

Die Monte-Carlo-Methode zur numerischen Auswertung von Pfadintegralen

Ausarbeitung zum Seminarvortrag vom 11.07.2012

Markus Michael

1 Einleitung

In den vorangegangenen Vorträgen haben wir eine Reihe verschiedener Einsatzmöglichkeiten für den Pfadintegralformalismus in Quantenmechanik und Quantenfeldtheorie kennengelernt. Trotz der häufig sehr anschaulichen Interpretationsmöglichkeiten des Funktionalintegralansatzes werden doch die zu berechnenden Ausdrücke schnell sehr unübersichtlich und sind häufig nicht exakt lösbar. So schreibt Richard MacKenzie in einem einführenden Aufsatz über besagten Formalismus sehr treffend, dass die Hauptarbeit beim Umgang mit Pfadintegralen darin bestehe, zu verhindern, überhaupt eines ausrechnen zu müssen. Tatsächlich wurde bei den bisher vorgestellten Lösungsstrategien in der Regel die direkte Berechnung des Pfadintegrals vermieden. Ein Ansatz war die Entwicklung der Pfade um den klassischen Pfad, wobei die Abweichung in quadratischer Näherung betrachtet wurde und somit lediglich ein gaußartiges Integral zu lösen war. An anderen Stellen wiederum wurde das eigentlich Pfadintegral als Normierungskonstante deklariert und nicht weiter ausgewertet. In vielen Fällen möchte man aber auch kompliziertere Ausdrücke wie Erwartungswerte von Variablen oder Korrelationsfunktionen berechnen und ist dann auf numerische Methoden angewiesen. Man bedient sich einer Simulationsmethode, die klassischerweise in der statistischen Physik zum Einsatz kommt, der Monte-Carlo-Methode. Im Folgenden soll diese Wahl motiviert werden, indem ein formaler Zusammenhang des Pfadintegrals mit der kanonischen Zustandssumme aus der statistischen Physik aufgezeigt wird, um dann in die eigentliche Simulationstechnik einzuführen und sie schließlich am Beispiel des harmonischen Oszillators zu illustrieren.

2 Pfadintegral und statistische Physik

Zunächst soll, wie bereits erwähnt, auf den engen Zusammenhang zwischen Pfadintegral und kanonischer Zustandssumme eingegangen werden.

Ausgehend vom im ersten Vortrag dieses Seminars hergeleiteten Ausdruck für den quantenmechanischen Propagator,

$$K(q', t; q, 0) = \langle q' | e^{-iHt} | q \rangle, \quad (1)$$

nimmt man zunächst die Ersetzung $t = -i\beta$ vor. Dies entspricht der in vergangenen Vorträgen schon angeklungenen Wickrotation, dem Übergang zu komplexen Zeiten. Die Variable β wurde bewusst in Anlehnung an die statistische Physik gewählt, zumal im hier

gewählten Einheitensystem ($c = \hbar = k_B = 1$) die Zeiteinheit einer inversen Energieeinheit entspricht. Aus (1) folgt somit

$$\begin{aligned} K(q', -i\beta; q, 0) &= \langle q' | e^{-iH(-i\beta)} | q \rangle = \langle q' | e^{-\beta H} \sum_j | j \rangle \langle j | q \rangle = \sum_j e^{-\beta E_j} \langle q' | j \rangle \langle j | q \rangle \\ &= \sum_j e^{-\beta E_j} \langle j | q \rangle \langle q' | j \rangle, \end{aligned} \quad (2)$$

wobei im zweiten Schritt eine Eins in Form eines vollständigen Systems von Eigenzuständen (des Hamiltonoperators) eingefügt wurde. Im letzten Schritt sieht man links schon die kanonische Zustandssumme in der vertrauten Form, die es nun noch gilt, zu isolieren. Das wird erreicht, indem man $q = q'$ setzt und über q integriert:

$$\begin{aligned} \int dq K(q, -i\beta; q, 0) &= \sum_j e^{-\beta E_j} \langle j | \int dq | q \rangle \langle q | j \rangle \\ \Leftrightarrow \int dq \int \mathcal{D}\tilde{q} \exp \left\{ - \int_0^\beta d\tau \left(\frac{m}{2} \left(\frac{d\tilde{q}}{d\tau} \right)^2 + V(\tilde{q}) \right) \right\} &= \sum_j e^{-\beta E_j} = Z \end{aligned} \quad (3)$$

Hier wurde ausgenutzt, dass in der ersten Zeile das Integral über die Ortszustände $|q\rangle$ wiederum eine Eins darstellt und außerdem die Eigenzustände $|j\rangle$ orthonormiert sind. Wesentlich ist, dass man durch die Wickrotation einen reellen Ausdruck erhält. Die einzelnen Pfade werden wie die unterschiedlichen Zustände in der Zustandssumme mit einem Exponentialfaktor gewichtet. Dass dieses Ergebnis richtig ist, kann man am Beispiel des harmonischen Oszillators überprüfen. Dort hatten wir ebenfalls in einem der Vorträge einen Ausdruck für den Propagator gefunden:

$$K(q', t; q, 0) = \left(\frac{m\omega}{2\pi i \sin(\omega t)} \right)^{\frac{1}{2}} \exp \left\{ \frac{im\omega}{2 \sin(\omega t)} ((q'^2 + q^2) \cos(\omega t) - 2q'q) \right\} \quad (4)$$

Mit $t = -i\beta$ folgt

$$K(q, -i\beta; q, 0) = \left(\frac{m\omega}{2\pi \sinh(\beta\omega)} \right)^{\frac{1}{2}} \exp \left\{ - \frac{m\omega q^2}{\sinh(\beta\omega)} (\cosh(\beta\omega) - 1) \right\}, \quad (5)$$

Gleichsetzen von q und q' sowie Integration über q liefern schließlich den bekannten Ausdruck für die Zustandssumme:

$$Z = \int dq K(q, -i\beta; q, 0) = \frac{e^{-\beta\omega/2}}{1 - e^{-\beta\omega}} = \sum_{j=0}^{\infty} e^{-\beta(j+\frac{1}{2})\omega} \quad (6)$$

Das letzte Ergebnis erhält man durch Rückwärtsanwendung der geometrischen Reihe. Nachdem nun also eine Brücke zwischen Pfadintegral und statistischer Physik geschlagen ist, soll von dort ein numerisches Hilfsmittel herangezogen werden, die Monte-Carlo-Simulationsmethode.

3 Einführung in die Monte-Carlo-Methode

3.1 Was ist Monte Carlo?

Unter einer Monte-Carlo-Simulation versteht man die Simulation eines stochastischen Systemverhaltens durch Verwendung von Zufallswerten für die einzelnen Systemparameter. Aus den Zufallswerten werden interessierende Größen berechnet und im Regelfall schließlich über eine Vielzahl solcher Ergebnisse gemittelt.

Diese Idee der “statistischen Probennahme” ist schon viel älter als die Erfindung des Computers. Bereits im 18. Jahrhundert war es offenbar ein verbreiteter Zeitvertreib der Bildungsgesellschaft, durch sogenannte Nadelwurfexperimente die Kreiszahl π möglichst genau zu bestimmen. Man hatte entdeckt, dass eine Nadel der Länge l , wenn sie auf zufällige Art und Weise auf ein mit Linien im Abstand $d \geq l$ versehenes Blatt geworfen wird, mit Wahrscheinlichkeit $2l/\pi d$ eine dieser Linien kreuzt.

Als die ersten Computer zur Verfügung standen, wurde die Simulationsmethode hauptsächlich zur Integration verwendet. Den Namen “Monte Carlo” erhielt sie schließlich, wohl in Anlehnung an das Glücksspielparadies in der gleichnamigen Stadt, vom Physiker N. Metropolis Ende der 1940er Jahre.

3.2 Ein erstes Beispiel

Eine erste einfache Illustration der Methode sei durch das folgende bekannte Beispiel gegeben:

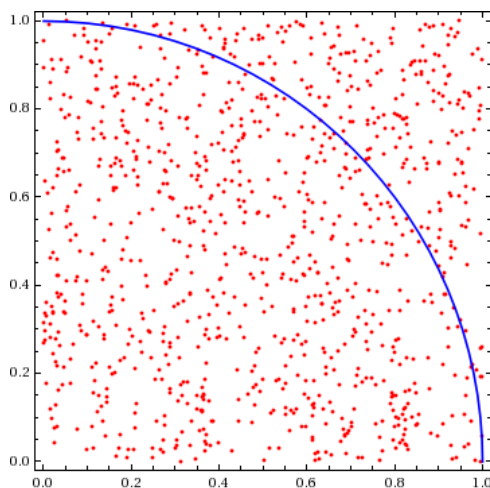


Abbildung 1: Bestimmung von π mithilfe von Zufallszahlen; [5]

Wie in Abbildung 1 zu sehen, legt man ein Viertel eines Einheitskreises in ein Einheitsquadrat. Innerhalb des Quadrats generiert man nun Zufallspunkte und bestimmt deren Abstand vom Ursprung. Ist dieser Abstand ≤ 1 , zählt man den Punkt zu N_i , der Anzahl der Punkte innerhalb des Kreises. Nach ausreichend häufiger Wiederholung erhält man eine mehr oder weniger gute Näherung für die Kreiszahl π , indem man N_i durch die Gesamtzahl der generierten Punkte dividiert:

$$\pi = 4 \cdot \frac{N_i}{N_{\text{ges}}} \quad (7)$$

3.3 Einfache Integration und importance sampling

Nach diesem sehr anschaulichen Beispiel wenden wir uns einer etwas nützlicheren Anwendung der Methode zu, der Integration von Funktionen. Zunächst beschränken wir uns auf den einfachsten Fall nur einer Dimension.

Es sei $f(X)$ Funktion einer kontinuierlichen Zufallsvariablen X . Der Erwartungswert von f ist wie folgt gegeben:

$$E(f(X)) = \int_{-\infty}^{\infty} f(s) p_X(s) ds \quad (8)$$

$p(X)$ ist die Verteilungsfunktion der Variablen X , ihr Integral ist auf 1 normiert. Die Form von (8) erinnert an die Definition des Erwartungswertes von Größen mit diskreter Zufallsvariable, wo einfach das Integral durch eine Summe über alle diskreten Werte ersetzt wird.

Für die Verteilungsfunktion soll zunächst folgende Form angenommen werden:

$$p_X(s) = \begin{cases} \frac{1}{|b-a|} & a \leq s \leq b \\ 0 & \text{sonst} \end{cases} \quad (9)$$

Das bedeutet, dass X jeden Wert auf dem Intervall $[a, b]$ mit der gleichen Wahrscheinlichkeit annimmt, die so gewählt wird, dass die Normierungsbedingung erfüllt ist. Man spricht auch von einer gleichverteilten Zufallsvariable.

Seien nun X_1, X_2, \dots, X_M M Werte der Zufallsvariablen, die mit der obigen Verteilung gewählt wurden. Dann gilt:

$$\langle f \rangle = \frac{1}{M} \sum_{j=1}^M f(X_j) \xrightarrow{M \rightarrow \infty} E(f(X)) = \frac{1}{b-a} \int_a^b f(s) ds \quad (10)$$

Im Grenzfall unendlich vieler generierter Zufallswerte wird also das Intervall $[a, b]$ komplett abgedeckt und der Erwartungswert korrekt angegeben. Da die Wahrscheinlichkeitsverteilung konstant ist, erhält man auch direkt das Integral über f :

$$(b-a) \lim_{M \rightarrow \infty} \langle f \rangle = \int_a^b f(s) ds \quad (11)$$

Ein großer Nachteil des beschriebenen Vorgehens ist, dass es nur für sehr flache Funktionen f effizient ist. Fluktuiert die Funktion jedoch auf einem Teilstück des Intervalls sehr stark, so braucht es an dieser Stelle sehr viele Zufallspunkte, um eine gute Näherung zu erreichen. Werden die Punkte nun nach einer Gleichverteilung gewählt, muss die Gesamtzahl der Zufallspunkte drastisch erhöht werden, um im Bereich starker Fluktuation die gewünschte Dichte zu erreichen. Wünschenswert wäre also eine Verteilung, die gerade an den interessierenden Stellen verstärkt Zufallspunkte liefert. Eine solche Anpassung der Verteilungsfunktion an die zu integrierende Funktion bezeichnet man als *importance sampling*.

In unserem eindimensionalen Beispiel bedeutet das konkret: Wir suchen eine Funktion $g(X) > 0$, die $\int_a^b g(s) ds = 1$ erfüllt und so gewählt sei, dass

$$h(X) \equiv \frac{f(X)}{g(X)} \approx \text{const} \quad (12)$$

Dann kann man schreiben:

$$\int_a^b f(s)ds = \int_a^b h(s)g(s)ds \approx \frac{1}{M} \sum_{j=1}^M h(X_j), \quad (13)$$

wobei die X_j hier mit der Verteilung $g(X)$ ausgewählt wurden. $h(X)$ ist eine flache Funktion und somit das Verfahren deutlich effizienter.

Das Problem an dieser Stelle ist, an Zufallszahlen zu kommen, die einer beliebigen Verteilung gehorchen. Gewöhnliche Zufallszahlgeneratoren geben stets gleichverteilte Werte aus. Zwar gibt es im eindimensionalen Fall für vereinzelte Verteilungen analytische ermittelbare Funktionen, die aus gleichverteilten Werten eine entsprechende Verteilung generieren, sobald man aber in höherdimensionale Gefilde gelangt, sind solche Methoden nicht mehr praktikabel.

3.4 Markov-Ketten

Die oben angestellten Überlegungen lassen sich leicht auf beliebig dimensionierte Integrale übertragen. Man findet im allgemeinen D -dimensionalen Fall

$$\int_V p(\vec{x})f(\vec{x})d^Dx \approx \langle f \rangle \pm \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{M}}. \quad (14)$$

Integriert wird über ein D -dimensionales Volumen V , und im Integranden stehen Funktionen von D -dimensionalen Zufallsvektoren \vec{x} (für unsere Zwecke werden dies später Zufallspfade sein, repräsentiert durch D diskrete Punkte). Diejenigen Zufallsvektoren, die zum Mittelwert $\langle f \rangle$ auf der rechten Seite der Gleichung beitragen, werden gemäß der Verteilung $p(\vec{x})$ ausgewählt. Der Fehler berechnet sich aus der Fluktuation der $f(\vec{x}_j)$ um den Mittelwert.

Wie generiert man nun aber solche Zufallsvektoren, die nach Möglichkeit auch noch einer beliebig gewählten Verteilung entsprechen sollen? Es lässt sich zeigen, dass (14) auch gültig ist, wenn die \vec{x} Folgenglieder eines sogenannten *stationären stochastischen Prozesses* sind. Was damit gemeint ist, sei im Folgenden kurz erläutert. Die diskrete Zeitentwicklung eines Systems sei durch eine Folge von Zuständen $X_t, X_{t+j_1}, X_{t+j_2}, \dots$ beschrieben. Eine solche Entwicklung wird als stationärer stochastischer Prozess bezeichnet, wenn die kumulierte Wahrscheinlichkeit einer Folge von N Zuständen unabhängig vom gewählten Zeitpunkt ist, d. h. $(X_t, X_{t+j_1}, \dots, X_{t+j_n})$ und $(X_{t+h}, X_{t+j_1+h}, \dots, X_{t+j_n+h})$ besitzen bezüglich der Zustände für jedes h die gleiche Häufigkeitsverteilung. Anders ausgedrückt hieße das, dass nach jedem Zeitschritt jedes neue Glied in der Sequenz nach den gleichen Wahrscheinlichkeitsgesetzen ausgewählt wird und damit unabhängig von den vorher durchlaufenen Zuständen ist. Dies führt uns auf den Begriff der *Markov-Ketten*.

Das Prinzip einer Markov-Kette soll an einem kurzen Beispiel erklärt werden. Gegeben sei ein System, das sich in R verschiedenen Zuständen s_1, s_2, \dots, s_R befinden kann. Zwischen diesen Zuständen wechselt das System hin und her, und zwar mit Wahrscheinlichkeit p_{ij} von s_i nach s_j . Die p_{ij} bilden eine $R \times R$ -Matrix, die sogenannte *Markov-Matrix*. In unserem Beispiel ist nun $R = 3$. Eine mögliche Markov-Matrix ist die folgende, hier

gegeben mit ihren drei linksseitigen Eigenvektoren:

$$P = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}, \quad \text{Eigenvektoren} \quad \begin{array}{ll} (2, 3, 2) & \lambda_1 = 1 \\ (-1, 0, 1) & \lambda_2 = \frac{1}{2} \\ (-3, -1, 4) & \lambda_3 = \frac{5}{12} \end{array} \quad (15)$$

Einige wichtige Eigenschaften lassen sich bereits an diesem Beispiel beobachten: Die Summe aller Elemente einer Matrixzeile ist 1, also

$$\sum_j p_{ij} \stackrel{!}{=} 1, \quad (16)$$

da sich das System nach dem Übergang ja auf jeden Fall in einem der Zustände (den Ausgangszustand eingeschlossen) befinden muss. Da die p_{ij} Wahrscheinlichkeiten sind, muss

$$0 \leq p_{ij} \leq 1 \quad (17)$$

gelten. Weiter lässt sich zeigen, dass der Betrag aller Eigenwerte immer kleiner gleich 1 ist, und dass in jedem Fall eine Eigenvektor mit dem Eigenwert 1 existiert, den man auch als invarianten Eigenvektor bezeichnet.

Die Markov-Matrix P beschreibt die (zeitdiskrete) Entwicklung eines Wahrscheinlichkeitsvektors w , dessen Komponenten w_i wiederum angeben, mit welcher Wahrscheinlichkeit sich das System im Zustand s_i befindet. Exemplarisch sei die Entwicklung für einen Vektor angegeben, der ein System beschreibt, dass sich zunächst mit Wahrscheinlichkeit 1 im Zustand s_1 befindet:

$$(1, 0, 0) \rightarrow \left(\frac{3}{4}, \frac{1}{4}, 0\right) \rightarrow \left(\frac{9}{16}, \frac{17}{48}, \frac{1}{12}\right) \rightarrow \dots \rightarrow \left(\frac{2}{7}, \frac{3}{7}, \frac{2}{7}\right) \quad (18)$$

Nach einmaligem Anwenden der Markov-Matrix befindet sich das System nur noch zu $\frac{3}{4}$ in s_1 , nach mehrmaligem Anwenden kann es auch in den Zustand s_3 gewechselt sein. Durch Ausführung unendlich vieler Iterationen erhält man den invarianten Einheitsvektor (in normierter Form). Das geschieht unabhängig vom Ausgangszustand, weil Anteile der anderen Eigenvektoren aufgrund des Eigenwertes, der kleiner als 1 ist, dann auf Null abgefallen sind. Die n -fache Iteration P^n bekommt im Grenzfall die Form

$$\lim_{n \rightarrow \infty} P^n = \frac{1}{7} \begin{pmatrix} 2 & 3 & 2 \\ 2 & 3 & 2 \\ 2 & 3 & 2 \end{pmatrix}. \quad (19)$$

Diese Matrix, angewendet auf einen beliebigen Wahrscheinlichkeitsvektor, ergibt tatsächlich stets den invarianten Eigenvektor, und mit jeder weiteren Iteration wird dieser Vektor zurückerhalten. Das System befindet sich also in einem Gleichgewicht, und jeder neue Zustand des Systems wird nach der gleichen Verteilung $\left(\frac{2}{7}, \frac{3}{7}, \frac{2}{7}\right)$ ausgewählt. Somit haben wir einen stationären stochastischen Prozess vorliegen und können die Glieder der Markov-Kette schon fast für unsere Monte-Carlo-Integration nach (14) verwenden. Es müssen allerdings noch ein paar weitere Anforderungen an die Markov-Kette gestellt werden:

- *Positive Rekurrenz*: Darunter versteht man, dass das System mit Wahrscheinlichkeit 1 nach endlich vielen Schritten in einen Ausgangszustand zurückkehrt. So ist garantiert, dass es überhaupt so etwas wie einen invarianten Eigenvektor geben kann.

- *Irreduzibilität*: Jeder Zustand s_j ist von jedem Zustand s_i aus nach endlicher Schrittzahl erreichbar. Dann ist der invariante Eigenvektor eindeutig. Diese ersten beiden Punkte fasst man auch unter dem Begriff *Ergodizität* zusammen.
- *Aperiodizität*: Es dürfen keine zyklischen Gleichgewichte auftreten, so dass die Folge wirklich gegen den einzigen invarianten Eigenvektor konvergiert.
- *Reversibilität*: Einmal im Gleichgewicht angelangt, muss die Wahrscheinlichkeit für einen Übergang von s_i nach s_j genauso wahrscheinlich wie der umgekehrte Übergang sein:

$$w_i p_{ij} \stackrel{!}{=} w_j p_{ji} \quad (20)$$

(Die Wahrscheinlichkeit, das System in s_i zu finden (w_i), multipliziert mit der Wahrscheinlichkeit für den Übergang zu s_j , p_{ij} , ist gleich der Wahrscheinlichkeit für den umgekehrten Fall) Diese Eigenschaft wird in der Fachsprache auch häufig als “*detailed balance*” bezeichnet.

Sind alle diese Bedingungen erfüllt, können wir analog zu (14) das Integral approximieren:

$$\int_V p(\vec{x}) f(\vec{x}) d^D x \approx \langle f \rangle \pm \sqrt{\frac{R(f)}{M}} \quad (21)$$

Die einzelnen Glieder \vec{x} der Markov-Kette sind in der Regel nicht völlig stochastisch unabhängig, sondern mehr oder weniger stark korreliert, was zu einem zusätzlichen Fehler bei der Integration führt. Die Fehlerangabe in (21) erfolgt daher durch eine Korrelationsfunktion $R(f)$, auf deren genaue Gestalt wir hier aber nicht eingehen wollen.

Unser Ziel ist nun also, eine Markov-Kette zu finden, die uns mit jedem Schritt einen Zufallspfad liefert. Die Auswahl der Pfade soll nach einer bestimmten Wahrscheinlichkeitsverteilung $p(\vec{x})$ erfolgen. Die nächste Frage wäre demnach, wie dieses $p(\vec{x})$ im Einzelnen aussehen soll.

3.5 Schätzfunktionen

Aus der statistischen Physik kennen wir die Berechnung von Erwartungswerten von Observablen wie Energie, Wärmekapazität oder Entropie. Diese ergeben sich im Allgemeinen in irgendeiner Form aus der kanonischen Zustandssumme. Es gilt:

$$\langle Q \rangle = \frac{\sum_j Q_j e^{-\beta E_j}}{\sum_k e^{-\beta E_k}} \quad (22)$$

Der Wert der Observablen Q in jedem Zustand der Zustandssumme, charakterisiert durch die Energie E_j , wird mit der entsprechenden Exponentialfunktion gewichtet, und der gesamte Term durch die Summe im Nenner normiert. In Pfadintegralschreibweise gilt analog

$$\langle Q \rangle = \frac{\int \mathcal{D}\phi Q[\phi] e^{-S[\phi]}}{\int \mathcal{D}\phi' e^{-S[\phi']}}. \quad (23)$$

Um den genauen Mittelwert zu bestimmen, müsste also die Observable für alle möglichen Systemzustände berechnet werden. Allerdings wird deren Anzahl schon für ein Vielteilchensystem mit wenigen Freiheitsgraden ungeheuer groß, und man muss sich darauf

beschränken, die Mittelung über einen kleinen Teil dieser Zustände durchzuführen. Dies geschieht mithilfe einer Schätzfunktion:

$$Q_M = \frac{\sum_{j=1}^M Q_j w_j^{-1} e^{-\beta E_j}}{\sum_{k=1}^M w_k^{-1} e^{-\beta E_k}} \quad (24)$$

Die Auswahl der Zustände erfolgt gemäß der Verteilung w_i , das sind die Komponenten der Wahrscheinlichkeitsvektoren aus dem vorherigen Abschnitt.

Wählen wir eine Gleichverteilung, d. h. alle w_i bekommen denselben Wert, so reduziert sich die Schätzfunktion zwar auf

$$Q_M = \frac{\sum_{j=1}^M Q_j e^{-\beta E_j}}{\sum_{k=1}^M e^{-\beta E_k}}, \quad (25)$$

es besteht aber das bereits angesprochene Problem, dass dieses Verfahren sehr ineffizient ist. Die Zahl der möglichen Zustände kann sehr groß werden, während sich das System eigentlich nur in einem verschwindend kleinen Bruchteil dieser Zustände wirklich befindet, und das sind diejenigen mit der niedrigsten Energie. Sinnvoll wäre demzufolge, solche Zustände mit besonderer großer Wahrscheinlichkeit auszuwählen. Dieses Prinzip haben wir bereits unter dem Namen “importance sampling” kennengelernt. Setzt man

$$w_i = \frac{e^{-\beta E_i}}{Z} \quad \text{bzw.} \quad p_\phi = \frac{e^{-S[\phi]}}{\int \mathcal{D}\phi' e^{-S[\phi']}}, \quad (26)$$

ergibt sich die Schätzfunktion einfach zu

$$Q_M = \frac{1}{M} \sum_{i=1}^M Q_i \quad (27)$$

Für eine effiziente Integration müssen wir also die Zufallspfade nach einer Boltzmannverteilung auswählen. Im folgenden Abschnitt soll ein Algorithmus vorgestellt werden, mit dessen Hilfe eine Markov-Kette mit entsprechender Gleichgewichtsverteilung realisiert werden kann, der sogenannte *Metropolis-Hastings-Algorithmus*.

3.6 Der Metropolis-Hastings-Algorithmus

Ein wichtiges Kriterium für unsere Markov-Kette war die Erfüllung von (20), auch als “detailed balance” bezeichnet. Die dort auftretenden w_i sollen die Boltzmannverteilung darstellen, wie in (26) festgelegt. Ein Algorithmus muss nun so konstruiert werden, dass die Übergangswahrscheinlichkeiten folgender Bedingung genügen:

$$\frac{p_{ij}}{p_{ji}} \stackrel{(20)}{=} \frac{w_j}{w_i} \stackrel{(26)}{=} e^{-\beta(E_j - E_i)} \quad (28)$$

Dazu behilft man sich mit folgendem Trick: Man teilt die Übergangswahrscheinlichkeit p_{ij} auf in die Wahrscheinlichkeit $g(i \rightarrow j)$, dass ein Übergang von s_i nach s_j vorgeschlagen wird, multipliziert mit der Wahrscheinlichkeit $A(i \rightarrow j)$ dafür, dass dieser Vorschlag akzeptiert wird. Dann ergibt sich mit der vorherigen Gleichung:

$$\frac{p_{ij}}{p_{ji}} = \frac{g(i \rightarrow j) A(i \rightarrow j)}{g(j \rightarrow i) A(j \rightarrow i)} \quad (29)$$

Die Vorschlagswahrscheinlichkeit g soll völlig gleichförmig gewählt werden. Das bedeutet, dass jeder Übergang mit gleicher Wahrscheinlichkeit vorgeschlagen werde, und insbesondere sei $g(i \rightarrow j) = g(j \rightarrow i)$. Gleichung (29) vereinfacht sich somit zu

$$\frac{p_{ij}}{p_{ji}} = \frac{A(i \rightarrow j)}{A(j \rightarrow i)} \stackrel{!}{=} e^{-\beta(E_j - E_i)} \quad (30)$$

Bei der genauen Form der Akzeptanzwahrscheinlichkeiten A besteht noch Wahlfreiheit. Je nach Wahl erhält man unterschiedliche Algorithmen. Zwei einfache Möglichkeiten sollen an dieser Stelle verglichen werden:

$$1) \quad A(i \rightarrow j) = A_0 e^{-\frac{1}{2}\beta(E_j - E_i)} \quad (31)$$

$$2) \quad A(i \rightarrow j) = \begin{cases} e^{-\beta(E_j - E_i)} & \text{falls } E_j > E_i \\ 1 & \text{sonst} \end{cases} \quad (32)$$

Beide Möglichkeiten erfüllen ganz offensichtlich die Bedingung (30). Im ersten Vorschlag ist die Konstante A_0 im Prinzip frei wählbar, mit der Einschränkung, dass der Gesamtausdruck für A , der ja eine Wahrscheinlichkeit darstellt, immer kleiner oder gleich 1 sein sollte. Der zweite Vorschlag bedeutet einfach, dass vorgeschlagene Übergänge in Zustände niedrigerer Energie grundsätzlich akzeptiert werden, während bei Erhöhung der Energie der neue Zustand nur mit geringerer Wahrscheinlichkeit angenommen wird. Diese Wahl wird letztlich Metropolis-Hastings-Algorithmus genannt.

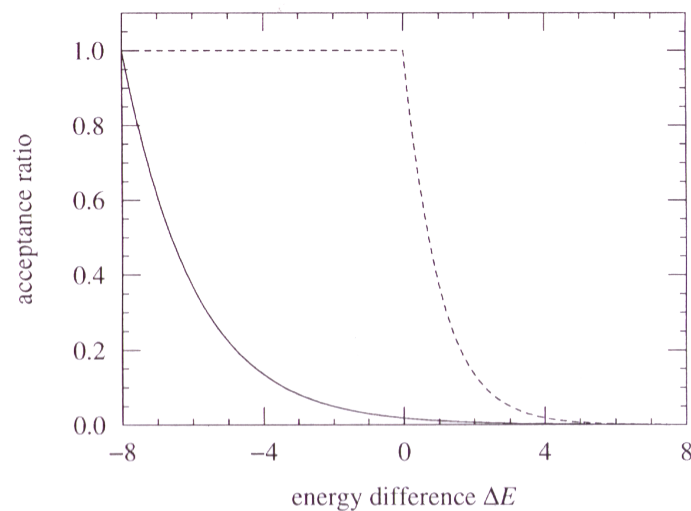


Abbildung 2: Die Akzeptanzwahrscheinlichkeiten (31) und (32) (gestrichelt) im Vergleich.

In Abbildung 2 werden die Akzeptanzwahrscheinlichkeiten beider Algorithmen in Abhängigkeit von der Energiedifferenz dargestellt. Erstrebenswert ist in jedem Falle eine hohe Akzeptanzrate auch noch bei positiven Energiedifferenzen, damit die verschiedenen Zustände des Systems schnell durchlaufen werden. Wie zu sehen ist, bietet in diesem Vergleich der Metropolis-Hastings-Algorithmus (gestrichelte Linie) beträchtliche Vorteile. Nach so vielen abstrakten Konzepten soll nun am Beispiel des harmonischen Oszillators eine konkrete Umsetzung dieses Algorithmus vorgestellt werden.

4 Monte-Carlo-Simulation des harmonischen Oszillators

4.1 Der MH-Algorithmus für den HO

Die euklidische Wirkung für den harmonischen Oszillator lautet, wie im entsprechenden Vortrag bereits gesehen,

$$S[x(\beta)] = \int_0^\beta d\tau \left(\frac{1}{2} m \dot{x}^2 + \frac{1}{2} m \omega^2 x^2 \right). \quad (33)$$

Die Zufallsvektoren in Gleichung (21) sind in unserem Fall diskretisierte Pfade. Das euklidische Zeitintervall der Länge β werde in N Teilintervalle der Länge ϵ unterteilt, also $\beta = N\epsilon$. Dann ist jeder Zufallspfad ein $N+1$ -komponentiger Vektor mit den Komponenten x_0, x_1, \dots, x_N , und die euklidische Wirkung berechnet sich gemäß

$$S = \frac{\epsilon}{2} \sum_{j=0}^{N-1} \left[\left(\frac{x_{j+1} - x_j}{\epsilon} \right)^2 + \omega^2 \left(\frac{x_{j+1} + x_j}{2} \right)^2 \right]. \quad (34)$$

Hier wurde zusätzlich zu den eingangs besprochenen Konventionen $m = 1$ gesetzt, für die potenzielle Energie wird jeweils über zwei benachbarte Diskretisierungsstellen x_j und x_{j+1} gemittelt. Der Übersicht halber führt man noch dimensionslose Koordinaten ein. Mit

$$u_j = \sqrt{\frac{4 + \omega^2 \epsilon^2}{\epsilon}} x_j \quad \text{und} \quad g = \frac{4 - \omega^2 \epsilon^2}{4 + \omega^2 \epsilon^2} \quad (35)$$

wird (34) zu

$$S = \frac{1}{2} (u_0^2 + u_N^2) + \sum_{j=1}^{N-1} u_j^2 - g \sum_{j=0}^{N-1} u_j u_{j+1}. \quad (36)$$

In dieser konkreten Anwendung könnte der Metropolis-Hastings-Algorithmus wie folgt aussehen:

1. Wähle einen zufälligen Punkt u_j für $j = 1, \dots, N$ des aktuellen Pfades aus.
2. Schlage eine beliebige Verschiebung dieses Punktes $u_j \rightarrow \tilde{u}_j = u_j + \delta$ vor, wobei δ ein Zufallswert aus dem Intervall $[-\Delta, \Delta]$ sei.
3. Berechne die Änderung in der Wirkung $\tilde{S} - S$, die diese Verschiebung verursachen würde:

$$\tilde{S} - S = \delta(\delta + 2u_j - g(u_{j-1} + u_{j+1})) \quad (37)$$

4. Akzeptiere die neue Konfiguration mit der Wahrscheinlichkeit

$$A(u \rightarrow \tilde{u}) = \begin{cases} e^{-\beta(\tilde{E}-E)} = e^{-(\tilde{S}-S)} & \text{falls } \tilde{S} > S \\ 1 & \text{sonst.} \end{cases} \quad (38)$$

Wird die Verschiebung akzeptiert, setze $u_j^{\text{neu}} = \tilde{u}_j$, ansonsten $u_j^{\text{neu}} = u_j$.

5. Wiederhole die Schritte 1 bis 4 N -mal, um einen "sweep", d. h. ein vollständiges Update eines Pfades, zu realisieren.

Ein wichtiger Parameter dieses Algorithmus ist die Größe des Intervalls $[-\Delta, \Delta]$, aus dem die zufällige Verschiebung gewählt wird. Ist Δ zu groß, werden zu häufig große Verschiebungen vorgeschlagen, die zu großen Energieänderungen führen und daher meist abgelehnt werden. In diesem Fall wird viel Rechenzeit durch zurückgewiesene Schritte verschwendet. Wählt man dagegen Δ zu klein, werden die kleinen Änderungen zwar fast alle angenommen, der Raum der unterschiedlichen Pfade wird allerdings nur sehr langsam durchlaufen. Idealerweise sollte eine Akzeptanzrate von etwa 50% vorliegen.

Weiterhin ist es wichtig, zu berücksichtigen, dass zwischen den Pfaden zweier aufeinanderfolgender sweeps starke Korrelationen bestehen. Infolgedessen wird der Fehler, wie in (21) angegeben, relativ groß. Diese Korrelationseffekte kann man senken, indem man nicht jedes Update für die Auswertung des Integrales benutzt, sondern nach einem verwendeten Pfad immer erst eine bestimmte Anzahl an sweeps abwartet. Die Autokorrelationsfunktion einer beliebigen Observablen O , ausgewertet für einen Pfad i , ist gegeben durch

$$\rho_O(\tau) = \frac{\langle O_i O_{i+\tau} \rangle - \langle O_i \rangle^2}{\langle O_i^2 \rangle - \langle O_i \rangle^2}. \quad (39)$$

In diesem Fall ist τ die Anzahl an sweeps, die zwischen den beiden Auswertungen liegt. Die Implementierung des Algorithmus in ein C++-Programm ermöglichte die Berechnung der Autokorrelation der Observablen $\langle x_0 x_5 \rangle$, das Ergebnis ist in Abbildung 3 graphisch dargestellt. Wie man sieht, sollte die Anzahl der Updates zwischen zwei für die Auswertung verwendeten Pfaden mindestens 200 betragen, um Autokorrelationen unter einen Wert von etwa 0,1 zu senken.

Eine weitere Möglichkeit, die Korrelationen zu veranschaulichen, ist Abbildung 4. Dort sind jeweils drei Zufallspfade abgebildet, wobei für die Zahl der Diskretisierungspunkte 200 gewählt wurde. Die oberen drei Pfade sind direkt aufeinanderfolgende Updates, die sehr korreliert, sich also noch sehr ähnlich sind. Zwischen den unteren drei Pfaden liegen hingegen jeweils 200 sweeps, die nicht berücksichtigt wurden. Man erkennt deutlich, dass sie allenfalls noch schwach korreliert sind. Ebenfalls zu sehen ist, dass Anfangs- und Endpunkt des Pfades übereinstimmen, so, wie es eingangs für den Zusammenhang zwischen Pfadintegral und Zustandssumme gefordert worden war.

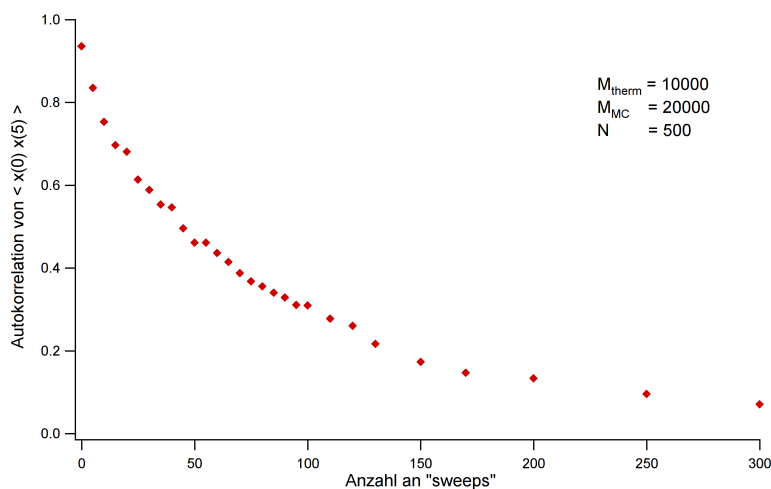


Abbildung 3: Autokorrelation der Observablen $\langle x_0 x_5 \rangle$.

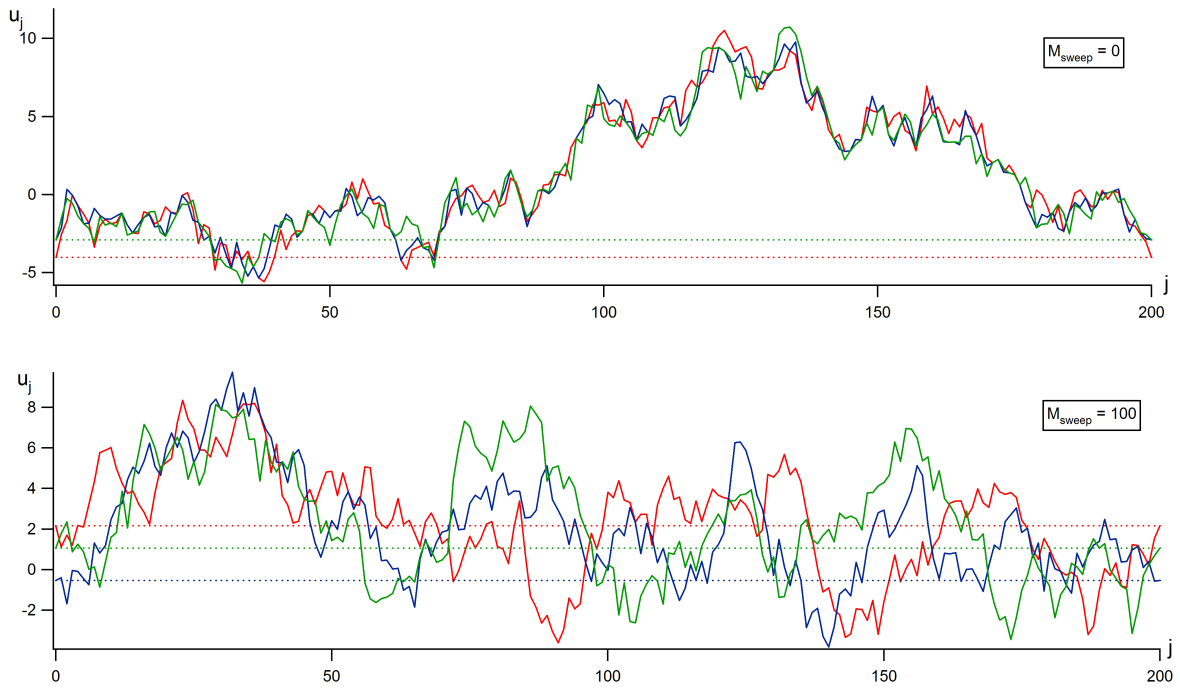


Abbildung 4: Zur Veranschaulichung der Korrelation: Zufallspfade mit $N = 200$

4.2 Monte-Carlo-Messungen der mittleren Energie

Nachdem wir nun über Boltzmann-verteilte, ausreichend dekorrelierte Zufallspfade verfügen, soll eine sogenannte Messung durchgeführt werden. Das bedeutet, dass gemäß (27) eine bestimmte Observable für jeden unserer Pfade berechnet und nach M Berechnungen gemittelt wird. Die Observable soll in diesem Fall die mittlere Energie des harmonischen Oszillators sein. Aus der statistischen Physik ist der Zusammenhang mit der Zustandssumme sowie das exakte Ergebnis bekannt:

$$\langle H \rangle = -\frac{\partial \ln Z}{\partial \beta} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = \frac{1}{2} \coth\left(\frac{1}{2}\omega\beta\right) \quad (40)$$

Im diskretisierten Fall wird aus der Ableitung nach β eine Ableitung nach ϵ :

$$\beta = \epsilon N \Rightarrow \frac{\partial}{\partial \beta} Z = \frac{1}{N} \frac{\partial}{\partial \epsilon} Z \quad (41)$$

An dieser Stelle wählen wir die Darstellung von Z als Pfadintegral. Da bei der Berechnung von Erwartungswerten die genaue Wahl von Anfangs- und Endpunkt keine Rolle spielt, wird auf die Integration über q , wie sie in (3) eingeführt wurde, verzichtet. Im Rahmen der Updates wird $u_0 = u_N$ ohnehin von Zeit zu Zeit geändert. Es gilt somit:

$$\langle H_T \rangle = \frac{\int \left(-\frac{1}{N} \frac{\partial}{\partial \epsilon}\right) \mathcal{D}\phi e^{-S[\phi]}}{\int \mathcal{D}\phi' e^{-S[\phi']}} = \frac{\int \mathcal{D}\phi H_T[\phi] e^{-S[\phi]}}{\int \mathcal{D}\phi' e^{-S[\phi']}} \quad (42)$$

$H_T[\phi]$ wird auch thermodynamische Schätzfunktion für die Energie genannt. Bei ihrer Berechnung muss beachtet werden, dass auch das Differential $\mathcal{D}\phi$ vom Diskretisierungsparameter ϵ abhängt. Mit

$$\mathcal{D}\phi = \left(\frac{1}{2\pi\epsilon}\right)^{\frac{N}{2}} dx_1 dx_2 \cdots dx_{N-1} \quad (43)$$

und der Gleichung (34) für die Wirkung ergibt sich für die Schätzfunktion:

$$H_T = \frac{1}{2\epsilon} - \frac{1}{2N} \sum_{j=0}^{N-1} \left[\left(\frac{x_{j+1} - x_j}{\epsilon} \right)^2 - \left(\frac{x_{j+1} + x_j}{2} \right)^2 \right] \quad (44)$$

Dieser Ausdruck wird bei jeder Messung für den entsprechenden Pfad ermittelt und der eigentliche Schätzwert wie besprochen erhalten:

$$\langle H_T \rangle = \frac{1}{M} \sum_{i=1}^M H_{T,i} \quad (45)$$

Ein Problem der soeben hergeleiteten Schätzfunktion ist, dass der zweite Term in (44), der kinetische Term, stark fluktuiert. Die Fluktuationen werden bei feiner werdender Diskretisierung immer größer; man kann zeigen, dass $\sigma_{T,N}$ proportional zu \sqrt{N} wächst. Dies ist eine sehr ungünstige Tatsache, da zunächst einmal N möglichst groß gewählt werden soll, um den kontinuierlichen Fall gut zu approximieren. Abhilfe schafft das Virialtheorem, das noch aus dem ersten Semester bekannt sein dürfte:

$$\langle H_V \rangle = \langle T \rangle + \langle V \rangle = 2 \langle V \rangle \quad (46)$$

Der Mittelwert der kinetischen Energie lässt sich auch nur durch den der potentiellen Energie ausdrücken. Daraus resultiert letztlich die sogenannte viriale Schätzfunktion,

$$H_V = \frac{1}{N} \omega^2 \sum_{j=0}^{N-1} x_j^2. \quad (47)$$

Von der Mittelwertbildung zwischen x_j und x_{j+1} wurde an dieser Stelle abgesehen. Der mit dieser Funktion berechnete Schätzwert hat den Vorteil, dass die Standardabweichung σ_V nur schwach von N abhängt, der stark fluktuierende kinetische Term wurde eliminiert. Abbildung 5 zeigt Messergebnisse der beiden Schätzfunktionen bei unterschiedlichen Temperaturen, wobei jeweils für N verschiedene Werte zwischen 10 und 1000 gewählt wurden, um die Entwicklung der Fluktuationen zu untersuchen. M_{therm} bezeichnet die Anzahl der sweeps, die zum Thermalisieren der Markov-Kette verwendet wurden. Dabei handelt es sich in diesem Fall um 1000 ‘‘Leerlaufsweps’’, in denen der Algorithmus von der beliebigen Anfangskonfiguration aus das Gleichgewicht erreichen konnte, bevor die eigentliche Messung begann. Die Anzahl der Messungen betrug dann jeweils $M_{\text{MC}} = 40000$. Klar zu erkennen ist in beiden Fällen, dass die Fluktuationen des thermodynamischen Schätzwertes mit wachsendem N zunehmen, während sich der Fehler beim virialen Schätzwert kaum ändert. Auch beobachtet man, dass für die höhere Temperatur die Fluktuationen deutlich stärker sind.

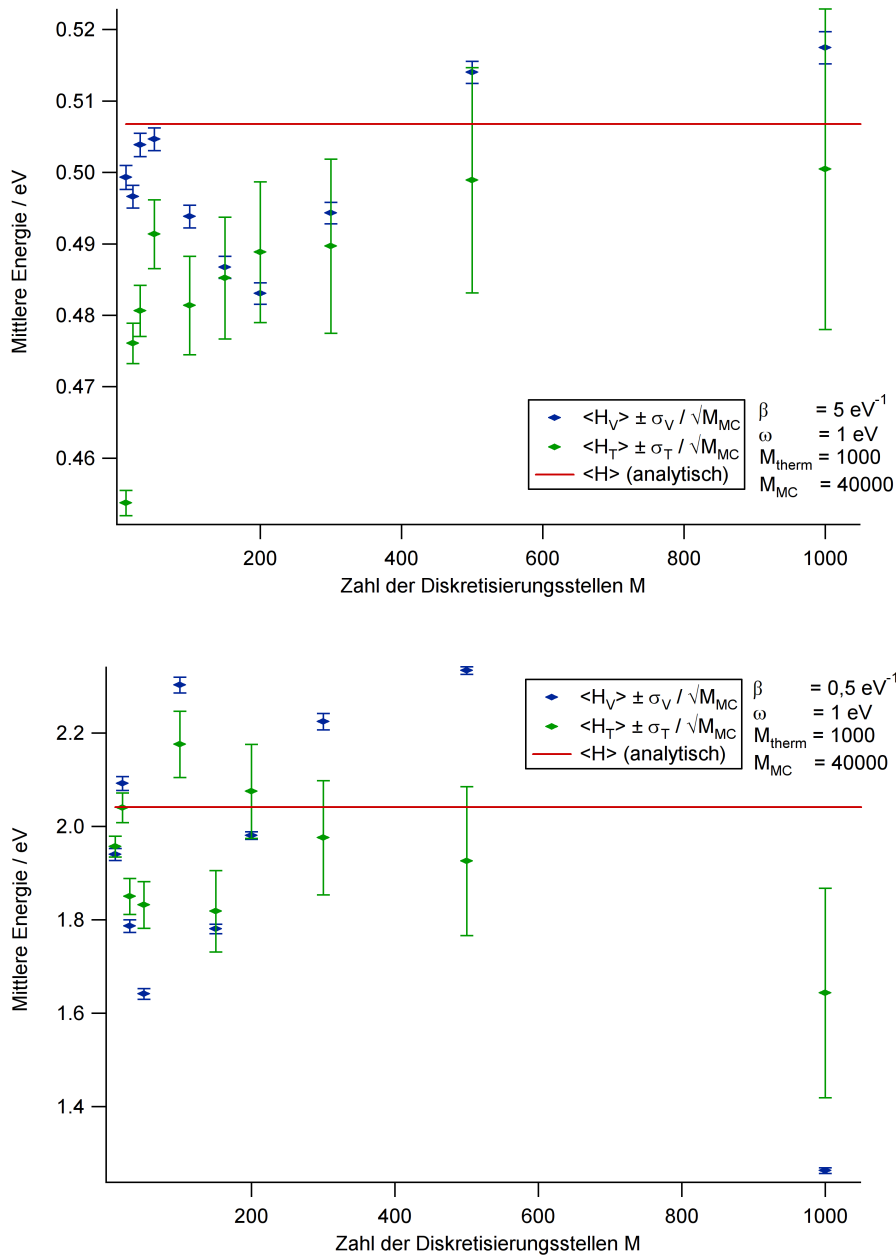


Abbildung 5: Messung der mittleren Energie mit thermodynamischer und virialer Schätzfunktion.

Die mitunter starken Abweichungen vom analytischen Ergebnis könnten der Tatsache geschuldet sein, dass diese Art der Zufallspfadgeneration sehr stark oszillierende Pfade hervorbringt. Alternative Ansätze zur Konstruktion glatter Pfade findet man beispielsweise in [7].

Deutlich weniger Fluktuation erhält man beispielsweise bei der Messung der Korrelationsfunktion $\langle x_j x_{j+\tau} \rangle$, wie in Abbildung 6 zu sehen ist. Für x_j wurden gleich mehrere Punkte aus der Pfadmitte verwendet, um die Statistik zu erhöhen. τ ist in Einheiten des Diskretisierungsparameters ϵ aufgetragen.

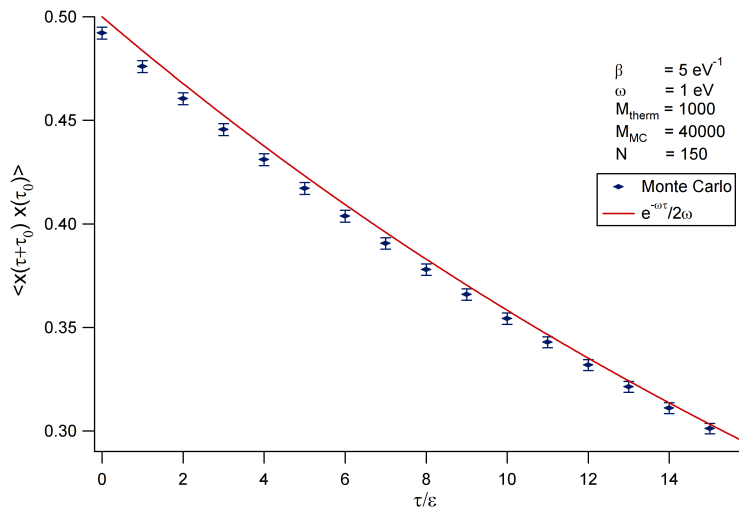


Abbildung 6: Messung der Korrelationsfunktion $\langle x_j x_{j+\tau} \rangle$.

Die Übereinstimmung mit der analytischen Kurve ist hier durchaus besser als bei den Energiemessungen, gleichwohl bleibt eine stete Abweichung nach unten. Möglicherweise hängt auch dies mit der etwas “zackigen” Gestalt der Zufallspfade zusammen, ebenso ist nicht auszuschließen, dass das verwendete Programm noch ein Fehler beinhaltet.

Abschließend bleibt zu sagen, dass der vorgestellte Metropolis-Hastings-Algorithmus in der aktuellen Forschung kaum noch Verwendung findet, weil er mit dem Nachteil der langen Korrelationszeiten behaftet ist. Stattdessen wurden, wieder ausgehend von der Forderung (20) der “detailed balance”, andere Algorithmen entwickelt, die in (29) nicht nur an der Akzeptanzwahrscheinlichkeit, sondern auch an der hier als gleichmäßig angenommenen Vorschlagswahrscheinlichkeit g drehen und so zu noch angepassteren und effizienteren Verfahren gelangen. Als Beispiel sei der sogenannte Cluster-Algorithmus genannt, der aktuell vielfach benutzt wird.

Literatur

- [1] R. MacKenzie, *Path Integral Methods and Applications*, arXiv:quant-ph/0004090v1 24 Apr 2000
- [2] C. Morningstar, *The Monte Carlo Method in Quantum Field Theory*, arXiv:hep-lat/0702020v1 20 Feb 2007
- [3] M. Newman, G. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford University Press, Korrigierte Fassung, 2002
- [4] M. Herman, E. Bruskin, B. Berne, *On path integral Monte Carlo simulations*, J. Chem. Phys. 76, 5150 (1982)
- [5] <http://physical-thought.blogspot.de>, 08.07.2012
- [6] L. Brualla i Barberà, *Path integral Monte Carlo - Algorithms and applications to quantum fluids*, PhD thesis, Universitat Politècnica de Catalunya, Barcelona, 2002
- [7] T. Sekihara, *New Approach to Continuum Path Integrals for Particles and Fields*, arXiv:1201.0055v3 [quant-ph] 19 Jun 2012