

Kapitel 3

Monte–Carlo–Methoden

Historisch wird der Begriff der „Monte–Carlo–Methode“ 1947 geprägt [38]¹ und zum ersten Mal zwei Jahre später im Titel einer Veröffentlichung verwendet [39]. Wie der Name anklingen lässt, können die Regeln der Monte–Carlo–Methoden als ein Modell für Zufallsbewegungen - erzeugt mit Hilfe von Zufallszahlen - betrachtet werden. Die einzelnen Berechnungsschritte hängen von der explizit verwendeten Methode ab. Beinhalten hierbei diese Berechnungsschritte Operationen von einfacher, fundamentaler Natur, so sind sie geeignet, von herkömmlichen Rechenmaschinen ausgeführt zu werden. Deren Stärke liegt nämlich darin, dass sie fundamentale Rechenoperationen und logische Operationen schnell und wiederholt ausführen können. Es ist somit nicht verwunderlich, dass nicht nur mit dem Bau der ersten „modernen Rechenmaschinen“ [39] Monte–Carlo entwickelt wird, sondern dass es gerade die Entwicklungen in der Computer–Technologie unserer Zeit sind, die den Methoden ihren Aufschwung zur Standard–Methode verleihen.

In diesem Kapitel werden die Methoden der Monte–Carlo–Simulation vorgestellt. In der einführenden Literatur [40, 41, 42] geschieht dies fast durchwegs anhand des Ising–Modells als Beispiel. Da in der Folge dieses Modell für uns keine weitere Bedeutung besitzt, wählen wir von Anfang an einen formalen Weg.

Wir beginnen mit der Methode des Simple–Sampling, die erste wichtige Ideen und grundlegende Prinzipien offenlegt. Diese Methode kann durch das Importance–Sampling für bestimmte Problemstellungen verbessert werden, welches zugleich die Basis für den Metropolis–Algorithmus bildet. Die Einführung dreier nützlicher Tricks für Monte–Carlo–Simulationen beendet dieses Kapitel, wobei noch auf die Kapitel des Anhangs verwiesen sei. In Anhang B wenden wir uns der Erzeugung von Zufallszahlen zu, in Anhang C der Theorie der Markov–Prozesse und in Anhang D machen wir uns Gedanken zum Thema der Fehlerrechnung und zu Fluktuationen.

Die Referenzen dieses Kapitels beziehen sich, wo nicht explizit anders angegeben, auf [38, 40, 41, 42].

3.1 Simple–Sampling

Ein Ziel der Statistischen Physik besteht darin, für einen Modell–Hamiltonian H die Mittelwerte gewünschter Observablen (z.B. der Energie E des betrachteten Systems) zu berechnen.

¹Der Artikel, der in erster Linie die grundlegenden Ideen der Methoden qualitativ umreißt, bezieht sich in seinem physikalischen Kern hauptsächlich auf Simulationsmöglichkeiten aus dem Gebiet der Teilchenphysik.

Zur Bestimmung des Mittelwertes der beliebigen Observablen $O(\pi)$ dient Formel

$$\langle O \rangle = \frac{1}{Z} \int_{\Gamma} d\Gamma \cdot O(\pi) \exp[-\beta H(\pi)] , \quad (3.1)$$

mit

$$Z = \int_{\Gamma} d\Gamma \cdot \exp[-\beta H(\pi)] \quad \text{und} \quad \beta = \frac{1}{kT} ,$$

wobei k die Boltzmann-Konstante und T die Temperatur ist. In Gleichung (3.1) steht π für einen Punkt im Phasenraum Γ und $d\Gamma = \prod_{i=1}^N d\pi_i$ für ein Phasenraum-Volumenelement, wobei N die Anzahl möglicher Freiheitsgrade des Systems angibt. Schreiben wir Gleichung (3.1) explizit aus, so folgt

$$\langle O \rangle = \frac{1}{Z} \underbrace{\int_{\Gamma} d\pi_N \cdots \int d\pi_1}_{\Gamma} \cdot O(\pi_1, \dots, \pi_N) \exp[-\beta H(\pi_1, \dots, \pi_N)] , \quad (3.2)$$

mit der Zustandssumme

$$Z = \underbrace{\int_{\Gamma} d\pi_N \cdots \int d\pi_1}_{\Gamma} \cdot \exp[-\beta H(\pi_1, \dots, \pi_N)] .$$

Die Anzahl möglicher Freiheitsgrade N liegt bei realen Systemen in der Regel bei Grössenordnungen von mehreren Zehnerpotenzen,² wodurch es unmöglich wird, die enorme Anzahl der entstehenden Integrale in Gleichung (3.2) exakt zu lösen. Es muss auf eine Näherung zurückgegriffen werden.

Die Idee der Monte-Carlo-Methoden ist es, Gleichung (3.1) durch den Ausdruck

$$\langle O \rangle \approx \bar{O} = \frac{\sum_{l=1}^M O(\pi_l) \exp[-\beta H(\pi_l)]}{\sum_{l=1}^M \exp[-\beta H(\pi_l)]} \quad (3.3)$$

zu nähern, wobei die Menge $\{\pi_1, \dots, \pi_M\}$ eine charakteristische Stichprobe aller Zustände des Phasenraumes Γ ist. Es sei betont, dass diese charakteristische Stichprobe aus Phasenraumelementen bestehen soll, die zufällig ausgewählt werden. Der entscheidende Vorteil der Monte-Carlo-Vorgehensweise im Vergleich zum exakten Ausintegrieren von (3.2) ist, dass die Anzahl M der Elemente der Stichprobe klein ist verglichen mit der Anzahl aller möglichen Konfigurationen. Dadurch wird Formel (3.3) handhabbar. Im Grenzfall $M \rightarrow \infty$ geht die Näherung (3.3) in (3.1) über, genau wie in der numerischen, eindimensionalen Integration.³

Wir sind nun in der Lage, den Algorithmus des Simple-Sampling, einer Variante der Monte-Carlo-Methoden zu formulieren. Er lautet [38]

²Betrachten wir 1mol eines idealen Gases, so beträgt die Anzahl Teilchen $N_A \approx 6 \cdot 10^{23}$. Besitzt jedes Teilchen die drei Freiheitsgrade des Ortes und die drei Freiheitsgrade des Impulses, so hat das ganze System $N = 6 \cdot N_A \approx 3.6 \cdot 10^{24}$ Freiheitsgrade.

³Für den Fall eindimensionaler Integrale ist die Monte-Carlo-Technik schwächer als direkte numerische Verfahren (z.B. Simpson's Regel). Für die multidimensionalen Integrale der statistischen Mechanik jedoch ist Monte-Carlo die einzig vernünftige Methode [38].

Algorithmus 1 (Simple-Sampling)

I. Initialisiere die Zählersumme $s_1 := 0$ und die Nennersumme $s_2 := 0$.

II. WIEDERHOLE folgende Schritte M Mal:

1. Wähle zufällig einen Punkt π_l des Konfigurationsraumes.
2. Berechne den Wert für den Hamiltonian $H(\pi_l)$ und den Wert für die Observable $O(\pi_l)$ für die gewählte Konfiguration.
3. Berechne den Boltzmanngewichtsfaktor

$$W_B := \exp \left[-\frac{H(\pi_l)}{kT} \right]$$

für eine gewünschte Temperatur T .

4. Addiere das Produkt $O(\pi_l) \cdot W_B$ zur Zählersumme s_1 .
5. Addiere W_B zur Nennersumme s_2 .

III. Berechne

$$O_{\text{mittel}} := \frac{s_1}{s_2}.$$

Der Wert O_{mittel} ist der gesuchte Mittelwert.

Führen wir uns die Vorteile des Simple-Sampling Algorithmus vor Augen [40]. Ein erster Vorteil besteht darin, dass die einzelnen Konfigurationen zufällig gewählt werden, also statistisch unkorreliert sind. Dadurch kann auf alle Daten des gesamten Simulationslaufs die Standard-Fehler-Analyse angewendet werden (Vgl. Anhang D). Alle Daten sind gleich wertvoll, da sie statistisch unabhängig sind. Weiter ist es von Vorteil, dass in einem einzigen Simulationslauf Informationen über einen weiten Bereich von Temperaturen gewonnen werden können. So muss die gewünschte Temperatur T , bei der der Lauf stattfinden soll, nicht am Laufanfang eindeutig festgelegt werden, sondern es können während des Laufs für die zufällige Konfiguration π_l die Boltzmanngewichte zu n verschiedenen Temperaturen T_1, \dots, T_n errechnet werden, um daraus die einzelnen $\overline{O}(T_j)$ mit $j = 1, \dots, n$ zu bestimmen.

Um anhand eines Beispiels ein wichtiges Manko des Simple Sampling Algorithmus erkennen zu können [40], wollen wir uns erst klar darüber werden, für welche Variablen die Größen H , O und π im von uns simulierten Modell stehen. In unserem Fall besteht der Modell-Hamiltonian H aus der freien Energie der Krümmung (2.20)⁴

$$H(\pi) = G = \int_{A_{\text{real}}} \left\{ \frac{1}{2} \kappa_0 J^2 + \kappa_1 (\nabla J)^2 + \kappa_2 K^2 + \kappa_4 K^4 \right\} \sqrt{\gamma} dA_{\text{proj}}. \quad (3.4)$$

Die Observable O ist in erster Linie die Energie E , aber auch die Membranfläche A_{real} und die maximale Auslenkungsamplitude a_{max} (das ist die z -Komponente des Abstands zwischen der

⁴Ausdruck (2.20) ist eine freie Energie pro Flächeneinheit. Die freie Energie einer Fläche \mathcal{A} lässt sich mittels Integration bestimmen. Es gilt

$$G = \int_{\mathcal{A}} g \sqrt{\gamma} dA_{\text{proj}}.$$

Stützstelle mit dem höchsten und der Stützstelle mit dem niedrigsten z -Wert.) kommen in Betracht. Die Zustände der Membran sind durch ihre Form gegeben, die in Monge-Darstellung durch die Höhenverteilung $z(u, v)$ beschrieben wird. Die Variable π kann somit mit der Höhenverteilung $z(u, v)$ identifiziert werden. Mögliche Höhenverteilungen bestehen aus einer Grundstruktur - der flachen Membran oder der Membran mit Sattelstruktur - auf der Undulationsmoden \vec{q} superponiert werden, die einen mittleren Energiebeitrag von $\langle E_q \rangle = kT/2$ pro Mode liefern.

Aufgrund des Hamiltonians (3.4) erwarten wir (Vgl. Kapitel 5), dass das Energieminimum durch eine sattelförmige Struktur beschrieben wird, deren Einheitszelle in Abhängigkeit der gewählten Parameter $\kappa_0, \kappa_1, \kappa_2$ und κ_4 sinnvollerweise bei einer Kantenlänge $L \geq 6.0\text{nm}$ liegt und deren Überschussfläche, d.h. das Verhältnis von realer Fläche A_{real} zur projizierten Basisfläche A_{proj} durch $A_{\text{real}}/A_{\text{proj}} \geq 1.20$ gegeben ist. Der Begriff der Einheitszelle bedeutet, ein Hoch und ein Tief in einer derartigen Anordnung in dieser Simulationszelle zu erwarten, dass daraus folgend zwei Sättel vorzufinden sind je einer in u - und v -Richtung. Zur Simulation der Einheitszelle einer solchen Struktur verwenden wir in der Regel $8 \times 8 = 64$ Stützstellen. Dies setzt eine Grundstruktur voraus, die sich deutlich von einer fluktuierenden, flachen Grundstruktur unterscheidet, bei der maximale Überschussflächen durch $A_{\text{real}}/A_{\text{proj}} \leq 1.05$ beschränkt sein sollen.

Wir kennen das Aussehen der Sattelstruktur von vornherein nicht. Ein Ziel dieser Arbeit ist es, ihre Form zu bestimmen. Wir können deshalb nicht von der Sattelstruktur als Grundstruktur ausgehen. Mit Hilfe von Zufallszahlen aber direkt in die sattelförmige Überstruktur als Zufallskonfiguration mit negativer Energie $E_{\text{Sattel}} < 0$ zu kommen, ist sehr unwahrscheinlich. Zustände, die mit Hilfe von Zufallszahlen direkt erreicht werden können, sind bestenfalls mit den zu erwartenden Undulationen der flachen Membran vergleichbar, sofern sie maximale Auslenkungen von $a_{\text{max}} \approx 1\text{nm}$ nicht überschreiten. Die Energiewerte dieser flachen Membran liegen grösstenteils um einen Mittelwert $\langle E_{\text{flach}} \rangle \geq 0$ herum und nicht um den gesuchten Mittelwert der Sattelstruktur $\langle E_{\text{Sattel}} \rangle < 0$. Überschneiden sich die Energieverteilungen der flachen $p(E_{\text{flach}})$ und der sattelförmigen Struktur $p(E_{\text{Sattel}})$ nicht wesentlich - und das ist zu erwarten - so müssen sehr viele Simulationsläufe gemacht werden ($M \rightarrow \infty$), damit Näherung (3.3) genügt, es ist sogar der Fall wahrscheinlich, dass kein zufällig erzeugter Zustand einen für die Wahrscheinlichkeitsverteilung $p(E_{\text{Sattel}})$ sinnvollen Wert hat.

Wie suchen eine Technik, die Konfigurationen π_l nicht vollständig zufällig erzeugt sondern vorzugsweise in dem Phasenraumbereich, der bei der Temperatur T wichtig ist. Die Zufallskonfigurationen sollen deshalb aus einer nichtuniformen Verteilung gewählt werden. Dies führt uns zur Methode des Importance-Sampling.

3.2 Importance-Sampling

In diesem Abschnitt wird erst das allgemeine Prinzip des Importance-Sampling formal erklärt. Dazu sind die Begriffe rund um Markov-Prozesse von Bedeutung, die im Anhang C ausführlicher beschrieben werden. Da in dieser Arbeit der Metropolis-Algorithmus verwendet wird, wollen wir uns nach der allgemeinen Einführung auf diesen konzentrieren und ihn ausführlich besprechen.

3.2.1 Allgemeines Prinzip

Das Problem, das von Importance-Sampling gelöst werden muss, besteht darin, eine Folge von Zufallszahlen zu erzeugen, die in Phasenraumbereichen liegt, für die $O(\pi_l)$ die wahrscheinlichsten Beiträge liefert, die somit die wichtigsten Bereiche der Observablen darstellen. Wenn ein Zustand mit der ihm eigenen, physikalischen Wahrscheinlichkeit $p(\pi_l)$ in der Folge der Zufallszahlen auftritt, so kann Näherung (3.3) ersetzt werden durch

$$\bar{O} = \frac{\sum_{l=1}^M O(\pi_l) \exp[-\beta H(\pi_l)]/p(\pi_l)}{\sum_{l=1}^M \exp[-\beta H(\pi_l)]/p(\pi_l)}, \quad (3.5)$$

wobei die Verteilung $p(\pi_l)$ die normierte Gleichgewichtsverteilung

$$p(\pi_l) = p_{\text{eq}}(\pi_l) = \frac{1}{Z} \exp[-\beta H(\pi_l)], \quad \text{mit } Z = \sum_{l=1}^M \exp[-\beta H(\pi_l)] \quad (3.6)$$

ist. Damit folgt anstelle von (3.5) das arithmetische Mittel

$$\bar{O} = \frac{1}{M} \sum_{l=1}^M O(\pi_l). \quad (3.7)$$

Bei der Methode des Simple-Sampling geht jeder Zustand, mit dem Gewicht der Wahrscheinlichkeit seines Auftretens multipliziert, in die Berechnung des Mittelwertes (3.3) ein. Im Gegensatz dazu soll beim Importance-Sampling jeder Zustand überhaupt nur mit der passenden Wahrscheinlichkeit auftreten können. In diesem Fall kann die Mittelwertberechnung problemlos durch die Bildung des arithmetischen Mittels ersetzt werden.

Die Idee von Metropolis ist, die gesuchte Verteilung der Zufallszahlen mit Hilfe einer Markov-Kette zu konstruieren, die zusätzlich die Bedingung des detaillierten Gleichgewichts erfüllt [43]. (Für eine genaue Definition und zur Erläuterung dieser Begriffe sei auf den Anhang C verwiesen.) Eine Markov-Kette ist eine Folge von Zuständen eines stochastischen Prozesses, bei der der Zustand π_l aus seinem Vorgänger π_{l-1} mit der Übergangswahrscheinlichkeit $W(\pi_l|\pi_{l-1})$ hervorgeht. Dabei besitzt sie die Eigenschaft, dass die Wahrscheinlichkeit $p(\pi_l)$ für grosse Zeiten ($t \rightarrow \infty$) gegen einen stationären Wert strebt, der gerade den Gleichgewichtswert des Prozesses darstellt

$$p_{\text{s}}(\pi_l) = p_{\text{eq}}(\pi_l).$$

Um den stationären Zustand auf einfache Art und Weise zu finden, verlangt Metropolis von den Übergangswahrscheinlichkeiten der Markov-Kette die starke Bedingung des detaillierten Gleichgewichts (Vgl. Anhang C)

$$W(\pi_i|\pi_j)p_{\text{eq}}(\pi_j, t) = W(\pi_j|\pi_i)p_{\text{eq}}(\pi_i, t). \quad (3.8)$$

Diese Bedingung ist ein Trick, um leicht in die Gleichgewichtsverteilung zu gelangen. Betrachten wir zur Verdeutlichung seiner Bedeutung die Zustände A , B und C . Detailliertes Gleichgewicht fordert, dass im Gleichgewicht Übergänge $A \rightarrow B$ mit der gleichen Häufigkeit

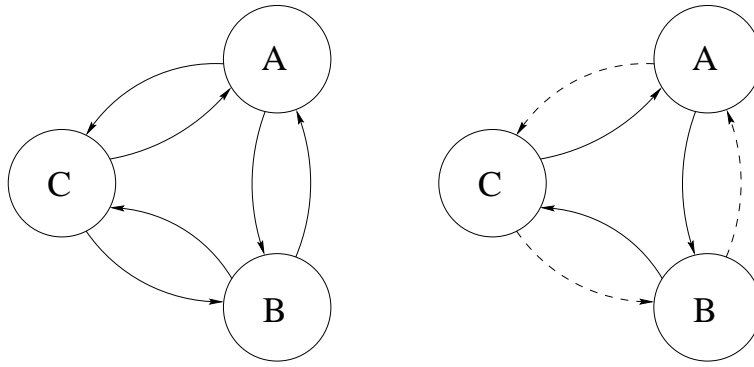


Abbildung 3.1: Detailliertes Gleichgewicht bedeutet: Ein Algorithmus darf nicht so gebaut sein, dass ein Kreislauf entsteht, wo ein Zustand A oft nach B , dieser oft nach C und dieser dann meist nach A übergeht (rechte Seite), ohne dass die umgekehrten Reaktionen im Gleichgewicht ebensooft auftreten (linke Seite).

auftreten, wie Übergänge in die umgekehrte Richtung $B \rightarrow A$. Gleiches gilt für die Übergänge zwischen A und C und zwischen B und C . Es dürfen also keine Übergänge in einer Richtung (z.B. $A \rightarrow B$) vor ihren Umkehrungen bevorzugt werden, wodurch Kreisläufe vermieden werden sollen (Abb. 3.1).

Es gibt verschiedene Methoden, die Bedingungen für Importance-Sampling, wie sie von Metropolis formuliert werden, zu erfüllen. Es gilt lediglich zu beachten, dass die Übergangsraten Bedingung (3.8) erfüllen

$$\frac{W(\pi_i|\pi_j)}{W(\pi_j|\pi_i)} = \frac{p_{\text{eq}}(\pi_i, t)}{p_{\text{eq}}(\pi_j, t)} = \exp[-\beta\Delta E],$$

wobei wir (3.6) verwenden und

$$\Delta E = H(\pi_i) - H(\pi_j)$$

einführen.

Eine häufig verwendete Methode, die von Metropolis selbst vorgeschlagen wird [43], und die gleichzeitig die wohl wichtigste Methode darstellt, legt die Übergangsmatrix zu

$$W(\pi_i|\pi_j) = \begin{cases} \alpha \exp[-\beta\Delta E] & , \text{ falls } \Delta E > 0 \\ \alpha & , \text{ falls } \Delta E \leq 0 \end{cases} \quad (3.9)$$

fest, wobei α für den Augenblick eine beliebige Proportionalitätskonstante sein soll, auf die wir weiter unten genauer eingehen (Vgl. Unterabschnitt 3.2.2). Festlegung (3.9) wird Metropolis-Sampling genannt. Eine andere häufig verwendete Methode ist das Barker-Sampling [44]⁵

$$W(\pi_i|\pi_j) = \alpha \frac{\exp[-\beta\Delta E]}{(1 + \exp[-\beta\Delta E])}.$$

⁵In [45] werden sowohl Metropolis- als auch Barker-Sampling als Grenzfälle einer allgemeineren Übergangswahrscheinlichkeitsmatrix identifiziert.

Beide Methoden sowohl Metropolis- als auch Barker-Sampling haben wir im Rahmen dieser Arbeit implementiert und getestet. Wie zu erwarten, sind aber keine qualitativen Unterschiede zwischen den Resultaten beider Methoden feststellbar. Die von uns in dieser Arbeit hauptsächlich verwendete Methode ist die des Metropolis-Sampling. Als Nächstes wollen wir deshalb genauer auf diese und ihren Algorithmus eingehen.

3.2.2 Der Metropolis-Algorithmus

Beziehung (3.9) legt die Übergangsmatrix für den Fall des Metropolis-Sampling fest. Mit dieser Festlegung lässt sich der Metropolis-Algorithmus leicht explizit angeben [43]. Störend könnte dabei der Faktor α in (3.9) wirken, über den wir noch etwas sagen wollen. Wir wissen, dass das Matrixelement $W(\pi_i|\pi_j)$ die Übergangswahrscheinlichkeit pro Zeiteinheit (Vgl. Anhang C) und somit von der Dimension $1/\text{Zeit}$ ist. Da die Exponentialfunktion dimensionslos ist, muss die Dimension im Faktor α ausgedrückt werden. Wir nennen den Faktor

$$\tau_{\text{MC}} = \frac{1}{\alpha}$$

die Monte-Carlo-Einheitszeit; die Zeit, die für einen Monte-Carlo-Schritt benötigt wird. Einen Monte-Carlo-Schritt pro Gitterplatz definieren wir als einen Versuch, die Konfiguration an einem Gitterpunkt zu verändern, also das Erzeugen eines Zufallspunktes des Konfigurationsraumes. Die Zeit für einen Monte-Carlo-Schritt pro Gitterplatz legen wir fest zu $\tau_{\text{MC}}/N = 1$, wobei N für die Anzahl Gitterplätze steht. Jetzt folgt der Wert für den Faktor α zu

$$\alpha = \frac{1}{N}.$$

Das Bild, das hinter diesen Überlegungen steht, wird mit Hilfe der in Anhang C eingeführten Mastergleichung (C.3) klarer. Diese folgt aus den Eigenschaften der Markov-Ketten und liefert eine dynamische Interpretation der Monte-Carlo-Simulation. Übergänge zwischen verschiedenen Zuständen, wie sie durch die Mastergleichung beschrieben werden, geschehen in Einheiten von Monte-Carlo-Zeiteinheiten. Es sei darauf hingewiesen, dass die Monte-Carlo-Zeiteinheit, wie sie durch die Mastergleichung festgelegt wird, nicht mit einer physikalischen Zeit verwechselt werden darf. Monte-Carlo-Simulationen machen keine Aussagen über die physikalische Dynamik eines Systems. Es können somit keinerlei Aussagen über echte, physikalische Zeitverläufe getroffen werden.⁶

Wir sind jetzt in der Lage, den Metropolis-Algorithmus formulieren zu können.

Algorithmus 2 (Metropolis-Algorithmus)

- I. Initialisiere arithmetisches Mittel $s := 0$, wähle Lauftemperatur T .
- II. Wähle zufällig einen Punkt π_{aktuell} des Konfigurationsraumes.
- III. WIEDERHOLE folgende Schritte M Mal:
 1. Wähle Gitterpunkt.

⁶Um Zeitaussagen machen zu können, muss auf Molekulardynamik-Simulationen zurückgegriffen werden. Dort wird die Zeitentwicklung physikalischer Systeme durch deterministische Gleichungen für die Variablen beschrieben.

2. Wähle zufällig einen Punkt π_{zufall} des Konfigurationsraumes.
3. Berechne die Energie

$$\Delta E := H(\pi_{\text{aktuell}}) - H(\pi_{\text{zufall}}) ,$$

die zur Veränderung des alten Zustands π_{aktuell} nötig ist.

4. FALLS $\Delta E < 0$: akzeptiere neue Konfiguration

$$\pi_{\text{aktuell}} := \pi_{\text{zufall}} .$$

SONST:

- i. Wähle Zufallszahl $Z_{\text{zufall}} \in [0; 1)$.
- ii. FALLS $Z_{\text{zufall}} < \exp[-\beta\Delta E]$: akzeptiere neue Konfiguration

$$\pi_{\text{aktuell}} := \pi_{\text{zufall}} .$$

5. Berechne den Wert für die Observable $O(\pi_{\text{aktuell}})$.
6. Addiere $O(\pi_{\text{aktuell}})$ zur Summe s .

IV. Berechne

$$O_{\text{mittel}} := \frac{s}{M} .$$

Der Wert O_{mittel} ist der gesuchte Mittelwert.

Zum Flussdiagramm (Abb. 3.2) des Metropolis-Algorithmus, wie er im Rahmen dieser Arbeit zur Anwendung kommt, wollen wir ein paar erläuternde Bemerkungen machen.

Nach anfänglichen Initialisierungsschritten, in denen auch die Ausgangsform der Membranfläche festgelegt wird,⁷ fallen im Flussdiagramm (Abb. 3.2) zwei Schleifen auf. Die äussere Schleife wird durch die Anzahl der gewünschten Sweeps festgelegt. Ein Sweep besteht aus einer Anzahl Formänderungsvorschläge, die der Anzahl der Stützstellen entspricht und repräsentiert somit die Monte-Carlo-Einheitszeit τ_{MC} . Was die tatsächliche Anzahl der Sweeps betrifft, so sind dazu noch einige Überlegungen notwendig, für die wir auf Anhang D, auf den Abschnitt über die Autokorrelationszeit verweisen.

Ein Sweep besteht aus der einmaligen Ausführung der inneren Schleife. Die innere Schleife wird - wie soeben erwähnt - so oft ausgeführt, wie die Anzahl der Stützstellen der simulierten Fläche beträgt. Um eine bestimmte Stützstelle auszuwählen, gibt es zwei Möglichkeiten. Einerseits können die Stützstellen sequentiell, in regelmässiger Reihenfolge abgefahren, alternativ dazu können sie zufällig ausgewählt werden. Die erste Methode erspart einen Zufalls-generatorzugriff und ist somit schneller als die zweite. Beide Arten der Stützstellenwahl sind im Programm zu dieser Arbeit implementiert. Wie zu erwarten [38, 40] spielt aber bei entsprechenden Testläufen die Art der Wahl keine Rolle für die Gleichgewichtseigenschaften der Simulation. Daher wird in den meisten Fällen die sequentielle, schnellere Art zur konkreten Berechnung während der Simulation gewählt.

Ist der zu ändernde Gitterpunkt festgelegt, so wird eine zufällige Änderung Δz bestimmt. Die Überlegungen, die zur Bestimmung einer optimalen Änderung führen, sind in Unterabschnitt 3.3.2 ausführlich beschrieben und sollen hier nicht vorweggenommen werden. Nach

⁷Die meistverwendete Ausgangsform ist die flache Membran mit $z = 0$ bei jeder Stützstelle. Auf die Art der Ausgangsform wird in dieser Arbeit bei jeder Problemstellung explizit verwiesen.

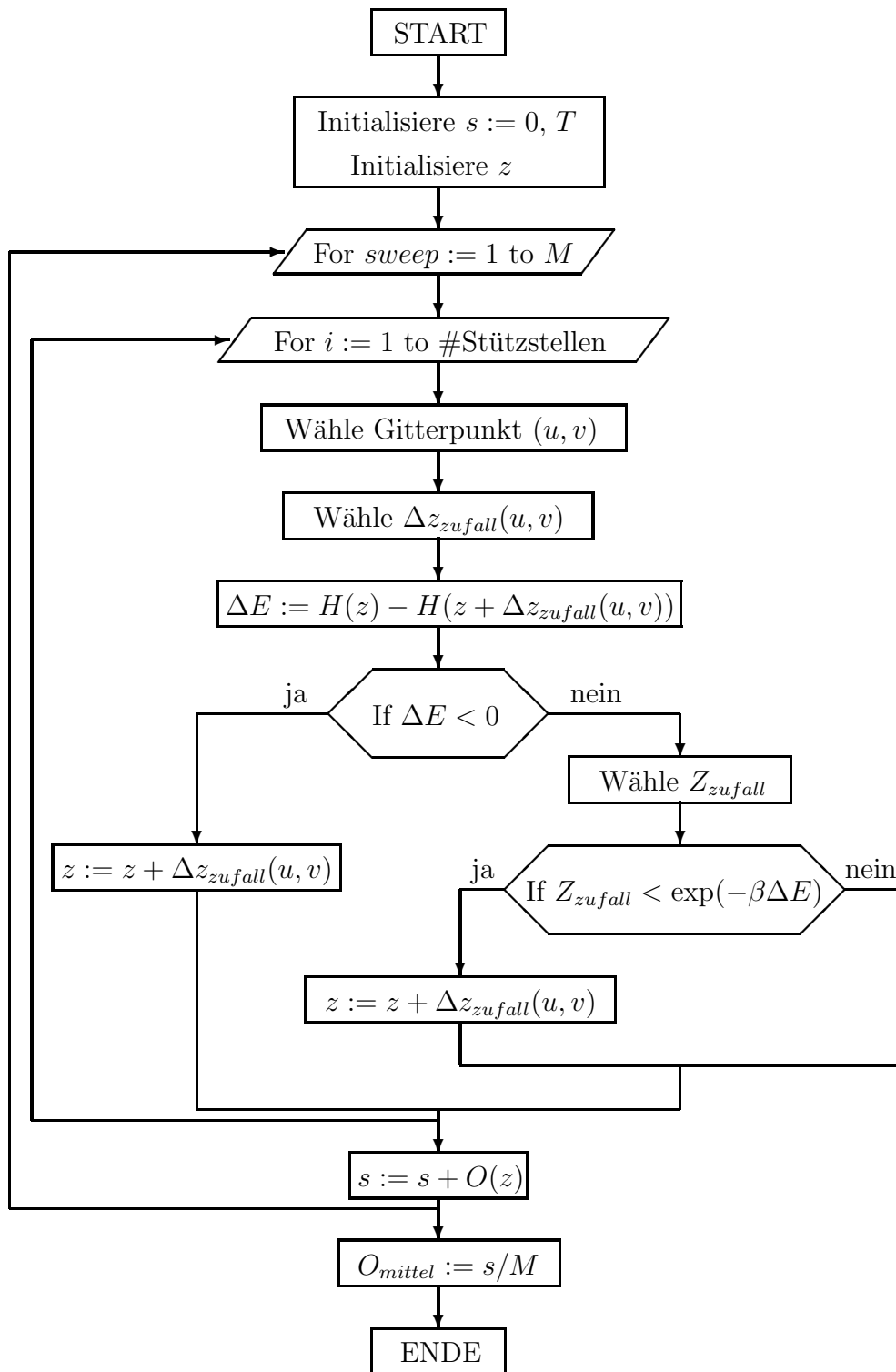


Abbildung 3.2: Flussdiagramm zum Metropolis-Algorithmus.

der Bestimmung der z -Wert-Änderung einer Stützstelle besteht der nächste Schritt darin, mit Hilfe des Hamiltonian (3.4) die Energie ΔE zu berechnen, die zur Veränderung der Membranform von der alten in die neue, zufällige Konfiguration benötigt wird.

Der nächstfolgende Block im Flussdiagramm (Abb. 3.2) kann mit Hilfe der Markov-Funktion

$$\mathcal{W} = \min(1, \exp[-\beta\Delta E])$$

zusammengefasst werden. Die zufällig erzeugte Membranform wird mit der Wahrscheinlichkeit \mathcal{W} der Markov-Funktion als neue Ausgangsmembranform akzeptiert. Formänderungen, die zu Energieabsenkungen ($\Delta E < 0$) führen, werden immer ausgeführt, während Änderungen, die zu einem Energieanstieg ($\Delta E \geq 0$) führen mit einer Wahrscheinlichkeit proportional dem Boltzmanngewichtsfaktor ausgeführt werden können. Der Vorteil dieser Vorgehensweise besteht darin, dass Membranformen, die ein lokales Energieminimum darstellen, verlassen werden können zugunsten der Membranformen des gesuchten globalen Minimums. Die neu erzeugte Membranform (auch bei verworfener Stützstellenänderung Δz) ist Ausgangsform für die weiteren Berechnungen, die innerhalb der Schleifen wieder von vorne beginnen.

Das Ende der inneren Schleife legt den Zeitpunkt fest, zu dem der Wert der gewünschten Observablen O berechnet und abgespeichert wird. Da dieser Schritt viele Operationsschritte beinhaltet und aufeinanderfolgende Konfigurationen eine starke Korrelation untereinander aufweisen können, ist es erlaubt, ihn nur nach grösseren, festen Zeitintervallen durchzuführen. Seine Durchführung soll auf jeden Fall und unabhängig von der spezifischen Anzahl der innerhalb der festgelegten Sweepzahl gemachten Änderungen der Stützstellenwerte vorgenommen werden [40]. Die Abstände $S \geq 1$ (als Zeitintervall interpretiert: $\tau_S = S \cdot \tau_{\text{MC}}$) zwischen der Abspeicherung zweier Werte einer Observablen werden bei Laufbeginn festgelegt.

3.3 Einige übliche Tricks

Drei übliche Simulationshilfen sollen in diesem Abschnitt dargestellt werden: Periodische Randbedingungen, dynamisch optimiertes Monte-Carlo und Simulated Annealing.

3.3.1 Periodische Randbedingungen

Bei Computersimulationen ist es in den meisten Fällen nur möglich, Objekte zu untersuchen, die aus sehr wenigen Teilchen - in unserem Fall Stützstellen - bestehen. Aber auch jeder reale, physikalische Körper hat eine endliche Ausdehnung und somit einen Rand. Daraus folgen meist schwer beschreibbare Randeffekte, die das physikalische Verhalten des Körpers zwar beeinflussen, die aber oft vernachlässigbar sind. Die üblicherweise benutzte Methode, vom Rand verursachte Störungen zu vermindern, ist die Verwendung periodischer Randbedingungen [38, 40].

Periodische Randbedingungen werden zum ersten Mal 1912 von den Physikern Born und von Karman vorgeschlagen. Für unseren Fall eines quadratischen Flächenstückes mit der Kantenlänge L tun wir dabei so, als ob an jede Seite des originalen Flächenstückes seine identische Kopie durch eine reine Translationsbewegung um ein Vielfaches der Kantenlänge in dieser Richtung angehängt wird. Für eine durch die u - und v -Position vorgegebene Höhenverteilung $z(u, v)$ gilt somit allgemein

$$z(u, v) = z(u + n_1 L, v + n_2 L), \quad n_1, n_2 \in \mathbb{Z}.$$

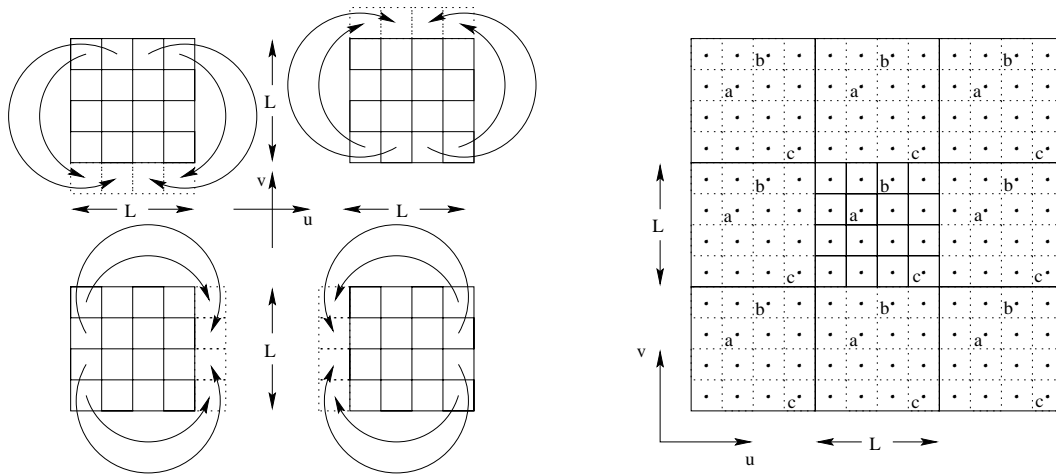


Abbildung 3.3: Realisierung periodischer Randbedingungen eines quadratischen Flächenstücks mit der Kantenlänge L und der Stützstellenzahl $N = 4 \times 4$. Im linken Bildteil wird für alle vier Richtungen der Ebene einzeln die Äquivalenz der Randzellen mit der Randzelle der periodisch fortgesetzten Basisfläche durch einen Pfeil dargestellt. Durch den rechten Bildteil wird deutlich, dass durch die wiederholte periodische Fortsetzung des Originals (Quadrat mit durchgezogenen Linien im Zentrum) eine unendlich ausgedehnte Fläche vollständig abgedeckt werden kann. Die Punkte in den einzelnen Zellen stellen die Lage der Stützstellen dar.

Bei Simulationsbeginn wird nicht nur die Kantenlänge L , sondern auch die Anzahl N der Stützstellen festgelegt, durch die das Quadrat beschrieben werden soll. Dabei ist es wichtig, dass jede Stützstelle in jeder Richtung der Basisvektoren u und v einen eindeutigen Nachbarn besitzt. Eine allgemeine Formel zur Bestimmung der nächsten Nachbarn in u -Richtung für einen beliebigen Punkt (u, v) kann mit Hilfe des Modulo-Operators berechnet werden zu

$$z(u, v) \rightarrow z((u \pm 1 + N_{\text{KI}}) \bmod N_{\text{KI}}, v), \quad \text{wobei } N_{\text{KI}} = \sqrt{N};$$

für die v -Richtung analog. Betrachten wir als Beispiel die zwei nächsten Nachbarn in u -Richtung des Randpunktes $(1, v)$, für ein beliebiges v . Sein Nachbar in positiver u -Richtung ist der Punkt $(2, v)$ und sein Nachbar in negativer u -Richtung der Punkt (N, v) . Für den Punkt (N, v) lassen sich die Nachbarn $(1, v)$ bzw. $(N - 1, v)$ in positiver bzw. negativer u -Richtung bestimmen (Abb. 3.3).

Die Methode des periodischen Fortsetzens der originalen Basiszelle entspricht einem Verbinden ihrer zwei gegenüberliegenden Kanten miteinander. Dadurch lässt sich die Topologie der Simulationsbasiszelle durch die Form eines Torus im dreidimensionalen Raum visualisieren.

Eine interessante und wichtige Frage ist, ob durch das Verwenden von periodischen Randbedingungen die physikalischen Eigenschaften eines Systems entscheidend verändert werden. Es ist klar, dass periodische Randbedingungen störende Effekte haben müssen bei Systemen, wo eine langreichweitige Ordnung (langreichweitig in Bezug auf die Kantenlänge L) eine Rolle spielt, beispielsweise langwellige Fluktuationen. So kann die kritische Divergenz der Korrelationslänge bei Phasenübergängen zweiter Ordnung zu Veränderungen führen, die unter dem Begriff der „Finite Size Effekte“ bekannt sind [40].

Aus der Literatur [38] geht hervor, dass periodische Randbedingungen weit weg von Phasenübergangspunkten „erfahrungsgemäss“ kaum einen Einfluss auf die Eigenschaften der Gleichgewichtsthermodynamik eines Systems besitzen. Diese Frage kann jedoch schlussendlich nicht beweiskräftig geklärt werden. Durch eine Variation der Kantenlänge der Simulationsfläche ist es aber möglich, ein Gefühl für den Einfluss der periodischen Randbedingungen auf das System zu bekommen. Weiter lässt das Bestimmen von Korrelationslängen eine quantitative Abschätzung zu.

3.3.2 Dynamisch optimiertes Monte-Carlo

In dem von uns realisierten Algorithmus der Monte-Carlo-Simulation werden lokale Änderungsvorschläge der Auslenkungsamplitude $z(u, v)$ mit Hilfe der Formel

$$\Delta z = \Delta z_{\max} \cdot (2\mathcal{Z} - 1) , \quad \mathcal{Z} \in [0; 1]$$

festgelegt, wobei \mathcal{Z} eine bei jedem Versuch neu erzeugte Zufallszahl und Δz_{\max} die maximale Schrittgrösse darstellt. Die vorgeschlagenen Änderungsmöglichkeiten der Auslenkungsamplitude liegen somit bei jedem Monte-Carlo-Schritt im Intervall $[-\Delta z_{\max}; \Delta z_{\max})$, wobei die tatsächliche Grösse des Parameters Δz_{\max} frei gewählt werden kann. Wird die maximale Schrittgrösse Δz_{\max} zu klein vorgegeben, wird ein hoher Anteil der vorgeschlagenen Bewegungen angenommen, aber aufeinanderfolgende Zustände der Membran sind stark miteinander korreliert. Wird der Parameter als zu gross angenommen, werden beinahe alle Bewegungsvorschläge verworfen, es finden also kaum Membranbewegungen statt. Auch dies hat hohe Korrelationen aufeinanderfolgender Membranzustände zur Folge.

Eine gängige Methode zur Festlegung der maximalen Schrittgrösse Δz_{\max} ist das dynamisch optimierte Monte-Carlo. Dabei wird während der Simulation die Akzeptanzrate, das ist das Verhältnis der Anzahl akzeptierter zur Gesamtzahl der gemachten Vorschläge, gemessen. Die maximale Schrittgrösse wird danach in Abhängigkeit einer gewünschten Akzeptanzrate angepasst. In den meisten Fällen wird eine Rate von 50% als sinnvoll erachtet, es ist aber von vornherein nicht klar, dass eine solche Grössenordnung optimale Ergebnisse liefert und so ist durchaus eine andere Akzeptanzrate denkbar [38].

Als Startwert legen wir in unserer Realisierung eines dynamisch optimierten Monte-Carlo-Laufs $\Delta z_{\max} = 1\text{nm}$ fest. Das Anpassen von Δz_{\max} bedeutet danach, dass das Programm nach jedem Sweep die Akzeptanzrate bestimmt. Unterschreitet das Verhältnis von akzeptierten zu allen Versuchen eine Schwelle von 30%, so ist Δz_{\max} zu klein und wird mit dem Faktor 10 multipliziert. Überschreitet dieses Verhältnis eine Schwelle von 60%, so wird Δz_{\max} als zu gross angenommen und wird durch den Faktor 10 dividiert. Im Rest der Fälle bleibt die maximale Schrittgrösse unverändert.

3.3.3 Simulated Annealing

Die Idee zur Methode des Simulated Annealing⁸ hat ihren Ursprung in der Thermodynamik [46]. Dort lässt sich beobachten, dass Flüssigkeiten, die langsam abgekühlt werden, beim Übergang in einen Festkörper eine kristalline Struktur annehmen können, für die die Energie des Systems sich in einem globalen Minimum befindet. Wird die Flüssigkeit schnell abgekühlt - zum Beispiel durch Quenchen (Vgl. Kapitel 1) - so findet sie das globale Energieminimum

⁸Das englische Verb „to anneal“ wird mit „ausglühen“ oder „tempern“ übersetzt.

während des Abkühlvorgangs nicht und endet in einem amorphen oder polykristallinen Zustand. Das liegt daran, dass das gesuchte Minimum versteckt liegt zwischen vielen kleineren lokalen Minima. Schnelles Kühlen kann also zu einem lokalen und muss nicht zum globalen Minimum führen, während beim langsamen Kühlen dem System Zeit gegeben wird, die Atome derart anzuordnen, dass das globale Minimum erreicht werden kann.

Auf dieser Beobachtung aufbauend besteht der Prozess des Simulated Annealing darin, dass das System, in unserem Fall die Membran, erst in einen Zustand hoher Temperatur T versetzt wird. Die Temperatur spielt dabei die Rolle eines Kontrollparameters, der anschliessend langsam und stufenweise erniedrigt wird, bis die Membran einfriert und keine Änderungen mehr auftreten. Es gilt dabei zu beachten, bei jeder Temperaturstufe die Simulation so lange laufen zu lassen, dass das System die Möglichkeit hat, einen Gleichgewichtszustand zu erreichen [46].

Für die Art der Temperaturerniedrigung gibt es kein allgemeingültiges Rezept. In unserem Fall werden die Temperaturstufen T_ν von einer Starttemperatur T_s ausgehend mit Hilfe der Formel

$$T_\nu = \epsilon^\nu T_s, \quad \nu \geq 0$$

bestimmt, wobei für den Parameter ϵ die Beziehung $1 > \epsilon \geq 0.75$ sinnvoll ist. Bei jeder Temperaturstufe wird dabei die volle, bei Simulationsbeginn festgelegte Sweepzahl abgearbeitet, bevor die Zählvariable ν um 1 erhöht wird. Wird eine bei Simulationsbeginn festgelegte Temperaturgrenze T_f unterschritten, so wird die Simulation beendet.

Simulated Annealing wird in unseren Simulationen nur zur Bestimmung der Einheitszellengrösse der Sattelstruktur verwendet. Dort ist es wichtig, dass eine Struktur entsteht, die bestmöglich geglättet ist, deren Form also durch möglichst wenige Undulationen gestört ist.

