

# Ein numerisches Verfahren zum Fitten von empirischen Potentialen mit quantenmechanischen Daten

Nadine Pastuszka

Geboren am 7. Juni 1989 in Brühl

7. Februar 2012

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Michael Griebel

INSTITUT FÜR NUMERISCHE SIMULATION

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER  
RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Moleküldynamik</b>	<b>5</b>
2.1. Die Schrödinger-Gleichung . . . . .	6
2.1.1. Die stationäre Schrödinger-Gleichung . . . . .	6
2.1.2. Die Born-Oppenheimer-Näherung . . . . .	7
2.1.3. Hartree-Fock-Verfahren . . . . .	8
2.1.4. Post-Hartree-Fock-Verfahren . . . . .	10
2.1.5. Die Bestimmung von Potentialflächen . . . . .	12
2.2. Das Verlet-Verfahren . . . . .	14
2.3. Das Programm PSI3 zur Berechnung quantenmechanischer Daten . . . . .	16
<b>3. Maschinelles Lernen</b>	<b>19</b>
3.1. Grundlagen von Lernalgorithmen . . . . .	19
3.1.1. Regressionen im überwachten Lernen . . . . .	20
3.2. Der Gauß-Prozess . . . . .	22
3.2.1. Die Gauß-Prozess-Regression . . . . .	23
3.2.2. Der Gauß-Kern . . . . .	30
3.2.3. Die Wahl der Parameter . . . . .	32
3.2.4. Die Gauß-Prozess-Regression mit Gradienten . . . . .	33
3.3. Die Anwendung auf Potentialkurven . . . . .	36
3.4. Das Learn on the Fly Verfahren . . . . .	37
3.4.1. Die Anwendung auf die Moleküldynamik . . . . .	39
<b>4. Numerische Ergebnisse</b>	<b>41</b>
4.1. Das Wasserstoffmolekül als Modellbeispiel . . . . .	41
4.2. Die Anwendung der Gauß-Prozess-Regression . . . . .	43
4.2.1. Die Wahl der Hyperparameter . . . . .	47
4.2.2. Fehleranalyse . . . . .	47
4.3. Die Anwendung für das Verlet-Verfahren . . . . .	48
<b>5. Zusammenfassung und Ausblick</b>	<b>53</b>
<b>A. Anhang</b>	<b>57</b>
A.1. Bayes-Statistik . . . . .	57
A.2. Kenngrößen für Zufallsvariablen . . . . .	58
A.3. Matrizen . . . . .	59



## 1. Einleitung

Die Moleküldynamik ist in vielen Bereichen der aktuellen Wissenschaft und Forschung von großer Bedeutung. Die effektiven makroskopischen Eigenschaften eines Materials sind meist in der Struktur auf der Nanoskala, also in den Wechselwirkungen von Atomen und Molekülen, begründet. Das Gebiet der Materialwissenschaften beschäftigt sich mit der Klassifikation, der Synthese und der Bestimmung der Eigenschaften von Materialien. Darüber hinaus besteht das Ziel darin neue, den jeweiligen Verwendungszwecken angepasste, Stoffe zu entwickeln.

Das Ziel der Moleküldynamik ist es die Vorgänge auf molekularer Ebene mit Algorithmen auf Computern möglichst genau simulieren zu können, da sich reale Experimente häufig als problematisch, sehr aufwändig, teuer oder sogar unmöglich erweisen.

Die Simulation in der Moleküldynamik basiert auf der Kenntnis der Wechselwirkungen zwischen Teilchen auf atomarer Ebene. Die entscheidende Kenngröße ist dabei die auf die Teilchen einwirkende Kraft, für deren Bestimmung eine Potentialfunktion notwendig ist. Diese beschreibt die Wechselwirkungen zwischen den Teilchen, aus denen das zu untersuchende Molekül besteht.

Da die Gesetze der klassischen Mechanik auf atomarer Ebene nicht mehr gelten, werden diese durch die Quantenmechanik ersetzt. Das zugrundeliegende mathematische Modell ist die *Schrödingergleichung*, deren Lösung eine Wellenfunktion ist, welche alle messbaren Größen des jeweils betrachteten quantenmechanischen Systems enthält. Da diese allerdings eine hochdimensionale Differentialgleichung darstellt und nur für wenige Spezialfälle analytisch gelöst werden kann, ist es notwendig eine Näherung zu bestimmen. Beispielsweise wird die Wellenfunktion in der *Born-Oppenheimer-Näherung* dadurch approximiert, dass nur die Elektronen der Atome über die Quantenmechanik beschrieben werden und die Kerne über die klassische Mechanik. Daraus resultiert die *elektronische Schrödingergleichung*. Durch weitere Näherungen lässt sich letztendlich zwar eine Lösung numerisch approximieren, in vielen Fällen werden allerdings trotz moderner Hochleistungsrechner immer noch ein enorm hoher Rechenaufwand und sehr lange Laufzeiten benötigt.

Die Wellenfunktion der Schrödingergleichung enthält unter anderem die Information über potentielle Energien, welche zur Bestimmung der hochdimensionalen Potentialfunktion benötigt werden. Für Mehrteilchensysteme bedeutet dies allerdings, dass alle Energien zwischen den Teilchen des Moleküls zu jedem möglichen Abstand im Raum betrachtet werden müssen, was ein schwer zu lösendes Problem darstellt. Da aus den Potentialen allerdings die resultierenden Kräfte berechnet werden können und sie deshalb entscheidend für die Simulation von Molekülbewegungen sind, wird nach effizienten Möglichkeiten gesucht diese mit einem geringeren Aufwand zu bestimmen.

An diesem Punkt setzt die vorliegende Arbeit an.

### Problemstellung

Die hochdimensionalen *Born-Oppenheimer-Potentialhyperflächen* sind ein wichtiger Bestandteil der Moleküldynamik. Für jede Teilchenkonfiguration ist zu deren Bestimmung eine approximative Lösung der *elektronischen Schrödingergleichung* notwendig. Die Potentialfunktion ist wegen

der hohen Dimension allerdings wie die *elektronische Schrödingergleichung* selber vom „Fluch der Dimensionen“ betroffen, was eine klassische Diskretisierung unmöglich macht.

Das Gesamtpotential eines Moleküls wird deshalb über eine Näherung bestimmt, welche die hochdimensionale Potentialfunktion über eine partikelweise Zerlegung beschreibt und aus den einzelnen Beiträgen der Wechselwirkungen aller Untermengen der im Molekül enthaltenen Teilchen entsteht. Dadurch wird der „Fluch der Dimensionen“ gebrochen.

Das Ziel dieser Arbeit ist es den ersten Schritt zur Bestimmung einer solchen Zerlegung zur effizienten Berechnungen der *Born-Oppenheimer-Moleküldynamik* durchzuführen.

Für ausgewählte Moleküle sind die Potentialflächen bereits bekannt und können auf solche mit ähnlichen Eigenschaften übertragen werden. Es existiert allerdings keine allgemeine Parameterdarstellung, weshalb die Bestimmung für beliebige Moleküle aufgrund der Komplexität zur approximativen Berechnung von Potentialfunktionen ein enormes Problem darstellt.

Ziel ist es diese nicht mehr ausschließlich über die Näherung der Lösung der *Schrödingergleichung* zu berechnen, sondern sie anhand ausgewählter quantenmechanischer Daten möglichst effizient und genau zu approximieren.

Ist es möglich die Anzahl der zu berechnenden Daten möglichst gering zu halten, so bietet dies eine wesentliche Zeitersparnis, weil die Ermittlung der hochdimensionalen, aufwändig zu berechnenden Potentialfläche somit durch eine endliche Anzahl an Approximationen der Lösung der *elektronischen Schrödingergleichung* auskommt.

Des weiteren ist die Kenngröße der Kraft notwendig für jedes moleküldynamische Verfahren, um diese zu bestimmen wird die partielle Ableitung der Potentialfunktion berechnet. Zudem wird für die Simulation der Bewegungen ein Zeitdiskretisierungsverfahren, wie beispielsweise das *Verlet-Verfahren*, benötigt.

Um in der Moleküldynamik mit beliebigen Molekülen effizient Simulationen durchführen zu können ist es notwendig die Potential- und Kraftfunktion anhand weniger Daten approximieren zu können.

## Lösungsansätze

Die Bestimmung der Potentialfunktion mit der Einbindung von quantenmechanischen Daten erfordert ein Approximationsverfahren für hochdimensionale Funktionen, welches auf den aus den Daten erhaltenen Informationen basiert.

In diesem Zusammenhang werden Algorithmen aus dem Bereich des *maschinellen Lernens* eingesetzt, welche aus einer gegebenen Anzahl an Eingabedaten, den sogenannten *Trainingsdaten*, eine Funktion lernen und diese Approximation durch jeden weiteren Datensatz verbessern können. Der als Regression bezeichnete Teilbereich der Algorithmen des *maschinellen Lernens* beschäftigt sich mit dem Lernen von Zusammenhängen zwischen Variablen, welche über quantitative Größen beschrieben werden können, also dem Lernen von Funktionen.

Für die Problemstellung der Bestimmung von Potentialflächen wird die *Gauß-Prozess-Regression* als ein Verfahren des maschinellen Lernens verwendet. Dieses ermöglicht eine Extrapolation, das bedeutet die Werte der zugrundeliegenden Funktion können an jeder beliebigen Stelle vorausgesagt werden. Zudem kann gleichzeitig ein lokaler Schätzer für den entstandenen Fehler berechnet werden, welcher ein Maß für die Genauigkeit der Voraussage darstellt. Speziell in Bezug auf die Problemstellung stellt dies einen entscheidenden Vorteil dar, da anhand dieser Information ein Kriterium entwickelt werden kann, sodass nur an den Stellen quantenmechani-

---

sche Daten berechnet werden müssen, an den die Voraussage zu ungenau ist. Somit lässt sich die Anzahl der zu berechnenden Näherungen der *elektronischen Schrödingergleichung* auf ein Minimum reduzieren.

Innerhalb des Modells der *Gauß-Prozess-Regression* ist die Grundlage die Annahme von normal- beziehungsweise gaußverteilten Daten. Dadurch sind diese eindeutig bestimmt durch den Erwartungswert und die Kovarianz, das bedeutet die Korrelationen zwischen den jeweiligen Daten. Durch die entsprechenden Modellannahmen lässt sich die Dimension des Problems zur Bestimmung der Voraussage auf die Anzahl der Trainingsdaten vereinfachen und ist nicht mehr durch Anzahl an Parametern der zugrundeliegenden Funktion bestimmt. Da in der Moleküldynamik häufig die Untersuchung von Mehrteilchensystemen von großem Interesse ist, ist dies von entscheidendem Nutzen.

Die Kovarianz in Bezug auf die Bestimmung von Potentialen festzulegen und die darin enthaltenen freien Parameter zu optimieren ist einer der zentralen Punkte dieser Arbeit. Des Weiteren stellt sich die Frage, wie die *Trainingsdaten* gewählt werden müssen um einen möglichst geringen Rechenaufwand und gleichzeitig ein genaues Ergebnis zu gewährleisten.

Durch eine Erweiterung der *Gauß-Prozess-Regression* ist es möglich, zusätzlich zu den Funktionswerten, auch den Wert der zugehörigen Gradienten in die Voraussage einzubinden und diese somit zu verbessern. Zudem kann die Funktion der Ableitung mit gelernt werden, was in Bezug auf die Moleküldynamik einen entscheidenden Vorteil darstellt, da die Ableitung die Funktion der Kraft darstellt und notwendig für das *Verlet-Verfahren* zur Simulation von Molekülen ist.

Demnach wird nach einer geeigneten Darstellung für die Kovarianzfunktion und die darin enthaltenen Parametern zur Beschreibung der Potentialfunktion gesucht und ein Verfahren benötigt die *Trainingsdaten* geschickt zu wählen.

Dazu werden vorhandene Informationen über Potentialkurven wie die Bindungslängen von Molekülen ausgenutzt und anschließend über ein *Learn on the Fly* Verfahren neue Daten hinzugefügt, sodass das Resultat sich im Hinblick auf die Verwendung innerhalb des *Verlet-Verfahrens* zur Simulation von Molekülen als effizient erweist.

## Eigener Beitrag

Im Folgenden sind die wesentlichen eigenen Beiträge dieser Arbeit zusammengefasst.

- Einarbeitung in die Grundlagen der Moleküldynamik.
- Einarbeitung in die wesentlichen Schritte von *Gauß-Prozessen*, insbesondere die *Gauß-Prozess-Regression* und deren Erweiterung mit Gradienten.
- Implementierung der *Gauß-Prozess-Regression* mit Gradienten.
- Implementierung des *Verlet-Verfahrens*, die Einbindung des gelernten Potentials und der gelernten Funktion für die Kraft und anschließende Erweiterung durch ein *Learn on the Fly* Verfahren.

## Aufbau der Arbeit

Die Arbeit ist wie folgt strukturiert.

**Kapitel 2** gibt einen Überblick zur Moleküldynamik und die verschiedenen Approximationsmöglichkeiten der Lösung der *Schrödingergleichung* unter der Annahme der *Born-Oppenheimer-Näherung* werden erläutert. Zudem werden die wesentlichen Schritte des *Verlet-Verfahrens* zur Simulation von Molekülen beschrieben und ein Programm vorgestellt, welches die konkrete Berechnung der potentiellen Energien und zugehörigen Kräfte mittels der beschriebenen Verfahren ermöglicht.

**Kapitel 3** beschreibt die Grundlagen des maschinellen Lernens und die *Gauß-Prozess-Regression* als Teilgebiet und Verfahren zur Extrapolation von Funktionen anhand von *Trainingsdaten*. Es werden Verfahren zur Verbesserung der Voraussagen von beliebigen Funktionswerten vorgestellt und anschließend wird der Bezug zur Anwendung auf die Simulation in der Moleküldynamik hergestellt.

**Kapitel 4** zeigt anschließend die numerischen Ergebnisse dieses Verfahrens anhand des Wasserstoffmoleküls als Modellbeispiel auf. Die Potentialfunktion wird über die *Gauß-Prozess-Regression* extrapoliert und als Eingabewert für das *Verlet-Verfahren* verwendet. Durch die Wahl von entsprechenden Fehlerschranken für den Fehlerschätzer können so sehr genaue Ergebnisse erzielt werden.

**Kapitel 5** fasst den Inhalt der Arbeit zusammen und erläutert eine Anwendung des, über das vorgestellte Verfahren bestimmten, Potentials. Des weiteren werden Anregungen für weiterführende Themen für künftige Arbeiten beschrieben.

## Danksagung

An dieser Stelle ist es mir ein großes Anliegen mich bei den Menschen zu bedanken, die zu der Entstehung dieser Arbeit beigetragen haben.

Professor Dr. Griebel danke ich für die Vergabe eines interessanten Themas und seine Anregungen zur Problemstellung.

Den Mitarbeitern des Instituts für Numerische Simulation danke ich für die stetige und immer bereitwillige Unterstützung.

Besonderer Dank gilt dabei Jan Hamaekers, Christian Neuen, Christian Rieger und Jens Oettershagen, die mir für auftretende Fragestellungen und Probleme viele Hilfestellungen leisteten.

Ein ganz besonderer Dank geht an meinen Betreuer Ralph Thesen, der mich in der Zeit während der Entstehung dieser Arbeit in jeglicher Hinsicht unterstützt und motiviert hat und mir bei Fragen und Problemen immer zur Seite stand.

Zum Schluss möchte ich mich auch bei meinen Freunden und meiner Familie für ihren kontinuierlichen Beistand, ihre Geduld und ihr Vertrauen ganz herzlich bedanken.

## 2. Moleküldynamik

Die Moleküldynamik beschäftigt sich mit Untersuchungen von Materialien auf atomarer Ebene durch Computersimulationen, in denen Wechselwirkungen zwischen den beteiligten Teilchen berechnet und für bestimmte Zeitabschnitte simuliert werden können.

Durch diese Simulationen haben wir die Möglichkeit Probleme bei der Durchführung von Experimenten zu vermeiden, wie zum Beispiel die hohen Kosten für Versuche, oder aber den Aufwand bei der Entwicklung neuer Materialien. Verschiedene Ansätze und Experimente können anhand vordefinierter Eigenschaften ausprobiert werden, ohne dass das Material selbst synthetisiert werden muss, was in den meisten Fällen mit großem Aufwand verbunden ist. Vor allem erhalten wir tiefere Einblicke in die konkreten Abläufe auf atomarer Ebene, weil wir die einzelnen Teilchen in der Simulation wesentlich genauer beobachten können als es in realen Experimenten überhaupt möglich ist.

Um die Teilchen auf atomarer Ebene simulieren zu können, müssen wir deren Eigenschaften und Bewegungsabläufe genauer untersuchen. Da sich Partikel gegenseitig beeinflussen ist in diesem Zusammenhang neben den jeweiligen Eigenschaften sowohl die Lage zueinander, als auch das Verhalten der einzelnen Partikel entscheidend. Wir wollen hierbei näher auf Moleküle eingehen. Diese bestehen aus zwei oder mehreren Atomen, die durch chemische Bindungen zusammengehalten werden. Die Atome wiederum bestehen aus einem positiv geladenen Atomkern, umgeben von einem System aus negativ geladenen Elektronen, welche sich in sogenannten *Orbitalen* befinden, mit deren Hilfe wir Aufenthaltswahrscheinlichkeiten der Elektronen angeben können. Das Verhalten von Elektronen kann beispielhaft durch das sogenannte *Doppelspalt-Experiment* verdeutlicht werden.

Wir betrachten hierbei ein Elektron, welches durch zwei eng beieinander liegende Spalten bewegt und dessen Verteilung, an welcher Position einer Kontrollfläche es sich befindet, gemessen wird. Dabei ist zu erkennen, dass das Elektron sich nicht so verhält, wie man es von einem Teilchen erwarten würde. Es entstehen keine zwei voneinander getrennten Häufungen, sondern die Verteilung weist ein Interferenzmuster auf. Der Erklärungsansatz hierfür ist der *Welle-Teilchen-Dualismus*, welcher besagt, dass sich Teilchen sowohl so, wie von ihnen erwartet, verhalten, als auch Welleneigenschaften aufweisen. Das bedeutet, dass man im Gegensatz zur klassischen Mechanik, den Ort und den Impuls eines Teilchens nicht mit den Newtonschen Bewegungsgleichungen vorhersagen kann. Aus diesem Grund beschreiben wir Elektronen durch Wellenfunktionen, mit welchen wir die Wahrscheinlichkeit bestimmen, mit der sich das Elektron an einem bestimmten Ort befindet.

Das mathematische Modell der Quantenmechanik, das als Grundlage der Simulation von Bewegungsabläufen von Partikeln dient, ist die *Schrödingergleichung*, auf die wir im Folgenden näher eingehen wollen.

## 2.1. Die Schrödingergleichung

Die *Schrödingergleichung* beschreibt den Zustand von Elektronen und Kernen in Form einer partiellen Differentialgleichung, deren Lösung eine komplexwertige Wellenfunktion ist, die wir im weiteren Verlauf durch  $\psi(\mathbf{x}, t)$  beschreiben werden, dabei bezeichne  $\mathbf{x} \in \mathbb{R}^n$  den Ort und  $t$  die Zeit. Die Wahrscheinlichkeitsdichte, dass sich beispielsweise ein einzelnes Elektron zum Zeitpunkt  $t$  am Ort  $\mathbf{x} \in \mathbb{R}^3$  befindet, kann mit Hilfe des Betragsquadrates  $\psi\bar{\psi}$  bestimmt werden, wodurch nun Aussagen über die Position des Elektrons möglich werden.

Im Falle eines Mehrteilchensystems mit  $n_{el}$  Elektronen und  $n_K$  Kernen, beinhaltet der Vektor  $\mathbf{x}$  dann die einzelnen Teilchen-Koordinaten, die wir mit  $\mathbf{r} = \{r_i\}_{i=1, \dots, n_{el}}, r_i \in \mathbb{R}^3$  für  $i = 1, \dots, n_{el}$ , als Elektron-Koordinaten, beziehungsweise mit  $\mathbf{R} = \{R_j\}_{j=1, \dots, n_K}, R_j \in \mathbb{R}^3$  für  $j = 1, \dots, n_K$ , als Kern-Koordinaten bezeichnen, woraus sich  $\mathbf{x} = (\mathbf{r}, \mathbf{R})$  ergibt.

Formal wird die *zeitabhängige Schrödingergleichung* durch

$$i \frac{\partial}{\partial t} \psi(\mathbf{x}, t) = \hat{\mathbf{H}} \psi(\mathbf{x}, t), \text{ für } \mathbf{x} \in \mathbb{R}^n \text{ und } t > 0 \quad (2.1)$$

ausgedrückt, dabei sei der *Hamiltonoperator*  $\hat{\mathbf{H}}$  definiert durch

$$\hat{\mathbf{H}} = -\frac{\hbar^2}{2m} \Delta + V(\mathbf{x}, t), \quad (2.2)$$

wobei  $m$  die Masse des Teilchens und  $\hbar = h/2\pi$  eine Konstante, das sogenannte *reduzierte plancksche Wirkungsquantum* beschreibt. Die Gesamtenergie eines Systems setzt sich der klassischen Mechanik nach aus kinetischer und potentieller Energie zusammen, das bedeutet  $E_{tot} = E_{kin} + E_{pot}$ . In der Darstellung des *Hamiltonoperators* werden die kinetische Energie durch  $-\frac{\hbar^2}{2m} \Delta$  und die potentielle Energie durch  $V(\mathbf{x}, t)$  jeweils als Operator beschrieben.

### 2.1.1. Die stationäre Schrödingergleichung

Hängt der *Hamiltonoperator* nicht explizit von der Zeit ab, so lässt sich ein Produktansatz für die Wellenfunktion anwenden, der die Abhängigkeit von Ort und Zeit separiert, das bedeutet wir betrachten

$$\psi(\mathbf{x}, t) = f(t) \tilde{\psi}(\mathbf{x}).$$

Setzen wir dies in die *Schrödingergleichung* (2.1) ein, so ergibt sich

$$\begin{aligned} i \frac{\partial}{\partial t} f(t) \tilde{\psi}(\mathbf{x}) &\stackrel{!}{=} \hat{\mathbf{H}} f(t) \tilde{\psi}(\mathbf{x}) \\ \stackrel{H, \tilde{\psi} \text{ zeitunabh.}}{\Leftrightarrow} i \tilde{\psi}(\mathbf{x}) \frac{\partial}{\partial t} f(t) &= f(t) \hat{\mathbf{H}} \tilde{\psi}(\mathbf{x}) \\ \Leftrightarrow i \frac{1}{f(t)} \frac{\partial}{\partial t} f(t) &= \frac{1}{\tilde{\psi}(\mathbf{x})} \hat{\mathbf{H}} \tilde{\psi}(\mathbf{x}). \end{aligned}$$

Somit haben wir einen Teil, der nur von  $\mathbf{x}$  abhängt, und einen, der nur abhängig ist von  $t$ , daraus folgt, dass beide Seiten gleich einer Konstanten  $E$  sein müssen. Wir erhalten somit

$$\begin{aligned} i \frac{1}{f(t)} \frac{\partial}{\partial t} f(t) &= \frac{1}{\tilde{\psi}(\mathbf{x})} \hat{\mathbf{H}} \tilde{\psi}(\mathbf{x}) = E \\ \Rightarrow \hat{\mathbf{H}} \tilde{\psi}(\mathbf{x}) &= E \tilde{\psi}(\mathbf{x}). \end{aligned} \quad (2.3)$$

Da  $\hat{H}$  hierbei einen Operator darstellt, folgt, dass  $E$  ein Eigenwert sein muss, und zwar der Energieeigenwert. Aus diesem Grund fassen wir die Wellenfunktion, die diese partielle Differentialgleichung löst, als Eigenfunktion auf. Wir erhalten somit ein Eigenwertproblem und mit (2.3) die sogenannte *stationäre Schrödingergleichung*.

Da der Eigenwert den Zustand des Systems beschreibt, ist in diesem Zusammenhang aus physikalischer Sicht der kleinste Eigenwert interessant, da dieser den Zustand mit der geringsten Energie und deshalb den von den Teilchen präferierten *Grundzustand* beschreibt. Im Folgenden werden wir den zugehörigen Energieeigenwert mit  $E_0$  bezeichnen. Die *stationäre Schrödingergleichung* auf diesen Zustand zu reduzieren ist eine weitere, häufig verwendete Näherung, bei der nur noch ein einziger Zustand der Wellenfunktion betrachtet werden muss, genauerer Informationen dazu sind in [10] zu finden.

Die *Schrödingergleichung* in ihrer ursprünglichen Form analytisch zu lösen ist nur in wenigen Spezialfällen möglich und eine Approximation durch numerische Verfahren würde zu lange dauern, als dass man dies als geeignete Methode in Betracht ziehen würde.

Aus diesem Grund verwenden wir Näherungen, um ein Verfahren zu entwickeln, mit welchem wir die Lösung der *Schrödingergleichung* approximieren und dann numerisch lösen können, ohne dass sich das, anhand des Ergebnisses resultierende Verhalten der zu betrachtenden Teilchen, wesentlich verändert. Der Ansatz, den wir dazu als erstes betrachten werden ist die sogenannte *Born-Oppenheimer-Näherung*.

### 2.1.2. Die Born-Oppenheimer-Näherung

Die Grundidee der *Born-Oppenheimer-Näherung* besteht darin, den Rechenaufwand zur Lösung der *Schrödingergleichung* zu reduzieren, indem die Variablenanzahl innerhalb der Differentialgleichung verringert wird.

Grundlage dafür ist die Erkenntnis, dass Elektronen eine wesentlich kleinere Masse besitzen als Kerne, aber dennoch vergleichbaren Anziehungs- und Abstoßungskräften ausgesetzt sind. Dies hat zur Folge, dass sie stärker auf äußere Einflüsse reagieren. Aus diesem Grund wird vorausgesetzt, dass sich die Elektronen an eine Änderung der Kernpositionen innerhalb eines vernachlässigbaren Zeitraumes anpassen und sich somit zu jedem Zeitpunkt im quantenmechanischen Grundzustand befinden, welcher zur jeweiligen Kernkonfiguration gehört.

Folglich trennt man die Bewegung der Kerne von der Bewegung der Elektronen, was konkret bedeutet, dass die trägen Kerne als ortsfest betrachtet werden und nur die Wechselwirkungen der Elektronen in Abhängigkeit des Kernpotentialfeldes durch die Wellenfunktion beschrieben werden. Die separierte Wellenfunktion kann formal folgendermaßen beschrieben werden

$$\tilde{\psi}(r_1, \dots, r_{n_{el}}, R_1, \dots, R_K) \approx \tilde{\psi}_{el, \mathbf{R}}(r_1, \dots, r_{n_{el}}) \tilde{\psi}_K(R_1, \dots, R_K). \quad (2.4)$$

Die Kerne werden hierbei als klassische Teilchen betrachtet und die Kern-Kern Wechselwirkungen über die Newtonsche Bewegungsgleichung der klassischen Mechanik berechnet. Dies führt uns für jeden Kern zu einer Einsparung des Rechenaufwands der Dimension der Kern-Koordinaten, also der Dimension drei.

In der *Born-Oppenheimer-Näherung* wird somit, unter der Annahme von festen Kernkoordinaten, nur der von den Elektronen abhängige Teil betrachtet und wir erhalten die zeitunabhängige

*elektronische Schrödingergleichung*

$$\hat{\mathbf{H}}_{el,\mathbf{R}}(\mathbf{r}) \tilde{\psi}_{el,\mathbf{R}_0}(\mathbf{r}) = E_0 \tilde{\psi}_{el,\mathbf{R}_0}(\mathbf{r}),$$

wobei  $\tilde{\psi}_{el,(\mathbf{R}_0)}(\mathbf{r})$  die elektronische Wellenfunktion des *Grundzustands* beschreibt. Die Lösung der *elektronischen Schrödingergleichung* beschreibt ein effektives Potential  $V_e^{BO}$ , welches uns nach weiteren Einschränkungen zur *Born-Oppenheimer-Moleküldynamik* [10] führt, diese genügt den Gleichungen

$$\begin{aligned} m_j \frac{\partial R_j(t)}{\partial t^2} &= -\nabla_{R_j} \min_{E_0} \left\{ \int \tilde{\psi}_{el,\mathbf{R}_0}^*(\mathbf{r}) \hat{\mathbf{H}}_{el,\mathbf{R}}(\mathbf{r}) \tilde{\psi}_{el,\mathbf{R}_0}(\mathbf{r}) \right\} \\ &=: -\nabla_{R_j} V_e^{BO}(R(t)). \end{aligned}$$

Können wir dieses Gleichungssystem lösen, so ist es uns möglich die für die Moleküldynamik wesentliche Kenngröße, die Kraft  $\mathbf{F}_j(t)$  zum Zeitpunkt  $t$ , aus den Kernbewegungen mittels der Bewegungsgleichung der klassischen Mechanik

$$\mathbf{F}_j(t) = m_j \frac{\partial R_j(t)}{\partial t^2}$$

zu bestimmen.

Die *Born-Oppenheimer-Näherung* der Lösung der *Schrödingergleichung* liefert uns schon eine beträchtliche Einsparung der Kosten, in Bezug auf konkrete Berechnungen ist diese allerdings immer noch sehr komplex und kann deshalb im Allgemeinen nur numerisch mit einem sehr hohem Rechenaufwand gelöst werden. Aus diesem Grund müssen Ansätze für weitere Approximationen betrachtet werden.

### 2.1.3. Hartree-Fock-Verfahren

Ausgehend von den Annahmen der *Born-Oppenheimer-Näherung* betrachten wir zunächst das *Hartree-Fock-Verfahren* [6] zur Approximation der Lösung der *elektronischen Schrödingergleichung*. Die Herausforderung besteht darin, eine Vorschrift für die Näherung der Wellenfunktion anzugeben, um numerische Berechnungen in angemessener Zeit durchführen zu können.

Bei den *Hartree-Fock-Verfahren* werden nur Elektronen betrachtet, die einen halbzahligen Spin aufweisen, die sogenannten *Fermionen*. Die Wellenfunktion in diesen Verfahren wird in Abhängigkeit der Quantenzahlen der Teilchen des betrachteten Systems dargestellt, das bedeutet wir betrachten die Wellenfunktion

$$\psi_{el(\mathbf{R},t)}^{HF}(x_1, \dots, x_{n_{el}}) = \tilde{\psi}_{el(\mathbf{R},t)}(r_1, \dots, r_{n_{el}}).$$

Dabei beschreiben die  $x_i$ ,  $i = 1, \dots, n_{el}$  die jeweiligen Quantenzahlen für die einzelnen Teilchen des Systems, also die Informationen über die Elektronen, wie zum Beispiel den Drehimpuls oder die Orientierung des Spins. Diese Quantenzahlen sind nach dem *Pauli-Ausschlussprinzip* [20] eindeutig und treten deshalb innerhalb eines Systems nie doppelt auf.

Aus diesem Grund müssen wir eine Zustandsfunktion entwickeln, die antisymmetrisch unter der Vertauschung zweier Elektronen ist, das bedeutet formal für die Wellenfunktion der *Hartree-Fock-Verfahren*

$$\psi_{el(\mathbf{R},t)}^{HF}(x_1, x_2, x_3, \dots, x_{n_{el}}) = -\psi_{el(\mathbf{R},t)}^{HF}(x_2, x_1, x_3, \dots, x_{n_{el}}). \quad (2.5)$$

Die Näherung der Wellenfunktion im *Hartree-Fock-Verfahren* besteht darin, ihre Komplexität über einen Produktansatz zu reduzieren. Dabei wird die Gesamtwellenfunktion als Produkt von Eielektronenwellenfunktionen approximiert

$$\psi_{el(\mathbf{R},t)}^{HF}(x_1, \dots, x_{n_{el}}) = \phi_1(x_1)\phi_2(x_2)\dots\phi_{n_{el}}(x_{n_{el}}).$$

Da die Gesamtwellenfunktion die Eigenschaft der Antisymmetrie bei Vertauschung von Elektronen aufweisen muss, müssen wir ein antisymmetrisches Produkt definieren. Hierfür verwenden wir die *Slater-Determinante* [20], welche definiert ist durch

$$\phi_1(x_1) \wedge \dots \wedge \phi_{n_{el}}(x_{n_{el}}) = \frac{1}{n_{el}!} \begin{vmatrix} \phi_1(x_1) & \phi_1(x_2) & \dots & \phi_1(x_{n_{el}}) \\ \phi_2(x_1) & \phi_2(x_2) & \dots & \phi_2(x_{n_{el}}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n_{el}}(x_1) & \phi_{n_{el}}(x_2) & \dots & \phi_{n_{el}}(x_{n_{el}}) \end{vmatrix}.$$

Für die Wellenfunktion folgt somit

$$\psi_{el(\mathbf{R},t)}^{HF}(x_1, x_2, \dots, x_{n_{el}}) = \phi_1(x_1) \wedge \phi_2(x_2) \wedge \dots \wedge \phi_{n_{el}}(x_{n_{el}}).$$

Aus der *Leibniz-Formel* zur Berechnung der Determinante folgt, dass (2.5) für alle  $x_i$ ,  $i \in \{1, \dots, n_{el}\}$ , erfüllt ist.

Die *Hartree-Fock-Methode* geht demnach von der Darstellung der Wellenfunktion durch eine einzige *Slaterdeterminante* aus, wodurch die Elektronen separat betrachtet und die Elektronenkorrelationen vernachlässigt werden. Bei der Lösung für Mehrteilchensysteme, beziehungsweise bei der Betrachtung großer Moleküle, werden die Eielektronenwellenfunktionen  $\phi_i$ ,  $i = 1, \dots, n_{el}$  als *Molekülorbitale* [6] betrachtet.

Das Ziel besteht darin optimale Funktionen für die  $\phi_i$ ,  $i = 1, \dots, n_{el}$  zu bestimmen.

Um auf dieses Optimierungsverfahren näher einzugehen, benötigen wir das *Variationsprinzip* [6], welches aussagt, dass die Energie jeder approximierten Wellenfunktion größer oder gleich der Energie der exakten Lösung ist. Demzufolge sind optimalen Näherungen der  $\phi_i$  diejenigen Eielektronenwellenfunktionen, die die Gesamtenergie des Systems minimieren.

Die konkrete Berechnung erfolgt nun iterativ über die Verwendung der *Hartree-Fock-Gleichungen* [6]

$$\hat{\mathbf{F}}_i \phi_i = \hat{\mathcal{E}}_i \phi_i, \quad i = 1, \dots, n_{el},$$

wobei  $\hat{\mathbf{F}}_i$  den *Fock-Operator* darstellt, welcher einen Eielektronen-Energie-Operator beschreibt, der die Bewegung des Elektrons  $i$  in Abhängigkeit der Kernpotentialfelder charakterisiert. Die Energie der Molekülorbitale  $\phi_i$  wird in der Gleichung durch  $\hat{\mathcal{E}}_i$  ausgedrückt.

Da der *Fock-Operator* eines Teilchens jeweils von den Potentialfeldern der übrigen Teilchen des Systems abhängig ist, werden wir zur Lösung ein iteratives Verfahren, die die sogenannte *Self-Consistent-Field (SCF)-Methode* [6], verwenden.

Zunächst wird mit einem Satz gewählter Startfunktionen  $\phi_i$  begonnen, anhand welcher der *Fock-Operator* für diese Funktionen bestimmt werden kann. Daraufhin wird durch das Lösen der *Hartree-Fock-Gleichungen* die resultierende Energie berechnet und anschließend wieder die somit berechenbaren Einelektronenfunktionen, also die  $\phi_i$ , die in diesem Verfahren als *SCF-Orbitale* bezeichnet werden.

Diese Iteration wird solange wiederholt, bis ein stationärer Zustand angenommen wird, das bedeutet, dass die Ergebnisse der Energie und der *SCF-Orbitale* sich nicht mehr verändern und somit das Ergebnis selbstkonsistent ist.

Da die *Hartree-Fock-Methoden* die, häufig bedeutungsvollen, Elektronenkorrelationen vernachlässigen ist sie für Anwendungen nur in geringem Maße geeignet, trotzdem können sie als Basis für eine Reihe von Verfahren verwendet werden, in denen die Korrelationen der Elektronen mit einbezogen werden. Aus diesem Grund werden wir im Folgenden die sogenannten *Post-Hartree-Fock-Verfahren* betrachten.

#### 2.1.4. Post-Hartree-Fock-Verfahren

*Post-Hartree-Fock-Verfahren* verwenden die Resultate der *Hartree-Fock-Verfahren* und verbessern sie im Hinblick auf die Berücksichtigung der Elektronenkorrelationen. Es wurden verschiedene wellenfunktionenbasierende Strategien entwickelt um eine Modellapproximation der *Schrödingergleichung* zu bestimmen. Die Elektronenwellenfunktion wird dabei nicht mehr nur über eine, sondern über Kombinationen verschiedener *Slaterdeterminanten* dargestellt. Die einzelnen Verfahren unterscheiden sich in ihrem Aufwand und ihrer Rechengenauigkeit und werden klassischerweise für Berechnungen der Zustände von Molekülen verwendet. Im Folgenden sollen die Grundlagen einiger ausgewählter *Post-Hartree-Fock-Verfahren* beschrieben werden.

#### Konfigurationswechselwirkungs-Verfahren

Bei den *Konfigurationswechselwirkungs(CI)-Verfahren* [13] wird anstelle einer einzigen *Slaterdeterminante* zur Bestimmung der Gesamtwellenfunktion, eine Linearkombination verschiedener *Slaterdeterminanten* verwendet somit zu einem additiven Ansatz führt. Durch  $\psi_{ab\dots}^{rs\dots}$  seien sogenannte „angeregte“ oder substituierte *Slaterdeterminanten* beschrieben. Diese werden, ausgehend von einer *Slaterdeterminanten*  $\psi_o$ , die nur besetzte Orbitale enthält, dadurch verändert, dass das jeweils a-te, b-te, ... besetzte Orbital durch das s-te, r-te, ... virtuelle Orbital ersetzt wird. Dies entspricht einer Anregung eines Elektrons in ein energiereicheres Orbital. Für weitere Details dieses Verfahrens sei verwiesen auf [13]. Für die Gesamtwellenfunktion des *CI-Verfahrens* gilt demnach

$$\psi_{el(\mathbf{R},t)}^{CI} = c_0 \psi_o + \sum_{a,r} c_r^a \psi_r^a + \sum_{a<b,r<s} c_{ab}^{rs} \psi_{ab}^{rs} + \sum_{a<b<c,r<s<t} c_{abc}^{rst} \psi_{abc}^{rst} + \dots$$

Für  $\psi_o$  wird dabei in der Regel die *Hartree-Fock-Slaterdeterminante*  $\psi_{el(R_1,\dots,R_K)}^{HF}$  angenommen [11]. Betrachten wir nicht den gesamten unendlich dimensionalen Orbitalraum, sondern nur einen endlichen Teilraum, der den Orbitalraum mit  $M$  Basisfunktionen nähert, so erhalten wir bei einer Anzahl von  $n_{el}$  Elektronen  $\binom{M}{n_{el}}$  Möglichkeiten zur Bildung einer *Slaterdeterminanten* [13].

Für  $\psi_o$  wird in der Regel die *Hartree-Fock-Slaterdeterminante*  $\psi_{el(\mathbf{R},t)}^{HF}$  angenommen [11].

### Møller-Plesset-Verfahren

Das *Møller-Plesset(MP)-Verfahren* [6, 12] gehört zu den störungstheoretischen Methoden der *Post-Hartree-Fock-Verfahren*. Der Hamiltonoperator der *stationären Schrödingergleichung* erhält dabei einen zusätzlichen Störungsoperator  $\lambda\hat{\mathbf{V}}$ , der folgendermaßen in den ungestörten Hamiltonoperator  $\hat{\mathbf{H}}$  eingefügt wird

$$\hat{\mathbf{H}}_{err}(\lambda) = \hat{\mathbf{H}} + \lambda\hat{\mathbf{P}}, \quad \lambda \in \mathbb{R}.$$

Die gestörte Wellenfunktion  $\psi_\lambda$  und die gestörte Energie  $E_\lambda$  können nun mit Hilfe von Taylorreihen entwickelt werden

$$\begin{aligned} \psi_\lambda &= \psi_0 + \lambda\psi_1 + \lambda^2\psi_2 + \lambda^3\psi_3 + \dots = \sum_{k=0}^{\infty} \lambda^k \psi_k \\ E_\lambda &= E_0 + \lambda E_1 + \lambda^2 E_2 + \lambda^3 E_3 + \dots = \sum_{k=0}^{\infty} \lambda^k E_k, \end{aligned}$$

wobei die Indizes die jeweilige Ordnung der Störung angeben [6].

Setzen wir dies nun in die *stationäre Schrödingergleichung* (2.3) ein so erhalten wir

$$(\hat{\mathbf{H}} + \lambda\hat{\mathbf{V}}) \left( \sum_{k=0}^{\infty} \lambda^k \psi_k \right) = \left( \sum_{k=0}^{\infty} \lambda^k E_k \right) \left( \sum_{k=0}^{\infty} \lambda^k \psi_k \right).$$

Wir unterscheiden nun die verschiedenen *MP-Verfahren* anhand der Ordnung der Taylorreihenentwicklung. Die gestörte Energie des Verfahrens bei Abbruch der Taylorentwicklung nach der ersten Ordnung (*MP1-Verfahren*) ergibt sich aus der Summe von  $E_1$  und  $E_2$  und entspricht der *Hartree-Fock-Energie*.

Die Elektronenkorrelationen werden deshalb erst bei der Betrachtung ab der zweiten Ordnung einbezogen. Der Rechenaufwand steigt hierbei jedoch mit der Ordnungszahl stark an und das Ergebnis kann bei signifikanten Fehleroperatoren schnell Oszillationen aufweisen. Deshalb ist die am häufigsten verwendete Methode das *MP2-Verfahren*, also der Abbruch der Taylorreihe nach der zweiten Ordnung, da es eine angemessene Balance zwischen Rechenaufwand und Genauigkeit aufweist.

Da beim *MP2-Verfahren* nur die Paarkorrelationen, also Störungsterme zweiter Ordnung, betrachtet werden ist es sinnvoll ein Näherungsverfahren zu ermitteln, welches es ermöglicht die Störungsterme auch in höheren Ordnungen mit einzubeziehen. Eine Möglichkeit hierfür bieten die sogenannten *Coupled-Cluster-Verfahren*.

### Coupled-Cluster-Verfahren

Die *Coupled-Cluster(CC)-Verfahren* [12] sind eine weitere Verbesserung der Modellapproximation der *Schrödingergleichung* und beschreiben die Wellenfunktion explizit über einen exponentiellen Ausdruck, abhängig von einer beliebigen Referenzdeterminante. Hierfür kann beispielsweise die Wellenfunktion als *Slaterdeterminante* aus dem *Hartree-Fock-Verfahren* verwendet

werden. Formal bedeutet dies, wir beschreiben die Wellenfunktion  $\psi_{el(\mathbf{R},t)}^{CC}$  des *Coupled-Cluster-Verfahrens* durch

$$\psi_{el(\mathbf{R},t)}^{CC} = e^{\hat{\mathbf{T}}}\psi_{el(\mathbf{R},t)}^{HF}$$

mit  $\hat{\mathbf{T}} = \sum_n \hat{\mathbf{T}}_n$ ,

wobei  $\hat{\mathbf{T}}_n$  den *Cluster-Operator* darstellt, der die Anregung von jeweils n Elektronen der *Slaterdeterminante* beschreibt. Somit werden auch die Elektronenkorrelationen für mehrere Elektronen berücksichtigt.

Die *Coupled-Cluster-single-double(-triple)(CCSD(T))-Methode* beispielsweise berücksichtigt die Wechselwirkungen zwischen einzelnen Elektronen und Elektronenpaaren mittels *Coupled-Cluster-Verfahren*, also  $\hat{\mathbf{T}} = \hat{\mathbf{T}}_1 + \hat{\mathbf{T}}_2$ , und die Korrelationen zwischen Elektronentripeln basierend auf störungstheoretischen Methoden.

Mit Hilfe der *Post-Hartree-Fock-Methoden* lassen sich demnach die Energiewerte für Moleküle approximieren, die Aussagekraft dieser Näherungen ist jedoch abhängig von der Genauigkeit der im voraus bestimmten *Hartree-Fock-Determinante*, welche innerhalb des jeweiligen Verfahrens verwendet wird.

Eine Alternative zu den *Post-Hartree-Fock-Verfahren* stellt die *Dichtefunktionaltheorie(DFT)* dar, welche hauptsächlich für Festkörper und periodische Randbedingungen verwendet wird wie es beispielsweise bei Kristallen der Fall ist. Diese Verfahren sollen im Rahmen dieser Arbeit allerdings nicht näher erläutert werden, für weitere Details sei auf [6] verwiesen.

Diese Approximationen ermöglichen es uns die Lösung der *Schrödingergleichung* durch diese nur von Parametern abhängige Potentialgleichung zu approximieren, welche wir im Gegensatz zur ursprünglichen Form durch numerische Verfahren lösen können.

Anhand der berechneten Energiewerte können wir dann eine Energiefunktion approximieren, aus welcher sich ein Potential entwickeln lässt.

### 2.1.5. Die Bestimmung von Potentialflächen

In Rahmen dieser Arbeit sind wir daran interessiert eine Methode zur effizienten Bestimmung der *Born-Oppenheimer-Potentialfläche* für Moleküle zu entwickeln.

Zur Bestimmung dieser Potentialfläche betrachten wir das in Abschnitt 2.1.1 beschriebene Potential im *Grundzustand* und setzen die *Born-Oppenheimer-Näherung* voraus.

Da Potentialflächen eine hochdimensionale Funktion beschreiben, die vom „Fluch der Dimensionen“ betroffen sind muss ein Ansatz verwendet werden um die Dimension des Problems zu reduzieren.

Angenommen, wir können die Lösung der *stationären elektronischen Schrödingergleichung* für bestimmte Abstände von Kernen im Raum berechnen, beispielsweise durch eines der bereits beschriebenen Approximationsverfahren.

Durch den *HDMR-Ansatz* [16] wird das hochdimensionale Problem, die potentielle Energie von Molekülen zu bestimmen, über einen additiven Ansatz vereinfacht und somit der „Fluch der Dimensionen“ gebrochen.

Der Grundgedanke besteht darin, die Eigenschaften der Moleküle, die durch ein mehrdimensionales System repräsentiert werden können, auf die Wechselwirkung bestimmter Paare, Tripel, Quadrupel, ... zu reduzieren. .

Konkret wird dabei mit den Korrelationen zwischen den jeweiligen Teilchen unabhängig voneinander begonnen, darauf folgen die Korrelationen von Paaren, von Tripeln, Quadrupeln,... bis hin zu den Korrelationen zwischen allen beteiligten Teilchen. Anschließend werden die einzeln berechneten Komponenten aufaddiert, woraus sich die Gesamtpotentialfläche ergibt.

Seien  $\mathbf{R} = (R_1, \dots, R_K)$  die Koordinaten der Kerne aller Teilchen des zu untersuchenden Moleküls, dann wird diese Approximation formal beschrieben durch

$$V_e^{BO} \approx V_e^{approx}(\mathbf{R}) = V_0 + \sum_l^K V_1(R_l) + \sum_{l<j}^K V_2(R_l, R_j) + \sum_{l<j<k}^K V_3(R_l, R_j, R_k) + \dots + V_K(\mathbf{R}), \quad (2.6)$$

wobei die  $V_l$ ,  $l = 0, \dots, K$  das Potential für die jeweilige Anzahl an betrachteten Kernen darstellt.

Das Paarpotential  $V_2(R_1, R_2)$  kann beispielsweise durch eine Abstandsfunktion der Form  $V_2(\|R_1 - R_2\|)$  ausgedrückt werden, wodurch sich das Problem auf die Bestimmung einer eindimensionalen Funktion beschränkt.

In diesem Zusammenhang sind in vielen physikalischen und chemischen Anwendungen die Terme niedriger Ordnung signifikant für die resultierende Energie, weshalb nur diese Terme in den Berechnungen berücksichtigt werden.

Sei  $q$  der Wert, sodass nur die Potentialfunktionen  $V_l$ ,  $l = 0, \dots, q$  betrachtet werden, dann reduziert sich das Problem der Approximation einer  $K$ -Partikel-Funktion auf ein Problem von  $q$  Partikelfunktionen, mit  $q \ll K$ .

Da die beschriebene *HDMR-Zerlegung* jedoch nicht eindeutig ist wird nach einer solchen gesucht, in der die Ordnung der zu betrachtenden Terme möglichst niedrig ist, um eine gute Approximation zu erzielen.

Jeder einzelne Summand der Approximation (2.6) der Potentialfläche müssen wir mit Hilfe der *elektronischen Schrödingergleichung* bestimmen, was es uns unmöglich macht, eine exakte Lösung dieses Problems anzugeben und selbst Näherungen benötigen viel zu lange Laufzeiten.

Können wir in konkreten Anwendungen die Lösungen zu bestimmten Kernkonfigurationen bestimmen, so ist es anhand dieser Werte möglich die einzelnen Potentiale mit Hilfe von *Force-matching* oder dem *Fitten analytischer Potentiale* an quantenmechanische oder experimentelle Daten [10] zu approximieren.

Das Ziel dieser Arbeit ist es, eine Methode für das *Fitten der analytischen Potentiale* an quantenmechanische Daten zu entwickeln, also die Potentialflächen bestimmter Moleküle so zu approximieren, dass man die potentielle Energie nur an bestimmten Punkten berechnet und die Potentiale daraufhin durch das Fitten einer Potentialfunktion approximiert. Solch ein Verfahren ermöglicht eine enorme Zeitersparnis im Hinblick auf die Moleküldynamik.

Diese Potentiale werden in der Moleküldynamik benötigt um die wirkenden Kräfte berechnen zu können, da sie die Grundlage für die Simulation der Bewegung der Moleküle darstellen. Wir benötigen demnach ein Verfahren, welches die Positionen der einzelnen Teilchen im Verlauf der Zeit bestimmt und werden deshalb im Folgenden auf das *Verlet-Verfahren* zur Zeitintegration eingehen.

## 2.2. Das Verlet-Verfahren

Die Grundlage für die Simulation von Molekülen ist die Vorhersage ihrer Bewegungsabläufe, die über die Lösung der Newtonschen Bewegungsgleichungen (2.7) bestimmt werden können.

Das bedeutet, wir müssen ein Verfahren entwickeln, das es uns ermöglicht zu jedem Zeitpunkt die einwirkenden Kräfte auf die einzelnen Teilchen zu berechnen. Wir werden im Folgenden das *Verlet-Verfahren* zur Zeitintegration betrachten.

Die Kerne werden in Abhängigkeit der Elektronen im *Grundzustand* betrachtet und wir in der *Born-Oppenheimer-Näherung* davon aus, dass diese über die Regeln der klassischen Mechanik berechnet werden können. Seien  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$  Positionen der Kerne der einzelnen Teilchen des zu untersuchenden Moleküls und  $t$  die Zeit, dann gilt für die Berechnung der Kraft nach den Newtonschen Bewegungsgleichungen

$$m_i \frac{\partial^2 \mathbf{x}_i(t)}{\partial t^2} = \mathbf{F}_i(t) = -\nabla_{\mathbf{x}_i} V(\mathbf{x}_1, \dots, \mathbf{x}_N). \quad (2.7)$$

Da dies ein kontinuierliches Problem darstellt, ist es nötig die Gleichung zunächst in geeigneter Form zu diskretisieren. Wir werden im Folgenden das *Verlet-Verfahren* zur Zeitdiskretisierung betrachten. Dazu verwenden wir diskrete Differenzenoperatoren wie in [10], womit für die Berechnung der Kraft nach dem zentralen Differenzenquotienten der zweiten Ableitung zum Zeitpunkt  $t_n$  folgt

$$m_i \frac{1}{\delta t^2} \left( \mathbf{x}_i(t_{n+1}) - 2\mathbf{x}_i(t_n) - \mathbf{x}_i(t_{n-1}) \right) \approx \mathbf{F}_i(t_n),$$

wobei die Positionen und die Kraft vom jeweiligen Zeitpunkt abhängig sind. Daraus ergibt sich für die Position des  $i$ 'ten Teilchens beim Übergang auf den nächsten Zeitschritt  $n+1$  mit Auflösen nach  $\mathbf{x}_i(t_{n+1})$

$$\mathbf{x}_i(t_{n+1}) \approx 2\mathbf{x}_i(t_n) - \mathbf{x}_i(t_{n-1}) + \delta t^2 \cdot \frac{\mathbf{F}_i(t_n)}{m_i}.$$

Hierbei ist zu beachten, dass große Rundungsfehler auftreten können, da der Faktor  $\delta t^2$  den Term  $\delta t^2 \cdot \frac{\mathbf{F}_i(t_n)}{m_i}$  stark verkleinert und auf den, von  $\delta t^2$  unabhängigen, restlichen Teil aufaddiert.

Zur Bestimmung der kinetischen Energie ist es notwendig, die Geschwindigkeit separat zu berechnen. Diese ergibt sich mit Hilfe des diskreten zentralen Differenzenoperators zu

$$\mathbf{v}_i(t_n) \approx \frac{\mathbf{x}_i(t_{n+1}) - \mathbf{x}_i(t_{n-1}))}{2\delta t}. \quad (2.8)$$

Der Nachteil dieser Methode besteht in der Berechnung der Geschwindigkeit über zwei Zeitschritte.

Eine Methode, welche die Geschwindigkeit und die Orte zum gleichen Zeitpunkt berechnet ist der *Geschwindigkeits-Verlet-Algorithmus*.

Aus der Approximation der Geschwindigkeit folgt für die Berechnung der Position von  $\mathbf{x}_i$  zum Zeitpunkt  $t_{n+1}$

$$\mathbf{x}_i(t_{n+1}) \approx \mathbf{x}_i(t_n) + \delta t \mathbf{v}_i(t_n) + \delta t^2 \cdot \frac{\mathbf{F}_i(t_n)}{2m_i} \quad (2.9)$$

**Algorithmus 1:** Geschwindigkeits-Verlet-Algorithmus

**Input :** Anfangspositionen  $\mathbf{x}$ , Anfangsgeschwindigkeiten  $\mathbf{v}$ , Endzeit  $t_{end}$ ,  
Zeitschrittweite  $\delta t$

**Output :** Alle Positionen  $\mathbf{x}$ , Geschwindigkeiten  $\mathbf{v}$  und Kräfte  $\mathbf{F}$  und die potentielle  $E_{pot}$   
und kinetische Energie  $E_{kin}$  zu jedem Zeitschritt

Setze  $t=0$ .

Berechne das Potential  $V(\mathbf{x})$  aus den Anfangspositionen  $\mathbf{x}(0)$  zum Zeitpunkt 0.

**for**  $i=1,\dots,N$  **do**

Berechne die Kräfte  $\mathbf{F}(0)$  zum Anfangszeitpunkt aus der partiellen Ableitung von  
 $V(\mathbf{x})$ ,

$$\mathbf{F}_i^{old} = -\nabla_{x_i} V(x_1, \dots, x_N).$$

**end**

**while**  $t < t_{end}$  **do**

$t = t + \delta t$

**for**  $i=1,\dots,N$  **do**

Berechne die neuen Positionen  $\mathbf{x}(t)$  aus der Kraft  $\mathbf{F}(t - \delta t)$ , der Geschwindigkeit  
 $\mathbf{v}(t - \delta t)$  und den Positionen  $\mathbf{x}(t - \delta t)$  zum vorherigen Zeitpunkt,

$$\mathbf{x}_i = \mathbf{x}_i + \delta t \mathbf{v}_i + \delta t^2 \cdot \frac{\mathbf{F}_i^{old}}{2m_i}.$$

**end**

Berechne das Potential  $V(\mathbf{x})$  aus den neuen Positionen  $\mathbf{x}(t)$ .

**for**  $i=1,\dots,N$  **do**

Berechne die neuen Kräfte  $\mathbf{F}(t)$  aus der partiellen Ableitung von  $V(\mathbf{x})$ ,

$$\mathbf{F}_i = -\nabla_{x_i} V(x_1, \dots, x_N).$$

**end**

**for**  $i=1,\dots,N$  **do**

Berechne die neuen Geschwindigkeiten  $\mathbf{v}(t)$  aus den Geschwindigkeiten  $\mathbf{v}(t - \delta t)$   
und der Kraft  $\mathbf{F}(t - \delta t)$  zum vorherigen Zeitpunkt und der Kraft  $\mathbf{F}(t)$  zum  
aktuellen Zeitpunkt,

$$\mathbf{v}_i = \mathbf{v}_i + \frac{(\mathbf{F}_i + \mathbf{F}_i^{old})\delta t}{2m_i}.$$

**end**

Berechne die kinetische Energie  $E_{kin}$  und die potentielle Energie  $E_{pot}$

$\mathbf{F}_i^{old} = \mathbf{F}_i$

return  $t$ ,  $\mathbf{x}(t)$ ,  $\mathbf{v}(t)$ ,  $\mathbf{F}(t)$ ,  $E_{kin}$ ,  $E_{pot}$

**end**

Setzen wir dies in die Approximation (2.8) von  $\mathbf{v}$  ein, so erhalten wir für die Geschwindigkeit zum Zeitpunkt  $t_n$

$$\mathbf{v}_i(t_n) \approx \frac{\mathbf{x}_i(t_n) - \mathbf{x}_i(t_{n-1})}{\delta t} + \frac{\mathbf{F}_i(t_n)}{2m_i} \delta t$$

Um nun die Geschwindigkeit zum Zeitpunkt  $t_{n+1}$  zu berechnen betrachten wir

$$\mathbf{v}_i(t_{n+1}) + \mathbf{v}_i(t_n) \approx \frac{x_i(t_{n+1}) - x_i(t_{n-1})}{\delta t} + \frac{(\mathbf{F}_i(t_{n+1}) + \mathbf{F}_i(t_n)) \delta t}{2m_i},$$

und durch Einsetzen von (2.8) folgt damit

$$\mathbf{v}_i(t_{n+1}) \approx \mathbf{v}_i(t_n) + \frac{(\mathbf{F}_i(t_{n+1}) + \mathbf{F}_i(t_n)) \delta t}{2m_i}. \quad (2.10)$$

Über die Terme (2.9) und (2.10) können wir die Positionen und Geschwindigkeiten, und damit auch alle daraus ableitbaren Größen wie die Kraft, die potentielle und kinetische Energie bestimmen, sie ergeben zusammen den *Geschwindigkeits-Verlet-Algorithmus* (Algorithmus 1).

Zur Berechnung der neuen Geschwindigkeit benötigen wir nach (2.10) sowohl die Kraft des vorherigen, als auch des aktuellen Zeitschrittes, was zusätzlichen Speicheraufwand im Gegensatz zur Standardform des *Verlet-Verfahrens* verursacht. Allerdings akzeptieren wir diesen zusätzlichen Aufwand in Anbetracht der Verbesserung des Ergebnisses.

### 2.3. Das Programm PSI3 zur Berechnung quantenmechanischer Daten

Das Programm *PSI3* [2] stellt eine C und C++ Implementation für quantenmechanische *ab initio* Berechnungen dar, es ist uns dadurch möglich bestimmte Eigenschaften von kleinen bis durchschnittlich großen Molekülen zu bestimmen. Innerhalb dieses Programms werden die bisher beschriebenen und außerdem weitere Verfahren zur Bestimmung einer approximativen Lösung der *stationären Schrödingergleichung* verwendet.

Wir haben somit die Möglichkeit Ergebnisse aus der Quantenchemie über Berechnungen auf dem Computer, anstelle von realen Experimenten, zu erfassen.

Zu den implementierten Verfahren zählen die beschriebenen *Hartree-Fock-Methoden*, determinantenbasierende *Configuration-Interaction-Verfahren* darunter CASSCF, RAS-CI, Full CI, *Møller-Plesset-Verfahren* MP2 und MP2-R12 und *Coupled-Cluster-Verfahren* CC2, CCSD, CCSD(T), und CC3 mit RHF, ROHF, UHF darüber hinaus auch Verfahren der *Dichtefunktionaltheorie(DFT)*.

Da *PSI3* auch dafür geeignet ist auf Parallel-Rechnern gestartet zu werden, können viele Rechnungen gleichzeitig durchgeführt und die Rechenzeit somit stark verbessert werden.

Im Rahmen dieser Arbeit benötigen wir zur Bestimmung der *Born-Oppenheimer-Potentialfläche* die Energien zwischen einzelnen Atomen in Abhängigkeit ihres Abstands im Raum zueinander. Diese Energiewerte können wir mit Hilfe von *PSI3* für einzelne Abstände bestimmen und damit für ein Verfahren zur Bestimmung des Potentials verwenden. Da im Hinblick auf die Moleküldynamik mit Hilfe des *Verlet-Verfahrens* auch die Kräfte, und somit die Gradienten zu den jeweils betrachteten Positionen, benötigt werden, sollen diese auch bestimmt werden.

Die Bestimmung der Potentialfläche nach (2.6) benötigt die einzelnen Potentialkurven für die jeweilige Anzahl zu betrachtender Kerne. Würden wir versuchen dieses Problem allein anhand der Berechnungen über *PSI3* zu lösen wären die resultierenden Laufzeiten viel zu lang. Es müssten alle Abstände betrachtet und deren Energie und Gradienten bestimmt werden und das jeweils für eine aufsteigende Anzahl an Teilchen immer wieder von neuem.

Aus diesem Grund wird im weiteren Verlauf ein Verfahren beschrieben um anhand weniger Ausgewerteter Energie- und Gradientenwerte effektiv eine gute Approximation der Potentialkurve und den zugehörigen Kräften bestimmen zu können. Dazu soll im Folgenden ein Verfahren aus dem Bereich des *maschinellen Lernens* erläutert werden.



### 3. Maschinelles Lernen

Lernen erfordert Übung und Training, mit genügend Geduld bereitet es Menschen und Tieren keine Probleme Verhaltensweisen, Wissen und Fertigkeiten zu behalten und auf ähnliche Situationen anzuwenden. Diese Fähigkeit, anhand von Erfahrungen und Beispielen zu lernen, erscheint deshalb als selbstverständlich, jedoch stellt es sich als Herausforderung dar derartige Prozesse mit Hilfe von Algorithmen zu beschreiben.

*Maschinelles Lernen* beschreibt die Entwicklung von Methoden, die Lernprozesse auch auf Maschinen übertragen können. Der Begriff des Lernens wird in diesem Zusammenhang als der Gewinn von Informationen aus Daten interpretiert. Die Grundlage dafür, dass diese Interpretation als sinnvoll erscheint besteht in der Annahme, dass die Daten nicht zufällig entstehen, sondern dass ein bestimmtes Muster zugrunde liegt. Sobald dieses erkannt wird, können Gesetzmäßigkeiten für die Entstehung von Daten hergeleitet und diese entweder analysiert oder auf neue, ähnliche Vorgänge angewendet werden.

Im Unterschied zum *Data Mining* geht es beim *maschinellen Lernen* um die Suche von bekannten Mustern in einer neuen Datenmenge und nicht um die Suche nach noch unbekanntem, neuen Mustern. Obwohl die zugehörigen Algorithmen verschiedene Problemstellungen behandeln, können sie trotzdem häufig zur Lösung von Problemen aus beiden Bereichen verwendet werden.

Wir werden im Folgenden den *Gauß-Prozess* als ein Verfahren aus dem Bereich des *Maschinellen Lernens* betrachten, welches es uns ermöglicht anhand ausgewählter Eingabedaten, eine Funktion zu lernen und zusätzlich eine Aussage über die Genauigkeit des Resultats zu machen.

Wir werden außerdem die Anwendung dieses Verfahrens in Bezug auf die Bestimmung von Potentialkurven und deren Ableitung beschreiben, indem empirische Daten als Information über die Funktion eingebunden werden.

Zunächst wird ein Überblick über die Klasse von Algorithmen gegeben, zu denen dieses Verfahren gehört und welche Vorteile wir aus deren Verwendung ziehen können.

#### 3.1. Grundlagen von Lernalgorithmen

Im *Maschinellen Lernen* werden Algorithmen eingesetzt, welche aus *Trainingsdaten* lernen und entweder die gewonnene Information über diese Daten bereitstellen oder es erlauben mit Hilfe dieser Informationen Aussagen über unbekannte Daten zu treffen. Dabei unterscheiden sie sich in ihrer Vorgehensweise, je nachdem wie die Eingabedaten geartet sind, in welcher Struktur und mit welchem Informationsgehalt diese vorkommen und welches Ergebnis gewünscht ist.

Im Wesentlichen werden drei verschiedene Formen von *maschinellem Lernen* unterschieden, welche jeweils von einem bestimmten Modell zur Beschreibung des Problems ausgehen.

Algorithmen für *überwachtes Lernen* [3] erhalten die *Trainingsdaten* in Form von Eingabe- und den zugehörigen Ausgabe-Werten als Beispiele und es sollen die Beziehungen zwischen

diesen Daten gelernt werden, um Vorhersagen für neue Eingabewerte in Form einer Funktion darzustellen. Die *Trainingsdaten* dienen hierbei als eine Art *Feedback* für die Vorhersage, weil man für diese die richtige Lösung schon kennt und mit der vom Lern-Algorithmus entwickelten Lösung vergleichen, beziehungsweise „überwachen“ kann.

Verfahren, welche zum überwachten Lernen gehören, sind beispielsweise die *Regression* und die *Klassifikation* [3], wobei diese sich in der Darstellungsart der Zielwerte unterscheiden. Die Regression gibt die Ausgabedaten als reellwertige Zahlenwerte aus, die Klassifizierung in Form eines sogenannten *Klassencodes*, welcher die Ausgabewerte in definierte Klassen einteilt.

*Unüberwachtes Lernen* [3] beschäftigt sich mit der Analyse von Strukturen in den *Trainingsdaten*, welche nur aus Eingabedaten bestehen. Hierbei können die Resultate nicht bewertet werden wie beim überwachten Lernen, da keine Aussagen über die Ausgabewerte vorliegen, der Algorithmus verläuft in Folge dessen „unüberwacht“. Der Grundgedanke besteht darin die Eingabedaten auf Gemeinsamkeiten zu untersuchen, diese über verschiedene Klassen zu definieren und anschließend die Daten daraufhin zu sortieren. Solch ein Verfahren wird als *Clustering* bezeichnet.

Das *bestärkende Lernen* [3] besitzt einen Entscheidungsträger, den sogenannten *Agenten*, der auf die jeweils durchgeführte Aktion durch eine Zustandsveränderung von seiner Umwelt eine Reaktion in Form von Belohnung oder Bestrafung erhält. Das Ziel besteht darin, dass der *Agent* anhand vorangegangener Reaktionen den Nutzen von beliebigen Aktionen erlernt. Anhand des erlernten Wissens wird der Zustand jeweils so angepasst, dass die erwartete Gesamtbelohnung maximiert wird.

In dieser Arbeit befassen wir uns mit der Bestimmung von Energiefunktionen von Potentialen aus der Moleküldynamik. Dabei soll für Partikel die potentielle Energie abhängig von ihrer Position im Raum bestimmt werden. Die Berechnung dieser Energie ist allerdings selbst mittels Näherungen an die Lösung der *Schrödingergleichung*, wie in Kapitel 2 beschrieben, immer noch sehr aufwendig und teuer, weshalb es wünschenswert ist, eine Methode zu entwickeln, sodass nur eine möglichst kleine Anzahl dieser Punkte konkret berechnet werden muss.

Unter der Verwendung einiger, durch das Lösen der *Schrödingergleichung*, berechneter Werte für verschiedene Abstände, wollen wir mittels Lernalgorithmen die Energie auch für beliebige Werte nähern und somit eine Energiefunktion über eine Form von *Extrapolation*, das bedeutet die Bestimmung von Funktionswerten an beliebigen, unbekanntenen Stellen, approximieren. Da wir als *Trainingsdaten* jeweils die Abstände als Eingabewerte und die zugehörige Energie als Ausgabewerte erhalten, bietet sich der Lernalgorithmus der Regression an und wir erhalten damit eine Möglichkeit die Energiefunktion mittels *unüberwachtem Lernen* zu fitten und damit zu approximieren.

#### 3.1.1. Regressionen im überwachten Lernen

Die Regression beschreibt ein Verfahren des überwachten Lernens welches die Zusammenhänge zwischen einer abhängigen und einer oder mehrerer unabhängiger Variablen aufzeigt. Dabei werden die Eingabedaten als unabhängige und die Ausgabedaten als abhängige Variablen bezeichnet.

Gegeben einer endlichen Menge an *Trainingsdaten* wird die Regression somit verwendet um die Beziehungen zwischen den Ein- und Ausgabedaten quantitativ zu beschreiben und gegebenenfalls die Werte der Ausgabedaten an noch unbekanntenen Stellen vorauszusagen. Damit ermöglicht

uns die Regression Prognosen über die zugrunde liegende Funktion aufzustellen.

Gegeben sei eine Menge von  $n$  Paaren an Trainingsbeispielen

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\},$$

mit  $x_i \in X$  als Eingabewerte und  $y_i \in Y$  als Ausgabewerte für  $i = 1, \dots, n$ .

Da die gegebenen Daten fehlerbehaftet sein dürfen, beschreiben wir den Zusammenhang von Ein- und Ausgabewerten mit

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \dots, n, \quad (3.1)$$

wobei  $f : X \rightarrow Y$  die Funktion darstellt, die den exakten Funktionswerten zugrunde liegt.

Ziel der Regression ist es, eine Funktion zu erlernen, die möglichst nah an der exakten Funktion  $f$  liegt, indem diese iterativ an die *Trainingsdaten* angepasst wird.

In diesem Zusammenhang gibt es verschiedene sukzessive Approximationsmethoden, wie zum Beispiel die *lineare*, die *nicht-lineare* oder die *nicht-parametrische* Regression [3]. Welche Methode verwendet wird hängt davon ab, welche Informationen über die Funktion  $f$  vorliegen. Abhängig von diesen Informationen wird ein *Hypothesenraum* erstellt, aus welchem die zu bestimmende Funktion stammt.

Um die diese näher zu beschreiben, verwenden wir eine Modellfunktion für die Annäherung an  $f$ , welche wir mit

$$g(\mathbf{x}; \boldsymbol{\rho}), \quad (3.2)$$

beschreiben werden und welche abhängig ist von sogenannten *freien Parametern*  $\boldsymbol{\rho} = (\rho_1, \dots, \rho_m)$ ,  $m \in \mathbb{N}$ .

Die Modellfunktion  $g(\cdot)$  repräsentiert durch zusätzliches Wissen erhaltene Informationen über die zugrundeliegende Funktion. Das bedeutet, je nachdem welche Approximationsmethode verwendet wird, erhält die Funktion  $g$  die dazu entsprechenden Eigenschaften.

Die lineare Regression setzt einen linearen Zusammenhang bezüglich der Parameter von den Eingabe- auf die Ausgabewerte voraus, die nicht-lineare nur einen von den Parametern abhängigen. Die nicht-parametrische Regression verläuft ohne konkrete Informationen über die Funktion  $f$  und bestimmt die Näherung allein durch die gegebenen *Trainingsdaten*.

Der maschinelle Lern-Algorithmus variiert die Parameter  $\rho_i$  nun so, dass der Näherungsfehler  $\epsilon_i$  schrittweise minimiert wird, das heißt ein Optimierungsverfahren zur Bestimmung der Parameter angewendet wird. Solch ein Verfahren wird als *Ausgleichsrechnung* oder *Fitting* bezeichnet.

Ein Beispiel für solch eine Fehlerminimierung ist die *Methode der kleinsten Quadrate* [22], bei der die Summe der Quadrate der Differenz zwischen den Schätz- und den exakten Funktionswerten minimiert wird, das heißt wir betrachten

$$\min_{\boldsymbol{\rho}} \sum_{i=1}^n (f(x_i) - g(x_i, \rho_i))^2.$$

Bei der Erstellung von Messdaten kann es vorkommen, dass durch zufällig auftretende Fehler einige Daten einen falschen Wert annehmen und das zu erschließende Muster damit in eine falsche Richtung gelenkt wird, solche Daten werden als *Ausreißer* bezeichnet. Das Regressionsverfahren gegen solche *Ausreißer* robust zu machen, ist Aufgabe von sogenannten *robusten Schätzverfahren* [22].

Ein anderes Problem, das bei Regressionen entstehen kann ist das sogenannte *Overfitting* [3], welches die Überanpassung der resultierenden Funktion an die *Trainingsdaten* bezeichnet. Genauer bedeutet dies, dass der Fehler in Bezug auf die *Trainingsdaten* zwar kleiner wird, der Fehler zu den *Testdaten* allerdings größer. Der *Hypothesenraum*, aus welchem die Modellfunktion stammt ist in diesem Fall zu groß. Dies ist beispielsweise der Fall, wenn die Anzahl der Parameter in der Modellfunktion gleich oder kleiner den gegebenen Trainingsdaten ist und die Regression eine Funktion bestimmt, die exakt durch alle diese Punkte verläuft. Die Schätzungen orientieren sich hierbei zu stark an den Beispieldaten, was in den meisten Fällen zu großen Ungenauigkeiten in Bezug auf die Voraussage der Funktionswerte an den noch unbekannt Stellen führt. Es müssen daher zusätzliche Verfahren verwendet werden durch welche sich feststellen lässt, ob weiteres Training die Aussagekraft der Regression verbessert oder verschlechtert, um das Problem des *Overfitting* zu vermeiden. Verfahren dazu sind beispielsweise die *Kreuz-Validierung* oder das *Pruning*[3].

In Bezug auf die von uns zu bestimmenden Potentialkurven hat sich eine Methode der Regression, die *Gauß-Prozess-Regression* schon in anderen Arbeiten [5] als sinnvoll und aussagekräftig erwiesen, da wir mit diesem Verfahren zusätzlich zu einer Voraussage für die Funktionswerte auch einen Schätzer für den entstehenden Fehler an beliebigen Stellen erhalten und diesen somit gegebenenfalls beeinflussen können. Außerdem ist die Rechenkomplexität nur abhängig von der Anzahl der gegebenen *Trainingsdaten* und nicht von der Dimension der verwendeten Variablen innerhalb der Modellfunktion, was in Bezug auf die Moleküldynamik den entscheidenden Vorteil bietet da wir so auch Moleküle betrachten können, die aus einer großen Anzahl an Teilchen bestehen.

Diesem Ansatz werden wir in dieser Arbeit folgen und untersuchen, wie sich der *Gauß-Prozess* effizient auf das von uns betrachtete Problem der Bestimmung von Energiekurven anwenden lässt. Betrachten wir nun das zugrundeliegende Verfahren im Detail.

### 3.2. Der Gauß-Prozess

Der *Gauß-Prozess* ist ein Verfahren des *überwachten maschinellen Lernens*, welches für Regressionen und Klassifikationen verwendet wird.

Der *Gauß-Prozess* beruht hauptsächlich auf den Ideen der *multivariaten Gauß-Verteilung* [14] und der *Bayes-Statistik* (A.1) und ermöglicht uns Schätzungen für unbekannte Werte einer Funktion vorherzusagen, deren Auswertung wir nur für eine endliche Anzahl von Stellen kennen. Wir können zudem die Genauigkeiten dieser Schätzungen bestimmen, was es uns zusätzlich möglich macht diese zu bewerten.

Der *Gauß-Prozess* [24] ist eine Verallgemeinerung der multivariaten Gauß-Verteilung, wie die Gauß-Verteilung selbst ist auch er eindeutig bestimmt durch den Erwartungswert und die Kovarianz. Der Unterschied besteht nun darin, dass diese beiden Komponenten der Gauß-Verteilung aus einem Vektor beziehungsweise einer Matrix bestehen, bei dem *Gauß-Prozess* handelt es sich bei beidem um Funktionen, was bedeutet, dass wir eine Verteilung über Funktionen betrachten.

**Definition 3.1** (Gauß-Prozess). *Sei  $X$  eine beliebige Indexmenge und  $g(x)$ ,  $x \in X$  ein stochastischer Prozess, das bedeutet eine Kollektion von Zufallsvariablen  $\{g(x)|x \in X\}$ . Dann wird  $g(x)$  als Gauß-Prozess bezeichnet, wenn für jedes  $n \in \mathbb{N}$  und  $x_1, \dots, x_n \in X$  die Zufallsvariablen  $g(x_1), \dots, g(x_n)$  normalverteilt sind.*

Wir werden uns in dieser Arbeit näher mit der *Gauß-Prozess-Regression* als eine Form der nicht-parametrischen Regression beschäftigen, da wir daran interessiert sind Schätzwerte für die potentielle Energie von Teilchen mit beliebigen Abständen zueinander bestimmen zu können und ihre Energiefunktion somit zu approximieren.

Nähere Informationen zur *Gauß-Prozess-Klassifikation* werden in [7] beschrieben.

Der Grundgedanke zur Lösung von Regressionsproblemen mit Hilfe des *Gauß-Prozesses* besteht darin, eine *a-priori Normalverteilung* als Modellannahme zu treffen und daraus die *a-posteriori Normalverteilung* für die *Testdaten* zu berechnen.

Nach der Definition des *Gauß-Prozesses* ist diese *a-priori Verteilung* eine Normalverteilung, was bedeutet, dass sie durch ihre Erwartungswert- und ihre Kovarianzfunktion festgelegt ist. Die Kovarianz ist eine positiv-semidefinite Matrix (A.3), kann ansonsten allerdings beliebig gewählt werden und wird bestimmt durch die Kovarianzfunktion, welche wiederum immer aus positiv-semidefiniten *Kernfunktionen* besteht.

Erwartungswert- und Kovarianzfunktion können außerdem *freie Parameter* enthalten, mit welchen sich die Eigenschaften der zu lernenden Funktion steuern lassen. Durch verschiedene Werte für diese Parameter können unterschiedliche Einflüsse oder Gegebenheiten des zu beschreibenden Vorgangs repräsentiert werden, daher müssen wir für jedes Problem einzeln entscheiden, wie die Parameter innerhalb der Funktionen zu wählen sind.

Die dabei resultierenden Funktionen können durch diese Wahl allerdings sowohl so simpel werden, dass die Ergebnisfunktion keine Aussagekraft in Bezug auf die gesuchte Funktion besitzt, als auch so komplex werden, dass die Durchführung von Berechnungen zu kompliziert wird. Ein weiteres Problem im Zusammenhang ist das, in Abschnitt 3.1.1 beschriebene *Overfitting*, welches auftritt, wenn die Parameter zu stark an die Trainingsdaten angepasst werden und somit die Aussagekraft des Ergebnisses verringern. Diesem Problem wird im *Gauß-Prozess* mit Hilfe von Verfahren wie der Maximierung der *Likelihood-Funktion*, wodurch die Parameter bestimmt werden die anhand der gegebenen Daten am wahrscheinlichsten angenommen werden, oder der *Kreuz-Validierung* entgegengewirkt.

Ziel ist es also, die Werte so zu wählen, dass eine Balance zwischen der Komplexität des Rechenaufwands und der Aussagekraft der Funktionen erreicht wird.

Die *Gauß-Prozess-Regression* beruht auf der Schätzung von Prior-Wahrscheinlichkeit der *Trainingsdaten*, wodurch die Posterior-Wahrscheinlichkeitsverteilung für unbekannte *Testdaten* berechnet werden kann. Anhand dieser Informationen lassen sich Erwartungswerte und Varianzen der *Testdaten* berechnen und somit eine Schätzung für die Funktionswerte an den betreffenden Stellen und für den resultierenden Fehler bestimmen.

Wie das Verfahren im Detail aussieht, wird in den Folgenden Abschnitten erklärt.

### 3.2.1. Die Gauß-Prozess-Regression

Die *Gauß-Prozess-Regression* wird verwendet um Voraussagen von Funktionswerten anhand von Trainingsdaten zu erzeugen. Durch die Berechnung von Erwartungswerten und Varianzen können Schätzungen für die zugrunde liegende Funktion berechnet und die möglichen Abweichungen beziehungsweise Fehlerraten bestimmt werden.

Sei  $f : \mathbb{R}^D \rightarrow \mathbb{R}$  die gesuchte Funktion und seien die Daten  $\mathbf{y} = \{y_i\}_{i=1}^n$  für die Stellen  $\mathbf{x} = \{x_i\}_{i=1}^n$  Menge

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1, \dots, n}$$

bezeichnet unsere *Trainingsdaten*. Wir gehen davon aus, dass diese mit einem Fehler  $\{\epsilon_i\}_{i=1}^n = \boldsymbol{\epsilon}$  behaftet sind, der additiv, unabhängig und gaußverteilt ist und die Daten  $\mathbf{y}$  deshalb über exakten Daten  $f(x_i)$  durch

$$y_i = f(x_i) + \epsilon_i \quad (3.3)$$

beschrieben werden können.

Das Ziel besteht darin auch eine Schätzung für die Voraussage unbekannter Testwerte  $\{f_i^*\}_{i=1}^{n^*} = \{f(x_i^*)\}_{i=1}^{n^*}$  der Funktion  $f$  an beliebigen Teststellen  $\mathbf{x}^* = \{x_i^*\}_{i=1}^{n^*}$  bestimmen zu können.

Um dieses Problem zu lösen verwenden wir eine Modellannahme, basierend auf den Formeln der *Bayes-Statistik*.

Nach der *Bayes-Regel* [15] gilt für die bedingte Wahrscheinlichkeit

$$\underbrace{P(f_i^* | \mathbf{y})}_{\text{Posterior}} = \frac{\overbrace{P(\mathbf{y} | f_i^*)}^{\text{Likelihood}} \overbrace{P(f_i^*)}^{\text{Prior}}}{P(\mathbf{y})} \propto P(\mathbf{y} | f_i^*) P(f_i^*). \quad (3.4)$$

Im Folgenden soll es darum gehen die Verteilung für  $P(f_i^* | \mathbf{y})$  zu bestimmen, um damit Voraussagen für die Werte von  $y_i^*$  machen zu können.

Wir werden hierbei näher auf das *lineare Modell* [17] für die Modellfunktion  $g$  wie in (3.2), eingehen.

### Das lineare Modell

Das *lineare Modell* stellt einen Spezialfall in der Voraussage mit Hilfe von *Gauß-Prozessen* dar. Seien

$$\phi_j : V_m \rightarrow \mathbb{R}^D, \quad j = 1, \dots, m,$$

Basisfunktionen und  $V_m = \text{span}\{\phi_j\}$  der *Hypothesenraum*, aus welchem die Modellfunktionen für  $f$  stammen. Innerhalb des linearen Modells nehmen wir an, dass wir die Funktion durch eine Linearkombination aus festen Basisfunktionen  $\phi_j$  und den zugehörigen Gewichtsfunktionen  $w_j \in \mathbb{R}, j = 1, \dots, m$  beschrieben werden kann. Diese können wir dann darstellen durch

$$\mathbf{f} = \sum_{j=1}^m \phi_j w_j = \boldsymbol{\phi}^\top \mathbf{w} \quad (3.5)$$

und der Wert der Funktion an der Stelle  $x_i$  der *Trainingsdaten* ist definiert durch

$$f_i := f(x_i) = \sum_{j=1}^m w_j \phi_j(x_i)$$

und  $\mathbf{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$ .

Für die weiteren Annahmen dieses Modells benötigen wir zunächst die Definition der Kovarianz und der Kovarianzmatrix, welche im Folgenden beschrieben werden.

**Definition 3.2** (Kovarianz). *Seien  $X$  und  $Y$  Zufallsvariablen und  $\mathbb{E}[X]$  der Erwartungswert (A.2) von  $X$ . Dann ist die Kovarianz von  $X$  und  $Y$  definiert durch*

$$\text{Cov}[X, Y] := \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

Hieraus ist zu erkennen, dass die Kovarianz zwischen einer Zufallsvariablen  $X$  zu sich selbst ihrer Varianz (A.2) entspricht, denn

$$\text{Cov}[X, X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{V}[X].$$

Betrachten wir diskrete Zufallsvektoren  $\mathbf{X} = (X_1, \dots, X_n)$ , so gilt die folgende Definition.

**Definition 3.3** (Kovarianzmatrix). *Sei  $\mathbf{X} = (X_1, \dots, X_n)$  ein diskreter Zufallsvektor. Dann ist die Kovarianzmatrix von  $\mathbf{X}$  definiert durch*

$$\text{Cov}[\mathbf{X}] := \begin{pmatrix} \text{Cov}[X_1, X_1] & \dots & \text{Cov}[X_1, X_n] \\ \vdots & \ddots & \vdots \\ \text{Cov}[X_n, X_1] & \dots & \text{Cov}[X_n, X_n] \end{pmatrix} = (\text{Cov}[X_i, X_j])_{i,j=1,\dots,n} \in \mathbb{R}^{n \times n},$$

mit den Varianzen  $\mathbb{V}[X_i] = \text{Cov}[X_i, X_i]$ ,  $i = 1, \dots, n$ , auf der Hauptdiagonalen.

Die Modellannahme besteht nun in einer *Prior-Verteilung* von  $\mathbf{w}$ , welche im *Gauß-Prozessen* eine Gauß- beziehungsweise Normalverteilung mit Erwartungswert Null und einer Kovarianzmatrix  $\Sigma$  darstellt, das heißt

$$\mathbf{w} \sim \mathcal{N}(0, \Sigma).$$

Zusätzlich rufen wir uns in Erinnerung, dass der Fehler  $\epsilon$  aus (3.3) additiv, unabhängig und gaußverteilt ist und nehmen darüber hinaus noch an, er habe Erwartungswert Null und die Varianz  $\sigma_n^2$ , also gilt

$$\epsilon_i \sim \mathcal{N}(0, \sigma_n^2).$$

Als Motivation sei hierbei genannt, dass wir Aussagen über die Funktion  $\mathbf{f}$  an unbekanntenen Stellen  $\mathbf{x}^*$  treffen wollen und aus diesem Grund zunächst die Verteilung für  $\mathbf{f}$  bestimmen müssen. Diese folgt jetzt aus der Verteilung von  $\mathbf{w}$  und wir benötigen zu deren eindeutigen Bestimmung lediglich den Erwartungswert und die Kovarianz, da es sich um eine Normalverteilung handelt.

Der Erwartungswert lässt sich bestimmen durch

$$\mathbb{E}[\mathbf{f}] = \mathbb{E}[\mathbf{w}^\top \boldsymbol{\phi}] = \mathbb{E}[\mathbf{w}^\top] \boldsymbol{\phi} = 0.$$

Sei  $\mathbf{Q}$  definiert als die Kovarianzmatrix von  $\mathbf{f}$ , dann sind die einzelnen Einträge definiert durch

$$\begin{aligned} \mathbf{Q}_{ij} &= \text{Cov}[f_i, f_j] = \mathbb{E} \left[ \left( \sum_{k=1}^n w_k \phi_k(x_i) \right) \left( \sum_{k=1}^n w_k \phi_k(x_j) \right) \right] - \underbrace{\mathbb{E} \left[ \sum_{k=1}^n w_k \phi_k(x_i) \right]}_{=0} \underbrace{\mathbb{E} \left[ \sum_{k=1}^n w_k \phi_k(x_j) \right]}_{=0} \\ &= \mathbb{E}[f_i f_j] \end{aligned}$$

und kann deshalb ausgedrückt werden durch

$$\mathbf{Q} = \mathbb{E}[\mathbf{f}\mathbf{f}^\top] = \boldsymbol{\phi}^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \boldsymbol{\phi} = \boldsymbol{\phi}^\top \boldsymbol{\Sigma} \boldsymbol{\phi} \quad (3.6)$$

Daraus folgt, dass  $\mathbf{f}$  gaußverteilt ist mit Erwartungswert Null und Kovarianzmatrix  $\mathbf{Q}$ , das heißt

$$\mathbf{f} \sim \mathcal{N}(0, \boldsymbol{\phi}^\top \boldsymbol{\Sigma} \boldsymbol{\phi}).$$

Wegen der Linearität des Erwartungswertes [14] und der Bilinearität der Kovarianz [14] folgt außerdem, dass

$$\begin{aligned} \text{Cov}[y_i, y_j] &= \text{Cov}[f(x_i) + \epsilon_i, f(x_j) + \epsilon_j] \\ &= \text{Cov}[f(x_i), f(x_j)] + \underbrace{\text{Cov}[f(x_i), \epsilon_j]}_{=0} + \underbrace{\text{Cov}[\epsilon_i, f(x_j)]}_{=0} + \text{Cov}[\epsilon_i, \epsilon_j] \\ &= \begin{cases} \text{Cov}[f(x_i), f(x_j)] + \sigma_n^2, & \text{falls } i = j, \\ \text{Cov}[f(x_i), f(x_j)], & \text{falls } i \neq j. \end{cases} \end{aligned} \quad (3.7)$$

Daraus ergibt sich die Verteilung für  $\mathbf{y}$  zu

$$\mathbf{y} \sim \mathcal{N}(0, \mathbf{Q} + \sigma_n^2 \mathbf{I}).$$

Sei die Kovarianzmatrix von  $\mathbf{y}$  definiert durch

$$\mathbf{C} = \mathbf{Q} + \sigma_n^2 \mathbf{I}, \quad (3.8)$$

dann folgt die Invertierbarkeit von  $\mathbf{C}$ , da  $\mathbf{Q}$  daraus, dass die Kovarianzmatrix symmetrisch positiv semidefinit ist und  $\sigma_n^2 \mathbf{I}$  symmetrisch positiv definit. Die Summe aus diesen beiden Matrizen ergibt folglich eine symmetrische positiv definite Matrix, was Invertierbarkeit impliziert.

Wir beschäftigen uns im Folgenden mit der Voraussage der Verteilung von  $f_i^*$ , das heißt für einen zusätzlichen, noch unbekanntem Punkt um später auf die allgemeine Verteilung einer beliebigen Anzahl an *Testdaten* zu schließen.

Dabei nutzen wir die Information der gegebenen *Trainingsdaten* aus und bestimmen die bedingte Verteilung von  $f_i^*$  gegeben den Trainingsdaten  $\mathbf{y}$ , dazu benötigen wir zunächst den folgenden Satz [14].

**Satz 3.1** (Formel von der totalen Wahrscheinlichkeit). *Sei  $I$  eine höchstens abzählbare Menge und  $(B_i)_{i \in I}$  paarweise disjunkte Mengen mit  $P\left[\bigcup_{i \in I} B_i\right] = 1$ . Dann gilt für jedes  $A \in \mathcal{A}$ , mit  $\mathcal{A}$  einer  $\sigma$ -Algebra,*

$$P[A] = \sum_{i \in I} P[A|B_i]P[B_i].$$

Hieraus folgt nun mit der *Bayes-Regel* (3.4), dass

$$P(f_i^* | \mathbf{y}) \propto P(\mathbf{y} | f_i^*) \quad (3.9)$$

mit  $[\mathbf{y} | f_i^*] = (y_1, \dots, y_n, f_i^*)$ .

Da dies wieder einer Normalverteilung beschreibt, deren Kovarianzmatrix wir mit  $\mathbf{C}_{n+1}$  bezeichnen, erhalten wir für die Dichtefunktion von  $[\mathbf{y}f_i^*]$

$$p([\mathbf{y}f_i^*]) = \exp\left(-\frac{1}{2}\left[\begin{pmatrix} \mathbf{y}^\top & f_i^* \end{pmatrix} \mathbf{C}_{n+1}^{-1} \begin{pmatrix} \mathbf{y} \\ f_i^* \end{pmatrix}\right]\right). \quad (3.10)$$

Es gilt also, dass  $[\mathbf{y}f_i^*] \sim \mathcal{N}(0, \mathbf{C}_{n+1})$  mit der Kovarianzmatrix

$$\mathbf{C}_{n+1} = \begin{pmatrix} \begin{pmatrix} \mathbf{C} \\ \mathbf{k}^\top \end{pmatrix} & \begin{pmatrix} \mathbf{k} \\ \kappa \end{pmatrix} \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}. \quad (3.11)$$

Die Matrix ist wegen der Symmetrie und der positiven Definitheit von  $\mathbf{C}$  wieder symmetrisch positiv definit, weshalb sie ebenfalls invertierbar ist.

Da die Matrix  $\mathbf{C}_{n+1}$  Blockgestalt besitzt, können wir die Berechnung der Inverse vereinfachen indem wir ausnutzen, dass ihre Inverse auch Blockgestalt haben muss. Sei

$$\mathbf{C}_{n+1}^{-1} = \begin{pmatrix} \mathbf{M} & \mathbf{m} \\ \mathbf{m}^\top & m \end{pmatrix}$$

diese Blockdarstellung, dann gilt

$$\mathbf{C}_{n+1} \mathbf{C}_{n+1}^{-1} = \begin{pmatrix} \mathbf{C}\mathbf{M} + \mathbf{k}\mathbf{m}^\top & \mathbf{C}\mathbf{m} + m\mathbf{k} \\ \mathbf{k}^\top\mathbf{M} + \kappa\mathbf{m}^\top & \mathbf{k}^\top\mathbf{m} + \kappa m \end{pmatrix} = \mathbf{I}_{n+1},$$

woraus wir die Teilmatrizen von  $\mathbf{C}_{n+1}^{-1}$  berechnen können.

Es gilt

$$\mathbf{C}\mathbf{M} + \mathbf{k}\mathbf{m}^\top = \mathbf{I}_n \quad (3.12)$$

$$\mathbf{C}\mathbf{m} + m\mathbf{k} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad (3.13)$$

$$\begin{aligned} \mathbf{k}^\top\mathbf{M} + \kappa\mathbf{m}^\top &= (0, \dots, 0) \\ \mathbf{k}^\top\mathbf{m} + \kappa m &= 1. \end{aligned} \quad (3.14)$$

Aus (3.13) folgt

$$\begin{aligned} \mathbf{C}\mathbf{m} &= -m\mathbf{k} \\ \Leftrightarrow \mathbf{m} &= -m\mathbf{C}^{-1}\mathbf{k}. \end{aligned} \quad (3.15)$$

Dies in (3.14) eingesetzt ergibt

$$\begin{aligned} \mathbf{k}^\top(-m\mathbf{C}^{-1}\mathbf{k}) + \kappa m &= 1 \\ \Leftrightarrow m(-\mathbf{k}^\top\mathbf{C}^{-1}\mathbf{k} + \kappa) &= 1 \\ \Leftrightarrow m &= (\kappa - \mathbf{k}^\top\mathbf{C}^{-1}\mathbf{k})^{-1} \end{aligned} \quad (3.16)$$

und aus (3.12) folgt

$$\begin{aligned}
 \mathbf{C}\mathbf{M} + \mathbf{k}\mathbf{m}^\top &= \mathbf{I}_n \\
 \Leftrightarrow \mathbf{M} &= \mathbf{C}^{-1}(\mathbf{I}_n - \mathbf{k}\mathbf{m}^\top) \\
 &\stackrel{(3.15)}{=} \mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{k}(-m\mathbf{C}^{-1}\mathbf{k})^\top \\
 &= \mathbf{C}^{-1} - \left(-\frac{1}{m}(m\mathbf{C}^{-1}\mathbf{k})(m\mathbf{C}^{-1}\mathbf{k})^\top\right) \\
 &= \mathbf{C}^{-1} + \frac{1}{m}\mathbf{m}\mathbf{m}^\top.
 \end{aligned} \tag{3.17}$$

Setzen wir dies in (3.10) ein, so erhalten wir

$$\begin{aligned}
 &\exp\left(-\frac{1}{2}\left[\begin{array}{cc|c} \mathbf{y}^\top, y_i^* & \begin{pmatrix} \mathbf{M} & \mathbf{m} \\ \mathbf{m}^\top & m \end{pmatrix} & \begin{pmatrix} \mathbf{y} \\ y_i^* \end{pmatrix} \end{array}\right]\right) \\
 &= \exp\left(-\frac{1}{2}\left[\mathbf{y}^\top\mathbf{M}\mathbf{y} + (y_i^*\mathbf{m}^\top\mathbf{y}) + (\mathbf{y}^\top\mathbf{m}y_i^*) + (y_i^*)^2m\right]\right) \\
 &= \exp\left(-\frac{1}{2}\left[(y_i^*)^2m - 2y_i^*(m\mathbf{k}^\top\mathbf{C}^{-1})\mathbf{y} + [\mathbf{y}^\top\mathbf{C}^{-1}\mathbf{y} + m(\mathbf{k}^\top\mathbf{C}^{-1}\mathbf{y})^\top(\mathbf{k}^\top\mathbf{C}^{-1}\mathbf{y})]\right]\right) \\
 &= \exp\left(-\frac{1}{2}\left[\mathbf{y}^\top\mathbf{C}^{-1}\mathbf{y} + m[y_i^* - (\mathbf{k}^\top\mathbf{C}^{-1}\mathbf{y})]^2\right]\right) \\
 &= \underbrace{\exp\left(-\frac{1}{2}\left[\mathbf{y}^\top\mathbf{C}^{-1}\mathbf{y}\right]\right)}_{\text{Konstante}} \exp\left(-\frac{1}{2}\left[m[y_i^* - (\mathbf{k}^\top\mathbf{C}^{-1}\mathbf{y})]^2\right]\right).
 \end{aligned}$$

Das bedeutet mit (3.9),  $f_i^*|\mathbf{y}$  auch gaußverteilt und es gilt

$$f_i^*|\mathbf{y} \sim \mathcal{N}(\mathbf{k}^\top\mathbf{C}^{-1}\mathbf{y}, \kappa - \mathbf{k}^\top\mathbf{C}^{-1}\mathbf{k}). \tag{3.18}$$

Für die Anwendung auf das lineare Modell verwenden wir das Resultat aus [7], welches die bedingte Verteilung von  $f_i^*$  an der Stelle  $x_i^*$ , gegeben den *Trainingsdaten*  $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}$  folgendermaßen beschreibt

$$f_i^*|x_i^*, \mathcal{D} \sim \mathcal{N}(\phi(x_i^*)^\top \Sigma \phi(\mathbf{x}) \mathbf{C}^{-1} \mathbf{y}, \phi(x_i^*)^\top \Sigma \phi(x_i^*) - \phi(x_i^*)^\top \Sigma \phi(\mathbf{x}) \mathbf{C}^{-1} \phi(\mathbf{x})^\top \Sigma \phi(x_i^*)).$$

Somit können wir über diese Verteilung die Erwartungswerte und Abweichungen für die neuen, unbekanntenen Werte  $f_i^*$  voraussagen.

Betrachten wir nun anstelle eines einzigen Punktes mehrere Stellen  $\mathbf{x}^*$ , an denen wir eine Voraussage machen wollen, so bestimmen wir die entsprechende Verteilung für  $\mathbf{f}^*|\mathbf{y}$ . Diese wird entsprechend (3.11) durch

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \mathbf{Q}(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I} & \mathbf{Q}(\mathbf{x}, \mathbf{x}^*) \\ \mathbf{Q}(\mathbf{x}^*, \mathbf{x}) & \mathbf{Q}(\mathbf{x}^*, \mathbf{x}^*) \end{pmatrix}\right) \tag{3.19}$$

beschrieben.

Dabei wird die Kovarianzmatrix für die Trainingsinputs  $\mathbf{x}$  durch

$$\mathbf{Q}(\mathbf{x}, \mathbf{x}) = \begin{pmatrix} Q(x_1, x_1) & \dots & Q(x_1, x_n) \\ \vdots & \ddots & \vdots \\ Q(x_n, x_1) & \dots & Q(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

bezeichnet und die Kovarianz zwischen den Trainingsinputs und einem Testinput durch

$$\mathbf{Q}(x_i^*, \mathbf{x}) = \begin{pmatrix} Q(x_i^*, x_1) \\ \vdots \\ Q(x_i^*, x_n) \end{pmatrix} \in \mathbb{R}^n.$$

Die Kovarianzmatrixmatrix zwischen den *Trainings-* und *Testdaten*  $\mathbf{Q}(\mathbf{x}, \mathbf{x}^*)$  sei dabei definiert durch

$$\mathbf{Q}(\mathbf{x}^*, \mathbf{x}) = \begin{pmatrix} Q(x_1^*, x_1) & \dots & Q(x_n^*, x_1) \\ \vdots & \ddots & \vdots \\ Q(x_1^*, x_n) & \dots & Q(x_n^*, x_n) \end{pmatrix} \in \mathbb{R}^{n^* \times n} \quad (3.20)$$

und entsprechend dazu gilt für die Kovarianzmatrix zwischen *Test-* und *Trainingsdaten*

$$\mathbf{Q}(\mathbf{x}, \mathbf{x}^*) = (\mathbf{Q}(\mathbf{x}^*, \mathbf{x}))^\top \in \mathbb{R}^{n \times n^*}.$$

Sei  $\mathbf{C}$  die, mit dem Messfehler  $\sigma_n^2$  behaftete, Kovarianzmatrix der *Trainingsdaten*, das heißt

$$\mathbf{C} = \mathbf{Q}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I},$$

dann ist der Erwartungswert und die Varianz von  $f_i^*$ ,  $i = 1, \dots, n$  somit nach (3.18) gegeben durch

$$\mathbb{E}[f_i^*] = \mathbf{Q}(x_i^*, \mathbf{x})^\top \mathbf{C}(\mathbf{x}, \mathbf{x})^{-1} \mathbf{y} \quad (3.21)$$

und

$$\mathbb{V}[f_i^*] = Q(x_i^*, x_i^*) - \mathbf{Q}(x_i^*, \mathbf{x})^\top \mathbf{C}(\mathbf{x}, \mathbf{x})^{-1} \mathbf{Q}(x_i^*, \mathbf{x}) \quad (3.22)$$

berechnet. Anhand der Vorschriften zur Berechnung des Erwartungswertes (3.21) und der Varianz (3.22) können wir feststellen, dass für die Voraussagen mit Hilfe der *Gauss-Prozess-Regression* lediglich eine konkrete Kovarianzfunktion bestimmt werden muss.

Für das lineare Modell bedeutet das in diesem Zusammenhang, dass wir nach (3.6) die Kovarianzfunktion, oder auch *Kern* [7] genannt, durch

$$k(x_i, x_j) := Q(x_i, x_j) = \boldsymbol{\phi}(x_i)^\top \boldsymbol{\Sigma} \boldsymbol{\phi}(x_j),$$

beziehungsweise die Kovarianzmatrix durch

$$\mathbf{Q} = (k(x_i, x_j))_{i,j=1,\dots,n}$$

beschreiben können. Der entscheidende Vorteil daran ist, dass die Voraussage nun nur noch von diesem inneren Produkt abhängt und damit nicht mehr dimensionsabhängig ist. Dies wird als der *Kerntrick* [7] bezeichnet.

Die Kovarianzfunktion zu wählen erweist sich allerdings häufig als problematisch, da berücksichtigt werden muss welchen Sachverhalt sie widerspiegeln soll. Die Kovarianzfunktion gibt

an, in welchem Zusammenhang die einzelnen Elemente der *Trainingsdaten* untereinander stehen und überträgt dies im Verlauf der Regression auch auf die *Testdaten*. Konkret müssen wir also diejenigen Informationen über den Zusammenhang der einzelnen Funktionswerte kennen, welche die Kovarianzfunktion letztendlich zum Ausdruck bringen soll.

Aus diesem Grund ist die Kovarianzfunktion der entscheidende Faktor für die Genauigkeit der Voraussagen mit der *Gauß-Prozess-Regression*.

Können wir nur grobe Aussagen über die zugrundeliegende Funktion treffen, sodass wir auf eine, von zusätzlichen Parametern abhängige, Kovarianzfunktion schließen können, so haben wir mit Hilfe der *Gauß-Prozess-Regression* die Möglichkeit diese sogenannten Hyperparameter an die *Trainingsdaten* anzupassen.

Es existiert eine Vielzahl von möglichen Funktionen, die verwendet werden können, da wir im Rahmen dieser Arbeit allerdings nur die zu unserer Problemstellung passende Kovarianzfunktion näher betrachten werden, sei deshalb in diesem Zusammenhang auf [7] verwiesen.

Im Folgenden soll ein Spezialfall der Kovarianzfunktion, dem sogenannten *Gauß-Kern* näher erläutert werden, welchen wir im weiteren Verlauf der Arbeit für die Approximation der Energiefunktionen benötigen werden.

#### 3.2.2. Der Gauß-Kern

Der *Gauß-Kern* stellt eine der möglichen Wahlen der Kovarianzfunktion innerhalb von *Gauß-Prozessen* dar, welcher über *Gaußsche Basisfunktionen* definiert ist.

Wir werden im weiteren Verlauf dieser Arbeit eine spezielle Kernfunktion, den *Gauß-Kern* betrachten welcher aus der Verwendung von *Gauß-Basen* resultiert.

Dieser ist abhängig von sogenannten *Hyperparametern*  $\theta = (\{M\}, \sigma_f)$ , wobei  $M$  eine Matrix darstellt und  $\{M\}$  ihre jeweiligen Elemente.

Die Kovarianzfunktion sei nun folgendermaßen definiert

$$G(x_i, x_j) := k(x_i, x_j) = \sigma_f^2 \cdot \exp\left(-\frac{1}{2}(x_i - x_j)^\top M(x_i - x_j)\right). \quad (3.23)$$

Wir erhalten somit nach (3.5) für die zugrunde liegende Funktion für eine beliebige Stelle  $x_i^*$  den folgenden Ausdruck

$$f_i^* = f(x_i^*) = \sum_{k=1}^n \alpha_k G(x_i^*, x_k), \quad (3.24)$$

wobei  $\{\alpha_k\}_{k=1}^n = \alpha$  die zugehörigen Interpolationskoeffizienten darstellt.

Das bedeutet wir können jeden beliebigen Punkt durch diese Linearkombination darstellen, in welcher die zugrunde liegenden *Trainingsdaten*  $\mathbf{x} = \{x_i\}_{i=1}^n$  einbezogen werden.

Die *Gauß-Prozess-Regression* kann nun analog zu den vorangegangenen Abschnitten mit einer konkreten Kovarianz- beziehungsweise Kernfunktion durchgeführt werden.

Der entscheidende Vorteil liegt in diesem Zusammenhang in der dimensionsunabhängigen Bestimmung der zugehörigen Interpolationskoeffizienten  $\alpha$ , worauf im Folgenden näher eingegangen werden soll.

Sei  $\mathbf{Q}$  die nach der Kernfunktion (3.23) berechnete und der *Gauß-Prozess-Regression* zugrunde liegende Kovarianzmatrix und  $G(\mathbf{x}, \mathbf{x})$  definiert durch

$$G(\mathbf{x}, \mathbf{x}) = \begin{pmatrix} G(x_1, x_1) & \dots & G(x_1, x_n) \\ \vdots & \ddots & \vdots \\ G(x_n, x_1) & \dots & G(x_n, x_n) \end{pmatrix},$$

dann gilt für die Kovarianzmatrix der fehlerbehafteten *Trainingsdaten*

$$\mathbf{C}(\mathbf{x}, \mathbf{x}) = G(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}.$$

Hierbei kommt also noch der Hyperparameter  $\sigma_n$  als ein Maß für den entstehenden Fehler hinzu.

Um die Interpolationskoeffizienten bestimmen zu können betrachten wir den Wert der Funktion an mehreren unbekanntem Daten  $\mathbf{x}^* = \{x_i^*\}_{i=1}^{n^*}$

$$\begin{aligned} \mathbf{f}^* = f(\mathbf{x}^*) &= \begin{pmatrix} \sum_{k=1}^n \alpha_k Q(x_1^*, \mathbf{x}_k) \\ \vdots \\ \sum_{k=1}^n \alpha_k Q(x_{n^*}^*, \mathbf{x}_k) \end{pmatrix} \\ &= \begin{pmatrix} Q(x_1^*, x_1) & \dots & Q(x_1^*, x_n) \\ \vdots & \ddots & \vdots \\ Q(x_{n^*}^*, x_1) & \dots & Q(x_{n^*}^*, x_n) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \\ &\stackrel{(3.20)}{=} \mathbf{Q}(\mathbf{x}^*, \mathbf{x}) \boldsymbol{\alpha}. \end{aligned}$$

Setzen wir nun

$$\boldsymbol{\alpha} = \mathbf{C}(\mathbf{x}, \mathbf{x})^{-1} \mathbf{y},$$

so erhalten wir

$$\mathbf{f}^* = \mathbf{Q}(\mathbf{x}^*, \mathbf{x})^\top \mathbf{C}(\mathbf{x}, \mathbf{x})^{-1} \mathbf{y}$$

und damit den Erwartungswert der *Gauß-Prozess-Regression* nach (3.21).

Um nun konkrete Voraussagen für die Funktionswerte an beliebigen Stellen machen zu können, ist es notwendig die Parameter der Kovarianzfunktion zu wählen. Da wir diese allerdings nicht immer über gegebene Informationen bestimmen können, ist es nötig Verfahren zu entwickeln, welche die optimalen Werte für die Parameter wählen.

Der Grundgedanke besteht hierbei darin, dass die *Hyperparameter* bestmöglich an die *Trainingsdaten* angepasst werden, um so eine repräsentative Schätzung für die Energiefunktion machen zu können. Das Verfahren, welches wir dazu verwenden werden, wird im Folgenden beschrieben.

### 3.2.3. Die Wahl der Parameter

Durch die Wahl der Kovarianzfunktion in 3.2.2 können wir zwar grobe Aussagen über die Relationen zwischen den Funktionswerten treffen, für Voraussagen ist es allerdings notwendig konkrete Werte für die *Hyperparameter* zu wählen. Die Idee hinter einer effizienten Wahl besteht darin, die Parameter den *Trainingsdaten* anzupassen, das bedeutet sie so zu bestimmen, dass die Wahrscheinlichkeit der *Trainingsdaten* gegeben der *Hyperparameter* mit den geschätzten Werten maximiert wird. Dieses Verfahren wird die *Maximum Likelihood Methode* genannt.

Sei  $\mathcal{D} = \{(x_i, y_i)\}_{i=1, \dots, n}$  die Menge der gegebenen *Trainingsdaten* und  $C(\boldsymbol{\theta})$  die zugehörige Kovarianzfunktion, abhängig von einem *Hyperparameter*, wie in Abschnitt 3.2.2 definiert.

Um diese nun anhand der *Trainingsdaten* zu optimieren werden wir der in [7] beschriebenen Methode für die *Gauß-Prozess-Regression* folgen.

Dazu müssen wir zunächst die Wahrscheinlichkeit für die *Trainingsdaten*  $\mathcal{D}$ , gegeben den Parametern  $\boldsymbol{\theta}$ , bestimmen. Da wir im *Gauß-Prozess* Gaußverteilungen verwenden wird diese Wahrscheinlichkeitsdichte entsprechend der *multivariaten Gaußverteilung* und wegen  $\mathbf{y} \sim \mathcal{N}(0, \mathbf{C})$  durch

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{C}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{y}^\top \mathbf{C}^{-1} \mathbf{y}\right)$$

beschrieben. Hierbei bezeichne  $|\mathbf{C}|$  die Determinante von  $\mathbf{C}$ .

Um die Werte der *Hyperparameter* zu bestimmen, die diese Wahrscheinlichkeitsdichte maximieren und somit die „wahrscheinlichsten“ Parameter zu bestimmen muss die Ableitung dieser Funktion nach  $\boldsymbol{\theta}$  bestimmt werden.

Dazu verwenden wir den natürlichen Logarithmus dieser Funktion, um die Berechnungen somit zu vereinfachen, da anstelle eines Produktes eine Summe entsteht. Die dabei entstehende Funktion wird als *Log Marginal Likelihood* bezeichnet und ist durch

$$\ln(p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})) = \underbrace{-\frac{1}{2} \ln|\mathbf{C}(\boldsymbol{\theta})|}_{\text{Complexity Penalty Term}} \quad \overbrace{-\frac{1}{2} \mathbf{y} \mathbf{C}^\top(\boldsymbol{\theta}) \mathbf{y}}^{\text{Data-Fit Term}} \quad \underbrace{-\frac{n}{2} \ln(2\pi)}_{\text{Normalization Term}}$$

gegeben, wobei der *Complexity Penalty Term* den Teil darstellt, der eine Art „Bestrafung“ für eine hohe Komplexität des Modells, also in diesem Fall für die Wahl der Parameter einfließen lässt und dabei außerdem ein Maß für diese darstellt. Das bedeutet, dass der Kompromiss zwischen ausreichender und zu hoher Komplexität hierdurch gewährleistet wird, so dass wir an dieser Stelle keinen zusätzlichen Term angeben müssen, der diese Gewichtung steuert, wie es beispielsweise bei der *Kreuz-Validierung* der Fall ist [7].

Der *Data-Fit Term* ist der einzige Term, der von den Funktionswerten der *Trainingsdaten*, also von  $\{y_i\}_{i=1, \dots, n}$ , abhängig ist, er gibt ein Maß dafür an, wie die *Hyperparameter* an die Daten angepasst sind.

Der *Normalization Term* stellt eine Normalisierungskonstante dar und ist für die Interpretation der *Log Marginal Likelihood* nicht weiter von Bedeutung.

Durch diese Umformung können wir auch den Gradienten der Funktion nach  $\boldsymbol{\theta}$  bestimmen,

dieser lautet

$$\frac{\partial \ln(p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}))}{\partial \theta_i} = \frac{1}{2} \text{tr}(\mathbf{C}^{-1}) \frac{\partial \mathbf{C}}{\partial \theta_i} - \frac{1}{2} \mathbf{y}^\top \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta_i} \mathbf{C}^{-1} \mathbf{y},$$

wobei  $\text{tr}(\mathbf{C}^{-1})$  die Spur der Inversen der Kovarianzmatrix  $\mathbf{C}$  bezeichne.

Ein Verfahren zur Bestimmung der optimalen *Hyperparameter* ist die *Maximum-Likelihood Methode*. Diese bestimmt das Maximum der *Likelihoodfunktion* und demnach den wahrscheinlichsten Wert der *Hyperparameter*  $\boldsymbol{\theta}$ .

Die Maximierung dieser Funktion kann nun mittels gradientenbasierter Optimierungsverfahren, wie zum Beispiel das *Verfahren des konjugierten Gradienten (CG)* oder die *Broyden-Fletcher-Goldfarb-Shanno (BFGS) Methode*, durchgeführt werden. Auf diese Verfahren soll im Rahmen dieser Arbeit nicht weiter eingegangen werden, für ausführliche Informationen sei dabei auf [18] verwiesen.

Die erwähnten Verfahren bestimmen jeweils das Minimum einer Funktion, weshalb in der Anwendung der Anpassung der *Hyperparameter* meist anstelle der *Log Marginal Likelihood* die *Negative Log Marginal Likelihood*

$$\mathcal{L}(\boldsymbol{\theta}) = -\ln(p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})).$$

und die entsprechenden Gradienten  $-\frac{\partial \mathcal{L}}{\partial \theta_i}$  verwendet werden.

Ein weiteres Verfahren zur Bestimmung der *Hyperparameter* wäre zum Beispiel die *Kreuz-Validierung*, welche sich jedoch bei einer hoher Anzahl an *Trainingsdaten* als ineffizient erweist [7] und Odeshalb hier nicht näher betrachtet werden soll.

Desweiteren besteht die Möglichkeit die Parameter über *Monte-Carlo-Verfahren* anzupassen, welche die Posterior Wahrscheinlichkeitsverteilung neuer Daten über die numerische Integration von

$$P(f_i^* | x_i^*, \mathcal{D}) = \int P(f_i^* | x_i^*, \boldsymbol{\theta}, \mathcal{D}) P(\boldsymbol{\theta} | \mathcal{D}) d\boldsymbol{\theta}$$

durch die *Markov-Chain-Monte-Carlo (MCMC) Methode* [9] berechnen. Wir erhalten somit die Voraussage über die Bestimmung dieses Integrals, jedoch keine konkreten Werte für die *Hyperparameter*. Zusätzlich muss die Inverse der Kovarianzmatrix  $\mathbf{C}$  zur Ermittlung des Erwartungswertes und der Varianz, wie in Kapitel 4.2.1, in jeder Iteration neu durchgeführt werden.

Auf diese Methode soll im Folgenden nicht näher eingegangen werden, für weitere Details der *MCMC Methode* in Bezug auf *Gauß-Prozesse* sei dabei auf [23] verwiesen.

Die bisher beschriebenen Verfahren beruhen alle darauf, dass die *Trainingsdaten* aus bekannten Funktionsauswertungen für eine bestimmte Anzahl von Eingabewerten bestehen.

Nehmen wir nun an, dass wir über weitere Informationen der Funktion an diesen Stellen verfügen, nämlich den Gradienten zu jedem Element der *Trainingsdaten*, dann können wir dies für die Voraussagen der resultierenden Funktion ausnutzen.

Wie dieses zusätzliche Wissen in die *Gauß-Prozess Regression* eingebunden wird und somit die Voraussagen der zugrundeliegenden Funktion verbessert, soll im Folgenden erläutert werden.

### 3.2.4. Die Gauß-Prozess-Regression mit Gradienten

In bestimmten Problemstellungen kann es sein, dass wir zusätzlich zu den Funktionswerten an bestimmten Stellen auch Informationen über die erste Ableitung an diesen oder anderen

Stellen besitzen. Wenn wir dieses zusätzliche Wissen in die *Gauß-Prozess-Regression* einbinden können, wäre es somit möglich Voraussagen über die Gradienten an beliebigen Stellen zu treffen beziehungsweise die Voraussagen für die Funktionswerte durch die Gradienteninformation zu verbessern.

Da die Differenzierung ein linearer Operator ist, beschreibt die Ableitung eines *Gauß-Prozesses* wieder einen *Gauß-Prozess*. Aus diesem Grund ist es uns tatsächlich möglich die Gradienteninformation in die Voraussagen der *Gauß-Prozess-Regression* einzubinden, indem wir die einzelnen Komponenten ableiten um Schätzungen für die Gradienten zu machen oder indem wir die beiden Prozesse miteinander verbinden.

Da wir mit Hilfe von *PSI3* zusätzlich zu den Energiewerten auch die Kraft, die auf das jeweilige Teilchen einwirkt ausgeben lassen können, werden wir uns näher mit dem Fall beschäftigen, dass die *Trainingsdaten* sowohl die Auswertung des Funktionswertes als auch den Gradienten an derselben Stelle beinhalten.

Gegeben sei der mit einem Fehler  $\kappa$  behaftete Wert des Gradienten

$$g_i := \frac{\partial f}{\partial x_i} + \kappa_i, \quad i = 1, \dots, n,$$

an jeder Stelle  $x_i$ ,  $i = 1, \dots, n$  der *Trainingsdaten*  $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\} = \{(x_i, y_i)\}_{i=1, \dots, n}$ . Dabei ist zu beachten, dass der Fehler  $\kappa_i$  der Gradienten einen unterschiedlichen Wert zu dem Fehler  $\epsilon$  der *Trainingsdaten* darstellt. Im Allgemeinen wird die Ableitung um eine Größenordnung schlechter genähert als der Funktionswert, weshalb der Fehler  $\kappa_i$  meist einen größeren Wert beschreibt.

Diese werden nun durch die Informationen über die Gradienten ergänzt, das bedeutet konkret, dass wir zusätzlich zu  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{x}^*$  und  $\mathbf{f}^*$ , wie in Kapitel 4.2.1, den Gradientenoutputvektor

$$\mathbf{g}^k = \{g_i^k\}_{i=1, \dots, n}, \quad k = 1, \dots, D,$$

betrachten, wobei  $k$  die Dimension bezeichne, nach der der Gradient bestimmt wird.

Um diese Information mit den *Trainingsdaten*  $\mathcal{D}$  zu verbinden, wird der Vektor  $\mathbf{y}$  durch die Gradienteninformation ergänzt. Das bedeutet wir betrachten

$$\mathbf{y}_{ext} = (y_1, \dots, y_n, g_1^1, \dots, g_n^1, \dots, g_1^D, \dots, g_n^D)$$

und wollen die Werte für  $\mathbf{f}^* = \{f_i^*\}_{i=1}^{n^*}$  bestimmen.

Sei  $Q$  die Kovarianzfunktion für  $\mathcal{D}$ , die Kovarianzmatrix für die ergänzten Daten entsteht über die Bildung der Kovarianzmatrix aus den Kovarianzfunktionen zwischen Gradient und Funktionswert und der Kovarianzfunktion zwischen Gradienten [19], das bedeutet wir betrachten

$$\frac{\partial Q(x_i, x_j)}{\partial x_i} =: Q_{yy^l}(x_i, x_j), \quad \frac{\partial Q(x_i, x_j)}{\partial x_j} =: Q_{y^k y}(x_i, x_j) \quad \text{und} \quad \frac{\partial^2 Q(x_i, x_j)}{\partial x_i \partial x_j} =: Q_{y^k y^l}(x_i, x_j),$$

wobei  $Q_{y^k y^l}(x_i, x_j)$  die partielle Ableitung der Kovarianzfunktion in der  $k$ -ten Dimension nach der ersten Variablen und in der  $l$ -ten Dimension nach der zweiten Variablen bezeichnet.

Wenn wir die Verteilung des Fehlers  $\kappa_i$ , analog des zu dem in Abschnitt 3.2.1 definierten Fehlers  $\epsilon_i$ , als additiv, unabhängig und gaußverteilt annehmen mit

$$\kappa_i \sim \mathcal{N}(0, \sigma_\kappa^2),$$

so können wir daraus die Kovarianzfunktion für die *Trainingsdaten*

$$C(x_i, x_j) = Q(x_i, x_j) + \sigma_\kappa \delta_{ij}$$

und daraus die Kovarianzmatrix  $\mathbf{C}^*(\mathbf{x}, \mathbf{x})$  der verbundenen *Gauß-Prozesse* für die *Trainingsdaten* bestimmen

$$\mathbf{C}^*(\mathbf{X}, \mathbf{X}) = \begin{pmatrix} \mathbf{C}(\mathbf{x}, \mathbf{x}) & \mathbf{C}_{yy^1}(\mathbf{x}, \mathbf{x}) & \dots & \mathbf{C}_{yy^D}(\mathbf{x}, \mathbf{x}) \\ \mathbf{C}_{y^1y}(\mathbf{x}, \mathbf{x}) & \mathbf{C}_{y^1y^1}(\mathbf{x}, \mathbf{x}) & \dots & \mathbf{C}_{y^1y^D}(\mathbf{x}, \mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{y^Dy}(\mathbf{x}, \mathbf{x}) & \mathbf{C}_{y^Dy^1}(\mathbf{x}, \mathbf{x}) & \dots & \mathbf{C}_{y^Dy^D}(\mathbf{x}, \mathbf{x}) \end{pmatrix}.$$

Diese kann mit den erweiterten Ausgabewerten der *Trainingsdaten* zur verbesserten Schätzung für  $\mathbf{f}^*$  verwendet werden.

Der zu  $\mathbf{C}^*$  gehörigen Erwartungswert und die Varianz ergeben sich aus einer Kombination aus den Ableitungen der der Formeln für die *Gauß-Prozess-Regression* ohne Gradienteninformation wie in (3.21) und (3.22) beschrieben und diesen Formeln selbst.

Betrachten wir dafür zunächst einmal den Fall, dass wir nur die Voraussage für die Gradientenwerte an beliebigen Stellen berechnen wollen. Wir definieren den Gradientenwert an der Stelle  $x_i^*$ ,  $i = 1, \dots, n$  als

$$\frac{\partial f}{\partial \mathbf{x}_i^*} =: \mathbf{g}_i^*.$$

Der Vektor mit den *Trainingsdaten* besteht aus diesem Grund aus  $\mathbf{x}$  und  $\mathbf{g} = (g_1^1, \dots, g_n^1, \dots, g_1^D, \dots, g_n^D)$ . Wir betrachten für die Berechnungen des Erwartungswertes und der Varianz nur den Teil der Kovarianzmatrix  $\mathbf{C}^*$ , der die Korrelationen zwischen den Gradienten beinhaltet, das bedeutet

$$\mathbf{C}^d(\mathbf{x}, \mathbf{x}) = \begin{pmatrix} \mathbf{C}_{y^1y^1}(\mathbf{x}, \mathbf{x}) & \dots & \mathbf{C}_{y^1y^D}(\mathbf{x}, \mathbf{x}) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{y^Dy^1}(\mathbf{x}, \mathbf{x}) & \dots & \mathbf{C}_{y^Dy^D}(\mathbf{x}, \mathbf{x}) \end{pmatrix}.$$

Der Erwartungswert des Gradienten von  $f_i^*$  ergibt sich nun zu

$$\mathbb{E}[g_i^*] = \mathbb{E}\left[\frac{\partial f}{\partial x_i^*}\right] = \frac{\partial \mathbb{E}[f_i^*]}{\partial x_i^*} \stackrel{(3.21)}{=} \frac{\partial}{\partial x_i^*} \left( Q(x_i^*, \mathbf{x})^\top \mathbf{C}^d(\mathbf{x}, \mathbf{x})^{-1} \mathbf{g} \right) = \frac{\partial \mathbf{Q}(x_i^*, \mathbf{x})}{\partial x_i^*} \mathbf{C}^d(\mathbf{x}, \mathbf{x})^{-1} \mathbf{g}$$

und die Varianz des Gradienten von  $y_i^*$  zu

$$\begin{aligned} \mathbb{V}[g_i^*] &= \mathbb{V}\left[\frac{\partial f}{\partial x_i^*}\right] = \frac{\partial^2 \mathbb{V}[f_i^*]}{\partial x_i^* \partial x_i^*} \stackrel{(3.22)}{=} \frac{\partial^2}{\partial x_i^* \partial x_i^*} \left( Q(x_i^*, x_i^*) - \mathbf{Q}(x_i^*, \mathbf{x}) \mathbf{C}^d(\mathbf{x}, \mathbf{x})^{-1} \mathbf{Q}(\mathbf{x}, x_i^*) \right) \\ &= \frac{\partial^2 Q(x_i^*, x_i^*)}{\partial x_i^* \partial x_i^*} - \frac{\partial \mathbf{Q}(x_i^*, \mathbf{x})}{\partial x_i^*} \mathbf{C}^d(\mathbf{x}, \mathbf{x})^{-1} \frac{\partial \mathbf{Q}(\mathbf{x}, x_i^*)}{\partial x_i^*}. \end{aligned}$$

Um nun Voraussagen für die Funktionswerte zu machen und die Gradienten mit einfließen zu lassen, müssen der Erwartungswert und die Varianz dementsprechend angepasst werden. Wir erhalten somit den Erwartungswert für  $f_i^*$ ,  $i = 1, \dots, n^*$  unter Einbeziehung der Gradienten  $\mathbf{g}_i$ ,

$i = 1, \dots, n$  indem wir den Vektor  $\mathbf{y}_{ext}$  Vektor bilden, der die Kovarianz zwischen den *Trainingsdaten* und einem Testpunkt, sowohl für die Funktionswerte, als auch für die Gradientenwerte enthält. Dieser wird dann analog zu (3.21) mit der Kovarianzmatrix  $\mathbf{C}^*$  und den bekannten Funktions- und Gradientenwerten multipliziert. Formal bedeutet dies

$$\mathbb{E}[f_i^*] = \begin{pmatrix} \mathbf{Q}(\mathbf{x}, x_i^*) \\ \frac{\partial \mathbf{Q}(\mathbf{x}, x_i^*)}{\partial x_i^*} \end{pmatrix}^\top \mathbf{C}^*(\mathbf{x}, \mathbf{x})^{-1} \mathbf{y}_{ext}.$$

Die Varianz von  $y_i$ ,  $i = 1, \dots, n$  unter Einbeziehung der Gradienten  $g_i$ ,  $i = 1, \dots, n$  wird analog zu (3.22) angepasst, wobei zu erwähnen ist dass die Kovarianz zwischen zwei Testeingabewerten keine Gradienteninformation enthält, weil wir uns auf die Voraussage der Funktionswerte unter Einbindung der Gradienteninformation beschränken. Daraus ergibt sich, dass

$$\mathbb{V}[f_i^*] = Q(x_i^*, x_i^*) - \begin{pmatrix} \mathbf{Q}(\mathbf{x}, x_i^*) \\ \frac{\partial \mathbf{Q}(\mathbf{x}, x_i^*)}{\partial x_i^*} \end{pmatrix}^\top \mathbf{C}^*(\mathbf{x}, \mathbf{x})^{-1} \begin{pmatrix} \mathbf{Q}(\mathbf{x}, x_i^*) \\ \frac{\partial \mathbf{Q}(\mathbf{x}, x_i^*)}{\partial x_i^*} \end{pmatrix}.$$

Zusammenfassend wird somit die Voraussage der zugrunde liegenden Funktion über die *Gauß-Prozess-Regression* durch die zusätzliche Information der Gradienten verbessert, da wir mehr Wissen über die zugrunde liegende Funktion einbinden können und deshalb genauere Ergebnisse erzielen.

Im Folgenden werden wir untersuchen, wie wir die *Gauß-Prozess-Regression* auf die Bestimmung von Potentialkurven anwenden können. Dazu ist es nötig zunächst eine passende Kovarianzfunktion zu wählen, unter Berücksichtigung dessen, dass wir bestimmte Eigenschaften der Energiefunktion bereits kennen.

### 3.3. Die Anwendung auf Potentialkurven

Um Energiefunktionen zu approximieren werden wir in dieser Arbeit den Ansatz über Gauß-Basen wie in Abschnitt 3.2.2 verwenden, welche sich als angemessene Wahl herausgestellt haben [4], da sie die Energiefunktion sinnvoll approximieren. Sie ermöglichen außerdem die analytische Berechnung von Integralen und gewährleisten somit die Differenzierbarkeit und damit auch die Glattheit der resultierenden Funktionen.

Da wir bei Energiefunktionen davon ausgehen können, dass sie diese Eigenschaften auch aufweisen [21], werden wir nachfolgend diesen Ansatz vertiefen.

Die Energiefunktion an einer beliebigen Stelle  $x_i^*$  kann aus diesem Grund nach (3.24) durch

$$E(x_i^*) \approx \sum_{k=1}^n \alpha_k G(x_i^*, x_k),$$

mit

$$G(x_i, x_j) = \sigma_f^2 \cdot \exp\left(-\frac{1}{2}(x_i - x_j)^\top M(x_i - x_j)\right)$$

approximiert werden mit den zugehörigen Interpolationskoeffizienten

$$\boldsymbol{\alpha} = \mathbf{C}(\mathbf{x}, \mathbf{x})^{-1} \mathbf{y}.$$

Wir haben also über die *Gauß-Prozess-Regression* eine Methode gefunden Potentialkurven anhand quantenmechanischer *Trainingsdaten* so zu nähern, dass wir sie zu Berechnungen in der Moleküldynamik verwenden können.

Konkret bestehen die *Trainingsdaten* aus den Abständen der jeweiligen Teilchen innerhalb des Moleküls. Betrachten wir eines, welches aus nur zwei Atomen besteht, so erhalten wir ein eindimensionales Problem mit einer Datenmenge  $\mathcal{D} = \{r, E(r)\}$ , wobei  $r$  den jeweiligen Abstand zwischen den Atomen im Raum bezeichnet und  $E(r)$  die daraus resultierende Energie. In der Anwendung auf die *Gauß-Prozess-Regression mit Gradienten* betrachten wir die Menge  $\mathcal{D} = \{r, E(r), G(r)\}$ , wobei  $G(r)$  die Menge der jeweiligen Gradienteninformation, wie in Abschnitt 3.2.4 beschrieben, beschreibt. Der *Hyperparameter*  $M$  ist in diesem Fall eine Zahl und wird für die Energiefunktion auf  $l^{-2}$  gesetzt. Dieser steuert die Korrelation zwischen den einzelnen Abständen der eingehenden Daten und wird deshalb auch als *Längenskalenparameter* bezeichnet.

Für Berechnungen in höheren Dimensionen verläuft die *Gauß-Prozess-Regression* analog unter Berücksichtigung der erhöhten Dimension der Eingabedaten, was bedeutet wir betrachten die jeweiligen Abstände aller Atome zueinander und das Problem erweitert sich auf die dabei entstehende Dimension  $D$ . Die Eingabedaten werden Dabei als eine Matrix der Form

$$\mathbf{r} = \begin{pmatrix} r_{11} & \cdots & r_{1D} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nD} \end{pmatrix}$$

beschrieben.

Der zugehörige *Gauß-Kern* ist in diesem Fall

$$G(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \cdot \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)\right).$$

Der *Hyperparameter*  $M$  ist in diesem Fall eine Diagonalmatrix  $M = \text{diag}(l)^{-2}$  und erhält somit die gleiche Bedeutung wie der *Längenskalenparameter* in einer Dimension.

Die Frage, mit der wir uns als nächstes befassen werden ist die, wie wir unsere *Trainingsdaten* für die Approximation der Potentialkurven am besten wählen. Da die Berechnungen der Energien sehr teuer werden können, müssen Kriterien entwickelt werden anhand derer man mit einer geringen Menge an Punkten startet und danach geschickt Stellen hinzunimmt, an denen die Funktionsauswertungen zusätzlich berechnet werden sollen. Eine Methode zur Durchführung dieses Verfahrens ist das *Learn on the Fly*.

### 3.4. Das Learn on the Fly Verfahren

Mit Hilfe der *Gauss-Prozess-Regression* haben wir eine Methode gefunden, anhand von *Trainingsdaten* eine Funktion zu lernen, jedoch müssen wir uns die Frage Stellen, wie wir diese Beispieldaten geschickt wählen.

---

**Algorithmus 2:** Learn on the Fly Algorithmus

---

**Input** : Fehlerschranke  $\epsilon_V$ , Anfangstrainingsdaten  $\{(x_i, y_i)\}_{i=1}^n$   
**Output** : Menge an Trainingsdaten  $\{(x_i, y_i)\}_{i=1}^{n+n_V}$   
 Setze  $\{(x_i, y_i)\}_{i=1}^{n+n_V} = \{(x_i, y_i)\}_{i=1}^n$   
 (1) *Starte die Gauß-Prozess-Regression mit den Trainingsdaten  $\{(x_i, y_i)\}_{i=1}^{n+n_V}$  und den Teststellen  $\{x_i^*\}_{i=1}^{n^*}$*   
**for**  $j = 1, \dots, n^*$  **do**  
     **if**  $V(x_j^*) > \epsilon_V$  **then**  
         1. *Bestimme den größten Wert  $V(x_k^*)$  für das Intervall  $I = [x_a, x_b]$  in dem sich  $x_j^*$  befindet, also mit  $x_j \in I$ .*  
         2. *Berechne die Energie an der Stelle  $x_k^*$  mit PSI3*  
         3. *Füge den neuen Punkt zu den Trainingsdaten hinzu*  
                  $\{(x_i, y_i)\}_{i=1}^{n+n_V} = \{(x_i, y_i), (x_k^*, y_k^*)\}_{i=1}^{n+n_V}$ .  
         4. *Gehe zu (1).*  
     **end**  
     **else**  
         | *Gehe zu (1).*  
     **end**  
**end**  
**return**  $\{(x_i, y_i)\}_{i=1}^{n+n_V}$

---

Da die Berechnungen der Energiewerte in Abhängigkeit des Abstandes und damit die verwendeten Punkte, unter Umständen sehr teuer werden können, ist es ratsam eine Methode zu bestimmen, die nur eine möglichst kleine Anzahl an *Trainingsdaten* auswählt.

Diese Daten sollten allerdings trotzdem aussagekräftig genug sein, um eine gute Approximation an die Energiefunktion zu gewährleisten.

Mit der Methode des *Learn on the Fly* wird die Möglichkeit beschrieben, etwas während einer laufenden Aktion zu lernen und so an den gegenwärtigen Prozess anzupassen.

Dieses Prinzip werden wir zur effizienten Auswahl der *Trainingsdaten* für die *Gauß-Prozess-Regression* benutzen. Das bedeutet in diesem Zusammenhang, einen Algorithmus zu entwickeln, der ausgehend von einer gewählten Startmenge an *Trainingspunkten*, nur an sinnvollen Stellen weitere Punkte hinzufügt, um ein genaueres Ergebnis zu erhalten.

Um dies zu realisieren ist es notwendig, ein Kriterium zu entwickeln, welches uns angibt, an welchen Stellen unser Ergebnis noch zu ungenau ist und deshalb verbessert werden muss.

Da die *Gauß-Prozess-Regression* mit der zu berechnenden Varianz über einen Schätzer für den resultierenden Fehler verfügt, muss erkannt werden, an welcher Stelle die Abweichung, das bedeutet die berechnete Varianz nach (3.22), zu groß ist um dort dann einen weiteren Punkt zu berechnen und zu den *Trainingsdaten* hinzuzufügen.

Wir formulieren deshalb eine Fehlerschranke

$$\mathbb{V}[f_i^*] \stackrel{!}{<} C_V, \quad i = 1, \dots, n^* \tag{3.25}$$

wobei  $x_i^*$  die Stelle des jeweils betrachteten *Testpunktes* beschreibt,  $f_i^* = f(x_i^*)$  die Funktionsauswertung an dieser Stelle und  $C_V$  eine Konstante beschreibt, welche den maximal tolerierten Wert der Varianz angibt.

Das Verfahren des *Learn on the Fly* startet demnach mit einer gegebenen Menge an *Anfangs-Trainingspunkten* und führt die *Gauß-Prozess-Regression* für diese wie in Abschnitt 3.2.1 aus.

Wir erhalten die jeweiligen Erwartungswerte nach (3.21) und Varianzen nach (3.22) zu jedem *Testpunkt* und können die Werte der Varianz mit der Fehlerschranke vergleichen. Jeder *Testpunkt* wird also durchlaufen und die Bedingung (3.25) wird somit an jeder Stelle überprüft.

Ist diese Bedingung an jedem *Testpunkt* erfüllt so ändert sich nichts.

Sobald diese allerdings nicht erfüllt ist, wird in dem aktuell betrachteten Intervall zwischen zwei *Trainingspunkten* ein zusätzlicher Punkt erst mit Hilfe von *PSI3* berechnet und dann in die *Trainingsdaten* eingefügt. Erfahrungsgemäß ist es in Bezug auf die Genauigkeit des *Gauß-Prozesses* am effektivsten an der Stelle einen Punkte einzufügen, an der die Varianz im jeweiligen Intervall den höchsten Wert besitzt.

Das Verfahren startet anschließend mit dem zusätzlich berechneten Punkt von neuem und wird solange wiederholt, bis die Bedingung an den Fehler für alle *Testdaten* erfüllt ist, siehe Algorithmus 2.

### 3.4.1. Die Anwendung auf die Moleküldynamik

In Bezug auf die Moleküldynamik wollen wir *Learn on the Fly* in Verbindung mit dem *Verlet-Verfahren* aus Kapitel 2.2 verwenden. Aus gegebenen Anfangspositionen und Anfangsgeschwindigkeiten eines Moleküls sollen dessen neue Positionen, Geschwindigkeiten, Kräfte und abgeleitete Größen wie die kinetische und potentielle Energie bestimmt werden.

Die Energiefunktion stellt das Potential in Abhängigkeit des Abstandes der Teilchen im Raum dar und ihre partielle Ableitung nach den einzelnen Positionen die einwirkenden Kräfte.

Mit Hilfe der *Gauß-Prozess-Regression* für Gradienten wie in Abschnitt 3.2.4 können wir den Erwartungswert der Ableitung der Gradienten der Energiefunktion bestimmen und erhalten somit eine Approximation für die resultierende Kraft.

Da wir die Ableitung direkt über den *Gauß-Prozess* erhalten, anstatt sie durch Differenzierung einer genäherten Funktion zu berechnen, wird der Approximationsfehler in diesem Fall nicht zusätzlich vergrößert.

Die Kombination des *Learn on the Fly* Verfahrens und dem *Verlet-Verfahren* beruht darauf, die aus Algorithmus 2 resultierende Funktion als Eingabewert zu verwenden und zusätzliche Verbesserungen nur an erforderlichen Stellen durchzuführen.

Durch die *Gauss-Prozess-Regression* für Gradienten erhalten wir aus den *Trainingsdaten* für die Funktionswerte und die Kräfte eine Approximation der Kraft zu jedem beliebigen Abstand, der zur Berechnung der Positionen und Energien zu jedem Zeitschritt benötigt wird.

Konkret verläuft das *Verlet-Verfahren* dann folgendermaßen. Zu jedem Zeitschritt wird im Algorithmus 2.2 an den Stellen, an denen die Berechnung der Kraft erfolgt eine zusätzliche Fehlerberechnung eingefügt, indem die Varianz der mit den aktuellen *Trainingsdaten* bestimmten Energiefunktion mit einem gewählten Grenzwert  $C_{ver}$  für den akzeptierten Fehler verglichen

wird. Überschreitet dieser Wert die Grenze an einem *Testpunkt*  $x_i^*$ ,  $i = 1, \dots, n^*$ , das bedeutet es gilt für die Varianz des geschätzten Funktionswertes  $f(x_i^*) = f_i^*$  an dieser Stelle

$$\mathbb{V}[f_i^*] > C_{ver},$$

so wird analog zu Algorithmus 2 ein zusätzlicher Punkt inklusive resultierender Kraft an dieser Stelle mit Hilfe von *PSI3* bestimmt. Daraufhin wird die *Gauß-Prozess-Regression* mit den um diesen neuen Datensatz erweiterten *Trainingsdaten* erneut durchgeführt und die Fehlerschranke von neuem überprüft. Ist der Wert der Varianz kleiner als  $C_{ver}$ , so ist das Ergebnis genau genug, um zur Berechnung von Molekülbewegungen verwendet zu werden und die Kraft wird aus dem Erwartungswert des Gradienten an der jeweiligen Stelle bestimmt.

Hierbei ist zu berücksichtigen, dass für die Anwendung des *Verlet-Verfahrens* die selben *Trainingsdaten*, sowohl für die *Gauß-Prozess-Regression* zur Bestimmung der Energiewerte als auch für die zur Bestimmung der Werte für die Kraft verwendet werden, da die Energiefunktion und ihre Ableitung konsistent zueinander sein müssen.

Durch die Verwendung dieser Methode gelingt es uns mit möglichst geringem Rechenaufwand ein sehr genaues Ergebnis für die Simulation von Molekülen zu ermöglichen. Dadurch können in vielen Anwendungen der Quantenchemie und Quantenphysik enorme Kosten gespart und Versuche in wesentlich geringerer Zeit durchgeführt werden.

## 4. Numerische Ergebnisse

Das folgende Kapitel zeigt die numerischen Ergebnisse der Anwendung von *Gauß-Prozess-Regressionen* für die Bestimmung von Potentialkurven und ihre Verwendung in der Moleküldynamik auf.

Die Bestimmung von Potentialkurven beruht auf einer Approximation der Lösung der *Schrödingergleichung*. Durch die Berechnung der potentiellen Energie in Abhängigkeit des jeweiligen Abstands der Atomkerne ist es möglich eine Energie-Hyperfläche zu entwickeln, anhand welcher wir den Zustand des jeweiligen Systems beschreiben können.

Da die in Kapitel 2 beschriebenen Verfahren zur Berechnung der Energien sehr teuer werden können, müssen wir eine Methode verwenden um diese mit weniger Rechenaufwand zu nähern und verwenden dabei den Ansatz die Energiefunktion anhand quantenmechanischer Daten zu fitten.

Für die Anwendung auf die Moleküldynamik ist die anschließende Bestimmung der Kraft, also der Ableitung des Potentials nach der jeweiligen Position der im System vorhandenen Teilchen, von entscheidender Bedeutung. Aus diesem Grund werden wir im Folgenden eine Methode entwickeln, die effizient sowohl die Funktion Energiewerte als auch die Funktion der Kräfte approximiert. Hierzu werden wir die in Kapitel 3 beschriebene *Gauß-Prozess-Regression* verwenden. Die Grundlage dafür sind die *Trainingsdaten*, welche so gewählt werden müssen, dass die Funktion möglichst genau mit nur wenigen Punkten approximiert werden kann.

Über bekannte Resultate aus der Quantenchemie können wir im voraus schon einige Aussagen über die Eigenschaften von Potentialkurven treffen, welche wir für die Ermittlung einer Näherung ausnutzen können. Dabei sind vor allem die Eigenschaft der Glattheit und die Kenntnis der Bindungslängen von sehr vielen Molekülen besonders aussagekräftig.

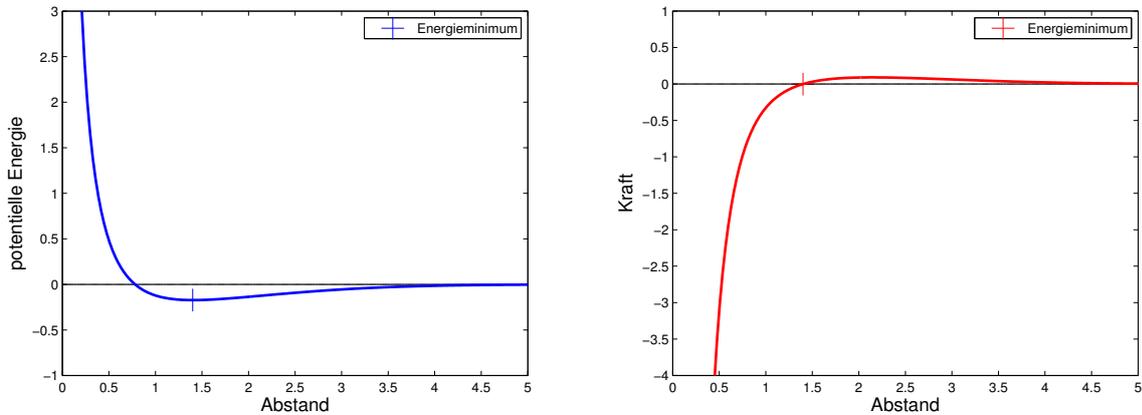
Um anschließend die Güte dieses Verfahrens zu messen benötigen wir ein Modellbeispiel, dessen Ergebnis wir kennen beziehungsweise berechnen können, um es anschließend mit dem Resultat der *Gauß-Prozess-Regression* zu vergleichen. Dazu betrachten wir im Folgenden das Wasserstoffmolekül.

### 4.1. Das Wasserstoffmolekül als Modellbeispiel

Das Wasserstoffmolekül besteht aus zwei Wasserstoffatomen und stellt das klassische und einfachste Beispiel innerhalb der Quantenchemie dar, es soll in diesem Zusammenhang als Modellbeispiel dienen. Wir werden die Effizienz der *Gauß-Prozess-Regression* zur Bestimmung von Approximationen der Potentialkurve und das Lernen von zusätzlichen Daten über das *Learn on the Fly* Verfahren aufzeigen.

Die Potentialkurve ergibt sich nun aus den Energien zu jedem Abstand, welche wir zunächst über eines der in Kapitel 2 beschriebenen Verfahren zur Lösung der *elektronischen Schrödingergleichung* konkret berechnen wollen. Wir betrachten außerdem die resultierende Ableitung, die den zugehörigen Kräften entspricht.

Die Berechnungen der Energien und Kräfte erfolgen mit Hilfe des in Abschnitt 2.3 beschriebenen Programms *PSI3*. Da es sich bei dem Wasserstoffmolekül um eine Bindung von nur zwei



(a) Die Energiefunktion.

(b) Die Kraftfunktion.

**Abbildung 4.1.:** Die Energie- und Kraftfunktion eines Wasserstoffmoleküls.

Atomen handelt und die Laufzeit hierbei noch kein wesentliches Problem darstellt, verwenden wir das *Coupled-Cluster-Verfahren*  $CCSD(T)$  als genaueste berechenbare Methode mit der *Restricted-Hartree-Fock* ( $RHF$ ) Determinante. Dabei lassen wir uns sowohl die Energiewerte, als auch die zugehörigen Kräfte in feiner Auflösung ausgeben um somit einen Vergleichswert zu der von uns bestimmten Näherung zu erhalten.

Die Abbildung 4.1 zeigt zum einen die resultierende Energiefunktion eines Wasserstoffmoleküls, welche die berechneten *Coupled-Cluster* Energiewerte in Hartrees für die Abstände zwischen 0.01 und 5 angibt und zum anderen die zugehörige Funktion der Kraft, also die Ableitung der Energiefunktion.

Anhand dieser Abbildungen lässt sich ein typischer Verlauf der potentiellen Energie und der zugehörigen Kraft erkennen, welcher sich folgendermaßen beschreiben lässt.

Ausgehend von einem großen Abstand besteht zunächst nur eine sehr geringe beziehungsweise gar keine potentielle Energie. Rücken die beiden Wasserstoffatome jedoch immer näher aneinander, so wirken sie aufeinander ein und es entsteht eine leichte Anziehungskraft, welche sich durch einen positiven Wert für die Kraft äußert.

Diese Anziehungskraft erhält einen immer höheren Energiewert bis zu der Stelle, an der sich das Energieminimum befindet, beziehungsweise im sogenannten *Grundzustand*. An dieser Stelle ist der Wert der Kraft Null und die Atome befinden sich im präferierten Abstand zueinander, welcher auch die Bindungslänge des Moleküls darstellt. Aus der Quantenchemie ist für das Wasserstoffmolekül über Berechnungen durch das  $CCSD(T)$ -Verfahren eine Bindungslänge von 1.417 Bohr [1] bekannt, was bedeutet, dass sich an dieser Stelle auch das Energieminimum befindet und mit dem Ergebnis in Abbildung 4.1 übereinstimmt.

Ist der Abstand geringer als dieses Minimum, so beginnen die Atome sich mit immer größerer Kraft abzustößen, was sich aus dem starken Abfall der Kraft in negative Richtung für kleiner werdende Abstände ergibt. Dabei ist zu erkennen, dass die Kraft mit der sich zwei immer näher liegende Atome abstoßen unendlich groß wird. Den Grenzfall für sehr geringe Abstände werden wir im weiteren Verlauf allerdings nicht näher betrachten, da sich Atome in der Realität nie unendlich nah kommen. Außerdem gehen wir von der Kenntnis langreichweitiger Potentiale aus, weshalb wir im Folgenden die Funktionen nur innerhalb eines Intervalls von Abständen zwischen

0.5 und 5 untersucht werden.

Nachfolgend werden wir aufzeigen, wie wir mit Hilfe der *Gauß-Prozess-Regression* diese Energiekurve und ihre Ableitung effizient approximieren können.

## 4.2. Die Anwendung der Gauß-Prozess-Regression

Die Grundlage für die *Gauß-Prozess-Regression* sind die *Trainingsdaten*, aus welchen die Energiefunktion und ihre Ableitung gelernt werden soll. Wir erhalten sie aus Energiewerten und den zugehörigen Kräften zum jeweiligen Abstand der Moleküle im Raum über quantenmechanische Berechnungen für bestimmte Kernkonfigurationen.

Zunächst beschäftigen wir uns mit der *Gauß-Prozess-Regression* für die Funktionswerte, also die Energien zu den jeweiligen Abständen.

Intuitiv wäre ein erster Ansatz für die Wahl der *Trainingsdaten* eine Menge an äquidistanten Punkten innerhalb des betrachteten Intervalls an Abständen. Dies erweist sich in der Anwendung allerdings als ineffizient, da Punkte an Stellen berechnet werden, die nicht zu einer Verbesserung der Approximation beitragen. Aus diesem Grund verwenden wir die allgemeine Kenntnis der Bindungslänge für viele Moleküle und insbesondere auch des Wasserstoffatoms, diese stellt das Minimum der Energiekurve und damit einen charakteristischen Punkt dar, welchen wir für die *Trainingsdaten* der *Gauß-Prozess-Regression* verwenden sollten.

Wir beginnen deshalb mit einer Datenmenge  $\mathcal{D}$ , bestehend aus dem Minimum und einem Punkt in jede Richtung in einem Abstand von 0.5 als Eingabewert für die *Trainingsdaten* in der *Gauß-Prozess-Regression*, wobei

$$\mathcal{D} = \left\{ (0.9, E(0.9)); (1.4, E(1.4)); (1.9, E(1.9)) \right\},$$

mit  $E(x)$  als der über *PSI3* berechnete Wert der Energie zum Abstand  $x$ .

Die Energiefunktion, wie in Abschnitt 3.3 definiert, impliziert für unser Beispiel die Kovarianzfunktion

$$Q(x_i, x_j) = \sigma_f^2 \cdot \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right)$$

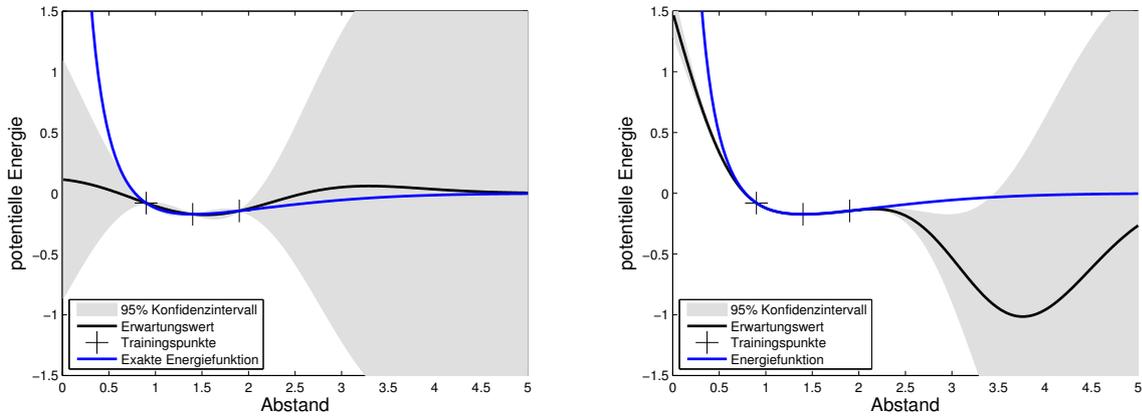
beziehungsweise die Kovarianzfunktion zwischen den Trainingsdaten

$$C(x_i, x_j) = \sigma_f^2 \cdot \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) + \sigma_n^2 \delta_{ij}.$$

Die darin enthaltenen *Hyperparameter* sind  $\boldsymbol{\theta} = \{\sigma_f, l, \sigma_n\}$ . Da der Messfehler  $\sigma_n$  über die Genauigkeit von *PSI3* vorgeben ist, können wir diesen auf  $\sigma_n = 10^{-3}$  festsetzen.

Wie die verbleibenden *Hyperparameter* angepasst werden können werden wird im weiteren Verlauf erläutert, zunächst setzen wir sie auf  $\boldsymbol{\theta} = \{\sigma_f, l\} = \{1, 1\}$ .

Das Ergebnis der *Gauß-Prozess-Regression* mit den *Trainingsdaten*  $\mathcal{D}$  ist in Abbildung 4.2(a) verdeutlicht. Die Energiefunktion stellt die über *PSI3* bestimmte Funktion wie in Abbildung 4.1 dar, der Erwartungswert und das Konfidenzintervall wurden über die *Gauß-Prozess-Regression* bestimmt.



(a) *Gauß-Prozess-Regression* ohne Gradienten.

(b) *Gauß-Prozess-Regression* mit Gradienten.

**Abbildung 4.2.:** Vergleich der Voraussage der *Gauß-Prozess-Regression* ohne und mit Gradienteninformation.

Eine Näherung an die exakte Energiefunktion ist zwar zu erkennen, allerdings wird der typische Verlauf in der Vorhersage der Funktion nicht deutlich. Weder der starke Abfall zu Beginn, noch das Energieminimum werden erfasst und diese Approximation ist deshalb für Anwendung und Berechnungen moleküldynamischer Prozesse noch nicht geeignet.

Folglich benötigen wir eine Verbesserung der Voraussage durch zusätzliche Informationen in den *Trainingsdaten*.

Wir wollen dazu ausnutzen, dass wir über das Programm *PSI3* auch die Gradienten zu den jeweiligen Abständen berechnen können und diese Information in die *Gauß-Prozess-Regression*, wie in Abschnitt 3.2.4 beschrieben, einbinden. Das bedeutet wir erweitern unsere *Trainingsdaten*  $\mathcal{D}$  um die Gradienten an der jeweiligen Stelle und erhalten damit

$$\mathcal{D}_g = \left\{ (0.9, E(0.9), G(0.9)); (1.4, E(1.4), G(1.4)); (1.9, E(1.9), G(1.9)) \right\},$$

mit  $G(x)$  als der über *PSI3* berechnete Wert des Gradienten an der Stelle  $x$ .

Die Kovarianzmatrix erweitern wir, wie in Abschnitt 3.2.4 beschrieben, durch die partiellen Ableitungen der Kovarianzfunktion

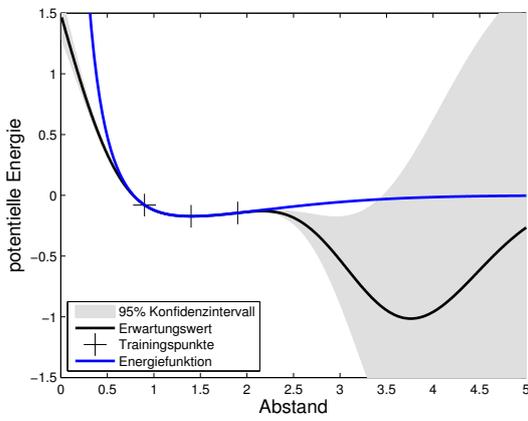
$$\frac{\partial Q(x_i, x_j)}{\partial x_i} = -\frac{\sigma_f^2}{l^2} (x_i - x_j) \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right),$$

$$\frac{\partial Q(x_i, x_j)}{\partial x_j} = \frac{\sigma_f^2}{l^2} (x_i - x_j) \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right)$$

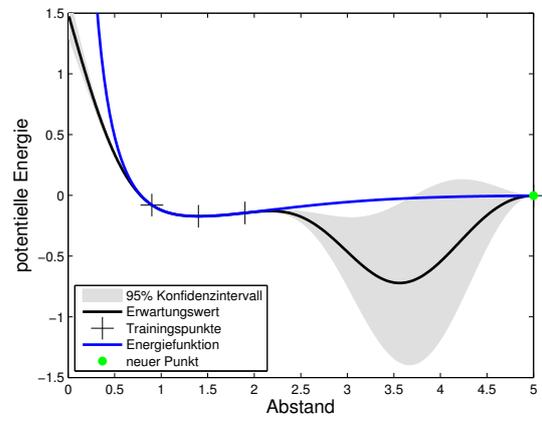
und

$$\frac{\partial^2 Q(x_i, x_j)}{\partial x_i \partial x_j} = -\frac{\sigma_f^2}{l^2} \left(1 - \frac{1}{l^2} (x_i - x_j)^2\right) \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right).$$

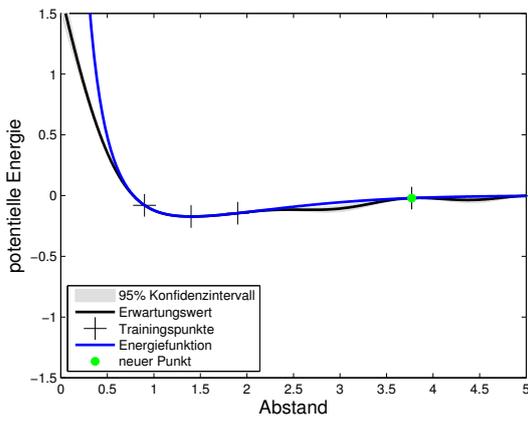
Da der Gradientenmessfehler  $\sigma_\kappa$  entsprechend Abschnitt 3.2.4 für die Berechnung der Kräfte für gewöhnlich eine Ordnung schlechter als der der Funktionswerte ist, setzen wir diesen auf  $\sigma_\kappa = 10^{-2}$ .



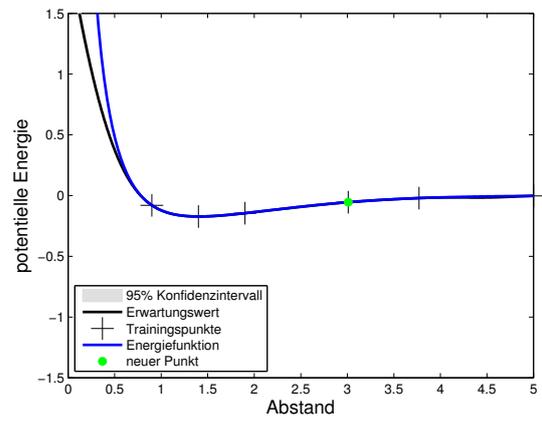
(a) Schritt 0.



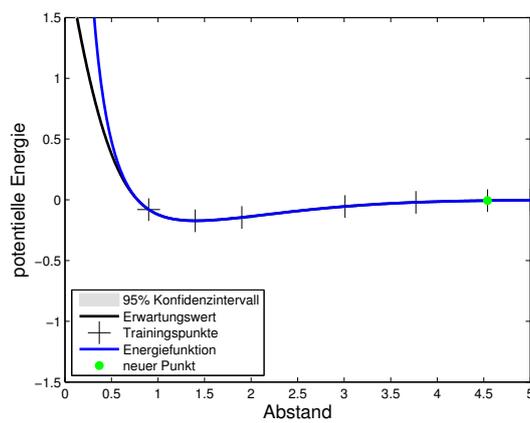
(b) Schritt 1.



(c) Schritt 2.



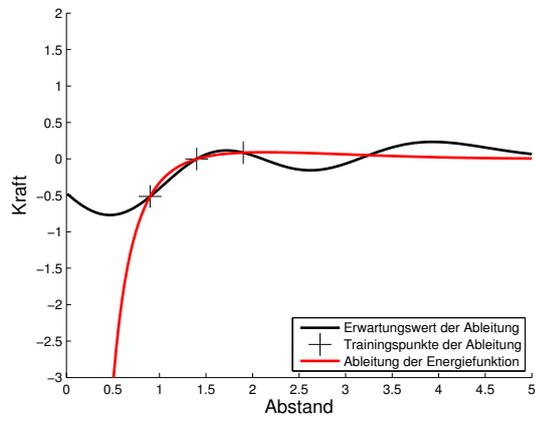
(d) Schritt 3.



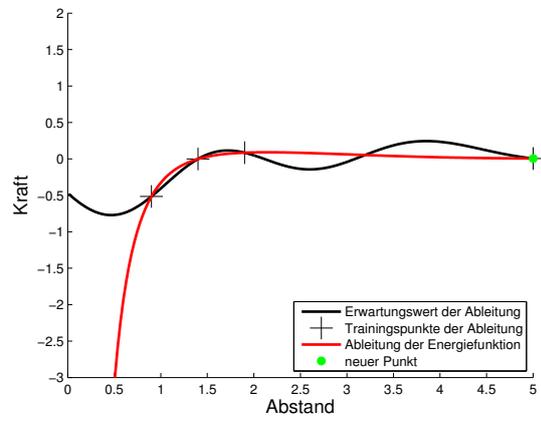
(e) Schritt 4.

Abbildung 4.3.: Learn on the Fly Verfahren zur Approximation der Energiefunktion.

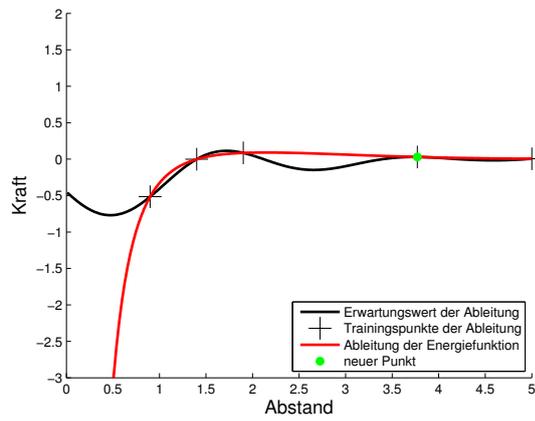
#### 4. Numerische Ergebnisse



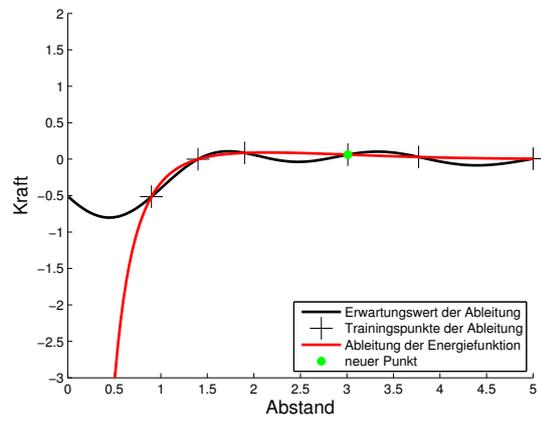
(a) Schritt 0.



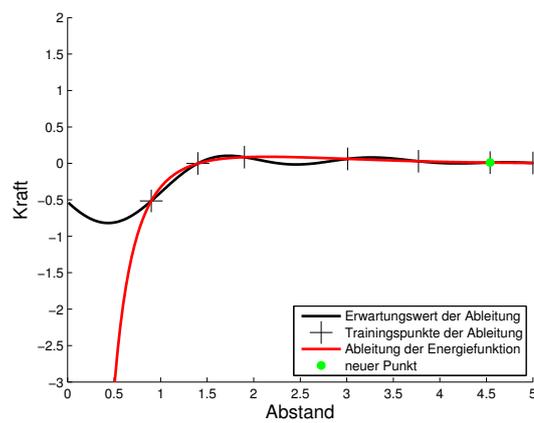
(b) Schritt 1.



(c) Schritt 2.



(d) Schritt 3.



(e) Schritt 4.

**Abbildung 4.4.:** *Learn on the Fly* Verfahren zur Approximation der Ableitung der Energiefunktion.

Das Ergebnis dieser verbesserten Vorhersage für die Energiefunktion ist Abbildung 4.2(b) dargestellt. Es ist zu erkennen, dass der typische Verlauf im Bereich zwischen 0.01 und 1.4 im Ansatz schon deutlich besser approximiert wird, allerdings werden Werte ab einem Abstand von 1.9 noch sehr ungenau und besitzen eine große Varianz. Für eine Verbesserung des Resultats an diesen Stellen müssen wir deshalb noch weitere Punkte zu den *Trainingsdaten* hinzufügen.

Dies werden wir mit Hilfe des in Abschnitt 3.4 beschriebenen *Learn on the Fly* Verfahrens umsetzen.

Dazu müssen wir einen Wert für die Fehlerschranke der Varianz  $C_V$  innerhalb des *Learn on the Fly* Verfahrens, analog zu Abschnitt 3.4, definieren, dieser soll im Folgenden  $C_V = 0.5 \cdot 10^{-7}$  betragen.

Die Abbildung 4.3 verdeutlicht die schrittweise Verbesserung der Approximation der Energiefunktion zweier Wasserstoffatome indem die *Trainingsdaten*, ausgehend von der Menge  $\mathcal{D}_g$ , durch geeignete Werte mittels *Learn on the Fly* Verfahren ergänzt werden.

Überschreitet der Wert der Varianz diese an einer *Teststelle*, so wird innerhalb des betrachteten Intervalls zwischen zwei *Trainingspunkten* diejenige neue *Teststelle* bestimmt, an der der größte Wert der Varianz vorliegt. Daraufhin wird dort ein neuer Energiewert und der zugehörige Gradient mit *PSI3* berechnet und zu den *Trainingsdaten* hinzugefügt.

Die Approximation der Ableitung der Energiefunktion verläuft nach dem gleichen Muster. Die aus den Schritten der Anpassung der Energiefunktion erhaltenen Trainingsdaten werden verwendet um die *Gauß-Prozess-Regression* mit Gradienten, wie in Abschnitt 3.2.4 beschrieben, durchzuführen. Das Resultat ist in Abbildung 4.4 aufgezeigt.

#### 4.2.1. Die Wahl der Hyperparameter

Die Anpassung der Hyperparameter der Kovarianzmatrix der *Gauß-Prozess-Regression* stellt eine weitere Möglichkeit zu Verbesserung der Voraussage dar.

Wie in Abschnitt 3.2.3 erläutert werden die *Hyperparameter* der Kovarianzfunktion angepasst, indem die zugehörige *Likelihoodfunktion* maximiert wird.

Dabei ist zu beachten, dass im Allgemeinen mehrere lokale Maxima der *Likelihoodfunktion* existieren können und das globale Maximum durch die beschriebenen Verfahren nicht immer gefunden wird. Aus diesem Grund sollten mehrere Startwerte ausprobiert werden um diejenigen zu finden, welche den größten Funktionswert besitzen.

Für das Beispiel des Wasserstoffmoleküls hat sich ein Startwert von  $\theta = \{\sigma_f, l\} \approx \{1, 0.5\}$  als beste Wahl erwiesen. Mittels des *BFGS-Verfahrens* zur Minimierung der *Negative Log Marginal Likelihood* Funktion wurden anschließend die optimalen Parameter  $\theta = \{\sigma_f, l\} \approx \{0.3, 1.0847\}$  bestimmt.

Um genauere Aussagen über die Genauigkeit des erhaltenen Ergebnisses machen zu können, führen wir eine Fehlermessung der Erwartungswerte der Energiefunktion und ihrer Ableitung an den *Trainingsdaten* ein, anhand derer wir die Genauigkeit unseres Ergebnisses messen können.

#### 4.2.2. Fehleranalyse

Zur Bestimmung der Genauigkeit der genäherten Funktion verwenden wir die mittlere Abweichung des über die *Gauß-Prozess-Regression* erhaltenen Ergebnisses von den exakt, über *PSI3*

berechneten, Funktionswerten. Das bedeutet, wir addieren die jeweiligen Abstände zwischen Auswertungen an gewählten Stellen der Näherungsfunktion und den entsprechend berechneten Auswertungen der exakten Funktion und teilen Anschließend durch die Anzahl der untersuchten Stellen. In unserem Fall entsprechen diese Stellen und zugehörigen Auswertungen den *Testdaten* der *Gauß-Prozess-Regression*.

Seien  $\mathbf{x}^* = (x_1^*, \dots, x_{n^*}^*)$  die betrachteten Stellen der *Testdaten*,  $f$  die exakte Funktion und  $\mathbb{E}[f_i^*]$  der Erwartungswert der *Gauß-Prozess-Regression* an der jeweiligen Stelle  $x_i^*$ , so wird der mittlere Fehler  $\bar{m}$  durch

$$\bar{m} = \sum_{i=1}^{n^*} \frac{|f(x_i^*) - \mathbb{E}[f_i^*]|}{n^*}$$

berechnet.

Die folgende Tabelle gibt die Fehler der mit Hilfe der *Gauß-Prozess-Regression* mit Gradienten vorausgesagten Funktionswerte für die Energie- und die Kraftfunktion ohne und mit Anpassung der *Hyperparameter* aus Abschnitt 4.2.1 an, indem *Testdaten* zwischen 0.5 und 5 mit einer Schrittweite von 0.01 betrachtet wurden.

	Fehler der Energiefunktion	Fehler der Kraftfunktion
ohne Hyperparameteranpassung	$8.51 \cdot 10^{-3}$	$1.24 \cdot 10^{-1}$
mit Hyperparameteranpassung	$2.75 \cdot 10^{-3}$	$0.89 \cdot 10^{-1}$

Es ist zu erkennen, dass über die Parameteranpassung die Voraussage sowohl für die Funktions-, als auch für die Gradientenwerte verbessert werden konnte.

Die Anpassung der *Hyperparameter* ermöglicht es uns folglich eine Verbesserung der Approximation ohne weitere *Trainingsdaten* hinzufügen zu müssen.

### 4.3. Die Anwendung für das Verlet-Verfahren

Die Positionen, Energien, Geschwindigkeiten und Kräfte der einzelnen Teilchen von Molekülen im Verlauf der Zeit können durch das in Abschnitt 2.2 beschriebene *Verlet-Verfahren* berechnet werden. Dies stellt die Grundlage der Simulation von Molekülen dar.

Wie in Algorithmus 1 beschrieben benötigen wir die Kraftauswertungen zu den jeweiligen Zeitpunkten um die Positionen und Geschwindigkeiten zu aktualisieren.

Da wir nun eine Methode entwickelt haben Potentialkurven und ihre zugehörigen Ableitungen anhand quantenmechanischer Daten zu fitten kann der Rechenaufwand durch Verwendung der daraus resultierenden Funktionen deutlich verringert werden.

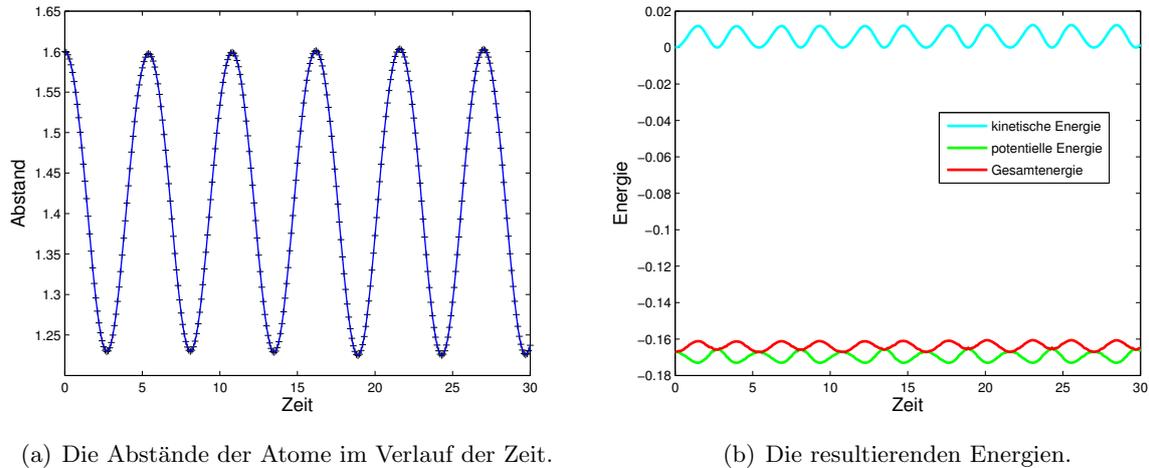
Im Folgenden wird die Effizienz dieser Methode anhand des Modellbeispiels des Wasserstoffmoleküls überprüft.

Dazu benötigen wir zunächst einmal, die für den *Verlet-Algorithmus* verwendeten, Startwerte. Dazu wählen wir zunächst für die Anfangspositionen

$$x_1 = (0 \ 0 \ 0) \quad \text{und} \quad x_2 = (0 \ 0 \ 1.6).$$

Da in diesem Abstand die potentielle Energie selbst für eine Bewegung sorgt, setzen wir die Geschwindigkeiten auf Null, das heißt

$$v_1 = (0 \ 0 \ 0) \quad \text{und} \quad v_2 = (0 \ 0 \ 0).$$



**Abbildung 4.5.:** Das Verlet-Verfahren für das Wasserstoffmolekül.

Wir verwenden im Folgenden dimensionslose Einheiten wie in [10] und setzen die Masse des Wasserstoffmoleküls auf  $m = 1$ .

Als Eingabe für den *Verlet-Algorithmus* verwenden wir nun das über *Learn on the Fly* Verfahren bestimmte Potential und die Funktion für die Kraftauswertungen, das bedeutet die Funktionen aus Abbildung 4.4 und 4.3.

Anschließend wird, wie in Abschnitt 3.4.1 beschrieben, eine weitere Fehlerüberprüfung durchgeführt um an den innerhalb des *Verlet-Verfahrens* benötigten Stellen auftretende Ungenauigkeiten mit einem zu hohen Wert zu verhindern.

Wir setzen dabei in dem Beispiel des Wasserstoffmoleküls diese Fehlerschranke  $C_{ver}$  auf  $C_{ver} = 0.1 \cdot 10^{-7}$ .

Für die oben gewählten Wert erhalten wir die in Abbildung 4.5 dargestellten Positionen für den Endzeitpunkt  $t_{end} = 30$  und eine Zeitschrittweite  $\delta t = 0.1$ . Die Genauigkeit ist zu diesem Startwert so hoch, dass die Fehlerschranke nicht überschritten wird und deshalb für diesen Fall kein neuer Punkt berechnet werden muss. In der Anwendung zeigt sich das erwartete Verhalten, zunächst ziehen sich die beiden Wasserstoffatome an, bis sie zum Energieminimum gelangen und anschließend ab einem Abstand unter 1.4 abstoßend wirken. Durch diese Kraft entfernen sich die Atome folglich wieder voneinander. Dadurch lässt sich der oszillierende Verlauf des in Abbildung 4.5(a) dargestellten Abstandes im Verlauf der Zeit erklären, was auch als „zittern“ bezeichnet wird, da die Atome um das Minimum in Bewegung bleiben.

Wir wollen des weiteren einen Fall betrachten, bei dem innerhalb des *Verlet-Verfahrens* ein neuer Punkt hinzu gelernt werden muss.

Dazu setzen wir die in Abschnitt 4.2 beschriebene Fehlerschranke für  $C_V$  auf  $C_V = 0.5 \cdot 10^{-3}$  herunter. Die Fehlerschranke für  $C_{ver}$  setzen wir auf  $C_{ver} = 0.1 \cdot 10^{-5}$ .

Bei der Anwendung des *Learn on the Fly* Verfahrens für den *Verlet-Algorithmus* entstehen dabei die in Abbildung 4.6(a) und (c) dargestellten Funktionen für das Potential mit dem wir starten. Zu erkennen ist, dass die Funktion der Kraft bei einem Abstand von 2.5 noch sehr ungenau genähert ist. Wenn wir diesen nun als Anfangsabstand für das *Verlet-Verfahren* wählen, also zum Beispiel die Startpunkte

$$x_1 = (0 \ 0 \ 0) \quad \text{und} \quad x_2 = (0 \ 0 \ 2.5),$$

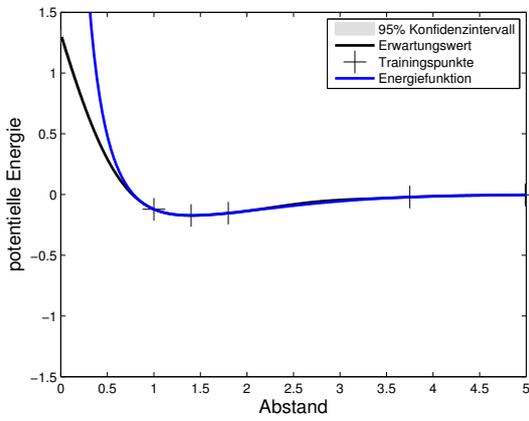
und die Anfangsgeschwindigkeiten und die Masse wie in dem Beispiel zuvor, so erhalten wir ein falsches Ergebnis für die Bewegung der Atome, dies ist in Abbildung 4.6(e) dargestellt. Da die Funktion der Kraft einen falschen Startwert zurück gibt stoßen sich die Atome direkt zu Beginn des *Verlet-Algorithmus* ab, diese Kraft verstärkt sich im weiteren Verlauf und führt letztendlich dazu, dass sie nicht das Energieminimum gelangen, was ein falsches Ergebnis darstellt.

Innerhalb des Verfahrens wird deshalb an dieser Stelle ein neuer Punkt mit dem zugehörigen Gradienten hinzulernt und die *Gauß-Prozess-Regression* mit den erweiterten Daten neu gestartet. Das Ergebnis für die Energie- und die Kraftfunktion ist in Abbildung 4.6(b) und (d) dargestellt, die neuen Werte für die Abstände im Verlauf der Zeit in (f).

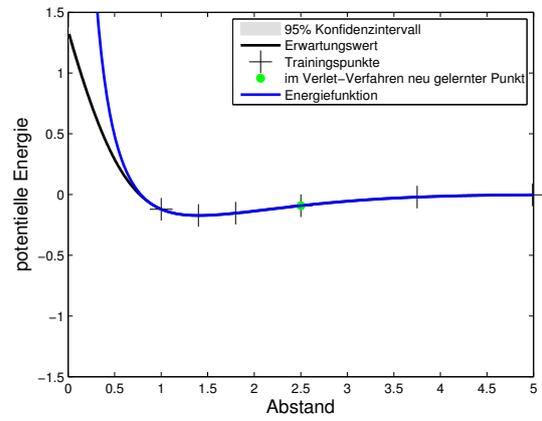
Hierbei zeigt sich wie zuvor in Abbildung 4.5 wieder das „zittern“ des Atoms. Durch das *Learn on the Fly* Verfahren haben wir also eine Methode gefunden entscheidende Fehler, die wir durch eine zu ungenaue Approximation der Energie- und Kraftfunktion verursachen, zu verhindern und somit auch nur dort konkret Daten berechnet werden müssen, an denen dies für das Ergebnis notwendig ist.

Dadurch können wir in Bezug auf größere Systeme von Teilchen enorm viel Rechenaufwand und Kosten einsparen, da die aufwendigen Berechnungen der Energien und Gradienten über die Lösung der *Schrödingergleichung* nur an wenigen Stellen benötigt werden.

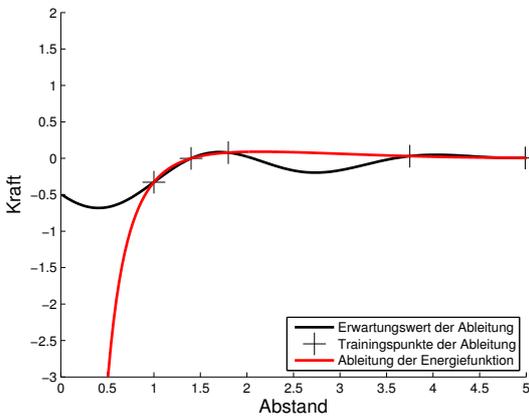
Hierbei ist zu beachten, dass die im vornherein durchgeführte Anpassung der Energie- und Kraftfunktion nur für Moleküle durchgeführt werden kann, die eine niedrigen Anzahl an Atomen besitzen. Da wir in den einzelnen Iterationsschritten jeweils alle Intervalle betrachten, würde in höheren Dimensionen der Rechenaufwand viel zu groß werden. Für höhere Dimensionen wird deshalb nur das *Learn on the Fly* Verfahren für das *Verlet-Verfahren* verwendet.



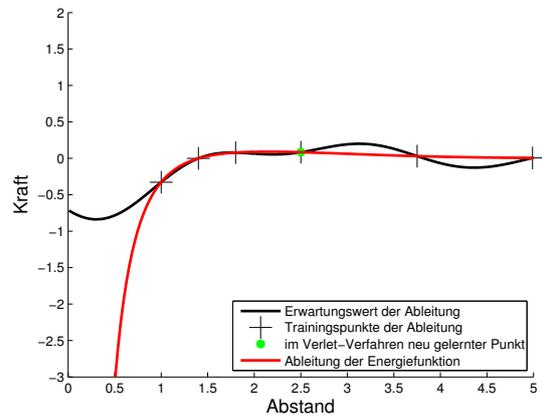
(a) Startfunktion für die Energie.



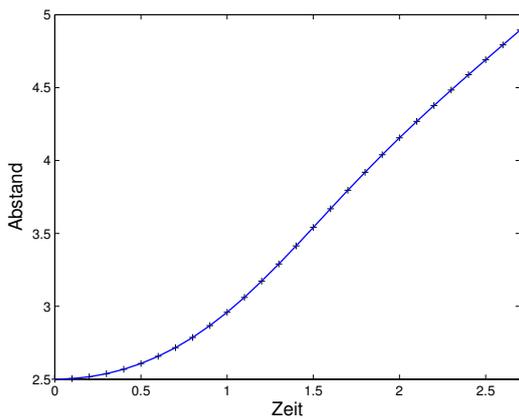
(b) Energiefunktion nach der Anpassung.



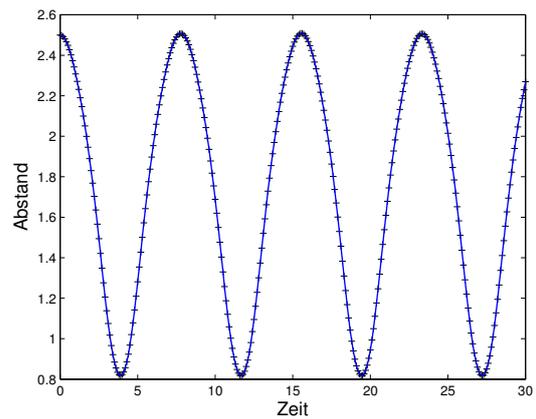
(c) Startfunktion für die Kraft.



(d) Kraftfunktion nach der Anpassung.



(e) Die Abstände der Atome im Verlauf der Zeit ohne neu gelerntem Punkt.



(f) Die Abstände der Atome im Verlauf der Zeit mit neu gelerntem Punkt.

**Abbildung 4.6.:** *Learn on the Fly* Verfahren innerhalb des *Verlet-Verfahrens* zur Approximation der Energie- und der Kraftfunktion.



## 5. Zusammenfassung und Ausblick

### Zusammenfassung

In dieser Arbeit beschäftigten wir uns mit dem Fitten von Potentialkurven an quantenmechanische Daten. Das Ziel war es die Potentialfläche, ausgehend von der *Born-Oppenheimer-Näherung* effizient zu approximieren, um diese für die Moleküldynamik verwenden zu können. Potentiale werden benötigt um Berechnungen und Computersimulationen auf molekularer Ebene durchzuführen, um somit den Aufwand und die Kosten für reale Experimente zu sparen oder Versuche durchzuführen, die in der Realität gar nicht möglich sind. Dies stellt einen entscheidenden Punkt in vielen Bereichen der Wissenschaft dar.

Wir beschrieben die Näherung der Gesamtpotentialfläche über eine Summe von Mehrkörperpotentialen, deren einzelne Summanden eine Lösung der *Schrödingergleichung* zu verschiedenen Kernkonfigurationen voraussetzen. Da diese allerdings zu komplex und analytisch nur für wenige Spezialfälle zu lösen ist, stellten wir verschiedene Näherungen für die Lösung vor.

Als Grundlage diente dazu die *Born-Oppenheimer-Näherung*, welche die Kerne als ortsfest betrachtet und lediglich die Bewegung der Elektronen über die Gesetze der Quantenmechanik berechnet. Da Lösungen allerdings immer noch zu aufwändig für konkrete Anwendungen sind, wurden die häufigsten Verfahren zur weiteren Approximation der *Schrödingergleichung*, die *Hartree-Fock-Verfahren* und *Post-Hartree-Fock-Verfahren*, beschrieben.

Weiterhin wurde das Programm *PSI3* beschrieben, welches uns die konkrete Berechnung von Näherungen der potentiellen Energien und den zugehörigen Gradienten vieler Kernkonfigurationen innerhalb von Molekülen ermöglichte. Aus der approximierten Lösung der *Schrödingergleichung* lassen sich die potentiellen Energien bestimmen, woraus sich anschließend ein Potential entwickeln lässt, welches für das *Verlet-Verfahren* zur Simulation von Molekülen benötigt wird, dessen wesentliche Schritte wir aufzeigten.

Dieses Potential exakt zu bestimmen, beziehungsweise fein aufgelöst auszurechnen, erfordert allerdings einen sehr hohen Rechenaufwand oder ist in vielen Fällen gar nicht möglich, weshalb eine effiziente Bestimmung anhand ausgewählter quantenmechanischer Daten von großer Bedeutung ist.

Wir wählten ein Verfahren aus dem *maschinellen Lernen* und beschrieben dazu zunächst die Grundlagen von Lernalgorithmen. Anschließend beschäftigten wir uns mit *Regressionen* im *maschinellen Lernen* und einem Modell zur Bestimmung von Funktionen anhand ausgewählter *Trainingsdaten*.

Als Verfahren zur Approximation der Energiefunktion verwendeten wir die *Gauß-Prozess-Regression*. Zunächst beschrieben wir dazu die wesentlichen Schritte und die zugrundeliegenden Modellannahmen dieser Form von maschinellem Lernalgorithmus zur nichtparametrischen Regression für das lineare Modell, welche es ermöglicht beliebige Funktionswerte anhand der *Trainingsdaten* zu approximieren. Die Voraussetzung ist die Annahme von *gaußverteilten* Daten, sodass wir anhand einer gewählten Erwartungswert- und Kovarianzfunktion eine Voraussage über die Berechnung von Erwartungswerten und den daraus resultierenden Abweichungen über Varianzen an ausgewählten *Testdaten* treffen können. Der wesentliche Vorteil in *Gauß-Prozessen*

besteht darin, dass wir Voraussagen treffen können, ohne die Funktion konkret beschreiben zu müssen, sondern lediglich die Erwartungswert- und Kovarianzfunktion. Dadurch beschränkt sich die Dimension des zu lösenden Problems nur noch auf die Anzahl der *Trainingsdaten*. Dies bringt für die Moleküldynamik einen entscheidenden Nutzen, da wir so auch Moleküle betrachten können, die aus beliebig vielen Teilchen bestehen.

Wir stellten den Spezialfall des *Gauß-Kerns* als mögliche Kovarianzfunktion vor, welche wir anschließend in unserem Modell zum Fitten der Energiekurven verwendeten.

Anschließend stellten wir ein mögliches Verfahren zur Anpassung der in den Kovarianzfunktionen enthaltenen *Hyperparameter* vor, indem wir die *Likelihoodfunktion* ermittelten, welche die Wahrscheinlichkeit angibt die *Trainingsdaten* zu erhalten, wenn gewisse Werte für die Parameter bereits gegeben sind. Durch die Maximierung dieser Funktion nach den Parametern erhalten wir die optimalen Werte in Bezug auf die gegebenen *Trainingsdaten*.

Weiterhin wurden die Voraussagen der Funktionswerte und Bestimmung der Varianzen durch die Verwendung der Gradienten an den betreffenden *Trainingsdaten* verbessert, was uns zudem eine Möglichkeit bot die direkte Voraussage der Ableitung der Funktion an den betreffenden Stellen mit Hilfe der *Gauß-Prozess-Regression* zu berechnen.

Daraufhin wurde das konkrete Modell in Bezug auf Energiefunktionen beschrieben, was es uns ermöglichte die Auswertungen der Potentialkurve und der zugehörigen Kraftfunktion anhand ausgewählter quantenmechanischer Daten zu nähern, ohne diese konkret parametrisieren zu müssen. Die Voraussage erfolgt allein durch die Wahl eines Erwartungswert- und einer Kovarianzfunktion der zugrunde liegenden Daten. Zudem erhalten wir an jeder Stelle eine Schätzung für den gemachten Fehler, wodurch wir ein Maß für die Genauigkeit des Resultats definieren können.

Um dieses Verfahren auf die für die Moleküldynamik anzuwenden entwickelten wir eine effiziente Methode, die *Trainingsdaten* so zu wählen, dass die Energiefunktion und die Funktion zur Bestimmung der Kräfte durch so wenig *Trainingsdaten* wie möglich sehr genau zu Fitten. Dazu verwendeten wir ein *Learn on the Fly* Verfahren, das ausgehend von einer gewählten Menge an Startwerten für die *Trainingsdaten* nur an solchen Stellen Punkte hinzufügt, an welchen der Fehler groß ist, um die *Gauß-Prozess-Regression* effizient für die Approximation ausnutzen zu können. Außerdem wurde eine Methode vorgestellt, in welcher innerhalb des *Verlet-Verfahrens* zusätzlich zu den *Trainingsdaten* neue Daten hinzugelern werden, wenn das Verfahren sie für die Berechnungen benötigt und die Varianz an der jeweils betrachteten Stelle zu groß ist.

Anschließend wurden die numerischen Ergebnisse anhand des Wasserstoffmoleküls als Modellbeispiel erläutert. Die Verbesserungen der Voraussagen der Funktionswerte der Potentialkurve über die Gradienteninformation, das Lernen von Stellen, an denen zusätzliche Daten hinzugefügt werden müssen und die Anpassung der Parameter wurden nacheinander vorgestellt und analysiert. Daraufhin zeigten wir die Anwendung auf das *Verlet-Verfahren*, indem das *Learn on the Fly* Verfahren für verschiedene Startwerte innerhalb der *Verlet-Verfahrens* durchgeführt wurde.

Es konnte gezeigt werden, dass die vorgestellte Methode sich als äußerst effizient in Bezug auf die Bestimmung von Potentialkurven und deren Verwendung innerhalb des *Verlet-Verfahrens* erweist. Ausgehend von bekannten Informationen der Quantenchemie wird eine geringe Anzahl an Daten als Startwert für die *Gauß-Prozess-Regression* gewählt und anschließend anhand des Fehlerschätzers an den Stellen ein neuer Datensatz hinzugefügt, der während des Verfahrens benötigt wird und an welchen der Fehler zu groß ist.

---

In der vorliegenden Arbeit wurde somit der erste Schritt zur Bestimmung der Approximation von *Born-Oppenheimer-Potentialhyperflächen* durchgeführt. Die Resultate anhand des Modellbeispiels, dem Wasserstoffmolekül, zeigten, dass das vorgestellte Verfahren durchaus aussichtsreich ist, auch in Bezug auf die Bestimmung der Potentialflächen von Mehrteilchensystemen. Dazu müssen die Potentiale für die Komponenten der über den *HDMR*-Ansatz bestimmten partikelweisen Zerlegung der Gesamt-Potentialfläche berechnet werden, woraus sich ein Ansatz für weiterführende Arbeiten ergibt.

Eine weitere Anwendung des, über die *Gauß-Prozess-Regression* gelernten, Potentials ist die sogenannte *vibrational Schrödingergleichung* [8]. Diese wird eingesetzt um die Eigenschwingungen eines Systems zu beschreiben und verwendet das elektronische Potential, also die durch unser Verfahren bestimmte *Born-Oppenheimer-Potentialfläche*.

## Ausblick

Wir haben in dieser Arbeit zunächst die Potentialkurven für Paarpotentiale betrachtet und ihre Anwendbarkeit für die Moleküldynamik aufgezeigt. Da die Gesamtpotentialfläche für die meisten Moleküle allerdings auch aus Potentialen von Tripeln, Quadrupeln, . . . gebildet wird stellt es eine interessante Erweiterung dar, die von uns vorgestellte Methode zum Fitten von Potentialen auch für höhere Dimensionen durchzuführen. In diesem Fall erhält die *Gauß-Prozess-Regression* wie schon beschrieben eine mehrdimensionale Eingabe, welche die einzelnen Abstände der Teilchen im betrachteten Molekül repräsentiert. Somit wäre es dann möglich Potentialhyperflächen für verschiedene mehratomige Moleküle zu nähern.

Eine weitere Fragestellung ergibt sich durch die nähere Betrachtung der Kovarianzfunktion und möglicher Alternativen. In dieser Arbeit verwendeten wir den *Gauß-Kern* als kanonische Kern- beziehungsweise Kovarianzfunktion. Die Parameter wurden im Rahmen dieser Arbeit als allgemeiner Faktor betrachtet, welcher für alle Punkte den gleichen Wert besitzt. Ein Ansatzpunkt für eine Verbesserung der Parameteranpassung und der Genauigkeit der daraus resultierenden Voraussagen wäre eine unterschiedliche Wahl in Bezug darauf wo sich die Punkte gerade befinden. Wie diese Parameter innerhalb der Kovarianzfunktion genau gewählt werden stellt dabei ein interessantes Problem dar. Des weiteren wäre eine adaptive Anpassung der Parameter während des Lernprozesses vorstellbar.

Obwohl sich die Wahl von *Gauß-Kernen* als durchaus sinnvoll erwies, wäre eine spannende Frage, ob eine alternative Wahl Verbesserungen in den Voraussagen ermöglicht, weshalb das systematische Ausprobieren weiterer Kovarianzfunktionen einen spannenden Ansatzpunkt für folgende Arbeiten darstellt.

In Bezug auf die *Gauß-Prozess-Regression* mit Gradienten wäre darüber hinaus die Frage interessant, ob sich die zweite Ableitung analog einbinden lässt, da es für bestimmte Verfahren zur Approximation einer Lösung der *Schrödingergleichung* auch möglich ist die zweite Ableitung beziehungsweise die *Hessematrix* zu bestimmen.

So verbleiben viele spannende Anregungen für die Problemstellung in künftigen Arbeiten.



## A. Anhang

### A.1. Bayes-Statistik

Die *Bayes-Statistik* [15] stammt aus dem Bereich der Inferenz-Statistik. Diese wird auch als schließende Statistik bezeichnet und beschäftigt sich mit dem Schluss von erhobenen Daten einer Stichprobe auf die Werte in der Grundgesamtheit.

Die Grundlage der *Bayes-Statistik* ist die Wahrscheinlichkeitstheorie. Neben der Berechnung der Wahrscheinlichkeiten zufälliger Ereignisse, ermöglicht diese Form der Statistik es ebenfalls die Wahrscheinlichkeit von Aussagen zu definieren, das heißt ein Maß für die Plausibilität einer Aussage anzugeben.

Der Grundgedanke besteht darin, vorhandenes Wissen in die Berechnung von Wahrscheinlichkeiten einzubauen. Konkret wird dabei zunächst vorhandenes Vorwissen in Form einer (subjektiven), sogenannten *Prior Verteilung*, quantifiziert und anschließend werden Daten erhoben, die weitere Informationen über dieses Ereignis enthalten. Mit Hilfe der *Regel von Bayes* kombiniert man diese Daten mit der *Prior Verteilung* und erhält somit die sogenannte *Posterior Verteilung*, die eine an die Daten „angepasste“ Wahrscheinlichkeitsverteilung beschreibt.

Der zentrale Punkt der *Bayes-Statistik* ist die *Regel von Bayes* und damit die Verwendung von bedingten Wahrscheinlichkeiten.

**Bemerkung A.1.** Wir bezeichnen die bedingte Wahrscheinlichkeit eines Ereignisses  $A$ , bei gegebenem Ereignis  $B$  mit

$$P(A|B).$$

Die *Regel von Bayes* beschreibt die Bestimmung der *Posterior Wahrscheinlichkeit* eines Ereignisses, anhand der angenommenen *Prior Wahrscheinlichkeit*, resultierend aus den über das Ereignis bereits bestehenden Informationen, der Wahrscheinlichkeit der Informationen unter der Bedingung, dass das Ereignis bereits eingetreten ist, der sogenannten *Likelihood*, und einer Normierungskonstanten.

**Satz A.1** (Regel von Bayes). Für zwei Ereignisse  $A$  und  $B$  mit  $P(B) > 0$  lautet die bedingte Wahrscheinlichkeit

$$\underbrace{P(A|B)}_{\text{Posterior}} = \frac{\overbrace{P(B|A)}^{\text{Likelihood}} \overbrace{P(A)}^{\text{Prior}}}{P(B)}.$$

Die Erweiterung des Theorems folgt aus der Anwendung der Produktregel der Wahrscheinlichkeitstheorie [15]. Die Wahrscheinlichkeit  $P(A|B, C)$  beschreibt hierbei die Wahrscheinlichkeit eines Ereignisses  $A$ , gegeben den Ereignissen  $B$  und  $C$  und kann durch die erweiterte *Regel von*

Bayes [17] folgendermaßen bestimmt werden

$$\underbrace{P(A|B, C)}_{\text{Posterior}} = \frac{\overbrace{P(B|A, C)}^{\text{Likelihood}} \overbrace{P(A|C)}^{\text{Prior}}}{P(B|C)} = \frac{P(B|A, C)P(A|C)}{\sum_{\substack{j \in I \\ P(A_j) \neq 0}} P(B|A_j, C)P(A_j|C)}.$$

Dies lässt sich analog zu Satz A.1 auch für Wahrscheinlichkeitsdichten von diskreten und stetigen Zufallsvariablen formulieren.

**Definition A.1.** Sei  $X$  eine reellwertige Zufallsvariable. Die Funktion  $p : \mathbb{R} \rightarrow \mathbb{R}$  heißt Wahrscheinlichkeitsdichte von  $X$ , falls gilt

$$P(a \leq X \leq b) = \int_a^b p(x) dx,$$

für reelle Zahlen  $a < b$ .

Betrachten wir mehrdimensionale Zufallsvariablen, also Zufallsvektoren, so ergibt sich die folgende Definition.

**Definition A.2.** Sei  $\mathbf{X} = (X_1, \dots, X_n)$  ein Zufallsvektor, so bezeichnet eine messbare Funktion  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  die Wahrscheinlichkeitsdichte von  $\mathbf{X}$ , falls gilt

$$P(\mathbf{X} \in [a_1, b_1] \times \dots \times [a_n, b_n]) = \int_{a_1}^{b_1} \dots \int_{a_n}^{b_n} p(x_1, \dots, x_n) dx_n \dots dx_1$$

Seien  $\mathbf{x}$  und  $\mathbf{y}$  diskrete oder stetige Zufallsvektoren, so ergibt sich die *verallgemeinerte Bayes Regel*

$$\underbrace{p(\mathbf{x}|\mathbf{y}, C)}_{\text{Posterior}} = \frac{\overbrace{p(\mathbf{y}|\mathbf{x}, C)}^{\text{Likelihood}} \overbrace{p(\mathbf{x}|C)}^{\text{Prior}}}{p(\mathbf{y}|C)} = \frac{p(\mathbf{y}|\mathbf{x}, C)p(\mathbf{x}|C)}{\int p(\mathbf{y}|\mathbf{x}', C)p(\mathbf{x}'|C) d\mathbf{x}'}$$

Hintergründe zu weiteren Einzelheiten dieses Themas können in [15] nachgelesen werden.

## A.2. Kenngrößen für Zufallsvariablen

Um Berechnungen mit Zufallsvariablen durchzuführen, benötigen wir Aussagen über ihre Eigenschaften. Zu den wichtigsten zählen hierbei der Erwartungswert und die Varianz [14].

**Definition A.3** (Erwartungswert). Sei  $X$  eine diskrete Zufallsvariable,  $I$  eine beliebige Indexmenge. Dann ist der Erwartungswert von  $X$  definiert durch

$$\mathbb{E}[X] := \sum_{k \in I} c_k P(X = x_k) = \sum_{k \in I} c_k p(x_k).$$

**Definition A.4** (Varianz). Sei  $X$  Zufallsvariable. Dann ist die Varianz von  $X$  definiert durch

$$\mathbb{V}[X] := \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2.$$

### A.3. Matrizen

**Definition A.5** (Positiv semidefinite Matrix). *Eine  $(n \times n)$ -Matrix  $M$  heißt positiv semidefinit, wenn gilt*

$$x^T Ax \geq 0, \text{ für alle } x \neq 0, x \in \mathbb{R}^n.$$



## Literaturverzeichnis

- [1] Webseite der *NIST Computational Chemistry Comparison and Benchmark Database*, 2011. <http://cccbdb.nist.gov/> (Letzter Zugriff 7. Februar 2012).
- [2] Webseite des *PSI3 Projekts*, 2011. <http://www.psicode.org/> (Letzter Zugriff 7. Februar 2012).
- [3] E. Alpaydin. *Maschinelles Lernen*. Oldenbourg Wissenschaftsverlag, 2008.
- [4] A.P. Bartók, M.C. Payne, R. Kondor, and G. Csányi. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Physical review letters*, 2010.
- [5] A. Bartók-Pártay. Gaussian approximation potential: an interatomic potential derived from first principles quantum mechanics. Dissertation, University of Cambridge, 2009.
- [6] U. Böhme. *Quantenchemische Studien zur Struktur und Reaktivität von Carben- und Silylenverbindungen des Titans und Eisens*. BoD-Books on Demand, 2005.
- [7] C. K. I. Williams C. E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.
- [8] O. Christiansen. Vibrational structure theory: new vibrational wave function methods for calculation of anharmonic vibrational energies and vibrational contributions to molecular properties. *Phys. Chem. Chem. Phys.*, 9(23):2942–2953, 2007.
- [9] M.N. Gibbs. Bayesian gaussian processes for regression and classification. Dissertation, 1997.
- [10] M. Griebel, S. Knapek, and G.W. Zumbusch. *Numerical simulation in molecular dynamics: numerics, algorithms, parallelization, applications*, volume 5. Springer Verlag, 2007.
- [11] J. Hamaekers. Ebene-Wellen basiertes, adaptives und paralleles Verfahren für die Dichtefunktionaltheorie. Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, Bonn, Germany, 2002.
- [12] A. Heßelmann. Die Berechnung von intermolekularen Wechselwirkungsbeiträgen mit Dichtefunktional- und Brueckner-Coupled-Cluster-Methoden. Dissertation, 2003.
- [13] T. Kelterbach. Zum Hartree-Fock-Verfahren und seiner numerischen Umsetzung. Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, 1999.
- [14] A. Klenke. *Wahrscheinlichkeitstheorie*. Springer, 2008.
- [15] K.R. Koch. *Einführung in die Bayes-Statistik*. Springer Verlag, 2000.
- [16] B. Li M. Y. Hayes and H. Rabitz. Estimation of molecular properties by high-dimensional model representation. *Journal of Physical Chemistry*, 110:264–272, 2006.

- [17] D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.
- [18] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Verlag, 1999.
- [19] C.E. Rasmussen. Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. 2003.
- [20] F. Schwabl. *Quantenmechanik für Fortgeschrittene (qm ii)*. Springer, 2008.
- [21] J. Stare and G.G. Balint-Kurti. Fourier grid hamiltonian method for solving the vibrational schrödinger equation in internal coordinates: Theory and test applications. *The Journal of Physical Chemistry A*, 107(37):7204–7214, 2003.
- [22] R. Storm. *Wahrscheinlichkeitsrechnung, mathematische Statistik und statistische Qualitätskontrolle*. Hanser Verlag, 2007.
- [23] C.K.I. Williams and C.E. Rasmussen. Gaussian processes for regression. 1996.
- [24] D. Wissel. Die Diskrete Gauß-Transformation - schnelle Approximationsverfahren und Anwendungen in hohen Dimensionen. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, April 2008.