2011) ist kostenpflichtig und online erwerbbar. Es wird heutzutage allerdings auf vielen Rechenclustern standardmäßig installiert. Die Implementierung basiert auf der MPI-Erweiterung *MPI-2*. Hauptbestandteile der Erweiterung sind eine dynamische Prozeßverwaltung, einseitige Kommunikation und Schnittstellen zu Programmiersprachen wie *C++* und *Fortran 90*.

MPI wurde in dieser Arbeit ausschließlich zur Parallelisierung von Molekularen Simulationen verwendet, und zwar im Falle von *Gromacs* (siehe Anhang F.1) und *ms2* (siehe Anhang F.3). Molekulare Simulationen werden insbesondere bei der Berechnung von Potentialen parallelisiert, was in Abschnitt 3.6.1 dargestellt wurde. Die Akkumulation der auf den miteinander kommunizierenden Prozessoren gesammelten Daten geschieht in diesem Fall über die arithmetische MPI-Operation `MPI_SUM`. Vorher werden die Daten allerdings mit der Reduktionsoperation `MPI_Reduce` zu einem Datum reduziert.

G.9 Python

python (Python, 2011) ist eine auf vielen Benutzeroberflächen verwendbare objektorientierte Skriptsprache, deren Quellcode in Bytecode übersetzt wird, welcher anschließend von einem Interpreter gelesen wird. Die Programmiersprache ist online frei verfügbar. Die *Open-Source*-Lizenz wurde anerkannt durch die *Open Source Initiative (OSI)*. Das geistige Eigentum obliegt der *Python Software Foundation (PSF)*, siehe (PSF, 2011). Letztere fördert, schützt und erweitert die Programmiersprache und vereinfacht die internationale Kommunikation von *python*-Programmierern.
In dieser Dissertation wurde neben der Objektorientiertheit der Vorteil von *python* genutzt, daß es durch sehr einfache Kommandos als Schnittstelle zu anderen Programmiersprachen fungieren kann: Durch die *python*-Bibliothek *os* wird es auf äußerst einfache und komfortable Art und Weise ermöglicht, Funktionalitäten des Betriebssystems zu nutzen, also kommandozeilenbasierte Skripte aufzurufen. Dies ist insbesondere dann nützlich, wenn die Simulationspakete aus Anhang F parallelisiert ausgeführt werden müssen, oder aber, wenn ein *R*-Skript (siehe Anhang G.10) zur Durchführung von einfachen numerischen Berechnungsroutinen aufzurufen ist. Verwendet wurden in dieser Arbeit die *python*-Versionen 2.4.3 und 2.6.6.

G.10 Statistikpaket *R*

R ist eine Skriptsprache und eine Umgebung für Statistik und Visualisierung. Es handelt sich um ein Projekt, welches unter der *GNU Public License* steht. Handbücher zur aktuellen Version 2.13.2 sind Venables u. Smith (2011) (Statistikpaket) und R Development Core Team (2011) (Skriptsprache). Die Skriptsprache *R* und die früher entwickelte Skriptsprache *S* (Becker u. a., 1988; Chambers u. Hastie, 1992) sind sehr ähnlich.
Verwendet wurden in dieser Arbeit die Versionen 2.7.1, 2.8.1, 2.9.2 und 2.10.1. Das Statistikpaket *R* wurde zumeist zur Durchführung einfacher mathematischer Algorithmen verwendet, wie zum Beispiel für die Lösung eines LGS oder die Eigenwertzerlegung einer Matrix. Weiterhin eignet sich *R* als Visualisierungspaket ebenfalls zur Erstellung von zwei- und dreidimensionalen Graphiken: Sämtliche Entwicklungen von LJ-Parametern während einer Optimierung (zum Beispiel Abbildungen 5.7 und 5.15), Box-Plots (zum Beispiel Abbildung 4.2) oder Temperaturfits (zum Beispiel Abbildung H.1) wurden mit *R* erstellt. Hierzu wurden mittels eines *R*-Skripts automatisch `.png`- und `.pdf`-Dateien erzeugt.

Es wurden die folgenden Pakete zur Ausführung von Glättungs- und Regularisierungsverfahren verwendet, welche online über das **C***ontributed* **R** *Archive* **N***etwork (CRAN)*, siehe CRAN (2011), frei verfügbar sind:

- *e1071:*
  - **Version:** 1.6
  - **Handbuch:** Dimitriadou u. a. (2011)
  - **Verwendung:** Support-Vector-Machines (Abschnitt 6.2.2)

  Das Paket *e1071* ist in der Bibliothek *LIBSVM* (Chang u. Lin, 2011) integriert, einer freien Bibliothek für Support-Vector-Machines. Letztere ist online frei verfügbar.

- *MASS:*
  - **Version:** 7.3-16
  - **Handbuch:** Ripley (2011)
  - **Verwendung:** Ridge-Regression, gewichtete Regression (Abschnitt 6.2.3)
- *lasso2:*
  - **Version:** 1.2-12
  - **Handbuch:** Lokhorst u. a. (2011)
  - **Verwendung:** LASSO-Verfahren (Abschnitt 6.2.3)
- *earth:*
  - **Version:** 3.2-1
  - **Handbuch:** Milborrow (2011)
  - **Verwendung:** Multi-Adaptive Regression Splines (Abschnitt 6.2.3)

Weiterhin wurden die folgenden Pakete verwendet:

- *Cairo:*
  - **Version:** 1.5-0
  - **Handbuch:** Urbanek (2011)
  - **Verwendung:** Erstellen von Graphiken mit hoher Qualität (`.png` und `.pdf`)
- *Hmisc:*
  - **Version:** 3.8-3
  - **Handbuch:** Harrell Jr (2010)
  - **Verwendung:** Einzeichnen von Fehlerbalken in Temperaturfits
- *panel:*
  - **Version:** 1.0.7
  - **Handbuch:** Gentleman (2010)
  - **Verwendung:** Berechnung von Eigenwerten einer Matrix

G.11 Java

Java (Java, 2011) ist eine plattformunabhängige objektorientierte Programmiersprache, bei der Bytecode erzeugt wird, welcher in einer Laufzeitumgebung ausgeführt und in den meisten Fällen kompiliert wird. *Java* zeichnet sich insbesondere durch die einfache Verwaltung von großen Softwareprojekten und die Wiederverwendbarkeit von Softwaremodulen aus. Der Objektzugriff geschieht über Referenzen, ähnlich den Zeigern in *C++*. Das zugehörige *Java Development Kit (JDK)*, welches vom Softwarehersteller *Oracle* stammt (Oracle, 2011), steht seit November 2006 unter der *GNU Public License* und ist online frei verfügbar.
In dieser Arbeit wurde die *Java*-Implementierung des evolutionären Algorithmus CMA-ES (siehe Anhang G.2) genutzt. Dabei war der Quellcode zum Erhalt einer Schnittstelle zu den Simulationsprogrammen aus Anhang F an einigen Stellen zu modifizieren. Verwendet wurde die *Java*-Version 1.6.0_20.

H Zusätzliche Verfahren und Beweise

H.1 Effiziente Gradientenberechnung mittels reduzierter Parameter

Der in Abschnitt 3.6.4 angesprochene Algorithmus zur effizienten Gradienten- und Hesse-Matrix-Berechnung wird im folgenden näher betrachtet. Zunächst wird auf die dort erwähnten Korrekturen bezüglich Temperatur und Druck eingegangen: Man kann sich der Methode der kleinsten Quadrate bedienen, um eine lineare beziehungsweise nichtlineare Regression auf die physikalischen Größen in Abhängigkeit von Temperatur und/oder Druck anzuwenden. Diese sind für experimentelle Werte bekannt und zum Beispiel in Poling u. a. (2000) zu finden. Für die Dichte kann für einen bestimmten Temperaturbereich eine lineare Abhängigkeit von der Temperatur angenommen werden. Weiterhin ist auch eine nichtlineare temperatur- und druckabhängige Formel für die Dichte einer Flüssigkeit bekannt:

$$\rho(T,P) = \frac{\rho_0(T,P_0)}{1 - A \ln \frac{B(T)+P}{B(T)+P_0}}. \tag{H.1}$$

Dabei sind $P_0 = 1$ bar der Atmosphärendruck, $\rho_0(T,P_0) = \rho_{00} + \rho_{01}T + \rho_{02}T^2 + \rho_{03}T^3$ und $B(T) = B_0 + B_1T + B_2T^2$. Die Regressionskoeffizienten sind $\rho_{00}, \rho_{01}, \rho_{02}, \rho_{03}, B_0, B_1.B_2, A \in \mathbb{R}$. An der Grenze zwischen der flüssigen und gasförmigen Phase verwendet man den sogenannten *Guggenheim-Fit* (Guggenheim, 1945) für die reduzierte Siededichte ρ_l^* und die reduzierte Dampfdichte ρ_v^* (siehe Gleichungen (3.62) beziehungsweise (3.63)). Dieser ist zwar nur temperaturabhängig, allerdings ist der Druck durch die Siedekurve festgelegt und ergibt sich gemäß Gleichung (3.64). In diesem Fall sind somit nur Korrekturen bezüglich der Temperatur vorzunehmen. Korrekturen bezüglich des Drucks ergeben sich automatisch. Die reduzierte Verdampfungsenthalpie ist in diesem Fall über die Clausius-Clapeyron-Gleichung (Gleichung (3.65)) bestimmbar.
Für die Verdampfungsenthalpie gibt es weiterhin die folgende Temperaturabhängigkeit:

$$\Delta_v H(T) = A(1 - T_r)^\beta \exp(-\alpha T_r), \tag{H.2}$$

wobei $T_r = \frac{T}{T_c}$ die mittels der kritischen Temperatur T_c reduzierte Temperatur. Die Regressionskoeffizienten sind $A, \beta \in \mathbb{R}$. Die kritische Temperatur geht ebenfalls als Parameter mit in die Berechnung ein. Dabei werden sukzessive verschiedene Werte für T_c eingesetzt, und für jedes T_c wird eine nichtlineare Regression durchgeführt. Diejenige kritische Temperatur mit kleinstem mittleren quadratischen Fehler wird verwendet.
Die Temperaturabhängigkeit des Dampfdrucks wird durch Gleichung (3.64) beschrieben.

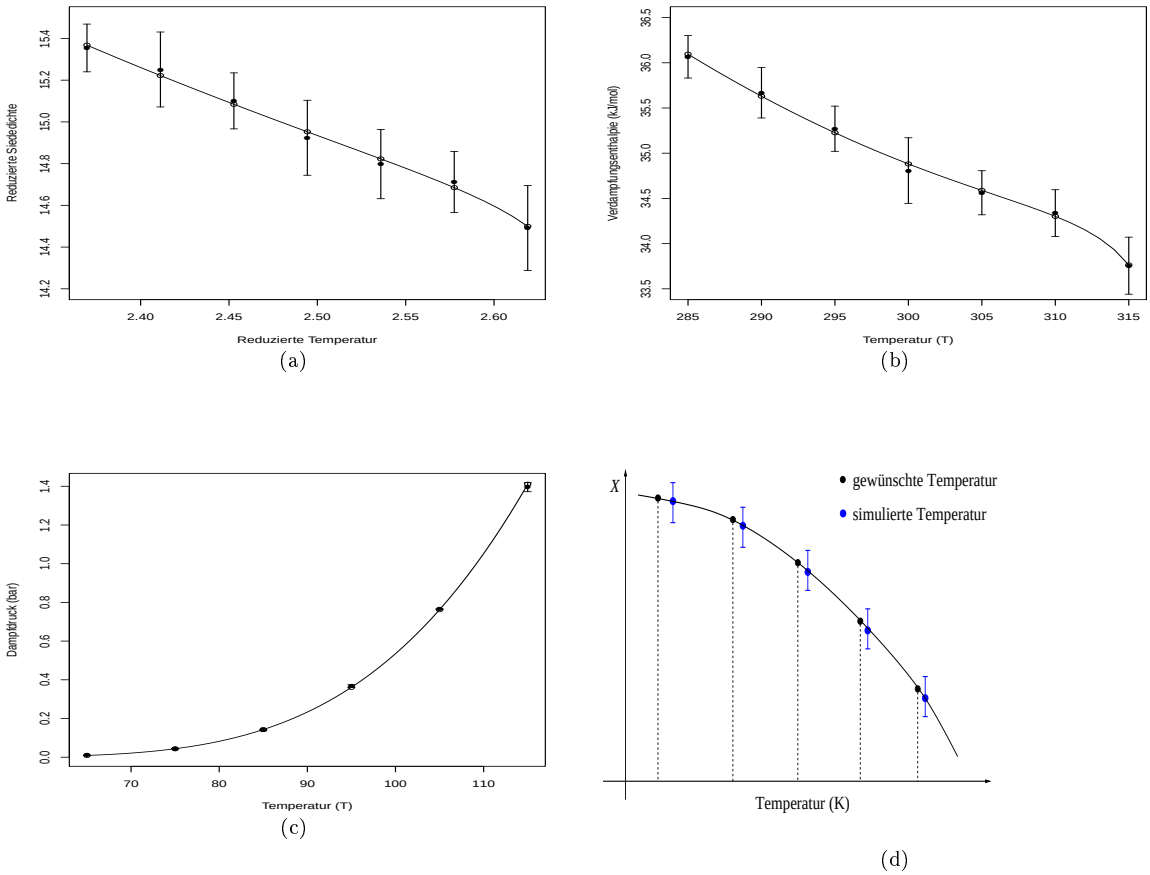


Abbildung H.1: Temperaturfits: Guggenheim-Fit für die reduzierte Siededichte (a), Verdampfungsenthalpie-Fit (b), Dampfdruck-Fit (c) und Temperaturkorrektur bei einer beliebigen Eigenschaft X (d). Die Temperaturfits für Siededichte, Verdampfungsenthalpie und Dampfdruck sind mithilfe von GROW (siehe Anhang G.1) erstellt worden.

Werden die korrekten Werte für T und P in diese Formeln eingesetzt, so erhält man gute Näherungen für die gewünschten physikalischen Eigenschaften. Obwohl diese Formeln experimentell bestimmt wurden, kann davon ausgegangen werden, daß der allgemeine Verlauf der die Abhängigkeiten beschreibenden Funktionen sich bei anderen LJ-Parametern nicht ändert. Die Regressionskoeffizienten sind jedoch bei jeder ausgelassenen Simulation neu zu bestimmen.
Falls nur temperaturabhängige Formeln für eine bestimmte physikalische Eigenschaft bekannt sind und diese verwendet werden, können die Fehler in Bezug auf den Druck nicht korrigiert werden. Auch in Bezug auf die Ladungen können keine Fehler korrigiert werden, da es keine ladungsabhängigen Formeln gibt. Wie groß sich diese Fehler auswirken und ob sie vernachlässigbar sind, wird zu prüfen sein. Die wichtigsten Temperaturfits sowie die Idee, Fehler bezüglich der Temperatur bei ausgelassenen Simulationen zu korrigieren, sind in Abbildung H.1 dargestellt: Abbildung H.1(a) zeigt den Guggenheim-Fit, Abbildung H.1(b) den Verdampfungsenthalpie-Fit, Abbildung H.1(c) den Dampfdruck-Fit und Abbildung H.1(d) die Temperaturkorrektur. Die Temperaturfits für Siededichte, Verdampfungsenthalpie und Dampfdruck sind dabei mit-

hilfe von GROW (siehe Anhang G.1) erstellt worden.

Für den Diffusionskoeffizienten ist ebenfalls eine Formel in Abhängigkeit von der Temperatur bekannt:

$$D(T) = A \exp\left(-\frac{B}{T}\right). \tag{H.3}$$

Dabei sind $A, B \in \mathbb{R}$ die Regressionskoeffizienten.

Im folgenden wird der Fall betrachtet, daß nicht nur jeweils ein σ und ε optimiert werden soll, sondern mehrere, das heißt für alle Wechselwirkungszentren. Es gelte also für einen bestimmten Kraftfeldparametervektor $x \in \mathbb{R}^{N_\varepsilon + N_\sigma}$, wobei es sich bei N_ε um die Anzahl an betrachteten ε und bei N_σ um die Anzahl an betrachteten σ handelt. Es ist zu beachten, daß in diesem Abschnitt bisher $N_\varepsilon = N_\sigma = 1$ galt. Für die Auswertung von $x = (\sigma_1, \sigma_2, \sigma_3, ..., \varepsilon_1, \varepsilon_2, \varepsilon_3, ...)^T$ wird folgende Simulation durchgeführt:

$$\sigma_1, \sigma_2, \sigma_3, ..., \varepsilon_1, \varepsilon_2, \varepsilon_3, ... \rightarrow \left\{ \begin{array}{ccc} \frac{\sigma_1}{\sigma_R} & = & \sigma_1^* \\ \frac{\sigma_2}{\sigma_R} & = & \sigma_2^* \\ \frac{\sigma_3}{\sigma_R} & = & \sigma_3^* \\ & \vdots & \\ \frac{\varepsilon_1}{\varepsilon_R} & = & \varepsilon_1^* \\ \frac{\varepsilon_2}{\varepsilon_R} & = & \varepsilon_2^* \\ \frac{\varepsilon_3}{\varepsilon_R} & = & \varepsilon_3^* \end{array} \right\} \overset{\text{Sim.},T,P,q}{\rightarrow} \left\{ \begin{array}{ccccc} \Delta_v H^* & = & \frac{\Delta_v H}{\varepsilon_R} & \rightarrow & \Delta_v H \\ \rho^* & = & \rho \sigma_R^3 & \rightarrow & \rho \\ D^* & = & \frac{D}{\sigma_R \sqrt{\varepsilon_R}} & \rightarrow & D \end{array} \right\}. \tag{H.4}$$

Wird nun beispielsweise ein um h verändertes $\tilde{\sigma}_1$ betrachtet, so ergibt sich der Referenzwert $\tilde{\sigma}_R$ aus der Gleichung $\frac{\tilde{\sigma}_1}{\tilde{\sigma}_R} = \sigma_1^*$. Dadurch ist auch jedes andere $\tilde{\sigma}_i$ für $i \neq 1$ festgelegt. Das bedeutet, daß die Kraftfeldparameter nicht unabhängig voneinander verändert werden können, falls Simulationen ersetzt werden sollen. Die klassischen Gradientenkomponenten, also die partiellen Ableitungen in Richtung der Einheitsvektoren $e_i,\ i = 1, ..., N_\varepsilon + N_\sigma$, können daher nicht berechnet werden. Gemäß dem folgenden Lemma können nur Richtungsableitungen in Richtungen der Form $v_i = \left(\frac{\sigma_i + h}{\sigma_i^*}\sigma_1^* - \sigma_1, \frac{\sigma_i + h}{\sigma_i^*}\sigma_2^* - \sigma_2, ..., \frac{\sigma_i + h}{\sigma_i^*}\sigma_{N_\sigma}^* - \sigma_{N_\sigma}, \underbrace{0, 0, ..., 0}_{N_\varepsilon} \right)^T,\ i = 1, ..., N_\sigma$, sowie $v_j = \left(\underbrace{0, 0, ..., 0}_{N_\sigma}, \frac{\varepsilon_i + h}{\varepsilon_i^*}\varepsilon_1^* - \varepsilon_1, \frac{\varepsilon_i + h}{\varepsilon_i^*}\varepsilon_2^* - \varepsilon_2, ..., \frac{\varepsilon_i + h}{\varepsilon_i^*}\varepsilon_{N_\varepsilon}^* - \varepsilon_{N_\varepsilon} \right)^T,\ j = 1, ..., N_\varepsilon$, ermittelt werden.
Es ist zu beachten, daß für die Komponenten v_{ii} beziehungsweise v_{jj} gilt: $v_{ii} = v_{jj} = h$.

Lemma H.1.1 (Effiziente Gradientenberechnung mit reduzierten Parametern)**.** *Zur effizienten Gradientenberechnung können statt der klassischen Vektoren* $(\sigma_1, \sigma_2, ..., \tilde{\sigma}_i, ..., \sigma_{N_\sigma}, \varepsilon_1, \varepsilon_2, ..., \varepsilon_{N_\varepsilon})^T$, $i = 1, ..., N_\sigma$, *beziehungsweise* $(\sigma_1, \sigma_2, ..., \sigma_{N_\sigma}, \varepsilon_1, \varepsilon_2, ..., \tilde{\varepsilon}_j, ..., \varepsilon_{N_\varepsilon})^T,\ j = 1, ..., N_\varepsilon$, *nur Vektoren*

der Form

$$w_i := \left(\frac{\sigma_i+h}{\sigma_i^*}\sigma_1^*, \frac{\sigma_i+h}{\sigma_i^*}\sigma_2^*, ..., \frac{\sigma_i+h}{\sigma_i^*}\sigma_{N_\sigma}^*, \varepsilon_1, \varepsilon_2, ..., \varepsilon_{N_\varepsilon}\right)^T,\ i = 1, ..., N_\sigma, \tag{H.5}$$

$$w_j := \left(\sigma_1, \sigma_2, ..., \sigma_{N_\sigma}, \frac{\varepsilon_i+h}{\varepsilon_i^*}\varepsilon_1^*, \frac{\varepsilon_i+h}{\varepsilon_i^*}\varepsilon_2^*, ..., \frac{\varepsilon_i+h}{\varepsilon_i^*}\varepsilon_{N_\varepsilon}^*\right)^T,\ j = 1, ..., N_\varepsilon, \tag{H.6}$$

verwendet werden. Für die Komponenten w_{ii} beziehungsweise w_{jj} gilt: $w_{ii} = \tilde{\sigma}_i,\ w_{jj} = \tilde{\varepsilon}_j$.

Beweis: Wird im ersten Schritt σ_1 verändert, ergibt sich $\tilde{\sigma}_R$ gemäß $\tilde{\sigma}_R = \frac{\tilde{\sigma}_1}{\sigma_1^*}$. Für alle anderen Indizes $i = 2, ..., N_\sigma$ gilt dann $\tilde{\sigma}_i = \tilde{\sigma}_R\sigma_i^* = \frac{\tilde{\sigma}_1}{\sigma_1^*}\sigma_i^* = \frac{\sigma_1+h}{\sigma_1^*}\sigma_i^*$, woraus sich der Vektor w_1 ergibt mit $w_{11} = \tilde{\sigma}_1$.
Verändert man ein beliebiges $\sigma_i,\ i = 1, ..., N_\sigma$, so ergibt sich $\tilde{\sigma}_R$ gemäß $\tilde{\sigma}_R = \frac{\tilde{\sigma}_i}{\sigma_i^*}$, woraus sich der Vektor w_i ergibt mit $w_{ii} = \tilde{\sigma}_i$. Da σ und ε unabhängig voneinander verändert werden können, ändern sich die $\varepsilon_j,\ j = 1, ..., N_\varepsilon$, nicht.
Umgekehrt ergeben sich die $w_j,\ j = 1, ..., N_\varepsilon$, durch Änderung von ε und Festhaltung von σ. Es gilt dann $w_{jj} = \tilde{\varepsilon}_j,\ j = 1, ..., N_\varepsilon$. □

Es ist zu beachten, daß auch Änderungen der Form $\tilde{\sigma} = \sigma + \kappa h$ beziehungsweise $\tilde{\varepsilon} = \varepsilon + \nu h$ betrachtet werden können, wobei $\kappa > -\frac{\sigma}{h}$ und $\nu > -\frac{\varepsilon}{h}$. Vektoren $v_i,\ i = 1, ..., N_\sigma$, und $v_j,\ j = 1, ..., N_\varepsilon$, die derartige Änderungen beinhalten, werden hier als *zulässige* Vektoren bezeichnet. Als Referenzwerte können $\sigma_R = \sigma_1$ und $\varepsilon_R = \varepsilon_1$ gewählt werden. Dann gilt $\sigma_1^* = \varepsilon_1^* = 1$. Eine weitere Möglichkeit besteht darin, den jeweils größten Kraftfeldparameter als Referenzwert zu wählen, so daß die reduzierten Kraftfeldparameter im Intervall $(0, 1]$ liegen. Die Wahl der Referenzwerte ist generell gesehen beliebig, allerdings sollte die Reduktion auch als eine Art Normalisierung angesehen werden, so daß numerische Probleme vermieden werden können.
Die Komponenten von w_i beziehungsweise w_j ergeben insgesamt $N_\varepsilon + N_\sigma = N$ neue physikalische Eigenschaften $\tilde{\rho}_p, \tilde{\Delta_v H_p}, (p_\sigma)_p,\ p = 1, ..., N$, allerdings zu anderen als den gewünschten Temperaturen. Mithilfe der bereits erwähnten (nicht-)linearen Regressionsfunktionen läßt sich dies korrigieren, indem die Regressionskoeffizienten mithilfe der eigentlichen Temperaturen $\tilde{T} = \frac{T^*\varepsilon_R}{k_B}$ ermittelt werden und die gewünschten Temperaturen T in die so erhaltenen Regressionsfunktionen eingesetzt werden. Die Richtungsableitungen $r_p,\ p = 1, ..., N$, der physikalischen Eigenschaften und somit der Gesamtfehlerfunktion in Richtung der Vektoren $v_p,\ p = 1, ..., N$, lassen sich über Differenzenquotienten ermitteln. Der Gradient der Fehlerfunktion $\nabla F(x)$ läßt sich nun über das folgende LGS bestimmen:

$$\begin{aligned} \langle \nabla F(x), v_1 \rangle &= r_1 \\ \langle \nabla F(x), v_2 \rangle &= r_2 \\ \vdots &= \vdots \\ \langle \nabla F(x), v_N \rangle &= r_N. \end{aligned} \tag{H.7}$$

Rein theoretisch gesehen läßt sich innerhalb eines Optimierungsablaufs für jede Iteration x^k der Gradient $\nabla F(x^k)$ ohne jede weitere Simulation zumindest näherungsweise berechnen. Approximationsfehler ergeben sich aus den Approximationsfehlern der nichtlinearen Regressionen sowie aus den nicht korrigierbaren Fehlern bezüglich der Drücke und Ladungen, die sich jedoch in der Praxis nicht sehr gravierend auswirken, was in Abschnitt 4.3 noch genauer dargelegt wird. Das Hauptproblem ist jedoch auch hier das statistische Rauschen. Die Vektoren $v_i,\ i = 1, ..., N$, müssen eine Basis bilden, und die Determinante der Matrix $V := (v_1|...|v_N)$ ist abhängig von der Wahl von h. Da Veränderungen eines Kraftfeldparameters nahezu dieselben Veränderungen in Bezug auf sämtliche andere Kraftfeldparameter der gleichen Sorte (σ oder ε) bewirken, sind die Vektoren $v_i,\ i = 1, ..., N_\sigma$, beziehungsweise $v_j,\ j = 1, ..., N_\varepsilon$, annähernd kollinear, was zu einem schlecht konditionierten Gleichungssystem (H.7) führt. Somit haben kleine Ungenauigkeiten in den Richtungsableitungen $r_p,\ p = 1, ..., N$, erhebliche Auswirkungen auf die Lösung $\nabla F(x)$.

H.2 Effiziente Berechnung der Hesse-Matrix mittels reduzierter Parameter

Auch die $N + \frac{N(N-1)}{2}$ Einträge der Hesse-Matrix können theoretisch betrachtet ohne Simulation berechnet werden. In diesem Abschnitt wird bewiesen, daß anstelle der Einheitsvektoren für die klassischen partiellen Ableitungen neben den Basisvektoren $v_i,\ i = 1, ..., N_\sigma$, und $v_j,\ j = 1, ..., N_\varepsilon$, auch beliebige Vielfache und Summen zweier Vektoren verwendet werden können, was für eine effiziente Berechnung der Hesse-Matrix notwendig ist. Um dies zu veranschaulichen, werden nur die v_i betrachtet. Die Betrachtung für die v_j ergibt sich analog.
Es sei oBdA $i = 1$, das heißt σ_1 wird zu $\sigma_1 + h$, und die anderen Komponenten von v_1 ergeben sich gemäß Lemma H.1.1. Es gilt also:

$$v_1 = \left(h, \frac{\sigma_1 + h}{\sigma_1^*}\sigma_2^* - \sigma_2, ..., \frac{\sigma_1 + h}{\sigma_1^*}\sigma_{N_\sigma}^* - \sigma_{N_\sigma}, \underbrace{0, 0, ..., 0}_{N_\varepsilon} \right)^T .$$

Zur Berechnung der Diagonalelemente wird die Auswertung von $2v_1 + x$ benötigt. Das bedeutet, es muß überprüft werden, ob $2v_1$ ein zulässiger Vektor ist. Es ist klar, daß der Vektor

$$\bar{v}_1 := \left(2h, \frac{\sigma_1 + 2h}{\sigma_1^*}\sigma_2^* - \sigma_2, ..., \frac{\sigma_1 + 2h}{\sigma_1^*}\sigma_{N_\sigma}^* - \sigma_{N_\sigma}, \underbrace{0, 0, ..., 0}_{N_\varepsilon} \right)^T$$

zulässig ist. Es gilt aber:

$$\begin{aligned} \forall_{i=1,...,N_\sigma}\ \frac{1}{2}\left(\frac{\sigma_1 + 2h}{\sigma_1^*}\sigma_i^* - \sigma_i\right) &= \frac{\frac{\sigma_1}{2} + h}{\sigma_1^*}\sigma_i^* - \frac{\sigma_i}{2} = \frac{\frac{\sigma_1}{2} + h}{\sigma_1}\sigma_i - \frac{\sigma_i}{2} \\ &= \left(\frac{1}{2} + \frac{h}{\sigma_1}\right)\sigma_i - \frac{\sigma_i}{2} = \left(1 + \frac{h}{\sigma_1}\right)\sigma_i - \sigma_i \\ &= \frac{\sigma_1 + h}{\sigma_1}\sigma_i - \sigma_i = \frac{\sigma_1 + h}{\sigma_1^*}\sigma_i^* - \sigma_i. \end{aligned}$$

Somit folgt $\frac{1}{2}\bar{v}_1 = v_1$, und $2v_1 = \bar{v}_1$ ist ein zulässiger Vektor. Das bedeutet, daß auch die Änderung sämtlicher Kraftfeldparameter um $2h$ ohne Simulation durchgeführt werden kann. Es gilt dann $\tilde{\sigma}_R = \frac{\sigma+2h}{\sigma^*}$ beziehungsweise $\tilde{\varepsilon}_R = \frac{\varepsilon+2h}{\varepsilon^*}$.
Für die Nichtdiagonalelemente werden Summen aus jeweils zwei Vektoren v_i beziehungsweise v_j benötigt, da diese auf der Änderungen zweier Kraftfeldparameter gleichzeitig um h beruhen. Es werden OBdA die Parameter σ_1 und σ_2 betrachtet. Für die Summe von v_1 und v_2 gilt:

$$
\begin{aligned}
v_1 + v_2 &= \begin{pmatrix} h \\ \frac{\sigma_1+h}{\sigma_1^*}\sigma_2^* - \sigma_2 \\ \frac{\sigma_1+h}{\sigma_1^*}\sigma_3^* - \sigma_3 \\ \vdots \\ \frac{\sigma_1+h}{\sigma_1^*}\sigma_{N_\sigma}^* - \sigma_{N_\sigma} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{\sigma_2+h}{\sigma_2^*}\sigma_1^* - \sigma_1 \\ h \\ \frac{\sigma_2+h}{\sigma_2^*}\sigma_3^* - \sigma_3 \\ \vdots \\ \frac{\sigma_2+h}{\sigma_2^*}\sigma_{N_\sigma}^* - \sigma_{N_\sigma} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \left(\frac{\sigma_1+h}{\sigma_1^*} + \frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_1^* - 2\sigma_1 \\ \left(\frac{\sigma_1+h}{\sigma_1^*} + \frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_2^* - 2\sigma_2 \\ \left(\frac{\sigma_1+h}{\sigma_1^*} + \frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_3^* - 2\sigma_3 \\ \vdots \\ \left(\frac{\sigma_1+h}{\sigma_1^*} + \frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_{N_\sigma}^* - 2\sigma_{N_\sigma} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.
\end{aligned}
$$

Für die erste Komponente von $v_1 + v_2$ gilt:

$$
\begin{aligned}
\left(\frac{\sigma_1+h}{\sigma_1^*} + \frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_1^* - 2\sigma_1 &= \frac{\sigma_2+h}{\sigma_2}\sigma_1 + h - \sigma_1 = \left(1 + \frac{h}{\sigma_2}\right)\sigma_1 + h - \sigma_1 = \frac{h}{\sigma_2}\sigma_1 + h \\
&= \frac{h\sigma_1 + h\sigma_2}{\sigma_2} + \sigma_1 - \sigma_1 = \sigma_1\left(\frac{h}{\sigma_2} + \frac{h\sigma_2}{\sigma_1\sigma_2} + 1\right) - \sigma_1 \\
&= \frac{\sigma_1 h + \sigma_2 h + \sigma_1\sigma_2}{\sigma_1^*\sigma_2}\sigma_1^* - \sigma_1 = \frac{\sigma_1 + h\left(1 + \frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^*}\sigma_1^* - \sigma_1.
\end{aligned}
$$

Für die zweite Komponente gilt analog (man vertausche σ_1 und σ_2):

$$\begin{aligned}\left(\frac{\sigma_1+h}{\sigma_1^*}+\frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_2^*-2\sigma_2 &= \left(\frac{\sigma_2+h}{\sigma_2^*}+\frac{\sigma_1+h}{\sigma_1^*}\right)\sigma_2^*-2\sigma_2\\ &= \frac{\sigma_2 h+\sigma_1 h+\sigma_1\sigma_2}{\sigma_1^*\sigma_2}\sigma_2^*-\sigma_2=\frac{\sigma_1+h\left(1+\frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^*}\sigma_2^*-\sigma_2.\end{aligned}$$

Für jede weitere Komponente $i\in\{3,...,N_\sigma\}$ gilt:

$$\begin{aligned}\left(\frac{\sigma_1+h}{\sigma_1^*}+\frac{\sigma_2+h}{\sigma_2^*}\right)\sigma_i^*-2\sigma_i &= \frac{\sigma_1+h}{\sigma_1^*}\sigma_i^*+\frac{\sigma_2+h}{\sigma_2}\sigma_i-2\sigma_i=\frac{\sigma_1+h}{\sigma_1^*}\sigma_i^*+\left(1+\frac{h}{\sigma_2}\right)\sigma_i-2\sigma_i\\ &= \frac{\sigma_1+h}{\sigma_1^*}\sigma_i^*+\frac{h}{\sigma_2^*}\sigma_i^*-\sigma_i=\left(\frac{\sigma_1+h}{\sigma_1^*}+\frac{h}{\sigma_2^*}\right)\sigma_i^*-\sigma_i\\ &= \frac{\sigma_1\sigma_2^*+h\sigma_2^*+h\sigma_1^*}{\sigma_1^*\sigma_2^*}\sigma_i^*-\sigma_i=\frac{\sigma_1+h\left(1+\frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^*}\sigma_i^*-\sigma_i.\end{aligned}$$

Werden somit zwei σ beziehungsweise zwei ε gleichzeitig verändert, so ist der resultierende Vektor zulässig, denn es gilt $\kappa=\kappa_{ij}=1+\frac{\sigma_i}{\sigma_j}$ beziehungsweise $\nu=\nu_{ij}=1+\frac{\varepsilon_i}{\varepsilon_j}$. Als Referenzwerte ergeben sich $\tilde{\sigma}_R^{ij}=\frac{\sigma_i+\kappa_{ij}h}{\sigma_i^*}$ beziehungsweise $\tilde{\varepsilon}_R^{ij}=\frac{\varepsilon_i+\nu_{ij}h}{\varepsilon_i^*}$.
Damit ist gezeigt, daß auch v_1+v_2 zulässig ist. Verändert man ein σ und ein ε gleichzeitig, so ist der resultierende Vektor ebenfalls zulässig, da diese Veränderungen unabhängig voneinander erfolgen. Also ist auch die Hesse-Matrix theoretisch gesehen ohne Simulationen berechenbar.

H.3 Interpolationsfehler bei Dünnen Gittern

Im folgenden werden die in Abschnitt 6.4.1 angegebenen Sätze für die Abschätzung des Dünn-Gitter-Interpolationsfehlers unter Verwendung der Kombinationsmethode aus Abschnitt 6.1.3 für den 2D- und 3D-Fall bewiesen. Die Beweise gehen auf Griebel u. a. (1990) zurück.

Satz H.3.1 (Dünn-Gitter-Interpolationsfehler in 2D). *Es seien $u\in S_{\bar{\ell}}^0$ die exakte Lösung des Interpolationsproblems auf einem vollen Gitter vom Level $\hat{\ell}$ und $\bar{\ell}=(\hat{\ell},\hat{\ell})\in\mathbb{N}^2$. Die Interpolation erfolge von einem Dünnen Gitter vom Level $\hat{\ell}$ auf das volle Gitter. Weiterhin existiere für alle $u_{i,j}\in T_{i,j}$ mit $i+j\in\{\hat{\ell},\hat{\ell}+1\}$ punktweise eine asymptotische Fehlerentwicklung vom Typ*

$$u-u_{i,j}=C_1(h_i)h_i^2+C_2(h_j)h_j^2+D(h_i,h_j)h_i^2h_j^2,$$

wobei $\forall_i\ |C_1(h_i)|\le\kappa$, $\forall_j\ |C_2(h_j)|\le\kappa$ und $\forall_{i,j}\ |D(h_i,h_j)|\le\kappa,\ \kappa>0$. Dann gilt

$$|u-\hat{u}_{\hat{\ell}}^c|=\mathcal{O}\left(h_{\hat{\ell}}^2\log(h_{\hat{\ell}})^{-1}\right).$$

Beweis: Es gilt:

$$
\begin{aligned}
u - \hat{u}_{\hat{\ell}}^c &\overset{(6.10)}{=} u - \sum_{i+j=\hat{\ell}+1} u_{i,j} + \sum_{i+j=\hat{\ell}} u_{i,j} \\
&= \sum_{i+j=\hat{\ell}+1} (u - u_{i,j}) - \sum_{i+j=\hat{\ell}} (u - u_{i,j})
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(6.54)}{=} \sum_{i+j=\hat{\ell}+1} C_1(h_i)h_i^2 + C_2(h_j)h_j^2 + D(h_i,h_j)h_i^2h_j^2 \\
&\quad - \sum_{i+j=\hat{\ell}} C_1(h_i)h_i^2 + C_2(h_j)h_j^2 + D(h_i,h_j)h_i^2h_j^2 \\
&\overset{i+j=\hat{\ell}+1 \Rightarrow\ i\le\hat{\ell}}{=} \sum_{i=1}^{\hat{\ell}} C_1(h_i)h_i^2 - \sum_{i=1}^{\hat{\ell}-1} C_1(h_i)h_i^2 \\
&\quad + \sum_{j=1}^{\hat{\ell}} C_2(h_j)h_j^2 - \sum_{j=1}^{\hat{\ell}-1} C_2(h_j)h_j^2 \\
&\quad + \sum_{i+j=\hat{\ell}+1} D(h_i,h_j)h_i^2h_j^2 - \sum_{i+j=\hat{\ell}} D(h_i,h_j)h_i^2h_j^2 \\
&\overset{h_i^2=\left(\frac{1}{4}\right)^i}{=} C_1(h_{\hat{\ell}})h_{\hat{\ell}}^2 + C_2(h_{\hat{\ell}})h_{\hat{\ell}}^2 \\
&\quad + \sum_{i+j=\hat{\ell}+1} \left(\frac{1}{4}\right)^{i+j} D(h_i,h_j) - \sum_{i+j=\hat{\ell}} \left(\frac{1}{4}\right)^{i+j} D(h_i,h_j) \\
&= C_1(h_{\hat{\ell}})h_{\hat{\ell}}^2 + C_2(h_{\hat{\ell}})h_{\hat{\ell}}^2 \\
&\quad + \underbrace{\left(\frac{1}{4}\right)^{\hat{\ell}+1}}_{=\frac{1}{4}h_{\hat{\ell}}^2} \sum_{i+j=\hat{\ell}+1} D(h_i,h_j) - \underbrace{\left(\frac{1}{4}\right)^{\hat{\ell}}}_{=h_{\hat{\ell}}^2} \sum_{i+j=\hat{\ell}} D(h_i,h_j) \\
&= \left(C_1(h_{\hat{\ell}}) + C_2(h_{\hat{\ell}}) + \frac{1}{4} \sum_{i+j=\hat{\ell}+1} D(h_i,h_j) - \sum_{i+j=\hat{\ell}} D(h_i,h_j) \right) h_{\hat{\ell}}^2.
\end{aligned}
$$

Somit folgt nach Voraussetzung:

$$\begin{aligned}
|u - \hat{u}^c_{\hat{\ell}}| &\leq \left(\left|C_1(h_{\hat{\ell}})\right| + \left|C_2(h_{\hat{\ell}})\right| + \left| \frac{1}{4} \sum_{i+j=\hat{\ell}+1} D(h_i, h_j) - \sum_{i+j=\hat{\ell}} D(h_i, h_j) \right| \right) h^2_{\hat{\ell}} \\
&\leq \left(\left|C_1(h_{\hat{\ell}})\right| + \left|C_2(h_{\hat{\ell}})\right| + \frac{1}{4} \sum_{i+j=\hat{\ell}+1} |D(h_i, h_j)| - \sum_{i+j=\hat{\ell}} |D(h_i, h_j)| \right) h^2_{\hat{\ell}} \\
&\leq \left(\kappa + \kappa + \frac{1}{4}\kappa\hat{\ell} - \kappa(\hat{\ell} - 1) \right) h^2_{\hat{\ell}} \\
&= \left(2 + \frac{1}{4}\hat{\ell} + \hat{\ell} - 1 \right) \kappa h^2_{\hat{\ell}} = \left(1 + \frac{5}{4}\hat{\ell} \right) \kappa h^2_{\hat{\ell}} \\
&\overset{h_{\hat{\ell}}=2^{-\hat{\ell}} \Rightarrow \hat{\ell}=\log\left(h^{-1}_{\hat{\ell}}\right)}{=} \left(1 + \frac{5}{4} \log\left(h^{-1}_{\hat{\ell}}\right) \right) \kappa h^2_{\hat{\ell}}.
\end{aligned}$$

Es folgt sofort die Behauptung. □

Satz H.3.2 (Dünn-Gitter-Interpolationsfehler in 3D). *Es seien $u \in S^0_{\bar{\ell}}$ die exakte Lösung des Interpolationsproblems auf einem vollen Gitter vom Level $\hat{\ell}$ und $\bar{\ell} = (\hat{\ell}, \hat{\ell}, \hat{\ell}) \in \mathbb{N}^3$. Die Interpolation erfolge von einem Dünnen Gitter vom Level $\hat{\ell}$ auf das volle Gitter. Weiterhin existiere für alle $u_{i,j,k} \in T_{i,j,k}$ mit $i + j + k \in \{\hat{\ell}, \hat{\ell} + 1, \hat{\ell} + 2\}$ punktweise eine asymptotische Fehlerentwicklung vom Typ*

$$\begin{aligned}
u - u_{i,j,k} &= C_1(h_i)h_i^2 + C_2(h_j)h_j^2 + C_3(h_k)h_k^2 \\
&+ D_1(h_i, h_j)h_i^2h_j^2 + D_2(h_i, h_k)h_i^2h_k^2 + D_3(h_j, h_k)h_j^2h_k^2 \\
&+ E(h_i, h_j, h_k)h_i^2h_j^2h_k^2,
\end{aligned}$$

wobei $\forall_{p=1,2,3}\ \forall_i\ |C_p(h_i)| \leq \kappa$, $\forall_{p=1,2,3}\ \forall_{i,j}\ |D_p(h_i, h_j)| \leq \kappa$ und $\forall_{i,j,k}\ |E(h_i, h_j, h_k)| \leq \kappa,\ \kappa > 0$. Dann gilt

$$|u - \hat{u}^c_{\hat{\ell}}| = \mathcal{O}\left(h^2_{\hat{\ell}} \left(\log(h_{\hat{\ell}})^{-1} \right)^2 \right).$$

Beweis: Es gilt:

$$u - \hat{u}^c_{\hat{\ell}} \overset{(6.11)}{=} u - \sum_{i+j+k=\hat{\ell}+2} u_{i,j,k} + 2 \sum_{i+j+k=\hat{\ell}+1} u_{i,j,k} - \sum_{i+j+k=\hat{\ell}} u_{i,j,k}$$

Aus $i + j + k = \hat{\ell} + 2$ folgt $i \leq \hat{\ell}$, aus $i + j + k = \hat{\ell} + 1$ folgt $i \leq \hat{\ell} - 1$ und aus $i + j + k = \hat{\ell}$ folgt

$i \leq \hat{\ell} - 2$. Wegen $\sum_{i=1}^{\hat{\ell}} i - 2\sum_{i=1}^{\hat{\ell}-1} i + \sum_{i=1}^{\hat{\ell}-2} i = \binom{\hat{\ell}+1}{2} - 2\binom{\hat{\ell}}{2} + \binom{\hat{\ell}-1}{2} = 1$ folgt somit:

$$
\begin{aligned}
u - \hat{u}_{\hat{\ell}}^c &= \sum_{i+j+k=\hat{\ell}+2} (u - u_{i,j,k}) - 2 \sum_{i+j+k=\hat{\ell}+1} (u - u_{i,j,k}) + \sum_{i+j+k=\hat{\ell}} (u - u_{i,j,k}) \\
&\overset{(6.58)}{=} \sum_{i=1}^{\hat{\ell}} (\hat{\ell} - i + 1) C_1(h_i) h_i^2 + \sum_{j=1}^{\hat{\ell}} (\hat{\ell} - j + 1) C_2(h_j) h_j^2 + \sum_{k=1}^{\hat{\ell}} (\hat{\ell} - k + 1) C_3(h_k) h_k^2 \\
&- 2 \left(\sum_{i=1}^{\hat{\ell}-1} (\hat{\ell} - i) C_1(h_i) h_i^2 + \sum_{j=1}^{\hat{\ell}-1} (\hat{\ell} - j) C_2(h_j) h_j^2 + \sum_{k=1}^{\hat{\ell}-1} (\hat{\ell} - k) C_3(h_k) h_k^2 \right) \\
&+ \sum_{i=1}^{\hat{\ell}-2} (\hat{\ell} - i - 1) C_1(h_i) h_i^2 + \sum_{j=1}^{\hat{\ell}-2} (\hat{\ell} - j - 1) C_2(h_j) h_j^2 + \sum_{k=1}^{\hat{\ell}-2} (\hat{\ell} - k - 1) C_3(h_k) h_k^2 \\
&+ \sum_{i+j \leq \hat{\ell}+1} D_1(h_i, h_j) h_i^2 h_j^2 + \sum_{i+k \leq \hat{\ell}+1} D_2(h_i, h_k) h_i^2 h_k^2 + \sum_{j+k \leq \hat{\ell}+1} D_3(h_j, h_k) h_j^2 h_k^2 \\
&- 2 \left(\sum_{i+j \leq \hat{\ell}} D_1(h_i, h_j) h_i^2 h_j^2 + \sum_{i+k \leq \hat{\ell}} D_2(h_i, h_k) h_i^2 h_k^2 + \sum_{j+k \leq \hat{\ell}} D_3(h_j, h_k) h_j^2 h_k^2 \right) \\
&+ \sum_{i+j \leq \hat{\ell}-1} D_1(h_i, h_j) h_i^2 h_j^2 + \sum_{i+k \leq \hat{\ell}-1} D_2(h_i, h_k) h_i^2 h_k^2 + \sum_{j+k \leq \hat{\ell}-1} D_3(h_j, h_k) h_j^2 h_k^2 \\
&+ \sum_{i+j+k=\hat{\ell}+2} E(h_i, h_j, h_k) h_i^2 h_j^2 h_k^2 - 2 \sum_{i+j+k=\hat{\ell}+1} E(h_i, h_j, h_k) h_i^2 h_j^2 h_k^2 \\
&+ \sum_{i+j+k=\hat{\ell}} E(h_i, h_j, h_k) h_i^2 h_j^2 h_k^2 \\
&= C_1(h_{\hat{\ell}}) h_{\hat{\ell}}^2 + C_2(h_{\hat{\ell}}) h_{\hat{\ell}}^2 + C_2(h_{\hat{\ell}}) h_{\hat{\ell}}^2 \\
&+ \left(\frac{1}{4} \sum_{i+j=\hat{\ell}+1} D_1(h_i, h_j) - \sum_{i+j=\hat{\ell}} D_1(h_i, h_j) \right) h_{\hat{\ell}}^2 \\
&+ \left(\frac{1}{4} \sum_{i+j=\hat{\ell}+1} D_2(h_i, h_j) - \sum_{i+j=\hat{\ell}} D_2(h_i, h_j) \right) h_{\hat{\ell}}^2 \\
&+ \left(\frac{1}{4} \sum_{i+j=\hat{\ell}+1} D_3(h_i, h_j) - \sum_{i+j=\hat{\ell}} D_3(h_i, h_j) \right) h_{\hat{\ell}}^2 \\
&+ \sum_{i+j+k=\hat{\ell}+2} E(h_i, h_j, h_k) h_i^2 h_j^2 h_k^2 - 2 \sum_{i+j+k=\hat{\ell}+1} E(h_i, h_j, h_k) h_i^2 h_j^2 h_k^2 \\
&+ \sum_{i+j+k=\hat{\ell}} E(h_i, h_j, h_k) h_i^2 h_j^2 h_k
\end{aligned}
$$

Somit folgt nach Voraussetzung und wegen $\sum_{i+j+k=\hat{\ell}+2} |E(h_i, h_j, h_k)| h_i^2 h_j^2 h_k^2 \leq \kappa \left(\sum_{i=1}^{\hat{\ell}} i \right) h_{\hat{\ell}+2}^2 =$

$\kappa\frac{\hat{\ell}(\hat{\ell}+1)}{2}h^2_{\hat{\ell}+2}$:

$$\begin{aligned}
|u-\hat{u}^c_{\hat{\ell}}| &\leq 3\kappa h^2_{\hat{\ell}} + 3\kappa\left(\frac{1}{4}\hat{\ell}+\hat{\ell}-1\right)h^2_{\hat{\ell}} \\
&+ \kappa\left(\frac{1}{16}\frac{\hat{\ell}(\hat{\ell}+1)}{2}+\frac{2}{4}\frac{\hat{\ell}(\hat{\ell}-1)}{2}+\frac{\hat{\ell}(\hat{\ell}-2)(\hat{\ell}-1)}{2}\right)h^2_{\hat{\ell}} \\
&= \left(1+\frac{65}{32}\hat{\ell}+\frac{25}{32}\hat{\ell}^2\right)\kappa h^2_{\hat{\ell}} \\
&= \left(1+\frac{65}{32}\log\left(h_{\hat{\ell}}\right)^{-1}+\frac{25}{32}\left(\log\left(h_{\hat{\ell}}\right)^{-1}\right)^2\right)\kappa h^2_{\hat{\ell}} = \mathcal{O}\left(h^2_{\hat{\ell}}\left(\log(h_{\hat{\ell}})^{-1}\right)^2\right).
\end{aligned}$$

□

H.4 Konvergenz des DGAFO-Verfahrens bei differenzierbarer Fehlerfunktion

Im folgenden wird der Konvergenzbeweis des DGAFO-Vefahrens aus Abschnitt 6.4.3 für den differenzierbaren Fall erweitert. Es sei F mindestens C^2, und für die Hesse-Matrizen D^2F und H_k gelte $||D^2F|| \leq \kappa_{HF}$ beziehungsweise $H_k \leq \kappa_{HG}$ mit $\kappa_{HF}, \kappa_{HG} > 0$, wobei $||\cdot||$ beispielsweise die Frobenius-Norm ist. Es sei weiterhin $\kappa_{\max} := \max(\kappa_{HF}, \kappa_{HG})$. Es ist dann zu zeigen, daß die durch das DGAFO-Verfahren definierte Folge $(x^k)_{k\in\mathbb{N}}$ gegen einen stationären Punkt von F konvergiert, genauer, daß für jeden Häufungspunkt $x^\star$ dieser Folge gilt: $\nabla F(x^\star) = 0$. Ist zusätzlich $D^2F(x^\star)$ spd, so handelt es sich bei $x^\star$ um ein lokales Minimum von F. Hierzu ist zunächst eine Abschätzung bezüglich der Differenz zwischen $\nabla F(x^k)$ und der Approximation g^k erforderlich:

Lemma H.4.1 (Abschätzung für $||\nabla F(x^k) - g^k||$)**.** *Es gilt*

$$\exists_{\tilde{\kappa}_g>0}\ \forall_{k\in\mathbb{N}}\ ||\nabla F(x^k) - g^k|| \leq \tilde{\kappa}_g f^g(\Delta_k), \tag{H.8}$$

wobei $f^g(\Delta_k) := \tilde{f}^\mu(\Delta_k) + \Delta_k + \Delta_k^2 \overset{\Delta_k\to 0}{\to} 0$.

Beweis:
1. Fall: $g_k = \nabla F(x^k)$. Dann ist nichts zu zeigen.
2. Fall: $g_k \neq \nabla F(x^k)$. Aus einer Taylor-Entwicklung für F folgt:

$$\forall_{h,||h||\leq\Delta_k}\ F(x^k+h) = F(x^k) + \langle\nabla F(x^k), h\rangle + \frac{1}{2}\langle h, D^2F(\xi^k)h\rangle, \tag{H.9}$$

wobei $\xi^k := \alpha_k h$ und $\alpha_k \in (0,1)$. Analog gilt für G:

$$\forall_{h,||h||\leq\Delta_k}\ G(x^k+h) = G(x^k) + \langle g^k, h\rangle + \frac{1}{2}\langle h, H_k h\rangle + \mathcal{O}\left(||h||^3\right). \tag{H.10}$$

Subtrahiert man Gleichung (H.10) von Gleichung (H.9), so erhält man:

$$\begin{aligned} F(x^k+h)-G(x^k+h)-\Big(F(x^k)-G(x^k)\Big) &= \langle \nabla F(x^k)-g^k,h\rangle + \frac{1}{2}\left\langle h,\left(D^2F(\xi^k)-H_k\right)h\right\rangle \\ &+ \mathcal{O}\left(||h||^3\right) \\ \Leftrightarrow\ \langle \nabla F(x^k)-g^k,h\rangle &= F(x^k+h)-G(x^k+h)-\Big(F(x^k)-G(x^k)\Big) \\ &- \frac{1}{2}\left\langle h,\left(D^2F(\xi^k)-H_k\right)h\right\rangle + \mathcal{O}\left(||h||^3\right). \end{aligned}$$

Somit folgt wegen $B_{\Delta_k}(x^k)\subset \mathcal{W}^{\Delta_k}(x^k)$:

$$\begin{aligned} |\langle \nabla F(x^k)-g^k,h\rangle| &\overset{(6.63)}{\leq} 2\kappa_\mu f^\mu(\Delta_k)+\frac{1}{2}\left|\left\langle h,D^2F(\xi^k)h\right\rangle\right|+\frac{1}{2}|\langle h,H_kh\rangle|+C||h||^3 \\ &\overset{\text{Cauchy}-\text{Schwarz}}{\leq} 2\kappa_\mu f^\mu(\Delta_k)+\kappa_{\max}\Delta_k^2+C||h||^3 \end{aligned}$$

für ein $C>0$. Setze nun $h:=\Delta_k\frac{\nabla F(x^k)-g^k}{||\nabla F(x^k)-g^k||}$ (Man beachte: $g_k\neq\nabla F(x^k)$). Dann ist $||h||=\Delta_k$, und es gilt:

$$\begin{aligned} ||\nabla F(x^k)-g^k|| &\leq 2\frac{\kappa_\mu}{\Delta_k}f^\mu(\Delta_k)+\kappa_{\max}\Delta_k+C\Delta_k^2 \\ &= 2\kappa_\mu\tilde{f}^\mu(\Delta_k)+\kappa_{\max}\Delta_k+C\Delta_k^2 \\ &\leq \tilde{\kappa}_g f^g(\Delta_k), \end{aligned}$$

wobei $\tilde{\kappa}_g:=\max(2\kappa_\mu,\kappa_{\max},C)$. □

Nun kann die eigentliche Konvergenz des DGAFO-Verfahrens im Falle eines glatten F bewiesen werden:

Satz H.4.2 (Konvergenz für endlich viele erfolgreiche Iterationen, glatter Fall)**.** *Es seien sämtliche Voraussetzungen von Satz 6.4.11 erfüllt. Falls es nur endlich viele erfolgreiche Iterationen $k^\star$ gibt, und ist $x^{k^\star}=x^\star$, so folgt $\nabla F(x^\star)=0$.*

Beweis: (durch Widerspruch). Annahme: $\nabla F(x^\star)\neq 0$. Es gilt $\forall_{j\in\mathbb{N}}\ x^{k^\star+j}=x^{k^\star}=x^\star$ sowie $\lim_{j\to\infty}\Delta_{k^\star+j}=0$. Nach Lemma H.4.1 gilt:

$$\begin{aligned} \forall_{j\in\mathbb{N}}\ ||\nabla F(x^\star)-g^{k^\star+j}|| &\leq \tilde{\kappa}_g f^g(\Delta_{k^\star+j}) \overset{j\to\infty}{\to} 0 \\ \Rightarrow\ \exists_{j_0}\ \forall_{j\geq j_0}\ ||g^{k^\star+j}|| &\geq \frac{1}{2}||\nabla F(x^\star)|| =: \kappa_g > 0. \end{aligned}$$

Nach Lemma 6.4.10 folgt dann:

$$\exists_{\kappa_\Delta>0}\ \forall_{j\geq j_0}\ \exists_{j^\star}\ \Delta_{k^\star+j,j^\star}>\kappa_\Delta,$$

das heißt, es gibt noch unendlich viele zusätzliche erfolgreiche Iterationen. Es gilt zum Beispiel $x^{k^\star+j_0+1}\neq x^\star$ ↯.
Damit ist die Annahme falsch, und es gilt $\nabla F(x^\star)=0$. □

Lemma H.4.3. *Es seien die Voraussetzungen von Korollar 6.4.7 und Lemma 6.4.9 für alle $k, l \in \mathbb{N}$ erfüllt, und es gebe unendlich viele erfolgreiche Iterationen. Weiterhin sei $(k_i)_{i\in\mathbb{N}}$ gemäß Theorem 6.4.12 eine Teilfolge mit $\lim_{i\to\infty} ||g^{k_i}|| = 0$. Dann gilt:*

$$\lim_{i\to\infty} ||\nabla F(x^{k_i})|| = 0. \tag{H.11}$$

Beweis: Aufgrund von Lemma H.4.1 gilt für alle $i \in \mathbb{N}$:

$$\begin{aligned} ||\nabla F(x^{k_i}) - g^{k_i}|| &\leq \tilde{\kappa}_g f^g(\Delta_{k_i}) \\ \Rightarrow\ ||\nabla F(x^{k_i})|| &\leq ||g^{k_i}|| + ||\nabla F(x^{k_i}) - g^{k_i}|| \\ &\leq ||g^{k_i}|| + \tilde{\kappa}_g f^g(\Delta_{k_i}). \end{aligned}$$

Nach Voraussetzung gilt $\lim_{i\to\infty} ||g^{k_i}|| = 0$. Da f^g stetig ist, gilt $\lim_{i\to\infty} f^g(\Delta_{k_i}) = f^g\left(\lim_{i\to\infty} \Delta_{k_i}\right) = f^g(0) = 0$. Und somit gilt auch für die linke Seite:

$$\lim_{i\to\infty} ||\nabla F(x^{k_i})|| = 0.$$

□

Satz H.4.4 (Konvergenz des DGAFO-Verfahrens, glatter Fall)**.** *Für jeden Häufungspunkt $x^\star$ der durch das DGAFO-Verfahren definierten Folge $(x^k)_{k\in\mathbb{N}}$ gilt:*

$$\nabla F(x^\star) = 0. \tag{H.12}$$

Beweis: (durch Widerspruch). Annahme: Es existiert eine Teilfolge $(t_i)_{i\in\mathbb{N}}$ erfolgreicher Iterationen, so daß

$$\exists_{\epsilon_0>0}\ \forall_{i\in\mathbb{N}}\ ||\nabla F(x_{t_i})|| \geq \epsilon_0. \tag{H.13}$$

Aufgrund von Lemma H.4.3 existiert dann ein $\epsilon > 0$, so daß für i groß genug gilt:

$$||g^{t_i}|| \geq 2\epsilon > \epsilon. \tag{H.14}$$

Es gelte oBdA $(2 + \tilde{\kappa}_g)\epsilon \leq \frac{1}{2}\epsilon_0$, ansonsten muß ϵ verkleinert, das heißt i vergrößert werden. Nach Theorem 6.4.12 gilt $\liminf_{k\to\infty} ||g^k|| = 0$. Daher existiert für alle t_i eine erste erfolgreiche Iteration $\tilde{t}_i > t_i$ mit $||g^{\tilde{t}_i}|| < \epsilon$ und $\forall_{t_i \leq \bar{t}_i < \tilde{t}_i}\ ||g^{\bar{t}_i}|| \geq \epsilon$. Es sei nun

$$\mathcal{T} := \{\bar{t}_i \in \mathcal{S} | t_i \leq \bar{t}_i \leq \tilde{t}_i\},$$

wobei $\mathcal{S}$ die Menge aller erfolgreicher Iterationen sei. Es folgt dann:

$$\begin{aligned} \forall_{\bar{t}_i\in\mathcal{T}}\ F(x^{\bar{t}_i}) - F(x^{\bar{t}_i+1}) &\geq \eta_1 \left(q_{\bar{t}_i}(0) - q_{\bar{t}_i}(d^{\bar{t}_i})\right) \\ &\overset{\text{Lemma 6.4.9}}{\geq} \kappa_q ||g^{\bar{t}_i}|| \Delta_{\bar{t}_i} \\ &\overset{\text{(H.14)}}{\geq} \kappa_q \epsilon \Delta_{\bar{t}_i}. \end{aligned}$$

Somit gilt aufgrund einer Teleskopsummation:

$$\sum_{j=t_i,\ j\in\mathcal{T}}^{\tilde{t}_i-1} \Delta_j \leq \frac{1}{\kappa_q \epsilon} \left(F(x^{t_i}) - F(x^{\tilde{t}_i}) \right). \tag{H.15}$$

Da die Folge $\left(F(x^{\bar{t}_i})\right)_{\bar{t}_i\in\mathcal{T}}$ monoton fallend und nach unten beschränkt ist, ist sie konvergent, und die rechte Seite von Gleichung (H.15) konvergiert für $i \to \infty$ gegen 0. Aus Stetigkeitsgründen folgt:

$$||\nabla F(x^{t_i}) - \nabla F(x^{\tilde{t}_i})|| \leq \epsilon$$

für i groß genug. Somit gilt auch, für i groß genug:

$$\begin{aligned} ||\nabla F(x^{t_i})|| &\leq ||\nabla F(x^{t_i}) - \nabla F(x^{\tilde{t}_i})|| + ||\nabla F(x^{\tilde{t}_i}) - g_{\tilde{t}_i}|| + ||g_{\tilde{t}_i}|| \\ &\overset{(H.8)}{\leq} \epsilon + \tilde{\kappa}_g f^g(\Delta_{\tilde{t}_i}) + \epsilon \\ &\leq (2 + \tilde{\kappa}_g)\epsilon \leq \frac{1}{2}\epsilon_0. \end{aligned}$$

Dies ist ein Widerspruch zu Ungleichung (H.13). ↯
Somit ist die Annahme falsch, und es folgt die Behauptung. □

Literaturverzeichnis

[Alder u. Wainwright 1957] Alder, B. J. ; Wainwright, T. E.: Phase Transition for a Hard Sphere System. In: Journal of Chemical Physics 27 (1957), S. 1208–1209

[Alder u. Wainwright 1959] Alder, B. J. ; Wainwright, T. E.: Studies in Molecular Dynamics. I. General Method. In: Journal of Chemical Physics 31 (1959), S. 459–466

[Alder u. Wainwright 1960] Alder, B. J. ; Wainwright, T. E.: Studies in Molecular Dynamics. II. Behaviour of a Small Number of Elastic Spheres. In: Journal of Chemical Physics 33 (1960), S. 1439–1451

[Allen u. Tildesley 1987] Allen, M. P. ; Tildesley, D. J.: Computer Simulation of Liquids. Oxford : Oxford Science Publications, 1987

[Andersen 1980] Andersen, H. C.: Molecular Dynamics Simulations at Constant Pressure and/or Temperature. In: Journal of Chemical Physics 72 (1980), S. 2384–2393

[Andersen 1983] Andersen, H. C.: Rattle: A 'Velocity' Version of the SHAKE Algorithm for Molecular Dynamics Calculations. In: Journal of Computational Physics 52 (1983), S. 24–34

[Armstrong 1998] Armstrong, M.: Basic Linear Geostatistics. Springer-Verlag, 1998

[Auger u. Hansen 2005] Auger, A. ; Hansen, N.: A Restart CMA Evolution Strategy With Increasing Population Size. In: MacKay, B. (Hrsg.): The 2005 IEEE International Congress on Evolutionary Computation (CEC'05), 2005, S. 1769–1776

[Barkla u. Pantoja 1996] Barkla, B. J. ; Pantoja, O.: Physiology of Ion Transport Across the Tonoplast of Higher Plants. In: Annual Review of Plant Physiology and Plant Molecular Biology 47 (1996), S. 159–164

[Barnes u. Gelb 2007] Barnes, B. C. ; Gelb, L. D.: Meta-Optimization of Evolutionary Strategies for Empirical Potential Development: Application to Aqueous Silicate Systems. In: Journal of Chemical Theory and Computation 3 (2007), S. 1749–1764

[Batra u. a. 1975] Batra, I. P. ; Bennett, B. I. ; Herman, F.: Simple Molecular Model for Crystalline Tetrathiofulvalene-Tetracyanoquinodimethane (TTF-TCNQ). In: Physical Review B 11 (1975), S. 4972–4934

[Bäuerle u. a. 2004] Bäuerle, C. ; Thole, C.-A. ; Trottenberg, U.: DesParO - A Design Parameter Optimisation Toolbox using an Iterative Kriging Algorithm. In: ERCIM News 56 (2004), S. 32–33

[Baxter 1982] Baxter, R. J.: Exact Solved Models in Statistical Mechanics. Academic Press, London, 1982

[Becker u. a. 1988] Becker, R. A. ; Chambers, J. M. ; Wilks, A. R.: The New S Language—A Programming Environment for Data Analysis and Graphics. Wadsworth & Brooks: Cole Advanced Books & Software, 1988

[Beeman 1976] Beeman, D.: Some Multiple Methods for Use in Molecular Dynamics Calculations. In: Journal of Computational Physics 20 (1976), S. 130–139

[Berendsen u. van Gunsteren 1987] Berendsen, H. J. ; Gunsteren, W. F.: GROMOS87 Manual, 1987. – Handbuch, Biomos, AG Groningen.

[Berendsen u. a. 1984] Berendsen, H. J. C. ; Postma, J. P. M. ; Gunsteren, W. F. ; di Nola, A. ; Haak, J. R.: Molecular Dynamics with Coupling to an External Bath. In: Journal of Chemical Physics 81 (1984), S. 3684–3690

[Bernal u. King 1968] Bernal, J. D. ; King, S. V.: Experimental Studies of a Simple Liquid Model. In: Temperley, H. N. V. (Hrsg.) ; Rowlinson, J. S. (Hrsg.) ; Rushbrooke, G. S. (Hrsg.): Physics of simple liquids, 1968, S. 231–252

[Berthelot 1898] Berthelot, D.: Sur le mélange des gaz. In: Comptes Rendues Hebdomadaires des Séances de l'Académie des Sciences 126 (1898), S. 1703–1706

[Beyer u. Schwefel 2002] Beyer, H.-G. ; Schwefel, H.-P.: Evolution Strategies: A Comprehensive Introduction. In: Natural Computing 1 (2002), S. 3–52

[Bien u. Chiriac 2004] Bien, D. E. ; Chiriac, V. A.: A Novel Molecular Approach to Modeling Phase Change in Micro-Fluidic Systems. In: Proceedings of the 9th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems Bd. 2, IEEE, 2004, S. 748–758

[Binder 1995] Binder, K.: Monte Carlo and Molecular Dynamics Simulations in Polymer Science. Oxford University Press, 1995

[Bishop 2007] Bishop, C. M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, 2007

[Bordat u. a. 2001] Bordat, P. ; Reith, D. ; Müller-Plathe, F.: The Influence of Interaction Details on the Thermal Diffusion in Binary Lennard-Jones Liquids. In: Journal of Chemical Physics 115 (2001), S. 8978–8982

[Born u. Oppenheimer 1927] Born, M. ; Oppenheimer, R.: Zur Quantentheorie der Molekeln. In: Annalen der Physik 84 (1927), S. 457–484

[Bourasseau u. a. 2003] Bourasseau, E. ; Haboudou, M. ; Boutin, A. ; Fuchs, A. H. ; Ungerer, P.: New Optimization Method for Intermolecular Potentials: Optimization of a New Anisotropic United Atoms Potential for Olefins: Prediction of Equilibrium Properties. In: Journal of Chemical Physics 118 (2003), S. 3020–3034

[Breneman u. Wiberg 1990] Breneman, C. M. ; Wiberg, K. B.: Determining Atom-Centered Monopoles from Molecular Electrostatic Potentials. The Need for High Sampling Density in Formamide Conformational Analysis. In: Journal of Computational Chemistry 11 (1990), S. 361–373

[Brenner 1990] Brenner, D. W.: Empirical Potential for Hydrocarbons for Use in Simulating the Chemical Vapor Deposition of Diamond Films. In: Physical Review B 42 (1990), S. 9458–9471

[Brooks u. a. 1983] Brooks, B. R. ; Bruccoleri, R. E. ; Olafson, B. D. ; States, D. J. ; Swaminathan, S. ; Karplus, M.: CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. In: Journal of Computational Chemistry 4 (1983), S. 187–217

[Brühl u. a. 2009] Brühl, B. ; Hülsmann, M. ; Borscheid, D. ; Friedrich, C. M. ; Reith, D.: A Sales Forecast Model for the German Automobile Market Based on Time Series Analysis and Data Mining Methods. In: Perner, P. (Hrsg.): Proceedings of the 9th Industrial Conference on Data Mining (ICDM), Advances in Data Mining : Applications and Theoretical Aspects, Springer-Verlag, 2009, S. 146–160

[Buckingham 1959] Buckingham, A. D.: Molecular Quadrupole Moments. In: Quarterly Review of Biophysics 13 (1959), S. 183–214

[Buckingham 1938] Buckingham, R. A.: The Classical Equation of State of Gaseous Helium, Neon and Argon. In: Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 168 (1938), S. 264–238

[Buckles u. a. 1999] Buckles, C. ; Chipman, P. ; Cubillas, M. ; Lakin, M. ; Slezak, D. ; Townsend, D. ; Vogel, K. ; Wagner, M.: Ethylene Oxide User's Guide, 1999. – `http://www.ethyleneoxide.com`, zuletzt besucht am 28. November 2011 (Paßwort erforderlich).

[Bungartz 1992] Bungartz, H.-J.: An Adaptive Poisson Solver Using Hierarchical Bases and Sparse Grids in Iterative Methods in Linear Algebra. In: de Groen, P. (Hrsg.) ; Beauwens, R. (Hrsg.): Proceedings of the IMACS International Symposium, Brüssel, Elsevier, 1992, S. 293–310

[Carlson u. Westrum Jr 1971] Carlson, H. G. ; Westrum Jr, E. F.: Methanol: Heat Capacity, Enthalpies of Transition and Melting, and Thermodynamic Properties from 5–300 K. In: Journal of Chemical Physics 54 (1971), S. 1464–1471

[Case u. a. 2008] Case, F. H. ; Brennan, J. ; Chaka, A. ; Dobbs, K. D. ; Friend, D. G. ; Gordon, P. A. ; Moore, J. D. ; Mountain, R. D. ; Olson, J. D. ; Ross, R. B. ; Schiller, M. ; Shen, V. K. ; Stahlberg, E. A.: The Fourth Industrial Fluid Properties Simulation Challenge. In: Fluid Phase Equilibria 274 (2008), S. 2–9

[Chambers u. Hastie 1992] Chambers, J. M. ; Hastie, T. J.: Statistical Models in S. Wadsworth & Brooks: Cole Advanced Books & Software, 1992

[Chang u. Lin 2011] Chang, C.-C. ; Lin, C.-J.: LIBSVM: A Library for Support Vector Machines. In: ACM Transactions on Intelligent Systems and Technology 2 (2011), S. 1–27. – Software verfügbar unter `http://www.csi.ntu.edu.tw/~cjlin/libsvm/`, zuletzt besucht am 28. November 2011.

[Chen u. a. 2001a] Chen, B. ; Potoff, J. J. ; Siepmann, J. I.: Monte Carlo Calculations for Alcohols and their Mixtures with Alkanes. Transferable Potentials for Phase Equilibria.

5. United-Atom Description of Primary, Secondary and Tertiary Alcohols. In: Journal of Physical Chemistry B 105 (2001), S. 3093–3104

[Chen u. a. 2001b] Chen, I. J. ; Yin, D. ; MacKerell Jr., A. D.: Combined *Ab initio*/Empirical Approach for Optimization of Lennard–Jones Parameters for Polar–Neutral Compunds. In: Journal of Computational Chemistry 23 (2001), S. 199–213

[Clementi 2008] Clementi, C.: Coarse-Grained Models of Protein Folding: Toy Models or Predictive Tools? In: Current Opinion in Structural Biology 18 (2008), S. 10–15

[CMA-ES Javadoc 2011] CMA-ES Javadoc: Java Documentation of CMA-ES, 2011. – Online-Dokumentation, `http://www.lri.fr/~hansen/javadoc/index.html`, zuletzt besucht am 21. November 2011.

[Coester u. Kümmel 1960] Coester, F. ; Kümmel, H.: Short-range Correlations in Nuclear Wave Functions. In: Nuclear Physics 17 (1960), S. 477–485

[Conn u. a. 1997] Conn, A. R. ; Scheinberg, K. ; Toint, P. L.: On the Convergence of Derivative-free Methods for Unconstrained Optimization. In: Iserles, A. (Hrsg.) ; Buhmann, M. (Hrsg.): Approximation Theory and Optimization: Tributes to M. J. D. Powell. Cambridge University Press, 1997, S. 83–108

[Coulomb 1788] Coulomb, C. A.: Premier Mémoir sur l'Electricité et le Magnétisme. In: Mémoires de l'Académie Royale des Sciences Pour l'Année 1785 (1788), S. 569–577

[CRAN 2011] CRAN: Contributed R Archive Network, 2011. – `http://cran.r-project.org/web/packages/`, zuletzt besucht am 28. November 2011.

[Davies 1945] Davies, C. N.: The Density and Thermal Expansion of Liquid Phosgene. In: Journal of Chemical Physics 14 (1945), S. 48–49

[Della u. Dongwei 2008] Della, C. N. ; Dongwei, S.: Mechanical Properties of Carbon Nanotubes Reinforced Ultra High Molecular Weight Polyethylene. In: Diffusion and defect data. Solid state data. Part B, Solid state phenomena 136 (2008), S. 45–48

[Deshpande u. Pandya 1967] Deshpande, D. D. ; Pandya, M. V.: Thermodynamics of Binary Solutions. Part 2. Vapour Pressures and Excess Free Energies of Aniline Solutions. In: Transactions of the Faraday Society 63 (1967), S. 2149–2157

[Deublein u. a. 2011] Deublein, S. ; Eckl, B. ; Stoll, J. ; Lishchuk, S. V. ; Guevara-Carrion, G. ; Glass, C. W. ; Merker, T. ; Bernreuther, M. ; Hasse, H. ; Vrabec, J.: ms2: A Molecular Simulation Tool for Thermodynamic Properties. In: Computer Physics Communications 182 (2011), S. 2350–2367. – Software verfügbar unter `http://www.ms-2.de/`, zuletzt besucht am 21. November 2011.

[Dimitriadou u. a. 2011] Dimitriadou, E. ; Hornik, K. ; Leisch, F. ; Meyer, D. ; A., Weingessel: The Package 'e1071', 2011. – Handbuch, `http://cran.r-project.org/web/packages/e1071/`, zuletzt besucht am 28. November 2011.

[DOW 1995] DOW: The Dow Chemical Company, 1995. – `http://www.dow.com/`, zuletzt besucht am 28. November 2011.

[Duan u. a. 2003] Duan, Y. ; Wu, C. ; Chowdhury, S. ; Lee, M. C. ; Xiong, G. ; Zhang, W. ; Yang, R. ; Cieplak, P. ; Luo, R. ; Lee, T. ; Caldwell, J. ; Wang, J. ; Kollman, P.: A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations. In: Journal of Computational Chemistry 24 (2003), S. 1999–2012

[Eastwood u. a. 1980] Eastwood, J. W. ; Hockney, R. W. ; Lawrence, D.: P3M3DP—the Three Dimensional Periodic Particle–Particle/Particle–Mesh Program. In: Computer Physics Communications 19 (1980), S. 215–261

[Eckl u. a. 2008a] Eckl, B. ; Vrabec, J. ; Hasse, H.: On the Application of Force Fields for Predicting a Wide Variety of Properties: Ethylene Oxide as an Example. In: Fluid Phase Equilibria 274 (2008), S. 16–26

[Eckl u. a. 2008b] Eckl, B. ; Vrabec, J. ; Hasse, H.: A Set of Molecular Models Based on Quantum Mechanical ab initio Calculations and Thermodynamic Data. In: Journal of Physical Chemistry B 112 (2008), S. 12710–12721

[Edberg u. a. 1986] Edberg, R. ; Evans, D. J. ; Morriss, G. P.: Constrained Molecular-Dynamics Simulations of Liquid Alkanes with a New Algorithm. In: Journal of Chemical Physics 84 (1986), S. 6933–6939

[Elliott 2011] Elliott, J. R., persönliche Kommunikation, 2011.

[Endres u. El Abedin 2006] Endres, F. ; El Abedin, S. Z.: Air and Water Stable Ionic Liquids in Physical Chemistry. In: Physical Chemistry Chemical Physics 8 (2006), S. 2101–2116

[Engin u. a. 2011] Engin, C. ; Vrabec, J. ; Hasse, H.: On the Difference between a Point Multipole and an Equivalent Linear Arrangement of Point Charges in Force Field Models for Vapor–Liquid Equilibria; Partial Charge Based Models for 59 Real Fluids. In: Molecular Physics 109 (2011), S. 1975–1982

[Esteve u. a. 2003] Esteve, X. ; Conesa, A. ; Coronas, A.: Liquid Densities, Kinematic Viscosities, and Heat Capacities of Some Alkylene Glycol Dialkyl Ethers. In: Journal of Chemical and Engineering Data 48 (2003), S. 392–397

[Evans 1983] Evans, D. J.: Computer Experiment for Nonlinear Thermodynamics of Couette Flow. In: Journal of Chemical Physics 78 (1983), S. 3297–3302

[Evans u. Morriss 1983] Evans, D. J. ; Morriss, G. P.: Isothermal Isobaric Molecular Dynamics Ensemble. In: Chemical Physics 77 (1983), S. 63–66

[Evans u. Morriss 1984] Evans, D. J. ; Morriss, G. P.: Non-Newtonian Molecular Dynamics. In: Computational Physics Reports 1 (1984), S. 297–344

[Ewald 1921] Ewald, P.: Die Berechnung optischer und elektrostatischer Gitterpotentiale. In: Annalen der Physik 64 (1921), S. 253–287

[Faller 2004] Faller, R.: Automatic Coarse Graining of Polymers. In: Polymer 45 (2004), S. 3869–3876

[Faller u. a. 1999] Faller, R. ; Schmitz, H. ; Biermann, O. ; Müller-Plathe, F.: Automatic Parameterization of Force Fields Liquids by Simplex Optimization. In: Journal of Computational Chemistry 20 (1999), S. 1009–1017

[Fan 2002] Fan, E.: Global Optimization of Lennard-Jones Atomic Clusters, McMaster University, Diplomarbeit, 2002

[Fehlner 2000] Fehlner, T. P.: Molecular Models of Solid State Metal Boride Structures. In: Journal of Solid State Chemistry 154 (2000), S. 110–113

[Ferguson u. Kollman 1991] Ferguson, D. M. ; Kollman, P. A.: Can the Lennard-Jones 6-12 Function Replace the 10-12 Form in Molecular Mechanics Calculations? In: Journal of Computational Chemistry 12 (1991), S. 620–626

[Foloppe u. MacKerell 2000] Foloppe, N. ; MacKerell, A. D.: All-Atom Empirical Force Field for Nucleic Acids: 1) Parameter Optimization Based on Small Molecule and Condensed Phase Macromolecular Target Data. In: Journal of Computational Chemistry 21 (2000), S. 86–104

[Forester u. Smith 1998] Forester, T. R. ; Smith, W.: SHAKE, Rattle, and Roll: Efficient Constraint Algorithms for Linked Rigid Bodies. In: Journal of Computational Chemistry 19 (1998), S. 102–111

[Franck u. Deul 1978] Franck, E. U. ; Deul, R.: Dielectric Behaviour of Methanol and Related Polar Fluids at High Pressures and Temperatures. In: Faraday Discussions of the Chemical Society 66 (1978), S. 191–198

[Freindorf u. a. 2005] Freindorf, M. ; Shao, Y. ; Furlani, T. R. ; Kong, J.: Lennard-Jones Parameters for the Combined QM/MM Method Using the B3LYP/G-31+G*/AMBER Potential. In: Journal of Computational Chemistry (2005), S. 1270–1278

[Frenkel u. Smit 2006] Frenkel, D. ; Smit, B.: Understanding Molecular Simulation: From Algorithms to Applications. Academic Press, 2006

[Friedman 1991] Friedman, J. H.: Multivariate Adaptive Regression Splines. In: The Annals of Statistics 19 (1991), S. 1–67

[Frisch u. a. 2004] Frisch, M. J. ; Trucks, G. W. ; Schlegel, H. B. ; Scuseria, G. E. ; Robb, M. A. ; Cheeseman, J. R. ; Montgomery, J. A. Jr. ; Vreven, T. ; Kudin, K. N. ; Burant, J. C. ; Millam, J. M. ; Iyengar, S. S. ; Tomasi, J. ; Barone, V. ; Mennucci, B. ; Cossi, M. ; Scalmani, G. ; Rega, N. ; Petersson, G. A. ; Nakatsuji, H. ; Hada, M. ; Ehara, M. ; Toyota, K. ; Fukuda, R. ; Hasegawa, J. ; Ishida, M. ; Nakajima, T. ; Honda, Y. ; Kitao, O. ; Nakai, H. ; Klene, M. ; Li, X. ; Knox, J. E. ; Hratchian, H. P. ; Cross, J. B. ; Bakken, V. ; Adamo, C. ; Jaramillo, J. ; Gomperts, R. ; Stratmann, R. E. ; Yazyev, O. ; Austin, A. J. ; Cammi, R. ; Pomelli, C. ; Ochterski, J. W. ; Ayala, P. Y. ; Morokuma, K. ; Voth, G. A. ; Salvador, P. ; Dannenberg, J. J. ; Zakrzewski, V. G. ; Dapprich, S. ; Daniels, A. D. ; Strain, M. C. ; Farkas, O. ; Malick, D. K. ; Rabuck, A. D. ; Raghavachari, K. ; Foresman, J. B. ; Ortiz, J. V. ; Cui, Q. ; Baboul, A. G. ; Clifford, S. ; Cioslowski, J. ; Stefanov, B. B. ; Liu, G. ; Liashenko, A. ;

Piskorz, P. ; Komaromi, I. ; Martin, R. L. ; Fox, D. J. ; Keith, T. ; Al-Laham, M. A. ; Peng, C. Y. ; Nanayakkara, A. ; Challacombe, M. ; Gill, P. M. W. ; Johnson, B. ; Chen, W. ; Wong, M. W. ; Gonzalez, C. ; Pople, J. A.: Gaussian 03, 2004. – Gaussian, Inc., Wallingford, CT, Software verfügbar unter `http://www.gaussian.com/`, zuletzt besucht am 5. Dezember 2011.

[Fukunaga u. a. 2002] Fukunaga, H. ; Takimoto, J. ; Doi, M.: A Coarse-Graining Procedure for Flexible Polymer Chains with Bonded and Nonbonded Interactions. In: Journal of Chemical Physics 116 (2002), S. 8183–8190

[Gear 1966] Gear, C. W.: The Numerical Integration of Ordinary Differential Equations of Various Orders / Argonne National Laboratory. 1966. – Forschungsbericht

[Gear 1971] Gear, C. W.: Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall, 1971

[Geiger u. Kanzow 1999] Geiger, C. ; Kanzow, C.: Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben. Springer-Verlag, 1999

[Gentleman 2010] Gentleman, R.: The Package 'panel', 2010. – Handbuch Version 1.0.7, `http://cran.r-project.org/web/packages/panel/`, zuletzt besucht am 28. November 2011.

[Giauque u. Jones 1948] Giauque, W. F. ; Jones, W. M.: Carbonyl Chloride. Entropy. Heat Capacity. Vapor Pressure. Heats of Fusion and Vaporization. Comments on Solid Sulfur Dioxide Structure. In: Journal of the American Chemical Society 70 (1948), S. 120–124

[Gillespie u. Robinson 2005] Gillespie, R. J. ; Robinson, E. A.: Models of molecular geometry. In: Chemical Society Reviews 34 (2005), S. 396–407

[Goldberg 1989] Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison–Wesley, 1989

[Gonnet 2007] Gonnet, P.: P-SHAKE: A Quadratically Convergent SHAKE in $\mathcal{O}(n^2)$. In: Journal of Computational Physics 220 (2007), S. 740–750

[Grest u. Kremer 1986] Grest, G. S. ; Kremer, K.: Molecular Dynamics Simulation for Polymers in the Presence of a Heat Bath. In: Physical Review A 33 (1986), S. 3628–3631

[Griebel u. a. 1990] Griebel, M. ; Schneider, M. ; Zenger, C.: A Combination Technique for the Solution of Sparse Grid Problems. 1990. – Forschungsbericht

[Gsponer u. Caflisch 2002] Gsponer, J. ; Caflisch, A.: Molecular Dynamics Simulations of Protein Folding from the Transition State. In: Proceedings of the National Academy of Sciences (PNAS) Bd. 99, 2002, S. 6719–6724

[Guest u. a. 2005] Guest, M. F. ; Bush, I. J. ; van Dam, H. J. J. ; Sherwood, P. ; Thomas, J. M. H. ; van Lenthe, J. H. ; Havenith, R. W. A. ; Kendrick, J.: The GÅMESS-UK Electronic Structure Package: Algorithms, Developments and Applications. In: Molecular Physics 103 (2005), S. 719–747

[Guevara-Carrion u. a. 2012] Guevara-Carrion, G. ; Hasse, H. ; Vrabec, J.: Thermodynamic Properties for Applications in Chemical Industry via Classical Force Fields. In: Kirschner, B. (Hrsg.) ; Vrabec, J. (Hrsg.): Topics in Current Chemistry: Multiscale Molecular Models in Applied Chemistry Bd. 307, Springer-Verlag, 2012, S. 201–249

[Guevara-Carrion u. a. 2008] Guevara-Carrion, G. ; Nieto-Draghi, C. ; Vrabec, J. ; Hasse, H.: Prediction of Transport Properties by Molecular Simulation: Methanol and Ethanol and their Mixture. In: Journal of Phyisical Chemistry B 112 (2008), S. 16664–16674

[Guggenheim 1945] Guggenheim, E. A.: The Principle of Corresponding States. In: Journal of Chemical Physics 13 (1945), S. 253–261

[Hansen 2011] Hansen, N.: The CMA Evolution Strategy: A Tutorial, 2011. – Handbuch, `http://www.lri.fr/~hansen/cmaesintro.html`, zuletzt besucht am 5. Dezember 2011.

[Hansen u. Ostermeier 1997] Hansen, N. ; Ostermeier, A.: Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The ...–CMA-ES. In: 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), 1997, S. 650–654

[Hansen u. Ostermeier 2001] Hansen, N. ; Ostermeier, A.: Completely Derandomized Self–Adaptation in Evolution Strategies. In: Evolutionary Computation 9 (2001), S. 159–195

[Harrell Jr 2010] Harrell Jr, E.: The Package 'Hmisc', 2010. – Handbuch Version 3.9-0, `http://cran.r-project.org/web/packages/Hmisc/`, zuletzt besucht am 28. November 2011.

[Hemmersbach 2011] Hemmersbach, J.: Regularisierte Approximationsverfahren für eine Dünn-Gitter-basierte, ableitungsfreie Methode zur Optimierung von Kraftfeldparametern, Universität zu Köln, Diplomarbeit, 2011

[Hess 2008] Hess, B.: P-LINCS: A Parallel Linear Constraint Solver for Molecular Simulation. In: Journal of Chemical Theory and Computation 4 (2008), S. 116–122

[Hess u. a. 1997] Hess, B. ; Bekker, H. ; Berendsen, H. C. ; Fraaije, J. G. E. M.: LINCS: A Linear Constraint Solver for Molecular Simulations. In: Journal of Computational Chemistry 18 (1997), S. 1463–1472

[Heyes 1983] Heyes, D. M.: Molecular Dynamics at Constant Pressure and Temperature. In: Chemical Physics 82 (1983), S. 285–301

[Hills Jr. u. a. 2010] Hills Jr., R. D. ; Lu, L. ; Voth, G. A.: Multiscale Coarse-Graining of the Protein Energy Landscape. In: PLoS Computational Biology 6 (2010), S. 1–12

[Ho u. Baumgärtner 1990] Ho, J.-S. ; Baumgärtner, A.: Simulations of Fluid Self-Avoiding Membranes. In: Europhysical Letters 12 (1990), S. 295–300

[Hockney 1970] Hockney, R. W.: The Potential Calculation and Some Applications. In: Methods in Computational Physics 9 (1970), S. 136–211

[Hockney u. Eastwood 1981] Hockney, R. W. ; Eastwood, J. W.: Computer Simulation using Particles. McGraw-Hill, 1981

[Hodgkin u. Huxley 1952] Hodgkin, A. ; Huxley, A.: A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. In: Journal of Physiology 117 (1952), S. 500–544

[Hoerl u. Kennard 1970] Hoerl, A. E. ; Kennard, R. W.: Ridge Regression: Biased Estimation for Nonorthogonal Problems. In: Technometrics 12 (1970), S. 55–67

[Hofmann 1992] Hofmann, K.: Parallelisierung einer dreidimensionalen Molekulardynamikaufgabe auf Multiprozessoren mit verteiltem Speicher. / Universität Erlangen-Nürnberg. 1992. – Forschungsbericht

[Hoover 1985] Hoover, W. G.: Canonical Dynamics: Equilibrium Phase-Space Distributions. In: Physical Review A 31 (1985), S. 1695–1697

[Hoover u. a. 1982] Hoover, W. G. ; Ladd, A. J. C. ; Moran, B.: High Strain Rate Plastic Flow Studied via Nonequilibrium Molecular Dynamics. In: Physical Review Letters 48 (1982), S. 1818–1820

[Hradetzky u. Lempe 2011] Hradetzky, G. ; Lempe, D. A.: MDB – Merseburger Datenbank für thermophysikalische Reinstoffdaten, 2011. – Handbuch Version 7.3, `http://www.forschung-sachsen-anhalt.de/`, zuletzt besucht am 13. Dezember 2011.

[Huang u. a. 2011] Huang, Y.-L. ; Heilig, M. ; Hasse, H. ; Vrabec, J.: Vapor–Liquid Equilibria of Hydrogen Chloride, Phosgene, Benzene, Chlorobenzene, Ortho-Dichlorobenzene, and Toluene by Molecular Simulation. In: AIChE Journal 57 (2011), S. 1043–1060

[Hülsmann 2006] Hülsmann, M.: Vergleich verschiedener kernbasierter Methoden zur Realisierung eines effizienten Multiclass-Algorithmus des maschinellen Lernens, Universität zu Köln, Diplomarbeit, 2006

[Hülsmann u. a. 2011a] Hülsmann, M. ; Borscheid, D. ; Friedrich, C. M. ; Reith, D.: General Sales Forecast Models for Automobile Markets based on Time Series Analysis and Data Mining Techniques. In: Perner, Petra (Hrsg.): Proceedings of the 11th Industrial Conference on Data Mining (ICDM), Advances in Data Mining : Applications and Theoretical Aspects, Springer-Verlag, 2011, S. 1–15

[Hülsmann u. Friedrich 2007] Hülsmann, M. ; Friedrich, C. M.: Comparison of A Novel Combined ECOC Strategy with Different Multiclass Algorithms Together with Parameter Optimization Methods. In: Perner, Petra (Hrsg.): Proceedings of the 5th International Conference on Machine Learning and Data Mining (MLDM), Machine Learning and Data Mining in Pattern Recognition, Springer-Verlag, 2007, S. 17–31

[Hülsmann u. a. 2012] Hülsmann, M. ; Hemmersbach, J. ; Reith, D.: An Efficient Derivative-Free Optimization Algorithm for the Development of Force Fields in Molecular Simulations. In: Computer Physics Communications (2012). – in Vorbereitung

[Hülsmann u. a. 2010a] Hülsmann, M. ; Köddermann, T. ; Vrabec, J. ; Reith, D.: GROW: A Gradient-based Optimization Workflow for the Automated Development of Molecular Models. In: Computer Physics Communications 181 (2010), S. 499–513

[Hülsmann u. a. 2011b] Hülsmann, M. ; Müller, T. J. ; Köddermann, T. ; Reith, D.: Automated Force Field Optimisation of Small Molecules using a Gradient-based Workflow Package. In: Molecular Simulation 36 (2011), S. 1182–1196

[Hülsmann u. a. 2010b] Hülsmann, M. ; Vrabec, J. ; Maass, A. ; Reith, D.: Assessment of Numerical Optimization Algorithms for the Development of Molecular Models. In: Computer Physics Communications 181 (2010), S. 887–905

[Humphrey u. a. 1996] Humphrey, W. ; Dalke, A. ; Schulten, K.: VMD – Visual Molecular Dynamics. In: Journal of Molecular Graphics 14 (1996), S. 33–38. – Software verfügbar unter `http://www.ks.uiuc.edu/Research/vmd/`, zuletzt besucht am 21. November 2011.

[Hünenberger u. van Gunsteren 1997] Hünenberger, P. H. ; Gunsteren, W. F.: Empirical Classical Interaction Functions for Molecular Simulation. In: Gunsteren, W. F. (Hrsg.) ; Weiner, P. K. (Hrsg.) ; Wilkinson, A. J. (Hrsg.): Computer Simulation of Biomolecular Systems—Theoretical and Experimental Applications, Kluwer Academic Publishers, 1997, S. 3–82

[Hunger u. Huttner 1999] Hunger, J. ; Huttner, G.: Optimization and Analysis of Force Field Parameters by Combination of Genetic Algorithms and Neural Networks. In: Journal of Computational Chemistry 20 (1999), S. 455–471

[IFPSC 2010] IFPSC: IFPSC (Industrial Fluid Property Simulation Challenge), 2010. – `http://www.ifpsc.com`, zuletzt besucht am 28. November 2011.

[IntelMPI 2011] IntelMPI: IntelMPI Library for Linux* OS, 2011. – Handbuch Version 4.0, `http://software.intel.com/en-us/articles/intel-mpi-library-documentation/`, zuletzt besucht am 28. November 2011.

[Java 2011] Java: The Java Tutorials, 2011. – Online-Tutorial Version SE 7, `http://docs.oracle.com/javase/tutorial/`, zuletzt besucht am 28. November 2011.

[Jensen 1999] Jensen, F.: Introduction to Computational Chemistry. John Wiley & Sons, 1999

[Ji u. a. 2007] Ji, W. R. ; Stiebing, E. ; Hradetzky, G. ; A., Lempe D.: Extrapolation of VLE data and Simultaneous Representation of Caloric and Volumetric Properties by Means of a Cubic 3-Parameter Equation of State. In: Fluid Phase Equilibria 260 (2007), S. 113–125

[Jiang u. a. 2004] Jiang, L. ; Byrd, R. H. ; Eskow, E. ; Schnabel, R. B.: A Preconditioned L-BFGS Algorithm with Application to Molecular Energy Minimization / Department of Computer Science, University of Colorado. 2004. – Forschungsbericht

[Jorgensen u. a. 1983] Jorgensen, W. L. ; Chandrasekhar, J. ; Madura, J. D. ; Impey, R. W. ; Klein, M. L.: Comparison of Simple Potential Functions for Simulating Liquid Water. In: Journal of Chemical Physics 79 (1983), S. 926–935

[Jorgensen u. a. 1984] Jorgensen, W. L. ; Madura, J. D. ; Swensen, C. J.: Optimized Intermolecular Potential Functions for Liquid Hydrocarbons. In: Journal of the American Chemical Society 106 (1984), S. 6638–6646

[Jorgensen u. a. 1996] Jorgensen, W. L. ; Maxwell, D. S. ; Tirado-Rives, J.: Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. In: Journal of the American Chemistry Society 118 (1996), S. 11225–11236

[Kaminski u. a. 2002] Kaminski, G. A. ; Stern, H. A. ; Berne, B. J. ; Friesner, R. A. ; Cao, Y. X. ; Murphy, R.B. ; Zhou, R. ; Halgren, T. A.: Development of a Polarizable Force Field For Proteins via Ab Initio Quantum Chemistry: First Generation Model and Gas Phase Tests. In: Journal of Computational Chemistry 23 (2002), S. 1515–1531

[Kirkpatrick u. a. 1983] Kirkpatrick, S. ; Gelatt, C. D. ; Vecchi, M. P.: Optimization by Simulated Annealing. In: Science 220 (1983), S. 671–680

[Kirschner 2010] Kirschner, K. N., persönliche Kommunikation, 2010.

[Kirschner u. a. 2008] Kirschner, K. N. ; Yongye, A. B. ; Tschampel, S. M. ; Gonzalez-Outeirino, J. ; Daniels, C. R. ; Foley, B. L. ; Woods, R. J.: GLYCAM06: A Generalizable Biomolecular Force Field. Carbohydrates. In: Journal of Computational Chemistry 29 (2008), S. 622–655

[Köddermann u. a. 2011] Köddermann, T. ; Kirschner, K. N. ; Vrabec, J. ; Hülsmann, M. ; Reith, D.: Liquid–Liquid Equilibria of Dipropylene Glycol Dimethyl Ether and Water by Molecular Dynamics. In: Fluid Phase Equilibria 310 (2011), S. 25–31

[Köddermann u. a. 2007] Köddermann, T. ; Paschek, D. ; Ludwig, R.: Molecular Dynamics Simulations of Ionic Liquids: A Reliable Description of Structure, Thermodynamics and Dynamics. In: Journal of Chemical Physics and Physical Chemistry 17 (2007), S. 2464–2470

[Köddermann 2008] Köddermann, Thorsten: Molekulardynamik-Simulation und FTIR-Spektroskopie: Neue Einsichten in Struktur, Dynamik und Thermodynamik Ionischer Flüssigkeiten, Technische Universität Dortmund, Diss., 2008

[Kolafa u. a. 2001] Kolafa, J. ; Nezbeda, I. ; Lisal, M.: Effect of Short- and Long-Range Forces on the Properties of Fluids. III. Dipolar and Quadrupolar Fluids. In: Molecular Physics 99 (2001), S. 1751–1764

[Kräutler u. a. 2001] Kräutler, V. ; van Gunsteren, W. F. ; Hünenberger, P. H.: A Fast SHAKE Algorithm to Solve Distance Constraint Equations for Small Molecules in Molecular Dynamics Simulations. In: Journal of Computational Chemistry 22 (2001), S. 501–508

[Kremer u. Grest 1990] Kremer, K. ; Grest, G. S.: Dynamics of Entangled Linear Polymer Melts: A Molecular–Dynamics Simulation. In: Journal of Chemical Physics 92 (1990), S. 5057–5086

[Kremer u. Müller-Plathe 2002] Kremer, K. ; Müller-Plathe, F.: Multiscale Simulation in Polymer Science. In: Molecular Simulation 28 (2002), S. 729–750

[Kriebel u. a. 1996] Kriebel, C. ; Muller, A. ; Mecke, M. ; Winkelmann, J. ; Fischer, J.: Prediction of Thermodynamic Properties for Fluid Nitrogen with Molecular Dynamics Simulations. In: International Journal of Themodynamics 17 (1996), S. 1349–1363

[Krieger u. a. 2002] Krieger, E. ; Koraimann, G. ; Vriend, G.: Increasing the Precision of Comparative Models with YASARA NOVA—a Self-Parameterizing Force Field. In: PROTEINS: Structure, Function, and Genetics 47 (2002), S. 393–402

[Kuypers 1993] Kuypers, F.: Klassische Mechanik. VCH Verlagsgesellschaft mbH, 1993

[Lambrakos u. a. 1989] Lambrakos, S. G. ; Boris, J. P. ; Oran, E. S. ; Chandrasekhar, I. ; Nagumo, M.: A Modified SHAKE Algorithm for Maintaining Rigid Bonds in Molecular Dynamics Simulations of Large Molecules. In: Journal of Computational Physics 85 (1989), S. 473–486

[Leach 2001] Leach, A. R.: Molecular Modelling: Principles and Applications. Prentice-Hall, 2001

[Lee u. a. 2005] Lee, S.-H. ; Palmo, K. ; Krimm, S.: WIGGLE: A New Constrained Molecular Dynamics Algorithm in Cartesian Coordinates. In: Journal of Computational Physics 210 (2005), S. 171–182

[Lennard-Jones 1931] Lennard-Jones, J. E.: Cohesion. In: Proceedings of the Physical Society 43 (1931), S. 461–483

[Levenberg 1944] Levenberg, K.: A Method for the Solution of Certain Problems in Least Squares. In: Quarterly of Applied Mathematics 2 (1944), S. 164–168

[Levitt 1976] Levitt, M.: A Simplified Representation of Protein Conformations for Rapid Simulation of Protein Folding. In: Journal of Molecular Biology 104 (1976), S. 59–107

[Levitt u. Warshe I 1975] Levitt, M. ; Warshe I, A.: Computer Simulation of Protein Folding. In: Nature 253 (1975), S. 694–698

[Lin u. a. 2003] Lin, S.-T. ; Blanco, M. ; Goddard III, W. A.: The Two-Phase Model for Calculating Thermodynamic Properties of Liquids from Molecular Dynamics: Validation for the Phase Diagram of Lennard-Jones Fluids. In: Journal of Chemical Physics 119 (2003), S. 11792–11805

[Liu u. a. 2011] Liu, Y. ; Tao, L. ; Lu, J. ; Xu, S. ; Maa, Q. ; Q., Duan: A Novel Force Field Parameter Optimization Method based on LSSVR for ECEPP. In: FEBS Letters 585 (2011), S. 888–892

[Lokhorst u. a. 2011] Lokhorst, J. ; Venables, B. ; Turlach, B.: The Package 'lasso2', 2011. – Handbuch Version 1.2-12, `http://cran.r-project.org/web/packages/lasso2/`, zuletzt besucht am 28. November 2011.

[Lopes u. Padua 2006] Lopes, J. N. A. C. ; Padua, A. A. H.: Using Spectroscopic Data on Imidazolium Cation Conformations to Test a Molecular Force Field for Ionic Liquids. In: Journal of Physical Chemistry B 110 (2006), S. 7485–7489

[Lorentz 1881] Lorentz, H. A.: Über die Anwendung des Satzes vom Virial in der kinetischen Theorie der Gase. In: Annalen der Physik 12 (1881), S. 127–136

[LUT 2011] LUT: Lappeenranta University of Technology, Department of Information Technology, 2011. – Graphik verfügbar unter `http://www.it.lut.fi/ip/evo/functions/node6.html`, zuletzt besucht am 21. November 2011

[Maaß 2011] Maaß, A., persönliche Kommunikation, 2011.

[Maaß u. a. 2010] Maass, A. ; Nikitina, L. ; Clees, T. ; Kirschner, K. N. ; D., Reith: Multi-objective Optimisation on the Basis of Random Models for Ethylene Oxide. In: Molecular Simulation 36 (2010), S. 1208–1218

[MacQueen 1967] MacQueen, J. B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967, S. 281–297

[Mangasarian u. a. 2004] Mangasarian, O. L. ; Rosen, J. B. ; Thompson, M. E.: Global Minimization via Piecewise-Linear Underestimation. In: Journal of Global Optimization 32 (2004), S. 1–9

[Marquardt 1963] Marquardt, D.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters. In: SIAM Journal of Applied Mathematics 11 (1963), S. 431–441

[Martin u. Siepmann 1998] Martin, M. G. ; Siepmann, J. I.: Transferable Potentials for Phase Equilibria. 1. United–Atom Description of n–Alkanes. In: Journal of Physical Chemistry B 102 (1998), S. 2567–2577

[Mazur 1999] Mazur, A. K.: Symplectic Integration of Closed Chain Rigid Body Dynamics with Internal Coordinate Equations of Motion. In: Journal of Chemical Physics 111 (1999), S. 1407–1414

[McDonald 1969] McDonald, I. R.: Monte Carlo Calculations for One- and Two-Component Fluids in the Isothermal-Isobaric Ensemble. In: Chemical Physics Letters 3 (1969), S. 241–243

[Merker u. a. 2012] Merker, T. ; Vrabec, J. ; Hasse, H.: Engineering Molecular Models: Efficient Parameterization and Cyclohexanol as Case Study, Soft Materials. In: Soft Materials 10 (2012), S. 3–24

[Methanex Corporation 2006] Methanex Corporation: Technische Informationen und Sicherheitsmerkblatt für den Umgang mit Methanol, 2006. – Handbuch, `http://www.methanex.com/`, zuletzt besucht am 16. Dezember 2011.

[Metropolis u. a. 1953] Metropolis, N. ; Rosenbluth, A.W. ; Rosenbluth, M.N. ; Teller, A.H. ; Teller, E.: Equation of State Calculations by Fast Computing Machines. In: Journal of Chemical Physics 21 (1953), S. 1087–1092

[Metropolis u. Ulam 1949] Metropolis, N. ; Ulam, S.: The Monte Carlo method. In: Journal of The American Statistical Association 44 (1949), S. 335–341

[Milborrow 2011] Milborrow, S.: The Package 'earth', 2011. – Handbuch Version 3.2-1, `http://cran.r-project.org/web/packages/earth/`, zuletzt besucht am 28. November 2011.

[Miyamoto u. Kollman 1992] Miyamoto, S. ; Kollman, P. A.: SETTLE: An Analytical Version of the SHAKE and RATTLE Algorithm for Rigid Water Models. In: Journal of Computational Chemistry 13 (1992), S. 952–962

[Møller u. Plesset 1934] Møller, C. ; Plesset, M. S.: Note on an Approximation Treatment for Many-Electron Systems. In: Physical Review 46 (1934), S. 618–622

[Möllhoff u. Sternberg 2001] Möllhoff, M. ; Sternberg, U.: Molecular Mechanics with Fluctuating Atomic Charges—A New Force Field with a Semi-Empirical Charge Calculation. In: Journal of Molecular Modeling 7 (2001), S. 90–102

[Moré 1983] Moré, J. J.: Recent Developments in Algorithms and Software for Trust Region Methods. In: Bachem, A. (Hrsg.) ; Grötschel, M. (Hrsg.) ; Korte, B. (Hrsg.): Mathematical Programming: The State of the Art, Springer-Verlag, 1983, S. 258–287

[Morrell u. Hildebrand 1936] Morrell, W. E. ; Hildebrand, J. H.: The Distribution of Molecules in a Model Liquid. In: Journal of Chemical Physics 4 (1936), S. 224–227

[Morse 1929] Morse, P. M.: Diatomic Molecules According to the Wave Mechanics. II. Vibrational Levels. In: Physical Review 34 (1929), S. 57–64

[MPI 1997] MPI: Message Passing Interface, 1997. – `http://www.mpi-forum.org/`, zuletzt besucht am 28. November 2011.

[Müller u. a. 2008] Müller, T. J. ; Roy, S. ; Zhao, W. ; Maass, A. ; Reith, D.: Economic Simplex Optimization for Broad Range Property Prediction: Strengths and Weaknesses of an Automated Approach for Tailoring of Parameters. In: Fluid Phase Equilibria 274 (2008), S. 27–35

[Müller-Plathe 1993] Müller-Plathe, F.: YASP: A Molecular Simulation Package. In: Computer Physics Communications 78 (1993), S. 77–94

[Müller-Plathe 1994] Müller-Plathe, F.: Permeation of Polymers—A Computational Approach. In: Acta Polymer 45 (1994), S. 259–293

[Müller-Plathe 2001] Müller-Plathe, F.: Computational Chemistry I, Molecular Simulation. Vorlesung, 2001

[Müller-Plathe u. Reith 1999] Müller-Plathe, F. ; Reith, D.: Cause and Effect Reversed in in Non-Equilibrium Molecular Dynamics: An Easy Route to Transport Coefficients. In: Computational and Theoretical Polymer Science 9 (1999), S. 203–209

[Murthy u. a. 1980] Murthy, C. S. ; Singer, K. ; Klein, M. L. ; McDonald, I.R.: Pairwise Additive Effective Potentials for nitrogen. In: Molecular Physics 41 (1980), S. 1387–1399

[MVAPICH 2011] MVAPICH, 2011. – Online-Dokumentation Version 1.2.7, `http://mvapich.cse.ohio-state.edu/download/mvapich/`, zuletzt besucht am 28. November 2011.

[Nelder u. Mead 1965] Nelder, J. A. ; Mead, R. A.: A Simplex Method for Function Minimization. In: Computer Journal 7 (1965), S. 308–313

[Newton 1726] Newton, Isaac: Philosophiae Naturalis Principia Mathematica, 1726. – Digitalisat 2007, Göttinger Digitalisierungszentrum.

[Nezbeda u. Kolafa 1991] Nezbeda, I. ; Kolafa, J.: A New Version of the Insertion Particle Method for Determining the Chemical Potential by Monte Carlo Simulation. In: Molecular Simulation 5 (1991), S. 391–403

[Nicolas u. a. 1979] Nicolas, J. J. ; Gubbins, K. E. ; Streett, W. B. ; Tildesley, D. J.: Equation of State for the Lennard-Jones Fluid. In: Journal of Molecular Physics 37 (1979), S. 1429–1454

[NIST 2011] NIST: NIST (National Institute of Standards and Technology) Chemistry Webbook, 2011. – `http://webbook.nist.gov/chemistry/`, zuletzt besucht am 28. November 2011.

[Nocedal u. Wright 1999] Nocedal, J. ; Wright, S. J.: Numerical Optimization. Springer-Verlag, 1999

[Norman u. Filinov 1969] Norman, G. E. ; Filinov, V. S.: Investigations of Phase Transistions by a Monte Carlo Method. In: High Temperature (USSR) 7 (1969), S. 216–222

[Nosé 1984] Nosé, S.: A Molecular Dynamics Method for Simulations in the Canonical Ensemble. In: Molecular Physics 52 (1984), S. 255–268

[O'Boyle u. a. 2011] O'Boyle, N. M. ; Banck, M. ; Craig, A. J. ; Morley, C. ; Vandermeersch, T. ; Hutchison, G. R.: Open Babel: An Open Chemical Toolbox. In: Journal of Cheminformatics 3 (2011), S. 33–58. – Software verfügbar unter `http://openbabel.org/wiki/Main_Page`, zuletzt besucht am 28. November 2011.

[Olson u. Wilson 2008] Olson, J. D. ; Wilson, L. C.: Benchmarks for the Fourth Industrial Fluid Properties Simulation Challenge. In: Fluid Phase Equilibria 274 (2008), S. 10–15

[Oostenbrink u. a. 2004] Oostenbrink, C. ; Villa, A. ; Mark, A. E. ; van Gunsteren, W. F.: A Biomolecular Force Field Based on the Free Enthalpy of Hydration and Solvation: The GROMOS Force–Field Parameter Sets 53A5 and 53A6. In: Journal of Computational Chemistry 25 (2004), S. 1656–1676

[Oracle 2011] Oracle: Oracle Technology Network, 2011. – `http://www.oracle.com/`, zuletzt besucht am 28. November 2011.

[Panagiotopoulos 1987] Panagiotopoulos, A. Z.: Direct Determination of Phase Coexistence Properties of Fluids by Monte Carlo Simulations in a New Ensemble. In: Molecular Physics 61 (1987), S. 813–826

[Panagiotopoulos u. a. 1988] Panagiotopoulos, A. Z. ; Quirke, N. ; Stapleton, M. ; Tildesley, D. J.: Phase Equilibria by Simulation of the Gibbs Ensemble. Alternative Derivation, Generalization, and Application to Mixture and Membrane Equilibria. In: Molecular Physics 63 (1988), S. 527–546

[Parrinello u. Rahman 1981] Parrinello, M. ; Rahman, A.: Polymorphic Transitions in Single Crystals: A New Molecular Dynamics Method. In: Journal of Applied Physics 52 (1981), S. 7182–7190

[Paschek u. Geiger 2003] Paschek, D. ; Geiger, A.: MOSCITO 4, Performing Molecular Dynamics Simulations, 2003. – Handbuch, `http://ganter.chemie.uni-dortmund.de/MOSCITO/moscito_man.html`, zuletzt besucht am 21. November 2011.

[Peguin u. a. 2009] Peguin, R. P. S. ; Kamath, G. ; Potoff, J. J. ; da Rocha, S. R. P.: All–Atom Force Field for the Prediction of Vapor–Liquid Equilibria and Interfacial Properties of HFA134a. In: Journal of Physical Chemistry B 113 (2009), S. 178–187

[Poling u. a. 2000] Poling, B. E. ; Prausnitz, J. M. ; O'Connell, J. P.: The Properties of Gases and Liquids. McGraw–Hill Professional, 2000

[Potter 1972] Potter, D.: Computational Physics. Wiley, 1972

[Powell 1987] Powell, M.: Radial Basis Functions for Multivariable Interpolation: A Review. In: Mason, J. C. (Hrsg.) ; Cox, M. G. (Hrsg.): Algorithms for Approximation, Clarendon Press, 1987, S. 143–167

[Praprotnik u. a. 2008a] Praprotnik, M. ; Hocevar, S. ; Hodoscek, M. ; Penca, M. ; Janezic, D.: New All-Atom Force Field for Molecular Dynamics Simulation of an AlPO4-34 Molecular Sieve. In: Journal of Computational Chemistry 29 (2008), S. 122–129

[Praprotnik u. a. 2008b] Praprotnik, M. ; Junghans, C. ; Delle Site, L. ; Kremer, K.: Simulation Approaches to Soft Matter: Generic Statistical Properties vs. Chemical Details. In: Computer Physics Communications 179 (2008), S. 51–60

[Press u. a. 1992] Press, W. H. ; Teukolsky, S. A. ; Vetterling, W. T. ; Flannery, B. P.: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 1992

[PSF 2011] PSF: Python Software Foundation, 2011. – `http://www.python.org/psf/`, zuletzt besucht am 28. November 2011.

[Python 2011] Python: Python Programming Language, 2011. – Online-Dokumentation Versionen 2.x und 3.x, `http://python.org/doc/`, zuletzt besucht am 28. November 2011.

[R Development Core Team 2011] R Development Core Team: R Language Definition, Version 2.13.2, 2011. – Handbuch, `http://www.r-project.org/`, zuletzt besucht am 28. November 2011.

[Rahman 1964] Rahman, A.: Correlations of Motion of Atoms in Liquid Argon. In: Physical Review 136A (1964), S. 405–411

[Reith u. Kirschner 2011] Reith, D. ; Kirschner, K. N.: A Modern Workflow for Force-Field Development—Bridging Quantum Mechanics and Atomistic Computational Models. In: Computer Physics Communications 182 (2011), S. 2184–2191

[Reith u. a. 2001] Reith, D. ; Meyer, H. ; Müller-Plathe, F.: Mapping Atomistic to Coarse-grained Polymer Models. In: Macromolecules 34 (2001), S. 2335–2345

[Reith u. a. 2002] Reith, D. ; Meyer, H. ; Müller-Plathe, F.: CG-OPT: A software package for automatic force field design. In: Computer Physics Communications 148 (2002), S. 299–313

[Reith u. a. 2003] Reith, D. ; Pütz, M. ; Müller-Plathe, F.: Deriving Effective Mesoscale Potentials from Atomistic Simulations. In: Journal of Computational Chemistry 24 (2003), S. 1624–1636

[Ripley 2011] Ripley, B.: The Package 'MASS', 2011. – Handbuch Version 7.3-16, `http://cran.r-project.org/web/packages/MASS/`, zuletzt besucht am 28. November 2011.

[Rogers u. Seddon 2003] Rogers, R. D. ; Seddon, K. R.: Ionic Liquids—Solvents of the Future? In: Science 302 (2003), S. 792–793

[Roothaan 1951] Roothaan, C. C. J.: New Developments in Molecular Orbital Theory. In: Reviews of Modern Physics 23 (1951), S. 69–89

[Rosenblatt 1957] Rosenblatt, F.: The Perceptron: A Perceiving and Recognizing Automaton / Cornell Aeronautical Lab. 1957. – Forschungsbericht

[Roweis 1996] Roweis, S.: Levenberg-Marquardt Optimization / University of Toronto. 1996. – Forschungsbericht

[Ryckaert u. a. 1977] Ryckaert, J.-P. ; Ciccotti, G. ; Berendsen, H. J. C.: Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes. In: Journal of Computational Physics 23 (1977), S. 327–341

[Schölkopf u. Smola 2002] Schölkopf, B. ; Smola, A.: Learning with Kernels: Suppor Vector Machine, Regularization, Optimization, and Beyond. MIT Press, 2002

[Schrödinger 1926] Schrödinger, E.: Quantisierung als Eigenwertproblem. In: Annalen der Physik 79 (1926), S. 361–376

[Schwefel 1995] Schwefel, H.-P.: Evolution and Optimum Seeking. John Wiley & Sons, Inc., 1995

[Shawe-Taylor u. Christianini 2004] Shawe-Taylor, J. ; Christianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, 2004

[Siepmann u. a. 1993] Siepmann, J. I. ; Karaborni, S. ; Smit, B.: Simulating the Critical Behaviour of Complex Fluids. In: Nature 365 (1993), S. 330–332

[Singer u. Nicolson 1972] Singer, S. J. ; Nicolson, Garth L.: The Fluid Mosaic Model of the Structure of Cell Membranes. In: Science 175 (1972), S. 720–731

[Smolyak 1963] Smolyak, S. A.: Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions. In: Soviet Mathematics Doklady 4 (1963), S. 240–243

[Snow u. a. 2005] Snow, Christopher D. ; Sorin, Eric J. ; Rhee, Young M. ; Pandel, S. Vijay: How well can Simulation Predict Protein Folding Kinetics and Thermodynamics? In: Annual Review of Biophysics and Biomolecular Structure 34 (2005), S. 43–69

[Span u. a. 1996] Span, R. ; ; Wagner, W.: A New Equation of State for Carbon Dioxide Covering the Fluid Region from the Triple-Point Temperature to 1100 K at Pressures up to 800 MPa. In: Journal of Physical and Chemical Reference Data 25 (1996), S. 1509–1596

[Span u. a. 2000] Span, R. ; Lemmon, E. W. ; Jacobsen, R. T. ; Wagner, W. ; Yokozeki, A.: A Reference Equation of State for the Thermodynamic Properties of Nitrogen for Temperatures from 63.151 to 1000 K and Pressures to 2200 MPa. In: Journal of Physical and Chemical Reference Data 29 (2000), S. 1361–1433

[Stoll 2005] Stoll, J.: Molecular Models for the Prediction of Thermophysical Properties of Pure Fluids and Mixtures, Universität Stuttgart, Diss., 2005

[Stoll u. a. 2003a] Stoll, J. ; Vrabec, J. ; Hasse, H.: Comprehensive Study of the Vapour–Liquid Equilibria of The Pure Two–Centre Lennard–Jones plus Pointdipole Fluid. In: Fluid Phase Equilibria 209 (2003), S. 29–53

[Stoll u. a. 2003b] Stoll, J. ; Vrabec, J. ; Hasse, H.: A Set of Molecular Models for Carbon Monoxide and Halogenated Hydrocarbons. In: Journal of Chemical Physics 119 (2003), S. 11396–11407

[Stoll u. a. 2001] Stoll, J. ; Vrabec, J. ; Hasse, H. ; Fischer, J.: Comprehensive Study of the Vapour–Liquid Equilibria of the Pure Two–Centre Lennard–Jones plus Pointquadrupole Fluid. In: Fluid Phase Equilibria 179 (2001), S. 339–362

[Stork u. a. 2008] Stork, A. ; Thole, C.-A. ; Klimenko, S. ; Nikitin, I. ; Nikitina, L. ; Astakhov, Y.: Towards Interactive Simulation in Automotive Design. In: The Visual Computer 24 (2008), S. 947–953

[Stubbs u. a. 2004] Stubbs, J. M. ; Potoff, J. J. ; Siepmann, J. I.: Transferable Potentials for Phase Equilibria. 6. United-atom Description for Ethers, Glycols, Ketones and Aldehydes. In: Journal of Physical Chemistry B 108 (2004), S. 17596–17605

[Sturm u. Pietruschka 1997] Sturm, M. ; Pietruschka, U.: Verallgemeinerte Radiale Basisfunktionen zur Approximation höherdimensionaler Funktionen. In: Informatik – Forschung und Entwicklung 12 (1997), S. 30–37

[Sun 2004] Sun, H.: Prediction of fluid densities using automatically derived VDW parameters. In: Fluid Phase Equilibria 217 (2004), S. 59–76

[Swope u. a. 1982] Swope, W. C. ; Andersen, H. C. ; Berens, P. H. ; Wilson, K. R.: A Computer Simulation Method for the Calculation of Equilibrium Constants for the Formation of Physical Clusters of Molecules: Application to Small Water Clusters. In: Journal of Chemical Physics 76 (1982), S. 637–649

[Tersoff 1988] Tersoff, J.: New Empirical Approach for the Structure and Energy of Covalent Systems. In: Physical Review B 37 (1988), S. 6991–7000

[Thole u. a. 2007] Thole, C.-A. ; Nikitina, L. ; Nikitin, I. ; Steffes-lai, D. ; Kersten, R. ; Bruns, J.: Constrained Optimization with DesParO. In: Proceedings of the Conference Virtual Product Development (VPD) in Automotive Engineering, 2007

[Tibshirani 1996] Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. In: Journal of the Royal Statistical Society, Series B 58 (1996), S. 267–288

[Tokuda u. a. 2005] Tokuda, H. ; Hayamizu, K. ; Ishii, K. ; Susan, A. B. H. ; Watanabe, M.: Physicochemical Properties and Structures of Room Temperature Ionic Liquids. In: Journal of Physical Chemistry B 109 (2005), S. 6103–6110

[Towhee 2011] Towhee: Monte Carlo Molecular Simulation Code, 2011. – Online-Dokumentation Version 6.2.14, `http://towhee.sourceforge.net/`, zuletzt besucht am 5. Dezember 2011.

[Tozzini 2005] Tozzini, V.: Coarse-Grained Models for Proteins. In: Current Opinion in Structural Biology 15 (2005), S. 144–150

[Ungerer u. a. 1999] Ungerer, P. ; Beauvais, C. ; Delhommelle, J. ; Boutin, A. ; Rousseau, B. ; Fuchs, A. H.: Optimization of the anisotropic united atoms intermolecular potential for n-alkanes. In: Journal of Computational Physics 112 (1999), S. 5499–5510

[Urbanek 2011] Urbanek, S.: The Package 'Cairo', 2011. – Handbuch Version 1.5-0, `http://cran.r-project.org/web/packages/Cairo/`, zuletzt besucht am 28. November 2011.

[Valiullin u. a. 2006] Valiullin, R. ; Naumov, S. ; Galvosas, P. ; Kärger, J. ; Woo, H.-J. ; Porcheron, F. ; Monson, P. A.: Exploration of Molecular Dynamics During Transient Sorption of Fluids in Mesoporous Materials. In: Nature 443 (2006), S. 965–968

[van der Spoel u. a. 2010] van der Spoel, D. ; Lindahl, E. ; Hess, B. ; Buuren, A. R. ; Apol, E. ; Meulenhoff, P. J. ; Tieleman, D. P. ; Sijbers, A. L. T. M. ; Feenstra, K. A. ; Drunen, R. van ; Berendsen, H. J. C.: Gromacs User Manual Version 4.5.4, 2010. – Handbuch, `http://www.gromacs.org/Documentation/Manual/`, zuletzt besucht am 21. November 2011.

[van Gunsteren u. Berendsen 1990] van Gunsteren, W. F. ; Berendsen, H.: Computer Simulation of Molecular Dynamics: Methodology, Applications and Perspectives in Chemistry. In: Angewandte Chemie – International Edition Bd. 29, 1990, S. 992–1023

[van Gunsteren u. a. 1996] van Gunsteren, W. F. ; Billeter, S. R. ; Eising, A. A. ; Hünenberger, P. H. ; Krüger, P. ; Mark, A. E. ; Scott, W. R. P. ; Tironi, I. G.: Biomolecular Simulation: The GROMOS96 Manual and User Guide, 1996. – Handbuch, vdf Hochschulverlag AG an der ETH Zürich.

[van Ness 1995] van Ness, H. C.: Thermodynamics in the Treatment of Vapor/Liquid Equilibrium (VLE) Data. In: Pure and Applied Chemistry 67 (1995), S. 859–872

[Vapnik 1995] Vapnik, V. N.: The Nature of Statistical Learning Theory. Springer-Verlag, 1995

[Venables u. Smith 2011] Venables, W. N. ; Smith, D. N.: An Introduction to R—Notes on R: A Programming Environment for Data Analysis and Graphics, Version 2.13.2, 2011. – Handbuch, `http://www.r-project.org/`, zuletzt besucht am 28. November 2011.

[Verlet 1967] Verlet, L.: Computer 'Experiments' on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. In: Physical Review 159 (1967), S. 98–103

[Verlet 1968] Verlet, L.: Computer 'Experiments' on Classical Fluids. II. Equilibrium Correlation Functions. In: Physical Review 165 (1968), S. 201–214

[Vrabec u. Gross 2008] Vrabec, J. ; Gross, J.: Vapor–Liquid Equilibria Simulation and an Equation of State Contribution for Dipole–Quadrupole Interactions. In: Journal of Physical Chemistry B 112 (2008), S. 51–60

[Vrabec u. Hasse 2002] Vrabec, J. ; Hasse, H.: Grand Equilibrium: Vapour–Liquid Equilibria by a New Molecular Simulation Method. In: Molecular Physics 100 (2002), S. 3375–3383

[Vrabec u. a. 2002] Vrabec, J. ; Kettler, M. ; Hasse, H.: Chemical Potential of Quadrupolar Two–Centre Lennard–Jones Fluids by Gradual Insertion. In: Chemical Physics Letters 356 (2002), S. 431–436

[Vrabec u. a. 2001] Vrabec, J. ; Stoll, J. ; Hasse, H.: A Set of Molecular Models for Symmetric Quadrupolar Fluids. In: Journal of Physical Chemistry B 105 (2001), S. 12126–12133

[Wang u. Kollman 2001] Wang, J. ; Kollman, P. A.: Automatic Parameterization of Force Field by Systematic Search and Genetic Algorithms. In: Journal of Computational Chemistry 22 (2001), S. 1219–1228

[Wang u. a. 2004] Wang, J. ; Wolf, R. M. ; Caldwell, J.W. ; Kollman, P. A. ; Case, D. A.: Development and Testing of a General Amber Force Field. In: Journal of Computational Chemistry 25 (2004), S. 1157–1174

[Wasserscheid u. Welton 2003] Wasserscheid, P. ; Welton, T.: Ionic Liquids in Synthesis. Wiley-VCH Weinheim, 2003

[Weiner u. a. 1984] Weiner, S. J. ; Kollman, P. A. ; Case, D. A. ; Singh, U. C. ; Ghio, C. ; Alagona, G. ; Profeta, S. ; Weiner, P.: A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins. In: Journal of the American Chemistry Society 106 (1984), S. 765–784

[Wendland 1996] Wendland, H.: Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger, Universität Göttingen, Diss., 1996

[Wendland 2005] Wendland, H.: Scattered Data Approximation. Cambridge University Press, 2005

[Widom 1963] Widom, B.: Some Topics in the Theory of Fluids. In: Journal of Chemical Physics 39 (1963), S. 2808–2812

[Wikipedia 2011a] Wikipedia: Wikipedia – Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/wiki/CMA-ES`. – zuletzt besucht am 21. November 2011. Die Graphik ist gemeinfrei.

[Wikipedia 2011b] Wikipedia: Wikipedia – Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/wiki/Benzol`. – zuletzt besucht am 21. November 2011. Die Graphik steht unter der *GNU Public License*. Urheber ist der Wikipedia-Nutzer *Moebius1*.

[Wikipedia 2011c] Wikipedia: Wikipedia – Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/wiki/Phosgen`. – zuletzt besucht am 21. November 2011. Die Graphik ist gemeinfrei.

[Wikipedia 2011d] Wikipedia: Wikipedia – Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/wiki/Methanol`. – zuletzt besucht am 21. November 2011. Die Graphik ist gemeinfrei.

[Wikipedia 2011e] Wikipedia: Wikipedia – Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/wiki/Kohlenstoffdisulfid`. – zuletzt besucht am 21. November 2011. Die Graphik ist gemeinfrei.

[Wikipedia 2011f] Wikipedia: Wikipedia – Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/wiki/Ethylenoxid`. – zuletzt besucht am 21. November 2011. Die Graphik ist gemeinfrei.

[Williams u. Kelley 2007] Williams, T. ; Kelley, C.: Gnuplot 4.2, 2007. – Handbuch, `http://www.gnuplot.info/documentation.html`, zuletzt besucht am 24. Oktober 2011.

[Wood 1968] Wood, W. W.: Monte Carlo Calculations for Hard Disks in the Isothermal-Isobaric Ensemble. In: Journal of Chemical Physics 48 (1968), S. 415–434

[Woolf 1982] Woolf, L. A.: Self–diffusion in Carbon Disulphide under Pressure. In: Journal of the Chemical Society, Faraday Transacions 1 78 (1982), S. 583–590

[Xfig 2007] Xfig, 2007. – Online-Dokumentation Version 3.2.5, `http://epb.lbl.gov/xfig/`, zuletzt besucht am 24. Oktober 2011.

[Xmgrace 2008] Xmgrace, 2008. – Online-Dokumentation Version 5.1.22, `http://plasma-gate.weizmann.ac.il/Grace/doc/UsersGuide.html`, zuletzt besucht am 24. Oktober 2011.

[Yin u. MacKerell Jr. 1998] Yin, D. ; MacKerell Jr., A. D.: Combined Ab initio/Empirical Approach for the Optimization of Lennard–Jones Parameters. In: Journal of Computational Chemistry 19 (1998), S. 334–348

[Yoneya u. a. 1994] Yoneya, M. ; Berendsen, H. C. ; Hirasawa, K.: A Noniterative Matrix Method for Constraint Molecular-Dynamics Simulations. In: Molecular Simulations 13 (1994), S. 395–405

[Yoshida u. a. 2008] Yoshida, K. ; Matubayasi, N. ; Nakahara, M.: Self–diffusion Coefficients for Water and Organic Solvents at High Temperatures along the Coexistence Curve. In: Journal of Chemical Physics 129 (2008), S. 214501–214509

[Zenger 1991] Zenger, C.: Sparse Grids. In: Hackbusch, W. (Hrsg.): Parallel Algorithms for Partial Differential Equations, Notes on Numerical Fluid Mechanics Bd. 31, 1991, S. 241–251

[Zhou u. Stell 1992] Zhou, Y. ; Stell, G.: Chemical Association in Simple Models of Molecular and Ionic Fluids. II. Thermodynamic Properties. In: Journal of Chemical Physics 96 (1992), S. 1504–1506

[Zou u. Hastie 2005] Zou, H. ; Hastie, T.: Regularization and Variable Selection via the Elastic Net. In: Journal of the Royal Statistical Society, Series B 67 (2005), S. 301–320

Eidesstattliche Erklärung

Ich versichere, daß ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; daß diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; daß sie – abgesehen von unten angegebenen Teilpublikationen – noch nicht veröffentlicht worden ist sowie, daß ich eine solche Veröffentlichung vor Abschluß des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Herrn Prof. Dr. Ulrich Trottenberg betreut worden.

Marco Hülsmann, Diplom-Mathematiker

Teilpublikationen der vorliegenden Dissertation:

1. Hülsmann, Marco; Köddermann, Thorsten; Vrabec, Jadran; Reith, Dirk: *GROW: A Gradient-based Optimization Workflow for the Automated Development of Molecular Models*, In: Computer Physics Communications 181 (2010), S. 499–513.
2. Hülsmann Marco; Vrabec, Jadran; Maaß, Astrid; Reith, Dirk: *Assessment of Numerical Optimization Algorithms for the Development of Molecular Models*, In: Computer Physics Communications 181 (2010), S. 887–905.
3. Hülsmann, Marco; Müller, Thomas J.; Köddermann, Thorsten; Reith, Dirk: *Automated Force Field Optimisation of Small Molecules Using a Gradient-based Workflow Package*, In: Molecular Simulation 36 (2011), S. 1182–1196.
4. Köddermann, Thorsten; Kirschner, Karl N.; Vrabec, Jadran; Hülsmann, Marco; Reith, Dirk: *Liquid–Liquid Equilibria of Dipropylene Glycol Dimethyl Ether and Water by Molecular Dynamics*, In: Fluid Phase Equilibria 310 (2011), S. 25–31.

Lebenslauf

Name:	Marco Hülsmann
Geburtstdatum:	26.09.1982
Geburtsort:	Düren-Birkesdorf
Staatsangehörigkeit:	deutsch
Schulbildung:	1988–1992: Grundschule Düren-Derichsweiler 1992–2001: Burgau-Gymnasium Düren Abschluß: Abitur
Studienverlauf:	WS 2001/02 bis SS 2003: Lehramtsstudium Mathematik und Chemie an der Universität zu Köln WS 2003/04 bis WS 2006/07: Studium der Mathematik mit Nebenfach Chemie an der Universität zu Köln, Vordiplom am 2.11.2004 Abschluß: Diplom (14.11.2006) Diplomarbeit im Fach Angewandte Mathematik seit SS 2008: Promotionsstudium der Mathematik an der Universität zu Köln
Berufliche Tätigkeiten:	September 2003 bis März 2007: Studentische Hilfskraft am Fraunhofer-Institut für Algorithmen und Wiss. Rechnen (SCAI) April 2007 bis März 2008: Wiss. Mitarbeiter am Fraunhofer-Institut SCAI April 2008 bis März 2010: Promotionsstipendiat an der Universität zu Köln seit April 2010: Wiss. Mitarbeiter an der Universität zu Köln

Marco Hülsmann, Diplom-Mathematiker

Danksagung

Beginnen möchte ich mit den Personen, denen die vorliegende Dissertation gewidmet ist: meinem Sohn Maurice Cürsgen, meiner Mutter Gisela Hülsmann sowie meinen besten Freunden Arash und Roozbeh Faroughi. Ohne ihre Existenz und grenzenlose emotionale Unterstützung wäre ich niemals in der Situation, in der ich jetzt bin. Sie haben mir stets die Kraft und den notwendigen Halt gegeben, was für die jahrelange Anfertigung dieser Arbeit von enormer Bedeutung war. Weiterhin möchte ich diese Arbeit meiner Tante Ingrid Hülsmann und *in memoriam* meinem Vater Jürgen Hülsmann widmen.

An zweiter Stelle möchte ich mich bei Herrn Prof. Dr. Ulrich Trottenberg, meinem Erstgutachter, ganz herzlich bedanken, vor allem für das Interesse an der Numerischen Mathematik, das er damals in mir geweckt hat. Er hat es mir vor acht Jahren ermöglicht, am Fraunhofer-Institut SCAI als Studentische Hilfskraft tätig zu sein, wo ich später meine Diplomarbeit und die vorliegende Dissertation anfertigen konnte.
Weiterhin möchte ich mich bei meinem Prüfungskomittee bedanken: bei Frau Prof. Dr. Caren Tischendorf für die Übernahme des Zweitgutachtens, bei Herrn Prof. Dr. Florian Müller-Plathe dafür, daß er sich gerne dazu bereiterklärt hat, als externer Drittgutachter und Experte für Molekulare Simulationen für die interdisziplinäre Doktorarbeit zu fungieren, bei Herrn Prof. Dr. Gerd Meyer für die Übernahme des Vorsitzes bei meiner anschließenden Disputation sowie bei Herrn Dr. Roman Wienands für die Übernahme des Beisitzes.

Ein ganz besonderer Dank gilt meinem Betreuer am Fraunhofer-Institut SCAI, Dr. Dirk Reith, und unserem befreundeten Kooperationspartner Prof. Dr.-Ing. Jadran Vrabec. Durch die beiden habe ich sehr viel über Physik, Thermodynamik und Molekulare Simulationen gelernt, und sie haben dadurch einen entscheidenden Beitrag zu meiner Dissertation geleistet. Auch möchte ich mich für unsere gemeinsamen Publikationen und die Mühe beim Lesen meiner sehr lang gewordenen Doktorarbeit ganz herzlich bedanken.

Ein weiterer besonderer Dank gilt meinem Kollegen Dr. Thorsten Köddermann für unsere tägliche Zusammenarbeit und seine ständig für mich zur Verfügung stehende Expertise als Chemiker. Er hat mich insbesondere bei den Simulationen Ionischer Flüssigkeiten unterstützt und auch sonst sehr viel vor allem zu meinem praktischen Wissen im Bereich Molekularer Simulationen beigetragen.
Weiterhin bedanke ich mich bei unserer Arbeitsgruppe *Computational Chemical Engineering (CoChE)*: bei Dr. Astrid Maaß für die Zusammenarbeit bezüglich DesParO und Ethylenoxid, bei Karl N. Kirschner, PhD, für die Bereitstellung von intramolekularen Modellparametern und Partialladungen, bei Dr. Anton Schüller und Dr. Gerd Winter für insbesondere mathematische Anregungen, bei Dr. Thomas Brandes und Prof. Dr. Axel Arnold (jetzt Uni Stuttgart) für ihre insbesondere technische Unterstützung sowie bei unserem Abteilungsleiter Dr. Johannes Linden, welcher stets sehr an dem Fortschritt meiner Arbeit interessiert war.

Ebenfalls bedanken möchte ich mich bei meinen Studenten Janina Hemmersbach, Sonja Kopp, Markus Huber und Andreas Krämer, deren Arbeiten ebenfalls einen wichtigen Beitrag im Bereich der Kraftfeldparameteroptimierung geleistet haben.
Besonders erwähnen möchte ich auch meine mittlerweile pensionierte Kollegin Angelika Weihermüller wegen der sehr netten, lustigen und abwechslungsreichen gemeinsamen Pausen über viele Jahre hinweg sowie meine befreundeten Kollegen Antje Wolf, Dr. Roman Klinger, Bernd Müller, André Oeckerath, Wolfgang Vonolfen, Dr. Oliver Wäldrich und Dr. Marc Zimmermann.

Als nächstes möchte ich mich bei unseren externen Kooperationspartnern bedanken, und zwar bei Herrn Prof. Dr.-Ing. Hans Hasse und Stephan Deublein (TU Kaiserslautern), Dr. Thomas J. Müller (ehem. TU Darmstadt), Prof. Dr.-Ing. Christoph M. Friedrich (FH Dortmund, ehem. SCAI) sowie Herrn Detlef Borscheid (BDW Automotive).

Selbstverständlich darf auch mein Freundeskreis in dieser Danksagung nicht fehlen: Bedanken möchte ich mich bei unserem Gitarren-Dinner, bestehend aus Arash und Roozbeh Faroughi, Marko Flod, Ralf Gossmann, Martin Ossendorf und Momo Ziaei, weiterhin bei Lena Becker, Jacqueline Fani-Musevi, Verena Schmitz, Julia Schuschke, Leo Nußbaum, Roberto Reith und Ruben Schuschke. Insbesondere möchte ich auch das Showballett Kruuschberger Funken hervorheben, wo ich seit 14 Jahren tanze und stets einen sehr guten sportlichen Ausgleich finden konnte. Hier möchte ich vor allem Marlies und Willi Plum erwähnen, in denen ich zusätzlich sehr gute Freunde gewonnen habe.

Zum Schluß möchte ich der Universität zu Köln für die fast vierjährige Finanzierung danken, dem Rechenzentrum der Uni Köln dafür, daß ich die Möglichkeit hatte, viele meiner Rechnungen auf dem neuen Hochleistungsrechencluster *Cheops* auszuführen, und Herrn Dr. Jörg Behrend dafür, daß er mir den Rechencluster des Mathematischen Instituts der Uni Köln zur Verfügung gestellt hat.

Analog zu meiner Diplomarbeit möchte ich auch meine Dissertation mit einem Zitat von der Kölner Musikgruppe HÖHNER schließen. Es handelt sich dabei um eine Lebensweisheit, die mich auch zu der Anfertigung meiner Doktorarbeit motiviert hat: Man darf niemals allzu lange auf sein Glück warten, und man muß im Leben stets jede Chance beim Schopf packen. Später bereut man vielleicht eher die Dinge, die man nicht getan hat, als die, die man getan hat:

Spar ding Dröum nit op för murje,
denn wer weiß, wat murje es!
Do kanns et Jlöck nit halde,
on fröher oder späder
litt dich jede Droum em Ress.
HÖHNER - Album 'Jetzt und hier' (2007)

Molekulare Simulationen ermöglichen es, den Einfluss mikroskopischer Prozesse auf makroskopische Phänomene zu studieren. Um Simulationen in den Naturwissenschaften erfolgreich anwenden zu können, müssen geeignete molekulare Modelle vorliegen. Das Fundament einer Simulation zur quantitativen Vorhersage von physikalischen Eigenschaften ist das sogenannte Kraftfeld. Die Hauptschwierigkeit liegt in dessen Parametrisierung. Dies manuell durchzuführen, ist sehr zeitaufwendig, da für jeden Parametersatz eine aufwendige Simulation durchzuführen ist.

In dieser Arbeit wird ein neues automatisiertes Parametrisierungsschema vorgeschlagen, welches auf der Lösung eines mathematischen Optimierungsproblems basiert. Innerhalb des Optimierungsprozesses werden Eigenschaften, die aus einer Molekularen Simulation resultieren, an ihre entsprechenden experimentellen Referenzdaten angepasst. Da Molekulare Simulationen eine hohe Komplexität aufweisen und die resultierenden Eigenschaften mit statistischem Rauschen behaftet sind, wird das Optimierungsproblem sowohl mit bereits vorhandenen als auch mit neuartigen, effizienten und robusten numerischen Algorithmen gelöst.

ISBN 978-3-8396-0426-7