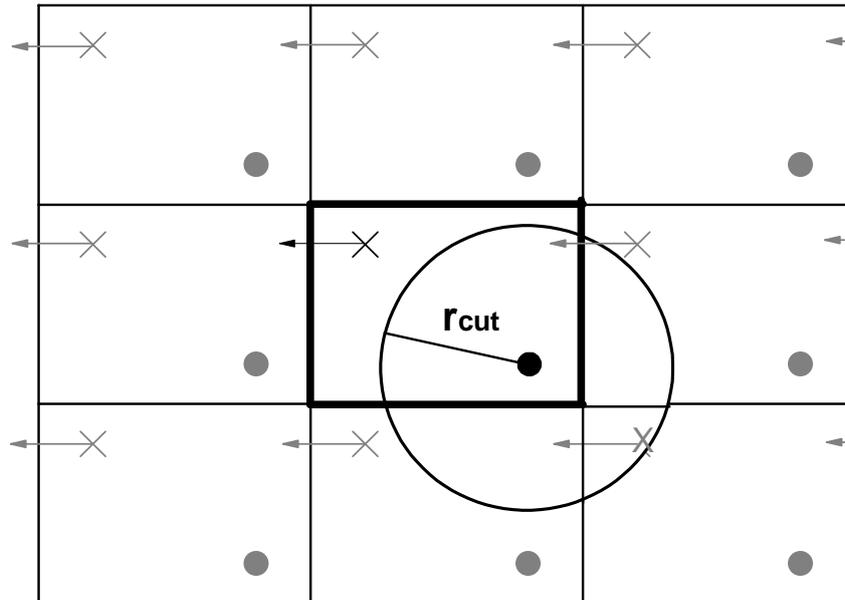


Monte Carlo Simulation von Gasen und Flüssigkeiten



Jonathan Hager & Reinhard Hentschke

6. August 2012

Inhaltsverzeichnis

1	Monte Carlo Integration	2
2	Metropolisalgorithmus	9
3	Simulation von Gasen und Flüssigkeiten mit wenigen Teilchen	14
4	MC-Simulation eines Lennard-Jones Fluids bestehend aus Punktteilchen	19
4.1	NVT-Ensemble	19
4.2	NPT-Ensemble	27
4.3	μ VT-Ensemble	33
4.4	Gibbs-Ensemble	38
5	MC-Simulation eines Lennard-Jones Fluids bestehend aus starren Molekülen	44
5.1	NVT-Ensemble	49
5.2	NPT-Ensemble	52
6	Quellen	53

1 Monte Carlo Integration

Im folgenden Kapitel wird es um sogenannte Monte Carlo Methoden gehen. Dies sind auf Zufallszahlen basierende Verfahren, mit denen unter anderem hochdimensionale Integrale gelöst werden können.

Zunächst werden Erwartungswerte bei diskreten Zufallsexperimenten betrachtet, dies wird dann auf kontinuierliche Fälle übertragen, um schließlich Integrale lösen zu können.

Erwartungswerte:

Der Erwartungswert einer Zufallsvariablen A , welche bei einem Zufallsexperiment die Werte (a_1, \dots, a_k) annehmen kann, lautet:

$$\langle A \rangle = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{i=1}^M A^{(i)} \quad (1.1)$$

$A^{(i)}$ bezeichnet den gemessenen Wert von A im i -ten Versuch.

Führt man die Häufigkeiten $h_1(M), \dots, h_k(M)$ ein, mit denen nach M Experimenten die Werte (a_1, \dots, a_k) angenommen werden, ergibt sich:

$$\langle A \rangle = \lim_{M \rightarrow \infty} \frac{1}{M} (h_1(M)a_1, \dots, h_k(M)a_k) \quad (1.2)$$

$$= \sum_{j=1}^k a_j w_j \quad (1.3)$$

Hierbei bezeichnet $w_i = \lim_{M \rightarrow \infty} \frac{h_i(M)}{M}$ die Wahrscheinlichkeit, dass bei dem Zufallsexperiment der Wert a_i auftritt. Für die Wahrscheinlichkeiten gilt:

$$\sum_{i=1}^k w_i = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{i=1}^k h_i \quad (1.4)$$

$$= 1 \quad (1.5)$$

Beispiel: Würfeln und Erwartungswert

Angenommen es wird unendlich oft gewürfelt. Was ist in diesem Fall die mittlere Augenzahl? Nach einem Wurf kann der Würfel 6 verschiedene Werte ($a_1 = 1, a_2 = 2, \dots, a_6 = 6$) annehmen. Die Wahrscheinlichkeit für jeden Wert betrage $w_i = \frac{1}{6} \Rightarrow \sum_{i=1}^6 w_i = 1$. Mit Gleichung (1.3) ergibt sich für den Erwartungswert $\langle A \rangle$:

$$\begin{aligned}\langle A \rangle &= \sum_{j=1}^k a_j w_j \\ &= \sum_{j=1}^6 a_j \frac{1}{6} \\ &= 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} \\ &= 3.5\end{aligned}$$

Kontinuierlicher Fall:

Wird nun eine kontinuierliche Funktion $f : R \rightarrow R$ mit der zugehörigen Wahrscheinlichkeitsdichtefunktion $w : R \rightarrow [0 : 1]$

$$\int_{-\infty}^{+\infty} w(x) dx = 1 \quad (1.6)$$

betrachtet, geht die Summe in Gleichung (1.3) über in ein Integral.

$$\langle f \rangle_w = \int_{-\infty}^{+\infty} f(x) w(x) dx \quad (1.7)$$

Der Erwartungswert kann dann gemäß Gleichung (1.1) approximiert werden, in dem M-viele x_i , welche nach $w(x)$ verteilt sind, gewählt, in $f(x)$ ausgewertet und gemittelt werden.

$$\langle f \rangle_w \approx \frac{1}{M} \sum_{i=1}^M f(x_i, w - \text{verteilt}) \quad (1.8)$$

Monte Carlo Integration:

Mit dem Wissen über Erwartungswerte lassen sich nun Integrale approximieren. Sei I das zu berechnende Integral:

$$I = \int_a^b f(x) dx \quad (1.9)$$

Erweitert man f mit einer Wahrscheinlichkeitsdichtefunktion $w(x)$, für die gilt:

- $\int_a^b w(x) = 1$
- $w(x) > 0$ für alle $x \in [a, b]$

ergibt sich für I :

$$I = \int_a^b \underbrace{\frac{f(x)}{w(x)}}_{g(x)} w(x) dx \quad (1.10)$$

Dies lässt sich dann wie oben beschrieben mit Gleichung (1.8) approximieren.

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_{MC} \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i, w - \text{verteilt})}{w(x_i, w - \text{verteilt})} \quad (1.11)$$

Die Wahl von $w(x)$ ist prinzipiell beliebig, jedoch gewinnt das Verfahren an Effizienz, wenn man $w(x)$ so wählt, dass aus Bereichen, in denen $f(x)$ große Funktionswerte annimmt, auch viele x_i gezogen werden. Entsprechend in Bereichen kleiner Funktionswerte weniger. Dieses Vorgehen nennt man **importance sampling**.

Beispiel: Monte Carlo Integration mittels Gleichverteilung

Die einfachste Variante einer Monte Carlo Integration ist eine Gleichverteilung.

$$w(x) = \frac{1}{b-a} \quad (1.12)$$

$$\Rightarrow I = \frac{b-a}{N} \sum_{i=1}^N f(x_i) \quad (1.13)$$

Diese Wahl hat allerdings im niederdimensionalen Fall keinen Vorteil gegenüber einer Integrationsmethode, bei der feste Stützstellen gewählt werden. Dies sieht man an Abbildung (1.2) am Beispiel von $f(x) = e^{-30x^2}$

Beispiel: Monte Carlo Integration mittels importance sampling

Wie im oberen Abschnitt beschrieben lässt sich ein Integral

$$I = \int_a^b f(x) dx$$

mit Hilfe einer Monte Carlo Integration

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i, w - \text{verteilt})}{w(x_i, w - \text{verteilt})}$$

approximieren.

Dieses Verfahren soll nun auf die Funktion $f(x) = e^{-30x^2}$ im Intervall $[0 : 1]$ angewendet werden. Die von Mathematica numerisch bestimmte Lösung lautet:

$$I = \int_0^1 e^{-30x^2} dx \approx 0.162$$

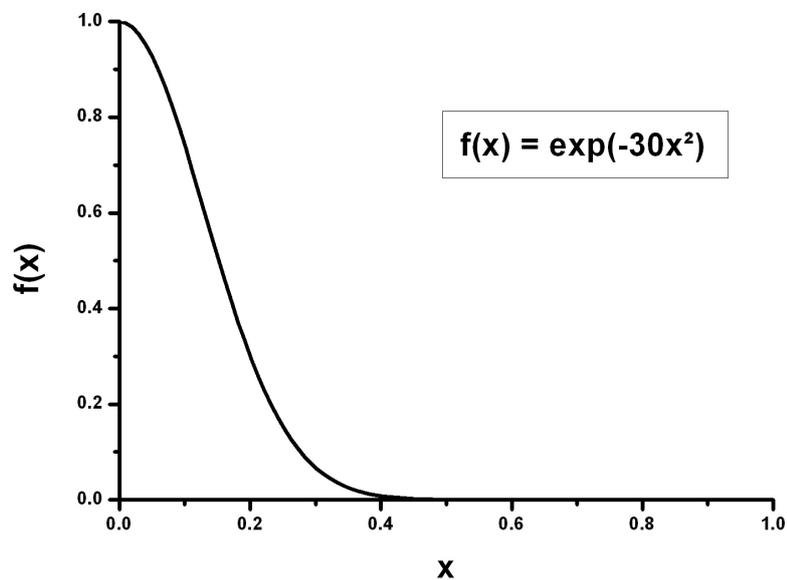


Abbildung 1.1: Verlauf der zu integrierenden Funktion $f(x) = e^{-30x^2}$.

In den Abbildungen (1.2) und (1.3) sind 3 verschiedene MC-Integrationen mit jeweils unterschiedlichen Wahrscheinlichkeitsfunktionen zu finden. Hier ist sehr gut zu erkennen, dass das Verfahren an Geschwindigkeit gewinnt (d.h. weniger MC-Schritte benötigt werden), wenn möglichst viele x_i in Bereichen "gezogen" werden, in denen die Funktion $f(x)$ große Werte annimmt (importance sampling). Es ist allerdings zu beachten, dass nur der Teil des Integrals berechnet wird, in dem $w(x) > 0$ ist. Eine Wahl von $w(x)$ wie in Abbildung (1.4) führt folglich zu einem falschen Ergebnis, da ein wesentlicher Teil des Integrals abgeschnitten wird.

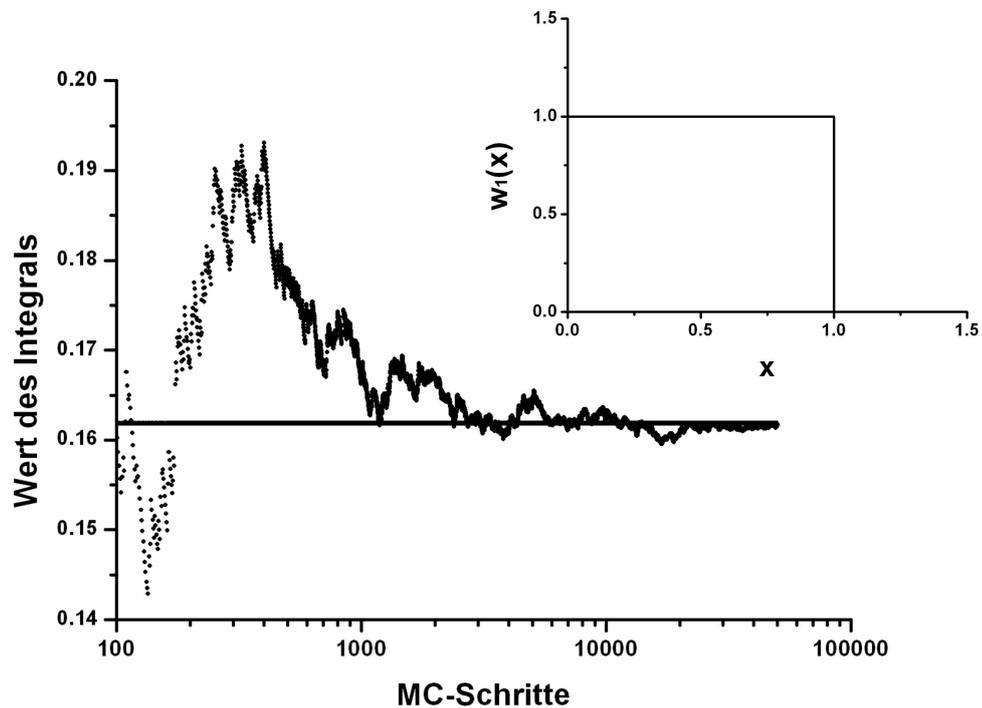


Abbildung 1.2: MC-Integration mit einer gleichverteilen Wahrscheinlichkeitsfunktion $w_1(x)$.

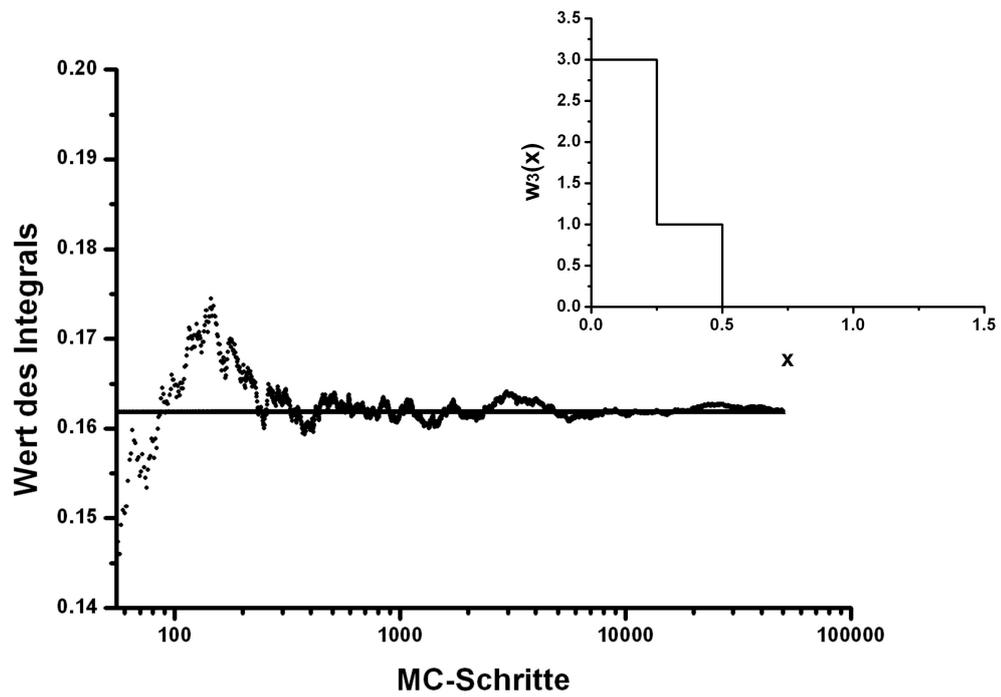
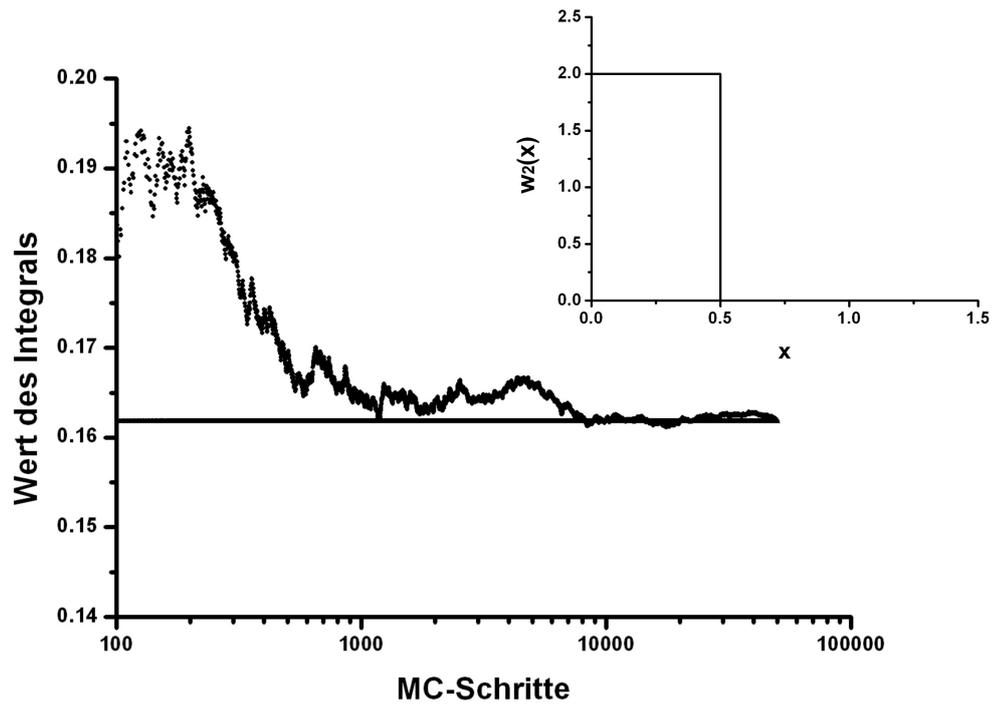


Abbildung 1.3: MC-Integration mit zwei weiteren Wahrscheinlichkeitsfunktion $w_2(x)$ und $w_3(x)$.

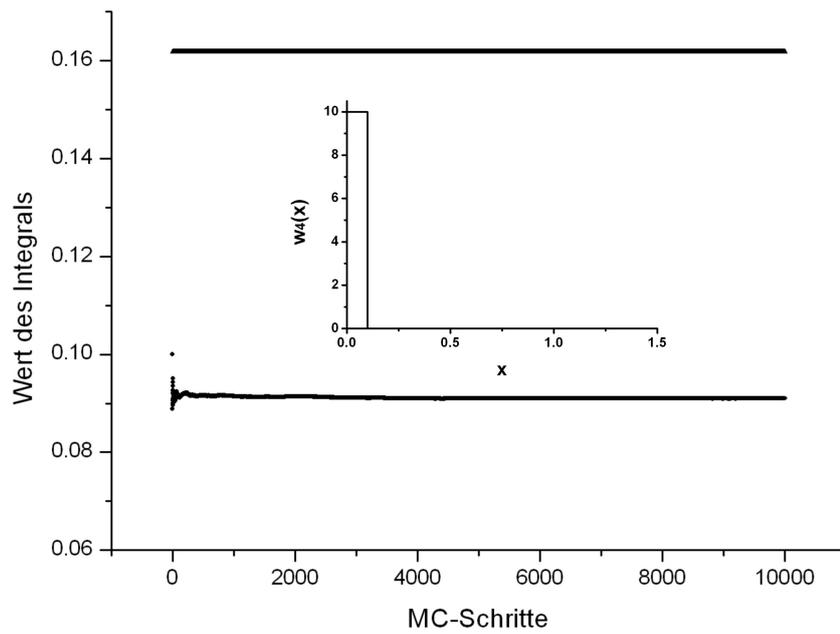


Abbildung 1.4: MC-Integration mit einer ungeeigneten Wahrscheinlichkeitsfunktion $w_4(x)$. Die ungenaue Approximation des Integrals kommt zu stande, da durch die Wahl der Wahrscheinlichkeitsfunktion $w_4(x)$ ein zu großer Teil des Integrals „abgeschnitten“ wird.

2 Metropolisalgorithmus

Ist man an Konfigurationen interessiert, welche unter einer vorgegebenen Verteilung erzeugt werden, bietet es sich an den Metropolisalgorithmus zu verwenden.

Metropolisalgorithmus:

1. Erzeuge zufällig eine Konfiguration f_m
2. Erzeuge zufällig eine weitere Konfiguration f_n
3. Erzeuge eine Zufallszahl $\xi \in [0, 1]$
4. Akzeptiere f_n , wenn $\min\left(1, \frac{f_n}{f_m}\right) \geq \xi$,
wenn nicht nehme wieder f_m
5. Gehe zu Schritt 2

An folgenden Beispielen soll die Funktionsweise des Metropolisalgorithmus verdeutlicht werden.

Beispiel: Ziffernfolge mit vorgegebener Ziffernverteilung

Zunächst soll ein diskreter Konfigurationsraum mit vier möglichen Konfigurationen betrachtet werden. Wir sind an der Erzeugung einer Zahlenkette aus den Ziffern 1-4 unter folgenden Voraussetzungen interessiert:

- Die Kette soll mit einer geraden Ziffer starten.
- Die Differenz zweier aufeinander folgender Ziffern darf nur die Werte (-1,0,1) annehmen.
- Nach der "1" darf die "4" folgen und nach der "4" die "1"
(Periodische Randbedingungen)
- Die geraden Ziffern (2,4) sollen doppelt so häufig auftreten wie die ungeraden Ziffern (1,3) $\left(\frac{f_2}{f_1} = \frac{f_4}{f_1} = \frac{f_2}{f_3} = \frac{f_4}{f_3} = 2 \text{ und } \frac{f_1}{f_3} = \frac{f_2}{f_4} = 1\right)$

Eine mögliche Zahlenkette wäre beispielsweise "2344121432234412221444".

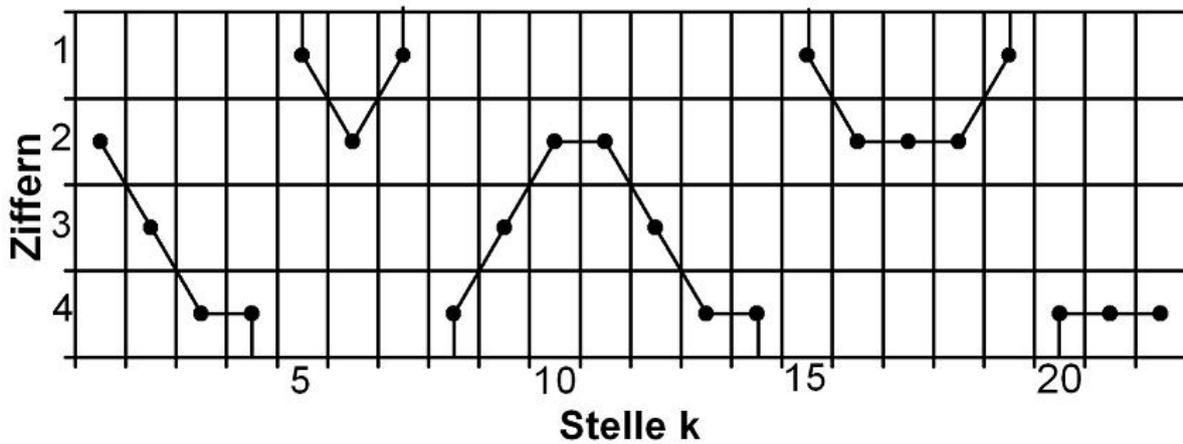


Abbildung 2.1: Möglicher Weg durch den Konfigurationsraum

Metropolisalgorithmus:

1. Erzeuge zufällig eine der Ziffern "1", "2", "3" oder "4".
2. Erzeuge eine neue Ziffer indem du zufällig -1,0,1 auf die alte Ziffer addierst.
Aus einer durch Addition entstandenen "0" mache eine "4" und aus einer durch Addition entstandenen "5" mache eine "1".
3. Erzeuge eine Zufallszahl $\xi \in [0, 1]$
4. Akzeptiere die neue Ziffer, wenn $\min\left(1, \frac{f_n}{f_m}\right) \geq \xi$,
wenn nicht nehme erneut die alte Ziffer.
5. Gehe zu Schritt 2

Um zu verstehen, dass mit dem Metropolisalgorithmus diese Zahlenketten erzeugt werden können, müssen der zweite und vierte Schritt des Algorithmusses näher betrachtet werden.

Schritt 2 sorgt zum Beispiel dafür, dass es gleichwahrscheinlich ist, nach einer "2" eine "1", "3" oder "4" zu erzeugen. Die Wahrscheinlichkeit beträgt jeweils $\frac{1}{3}$.

Schritt 4 sorgt allerdings dafür, dass nicht jede neue Konfiguration akzeptiert wird. Die Wahrscheinlichkeit für das Auftreten einer "3" nach einer "2" beträgt auf Grund des

Akzeptanzkriteriums $\min(1, \frac{1}{2}) \geq \xi$ beispielsweise lediglich $\frac{1}{2}$.

Kombiniert liefert dies folgende Wahrscheinlichkeiten für die Nachfolger einer "2":

$$\begin{aligned}P(2 \rightarrow 1) &= \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6} \\P(2 \rightarrow 2) &= \frac{1}{3} \cdot 2 = \frac{2}{3} \\P(2 \rightarrow 3) &= \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}\end{aligned}$$

Bei $P(2 \rightarrow 2)$ ging zusätzlich noch die Bedingung ein, dass die alte Konfiguration erneut verwendet wird, sollte die neue Konfiguration abgelehnt worden sein.

Schreibt man nun die Wahrscheinlichkeiten aller möglichen Übergänge $P(m \rightarrow n)$ in eine Übergangsmatrix π , wobei in der Komponente π_{ij} die Wahrscheinlichkeit $P(i \rightarrow j)$ mit $i, j \in \{1, 2, 3, 4\}$ steht, ergibt sich:

$$\pi = \begin{pmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 1/6 & 2/3 & 1/6 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/6 & 0 & 1/6 & 2/3 \end{pmatrix}$$

Wählt man nun als Startvektor \vec{v}_1 , dass die erste Ziffer beliebig ist,

$$\vec{v}_1 = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}$$

lassen sich die Wahrscheinlichkeiten für die Ziffer an der k-ten Stelle der Zahlenfolge wie folgt bestimmen:

$$\begin{aligned}\vec{v}_k &= \pi \cdot \vec{v}_{k-1} \\ &= \pi \cdot (\pi \vec{v}_{k-2}) \\ &= \pi \cdot \dots \cdot (\pi \cdot (\pi \cdot \vec{v}_1))\end{aligned}$$

Hierbei zeigt sich schon ab der zwanzigsten Stelle ($k=20$), dass die Ziffern mit der vorgegeben Verteilung auftreten:

$$\vec{v}_{20} = \begin{pmatrix} 1/6 \\ 2/6 \\ 1/6 \\ 2/6 \end{pmatrix}$$

Beispiel: Unabhängigkeit von der Startkonfiguration

Ähnlich wie im vorigen Beispiel soll wieder eine Zahlenkette erzeugt werden diesmal allerdings mit den Ziffern 1-9 und mit einer schwierigeren Verteilung. Die Bedingungen an die Zahlenkette lauten diesmal:

1. Es sollen die Ziffern 1 bis 9 verwendet werden.
2. Die Differenz zweier aufeinander folgender Ziffern darf nur die Werte $(-1,0,1)$ annehmen.
3. Nach der "1" darf die "9" folgen und nach der "9" die "1"
(Periodische Randbedingungen).
4. Die Ziffern sollen mit folgenden Häufigkeiten auftreten :
 $\frac{f_5}{f_4} = \frac{f_5}{f_6} = 2$ und $\frac{f_4}{f_3} = \frac{f_6}{f_7} = 2$ und $\frac{f_3}{f_2} = \frac{f_7}{f_8} = 2$ und $\frac{f_2}{f_1} = \frac{f_8}{f_9} = 2$
(die "5" soll beispielsweise doppelt so häufig auftreten wie die "4" und die "6")

In den Abbildung (2.2) und (2.3) sind die Simulationsergebnisse mit unterschiedlichen Startvektoren zu finden.

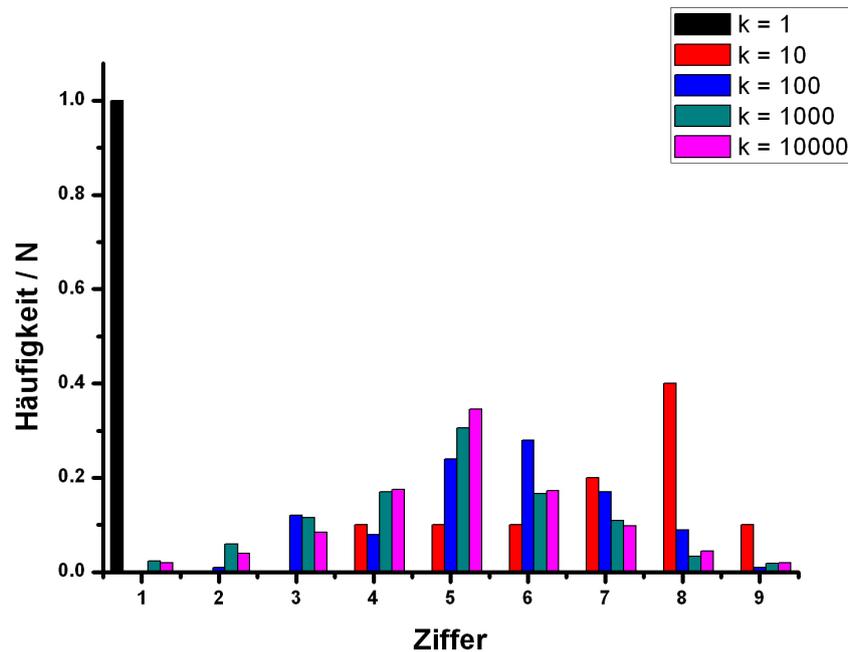


Abbildung 2.2: Simulationsergebnis des Metropolisalgorithmus für die im Text beschriebene Zahlenfolge.

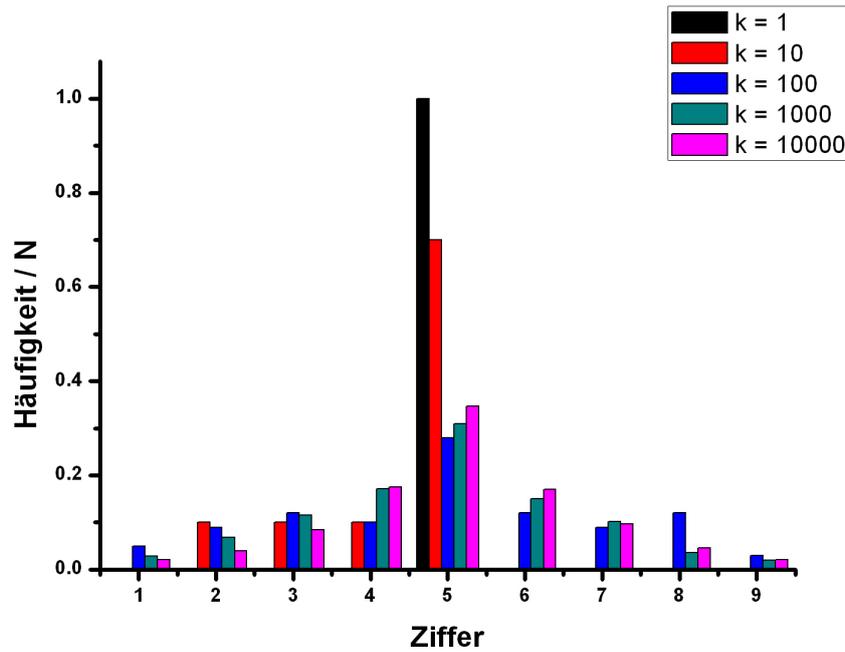


Abbildung 2.3: Simulationsergebnis des Metropolisalgorithmus für die im Text beschriebene Zahlenfolge mit einem anderen Startvektor.

Anmerkungen

Wie in den Abbildungen (2.2), (2.3) zu sehen, ist die Grenzverteilung unabhängig von dem gewählten Startvektor. Dieser hat lediglich Einfluss darauf, wie lange es dauert, bis sich die gewünschte Verteilung einstellt.

Der Metropolisalgorithmus benötigt eine gewisse Zeit, bis er Konfigurationen mit der vorgegebenen Verteilung erzeugt. Diese Zeit wird Thermalisierungszeit genannt. Es ist darauf zu achten, dass erst nach der Thermalisierungszeit an der Simulation Messungen ausgeführt werden.

3 Simulation von Gasen und Flüssigkeiten mit wenigen Teilchen

Ist man daran interessiert ein Teilchengas mit einer typischen Teilchenanzahl in der Größenordnung 10^{23} auf einem Computer zu simulieren, steht man schnell vor einem Problem. Allein der Versuch, jedes Teilchen mit einem Byte auf dem Computer abzuspeichern, würde bedeuten, dass der Computer über 10^{11} Terabyte verfügen müsste -eine Größenordnung, die vernab von heutigen Rechnerleistungen ist. Es muss also ein Weg gefunden werden, dieses quasi unendlich große System auf einem endlichen Computer zu simulieren. Eine gute Möglichkeit hierfür ist der Gebrauch von sogenannten "Periodischen Randbedingungen" (PRB). Es wird davon ausgegangen, dass wenn ein Teilchen das Simulationsvolumen verlässt, sein Spiegelbild an der gegenüberliegenden Seite wieder eintritt (siehe Abbildung (3.1)). Somit ist es möglich ein konstantes Volumen mit konstanter Teilchenzahl ohne Randeffekte zu realisieren.

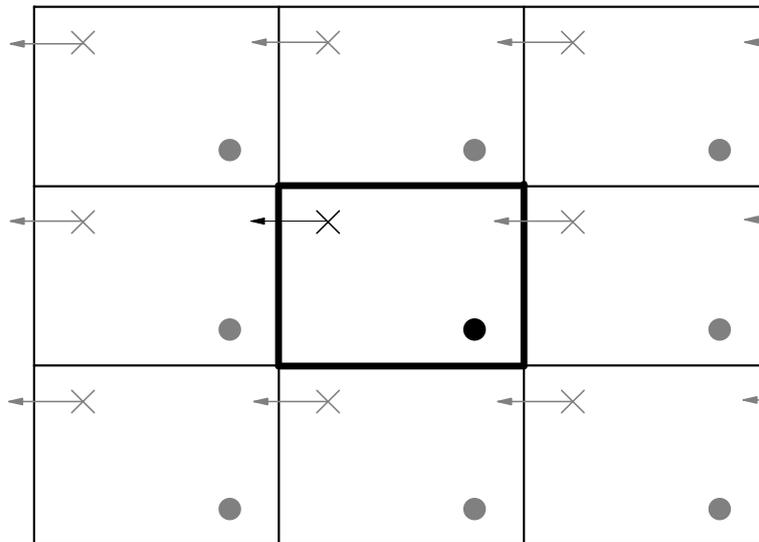


Abbildung 3.1: Simulationsvolumen mit zwei Teilchen umgeben von seinen Bildern. Visualisierung von PRB in zwei Dimensionen.

Im nächsten Schritt muss ausgeschlossen werden, dass ein Teilchen i mit mehreren Bildern von Teilchen j wechselwirkt. Um dies zu gewährleisten führt man die sogenannte "minimum image" Konvention ein. Wie der Name schon verrät, bedeutet dies, dass ein Teilchen i nur mit dem nächsten Bild von Teilchen j wechselwirkt (siehe Abbildung (3.2)).

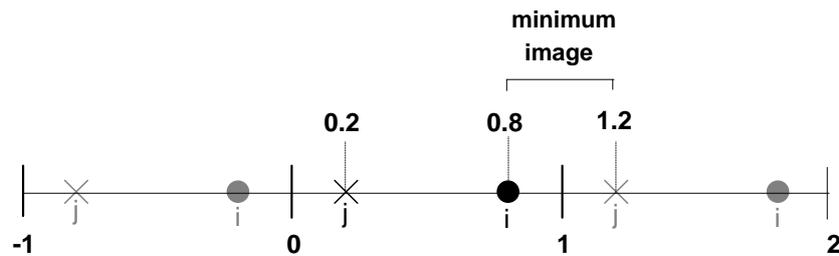


Abbildung 3.2: Visualisierung der "minimum image" Konvention in einer Dimension

Anmerkung:

Der C++ Quellcode könnte lauten:

```
int runden (double a) // Rundet bis 0.5 ab und ab 0.5 auf
{
    a=a+0.5;
    int b= floor(a);
    return b;
}

xij= fabs(i-j); //Abstand zwischen i und j
xij_min = fabs(runden(xij)-xij); // kürzester Abstand von i
// zum Bild von j
```

Als nächstes wird ein Modell benötigt, welches Wechselwirkungen zwischen den Teilchen beschreibt. Bei der Simulation von ungeladen Teilchen wird häufig das Lennard-Jones Potential verwendet:

$$U_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \quad (3.1)$$

Hierbei bezeichnet r_{ij} den Abstand zwischen zwei Teilchen i und j . Bei großen Entfernungen der Teilchen wirkt das LJ-Potential anziehend und ab einem Abstand von σ abstoßend (siehe Abb.(3.3)).

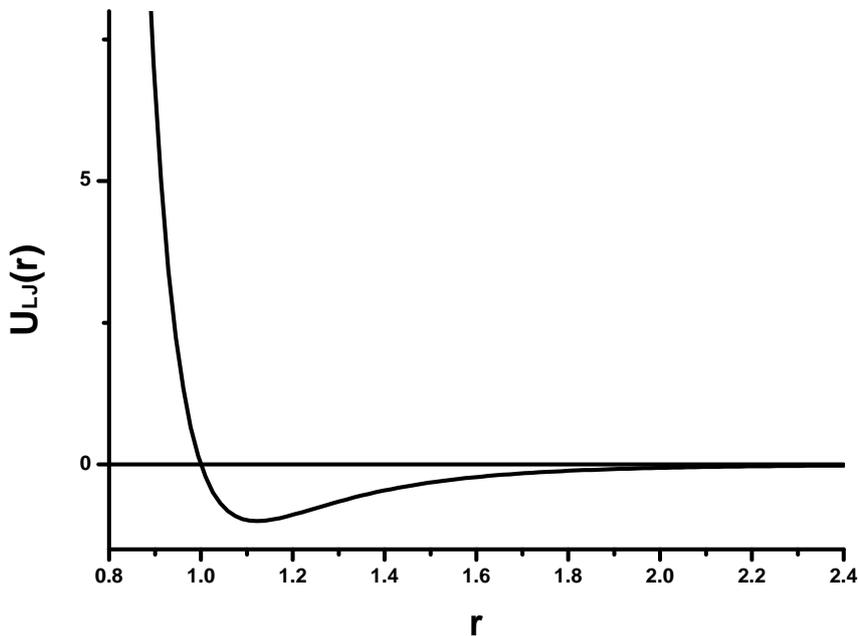


Abbildung 3.3: Lennard Jones Potential für $\epsilon = \sigma = 1$

Da der größte Rechenaufwand in der Berechnung der potentiellen Energien zwischen den Teilchen besteht, wäre die Einführung eines Abschneideradius r_{cut} vorteilhaft, welcher festlegt, bis zu welchem Abstand zwei Teilchen wechselwirken. Dies hätte den Vorteil, nicht mehr "N-1" Wechselwirkungen des Teilchens i mit allen anderen Teilchen berechnen zu müssen, sondern nur die mit $r_{ij} < r_{cut}$. Dies würde eine deutliche Verkürzung der Rechenzeit bedeuten. Die Frage ist nun, wie sich die Einführung von r_{cut} physikalisch begründen lässt. Die naheliegendste Begründung, dass das Lennard-Jones Potential schon ab einem Abstand von einigen σ vernachlässigbar gering wird, mag im ersten Moment richtig erscheinen, beinhaltet allerdings einen Haken. Mit größer werdendem Abstand geht das Lennard-Jones Potential zwar gegen Null, es steigt jedoch auch die Anzahl an

Wechselwirkungspartnern, welche in der Summe wieder einen nicht vernachlässigbaren Anteil beitragen könnten. Es wird also eine Abschätzung benötigt, um sagen zu können, welcher Effekt überwiegt. Dafür soll die mittlere potentielle Energie pro Teilchen bei großen Abständen r_{ij} berechnet werden.

$$\frac{\langle U \rangle}{N} = \frac{1}{N} \left\langle \underbrace{\sum_i u_{LJ}(r_{ij})}_{i < j} \right\rangle \quad (3.2)$$

Diese Gleichung lässt sich aufteilen in einen Anteil mit Teilchen, welche innerhalb der Abschneidekugel sind und einen Anteil mit Teilchen außerhalb dieser.

$$\frac{\langle U \rangle}{N} = \underbrace{\frac{1}{N} \left\langle \sum_{i < j} u_{LJ}(r_{ij}) \right\rangle}_{r_{ij} < r_{cut}}^{(1)} + \underbrace{\frac{1}{N} \left\langle \sum_{i < j} u_{LJ}(r_{ij}) \right\rangle}_{r_{ij} \geq r_{cut}}^{(2)}$$

Es soll nun der Summand (2) betrachtet werden. Hier geht ein, dass für den Mittelwert einer Funktion $k(r_{ij})$ gilt:

$$\langle k(r_{ij}) \rangle = \frac{1}{V} \int d^3 r_{ij} k(r_{ij}) g_2(r_{ij})$$

Wobei es sich bei $g_2(r_{ij})$ um die Paarkorrelationsfunktion handelt. Damit folgt:

$$\underbrace{\frac{1}{N} \left\langle \sum_{i < j} u_{LJ}(r_{ij}) \right\rangle}_{r_{ij} \geq r_{cut}} = \frac{1}{V} \int_{r_{ij} \geq r_{cut}} d^3 r_{ij} \frac{1}{N} \sum_{i < j} u_{LJ}(r_{ij}) g_2(r_{ij})$$

Vertauscht man nun das Integral und die Summe und beachtet, dass für die Anzahl an Wechselwirkungspaaren $\frac{N(N-1)}{2} \approx \frac{N^2}{2}$ gilt, folgt:

$$\underbrace{\frac{1}{N} \left\langle \sum_{i < j} u_{LJ}(r_{ij}) \right\rangle}_{r_{ij} \geq r_{cut}} \approx \frac{\rho}{2} \int_{r_{ij} \geq r_{cut}} d^3 r_{ij} u_{LJ}(r_{ij}) g_2(r_{ij})$$

Da wir den Fall für große r_{ij} betrachten, gilt $g_2(r_{ij}) = 1$. Außerdem ist u_{LJ} rotations-symmetrisch und durch Verwendung von Kugelkoordinaten ergibt sich:

$$\underbrace{\frac{1}{N} \left\langle \sum_{i < j} u_{LJ}(r_{ij}) \right\rangle}_{r_{ij} \geq r_{cut}} \approx \frac{4\pi}{2} \rho \int_{r_{cut}}^{\infty} dr_{ij} \cdot u_{LJ}(r_{ij}) r_{ij}^2$$

Setzen wir nun das oben beschriebene Lennard-Jones Potential ein und berechnen das Integral, so ergibt sich:

$$\frac{1}{N} \left\langle \underbrace{\sum_{i < j} u_{LJ}(r_{ij})}_{r_{ij} >= r_{cut}} \right\rangle = \frac{8\pi\epsilon\sigma^6\rho}{3} \left[\frac{\sigma^6}{3r_{cut}^9} - \frac{1}{r_{cut}^3} \right]$$

Somit wird klar, dass Teilchen außerhalb von r_{cut} für einen konstanten Korrekturterm sorgen. Dies legitimiert die Einführung eines Abschneideradius.

Abschließend ist es nun möglich periodische Randbedingungen, "minimum image" Konvention und Abschneideradius zu kombinieren (siehe Abbildung (3.4)). Die Wechselwirkung von Teilchen i mit den Bildern von Teilchen j wird nur dann berechnet, wenn der kürzeste Abstand zwischen den Teilchen kleiner als r_{cut} ist.

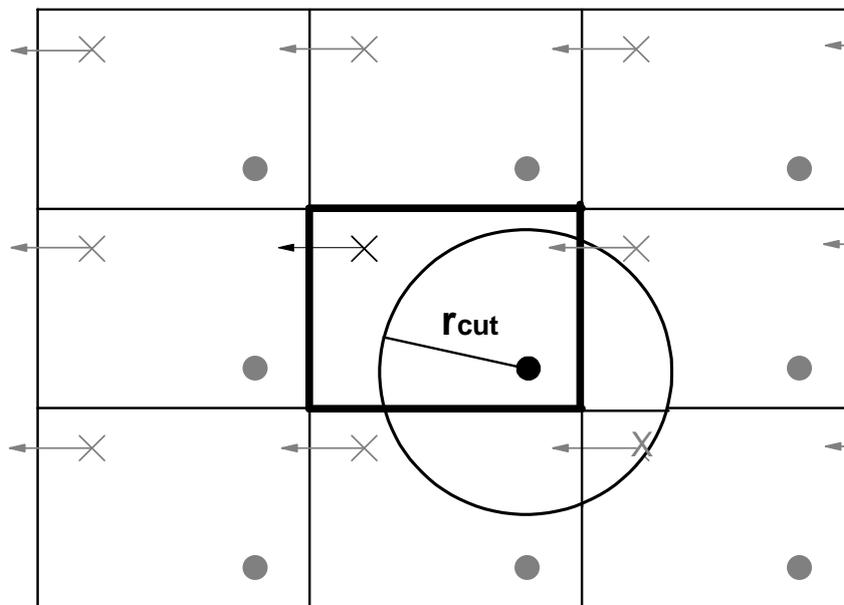


Abbildung 3.4: Kombination von Periodischen Randbedingungen, "minimum image" Konvention und Abschneideradius

4 MC-Simulation eines Lennard-Jones Fluids bestehend aus Punktteilchen

Im folgenden Kapitel geht es um die Simulation eines Lennard-Jones Fluids in verschiedenen Ensembles. Zunächst soll der einfachste Fall eines kanonischen Ensembles (NVT-Ensemble) betrachtet werden.

4.1 NVT-Ensemble

Als kanonisches bzw. NVT-Ensemble wird ein thermodynamisches System mit konstanter Teilchenanzahl N , konstantem Volumen V und konstanter Temperatur T bezeichnet. Die klassische Zustandssumme für "Punkt-" Teilchen lautet:

$$Z_{kanonisch} = \int d^{3N} p \cdot d^{3N} r \cdot e^{-\beta H} \quad (4.1)$$

Für die Wahrscheinlichkeitsdichtefunktion ergibt sich dann:

$$f(\{\vec{r}\}, \{\vec{p}\}) = \frac{e^{-\beta H(\{\vec{r}\}, \{\vec{p}\})}}{\int d^{3N} p \cdot d^{3N} r \cdot e^{-\beta H(\{\vec{r}\}, \{\vec{p}\})}} \quad (4.2)$$

Die Schreibweise d^{3N} bedeutet, dass über die 3 Impulskomponenten der N Teilchen integriert werden soll. $H(\{\vec{r}\}, \{\vec{p}\}) = K(\{\vec{p}\}) + U(\{\vec{r}\})$ ist die Summe aus kinetischer Energie K und potentieller Energie U und wird als Hamiltonfunktion bezeichnet. Da K nur von den Impulsen abhängt und U von den Ortskoordinaten, lässt sich die Gleichung (4.2) in ein Produkt zerlegen.

$$f(\{\vec{r}\}, \{\vec{p}\}) = \frac{e^{-\beta K(\{\vec{p}\})}}{\int d^{3N} p \cdot e^{-\beta K(\{\vec{p}\})}} \cdot \frac{e^{-\beta U(\{\vec{r}\})}}{\int d^{3N} r \cdot e^{-\beta U(\{\vec{r}\})}} \quad (4.3)$$

$$= f_r(\{\vec{r}\}) \cdot f_p(\{\vec{p}\}) \quad (4.4)$$

Soll nun der Erwartungswert einer Observablen $A(\{\vec{r}\})$, welche nur von den Ortskoordinaten abhängt, berechnet werden

$$\langle A(\{\vec{r}\}) \rangle = \int d^{3N} p \cdot d^{3N} r \cdot A(\{\vec{r}\}) \cdot f(\{\vec{r}\}, \{\vec{p}\}) \quad (4.5)$$

lässt sich dies mit Hilfe von Gleichung (4.4), der Bedingung $\int d^{3N} p \cdot f_p(\{\vec{p}\}) = 1$ und dem Wissen über MC-Integration vereinfachen:

$$\langle A(\{\vec{r}\}) \rangle = \int d^{3N} r \cdot A(\{\vec{r}\}) \cdot f_r(\{\vec{r}\}) \approx \frac{1}{M} \sum_{i=1}^M A(\{\vec{r}\}_i) \quad (4.6)$$

Hier steht $\{\vec{r}\}_i$ für die i -te Konfiguration aus einer Abfolge von Konfigurationen, die entsprechend $f_r(\{\vec{r}\})$ gewichtet sind. Wie diese erzeugt werden, hatten wir anhand des Beispiels SZiffernfolge mit vorgegebener Ziffernverteilung in Kapitel zwei gesehen. Im vorliegenden Fall ist $\frac{f_n}{f_m}$ durch

$$\frac{f_n}{f_m} = \frac{f(\{\vec{r}_n\})}{f(\{\vec{r}_m\})} = e^{-\beta[U(\{\vec{r}_n\}) - U(\{\vec{r}_m\})]} \quad (4.7)$$

gegeben. Der Zustand n tritt also um den Faktor $\frac{f_n}{f_m}$ häufiger auf, als der Zustand m . $e^{-\beta U}$ wird dabei als Boltzmanfaktor bezeichnet. Man beachte, dass sich der komplizierte Nenner, also $\int d^{3N} r \cdot e^{-\beta u(\{\vec{r}\})}$ herausgekürzt hat. Ein entsprechender Algorithmus lautet:

Metropolisalgorithmus im NVT-Ensemble:

1. Erzeuge zufällig eine Konfiguration f_m .
2. Erzeuge zufällig eine weitere Konfiguration f_n .
3. Erzeuge eine Zufallszahl $\xi \in [0, 1]$.
4. Akzeptiere f_n , wenn $\min\left(1, \frac{f_n}{f_m}\right) \geq \xi$ mit $m \neq n$, wenn nicht, nehme wieder f_m .
5. Gehe zu Schritt 2.

Anmerkung:

Dies bedeutet unter anderem, dass eine neue Konfiguration n immer dann akzeptiert wird, wenn n eine niedrigere potentielle Energie besitzt als m .

Beim Metropolisalgorithmus gilt es, zwei Voraussetzungen zu beachten. Zum einen müssen alle möglichen Konfigurationen des Systems erzeugt werden können (Ergodizität), zum anderen muss der Übergang von m zu n mit gleicher Wahrscheinlichkeit erzeugt werden wie der Übergang von n zu m ("detailed balance" Bedingung).

Beispiel: Monte Carlo Simulation eines NVT-Ensembles

Es soll eine NVT-Simulation in drei Dimensionen erstellt werden, welche es ermöglicht, den Erwartungswert der potentiellen Energie $\langle U_{pot} \rangle$ zu bestimmen.

Hauptprogramm:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<time.h>

#define L 8      //Volumengröße
#define N 200   //Teilchenanzahl
#define n 100000 // MC-Schritte

typedef struct{
    double x;
    double y; // Schwerpunkt in Kartesischen Koordinaten
    double z;
} Teilchen;

//globale Variablen
Teilchen liste[N]={0};
Teilchen liste_move[N]={0};
double U_matrix [N] [N]={0};
double U_matrix_neu [N] [N]={0};
double Upot_alt;
double Upot_neu;

int main(void)
{ int i,j,k,z;
  startkonfi(liste,list_move); //Lässt die Teilchen auf einem
  Upot_alt=upot_start();      Gitter starten

  for(k=1;k<=n;k++)          // Hauptschleife
  {
    z=rand()%N;               // wählt eine Zufallszahl zwischen 0 und N
    move(liste_move,z);       // Bewege das zufällig gewählte Teilchen
    Upot_neu=upot_neu(z);     // Bestimme neue potentielle Energie
    metropolis();             // Metropolis-kriterium
  }
  return 0;}
```

Verwendete Funktionen:

Es bietet sich an, die Teilchen auf einem Gitter starten zu lassen. Dies hat im Gegensatz zu einer zufälligen Wahl der Startkonfiguration den Vorteil, dass die Teilchen sich nicht „nahe“ kommen und die potentielle Energie nicht divergiert. Hierbei ist darauf zu achten, dass das Gitter nicht die Ränder der Box berührt, da es sonst zu Schwierigkeiten mit den periodischen Randbedingungen kommt.

```
void startkonfi (Teilchen feld[],Teilchen feld_move[])
{
    double i,j,k;
    int counter=0;
    double ende = pow(N,(double)1/3);
    double max=(double)(L-2)/(ende);
    for(i=0;i<=L-2;i=i+max)
    {
        for(j=0;j<=L-2;j=j+max)
        {
            for (k=0;k<=L-2;k=k+max)
            {
                if(counter<=N-1)
                {feld[counter].x=1+k;
                 feld[counter].y=1+j;
                 feld[counter].z=1+i;
                 feld_move[counter].x=feld[counter].x;
                 feld_move[counter].y=feld[counter].y;
                 feld_move[counter].z=feld[counter].z;
                 counter=counter+1;
                }
            }
        }
    }
}
```

Das Verschieben eines Teilchens mit periodischen Randbedingungen in drei Dimensionen könnte so aussehen:

```
void move (Teilchen feld[],int k)
{
    feld[k].x=feld[k].x+(1-(double)2*rand()/RAND_MAX)*r_max;
    if(feld[k].x>L) feld[k].x=feld[k].x-L;
    if(feld[k].x<=0) feld[k].x=feld[k].x+L;
```

```

    feld[k].y=feld[k].y+(1-(double)2*rand()/RAND_MAX)*r_max;
    if(feld[k].y>L) feld[k].y=feld[k].y-L;
    if(feld[k].y<=0) feld[k].y=feld[k].y+L;

    feld[k].z=feld[k].z+(1-(double)2*rand()/RAND_MAX)*r_max;
    if(feld[k].z>L) feld[k].z=feld[k].z-L;
    if(feld[k].z<=0) feld[k].z=feld[k].z+L;
}

```

Die Berechnung der potentiellen Energie zwischen zwei Teilchen unter Verwendung der "minimum image" Konvention und des Abschneideradius lautet:

```

double u_lj (Teilchen i,Teilchen j)
{
    double u_lj;
    // Bestimme kleinsten Abstand rij_min zwischen i und j
    // bzw. i und den Bildern von j
    double xij= fabs(i.x-j.x);
    double xij_min = fabs(L*runden(xij/L)-xij);

    double yij = fabs(i.y-j.y);
    double yij_min=fabs(L*runden(yij/L)-yij);

    double zij=fabs(i.z-j.z);
    double zij_min=fabs(L*runden(zij/L)-zij);

    // r Betrag:
    double r_ij=sqrt(pow(xij_min,2)+pow(yij_min,2)+pow(zij_min,2));

    if (r_ij<=r_cut) // Abfrage ob r-Betrag kleiner r_cut ist.
    {
        u_lj=4*epsilon*(pow((sigma/r_ij),12)-pow((sigma/r_ij),6));
        return u_lj;
    }
    else
        return 0;
}

```

Für das Metropolis-kriterium ergibt sich:

```
void metropolis (void)
{
  double e_hoch=exp(-(Upot_neu-Upot_alt)/(k_b*T)); // Metropolis-kriterium

  if(min(1,e_hoch)>= (double)rand()/RAND_MAX)
  { Upot_alt=Upot_neu;
    gleich_setzen(liste,list_move);
    Energiematrix_gleich(U_matrix,U_matrix_neu);
  }
  else
  {Upot_alt=Upot_alt;
    gleich_setzen(liste_move,list);
    Energiematrix_gleich(U_matrix_neu,U_matrix);
  }
}
```

Anmerkung 4.2:

Der größte Rechenaufwand der Simulation liegt in der Berechnung der potentiellen Energien zwischen den einzelnen Teilchen. Wie oben beschrieben, wird bei jedem MC-Schritt eine neue Konfiguration erzeugt, indem ein Teilchen zufällig ausgewählt und verschoben wird. Da es sich als sehr rechenaufwendig erweist, bei jedem MC-Schritt alle potentiellen Energien neu zu berechnen, bietet sich folgendes Verfahren an.

- Berechne am Anfang der Simulation die potentiellen Energien zwischen allen Teilchen und speichere sie in einer $(N \times N)$ -Matrix ab. Hierbei steht in der Komponente U_{ij} die potentielle Energie zwischen Teilchen i und j .
- Wähle zufällig ein Teilchen i .
- Lösche die i -te Zeile und Spalte.
- Verschiebe das i -te Teilchen zufällig.
- Fülle die i -te Zeile und Spalte mit den für Teilchen i neu berechneten potentiellen Energien.
- Erhalte die gesammte potentielle Energie durch Aufsummieren der Komponenten der oberen Dreiecksmatrix ohne Diagonale.

Ein Beispiel für 4 Teilchen, bei dem Teilchen "3" zufällig ausgewählt wird, verdeutlicht das Verfahren:

$$\begin{pmatrix} U_{11} & U_{12} & U_{13} & U_{14} \\ U_{21} & U_{22} & U_{23} & U_{24} \\ U_{31} & U_{32} & U_{33} & U_{34} \\ U_{41} & U_{42} & U_{43} & U_{44} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & 0 & U_{14} \\ U_{21} & U_{22} & 0 & U_{24} \\ 0 & 0 & 0 & 0 \\ U_{41} & U_{42} & 0 & U_{44} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \tilde{U}_{13} & U_{14} \\ U_{21} & U_{22} & \tilde{U}_{23} & U_{24} \\ \tilde{U}_{31} & \tilde{U}_{32} & \tilde{U}_{33} & \tilde{U}_{34} \\ U_{41} & U_{42} & \tilde{U}_{43} & U_{44} \end{pmatrix}$$

$$\Rightarrow U_{gesamt}^{neu} = U_{12} + \tilde{U}_{13} + U_{14} + \tilde{U}_{23} + U_{24} + \tilde{U}_{34}$$

In Abbildung (4.1) sind die Simulationsergebnisse für ein mit obigem Quellcode simuliertes NVT-Ensemble. Die Energiehistogramme für verschiedene Teilchenanzahlen sind in Abbildung (4.2) zu finden. Es ist gut zu erkennen, dass das Verteilungsmaximum mit steigender Teilchenanzahl schmaler und höher wird. Dies legt den Gedanken nahe, im thermodynamischen Limes einen Deltapeak zu erwarten. Passend dazu ist in Abbildung (4.3) das Verhalten des Erwartungswertes der Energie bei Erhöhung der Teilchenanzahl und gleichzeitig konstanter Teilchendichte zu finden. Im thermodynamischen Limes ist ein $\frac{1}{N}$ Verhalten beobachtbar.

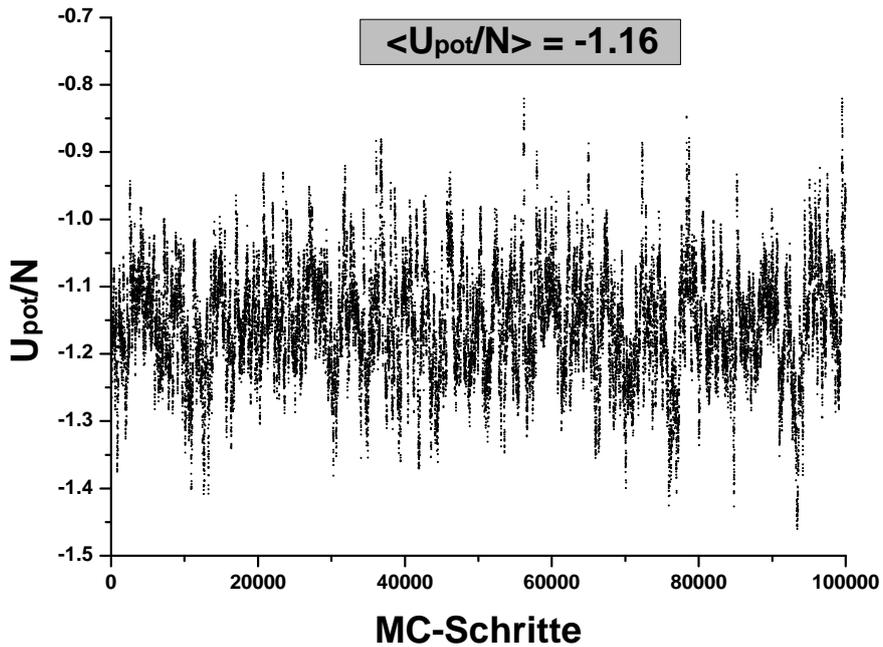


Abbildung 4.1: $\frac{U_{pot}}{N}$ als Funktion der MC-Schritte für $N=100$, $L=8$, $T=2$ und $r_{cut}=2.5$.

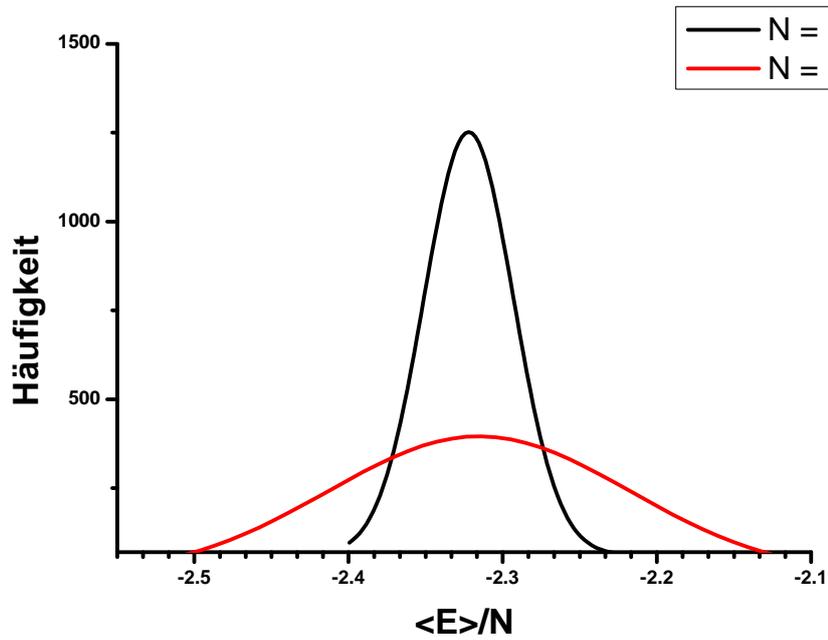


Abbildung 4.2: Energiehistogramme für verschiedene Teilchenanzahlen ($N = 100$ und $N = 1000$).

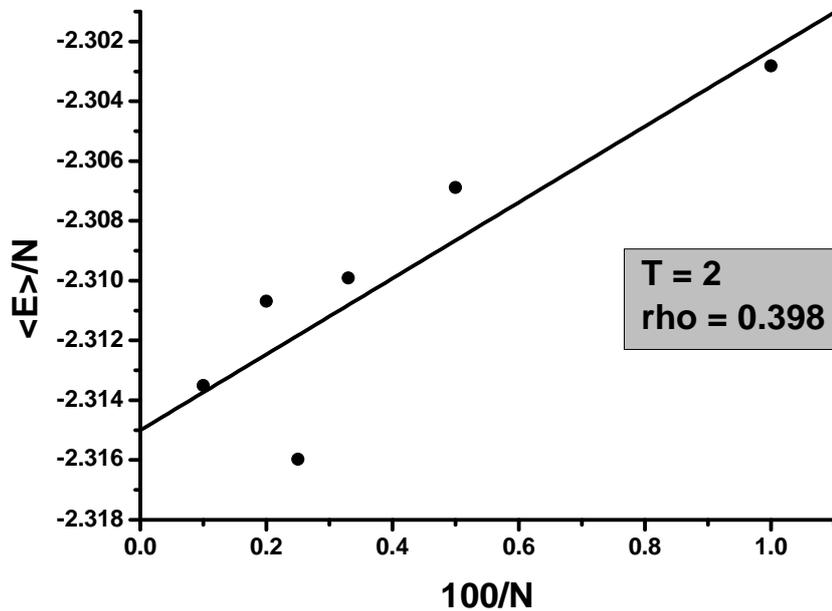


Abbildung 4.3: Verhalten von $\frac{\langle E \rangle}{N}$ im themodynamischen Limes.

4.2 NPT-Ensemble

Im vorigen Abschnitt haben wir gesehen, wie mit Hilfe des Metropolis Algorithmus ein NVT-Ensemble simuliert werden kann. Nun wollen wir uns einem System mit konstanter Teilchenanzahl, konstantem Druck und konstanter Temperatur (NPT-Ensemble) widmen. In diesem Ensemble wird das Volumen V als zusätzlicher Konfigurationsfreiheitsgrad behandelt. Hier ist es wichtig, zunächst zu verstehen, dass die potentielle Energie im Falle periodischer Randbedingungen nicht mehr nur von den Orten der Teilchen, sondern zusätzlich noch von der Größe des Simulationsvolumens abhängt. Dies wird in Abbildung (4.4) verdeutlicht.

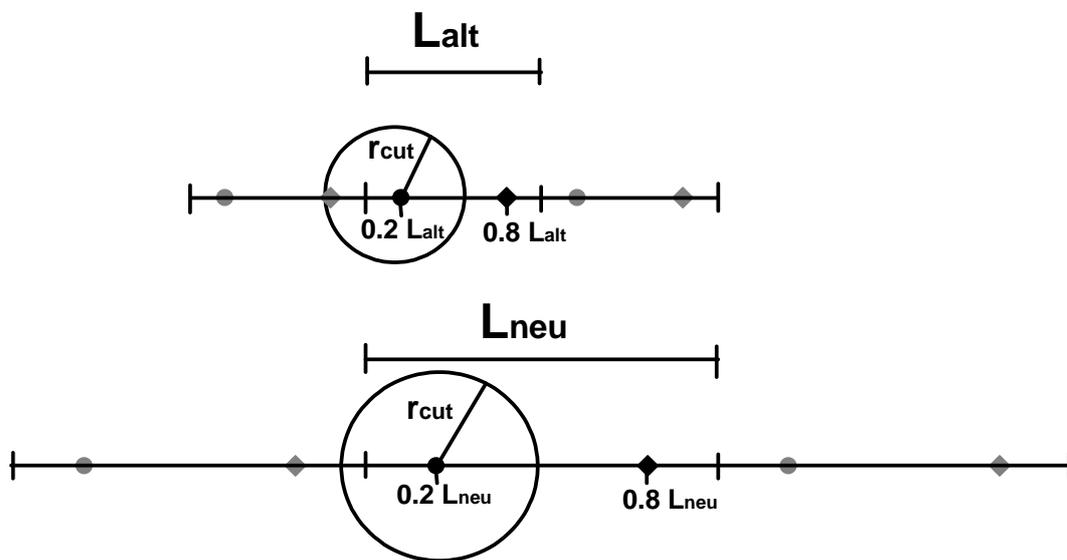


Abbildung 4.4: Visualisierung der Tatsache, dass die potentielle Energie zusätzlich eine Funktion von V ist $\Rightarrow U = U(\{\vec{s}\}, V)$

Desweiteren muss die Auswirkung einer Volumensänderung auf den Erwartungswert einer Observablen untersucht werden. Dafür betrachten wir den Erwartungswert einer Observablen A , welche nur von den Orten abhängt.

$$\langle A(\{\vec{r}\}) \rangle = \int d^{3N} p \int_V d^{3N} r \cdot A(\{\vec{r}\}) \cdot f(\{\vec{r}\}, \{\vec{p}\}) \quad (4.8)$$

Hier sehen wir, dass sich bei einer Volumensänderung das Integrationsvolumen V ändert. Soll nun später das Metropoliskriterium $\frac{f_m}{f_n}$ angewendet werden, muss vorher sicher gestellt sein, dass über dasselbe Integrationsvolumen V integriert wurde.

Dies wird durch die Einführung von dimensionslosen Koordinaten \vec{s} erreicht:

$$s_x = \frac{r_x}{L_x}$$

$$s_y = \frac{r_y}{L_y}$$

$$s_z = \frac{r_z}{L_z}$$

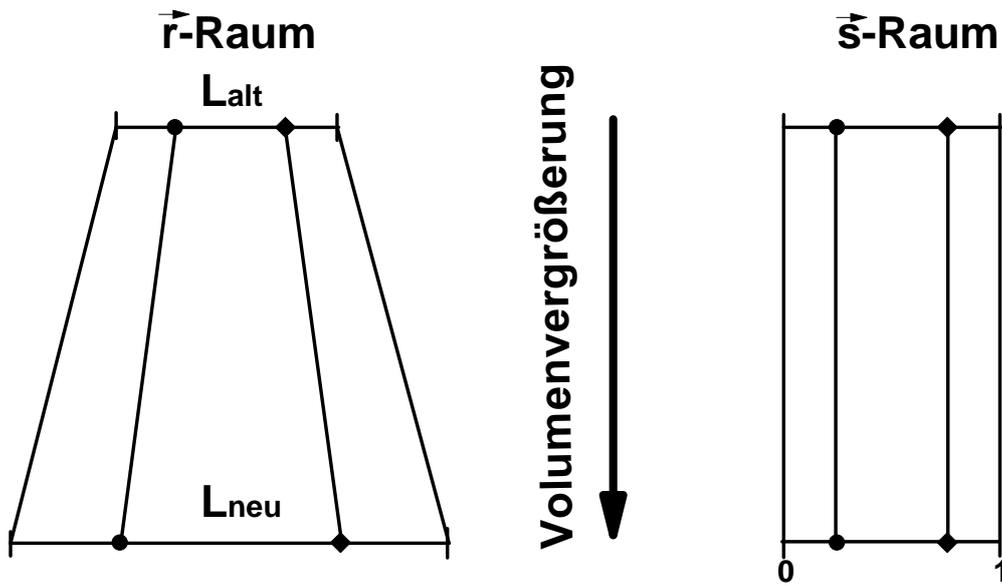


Abbildung 4.5: Volumensvergrößerung im \vec{r} - und \vec{s} -Raum

Somit ist sichergestellt, dass das Simulationsvolumen in \vec{s} -Koordinaten immer die Länge $L = 1$ hat. Dies muss allerdings bei allen Zusammenhängen, welche mit Längen zu tun haben, beispielsweise den Periodischen Randbedingungen oder dem Abschneideradius, berücksichtigt werden.

Für die Wahrscheinlichkeitsfunktion im NPT-Ensemble gilt dann:

$$f_{NPT}(\{\vec{s}\}, V) \propto V^N \cdot e^{-\beta[U(\{\vec{s}\}, V) + PV]} \quad (4.9)$$

Wobei der Faktor V^N aus der Substitution von \vec{r} durch \vec{s} stammt.

Der Metropolis Algorithmus für ein kubisches Simulationsvolumen ($L_x = L_y = L_z = L$) im NPT-Ensemble lautet:

Metropolisalgorithmus im NPT-Ensemble:

1. Erzeuge zufällig eine Konfiguration f_m .
2. Erzeuge zufällig eine weitere Konfiguration f_n , indem du entweder
 - (a) zufällig ein Teilchen verschiebst oder
 - (b) zufällig das Simulationsvolumen V veränderst
3. Erzeuge eine Zufallszahl $\xi \in [0, 1]$.
4. Akzeptiere f_n , wenn

$$\min\left(1, \exp\left\{-\beta [U^{neu} - U^{alt} + P(V^{neu} - V^{alt})] + N \ln\left(\frac{V^{neu}}{V^{alt}}\right)\right\}\right) \geq \xi,$$
 wenn nicht nehme wieder f_m .
5. Gehe zu Schritt 2.

Der Fall (a) ist identisch zu behandeln wie im NVT-Ensemble. Daher widmen wir uns jetzt Fall (b), bei welchem das Volumen zufällig geändert und die neue potentielle Energie berechnet wird. Die zufällige Volumensänderung kann wie folgt realisiert werden:

$$V^{neu} = V^{alt} + (2\xi - 1)\delta V_{max} \quad (4.10)$$

Hierbei ist ξ wieder eine gleichverteilte Zufallszahl zwischen 0 und 1 und V_{max} die maximale Volumensänderung. Der Abstand zwischen zwei Teilchen lässt sich in den \vec{s} -Koordinaten wie folgt ausdrücken:

$$r_{ij} = |\vec{r}_i - \vec{r}_j| = L |\vec{s}_i - \vec{s}_j| = s_{ij}L \quad (4.11)$$

Bei der Berechnung der neuen potentiellen Energie nach einer Volumensänderung zeigt sich nun ein weiterer Vorteil der \vec{s} -Koordinaten.

$$U^{(neu)} = \sum_{i<j} \left[4\epsilon \left(\frac{\sigma}{r_{ij}^{(neu)}} \right)^{12} - 4\epsilon \left(\frac{\sigma}{r_{ij}^{(neu)}} \right)^6 \right] \quad (4.12)$$

$$= \sum_{i<j} \left[4\epsilon \left(\frac{\sigma}{s_{ij}L^{(neu)}} \right)^{12} - 4\epsilon \left(\frac{\sigma}{s_{ij}L^{(neu)}} \right)^6 \right] \quad (4.13)$$

$$= 4\epsilon \left[\sum_{i<j} \left(\frac{\sigma}{s_{ij}} \right)^{12} \right] \left(\frac{1}{L^{(neu)}} \right)^{12} - 4\epsilon \left[\sum_{i<j} \left(\frac{\sigma}{s_{ij}} \right)^6 \right] \left(\frac{1}{L^{(neu)}} \right)^6 \quad (4.14)$$

Die potentielle Energie der neuen Konfiguration setzt sich also aus den einzelnen Anteilen der alten Konfigurationen zusammen, die nur unterschiedlich skaliert werden müssen. Dies bedeutet, dass auf die rechenintensive Summation über alle Wechselwirkungspartner verzichtet werden kann.

Beispiel: Monte Carlo NPT-Simulation für Methan

Mit den oben eingeführten Methoden wurde eine NPT-Simulation für ein Lennard-Jones-System mit 108 Teilchen erstellt. Es soll die Temperaturabhängigkeit der Dichte von Methan bei einem konstanten Druck von 10MPa ($P=0.2573$) untersucht werden. Die Simulationswerte werden mit experimentellen Daten verglichen. Dafür wurden die experimentellen Daten mit den Parametern $\epsilon = 141.185\text{K}$ und $\sigma = 3.688$ umgerechnet. In Abbildung (4.6) sind die Daten der Simulation und des Experiments zu finden. Es wurde $s_{cut} = 0.4$ gewählt, was wiederum bedeutet, dass sich r_{rut} mit dem Simulationsvolumen ändert. Daher ist es wichtig, die Langreichweitenkorrekturen des Lennard-Jones Potentials zu berücksichtigen und bei jeder Volumensänderung neu zu berechnen.

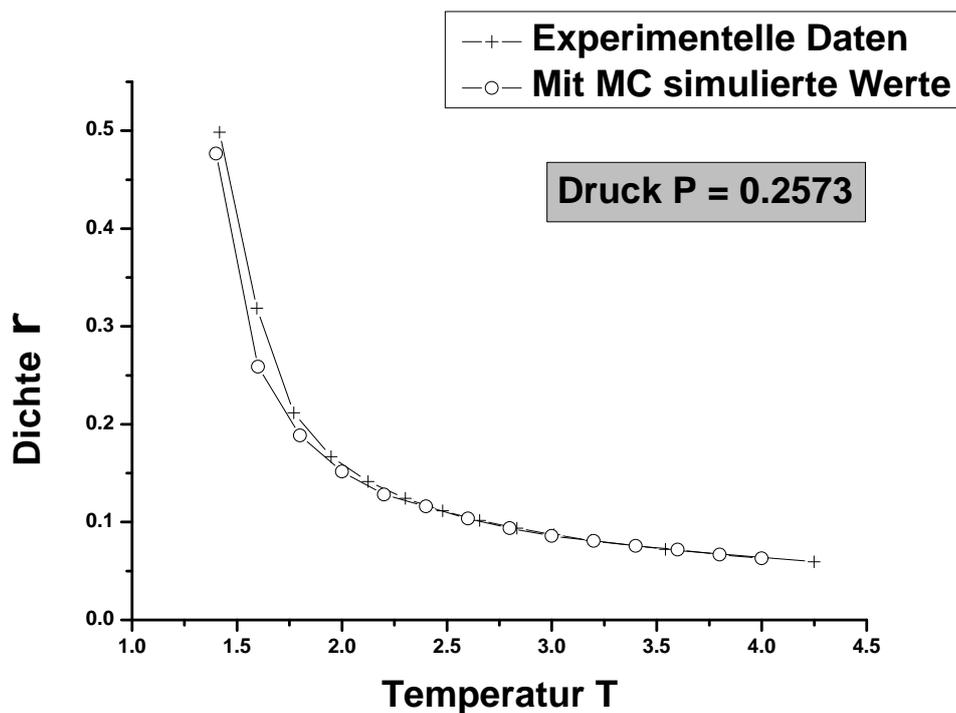


Abbildung 4.6: Temperaturabhängigkeit der Dichte von Methan.

Anmerkung:

Möchte man eine thermodynamische Größe gegen eine andere auftragen, ist es unpraktisch für jeden Simulationswert eine neue Simulation zu starten. Eleganter und effektiver ist es, die vorige Konfiguration als Startkonfiguration für den nächsten Messpunkt zu nehmen. Dieses Verfahren wurde bei der Simulation in Abbildung (4.6) verwendet. Hier wurde bei einer Temperatur von $T = 4$ gestartet und bis $T = 1.4$ Schritt für Schritt simuliert. Dennoch ist es wichtig, bei jedem Temperaturschritt darauf zu achten, dass sich das System wieder im Gleichgewicht befindet, bevor mit der neuen Messung begonnen wird. Diese Thermalisierungszeit ist je nach Konfiguration verschieden.

Beispiel: Monte Carlo NPT-Simulation für Argon

Mit der obigen NPT-Simulation soll diesmal die Temperaturabhängigkeit der Enthalpie H von Argon bei einem konstanten Druck von 10MPa ($P = 0.2478$) bestimmt werden. Die Simulationswerte werden mit experimentellen Daten verglichen. Dafür wurden die experimentellen Daten mit den Parametern $\epsilon = 111.756\text{K}$ und $\sigma = 3.369$ umgerechnet. Aus der Simulation wurden die innere Energie E und die Dichte ρ entnommen. Mit folgender Relation lässt sich dann die Enthalpie bestimmen:

$$\frac{H}{N} = \frac{E}{N} + \frac{P}{\rho}$$

Außerdem wurde mit Hilfe der Enthalpiewerte die Wärmekapazität $\frac{c_p}{R}$ als Temperaturableitung bestimmt:

$$\frac{c_p}{R} = \frac{1}{N} \left(\frac{dH}{dT} \right)_p$$

In Abbildung (4.7) sind die Daten der Simulation und des Experiments zu finden.

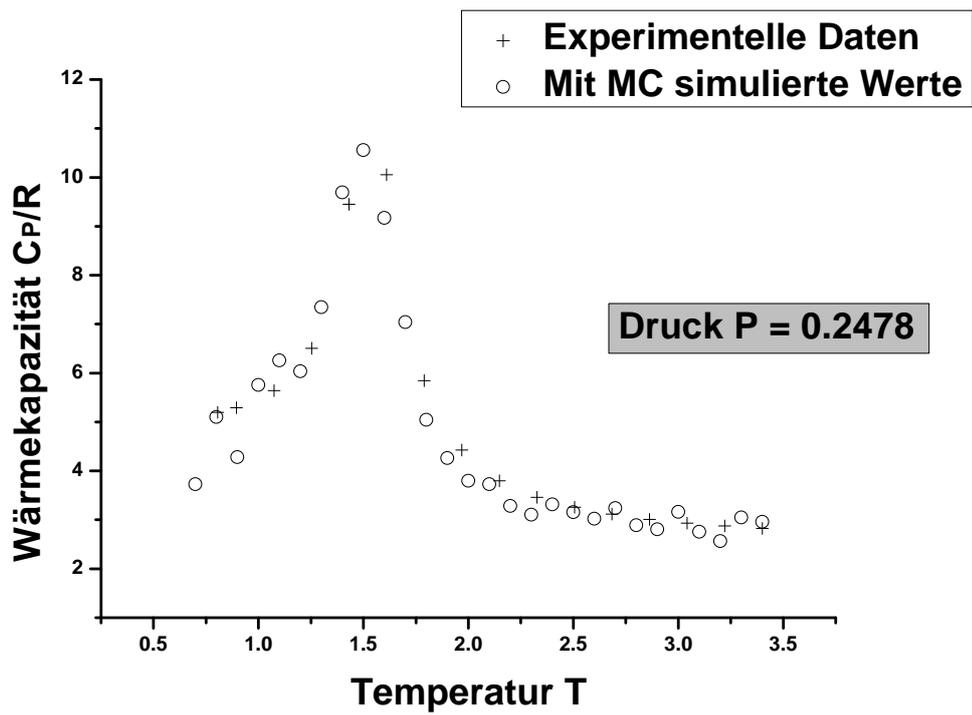
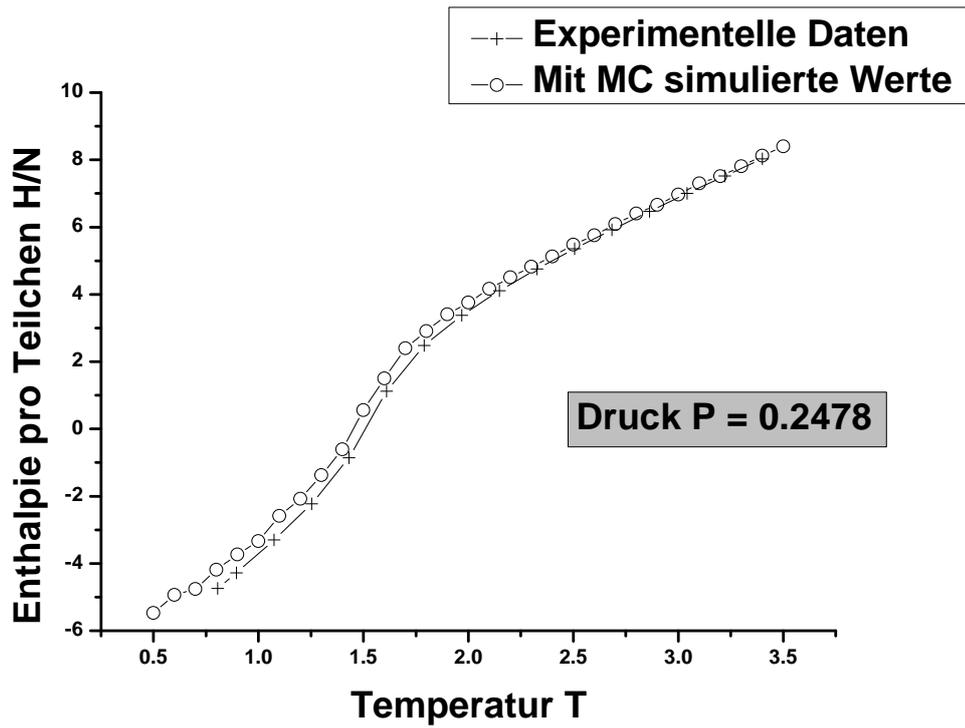


Abbildung 4.7: Temperaturabhängigkeit der Enthalpie $\frac{H}{N}$ und der Wärmekapazität $\frac{c_p}{R}$ von Argon.

4.3 μ VT-Ensemble

mit variabler Teilchenanzahl erzeugt werden. Hierfür werden das Volumen V , die Temperatur T und das chemische Potential μ vorgegeben. Dieses Ensemble wird als großkanonisches bzw. μ VT-Ensemble bezeichnet. Ausgangspunkt für das Metropolis-kriterium ist wieder die Wahrscheinlichkeitsfunktion. Diese lautet im großkanonischen Ensemble:

$$f \propto \frac{V^N}{N!} \cdot \Lambda_T^{-3N} e^{-\beta U_{pot}} e^{\beta \mu N} \quad (4.15)$$

Hierbei bezeichnet Λ_T die thermische Wellenlänge, welche das Ergebnis der Impulsintegration darstellt. Diese muss aufgrund der nicht konstanten Teilchenanzahl berücksichtigt werden. Für eine mögliche Änderung der Konfiguration kommen nun drei Möglichkeiten in Frage.

1. **Bewegung** eines zufällig gewählten Teilchens an einen zufälligen Platz im Simulationsvolumen. Hiefür ist das Verhältnis $\frac{f_{neu}}{f_{alt}}$ dasselbe wie im NVT-Ensemble:

$$\frac{f_{neu}}{f_{alt}} = e^{-\beta(U_{neu} - U_{alt})} \quad (4.16)$$

2. **Erzeugung** eines Teilchens an einem zufälligen Ort im Simulationsvolumen:

$$\begin{aligned} \frac{f_{neu}}{f_{alt}} &= \frac{\frac{\Lambda_T^{-3(N+1)}}{(N+1)!} V^{N+1} \exp[-\beta U_{neu} - \mu(N+1)]}{\frac{\Lambda_T^{-3N}}{N!} V^N \exp[-\beta U_{alt} - \mu N]} \\ &= \frac{\Lambda_T^{-3}}{N+1} V \exp[-\beta(U_{neu} - U_{alt} - \mu)] \\ &= \exp \left[-\beta(U_{neu} - U_{alt}) + \ln \left(\frac{aV}{N+1} \right) \right] \end{aligned} \quad (4.17)$$

Die Größe a bezeichnet die Aktivität und ist gegeben durch das chemische Potential μ und die thermische Wellenlänge λ_T :

$$a = \Lambda_T^{-3} e^{\beta \mu}$$

3. **Vernichtung** eines zufällig gewählten Teilchens:

$$\begin{aligned} \frac{f_{neu}}{f_{alt}} &= \frac{\frac{\Lambda_T^{-3(N-1)}}{(N-1)!} V^{N-1} \exp[-\beta U_{neu} - \mu(N-1)]}{\frac{\Lambda_T^{-3N}}{N!} V^N \exp[-\beta U_{alt} - \mu N]} \\ &= \frac{N}{\Lambda_T^{-3} V} \exp[-\beta(U_{neu} - U_{alt} + \mu)] \\ &= \exp \left[-\beta(U_{neu} - U_{alt}) + \ln \left(\frac{N}{aV} \right) \right] \end{aligned} \quad (4.18)$$

Der Metropolisalgorithmus im μVT -Ensemble lautet folglich:

Metropolisalgorithmus im μVT -Ensemble:

1. Erzeuge zufällig eine Konfiguration f_m .
2. Erzeuge zufällig eine weitere Konfiguration f_n , indem du entweder
 - (a) zufällig ein Teilchen verschiebst
 - (b) zufällig ein Teilchen erzeugst
 - (c) zufällig ein Teilchen vernichtest.
3. Erzeuge eine Zufallszahl $\zeta \in [0, 1]$.
4. Akzeptiere f_n , wenn (je nach Fall)
 - (a) $\min(1, \exp[-\beta(U_{neu} - U_{alt})]) \geq \zeta$
 - (b) $\min(1, \exp[-\beta(U_{neu} - U_{alt}) + \ln(\frac{aV}{N+1})]) \geq \zeta$
 - (c) $\min(1, \exp[-\beta(U_{neu} - U_{alt}) + \ln(\frac{N}{aV})]) \geq \zeta$wenn nicht nehme erneut f_m .
5. Gehe zu Schritt 2.

Anmerkung:

Da im μVT -Ensemble die Teilchenanzahl nicht konstant ist, bietet sich eine dynamische Programmierung an. Dynamische Vektoren, d.h. Vektoren deren Längen während der Simulation verändert werden können, sind beispielsweise in der C++ Bibliothek zu finden. Mehr Informationen sind unter www.cplusplus.com zu finden. Die für eine μVT -Simulation wichtigsten Eigenschaften sind folgende:

```
#include <vector>
using namespace std;

typedef struct{
    double x;
    double y; // Schwerpunkt in s-Koordinaten s=r/L
    double z;
} Teilchen;

//Erzeuge einen Vektor mit dem Namen teilchen_liste
//und Teilchen als Elemente
```

```

vector<Teilchen> teilchen_liste;

int main (void)
{
    // Definiere ein Teilchen mit den Koordinaten (0/0/0)
    Teilchen start_teilchen;
    start_teilchen.x=0;
    start_teilchen.y=0;
    start_teilchen.z=0;

    // Hänge ein Teilchen mit den Koordinaten (0/0/0)
    // an das Ende des Vektors
    teilchen_liste.push_back (start_teilchen);

    // Setze die y-Koordinate des N-ten Teilchens gleich 0.5
    teilchen_liste[N].y=0.5

    // Lösche das N-te Teilchen
    teilchen_liste.erase(liste.begin() + N);
}

```

Beispiel: Adsorption von Methan an einer Graphitoberfläche

Mit dem bisherigen Wissen über μ VT-Simulationen lässt sich die Adsorption von Methan an einer Graphit Oberfläche untersuchen. Dafür wird zunächst die Funktionalität der Simulation getestet. In Quelle [3] auf Seite 179 ist in Abbildung 4.4 der Verlauf des externen Anteils des chemischen Potentials μ_{ex} zu finden. Dieser beträgt bei einer Dichte von $\rho = 0.6365$ und einer Temperatur von $T = 1.4$ gerade $\mu_{ex} = -1.81$. Somit kann unsere Simulation überprüft werden, indem bei $T = 1.4$ und $\mu_{ex} = -1.81$ die Dichte ρ bestimmt wird. Für diese sollte dann natürlich $\rho \approx 0.6365$ gelten. In Abbildung (4.8) ist der zugehörige Verlauf der Dichte zu finden.

Anmerkung:

An dieser Stelle noch ein paar kurze Anmerkungen zum chemischen Potential μ . Dieses setzt sich aus einem wechselwirkungsfreien Anteil μ_{id} und einem Wechselwirkungsanteil μ_{ex} zusammen.

$$\begin{aligned}\mu &= \mu_{id} + \mu_{ex} \\ &= T \cdot \ln[\Lambda_T^3 \rho] + \mu_{ex}\end{aligned}$$

Somit folgt für die oben eingeführte Aktivität a :

$$\begin{aligned}a &= \Lambda_T^{-3} \exp[\beta\mu] \\ &= \Lambda_T^{-3} \exp[\beta(\mu_{id} + \mu_{ex})] \\ &= \rho \cdot \exp[\beta\mu_{ex}]\end{aligned}$$

Dies hat den Vorteil, dass die thermische Wellenlänge Λ_T bei der Simulation nicht verwendet werden muss.

Abbildung 4.8: Verlauf der Dichte ρ bei $T = 1.4$ und $\mu_{ex} = -1.81$. Der bestimmte Wert der Dichte beträgt $\rho = 0.64$.

Nachdem wir wissen, dass unsere Simulation funktioniert, ist es möglich, die radiale Paarkorrelationsfunktion $g_2(r)$ nach der Thermalisierung unseres Systems zu berechnen.

$$g_2(r) = \frac{\rho(i)}{\rho_{bulk}} \quad \text{mit} \quad \rho_{bulk} = \frac{N}{V} \quad \text{und} \quad \rho(i) = \frac{n(r, \Delta r)}{4\pi r^2 \Delta r} = \frac{n(i \cdot \Delta r, \Delta r)}{4\pi(i \cdot \Delta r)^2 \Delta r} \quad (4.19)$$

Hierbei ist $n(r, \Delta r)$ die mittlere Anzahl an Teilchen in einer Kugelschale der Breite Δr , in deren Zentrum sich ein zufälliges Teilchen befindet. In Abbildung (4.9) ist die radiale Paarkorrelationsfunktion $g_2(r)$ für $\Delta r = 0.05$ zu finden.

Die "kleine DichteNäherung von $g_2(r)$ besagt, dass sich die Teilchen am "liebsten in einem Abstand von ($r = 1$) zueinander aufhalten. Bei der von uns bestimmten radialen Paarkorrelationsfunktion $g_2(r)$ ist zusätzlich zu dem großen Peak bei ($r = 1$) noch ein weiterer kleinerer Peak bei ($r=2$) zu erkennen. Dies bedeutet, dass sich die Teilchen zusätzlich vermehrt in einem Abstand von $r = 2$ zueinander aufhalten.

Im nächsten Schritt betrachten wir ein Simulationsvolumen (L, L, L_z), in welchem sich an der Stelle $z = 0$ eine zusätzliche Oberfläche befindet. Zu dem Lennard-Jones-Potential kommt somit zusätzlich ein Potential, welches die Wechselwirkung zwischen den Teilchen und der Oberfläche beschreibt.

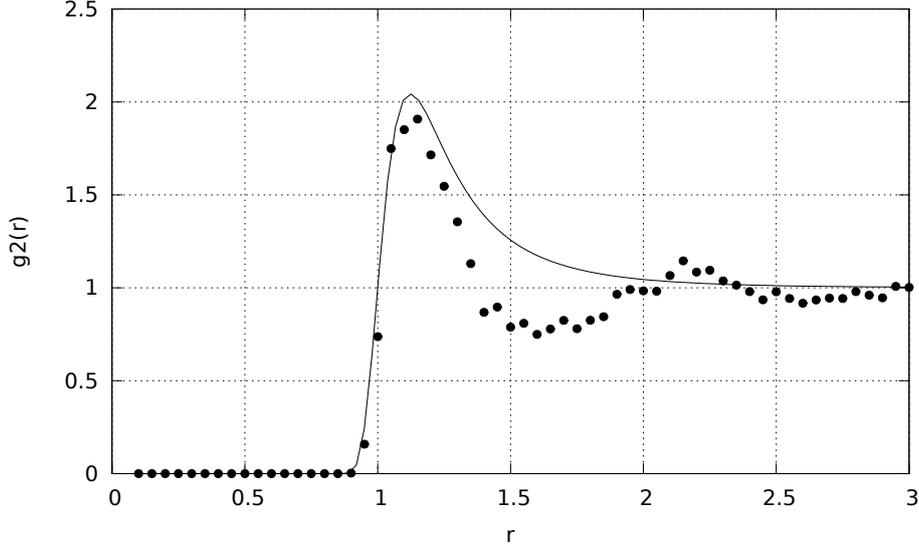


Abbildung 4.9: Simulationsergebnisse für die radiale Paarkorrelationsfunktion $g_2(r)$ (ausgefüllte Kreise) und "kleine Dichte Näherung" $g_2(r) \approx \exp[-\beta u_{LJ}(r)]$ (durchgezogene Linie).

$$U(\vec{r}) = \sum_{i < j} \underbrace{4(r_{ij}^{-12} - r_{ij}^{-6})}_{u_{LJ}(r_{ij})} + \sum_{i=1}^N \underbrace{17.908 \left[\frac{2}{5} \left(\frac{1.034}{z_i} \right)^{10} - \left(\frac{1.034}{z_i} \right)^4 \right]}_{u_{\text{Oberfläche}}(z_i)}$$

Die hier verwendeten Parameter simulieren die Wechselwirkung zwischen Methan und einer Graphitoberfläche. Zwei weitere Änderungen in unserer Simulation sind noch durchzuführen. Zum einen muss aufgrund der Graphitoberfläche bei $z=0$ die "minimum image" Konvention in Z -Richtung und zum anderen aufgrund der gebrochenen Kugelsymmetrie die Langreichweitekorrektur der potentiellen Energie ausgeschaltet werden. Mit Hilfe von $\beta\mu \approx \ln[\rho_{bulk}\Lambda_T^3] + 2\rho_{bulk}B_2(T)$, wobei die Dichte ρ_{bulk} weit entfernt von der Oberfläche ist und $B_2(T)$ (siehe Gleichung 2.80 in Quelle [3]) den zweiten Virialkoeffizienten bezeichnet, lässt sich μ_{ex} wie folgt berechnen:

$$\mu_{ex} = 2\rho_{bulk}B_2(T) \quad (4.20)$$

In Abbildung (4.10) ist die Dichte einer Scheibe mit Volumen ($V = L \cdot L \cdot \Delta L_z$) als Funktion der Höhe zu finden. Hierbei sind einerseits zwei Dichtepicks, welche für die zwei Adsorbenschichten stehen, und andererseits die Annäherung an ρ_{bulk} für große Abstände von der Graphitoberfläche bei $z=0$ zu erkennen.

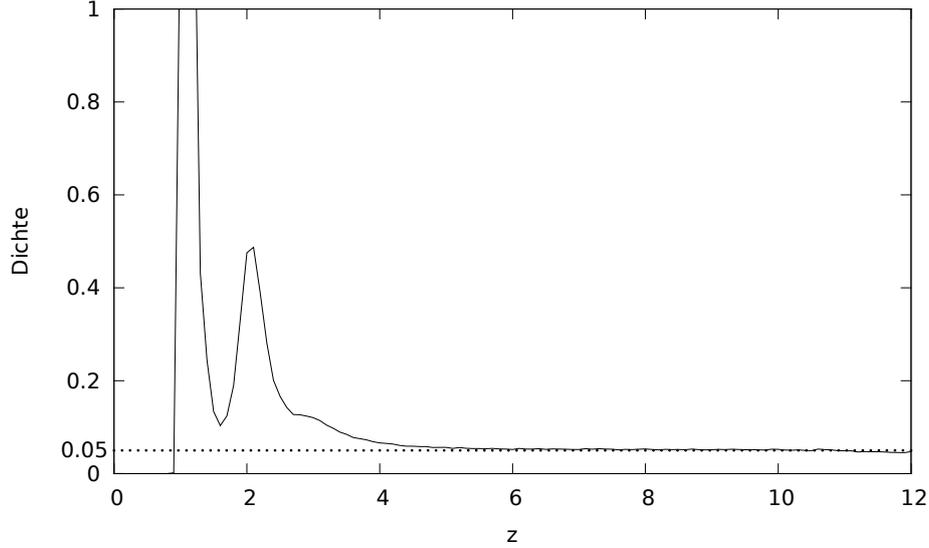


Abbildung 4.10: Verlauf der Dichte in Abhängigkeit von der Höhe z bei der Adsorption von Methan an einer Graphitoberfläche bei $z=0$.

($T = 1.53$, $\rho_{bulk} = 0.05$, $L = 6$, $L_z = 12$, $r_{cut} = 3$, $\Delta z = 0.1$)

4.4 Gibbs-Ensemble

Das in diesem Kapitel behandelte Gibbs-Ensemble stellt eine Kombination der vorrigen drei Ensemble (NVT, NPT, μ VT) dar. Hierfür wird eine weitere Simulationsschachtel mit Volumen V_2 verwendet, in der simultan simuliert wird. Die äußeren Randbedingungen sind, dass das Gesamtvolumen $V = V_1 + V_2$ und die Gesamtteilchenanzahl $N = N_1 + N_2$ konstant bleiben. In der Vorstellung grenzen die beiden Simulationsschachteln aneinander und können mit bestimmten Wahrscheinlichkeiten Teilchen- und Volumentransfers durchführen. Aufgrund dieser Transfermöglichkeiten entsteht eine Angleichung des chemischen Potentials und des Drucks in beiden Simulationsschachteln. Somit ist diese Methode beispielsweise gut für die Untersuchung von Phasenübergängen und Koexistenzgebieten geeignet. Durch den Angleich von μ und P werden sich in den beiden Volumina jeweils die verschiedenen Phasen bilden.

Um die Metropolis-kriterien herzuleiten betrachten wir zunächst die Wahrscheinlichkeit einer bestimmten Konfiguration mit N_1 Teilchen im Volumen V_1 und den Positionen $\{\vec{r}_1\}$ in V_1 , sowie $\{\vec{r}_2\}$ in V_2 .

$$f_{Gibbs} \propto \frac{V_1^{N_1} (V - V_1)^{N - N_1} N!}{N_1! (N - N_1)!} \exp[-\beta(U(\{\vec{r}_1\}) + U(\{\vec{r}_2\}))] \quad (4.21)$$

Im Gibbs-Ensemble werden nun drei mögliche Konfigurationsänderungen betrachtet.

1. **Bewegung** eines zufällig gewählten Teilchens innerhalb einer der Simulations-

schachteln (z.B. in V_1). Hierfür ist das Verhältnis $\frac{f_{Gibbs}^{neu}}{f_{Gibbs}^{alt}}$ das gleiche wie im NVT-Ensemble:

$$\frac{f_{Gibbs}^{neu}}{f_{Gibbs}^{alt}} = \exp[-\beta(U(\{\vec{r}_1\})^{neu} - U(\{\vec{r}_1\})^{alt})] \quad (4.22)$$

2. **Volumentransfer** zwischen Box 1 und Box 2. Aufgrund der Erhaltung des Gesamtvolumens bewirkt eine Volumensänderung in Box 1 um ΔV eine Volumensänderung in Box 2 um $-\Delta V$. Für $\frac{f_{Gibbs}^{neu}}{f_{Gibbs}^{alt}}$ folgt somit:

$$\frac{f_{Gibbs}^{neu}}{f_{Gibbs}^{alt}} = \frac{(V_1^{neu})^{N_1} (V - V_1^{neu})^{N-N_1}}{(V_1^{alt})^{N_1} (V - V_1^{alt})^{N-N_1}} \exp[-\beta(U(\{\vec{r}\})^{neu} - U(\{\vec{r}\})^{alt})] \quad (4.23)$$

3. **Teilchentransfer** z.B. aus Box 1 in Box 2. Aufgrund der Erhaltung der Gesamtteilchenanzahl bewirkt ein Teilchentransfer von Box 1 in Box 2 eine Änderung von $N_1^{neu} = N_1^{alt} - 1$ bzw. $N_2^{neu} = N_2^{alt} + 1$. Somit ergibt sich:

$$\frac{f_{Gibbs}^{neu}}{f_{Gibbs}^{alt}} = \frac{N_1(V - V_1)}{V_1(N - N_1 + 1)} \exp[-\beta(U(\{\vec{r}\})^{neu} - U(\{\vec{r}\})^{alt})] \quad (4.24)$$

Der Metropolisalgorithmus im Gibbs-Ensemble lautet folglich:

Metropolisalgorithmus im Gibbs-Ensemble:

1. Erzeuge zufällig eine Konfiguration f_m .
2. Erzeuge zufällig eine weitere Konfiguration f_n , indem du entweder
 - (a) zufällig ein Teilchen innerhalb einer der Boxen verschiebst
 - (b) zufällig einen Volumentransfer durchführst
 - (c) zufällig einen Teilchentransfer durchführst.
3. Erzeuge eine Zufallszahl $\zeta \in [0, 1]$.
4. Akzeptiere f_n , wenn (je nach Fall)

$$(a) \min(1, \exp[-\beta(U_{neu} - U_{alt})]) \geq \zeta$$

$$(b) \min\left(1, \frac{(V_1^{neu})^{N_1} (V - V_1^{neu})^{N-N_1}}{(V_1^{alt})^{N_1} (V - V_1^{alt})^{N-N_1}} \exp[-\beta(U(\{\vec{r}\})^{neu} - U(\{\vec{r}\})^{alt})]\right) \geq \zeta$$

$$(c) \min\left(1, \frac{N_1(V - V_1)}{V_1(N - N_1 + 1)} \exp[-\beta(U(\{\vec{r}\})^{neu} - U(\{\vec{r}\})^{alt})]\right) \geq \zeta$$

wenn nicht, nehme erneut f_m .

5. Gehe zu Schritt 2.

In Abbildung 4.11 sind die verschiedenen Möglichkeiten einer Konfigurationsänderung im Gibbs-Ensemble zu finden.

Anmerkung:

Soll mit Hilfe des Gibbs-Ensembles die Phasenseparation eines Stoffes untersucht werden, ist zu beachten, dass es rein zufällig ist, in welcher Box sich welche Phase bilden wird.

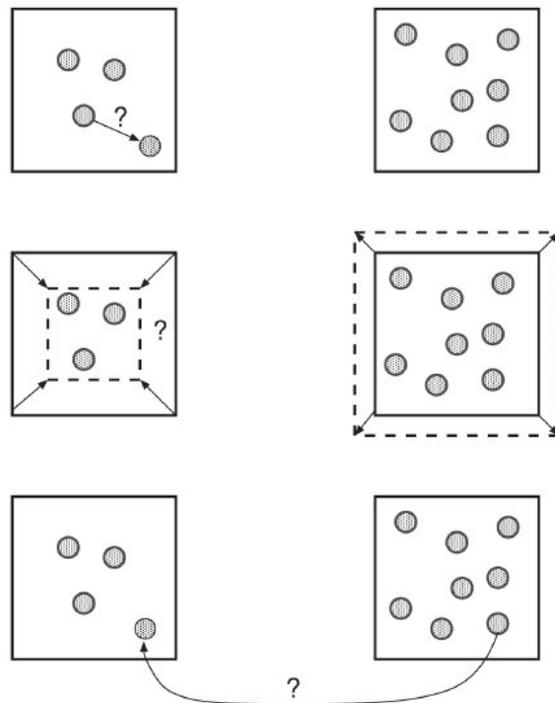


Abbildung 4.11: Möglichkeiten für eine Konfigurationsänderung im Gibbs-Ensemble (Entnommen aus Quelle [1])

Beispiel: Phasendiagramm eines Lennard-Jones Fluids in der Temperatur-Dichte-Ebene

Ziel ist die Erstellung eines Phasendiagramms des Lennard-Jones Fluids ($\sigma = \epsilon = 1$) in der Temperatur-Dichte-Ebene. Hierfür werden mehrere Gibbs-Ensemble-Simulationen für verschiedene Temperaturen von $T = 1$ bis $T = 1.5$ durchgeführt. Die weiteren Parameter sind $N = 256$, $r_{cut} = 3$ und $\rho_{start} = 0.3$.

In beiden Boxen wird jeweils mit einer Teilchenanzahl von $N_1 = N_2 = 128$ und einer Dichte $\rho_1 = \rho_2 = 0.3$ gestartet. Aufgrund der Gibbs-Ensemble Schritte wird es zu einem Angleich des chemischen Potentials und des Drucks in beiden Boxen kommen. Dies hat zur Folge, dass sich in der einen Box die flüssige Phase und in der anderen Box die gasförmige Phase bilden werden.

In Abbildung (4.13) sind beispielhaft die Verläufe der Dichten der beiden Boxen zu verschiedenen Temperaturen zu finden. Hier ist sehr gut zu sehen, dass bei niedrigen Temperaturen die Phasen in den Boxen separieren. Kommt man allerdings zu höheren Temperaturen beginnen die Dichteschwankungen größer zu werden, bis schließlich keine Phasenseparation mehr stattfindet. Dieses Verhalten ist ein deutliches Indiz für eine Annäherung an den kritischen Punkt des Lennard-Jones Fluids. Es ist zu erwarten, dass die Phasen unterhalb der kritischen Temperatur separieren und ab der kritischen Temperatur nicht mehr zu unterscheiden sind. Dies gilt natürlich nur, wenn man sich gleichzeitig im Bereich der kritischen Dichte befindet, welche in unserem Fall in der Nähe von $\rho \approx 0.3$ liegt. Mit Hilfe dieser Dichtemessungen ist es nun möglich das Phasendiagramm in der Temperatur-Dichte-Ebene aufzustellen (siehe Abbildung (4.12)).

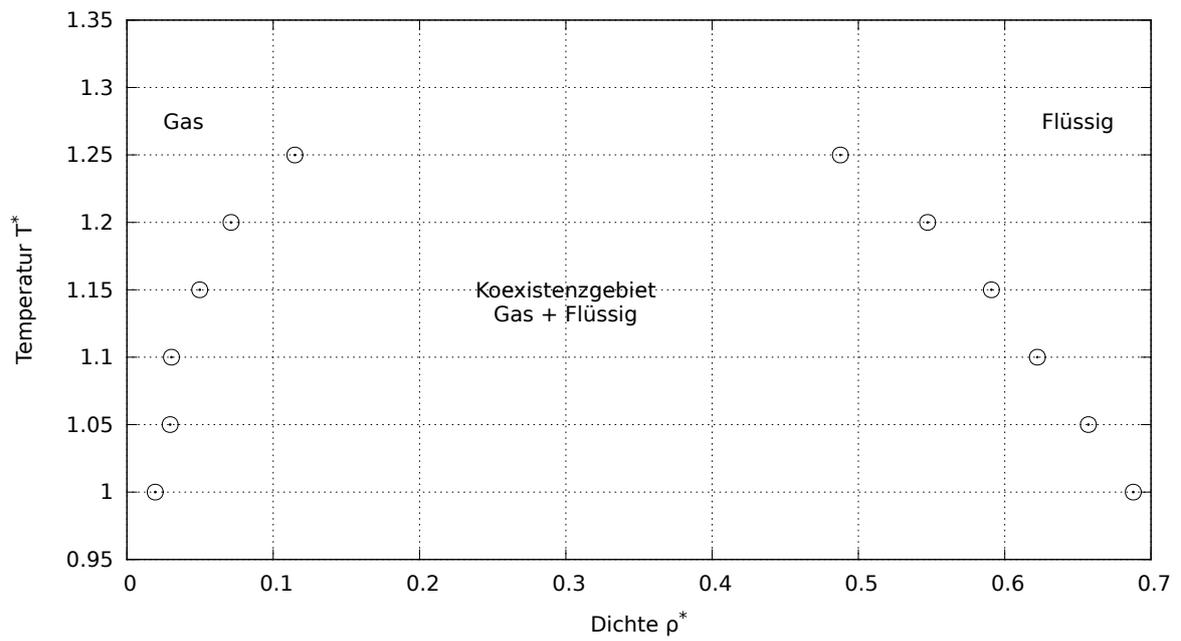


Abbildung 4.12: Phasendiagramm des Lennard-Jones Fluids ($\sigma = \epsilon = 1$) in der Temperatur-Dichte Ebene

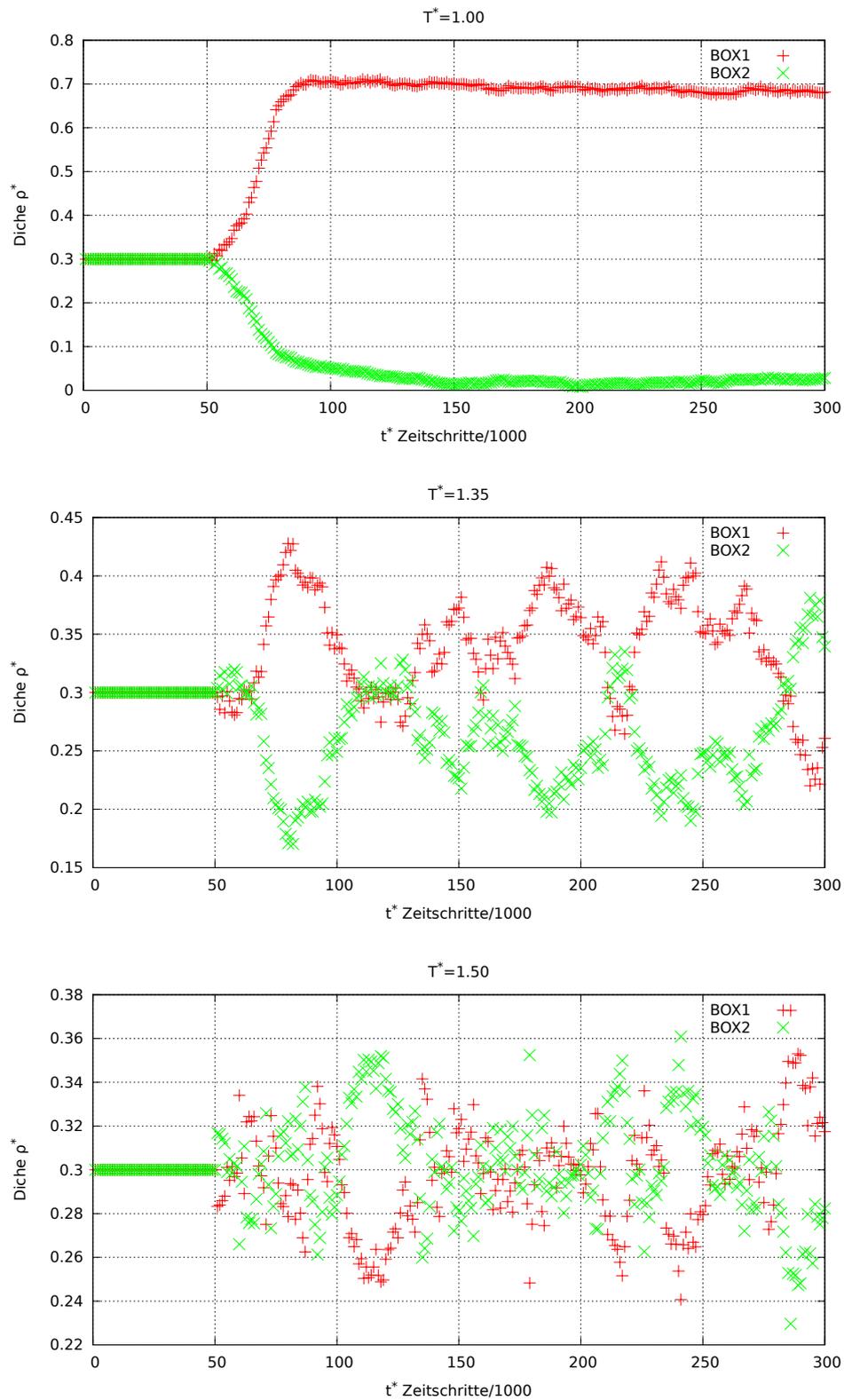


Abbildung 4.13: Dichteverläufe in den beiden Gibbs-Ensemble Boxen für verschiedene Temperaturen ($T=1$, $T=1.35$, $T=1.50$).

5 MC-Simulation eines Lennard-Jones Fluids bestehend aus starren Molekülen

Nachdem wir uns in den vorangegangenen Kapiteln mit der Simulation von punktförmigen Teilchensystemen beschäftigt haben, wollen wir diese Ergebnisse nun auf Systeme mit ausgedehnten Molekülen erweitern.

Hierbei beschäftigen wir uns zunächst mit der Simulation von starren Molekülen (siehe Abbildung (5.1)). „Starr“ bedeutet, dass die Vibrationen bzw. Normalschwingungen der Moleküle vernachlässigt werden. Dies gilt ebenfalls für Deformationen aufgrund äußerer Kräfte. Näherungsweise gilt dies für kleine Moleküle wie H_2O , CO_2 , CO , N_2 , NH_3 , C_6H_6 und weitere.

Bevor wir uns um deren Beschreibung kümmern, möchten wir an dieser Stelle kurz den Symmetriefaktor σ einführen, welcher, wie es der Name schon verrät, die Anzahl an Symmetrien des Moleküls angibt. Beispielsweise ist der Symmetriefaktor im Fall von Wasser $\sigma = 2$ und im Fall eines linearen Moleküls mit unterschiedlichen Atomen $\sigma = 1$. Um bei späteren Integrationen über den gesamten Phasenraum Mehrfachzählungen zu vermeiden, müssen diese Symmetrien berücksichtigt werden und durch ihre Anzahl σ geteilt werden (ähnlich wie der Faktor $\frac{1}{N!}$ bei ununterscheidbaren Teilchen). Die Ausdehnung der Moleküle hat zur Folge, dass wir zusätzlich zu den Translationen auch Rotationen betrachten müssen. Hierbei spielt die Orientierung dieser eine wichtige Rolle.

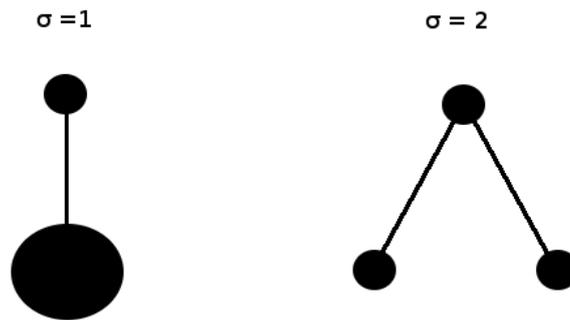


Abbildung 5.1: Beispiele für starre Moleküle mit jeweiligem Symmetriefaktor σ .
 links: lineares Molekül mit unterschiedlichen Atomen
 rechts: Wasser

Eulersche Winkel:

Für eine mögliche Beschreibung der Orientierung eines starren Körpers im Raum eignet sich die Verwendung der Eulerschen Winkel:

$$\varphi \in [0, 2\pi]$$

$$\Theta \in [0, \pi]$$

$$\Psi \in [0, 2\pi]$$

Diese stellen die Verbindung zwischen einem parallel zum Laborsystem ausgerichteten System (x,y,z) und einem mit dem Molekül fest verbundenen System (x',y',z') her. Diese Verbindung funktioniert wie folgt (siehe Abbildung (5.2)):

1. Lege beide Koordinatensysteme deckungsgleich übereinander.
2. Drehe das gestrichene System mit dem Winkel φ um die z -Achse.
3. Drehe das gestrichene System mit dem Winkel Θ um die vorläufig entstandene x' -Achse.
4. Drehe das gestrichene System mit dem Winkel Ψ um die entstandene z' -Achse.

Somit ist es nun möglich jede beliebige Orientierung eines Moleküls bzw. starren Körpers im Raum mit Hilfe der Eulerschen Winkel zu beschreiben. Im Folgenden bezeichnen

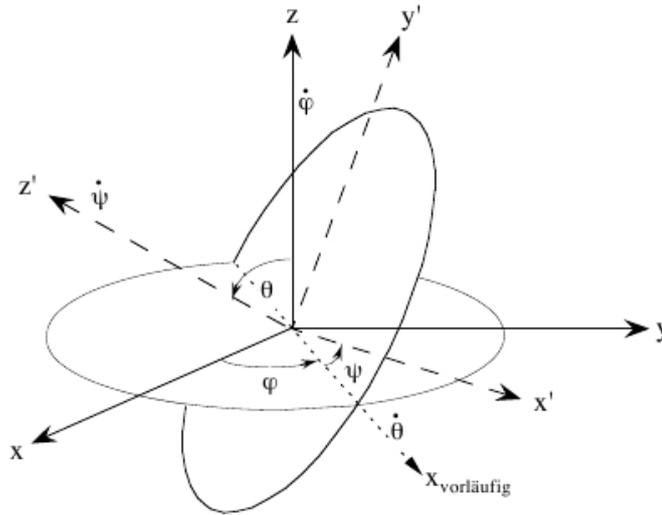


Abbildung 5.2: Visualisierung der Eulerschen Winkel (entnommen Quellen [4]).

Ω_i ($i=1,2,3$) die Winkelgeschwindigkeiten des Moleküls bezüglich seiner gestrichenen Achsen x' , y' und z' . Diese können gemäß

$$\begin{aligned}\Omega_1 &= \dot{\Theta} \cos(\Psi) + \dot{\varphi} \sin(\Theta) \sin(\Psi) \\ \Omega_2 &= -\dot{\Theta} \sin(\Psi) + \dot{\varphi} \sin(\Theta) \cos(\Psi) \\ \Omega_3 &= \dot{\Psi} + \dot{\varphi} \cos(\Theta)\end{aligned}$$

durch die Eulerschen Winkel und ihren Zeitableitungen ausgedrückt werden (siehe Quelle [4]). Wir benötigen diese Gleichungen, um die Lagrange-Funktion und anschließend die Hamilton Funktion des Moleküls durch die Eulerschen Winkel und ihre Zeitableitungen bzw. ihre konjugierten Impulse auszudrücken.

Lagrangeleichung starrer Moleküle:

Im ersten Schritt führen wir ein raumfestes Koordinatensystem (x,y,z) und ein körperfestes Koordinatensystem (x',y',z') ein. Der Schwerpunkt (S) des starren Moleküls, welches aus N Atomen besteht, sei im Ursprung des körperfesten Systems (x',y',z') (siehe Abbildung (5.3)).

In Abbildung (5.3) ist leicht zu sehen, dass $\vec{r}_i = \vec{R} + \vec{r}'_i$ gilt. Damit ergibt sich folgende Relation für die Geschwindigkeit des i -ten Atoms, deren Herleitung in Quelle [4] zu finden ist:

$$\vec{v}_i = \vec{V}_S + \vec{\Omega} \times \vec{r}'_i \quad (5.1)$$

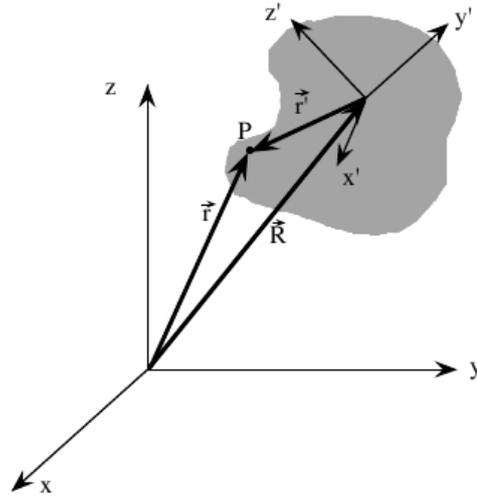


Abbildung 5.3: Ruhendes Bezugssystem (ungestrichen) sowie mit dem starren Körper verbundenes System (gestrichen) (entnommen Quellen [4])

Hierbei bezeichnet \vec{V}_S die Geschwindigkeit des Schwerpunkts des Moleküls und $\vec{\Omega}$ die oben eingeführte Winkelgeschwindigkeit. Betrachtet man nun die kinetische Energie K des starren Moleküls, bestehend aus N Atomen mit $m = \sum_{i=1}^N m_i$, so folgt:

$$\begin{aligned}
 K &= \frac{1}{2} \sum_{i=1}^N m_i \vec{v}_i^2 \\
 &= \frac{1}{2} \sum_{i=1}^N m_i \left(\vec{V}_S + \vec{\Omega} \times \vec{r}'_i \right)^2 \\
 &= \underbrace{\frac{1}{2} m \vec{V}_S^2}_{K_S} + \underbrace{\vec{V}_S \times \vec{\Omega} \sum_{i=1}^N m_i \vec{r}'_i}_{=0(*)} + \underbrace{\sum_{i=1}^N \frac{m_i}{2} \left(\vec{\Omega}^2 r_i'^2 - (\vec{\Omega} \cdot \vec{r}'_i)^2 \right)}_{K_{rot}} \quad (5.2)
 \end{aligned}$$

((*): da Ursprung im Schwerpunkt!), wobei der Beitrag der Rotation bezogen auf den Schwerpunkt K_{rot} gegeben ist durch:

$$K_{rot} = \frac{1}{2} \sum_{\alpha, \beta} \Omega_\alpha \Omega_\beta T_{\alpha, \beta} \quad (5.3)$$

mit dem Trägheitstensor

$$T_{\alpha, \beta} = \sum_i m_i (r_i'^2 \delta_{\alpha, \beta} - r'_{\alpha, i} r'_{\beta, i}) \quad (5.4)$$

Gehen wir nun von dem Fall aus, dass der Trägheitstensor diagonal ist und die Hauptträgheitsmomente (I_1, I_2, I_3) auf der Diagonalen stehen, dann folgt für die Lagrangegleichung:

$$\begin{aligned}
L &= K - U \\
&= K_S + K_{Rot} - U \\
&= \frac{m}{2} \vec{V}_S^2 + \frac{1}{2} (I_1 \Omega_1^2 + I_2 \Omega_2^2 + I_3 \Omega_3^2) - U
\end{aligned} \tag{5.5}$$

Konjugierte Impulse des starren Moleküls:

Für die konjugierten Impulse gilt allgemein:

$$P_{x_i} = \frac{\partial L}{\partial \dot{x}_i}$$

Damit folgt in unserem Fall des starren Moleküls bzw. Rotators unter der Berücksichtigung, dass die Winkelgeschwindigkeit $\vec{\Omega}$ durch die Eulerschen Winkel ausgedrückt werden kann:

$$\begin{aligned}
P_\varphi &= \frac{\partial L}{\partial \dot{\varphi}} = I_1 \Omega_1 \frac{\partial \Omega_1}{\partial \dot{\varphi}} + I_2 \Omega_2 \frac{\partial \Omega_2}{\partial \dot{\varphi}} + I_3 \Omega_3 \frac{\partial \Omega_3}{\partial \dot{\varphi}} \\
&= I_1 \Omega_1 \sin(\Theta) \sin(\Psi) + I_2 \Omega_2 \sin(\Theta) \cos(\Psi) + I_3 \Omega_3 \cos(\Theta) \\
P_\Theta &= \frac{\partial L}{\partial \dot{\Theta}} = I_1 \Omega_1 \frac{\partial \Omega_1}{\partial \dot{\Theta}} + I_2 \Omega_2 \frac{\partial \Omega_2}{\partial \dot{\Theta}} + I_3 \Omega_3 \frac{\partial \Omega_3}{\partial \dot{\Theta}} \\
&= I_1 \Omega_1 \cos(\Psi) - I_2 \Omega_2 \sin(\Psi) \\
P_\Psi &= \frac{\partial L}{\partial \dot{\Psi}} = I_1 \Omega_1 \frac{\partial \Omega_1}{\partial \dot{\Psi}} + I_2 \Omega_2 \frac{\partial \Omega_2}{\partial \dot{\Psi}} + I_3 \Omega_3 \frac{\partial \Omega_3}{\partial \dot{\Psi}} \\
&= I_3 \Omega_3
\end{aligned}$$

Da sich später bei der Bestimmung der Zustandssumme Q_{NVT} herausstellen wird, dass wir eine Koordinatentransformation von $dP_\varphi dP_\Theta dP_\Psi$ nach $d\Omega_1 d\Omega_2 d\Omega_3$ benötigen, wollen wir uns nun dieser Transformation widmen.

Hierfür stellen wir zunächst die Jakobimatrix J auf:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial P_\varphi}{\partial \Omega_1} & \frac{\partial P_\varphi}{\partial \Omega_2} & \frac{\partial P_\varphi}{\partial \Omega_3} \\ \frac{\partial P_\Theta}{\partial \Omega_1} & \frac{\partial P_\Theta}{\partial \Omega_2} & \frac{\partial P_\Theta}{\partial \Omega_3} \\ \frac{\partial P_\Psi}{\partial \Omega_1} & \frac{\partial P_\Psi}{\partial \Omega_2} & \frac{\partial P_\Psi}{\partial \Omega_3} \end{pmatrix} = \begin{pmatrix} I_1 \sin(\Theta) \sin(\Psi) & I_2 \sin(\Theta) \cos(\Psi) & I_3 \cos(\Theta) \\ I_1 \cos(\Psi) & -I_2 \sin(\Psi) & 0 \\ 0 & 0 & I_3 \end{pmatrix}$$

Somit folgt für die Koordinatentransformation:

$$\begin{aligned} dP_\varphi dP_\Theta dP_\Psi &= |\det J| d\Omega_1 d\Omega_2 d\Omega_3 \\ &= I_1 I_2 I_3 \sin(\Theta) d\Omega_1 d\Omega_2 d\Omega_3 \end{aligned} \quad (5.6)$$

Damit ist es nun möglich die Zustandssumme für Ensemble starrer Moleküle aufzustellen.

5.1 NVT-Ensemble

Wie oben beschrieben stellen wir uns jedes Molekül als einen starren Rotator vor, welcher durch seinen Schwerpunkt (S) und seine Orientierung festgelegt ist. Hierbei bietet es sich an, den Hamiltonoperator in einen Translationstherm H_{trans} und einen Rotationstherm H_{rot} aufzuteilen. Die Translationen beziehen sich jeweils auf die Schwerpunktbewegung und die Rotationen auf die Orientierung. Damit folgt für die Zustandssumme im NVT-Ensemble:

$$Q_{NVT}^{(N)} = \frac{1}{N!} \underbrace{\int \frac{d^{3N} r_s d^{3N} p_s}{h^{3N}} e^{-\beta H_{trans}}}_{(*)} \underbrace{\int \frac{d^N \varphi d^N \Theta d^N \Psi d^N P_\varphi d^N P_\Theta d^N P_\Psi}{\sigma^N h^{3N}} e^{-\beta H_{rot}}}_{(**)}$$

Wobei σ den eingeführten Symmetriefaktor bezeichnet.

Zuerst wollen wir uns um den Faktor (*) kümmern, welcher analog zu dem Falle punktförmiger Teilchen behandelt werden kann.

$$\begin{aligned} (*) &= \frac{V^N}{N! h^{3N}} \int d^{3N} s_s d^{3N} p_s e^{-\beta(K_{trans}^{kin} + U_{trans})} \\ &= \frac{V^N}{N! h^{3N}} \underbrace{\int d^{3N} p_s e^{-\beta K_{trans}^{kin}}}_{\Lambda_T^{-3N}} \int d^{3N} s_s e^{-\beta U_{trans}} \\ &= \frac{V^N}{N! h^{3N} \Lambda_T^{3N}} \int d^{3N} s_s e^{-\beta U_{trans}} \end{aligned}$$

Bei der Behandlung des zweiten Faktors (**) können wir das Wissen über das Transformationsverhalten der konjugierten Impulse anwenden.

$$\begin{aligned}
(**) &= \int \frac{d^N \varphi d^N \Theta d^N \Psi d^N P_\varphi d^N P_\Theta d^N P_\Psi}{\sigma^N \hbar^{3N}} e^{-\beta H_{rot}} \\
&= \frac{1}{\sigma^N \hbar^{3N}} \int d^N P_\varphi d^N P_\Theta d^N P_\Psi e^{-\beta \frac{1}{2}(I_1 \Omega_1^2 + I_2 \Omega_2^2 + I_3 \Omega_3^2)} \int d^N \varphi d^N \Theta d^N \Psi e^{-\beta U_{rot}(\varphi, \Theta, \Psi)} \\
&= \frac{1}{\sigma^N \hbar^{3N}} \underbrace{\int d^N \Omega_1 d^N \Omega_2 d^N \Omega_3 e^{-\beta \frac{1}{2}(I_1 \Omega_1^2 + I_2 \Omega_2^2 + I_3 \Omega_3^2)}}_{\lambda_T^{-N}} \int d^N \varphi d^N \Theta d^N \Psi \sin^N(\Theta) e^{-\beta U_{rot}(\varphi, \Theta, \Psi)} \\
&= \frac{1}{\lambda_T^N} \int d^N \varphi d^N \Theta d^N \Psi \sin^N(\Theta) e^{-\beta U_{rot}(\varphi, \Theta, \Psi)}
\end{aligned}$$

Wobei gilt:

$$\begin{aligned}
\sin^N(\Theta) &= \sin(\Theta_1) \cdot \sin(\Theta_2) \cdots \sin(\Theta_N) \\
\lambda_T &= \frac{\sigma}{(I_1 I_2 I_3)^{1/2}} \left(\frac{\beta \hbar^2}{2\pi} \right)^{3/2}
\end{aligned}$$

Die Integrationen über die Winkel wollen wir nun normieren. Dies verläuft ähnlich zu der Einführung der s-Koordinaten im NPT-Ensemble. Wir führen folgende Substitution durch:

$$\varphi' = \frac{\varphi}{2\pi} \Rightarrow d\varphi = 2\pi d\varphi' \quad \Theta' = \frac{\Theta}{\pi} \Rightarrow d\Theta = \pi d\Theta' \quad \Psi' = \frac{\Psi}{2\pi} \Rightarrow d\Psi = 2\pi d\Psi'$$

Damit folgt dann:

$$\begin{aligned}
&\frac{1}{\lambda_T^N} \int_0^{2\pi} d^N \varphi \int_0^\pi d^N \Theta \int_0^{2\pi} d^N \Psi \sin^N(\Theta) e^{-\beta U_{rot}(\varphi, \Theta, \Psi)} \\
&= \frac{1}{\lambda_T^N} \int_0^1 (2\pi)^N d^N \varphi' \int_0^1 \pi^N d^N \Theta' \int_0^1 (2\pi)^N d^N \Psi' \sin^N(\pi \Theta') e^{-\beta U_{rot}(\varphi', \Theta', \Psi')}
\end{aligned}$$

Es ergibt sich also für die Zustandssumme $Q_{NVT}^{(N)}$:

$$Q_{NVT}^{(N)} = \frac{(4\pi^3)^N V^N}{N! \hbar^{3N} \Lambda_T^{3N} \lambda_T^N} \int d^{3N} s_s e^{-\beta U_{trans}} \int d^N \varphi' d^N \Theta' d^N \Psi' \sin^N(\pi \Theta') e^{-\beta U_{rot}(\varphi', \Theta', \Psi')} \quad (5.7)$$

Damit ist es uns gelungen, die Wahrscheinlichkeitsfunktion $f_{NVT}^{(N)}$ für starre Moleküle im NVT-Ensemble herzuleiten, mit welcher es nun möglich ist, den Metropolis Algorithmus anzuwenden.

$$f_{NVT}^{(N)} \propto \frac{(4\pi^3)^N V^N}{N! \hbar^{3N} \Lambda_T^{3N} \lambda_T^N} \sin^N(\pi \Theta') e^{-\beta(U_{trans} + U_{rot}(\varphi', \Theta', \Psi'))} \quad (5.8)$$

Neben der zufälligen Verschiebung eines Moleküls kommt nun noch eine zufällige Änderung der Orientierung hinzu. Dies führt zu einem zusätzlichen MC-Schritt.

Zusätzlicher MC-Schritt im NVT-Ensemble:

1. Erzeuge zufällig eine neue Konfiguration f_{NVT}^{neu} , indem du zufällig ein Molekül auswählst und zufällig neue Eulersche Winkel des Moleküls bestimmst:

$$\varphi_{neu} \in [0, 2\pi]$$

$$\Theta_{neu} \in [0, \pi]$$

$$\Psi_{neu} \in [0, 2\pi]$$

2. Für $\frac{f_{NVT}^{neu}}{f_{NVT}^{alt}}$ gilt dann:

$$\frac{f_{NVT}^{neu}}{f_{NVT}^{alt}} = \frac{\sin(\Theta^{neu})}{\sin(\Theta^{alt})} e^{-\beta(U_{trans}^{neu} + U_{rot}^{neu} - U_{trans}^{alt} - U_{rot}^{alt})}$$

Spezialfall eines linearen Moleküls:

Im Fall eines linearen Moleküls gilt $\Psi = 0$, da eine Rotation um den Winkel Ψ keine "neue" Konfiguration liefern würde. Setzt man $\Psi = 0$ nun in die Jakobimatrix ein folgt:

$$J = \begin{pmatrix} 0 & I_2 \sin(\Theta) & I_3 \cos(\Theta) \\ I_1 & 0 & 0 \\ 0 & 0 & I_3 \end{pmatrix}$$

Der Betrag der Jakobideterminanten ändert sich allerdings nicht:

$$|\det J| = I_1 I_2 I_3 \sin(\Theta) \quad (5.9)$$

und darf daher nicht weggelassen werden.

Anmerkung:

Will man einen zufälligen Punkt auf der Einheitskugel wählen, ist der erste Gedanke wahrscheinlich, einfach zufällige Polarkoordinaten zu bestimmen. Dies führt

allerdings dazu, dass die Orientierungen nicht gleichverteilt sind. Man kann sich überlegen, dass bei diesem Verfahren auf jedem Breitengrad gleich viele zufällige Punkte gewählt werden. Allerdings werden die Breitengrade in Richtung der Pole immer "kürzer" und es entsteht so eine ungewollte Gewichtung der Polrichtungen. Diese Gewichtung wird gerade durch den Term " $\sin(\Theta)$ " verhindert.

5.2 NPT-Ensemble

Die Berechnung der Wahrscheinlichkeitsfunktion des NPT-Ensembles verläuft weitestgehend analog zu der Rechnung für das NVT-Ensemble. Im Gegensatz zu einem Ensemble mit Punktteilchen, kommt für den Fall starrer Moleküle allerdings folgender MC-Schritt hinzu.

Zusätzlicher MC-Schritt im NPT-Ensemble:

1. Erzeuge zufällig eine neue Konfiguration f_{NPT}^{neu} , indem du zufällig ein Molekül auswählst und zufällig neue Eulersche Winkel des Moleküls bestimmst:

$$\varphi_{neu} \in [0, 2\pi]$$

$$\Theta_{neu} \in [0, \pi]$$

$$\Psi_{neu} \in [0, 2\pi]$$

2. Für $\frac{f_{NPT}^{neu}}{f_{NPT}^{alt}}$ gilt dann:

$$\frac{f_{NPT}^{neu}}{f_{NPT}^{alt}} = \frac{\sin(\Theta^{neu})}{\sin(\Theta^{alt})} \exp \left\{ -\beta [U_{trans}^{neu} + U_{rot}^{neu} - U_{trans}^{alt} - U_{rot}^{alt} + P(V^{neu} - V^{alt})] + N \ln \left(\frac{V^{neu}}{V^{alt}} \right) \right\}$$

6 Quellen

- [1] R. Hentschke, E.M. Aydt, B. Fodi, E. Stöckelmann: "*Molekulares Modellieren mit Kraftfeldern*" 24. März 2004 <http://constanze.materials.uni-wuppertal.de>
- [2] M.E.J. Newman, G.T. Barkema (1999): "*Monte Carlo Methods in Statistical Physics*". New York: Oxford University Press.
- [3] R. Hentschke (2004) - Statistische Mechanik, Wiley-VCH
- [4] R. Hentschke Vorlesungsskript "*Klassische Mechanik*"