



Institut für Theoretische Physik

# Dimensionelles Crossover im Ising-Modell

Bachelorarbeit im Studiengang Physik

Leopold Kellers

l.k@wwu.de

4. Dezember 2013

Betreut von Prof. Dr. G. Münster

Diese Version der Bachelorarbeit unterscheidet sich nur unwesentlich von der Version, die am 4. Dezember 2013 beim Prüfungsamt gedruckt vorgelegt wurde.

# Inhaltsverzeichnis

<b>1. Einführung</b>	<b>1</b>
<b>2. Theoretische Grundlagen</b>	<b>1</b>
2.1. Das Ising-Modell . . . . .	1
2.1.1. Definition . . . . .	1
2.1.2. Analytisch bekannte Eigenschaften . . . . .	3
2.1.3. Der Phasenübergang (nicht nur) im Ising-Modell . . . . .	5
2.1.4. Finite Size Effekte . . . . .	8
2.2. Die Monte-Carlo-Methode . . . . .	8
2.2.1. Markov-Ketten-Monte-Carlo . . . . .	9
2.2.2. Der Metropolis-Algorithmus . . . . .	10
2.2.3. Systematische und statistische Fehler . . . . .	11
2.2.4. Critical Slowing Down . . . . .	13
<b>3. Problemstellung und Herangehensweise</b>	<b>14</b>
3.1. Numerische Bestimmung von $\nu$ und $\beta_c$ . . . . .	14
3.2. Die Zeitscheibenkorrelationsfunktion . . . . .	16
3.3. Cluster-Algorithmen . . . . .	18
3.4. Verbesserte Schätzer . . . . .	24
3.5. Behandlung der Autokorrelation . . . . .	25
<b>4. Implementierung</b>	<b>27</b>
4.1. Programmwurf . . . . .	27
4.2. Monte-Carlo . . . . .	28
4.3. Statistik . . . . .	30
4.4. Test . . . . .	32
4.5. Programmbenutzung . . . . .	34
<b>5. Ergebnisse</b>	<b>35</b>
5.1. Validierung der Implementierung . . . . .	35
5.2. Korrelationslängen . . . . .	39
5.3. Kritischer Exponent und kritische Temperatur . . . . .	39
5.4. Diskussion . . . . .	43
<b>6. Zusammenfassung</b>	<b>47</b>
<b>Anhang A: Fits zur Bestimmung von <math>\nu</math> und <math>\beta_c</math></b>	<b>48</b>
<b>Anhang B: Gewonnene Daten in Tabellenform</b>	<b>53</b>

## 1. Einführung

Das Ising-Modell ist ein sehr einfaches Modell aus der statistischen Physik, mit dem sich das Phänomen des Ferromagnetismus erstmals erklären ließ. Trotz seiner Einfachheit zeigt es in mehr als einer Dimension einen Phasenübergang zwischen einer magnetisierten und einer unmagnetisierten Phase bei einer bestimmten Temperatur. Das Ising-Modell ist äquivalent zu verschiedenen anderen Modellen, etwa dem des Gittergases, und letztlich ein Spezialfall einer ganzen Klasse von Gittermodellen, den sogenannten  $n$ -Vektor-Modellen sowie Potts-Modellen[1] und hat sogar Anwendungen in der Biologie[2]. Aus diesen Gründen wird es häufig als „Spielwiese“ zum Testen neuer Methoden der statistischen Physik benutzt. Man könnte auch sagen, es handelt sich beim Ising-Modell quasi um das Schokoladeneis der statistischen Feldtheorie - man greift immer wieder gerne darauf zurück<sup>1</sup>!

Nichtsdestotrotz hält das Ising-Modell noch einige ungeklärte Fragen bereit. Allen voran mangelt es an einer exakten Lösung im dreidimensionalen Fall, während von Onsager 1944 eine analytische Lösung für den zweidimensionalen Fall gefunden wurde[4]. Man kann sogar zeigen[5], dass das Problem der Lösbarkeit in diesen beiden Fällen dem großen ungelösten Problem  $P \stackrel{?}{=} NP$  der theoretischen Informatik äquivalent ist.

Diese Arbeit beschäftigt sich mit einer numerischen Simulation des Verhaltens bestimmter Größen im Ising-Modell beim Übergang von einem (quasi-) zweidimensionalen zu einem dreidimensionalen System. Dazu wird ein Programm entwickelt, das effiziente Monte-Carlo-Algorithmen für das Ising-Modell implementiert, und die damit erhaltenen Resultate analysiert. Zuvor jedoch folgt eine Einführung in die theoretischen Grundlagen des hier behandelten Stoffes.

## 2. Theoretische Grundlagen

Bevor ein Einstieg in die eigentliche Thematik möglich ist, sollten zunächst durch Klärung der verwendeten Begrifflichkeiten gemeinsame Grundlagen geschaffen werden. Dies betrifft das Ising-Modell im Allgemeinen, die Theorie der Phasenübergänge und die Monte-Carlo-Methode. Letztere kann benutzt werden, um numerische Ergebnisse zu nicht analytisch lösbaeren Problemen zu erhalten.

### 2.1. Das Ising-Modell

Das Ising-Modell wurde erstmals von Ernst Ising in seiner Doktorarbeit im Jahre 1924 für den eindimensionalen Fall untersucht und die Ergebnisse später veröffentlicht [6]. Hier interessiert aber die allgemeine Version dieses Gittermodells in mehr als einer Dimension.

#### 2.1.1. Definition

Daher sei zunächst ein vorerst endliches, kubisches,  $d$ -dimensionales *Gitter*  $\Lambda$  mit Seitenlängen  $L_i \in \mathbb{N}$  mit  $i \in \{1, 2, \dots, d\}$  und dem Abstand 1 zwischen den einzelnen Gittereinträgen definiert. Das Volumen  $V$  des Gitters ergibt sich zu  $V = \prod_{i=1}^d L_i$ . Nun werde jedem der  $V$  Gitterplätze  $x \in \Lambda$  ein *Spin*  $s_x$ , also eine elementare Magnetisierung, mit einem der beiden diskreten Werte aus  $\{-1; 1\}$  zugeordnet, was in einem bestimmten

---

<sup>1</sup>Vergleiche dazu die Analogie zwischen Vanilleeis und harmonischem Oszillator in [3].

Zustand  $Z$  des Ising-Modells resultiert. Der Raum aller möglichen Zustände sei mit  $S = \{-1; 1\}^V$  bezeichnet.

Jeder Spin auf einem Gitterplatz  $x$  möge nur mit seinen direkten Nachbarn  $y$  sowie einem eventuell vorhandenen äußeren Magnetfeld  $H$  wechselwirken, sodass sich für jeden Zustand die *Hamiltonfunktion*

$$\mathcal{H}(Z) = -J \sum_{\substack{x,y \in \Lambda \\ |x-y|=1}} s_x s_y - H \sum_{x \in \Lambda} s_x \tag{1}$$

mit einer positiven *Kopplungskonstante*  $J$  angeben lässt. Die im System befindliche Energie ist also maximal, wenn alle Nachbarn eines Spins entgegengesetztes Vorzeichen haben, bzw. minimal, wenn alle Spins parallel stehen. In Abwesenheit eines äußeren Magnetfeldes ist keine der beiden Möglichkeiten für die Richtung eines Spins ausgezeichnet. Für  $H \neq 0$  wird diese Symmetrie allerdings gebrochen und eine Stellung parallel zum Feld ist energetisch günstiger.

Es existieren insgesamt  $2^V$  Zustände, in denen sich das System befinden kann. Die Wahrscheinlichkeit, es zu einer gegebenen Temperatur  $T$  in einem bestimmten Zustand  $Z$  anzutreffen, ist durch

$$p(Z) = \frac{1}{\mathcal{Z}} e^{-\frac{\mathcal{H}(Z)}{k_B T}} \tag{2}$$

mit der Boltzmann-Konstante  $k_B$  und der *Zustandsumme*

$$\mathcal{Z} = \sum_{Z \in S} e^{-\beta \mathcal{H}(Z)} \tag{3}$$

gegeben. Wie üblich bezeichne  $\beta = \frac{1}{k_B T}$  die *inverse Temperatur*.

Nun wurde bei der bisherigen Betrachtung unterschlagen, wie mit den am Rand sitzenden Spins, welche weniger als  $2d$  Nachbarn besitzen, zu verfahren ist. Dieses Problem wird durch das Einführen von *periodischen Randbedingungen* gelöst, das heißt, die in Bezug auf eine bestimmte Richtung äußersten Gitterplätze seien mit den jeweils gegenüberliegenden in dieser Richtung verbunden. Damit wird das System auch translationsinvariant in alle Richtungen des Gitters. Im eindimensionalen Fall erhält man also einen Ring, im allgemeinen Fall einen Hypertorus. Zur Veranschaulichung, wie man sich das eben Beschriebene vorstellen kann, dient Abbildung 1, wo der zweidimensionale Fall dargestellt ist.

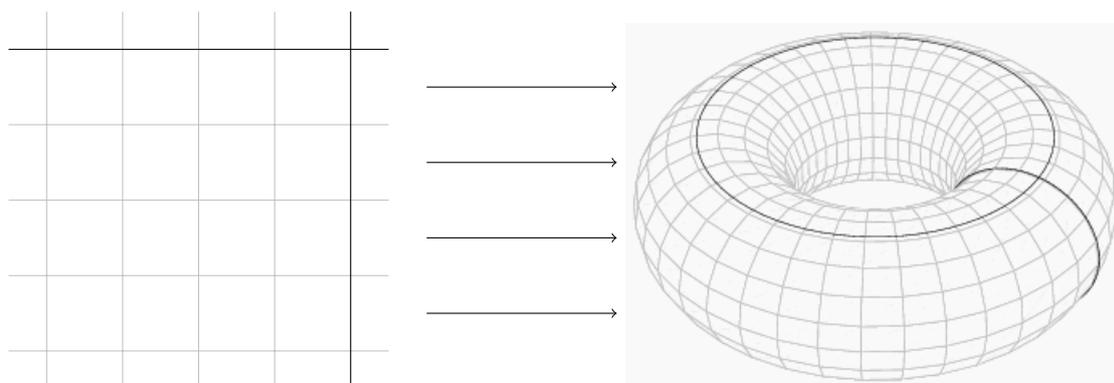


Abbildung 1: Visualisierung der periodischen Randbedingungen des zweidimensionalen Ising-Modells. Quelle des rechten Teilbildes (Torus): [7].

In dieser Arbeit wird grob gesagt das Verhalten der Korrelation zwischen den verschiedenen Gitterstellen in Abhängigkeit von vorgegebenen Parametern untersucht. Daher macht es Sinn, an dieser Stelle auch gleich eine *Korrelationsfunktion*  $g(x, y)$  zwischen den Spins  $s_x$  und  $s_y$  an den Gitterplätzen  $x$  und  $y$  zu definieren. Es sei

$$g(x, y) = \langle s_x s_y \rangle - \langle s_x \rangle \langle s_y \rangle, \tag{4}$$

wobei  $\langle \cdot \rangle$  den Erwartungswert bezüglich der durch  $\mathcal{Z}$  induzierten Wahrscheinlichkeitsverteilung bezeichnet. Da das System translationsinvariant definiert wurde, gilt

$$\langle s_x \rangle = \langle s_y \rangle =: M \tag{5}$$

und  $g(x, y)$  hängt nur vom Abstand  $|x - y|$  zwischen den beiden Gitterstellen ab. Der Erwartungswert  $\langle s_x \rangle$  für einen beliebigen Spin  $s_x$  wird auch als *Magnetisierung pro Spin*  $M$  bezeichnet.

Die Korrelationsfunktion beschreibt, wie stark das Verhalten des Spins am Punkt  $x$  mit dem Verhalten des Spins an Punkt  $y$  gekoppelt ist bzw. wie stark diese sich gegenseitig beeinflussen. Die Längenskala, auf der diese Beeinflussung stattfindet, bezeichnet man als *Korrelationslänge*  $\xi$ . Falls  $x$  und  $y$  überhaupt nicht korreliert sind, so faktorisiert der erste Term  $\langle s_x s_y \rangle$  zum Produkt  $\langle s_x \rangle \langle s_y \rangle$  und die Korrelationsfunktion verschwindet.

### 2.1.2. Analytisch bekannte Eigenschaften

Wie bereits erwähnt, sind für das Ising-Modell im ein- und im zweidimensionalen Fall analytische Lösungen bei Abwesenheit äußerer Magnetfelder  $H = 0$  bekannt. Analytische Lösung bedeutet, dass im Fall eines (in alle Richtungen) unendlich großen Gitters  $\Lambda$  ein geschlossener Ausdruck für die freie Energie pro Spin  $f$  oder eines bezüglich Legendre-Transformation äquivalenten thermodynamischen Potentials angegeben werden kann. Die freie Energie pro Spin  $f$  ergibt sich dabei im thermodynamischen Limes, bei dem das Volumen in alle Richtungen gegen Unendlich strebt, aus der Zustandssumme via

$$f = - \lim_{V \rightarrow \infty} \frac{1}{V} \frac{\ln \mathcal{Z}}{\beta}. \tag{6}$$

Unter der Annahme, dass die freie Energie  $F$  des Systems mit der Systemgröße skaliert, sollte sich für  $f$  ein endlicher Wert ergeben.

Wenn die freie Energie pro Spin bekannt ist, kann man daraus die Magnetisierung pro Spin

$$M = - \frac{\partial f}{\partial H} \tag{7}$$

bestimmen.

Wie bereits angemerkt, wurden die Lösungen bei verschwindendem äußeren Magnetfeld  $\lim H \rightarrow 0^2$  für das eindimensionale Modell von Ising selbst und für das zweidimensionale von Onsager erstmals berechnet. Die folgenden Ausführungen fassen jedoch die Ergebnisse aus [8] zusammen, wo die Berechnungen konsequent mit der sogenannten Transfermatrixmethode durchgeführt wurden.

### Analytische Ergebnisse für das eindimensionale Ising-Modell

Das Ergebnis dieser Berechnungen ist, dass für die freie Energie pro Spin der Ausdruck

$$f(H, \beta) = - \frac{1}{\beta} \ln \left( e^{\beta J} \cosh(\beta H) + \sqrt{e^{2\beta J} \sinh^2(\beta H) + e^{-2\beta J}} \right) \tag{8}$$

<sup>2</sup>Man beachte, dass hier die Reihenfolge der Limites relevant ist, also immer  $\lim_{H \rightarrow 0} \lim_{V \rightarrow \infty}$  zu nehmen ist.

gilt. Benutzung von (7) führt für die Magnetisierung  $M$  auf

$$M(H, \beta) = \frac{e^{\beta J} \sinh \beta H}{\sqrt{e^{2\beta J} \sinh^2(\beta H) + e^{-2\beta J}}}, \quad (9)$$

was für alle  $H$  und  $\beta < \infty$  eine analytische Funktion ist.

Für den Fall  $H = 0$  erhält man für die Korrelationsfunktion

$$g(x, y) = \tanh(\beta J)^{|x-y|}. \quad (10)$$

Diese fällt für alle von Null verschiedenen Temperaturen gegen Null, wenn man den Abstand zwischen  $x$  und  $y$  gegen Unendlich laufen lässt, weil die Basis stets betragsmäßig kleiner 1 ist. Bei  $T = 0 \Leftrightarrow \beta = \infty$  jedoch nimmt die Korrelation den konstanten Wert 1 an. Folglich haben alle Spins der Kette entweder den Wert 1 oder -1 und die Korrelationslänge  $\xi$ , die ja angibt, auf welcher Längenskala die Spins miteinander korrelieren, divergiert nach  $\infty$ . Einen Satz von Parametern, hier  $T = 0$ , für den ein thermodynamisches System dieses Verhalten der Korrelationslänge zeigt, nennt man *kritischen Punkt*. Es bleibt also festzuhalten, dass das eindimensionale Ising-Modell einen kritischen Punkt, aber keinen Phasenübergang aufweist.

### Analytische Ergebnisse für das zweidimensionale Ising-Modell

Der zweidimensionale Fall ist mathematisch wesentlich anspruchsvoller und eine ausführliche Behandlung würde leicht den Rahmen dieser Arbeit sprengen.

Es lässt sich zeigen, dass es wieder einen kritischen Punkt gibt, bei dem die Korrelationslänge  $\xi$  divergiert, diesmal allerdings bei einer endlichen und von null verschiedenen Temperatur  $T_c \Leftrightarrow \beta_c < \infty$  mit

$$J\beta_c = \frac{J}{k_B T_c} = \frac{\ln(1 + \sqrt{2})}{2} \approx 0.4404687. \quad (11)$$

Diese wurde erstmalig von Kramers und Wannier 1941 bestimmt [9], also noch vor der Lösung durch Onsager.

Die freie Energie und die Magnetisierung lassen sich nun selbst im Fall  $H \rightarrow 0$  nicht mehr so leicht durch elementare Funktionen darstellen wie im eindimensionalen Fall und man muss zwischen einer Hoch- und einer Tieftemperaturentwicklung unterscheiden, die sich im kritischen Punkt stetig fortsetzen lässt.

Bei der Auswertung der Magnetisierung nach Gleichung (5) im Tieftemperaturregime (also für  $T < T_c \Leftrightarrow \beta > \beta_c$ ) ist nun besondere Vorsicht geboten. Denn anders als im Fall von endlichen Gittern, bei denen der Spinerwartungswert  $\langle s_x \rangle$  und damit die Magnetisierung  $M$  für  $H \rightarrow 0$  verschwindet<sup>3</sup>, existiert hier eine *spontane Magnetisierung*  $\pm M_0$ ,

$$M_0 = \sqrt[8]{1 - \sinh^{-4}(2\beta J)}, \quad (12)$$

deren Vorzeichen durch das Vorzeichen von  $H$ , während  $H$  gegen Null geht, festgelegt ist.

Des Weiteren ergibt sich, dass abseits des kritischen Punktes die Korrelationsfunktion für unendlich groß werdende Abstände zwischen Gitterpunkten zur Null hin abfällt. Aus Gleichung (4) ergibt sich damit für das Tieftemperaturregime

$$\lim_{|x-y| \rightarrow \infty} \langle s_x s_y \rangle \rightarrow \langle s_x \rangle^2 = \langle s_y \rangle^2. \quad (13)$$

<sup>3</sup>Der Grund ist, dass ohne äußeres Magnetfeld zu jedem Zustand  $Z$  ein gleichwahrscheinlicher Zustand existiert, der aus  $Z$  hervorgeht, indem man bei allen Spins das Vorzeichen invertiert.

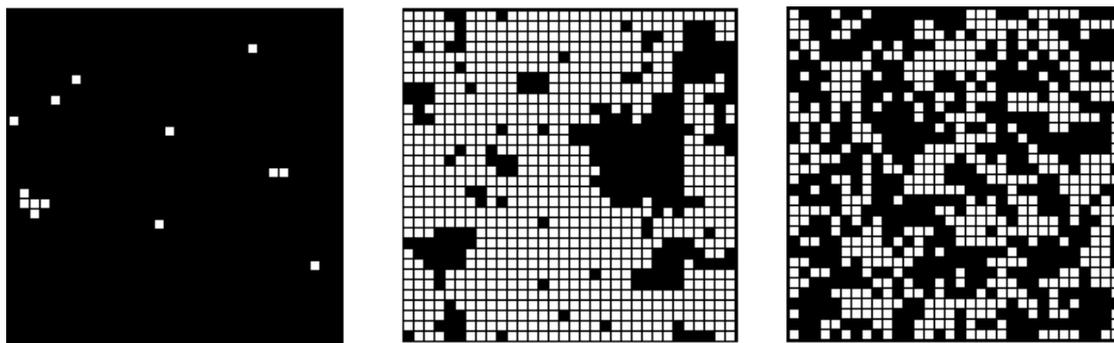


Abbildung 2: Das (hier natürlich nur endlich ausgedehnte) zweidimensionale Ising-Modell in den verschiedenen Zuständen: geordnete Tieftemperaturphase (links), kritischer Punkt (mittig) und ungeordnete Hochtemperaturphase (rechts). Die zwei möglichen Zustände eines Spins sind durch schwarze bzw. weiße Kacheln dargestellt. Erzeugt mit Hilfe von [10].

Für das Hochtemperaturregime, wo die Magnetisierung stets Null ist, fällt die Korrelationsfunktion exponentiell ab. Die Korrelationslänge ist dabei ein Maß dafür, wie schnell dies bei gegebener Temperatur vonstatten geht:

$$\langle s_x s_y \rangle \propto e^{-\frac{|x-y|}{\xi(\beta)}} \xrightarrow{|x-y| \rightarrow \infty} 0. \quad (14)$$

Man erhält also in beiden Temperaturbereichen ein qualitativ grundsätzlich unterschiedliches Verhalten, wie auch in Abbildung 2 zu sehen ist.

Nichtsdestotrotz gilt für die Korrelationslänge  $\xi$  in der Nähe des kritischen Punktes

$$\xi \propto \left| \frac{T_c - T}{T_c} \right|^{-1} \quad (15)$$

unabhängig davon, auf welcher Seite des kritischen Punktes das System sich befindet.

Dieses letzte Resultat aus [8] ist bereits ein Beispiel für einen sogenannten *kritischen Exponenten*, hier  $\nu = 1$ , der das Verhalten eines Systems nahe dem Phasenübergang charakterisiert. Auf diese Exponenten wird im nächsten Abschnitt näher eingegangen.

Das zweidimensionale Ising-Modell zeigt also einen Phasenübergang, bei dem auf der einen Seite des kritischen Punktes eine spontane Magnetisierung existiert, auf der anderen Seite jedoch nicht. Es sei betont, dass dies nur im thermodynamischen Limes, also dem Fall eines unendlich großen Gitters gilt, da sich die Zustandssumme  $\mathcal{Z}$  sonst als endliche Kombination von Exponentialfunktionen darstellen lässt. Diese wäre natürlich analytisch, sodass keine Größe Singularitäten aufweisen könnte.

### 2.1.3. Der Phasenübergang (nicht nur) im Ising-Modell

Ein Beispiel für einen Phasenübergang, der jedermann bekannt sein dürfte, ist der zwischen Eis und flüssigem Wasser bzw. flüssigem Wasser und Wasserdampf. Bei einem solchen Übergang ändert sich der Grad der Ordnung des beteiligten Stoffes, in diesem Fall die Anordnung der Moleküle.

Der strukturell etwas einfachere Phasenübergang zwischen der spontan magnetisierten und der nicht magnetisierten Phase des Ising-Modells im thermodynamischen Limes wurde

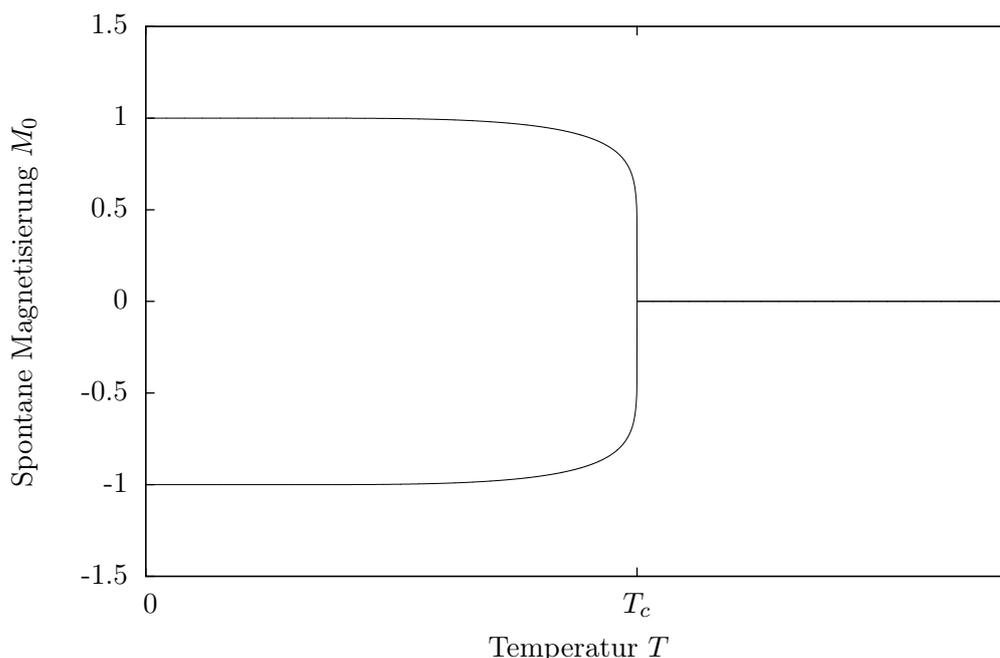


Abbildung 3: Die spontane Magnetisierung verschwindet für  $T > T_c$  und spaltet sich für  $T < T_c$  in zwei symmetrische Äste auf. Die Gestalt ergibt sich aus den Gleichungen (11) und (12)

bereits beschrieben. Weil sich das magnetische Verhalten grundlegend ändert, nennt man  $M$  einen *Ordnungsparameter*. Im Folgenden soll, basierend auf der Abhandlung in [11], die Theorie der Phasenübergänge noch etwas genauer dargestellt werden.

Man nennt den Übergang im Ising-Modell einen Übergang *zweiter Ordnung* oder auch *kontinuierlichen Übergang*, weil der Ordnungsparameter selbst, die Magnetisierung pro Spin  $M$ , an der kritischen Temperatur  $T_c$  zwar noch stetig ist, die Ableitung  $\frac{\partial M}{\partial T}$  aber bereits nicht mehr. Abbildung 3 zeigt die funktionale Abhängigkeit der Magnetisierung von der Temperatur.

Dieser Übergang tritt nicht nur im Ising-Modell so auf. Gemäß der *Universalitätshypothese* [12] gibt es eine ganze Klasse von Modellen, die sich in einer Umgebung des kritischen Punktes, dem sogenannten *kritischen Bereich* gleich verhalten. Konkret besagt diese Hypothese, dass, sobald die Korrelationslänge  $\xi$  genügend groß wird, die effektive Wechselwirkung trotz eigentlich kurzreichweitigen Wechselwirkungen zwischen den Konstituenten des Systems langreichweitig ist. Mehrere Konstituenten des Systems klumpen quasi zusammen und der „Klumpen“ verhält sich wie ein einziger großer Block. In diesem Fall treten die Details in der Hamiltonfunktion, wie z.B. die genaue Struktur des Gitters, in den Hintergrund, und das Verhalten des Systems wird maßgeblich nur noch von drei Bedingungen bestimmt:

- der Dimension  $d$  des Systems,
- der Reichweite der Wechselwirkung (nächster Nachbar im Ising-Modell)
- und der Spindimensionalität (im Ising-Modell ist diese eins).

Diese drei Bedingungen legen die kritischen Exponenten fest und alle Modelle, deren Observablen in der Umgebung des kritischen Punktes den selben Potenzgesetzen gehorchen, werden zu einer *Universalitätsklasse* zusammengefasst. Die vorherrschenden Potenzgesetze sind durch die kritischen Exponenten definiert. Für die Korrelationslänge im zweidimensionalen Ising-Modell wurde der zugehörige Exponent in Abschnitt 2.1.2 bereits erwähnt. Allgemein gilt

$$\xi \propto \left| \frac{T_c - T}{T_c} \right|^{-\nu} \quad (16)$$

Nahe dem kritischen Punkt kann man aber auch

$$T = T_c + \delta T \quad \text{mit} \quad \left| \frac{\delta T}{T_c} \right| \ll 1 \quad (17)$$

schreiben. Damit lässt sich eine Äquivalenz des Temperaturbilds und dem Bild der inversen Temperatur herleiten, denn es ist in erster Näherung

$$\begin{aligned} \frac{\beta - \beta_c}{\beta_c} &= \frac{\frac{1}{T} - \frac{1}{T_c}}{\frac{1}{T_c}} = \frac{T_c(T_c - T)}{T_c T} = \frac{T_c - T_c - \delta T}{T_c + \delta T} = \frac{-\frac{\delta T}{T_c}}{1 + \frac{\delta T}{T_c}} = \\ &= -\frac{\delta T}{T_c} + \mathcal{O}\left(\left(\frac{\delta T}{T_c}\right)^2\right) = \frac{T_c - T_c - \delta T}{T_c} + \mathcal{O}\left(\left(\frac{\delta T}{T_c}\right)^2\right) \approx \frac{T_c - T}{T_c}. \end{aligned} \quad (18)$$

Somit wird Gleichung (16) für infinitesimale Umgebungen des kritischen Punktes gleichwertig zu

$$\xi \propto \left| \frac{\beta - \beta_c}{\beta_c} \right|^{-\nu}. \quad (19)$$

Im zweidimensionalen Ising-Modell im thermodynamischen Limes und Limes  $H \rightarrow 0$  war  $\nu = 1$ . Für das entsprechende dreidimensionale System ist der mit theoretischen Methoden numerisch bestimmte Wert  $\nu \approx 0.63$ .

Es gibt noch weitere Observablen mit zugehörigen kritischen Exponenten, wobei allen die Basis  $|\beta_c - \beta|/\beta_c$  gemein ist. Diese divergieren also bei der kritischen Temperatur. Beispiele wären die isochore Wärmekapazität  $c_V$  mit Exponent  $\alpha$  oder die magnetische Suszeptibilität  $\chi$  mit zugehörigem  $\gamma$ . Diese sind sogar über Skalengesetze miteinander verknüpft, was hier aber nicht weiter ausgeführt wird, da diese Zusammenhänge für die vorliegende Arbeit nur von verschwindender Bedeutung sind.

Die Universalitätshypothese wurde ursprünglich aufgrund der phänomenologischen Erkenntnis, dass sich sehr viele verschiedene Systeme eben in Universalitätsklassen zusammenfassen lassen, aufgestellt. Durch die Wilsonsche Renormierungstheorie wird die Hypothese jedoch theoretisch untermauert. Ihr liegt die Idee der Kadanoff-Konstruktion zugrunde, Spins zu Blöcken zusammenzufassen, die sich zusammen wie Cluster korrelierter Einzelspins verhalten. Ein wichtiges Resultat daraus ist, dass die Exponenten davon unabhängig sind, aus welcher Richtung man sich dem kritischen Punkt nähert. Eine tiefer gehende Behandlung ist im Rahmen dieser Arbeit allerdings nicht möglich. Für eine kurze Einführung zur Renormierungsgruppe am Ising-Modell siehe zum Beispiel [13].

Es gibt auch noch weitere Theorien zu Phasenübergängen, wie die klassische Landau-Theorie oder die Molekularfeldnäherung. Diese beruhen jedoch auf teils sehr stark einschränkenden Annahmen, aufgrund derer sie für die hier interessierenden Systeme nur schlechte bis falsche Vorhersagen liefern.

### 2.1.4. Finite Size Effekte

Die bisher aufgelisteten Ergebnisse gelten alle nur im thermodynamischen Limes. Wenn eine analytische Lösung nicht verfügbar ist, was in den meisten Fällen der Fall ist, dann muss man sich mit numerischen Methoden behelfen. Diese können jedoch prinzipiell nur endliche Systeme beschreiben, daher ist es wichtig zu verstehen, wie sich ein endliches Gitter auf die obigen Erkenntnisse auswirkt.

Der allerwichtigste Aspekt ist, dass der Phasenübergang verschwindet, das heißt, die Korrelationslänge ist auf einen endlichen Wert, genauer gesagt die Größe des Gitters, beschränkt. Darüber hinaus verschwindet, wie bereits angemerkt, die spontane Magnetisierung, sodass die Magnetisierung pro Spin nun als

$$M = \left\langle \left| \frac{1}{V} \sum_{x \in \Lambda} s_x \right| \right\rangle \quad (20)$$

definiert werden muss, um mindestens einen der beiden Äste in Abbildung 3 zu erhalten. Der Knick, den die Magnetisierung eigentlich aufweist, schmilzt aus und die Ableitung wird stetig.

Dennoch ist ein endliches System eine gute Näherung für ein unendliches System, solange die Korrelationslänge viel kleiner als die Systemausdehnung ist, also abseits des kritischen Punktes. Diese Aussage gilt vor allem dann, wenn periodische Randbedingungen den Einfluss, den die sonst am Rand sitzenden Spins hätten, auf ein Minimum reduzieren.

Ein weiterer Effekt ist, dass die Position des (nun pseudo-)kritischen Punktes, bei dem  $\xi(\beta)$  eine abgerundete Spitze hat, sich in Abhängigkeit der Seitenlänge  $L$  des Gitters verändert. Gemäß [14] findet man die *Skalierungsrelation*:

$$\left| \frac{T_c(L) - T_c(L = \infty)}{T_c(L = \infty)} \right| \propto L^{-\frac{1}{\nu}} \xrightarrow{L \rightarrow \infty} 0. \quad (21)$$

Bestimmt man nun den pseudokritischen Punkt für verschieden große Gitter, kann man daraus den kritischen Exponenten des unendlich großen Gitters extrapolieren.

## 2.2. Die Monte-Carlo-Methode

Für das dreidimensionale Ising-Modell ist bisher keine analytische Lösung bekannt und aufgrund der eingangs erwähnten NP-Vollständigkeit ist fraglich, ob diese in absehbarer Zeit gefunden wird. Nichtsdestotrotz geben eine Vielzahl Ergebnisse numerischer Untersuchungen Einblick in das Verhalten des dreidimensionalen Systems, das wie das zweidimensionale auch einen Phasenübergang zweiter Ordnung bei ungefähr halbem  $\beta_c$  aufweist. Die populärste, wenn auch eine der anspruchsvollsten dieser numerischen Methoden, ist die *Monte-Carlo-Simulation*, ein stochastischer Ansatz, der den Zustandsraum mit Hilfe von Zufallszahlen erkundet und mit statistischen Fehlern behaftete Ergebnisse liefert.

Andere, exakte Methoden wie das stumpfe Auflisten aller möglichen Konfigurationen oder kombinatorische Ansätze scheitern sehr schnell an der Komplexität selbst relativ kleiner Systeme. Man erinnere sich nur daran, dass für das Ising-Modell die Zahl der möglichen Zustände gemäß  $2^V$  mit der Zahl der Gitterplätze  $V$  skaliert. Für ein nur  $7 \times 7 \times 7$  großes System sind das bereits mehr als  $10^{100}$  mögliche Zustände, was schon wesentlich mehr ist als die von Eddington auf  $\approx 10^{80}$  geschätzte Anzahl an Protonen und Neutronen im beobachtbaren Universum!

Die Monte-Carlo-Methode erlaubt die Bestimmung der Erwartungswerte von Observablen eines thermodynamischen Ensembles. Dazu werden, hier im Fall des Ising-Modells<sup>4</sup>, möglichst viele verschiedene, zufällige Zustände erzeugt, in denen sich das System befinden kann.

Im einfachsten Fall könnte man Zufallszahlen benutzen um sehr viele verschiedene Zustände zu erzeugen und die interessierenden Observablen, beispielsweise die Energie, die Magnetisierung oder die Korrelationsfunktion, in diesen Zuständen berechnen. Den Erwartungswert erhält man durch Mittelung über die so erhaltenen Werte, über die Varianz lassen sich Standardabweichung und -fehler bestimmen. Das Problem an diesem Vorgehen ist, dass es äußerst ineffizient ist und der Fehler nur sehr langsam im Gesetz der großen Zahlen gegen Null konvergiert.

Eine erste Verbesserung stellt das sogenannte *Importance Sampling* dar, bei dem nicht alle Zustände mit der gleichen Wahrscheinlichkeit, sondern gemäß einer Wahrscheinlichkeitsverteilung erzeugt werden. Für die Auswertung von Erwartungswert und Varianz müssen die einzelnen Stichproben dann entsprechend der Verteilung gewichtet werden. Wenn die Wahrscheinlichkeitsverteilung, die zum Erzeugen der Systemzustände benutzt wird qualitativ ähnlich der tatsächlichen Verteilung ist, dann konvergiert das Verfahren wesentlich schneller. Ein Problem ist jedoch, dass die zu gegebener Temperatur zu erwartende Wahrscheinlichkeitsverteilung in der Regel nicht leicht einzuschätzen ist. In diesem Fall ist das beschriebene Verfahren, *direktes Sampling* genannt, nicht sinnvoll anwendbar, und ein etwas komplizierterer Ansatz ist notwendig.

### 2.2.1. Markov-Ketten-Monte-Carlo

Die Ausführungen in diesem Kapitel stützten sich bisher auf [15], von nun an folgt die Abhandlung aber [1], vgl. auch [16]. Bei der Methode des direkten Samplings wurde eine Folge von  $n$  statistisch unabhängigen Zufallsvariablen  $X_i$  mit  $1 \leq i \leq n$  erzeugt. Wenn man die Unabhängigkeit der Folgenglieder durch Abhängigkeit vom (und nur vom) direkt vorhergehenden Folgenglied ersetzt<sup>5</sup>, erhält man eine sogenannte *Markov-Kette*.

Manchmal wird der Index der einzelnen Folgenglieder als „Zeit“ bezeichnet, und der Übergang von einem Glied  $X_i$  zu einem neuen Glied  $X_{i+1}$  als „Evolution“ der Markov-Kette. Dies hat jedoch nur etwas mit der Ausführung des erzeugenden numerischen Algorithmus und nichts mit dem physikalischen Zeitbegriff des Systems zu tun, insbesondere falls das untersuchte System sich im thermodynamischen Gleichgewicht befindet, in welchem eine Zeitangabe ohnehin nicht viel Sinn macht.

Ziel ist es nun, einen Algorithmus zu konstruieren, der eine Markov-Kette erzeugt, die sich im Limes unendlicher Länge der Wahrscheinlichkeitsverteilung des untersuchten Systems annähert, um dann durch Mittelung wieder den Erwartungswert der Observablen bestimmen zu können. Dazu ist es hilfreich, zu überlegen, dass Markov-Ketten sich durch ein vorgegebenes Anfangsglied  $X_1$  und eine  $2^V \times 2^V$  Übergangsmatrix  $P = (p(a \rightarrow b))_{ab}$  definieren lassen.  $a, b \in S$  sind dabei beliebige Elemente des Zustandsraums des Systems. Es muss gelten, dass alle Matrixelemente  $p(a \rightarrow b) \geq 0$  sind und  $\sum_b p(a \rightarrow b) = 1$  ist. Dann lässt sich  $p(a \rightarrow b)$  als Wahrscheinlichkeit interpretieren, dass die Markov-Kette vom Zustand  $X_i = a$  nach  $X_{i+1} = b$  übergeht. Die beiden Forderungen stellen sicher, dass die Wahrscheinlichkeit nicht negativ ist und dass die Wahrscheinlichkeit, von einem beliebigen Zustand  $a$  aus in irgendeinem Zustand  $b$  zu landen, genau 1 ist. Man kann

<sup>4</sup>Allgemein kann das Monte-Carlo-Verfahren dazu benutzt werden, um (schwierige, vor allem höherdimensionale) Integrale auszuwerten.

<sup>5</sup>Diese Eigenschaft wird auch als Gedächtnislosigkeit der Markov-Kette bezeichnet.

zeigen, dass eine Markov-Kette zwei bestimmte Eigenschaften erfüllen muss, damit sich die Kette einer gegebenen Wahrscheinlichkeitsverteilung  $\pi = \{\pi_a | \pi_a \text{ ist Wahrscheinlichkeit für den Zustand } a\}$  annähert, nämlich Ergodizität und Gleichgewichtserhaltung<sup>6</sup>. Diese bedeuten insbesondere:

**Ergodizität:**

$$\forall a, b \in S \exists n \in \mathbb{N}, c_i \text{ mit } 1 < i < n : p(a \rightarrow c_1)p(c_1 \rightarrow c_2) \dots p(c_{n-1} \rightarrow b) > 0 \quad (22)$$

Oder mit anderen Worten: für alle Zustände  $a$  und  $b$  aus  $S$  existiert ein natürliches  $n$ , sodass der Übergang von  $a$  nach  $b$  in  $n$  Schritten über Zwischenzustände  $c_i$  mit einer von null verschiedenen Gesamtwahrscheinlichkeit möglich ist.

**Gleichgewichtserhaltung:**

$$\forall b \in S : \sum_a \pi_a p(a \rightarrow b) = \pi_b \quad (23)$$

Eine hinreichende, aber nicht notwendige Bedingung um diese Eigenschaft zu erfüllen ist das sogenannte *detaillierte Gleichgewicht*

$$\forall a, b \in S : \pi_a p(a \rightarrow b) = \pi_b p(b \rightarrow a), \quad (24)$$

was so viel bedeutet wie: die Wahrscheinlichkeit im Zustande  $a$  zu sein und zum Zustand  $b$  überzugehen hängt nicht von der Richtung des Übergangs ab. Die Implikation folgt direkt aus der Forderung, dass die Wahrscheinlichkeit aus einem Zustand  $b$  in irgendeinen Zustand  $a$  zu kommen genau eins sein muss:

$$\sum_a \pi_a p(a \rightarrow b) = \sum_a \pi_b p(b \rightarrow a) = \pi_b. \quad (25)$$

Nachdem nun bekannt ist, wie ein Algorithmus zu konstruieren ist, der korrekt arbeitet, verbleibt noch die Frage nach einem Algorithmus, der auch halbwegs effizient ist. Dies sei Gegenstand des nächsten Abschnittes.

**2.2.2. Der Metropolis-Algorithmus**

Der Standardalgorithmus um Markov-Ketten zu erzeugen wurde bereits 1953 von Metropolis et al. entwickelt [17]. Für das Ising-Modell bei  $H = 0$  im Zustand  $a$  wird ein zufälliger Spin an der Stelle  $x$  ausgewählt und invertiert. Dies führt mit einer gewissen Wahrscheinlichkeit zum neuen Zustand  $b$ . Die Übergangswahrscheinlichkeit sei definiert als

$$p(a \rightarrow b) = \min \left( 1, \frac{\pi_b}{\pi_a} \right) = \min \left( 1, e^{-(\mathcal{H}(b) - \mathcal{H}(a))} \right) = \min \left( 1, e^{\left( 2\beta J \sum_{|x-y|=1} s_x s_y \right)} \right). \quad (26)$$

Bei den Umformungen wurde ausgenutzt, dass sich die Zustandssummen herauskürzen und sich nur an der Stelle  $x$  genau ein Spin ändert.

Es ist leicht zu sehen, dass der Algorithmus ergodisch ist, weil jeder Spin zufällig mit Wahrscheinlichkeit  $\frac{1}{V}$  ausgewählt werden kann, und die Übergangswahrscheinlichkeit

<sup>6</sup>Man nennt diese beiden auch Irreduzibilität und Stationarität.

**Data:** Anfangszustand  $a_0$   
**Result:** Markov-Kette  $a_1, \dots, a_n$

```

1 for  $i \leftarrow 1$  to  $n$  do
2    $x \leftarrow$  zufällige Gitterstelle;
3    $s_x \leftarrow -s_x$ ;
4    $p \leftarrow \exp \left( 2\beta J \sum_{\substack{y \in \Lambda \\ |x-y|=1}} s_x s_y \right)$ ;
5   if  $p \geq 1 \vee p > \text{Zufallszahl\_aus}([0;1])$  then
6      $a_i \leftarrow$  neue Konfiguration
7   end
8   else
9      $a_i \leftarrow$  alte Konfiguration
10  end
11 end

```

**Algorithmus 1:** Der Metropolis-Algorithmus für das Ising-Modell

endlich ist. Auch die detaillierte Gleichgewichtsbedingung ist erfüllt, wie sich leicht durch Fallunterscheidung  $\pi_a$  größer bzw. kleiner  $\pi_b$  in

$$\pi_a \min \left( 1, \frac{\pi_b}{\pi_a} \right) = \pi_b \min \left( 1, \frac{\pi_a}{\pi_b} \right) \quad (27)$$

nachprüfen lässt.

Der Algorithmus, der eine Markov-Kette der Länge  $n$  erzeugt, ist in Algorithmus 1 als Pseudocode dargestellt. Die Auswahl, welcher Zustandsübergang genau stattfindet, wird also durch *Zufallszahlen* realisiert. Ist die im Intervall  $[0; 1)$  generierte Zufallszahl kleiner als die Übergangswahrscheinlichkeit, so wird der vorgeschlagene Zustandsübergang akzeptiert, ansonsten wird er zurückgewiesen und der Zustand ändert sich nicht. Die Qualität der Ergebnisse hängt demnach mit der Qualität des Zufallszahlengenerators zusammen. Die Theorie der (Pseudo-)Zufallszahlen und ihrer Generierung ist ein weites und nichttriviales Feld, das hier jedoch nicht weiter behandelt werden kann.

Der Metropolis-Algorithmus liefert prinzipiell korrekte Ergebnisse, allerdings sind die Glieder der resultierenden Markov-Kette stark korreliert. Welche negativen Folgen daraus entstehen, sei im nächsten Abschnitt erläutert.

### 2.2.3. Systematische und statistische Fehler

Monte-Carlo-Simulationen liefern statistische, fehlerbehaftete Ergebnisse zu numerischen Fragestellungen. In vielerlei Hinsicht hat die Durchführung etwas mit der Durchführung eines gewöhnlichen Experimentes gemein, nur dass sich der gesamte „Messaufbau“ im Computer befindet. Daher ist es nur natürlich, die selben, strengen Maßstäbe zur Abschätzung des Fehlers der Resultate anzulegen wie in der Experimentalphysik[18]. Die Darstellung in diesem Abschnitt folgt ab jetzt im Wesentlichen [19].

*Systematische Fehler* können auf zwei Arten in das Ergebnis eingehen. Ein Entstehungsgrund sind große Autokorrelationen<sup>7</sup> bzw. zu kurze Messreihen. Diese Umstände werden

<sup>7</sup>Man beachte, dass diese trotz ähnlichem Namens abgesehen von der mathematischen Struktur nichts mit der bisher betrachteten Korrelation zwischen Spins auf dem Gitter zu tun haben!

bald erklärt, zunächst jedoch sei das Augenmerk auf die andere Möglichkeit gerichtet, systematische Fehler zu erhalten.

In der Regel sind bestimmte Observablen in einem System von Interesse, das sich bei einer vorgegebenen Temperatur im thermodynamischen Gleichgewicht befindet. Nun ist es allgemein nicht möglich, einen geeigneten Anfangszustand vorzugeben. Im Fall des Ising-Modells hat man typischerweise zwei Möglichkeiten: entweder alle Spins parallel ausgerichtet oder jeder einzelne Spin zufällig. Diese Zustände erwartet man bei niedrigen bzw. hohen Temperaturen, daher werden diese Anfangsbedingungen oft Hot bzw. Cold Start genannt. Offensichtlich sind in diesem Fall gewonnene Messwerte nicht repräsentativ für die interessierende Temperatur des Systems. Daher sollte eine gewisse Mindestanzahl an *Sweeps*, das sind Intervalle der Markov-Kette, bei denen ungefähr jeder Spin des Gitters einmal den Zustand gewechselt hat, zu Beginn aussortiert werden. Je nachdem, wie sehr die Kettenglieder korreliert sind, wie groß das System ist, und wie nah sich die Temperatur an einem kritischen Punkt befindet, können davon mehr oder weniger notwendig sein. Um abzuschätzen, wann das System im Gleichgewichtszustand angekommen ist<sup>8</sup>, kann man beispielsweise in beiden möglichen Startkonfigurationen starten und beobachten, ab wann die interessierende Observable für beide Startbedingungen ähnliche Werte annimmt. Bei Kenntnis der Autokorrelation, die nun im Rahmen der *statistischen Fehler* besprochen werden soll, kann man auch genauere Angaben zur benötigten Equilibrierungszeit machen.

Es sollte zunächst beleuchtet werden, wie man an die Ergebnisse kommt, deren statistische Fehler man analysieren möchte. Bekannt ist bereits, wie eine Markov-Kette der Länge  $N$  von Zuständen  $X_i$  des Systems zu erzeugen ist. Sei  $O$  eine Observable des Systems, etwa die durch die Hamiltonfunktion gegebene Energie. Diese hat im thermodynamischen Gleichgewicht den Erwartungswert  $\langle O \rangle$  und Varianz  $\sigma_O = \langle (O - \langle O \rangle)^2 \rangle$ . Ein geeigneter *Schätzer* ist nun für den Erwartungswert von  $O$  durch

$$\hat{O} = \frac{1}{N} \sum_{i=1}^N O(X_i) \quad (28)$$

und für die Varianz durch

$$\hat{\sigma}_O^2 = \frac{1}{N-1} \sum_{i=1}^N (O(X_i) - \hat{O})^2 \quad (29)$$

gegeben.  $O(X_i)$  wird als *Sample* zur „Zeit“  $i$  bezeichnet.

Falls die Glieder der Markov-Kette unkorreliert sind, wie beim direkten Sampling, kann man das Ergebnis mit statistischem Fehler direkt angeben:

$$\langle O \rangle \pm \sigma = \hat{O} \pm \frac{\hat{\sigma}_O}{\sqrt{N}}. \quad (30)$$

Besondere Beachtung verdient an dieser Stelle der Faktor  $\frac{1}{\sqrt{N}}$  im obigen Ausdruck. Darin äußert sich eine ganz allgemeine und zentrale Eigenschaft von Monte-Carlo-Methoden. Die Qualität der Ergebnisse skaliert bestenfalls mit der reziproken Wurzel des Budgets an Rechenzeit. Daher macht es nur Sinn, diese Methode einzusetzen, wenn alle anderen möglichen Ansätze noch ineffizienter sind.

Falls die einzelnen Glieder der Markov-Kette nennenswert korreliert sind, erhält man zwar immer noch den richtigen Erwartungswert, aber der statistische Fehler in Gleichung

<sup>8</sup>Man sagt dazu auch, dass das System equilibriert ist.

(30) unterschätzt den wahren Fehler. Um quantifizieren zu können, was „nennenswert“ bedeutet, benötigt man die *Autokorrelationsfunktion*

$$C(O(X_i), O(X_{i+t})) = \langle (O(X_i) - \langle O(X_i) \rangle)(O(X_{i+t}) - \langle O(X_{i+t}) \rangle) \rangle =: C_O(t). \quad (31)$$

Man kann eine mit  $C_O(0) = \sigma_O^2$  normalisierte Variante einführen,

$$\Gamma_O(t) = \frac{C_O(t)}{C_O(0)} \propto e^{-\frac{t}{\tau_{O, \text{exp}}}} + \text{h.o.t.}, \quad (32)$$

die in erster Ordnung typischerweise exponentiell abfällt. Man nennt  $\tau_{O, \text{exp}}$  die *exponentielle Autokorrelationszeit* zur Observablen  $O$ . Die Korrelation der Werte führt zu systematischen Fehlern der Größenordnung  $\mathcal{O}(\exp(-\frac{t}{\tau_{\text{exp}}}))$ , wenn zwischen dem Aufnehmen der Samples je  $t$  Kettenglieder ausgelassen werden.

Außerdem kann man die *integrierte Autokorrelationszeit*

$$\tau_{O, \text{int}} = \frac{1}{2} + \sum_{t=1}^N \Gamma_O(t) \quad (33)$$

definieren und unter Anwendung einiger Sätze aus der Stochastik zeigen, dass sich die Zahl unabhängiger Samples auf

$$N_{\text{unabh.}} = \frac{N}{2\tau_{O, \text{int}}} \quad (34)$$

verringert und der tatsächliche Fehler durch

$$\sigma_{O, \text{korr.}}^2 = 2\tau_{O, \text{int}} \sigma_O^2 \quad (35)$$

gegeben ist.

Jedoch ist es in der Regel recht schwierig, die Autokorrelationszeit  $\tau \approx \tau_{\text{int}} \approx \tau_{\text{exp}}$  präzise zu bestimmen, da üblicherweise mehr als  $1000\tau$  Samples nötig sind, um  $\tau$  selbst abzuschätzen. Wenn man schließlich einen Wert für  $\tau$  kennt, so kann man in der Equilibrierungsphase je nach Anforderung die ersten  $20\tau$  bis  $1000\tau$  auslassen. Für Ergebnisse mit 1% Genauigkeit wird mindestens eine Anzahl Samples in der Größenordnung  $10^5\tau$  oder besser benötigt.

#### 2.2.4. Critical Slowing Down

Zuletzt sei in diesem Kapitel noch erwähnt, wie die Autokorrelationszeit von den Parametern des untersuchten Systems abhängt. Für Gittermodelle hat sich empirisch herausgestellt, dass sie polynomiell mit der Korrelationslänge  $\xi_O$  zur Observablen  $O$  skaliert:

$$\tau_{O, \text{int}} \propto (\xi_O)^z. \quad (36)$$

$z$  wird als *dynamischer kritischer Exponent*<sup>9</sup> bezeichnet und hängt vom verwendeten Algorithmus ab. In der Nähe des kritischen Punktes skaliert  $\xi$  bekanntlich mit dem Gittervolumen, sodass für ein Gitter mit Seitenlänge  $L$  der nötige Rechenaufwand gemäß  $L^z$  anwachsen kann.

Der Metropolis-Algorithmus ist mit einem Exponenten  $z \approx 2$  besonders anfällig für dieses *Critical Slowing Down* genannte Phänomen. In späteren Abschnitten des folgenden Kapitels werden alternative Algorithmen besprochen, bei denen das Problem mit  $z < 1$  viel weniger ausgeprägt ist [20].

<sup>9</sup>Diese Namensgebung ist etwas unglücklich gewählt. Kein Zusammenhang besteht zu den kritischen Exponenten aus Abschnitt 2.1.2!

### 3. Problemstellung und Herangehensweise

Gegenstand der Untersuchung in dieser Arbeit ist, wie sich die Eigenschaften des Ising-Modells verändern, wenn man Schritt für Schritt von einem endlichen, effektiv zweidimensionalen, also planaren Gitter zu einem entsprechenden dreidimensionalen, kubischen Gitter übergeht. Als ein Kompromiss, zwischen geringen Finite Size Effekten und erträglicher Rechenzeit wird hierzu ein  $64 \times 64 \times T$ -Gitter gewählt, wobei  $T$  in diskreten Abständen von 1 nach 64 läuft.

Die zu bestimmenden Größen, die das System charakterisieren, seien der kritische Exponent der Korrelationslänge  $\nu$  und die pseudokritische Temperatur  $\beta_c$  in Abhängigkeit vom Parameter  $T$ . Dieses Kapitel beschäftigt sich mit der Art und Weise, wie die beiden Größen aus der Korrelationsfunktion  $g(x, y)$  in der Hochtemperaturphase numerisch bestimmt werden können.

Bevor es jedoch zur Sache geht, noch ein paar Festlegungen vorneweg: von nun an gelten  $k_B = 1$ ,  $J = 1$  und  $H = 0$ .  $T$  bezeichnet ab jetzt nur noch die Ausdehnung des Ising-Gitters in der ausgezeichneten Richtung und ist nicht mit der Temperatur zu verwechseln. Für letztere wird ab jetzt nur noch die reziproke Darstellung  $\beta$  benutzt werden!

#### 3.1. Numerische Bestimmung von $\nu$ und $\beta_c$

Weil sowohl  $\nu$  als auch  $\beta_c$  von den genauen Seitenlängen des betrachteten Gitters abhängen, müssen beide für jedes untersuchte Gitter erneut bestimmt werden. Dies ist aus „Messwerten“ (numerischen Ergebnissen) für die Korrelationslänge  $\xi$ , die in der Hochtemperaturphase<sup>10</sup> nahe dem kritischen Punkt gewonnen werden, wie folgt möglich. Gleichung (19) lautet für  $\beta \lesssim \beta_c$  und bei Einführung eines konstanten Vorfaktors  $\xi_0$ :

$$\xi \approx \xi_0 \left( \frac{\beta_c - \beta}{\beta_c} \right)^{-\nu}. \quad (37)$$

Davon kann man zuerst den Logarithmus

$$\ln(\xi) \approx \ln(\xi_0) - \nu (\ln(\beta_c - \beta) - \ln(\beta_c)) \quad (38)$$

und dann die Ableitung

$$\frac{1}{\xi} \frac{\partial \xi}{\partial \beta} = \frac{\partial \ln(\xi)}{\partial \beta} \approx -\nu \frac{-1}{\beta_c - \beta} \quad (39)$$

bilden. Bildung des Kehrwerts schließlich führt auf die Geradengleichung

$$\xi \left( \frac{\partial \xi}{\partial \beta} \right)^{-1} \approx \frac{1}{\nu} (\beta_c - \beta), \quad (40)$$

mit der sich  $\nu$  und  $\beta_c$  aus der Steigung bzw. dem Schnittpunkt mit der  $x$ -Achse bestimmen lassen.

Da ein geschlossener Ausdruck weder für  $\xi(\beta)$  noch für die Ableitung  $\xi'(\beta)$  bekannt ist, muss letztere auch numerisch bestimmt werden, indem man zu jedem  $\xi(\beta)$  noch die

<sup>10</sup>Der Grund, weswegen die Hochtemperaturphase zur numerische Bestimmung benutzt wird, ist schlicht, dass die Messwerte in der Tieftemperaturphase wesentlich schwieriger zu erhalten sind. Der Grund dafür wird im nächsten Abschnitt klarer werden.

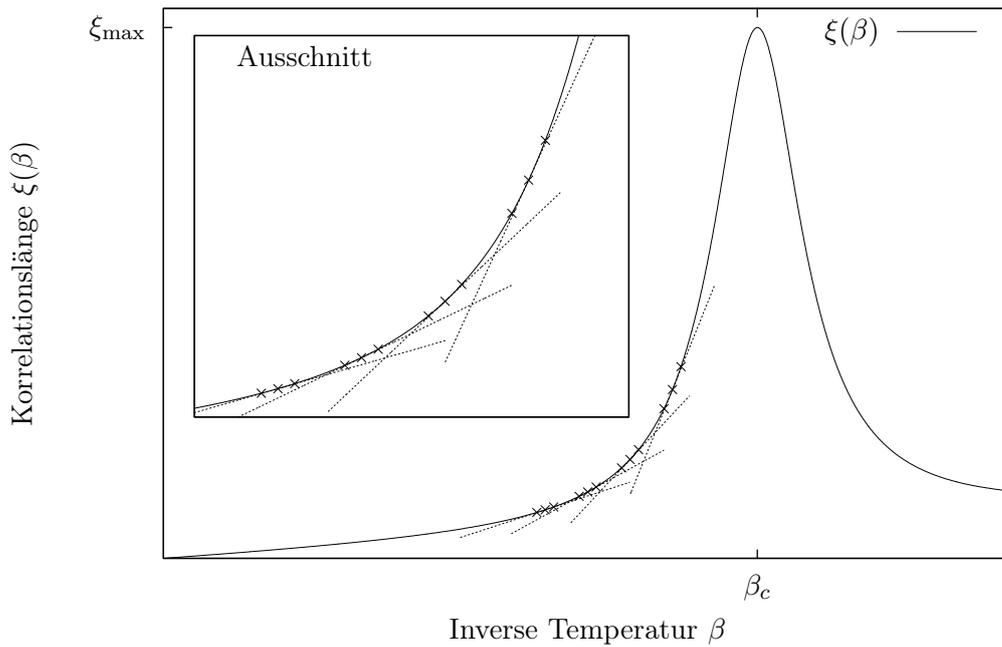


Abbildung 4: Ungefährer Verlauf von  $\xi(\beta)$  und Verdeutlichung der Wahl der  $\beta$ , für die die Korrelationslängen samt Ableitung (angedeutet durch die gestrichelten Geraden) bestimmt werden sollen.

Werte von  $\xi(\beta - \Delta\beta)$  und  $\xi(\beta + \Delta\beta)$  für ein kleines  $\Delta\beta$  erfasst. Der Ausdruck für die zentrale Differenzmethode [21] ist

$$\xi'(\beta) = \frac{\xi(\beta + \Delta\beta) - \xi(\beta - \Delta\beta)}{2\Delta\beta} - \frac{(2\Delta\beta)^2}{3!} \xi'''(\tilde{\beta}) \approx \frac{\xi(\beta + \Delta\beta) - \xi(\beta - \Delta\beta)}{2\Delta\beta}, \quad (41)$$

wobei im vernachlässigten Restglied das  $\tilde{\beta}$  nicht genauer als  $\tilde{\beta} \in [\beta - \Delta\beta; \beta + \Delta\beta]$  bekannt ist.

Abbildung 4 veranschaulicht, wie der allgemeine, in Wahrheit unbekanntes Verlauf von  $\xi(\beta)$  zu erwarten ist und bei welchen Werten für  $\beta$  die einzelnen  $\xi(\beta)$  zu bestimmen sind. Die  $\beta$  sind so zu wählen, dass  $1 \ll \xi(\beta) \ll 64$  gilt. Die obere Grenze ist bedingt durch sonst auftretende Finite Size Effekte wegen der Abrundung der Spitze am pseudokritischen Punkt, wohingegen die untere Grenze die Gültigkeit der Näherungsgleichung (19) sicher stellt.

Der Fehler  $\Delta\xi'$ , mit dem das Ergebnis aus Gleichung (41) vorliegt, setzt sich aus zwei Komponenten  $\Delta\xi'_1 + \Delta\xi'_2$  zusammen. Die eine ist der statistische Fehler

$$\Delta\xi'_1(\beta) = \frac{\sqrt{(\Delta\xi(\beta - \Delta\beta))^2 + (\Delta\xi(\beta + \Delta\beta))^2}}{2\Delta\beta} \approx \frac{\Delta\xi(\beta)}{\sqrt{2}\Delta\beta}, \quad (42)$$

der bei der Bestimmung der einzelnen Korrelationslängen mit der Monte-Carlo-Methode auftritt. Der andere folgt aus der Vernachlässigung des Restgliedes

$$\Delta\xi'_2(\beta) = \frac{2(\Delta\beta)^2}{3} \xi'''(\tilde{\beta}) \quad (43)$$

in Gleichung (41). Der Fehler, der durch die endliche Maschinengenauigkeit verursacht wird, ist wesentlich kleiner als diese beiden und kann daher vernachlässigt werden.

Man kann eine optimale Schrittweite  $\Delta\beta$  finden, bei der der Gesamtfehler minimal wird. Dazu leitet man den Gesamtfehler nach der Schrittweite  $\Delta\beta$  ab,

$$\frac{\partial\Delta\xi'}{\partial(\Delta\beta)} = \frac{\partial}{\partial(\Delta\beta)} \left( \frac{\Delta\xi}{\sqrt{2}\Delta\beta} + \frac{2(\Delta\beta)^2}{3}\xi''' \right) = \frac{4\Delta\beta}{3}\xi''' - \frac{\Delta\xi}{\sqrt{2}(\Delta\beta)^2} \quad (44)$$

und findet die Nullstelle der Ableitung, die bei

$$\Delta\beta_{\text{opt}} = \sqrt[3]{\frac{3\Delta\xi}{4\sqrt{2}\xi'''}} \quad (45)$$

liegt. Die dritte Ableitung in diesem Ausdruck ist allerdings nicht für die einzelnen  $\xi(\beta)$  bekannt, da zur Bestimmung mindestens vier Werte in gleichem Abstand bezüglich  $\beta$  benötigt würden. Für die erste Ableitung liegen aber nur der linke und rechte Nachbar zu jedem  $\xi(\beta)$  vor. Nichtsdestotrotz lässt sich bei Betrachtung von je vier Samples, die in Abbildung 4 in den großen Abständen zueinander liegen, die Größenordnung der dritten Ableitung und damit die der optimalen Schrittweite abschätzen. Dem liegt die Annahme zugrunde, dass  $\xi(\beta)$  ungefähr dem in der Abbildung angedeuteten Verlauf folgt, sodass bezüglich der dritten Ableitung in dem untersuchten Temperaturbereich noch keine unangenehmen Überraschungen zu erwarten sind.

### 3.2. Die Zeitscheibenkorrelationsfunktion

Nachdem nun bekannt ist, wie man aus den numerischen Werten für  $\xi(\beta)$  den kritischen Exponenten und die kritische Temperatur extrahiert, stellt sich natürlich die Frage, wie man überhaupt zu den Werten für  $\xi(\beta)$  gelangt. Die dazu verwendete Methode wurde bereits in einer Diplomarbeit [16] zu einem ähnlichen Thema in der gleichen Arbeitsgruppe benutzt, wo sich auch eine Herleitung mit Hilfe des Transfermatrixformalismus findet.

Zunächst wird eine Funktion  $\tilde{c}(x, y)$  definiert, die die Korrelationsfunktion  $g(x, y)$  zwischen den Gitterstellen  $x$  und  $y$  über alle Komponenten der Vektoren  $x$  und  $y$  bis auf die jeweils ersten Komponenten  $x_1$  und  $y_1$  aufsummiert:

$$\tilde{c}(x, y) = \sum_{\substack{x' \in \Lambda: x'_1 = x_1 \\ y' \in \Lambda: y'_1 = y_1}} \langle s_{x'} s_{y'} \rangle. \quad (46)$$

Da nur endliche Systeme auf diese Weise untersucht werden können verschwindet die Magnetisierung pro Spin  $\langle s_x \rangle$  nämlich. Man sagt, die Menge der Vektoren, die nur durch die erste Komponente festgelegt sind, bilden eine *Zeitscheibe*. Abbildung 5 zeigt, wie das im dreidimensionalen Fall für das  $L \times L \times T$ -Gitter, welches in dieser Arbeit von Interesse ist, aussieht. Speziell ist zu sehen, dass über die variable Seitenlänge  $T$  summiert wird!

$\tilde{c}(x, y)$  hängt also nur von den Komponenten  $x_1$  und  $y_1$  ab, sodass man auch  $\tilde{c}(x_1, y_1)$  schreiben kann. Wegen der Translationsinvarianz aufgrund der periodischen Randbedingungen reduziert sich die Abhängigkeit sogar auf den Abstand  $t = |x_1 - y_1|$ , sodass man über die gesamte Länge  $L_1$ , das ist die Ausdehnung des Gitters senkrecht zu den Zeitscheiben, mitteln kann. Wenn man dann noch mit dem Gittervolumen  $V$  normiert,

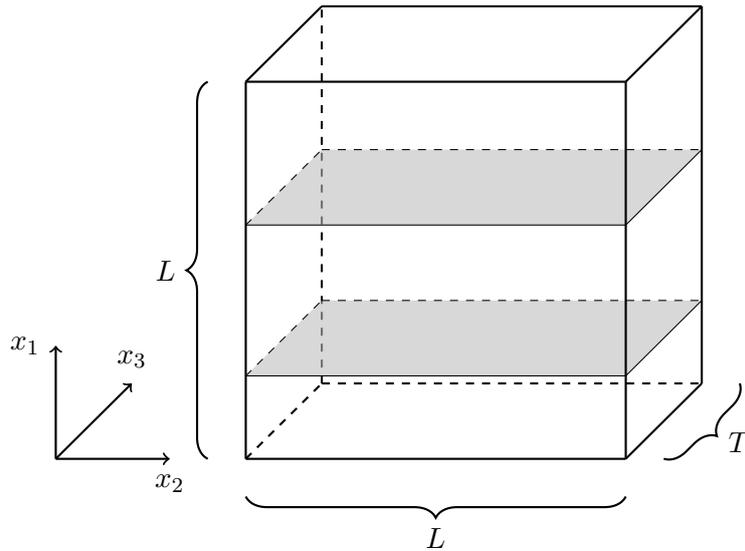


Abbildung 5: Lage der Zeitscheiben im  $L \times L \times T$ -Gitter.

erhält man die *Zeitscheibenkorrelationsfunktion*<sup>11</sup>

$$c(t) = \frac{1}{VL_1} \sum_{i=1}^{L_1} \tilde{c}(x_i, x_{i+t}). \tag{47}$$

Man kann zeigen [23], dass  $c(t)$  sich im Gegensatz zu  $g(x, y)$  im endlichen Fall als Reihe von Exponentialfunktionen mit konstanten Koeffizienten  $c_i$  entwickeln lässt,

$$c(t) = \sum_i c_i \left( e^{-m_i t} + e^{-m_i(L_1-t)} \right). \tag{48}$$

Ein weiterer Unterschied zu  $g(x, y)$  ist, dass bezüglich  $c(t)$  eine Richtung ausgezeichnet ist. Im konkret untersuchten Fall kann daher  $t$  immer von 0 bis  $L_1 - 1 = 63$  variiert werden und ist nicht durch die variable Seitenlänge  $T$  beschränkt.

Interessant an Gleichung (48) ist, dass der Koeffizient im ersten Exponenten  $m_1$  genau der reziproken Korrelationslänge, also  $\frac{1}{\xi}$  entspricht und die Koeffizienten in den höheren Summengliedern alle wesentlich größer als  $m_1$  sind. Beispielsweise ist üblicherweise bereits  $m_2 \approx 2m_1$ . Damit lässt sich festhalten:

$$c(t) \approx c_1 \left( e^{-\frac{t}{\xi}} + e^{-\frac{L_1-t}{\xi}} \right) \tag{49}$$

Dieser Ausdruck kann verwendet werden, um  $\xi(\beta)$  in den untersuchten Ising-Systemen bei gegebener Temperatur  $\beta$  zu bestimmen.

<sup>11</sup>Hier tritt wieder das Wort „Zeit“ im Namen auf, obwohl kein Zusammenhang zu einem Zeitbegriff im hier untersuchten thermodynamischen System besteht. Der Name rührt daher, dass das Ising-Modell äquivalent zu bestimmten Raumzeit-Gittermodellen in der Quantenfeldtheorie ist. Die hier betrachtete Zeitscheibenkorrelationsfunktion ist ein Spezialfall zum Impuls null im dortigen Formalismus. Eine ausführliche theoretische Beschreibung findet sich z. B. in [22].

### 3.3. Cluster-Algorithmen

Gemäß [24] gab es zwei Hauptphasen in der Geschichte der Untersuchung des Ising-Modells. Die erste begann mit der Feststellung, dass die Ergebnisse von Monte-Carlo-Simulationen in der Nähe des kritischen Punktes eher unpräzise Ergebnisse lieferten. Es dauerte lange Zeit, bis die beiden Hauptgründe dafür verstanden waren. Diese Gründe sind die in Abschnitt 2.1.4 angerissenen Finite Size Effekte und das in Abschnitt 2.2.4 beschriebene Critical Slowing Down des Metropolis-Algorithmus und verwandter Ansätze<sup>12</sup> in Abhängigkeit von der Korrelationslänge, die sich in der Umgebung des pseudokritischen Punktes den Gitterabmessungen annähert.

Die zweite, wichtige Phase begann Ende der Achtziger Jahre, als man feststellte, dass Möglichkeiten existieren, das Critical Slowing Down zu umgehen. Namentlich sind Swendsen und Wang zu nennen, die realisierten, dass man effiziente Clusteralgorithmen, welche einige Zeit zuvor von Fortuin und Kasteleyn für Perkolationsmodelle entwickelt worden waren, auch auf eine Klasse von Modellen übertragen konnte, von denen das Ising-Modell ein Spezialfall ist [26]. Eine Optimierung dieses Algorithmus [27] wurde von Wolff entwickelt.

Diese Algorithmen umgehen geschickt das Problem, dass es schwierig ist, bei großer Korrelationslänge einen gemeinsam agierenden „Klumpen“ von Spins mit einem Algorithmus zu beeinflussen, der immer nur versucht einzelne Spins umzuklappen. Grund dafür ist, dass die Wahrscheinlichkeit für ein Zurückweisen des Umklappens sehr groß werden kann. Das ist der Grund, warum der Metropolis-Algorithmus aus Abschnitt 2.2.2 zum Erreichen des Ziels dieser Arbeit ungeeignet ist.

Die Lösung des Problems durch die Cluster-Algorithmen besteht darin, gezielt diese „Spinklumpen“ nachzubauen und auf einen Schlag umzuklappen. Dies verbessert das Konvergenzverhalten des Algorithmus enorm. Wie das genau funktioniert, soll nun gezeigt werden, allerdings basieren die Erklärungen nicht auf der Originalliteratur, sondern auf ausführlicheren Sekundärquellen mit praktischem Anwendungsbezug, nach denen sich auch die Implementierung richtet. Für den Swendsen-Wang-Algorithmus ist das [28], für den Wolff-Algorithmus [24].

#### Der Swendsen-Wang-Algorithmus

Um die Idee, Bereiche gemeinsam wechselwirkender Spins nachzubauen, zu verwirklichen, wird der Begriff der *Bindung* zwischen gleich ausgerichteten Spins eingeführt. Dabei handelt es sich keinesfalls um eine energetische Bindung, sondern lediglich eine Beschreibung dessen, wie die Cluster zu definieren sind.

Man iteriert nun über das gesamte Gitter, und weist jedem Nachbarn, dessen Spin die gleiche Richtung hat wie die aktuelle Gitterstelle, eine Bindung mit der später zu definierenden, temperaturabhängigen Wahrscheinlichkeit  $0 < p(\beta) < 1$  zu. Danach liegt zusätzlich zum Ausgangszustand  $Z$  ein Clusterzustand  $Z_C$  vor, in dem jede Gitterstelle Teil eines Clusters ist. Abbildung 6 verdeutlicht dies. Nun wird jedem Cluster ein zufälliger Spin zugewiesen oder es werden alle Spins in einem Cluster mit einer Wahrscheinlichkeit von 50% umgeklappt, was im Endresultat keinen Unterschied macht. Nach dem Entfernen der Bindungen ist man nun bei einem neuen Spinzustand angelangt, der sich stark vom ursprünglichen Zustand unterscheidet.

<sup>12</sup>Als ein Beispiel dafür wäre der Wärmebad-Algorithmus zu nennen. Lange Zeit galten alle Algorithmen mit lokalen Updates als anfällig für Critical Slowing Down, das Aufkommen von lokalen Wurm-Algorithmen in letzter Zeit widerlegt jedoch diese Annahme [25]. Letztere sind sogar noch etwas effizienter als die hier vorgestellten Algorithmen, basieren aber auf einem anderen Ansatz und werden in dieser Arbeit nicht weiter verfolgt.

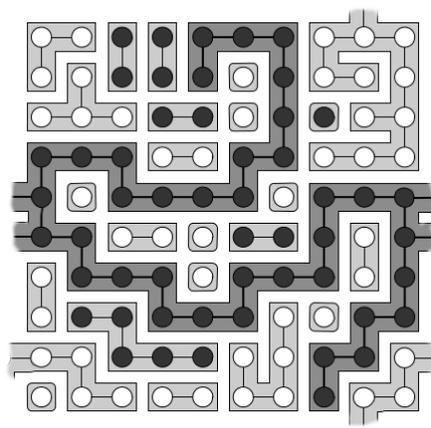


Abbildung 6: Beispielhafter Spinzustand  $Z$  und ein dazu möglicher Clusterzustand  $Z_C$  mit Bindungen zwischen bestimmten, gleich ausgerichteten Spins für ein quadratisches  $10 \times 10$ -Gitter in der Hochtemperaturphase etwas abseits des kritischen Punkts. Die Spins sind je nach Richtung mit schwarzen oder weißen Punkten codiert, zusammenhängende Cluster durch graue Polygone. Abgeleitet von einer Abbildung aus [28].

Man sieht leicht, dass der Algorithmus ergodisch ist, da eine von null verschiedene Wahrscheinlichkeit existiert, jedem Spin ein eigenes Cluster der Größe eins zuzuweisen und damit in jeden anderen Zustand zu gelangen. Die detaillierte Gleichgewichtsbedingung ist etwas schwieriger zu zeigen, der Beweis lässt sich aber analog zu dem beim Wolff-Algorithmus führen, weswegen darauf an dieser Stelle verzichtet wird. Gleiches gilt für die Wahl der temperaturabhängigen Wahrscheinlichkeit  $p(\beta) = 1 - e^{-2\beta}$ . Bevor diese beiden noch offenen Punkte im nächsten Abschnitt genauer behandelt werden, ist noch die Frage zu klären, wie man die Darstellung der Cluster algorithmisch am besten handhabt.

Eine verbreitete Möglichkeit, die Cluster zu identifizieren, ist der *Hoshen-Kopelman*-Algorithmus [29], ein Spezialfall aus der Klasse der Union-Find-Algorithmen (basierend auf den Operationen `find`, `union` und `make-set`), die allgemein dazu genutzt werden können, um Zusammenhangskomponenten in Graphen aufzufinden. Ziel ist es, jeder Gitterstelle ein *Label*, sprich eine Nummer zuzuweisen, wodurch die Clusterzugehörigkeit eindeutig festgelegt wird. Dazu ist es hilfreich, eine Hilfsstruktur zu definieren, die Mengen bezüglich Elementanzahl und Vereinigung von Mengen verwalten kann. Diese wird für jedes mögliche Label mit 0 initialisiert. Es sind genau so viele unterschiedliche Labels möglich, wie das Gitter Spins besitzt, da jedes Cluster schlimmstenfalls die Größe 1 haben kann.

Man beginne an einer beliebigen Gitterstelle und weise dieser ein noch nicht vergebenes Label  $l$  zu. Anschließend setze man den Eintrag in der Hilfsstruktur für das  $l$ -te Element auf 1 um zu signalisieren, dass sich genau ein Element mit dem Label  $l$  im Cluster befindet (`make-set`). Dann prüfe man alle Nachbarn, ob diese den gleichen Spin haben und ob  $p$  größer als eine Zufallszahl aus dem Intervall  $[0; 1)$  ist. Wenn dies für einen Nachbarn der Fall ist, bestimme man dessen Label über die Hilfsstruktur (`find`). Es gibt dann drei zu unterscheidende Fälle:

1. Der Nachbar hat noch kein Label. Dann weise man dem Nachbarn ebenfalls das

**Data:** Anfangszustand  $a_0$ , Bindungswahrscheinlichkeit  $p$   
**Result:** Markov-Kette  $a_1, \dots, a_n$

```

1 for  $i \leftarrow 1$  to  $n$  do
2    $\{s_1, \dots, s_V\} \leftarrow a_{i-1}$ ; // Die Spinkonfiguration aus dem Anfangszustand
3    $l \leftarrow (0, \dots, 0)$ ; // Die Labels zu jedem Index
4    $ll \leftarrow (0, \dots, 0)$ ; // Die Hilfsstruktur zu jedem Label
5    $L \leftarrow 0$ ; // Das höchste bisher vergebene Label
6   for  $x \leftarrow 1$  to  $V$  do
7     while  $ll_{l_x} < 0$  do
8        $l_x = -ll_{l_x}$ ; // find
9     end
10    if  $l_x = 0$  then
11       $L \leftarrow L + 1$ ; // make-set
12       $l_x \leftarrow L$ ;
13       $ll_{l_x} \leftarrow 1$ ;
14    end
15    for  $\forall y \in \{z | z \in \Lambda, \text{ sind\_Nachbar}(x, z), s_x = s_z\}$  do
16      if  $p > \text{Zufallszahl\_aus}([0;1])$  then
17        while  $ll_{l_y} < 0$  do
18           $l_y = -ll_{l_y}$ ; // find
19        end
20        if  $l_x = l_y$  then
21          /* Nichts zu tun */
22        end
23        else if  $l_y = 0$  then
24          /* Nachbar bekommt gleiches Label */
25           $l_y \leftarrow l_x$ ; // make-set + union
26           $ll_{l_x} \leftarrow ll_{l_x} + 1$ ;
27        end
28        else
29          /* Niedrigeres Label gewinnt */
30           $l_{<} \leftarrow \min l_x, l_y$ ; // union
31           $l_{>} \leftarrow \max l_x, l_y$ ;
32           $ll_{l_{<}} \leftarrow ll_{l_{<}} + ll_{l_{>}}$ ;
33           $ll_{l_{>}} \leftarrow -l_{<}$ ;
34           $l_x \leftarrow l_{<}$ ;
35        end
36      end
37    end
38  end
39   $a_i \leftarrow \{s_1, \dots, s_V\}$ ;
40 end

```

**Algorithmus 2:** Der Swendsen-Wang-Algorithmus für das Ising-Modell

- Label  $l$  zu und inkrementiere den  $l$ -ten Eintrag in der Hilfsstruktur (abgekürzte Kombination aus `make-set` und `union`).
2. Der Nachbar hat das gleiche Label. Dann ist nichts weiter zu tun.
  3. Der Nachbar hat bereits ein Label, aber ein anderes. Dann füge man beide Teilcluster zu einem zusammen (`union`.) Dazu wähle man als neues Label für beide das niedrigere Label  $l_<$  aus. Der Eintrag in der Hilfsstruktur für das größere Label  $l_>$ , der die Größe des Clusters mit dem bisherigen Label  $l_>$  angibt, wird zum Eintrag von  $l_<$  hinzuaddiert. Schließlich setze man den Eintrag von  $l_>$  auf  $-l_<$ .

Diese Vorgehensweise iteriere man über das gesamte Gitter.

Man sieht, dass die Hilfsstruktur zweierlei Funktion erfüllt. Zum einen findet man zu jedem Index das dazugehörige Label, indem man eventuellen negativen Einträgen solange folgt, bis man bei einem positiven landet. Zum anderen geben positive Einträge zum Index  $l$  die Größe des Clusters an. Der gesamte Algorithmus ist nochmal als Pseudocode in Algorithmus 2 zusammengefasst.

Der Swendsen-Wang-Algorithmus konstruiert also immer sämtliche Cluster und nur die Hälfte davon wird im Mittel umgeklappt. Außerdem tragen besonders kleine Cluster nicht viel zu den gewünschten großen Schritten im Zustandsraum bei. Diese beiden Mankos geht der Wolff-Algorithmus an.

### **Der Wolff-Algorithmus**

Anstatt alle Cluster zu konstruieren, beschränkt sich der Wolff-Algorithmus auf eines, das dann aber ohne Bedingung umgeklappt wird. Das macht den Wolff-Algorithmus bereits effizienter und obendrein ist dieses eine Cluster im Mittel sogar größer als das die mittlere Clustergröße beim Swendsen-Wang-Algorithmus. Außerdem vereinfacht sich die Clusterkonstruktion stark.

Man starte an einer zufälligen Stelle  $x$  und lege diese in einer Liste  $P$  ab.  $P$  enthält alle Gitterstellen, von denen aus das Cluster noch wachsen kann. In jeder Iteration des Konstruktionsschrittes wählt man ein beliebiges (beispielsweise das erste) Element  $k$  aus  $P$  aus und transferiert es in die Liste  $C$ , die am Ende alle Elemente des Clusters enthalten wird. Dann betrachtet man alle Nachbarn von  $k$  mit gleichem Spin und fügt diese mit der gleichen Wahrscheinlichkeit  $p(\beta)$  wie im Swendsen-Wang-Fall zu  $P$  hinzu, falls sie nicht sowieso schon in  $P$  oder  $C$  enthalten sind. Wenn  $P$  keine Elemente mehr enthält oder das komplette Gitter Teil des Clusters ist, ist die Clusterkonstruktion beendet und alle Spins im Cluster werden umgeklappt. Algorithmus 3 enthält nochmals den Pseudocode zu dieser Beschreibung, Abbildung 7 veranschaulicht den Prozess.

Ein Unterschied zum Swendsen-Wang-Algorithmus ist, dass pro Update nicht mehr ein ganzer Sweep durchgeführt wird, sondern nur der Bruchteil  $\frac{|C|}{V}$ . Dies ist jedoch in den meisten Fällen unproblematisch, insbesondere da man auch durch niedrigere Autokorrelationen und schnellere Konvergenz bei der Benutzung von Schätzern im Clusterbild (werden im nächsten Abschnitt behandelt) belohnt wird.

Auch beim Wolff-Algorithmus ist die Ergodizität wieder offensichtlich, denn jeder Spin kann mit Wahrscheinlichkeit  $\frac{1}{V}$  als erstes ausgewählt werden. Die Wahrscheinlichkeit, dass ein Cluster der Größe 1 erzeugt ist, ist nicht null und da dieses Cluster immer umgeklappt wird, kann man theoretisch in  $V$  Schritten von jedem Zustand zu einem beliebigen anderen Zustand kommen.

Die detaillierte Gleichgewichtsbedingung zu zeigen erfordert etwas mehr Anstrengung. Schließlich muss man die Wahrscheinlichkeit, dass der Konstruktionsprozess an einer

**Data:** Anfangszustand  $a_0$ , Bindungswahrscheinlichkeit  $p$

**Result:** Markov-Kette  $a_1, \dots, a_n$

```

1 for  $i \leftarrow 1$  to  $n$  do
2    $\{s_1, \dots, s_V\} \leftarrow a_{i-1}$ ;
3    $x \leftarrow$  Index zufälliger Gitterstelle  $\in 1, \dots, V$ ;
4    $C \leftarrow \{\}$ ; // Das eigentliche Cluster
5    $P \leftarrow \{x\}$ ; // Die Ränder des Clusters, können  $C$  noch wachsen lassen
6   while  $P \neq \{\}$  do
7      $k \leftarrow$  erstes Element von  $P$ ;
8      $P \leftarrow P \setminus \{k\}$ ;
9      $C \leftarrow C \cup \{k\}$ ;
10    for  $\forall l \in \{y | y \in \Lambda, y \notin C, y \notin P, \text{ sind\_Nachbar}(k, y), s_y = s_k\}$  do
11      if  $p > \text{Zufallszahl\_aus}([0;1])$  then
12         $P \leftarrow P \cup \{l\}$ ;
13      end
14    end
15  end
16  for  $\forall k \in C$  do
17     $s_k \leftarrow -s_k$ ;
18  end
19   $a_i \leftarrow \{s_1, \dots, s_V\}$ ;
20 end
    
```

**Algorithmus 3:** Der Wolff-Algorithmus für das Ising-Modell.

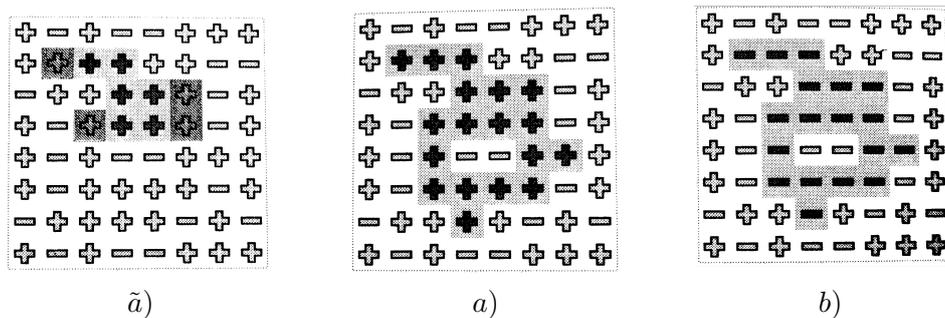


Abbildung 7: Ein Iterationsschritt des Wolff-Algorithmus für ein  $8 \times 8$ -Ising-Gitter. Die Werte  $\pm 1$ , die von den Spins angenommen werden können, sind mit  $+$  und  $-$  gekennzeichnet. In  $\tilde{a}$ ) ist das Cluster noch im Wachstum, hellgrau unterlegt sind die Elemente in  $C$ , dunkelgrau die Elemente in  $P$ . Teil  $a$ ) in der Mitte zeigt das fertige Cluster  $C$ , das zum Ausgangszustand  $a$  konstruiert wurde, und  $b$ ) auf der rechten Seite das Resultat nach dem Umklappen, also den Endzustand  $b$ . Die Grafiken sind aus [24] entnommen.

bestimmten Stelle endet, miteinbeziehen. Man betrachte dazu Abbildung 7. Jedes Element, das sich im Cluster  $C$  in  $7\tilde{a}$ ) (hellgrau schraffiert) befindet, muss zu irgendeinem Zeitpunkt Element von  $P$  (dunkelgrau schraffiert) gewesen sein. Jeder Nachbar mit gleichem Spin zu jedem Element in  $P$  wird mit Wahrscheinlichkeit  $p$  zu  $P$  hinzugefügt, bzw. mit Wahrscheinlichkeit  $(1 - p)$  nicht hinzugefügt.

Wenn das Cluster wie in  $7a)$  fertig konstruiert ist, bezeichne  $\partial C$  dessen Rand. Die Spins im Inneren von  $C$  haben beim Umklappen keinen Einfluss auf den Energieunterschied zwischen den Zuständen, denn das Ising-Modell wurde ja so konstruiert, dass in Abwesenheit äußerer Magnetfelder nur das Produkt  $s_x s_y$  der Spins benachbarter Gitterstellen  $x$  und  $y$  zur Hamiltonfunktion beiträgt, und  $(-1) \cdot (-1) = 1 \cdot 1$  ist offensichtlich. Es reicht also aus, nur die Spins auf dem Rand  $\partial C$  des Clusters und deren direkte Nachbarn außerhalb des Clusters zu betrachten. Bezeichne  $n_-$  die Anzahl der Nachbarschaftsbeziehungen zu negativen Spins außerhalb des Clusters und  $n_+$  entsprechend zu positiven Spins. Im Zustand  $a$  in  $7a)$  ist, wie man leicht nachzählen kann,  $n_- = 16$  und  $n_+ = 14$ . Die Wahrscheinlichkeit dafür, dass im Zustand  $a$  die Konstruktion des Clusters genau in diesem Clusterzustand abgebrochen wird, nachdem mit Wahrscheinlichkeit  $\frac{|C|}{V}$  ein passender Startpunkt ausgewählt wurde, ist  $(1 - p)^{n_+}$ , da die Spins im Cluster alle positives Vorzeichen haben.

Der mit dem Rand assoziierte Energiebeitrag ist  $H_{|\partial C} = n_- - n_+$  nach Gleichung (1). Dieser wechselt beim Übergang zu Zustand  $b$  in  $7b)$  das Vorzeichen, da alle Spins im Cluster das Vorzeichen wechseln, die Spins außerhalb jedoch nicht. Das bedeutet, dass sich die Wahrscheinlichkeiten  $\pi_a$  und  $\pi_b$  für das Antreffen der Zustände  $a$  und  $b$  nur im Vorzeichen des Boltzmannfaktors  $\exp(\mp \beta E_{|\partial C})$  unterscheiden und alle anderen Beiträge genau gleich bleiben.

Um vom Zustand  $b$  in einem Schritt in den Zustand  $a$  überzugehen, muss genau das von der Form her gleiche Cluster konstruiert werden, siehe  $7b)$ . Für die Wahrscheinlichkeit, dass der Konstruktionsprozess genau in diesem Clusterzustand abbricht ergibt sich nun  $(1 - p)^{n_-}$ , da die Spins im Cluster alle negative Vorzeichen haben. Die Werte für  $n_-$  und  $n_+$  unterscheiden sich natürlich nicht von denen in  $a$ .

Schließlich erhält man für die detaillierte Gleichgewichtsbedingung, vgl. Gleichung (24), den Ausdruck

$$e^{-\beta(n_- - n_+)}(1 - p)^{n_+} p(a \rightarrow b) = e^{-\beta(n_+ - n_-)}(1 - p)^{n_-} p(b \rightarrow a). \quad (50)$$

Analog zum Vorgehen beim Metropolis-Algorithmus in Abschnitt 2.2.2 macht es Sinn, die Übergangswahrscheinlichkeit als

$$p(a \rightarrow b) = \min \left( 1, \frac{e^{-\beta(n_+ - n_-)}(1 - p)^{n_-}}{e^{-\beta(n_- - n_+)}(1 - p)^{n_+}} \right) = \min \left( 1, \left( \frac{e^{-2\beta}}{1 - p} \right)^{n_+} \cdot \left( \frac{1 - p}{e^{-2\beta}} \right)^{n_-} \right) \quad (51)$$

zu definieren. Bei der Einführung des Wolff-Algorithmus weiter oben war festgelegt worden, dass jedes konstruierte Cluster unkonditional umgeklappt werde. Dies ist äquivalent zu einer Übergangswahrscheinlichkeit von  $p(a \rightarrow b) = 1$  für alle Zustände  $a$  und  $b$ . Aus Gleichung (51) ist sofort offensichtlich, dass diese Forderung zu

$$p = 1 - e^{-2\beta} \quad (52)$$

äquivalent ist. Eine Bindungswahrscheinlichkeit  $p$  zwischen gleich ausgerichteten Nachbarspins mit diesem Wert garantiert also einen maximal effizient arbeitenden Algorithmus, bei dem keine Zustandsänderungen zurückgewiesen werden.

Dieses häufige Zurückweisen von vorgeschlagenen Zustandsübergängen ist ja genau das große Problem beim Metropolis-Algorithmus, zusammen mit der Tatsache, dass sich die erzeugten neuen Zustände nur sehr wenig von den alten unterscheiden. Während Metropolis bildlich gesprochen nur mit Trippelschritten in zufällige Richtungen im Zustandsraum vorankommt und sehr selten zwischen den wahrscheinlichsten Zuständen wechseln kann, pflügen die Clusteralgorithmen geradezu mit Siebenmeilenstiefeln hin und her, wobei die Schrittweite genau dem von außen vorgegebenen Parameter Temperatur angepasst ist. Ein Critical Slowing Down ist fast gar nicht zu beobachten.

Zusammen mit den im nächsten Abschnitt behandelten verbesserten Schätzern sind diese Clusteralgorithmen ein Paradebeispiel dafür, wie speziell an die Problemstellung angepasste Algorithmen zwar dem Entwickler mehr Geistesleistung abverlangen, im Gegenzug aber durch um Größenordnungen verbesserte Effizienz Ressourcen sparen und die Geduld des Endanwenders schonen können.

### 3.4. Verbesserte Schätzer

Bei der bisherigen Betrachtung waren die durch Bindungen zwischen benachbarten Gitterstellen geformten Cluster nur Mittel zum Zweck, um effizient von einem Zustand zum nächsten zu kommen. Eine genauere Untersuchung zeigt jedoch, dass das Clusterbild  $\mathcal{C}$  für die hier interessierende Observable, nämlich die Zeitscheibenkorrelationsfunktion  $c(t)$ , die gleiche Information enthält wie das Spinbild  $\mathcal{S}$ . Es gilt also

$$\langle c(t) \rangle_{\mathcal{S}} = \langle c(t) \rangle_{\mathcal{C}}, \quad (53)$$

wobei der Erwartungswert im jeweiligen Bild zu bilden ist. Mehr noch, der Schätzer für den Erwartungswert im Clusterbild erweist sich als vorteilhafter als der klassische Schätzer im Spinbild, da die damit assoziierten statistischen Fehler bei der Verwendung von Ersterem geringer sind, während der Erwartungswert unverändert bleibt.

Um die Informationen im Clusterbild zu verarbeiten, ist zunächst die Definition der *charakteristischen Funktion*  $\Theta_C(x)$  zum Cluster  $C \in \mathcal{C}$  notwendig. Diese ist gegeben durch

$$\Theta_C(x) = \begin{cases} 1 & \text{für } x \in C \\ 0 & \text{sonst} \end{cases}. \quad (54)$$

Eine ausführliche Rechnung in [30] zeigt, dass sich der Erwartungswert der Spin-Spin-Korrelation  $\langle s_x s_y \rangle$  im Clusterbild des Wolffschen Einzelcluster-Algorithmus durch

$$\langle s_x s_y \rangle_{\mathcal{C}} = \left\langle \frac{V}{|C|} s_x s_y \Theta_C(x) \Theta_C(y) \right\rangle_{\mathcal{C}} \quad (55)$$

ausdrücken lässt. Wenn  $x$  und  $y$  sich in einem Markov-Kettenglied im gleichen Cluster befinden, erhält man für das Sample also einen mit der reziproken Clustergröße gewichteten, immer positiven Wert, da Spins im gleichen Cluster immer das gleiche Vorzeichen haben. Ansonsten erhält man für das Sample den Beitrag null.

Man kann sich leicht überlegen, warum dies vorteilhaft ist. Schließlich werden die Cluster ja genau so konstruiert, dass sie gemeinsam wechselwirkende Spins abbilden. Die Varianz bei der Verwendung dieser Methode ist deswegen geringer, weil im klassischen Spinbild unkorrelierte Spins zufällige Beiträge von entweder  $+1$  oder  $-1$  beitragen, was sich für eine große Zahl an Samples schließlich weg mittelt, während die Beiträge unkorrelierter Spins im Clusterbild direkt  $0$  sind.

Die Zeitscheibenkorrelationsfunktion aus Gleichung (48) lässt sich somit vereinfacht durch

$$c(t) = \left\langle \frac{1}{|C|L_1} \sum_{i=1}^{L_1} n_i n_{i+t} \right\rangle_c \quad (56)$$

ausdrücken. Dabei bezeichnet  $n_i$  die Anzahl der Elemente  $x$  im Cluster, deren erste Komponente  $x_1 = i$  ist, sprich die Menge der Clusterelemente in der  $i$ -ten Zeitscheibe. Wenn man schon bei der Clusterkonstruktion die  $n_i$  direkt mitbestimmt, lässt sich die Zeitscheibenkorrelationsfunktion nahezu ohne zusätzlichen Rechenaufwand bestimmen.

Für den Swendsen-Wang-Algorithmus lässt sich die Zeitscheibenkorrelationsfunktion auf ähnlichem Wege ermitteln, wenn man zusätzlich noch bei jedem Sweep über sämtliche Cluster mittelt. Für Details siehe [16].

Es gibt auch für andere Größen noch weitere verbesserte Schätzer im Clusterbild, zum Beispiel für die magnetische Suszeptibilität  $\chi$ . Diese sind aber nicht nötig, um die Korrelationslänge zu bestimmen, was der Fragestellung entspricht, auf die sich die vorliegende Arbeit konzentriert.

### 3.5. Behandlung der Autokorrelation

Im letzten Abschnitt dieses Kapitels soll noch darauf eingegangen werden, wie bei der Bestimmung der Autokorrelation zu verfahren ist. In Abschnitt 2.2.3 wurde bereits aufgezeigt, wie die integrierte Autokorrelationszeit  $\tau_{\text{int}}$  durch Summation über die normalisierte Autokorrelationsfunktion  $\Gamma(t)$  bestimmt werden kann. Aber es wurde nichts darüber ausgesagt, über welchen Bereich sich die Summe erstrecken soll.

Glücklicherweise muss diese schwierige Frage hier nicht beantwortet werden, denn es liegt in dieser Arbeit der Fall einer Kombination aus geringer Autokorrelation und einer mit Bezug auf die Rechenzeit „billigen“ Messmethode vor. Dafür gibt es eine praktikable Möglichkeit, mit der sich die Korrelationszeit auf einfacherem Wege abschätzen lässt.

Der Argumentation in [24] folgend, fasse man die aus den einzelnen Markov-Kettengliedern  $X_i$  gewonnenen Samples  $o_i$  zu einer Observablen  $O$  zum einem neuen Sample  $o'_i$  durch Mittelwertbildung mit dem direkten Nachbarn gemäß

$$o'_i = \frac{o_{2i} + o_{2i+1}}{2} \quad (57)$$

zusammen. Da sich dabei die Anzahl an verbleibenden Samples jeweils halbiert, macht es Sinn, zu Beginn eine Markov-Kette der Länge  $2^N$  zu erzeugen, damit man keine übriggebliebenen Samples gesondert behandeln muss.

Was hier passiert, ist, dass in der  $j$ -ten Iteration dieser Vorgehensweise die ursprünglich  $2^N$  Samples zu  $2^{N-j}$  Blöcken der Länge  $2^j$  zusammengefasst und gemittelt werden. Der Mittelwert über alle so gebildeten  $o'_i$  bleibt in jeder Iteration der gleiche, weil es sich beim Mittelwert um eine lineare Größe handelt. Allerdings gilt dies nicht unbedingt für den Standardfehler dieses Datensatzes.

Wenn aufeinanderfolgende Glieder der Markov-Kette miteinander korreliert sind, so ist zu erwarten, dass der Abstand zwischen aufeinanderfolgenden Observablen  $o_i$  geringer ist, als man das für komplett zufällig verteilte Werte im Mittel erwarten würde. Die Methode fasst nun stets zwei korrelierte Samples  $o_{2i}$  und  $o_{2i+1}$  zu einem  $o'_i$  zusammen, was die Gesamtkorrelation verringert. Somit wächst der Standardfehler in jeder Iteration  $j$  je nach Stärke der verbleibenden Korrelationen an, während die Korrelationen zwischen den Blöcken mit zunehmender Blocklänge selbst zurückgehen.

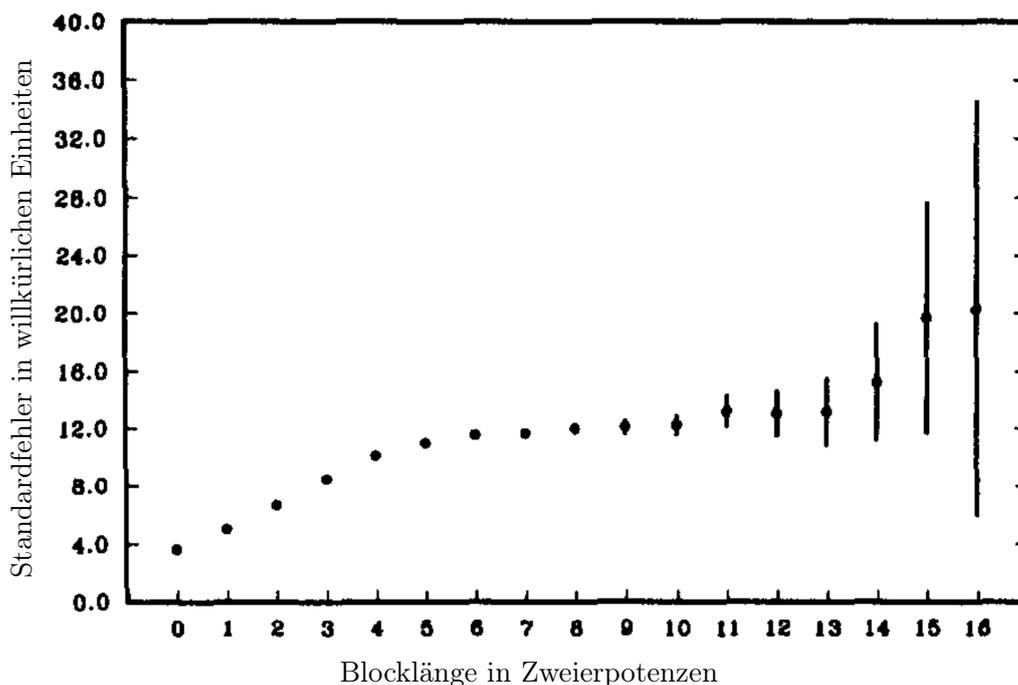


Abbildung 8: Der numerische Fehler inklusive dem Fehler des Fehlers als Funktion der Blocklänge. Für kleine Blocklängen wächst der Fehler zunächst an, um dann ab einer gewissen Blocklänge seinen tatsächlichen, konstanten Wert anzunehmen. Bei sehr großen Blocklängen sinkt die Aussagekraft aufgrund der dünner werdenden Datenlage. Die Grafik ist aus [31] entnommen.

Für unkorrelierte Samples hingegen ist zu erwarten, dass der Standardfehler konstant bleibt, weil die natürliche Streuung der unkorrelierten Messwerte auf den Standardfehler keinen Einfluss hat.

Der Standardfehler als Funktion der Blocklänge sollte also für korrelierte Daten von Anfang an anwachsen und schließlich in ein Plateau übergehen, vorausgesetzt zu jeder Blocklänge gibt es noch genügend Blöcke, sodass der Fehler des Standardfehlers sich noch in Grenzen hält. Dies ist beispielhaft in Abbildung 8 dargestellt. Aus der Blocklänge, bei der der Übergang zum Plateau auftritt, lässt sich die Autokorrelationszeit abschätzen. Im abgebildeten Beispiel ist dies bei  $\approx 2^6$  der Fall. Diese Technik wird oft *Data Binning*, *Data Blocking* oder *Data Bunching* genannt.

## 4. Implementierung

Nun da bekannt ist, auf welchen theoretischen Grundlagen die Untersuchung des kritischen Verhalten fußt und mit welcher Methode die Problemstellung anzugehen ist, ist es an der Zeit, der Frage nachzugehen, wie die zugehörige praktische Umsetzung aussieht. Bei dieser handelt es sich um die Implementierung der im letzte Kapitel behandelten Algorithmen. Die Diskussion der Implementierung ist Gegenstand dieses Kapitels. Dazu wird in jedem Abschnitt zuerst ein grober Überblick über die jeweiligen Bausteine gegeben und dann auf teils recht spezielle, aber durchaus interessante Details eingegangen.

### 4.1. Programmwurf

Beim Endergebnis des Programmiereteiles dieser Arbeit handelt es sich um gute 2000 Zeilen Quellcode, die Monte-Carlo-Simulationen mittels Clusteralgorithmus zur Bestimmung der Zeitscheibenkorrelationsfunktion für ein beliebigdimensionales Ising-Modell ausführen können. Die Quellen sind unter LGPLv2-Lizenz auf <https://github.com/LeopoldKellers/Bachelorarbeit> verfügbar.

Programmiersprache der Wahl ist C in der Variante C99, mit der der Autor schon jahrelange Erfahrung hat. Weitere Gründe für diese Wahl sind schlicht, dass die C++-Kenntnisse des Autors als für ein solches Projekt ungenügend eingeschätzt wurden, Fortran nach wie vor als unästhetisch empfunden wird und die Erfahrungen des Autors mit Java bei leistungsintensiven Anwendungen bisher eher durchwachsen ausfielen. Zum letzten Punkt kommt zusätzlich hinzu, dass auf der Zielplattform auch mangels Laufzeitumgebungen gar kein Bytecode ausgeführt werden kann.

Mit dem Aufkommen von Mehrkernprozessoren und der Möglichkeit, Gleitkomma-rechnungen auf Grafikkarten auszulagern, scheint die Begeisterung für die parallele Abarbeitung rechenintensiver Probleme, die im wissenschaftlichen Bereich schon immer vorhanden war, seit es Parallelrechner gibt, nochmals deutlich zugenommen zu haben. Jedoch eignen sich manche Probleme besser zur Parallelisierung als andere, und durch die globale Natur der Clusterupdates<sup>13</sup> ist es schwierig, Nebenläufigkeit bei der Programmausführung effizient zu nutzen. Vor diesem Hintergrund und der Tatsache, dass in heutigen Hochleistungsrechnern anstatt Vektorprozessoren einfach eine Vielzahl an Standardprozessoren verbaut wird, wurde auf eine parallele Implementierung komplett verzichtet. Nichtsdestotrotz lässt sich auf triviale Weise ein hoher Grad an nahezu optimaler Parallelisierung erreichen, wenn das gleiche Programm mit unterschiedlichen Parametern auf vielen Prozessorkernen gleichzeitig läuft, was dank heute in großen Mengen preiswert verfügbarem Arbeitsspeicher kein Problem darstellt.

Das Programm an sich ist in mehrere Module, die jeweils eine spezielle Funktionalität kapseln, unterteilt. Diese Module sind im Einzelnen:

**MAIN:** Enthält das Hauptprogramm. Dieses nimmt die Parameter der Simulation wie die Gittergröße, Temperatur etc. als Kommandozeilenargumente entgegen, und führt die Simulation mit anschließender Auswertung der Daten unter Benutzung der Module `MONTE_CARLO` und `DATA_ANALYSIS` aus. Dieses Modul wird in Abschnitt 4.5 genauer beschrieben.

<sup>13</sup>Im Gegensatz dazu ermöglichen lokale Updates wie z.B. im Metropolis-Algorithmus viel bessere Möglichkeiten der Parallelisierung. Die angesprochenen systematischen Probleme dieses Algorithmus, insbesondere das Critical Slowing Down können den so erzielten Leistungsgewinn dennoch nicht wettmachen. Ein interessanter Lösungsansatz ist der in [32] besprochene Hybrid-Algorithmus.

MONTE\_CARLO: Enthält alles, was nötig ist, um das Ising-Modell mit Wolff- oder Swendsen-Wang-Algorithmus zu simulieren und die Observablen Energie pro Spin sowie Zeitscheibenkorrelationsfunktion zu messen. Eine ausführliche Beschreibung folgt in Abschnitt 4.2.

DATA\_ANALYSIS: Enthält die Funktionalität zur statistischen Auswertung der einzelnen Samples. Abschnitt 4.3 behandelt spezielle Details der Implementierung.

TEST: Enthält Testfälle, um konkrete Untereinheiten des Programms auf Korrektheit zu prüfen. Die Schwierigkeiten beim Testen, die mit stochastischen Algorithmen einhergehen und Ansätze zu deren Lösung sind Gegenstand von Abschnitt 4.4

## 4.2. Monte-Carlo

Das Modul MONTE\_CARLO bildet das Herzstück der Software. In diesem sind sowohl der Wolff- als auch der Swendsen-Wang-Algorithmus implementiert<sup>14</sup>.

Weil die Qualität der mit Monte-Carlo-Methoden erhaltenen Ergebnisse mit der Qualität der verwendeten Zufallszahlen stehen und fallen kann, ist eine hohe Qualität des verwendeten Generators essentiell für gute Ergebnisse. Daher enthält das Modul eine Wrapperfunktion, über die verschiedene Bibliotheken zur Erzeugung von Zufallszahlen eingebunden werden können. Momentan sind dies der *Double precision SIMD-oriented Fast Mersenne Twister* (dSFMT) [33], eine schnelle Variante des Mersenne-Twister-Algorithmus und der *Well Equidistributed Long-period Linear* (WELL) Generator. Der Mersenne-Twister wurde für Monte-Carlo-Simulationen als Reaktion auf Schwächen in konventionellen Generatoren entwickelt [34] und scheint sich bis heute zum de-facto-Standard für sehr viele Bibliotheken entwickelt zu haben. Der WELL-Generator merzt die bekannte Schwäche des Mersenne-Twisters aus, bei schlechter Initialisierung anfangs schlechte Zufallsverteilungen zu erzeugen. Er erzeugt etwas bessere Zufallszahlen, allerdings bei etwas langsamerer Geschwindigkeit [35].

Beide sind 623-dimensionale Pseudo-Zufallszahlen-Generatoren mit einer Periodendauer von  $2^{19937}$ , was für die hier beabsichtigten Zwecke allemal ausreichen sollte. Die beiden Generatoren lassen sich mit einem vorgegebenen Anfangszustand (Seed) aus einer Datei initialisieren. Das ermöglicht es zum einen, wegen der deterministischen Erzeugung der Pseudozufallszahlen reproduzierbare Markov-Ketten und damit Simulationsergebnisse zu erhalten, eine Eigenschaft, die bei der Fehlersuche essentiell ist. Zum anderen kann man zur Initialisierung von Produktivläufen unter Linux einfach die Datei `/dev/random` übergeben, die wesentlich mehr Entropie bereitstellt als gängige Alternativen wie die Uhrzeit beim Programmstart.

Wie bereits erwähnt, besitzt der Mersenne-Twister eine Achillesferse, was schlechte Initialisierung betrifft. Deswegen werden die Generatoren gemäß den Empfehlungen in [36] durch ein Wegwerfen der ersten Million generierter Zahlen aufgewärmt, was mit einer Zehntelsekunde im Vergleich zur gesamten Simulationsdauer nicht groß ins Gewicht fällt.

Wie schon bei der Beschreibung der Algorithmen in Abschnitt 3.3 ist die Implementierung des Swendsen-Wang-Algorithmus an den Beschreibungen in [28], die des Wolff-Algorithmus an [24] orientiert. Für beide Algorithmen stehen jeweils Funktionen zur Verfügung, um eine Instanz des Ising-Modells der gewünschten Größe zu erzeugen

<sup>14</sup>Grund dafür ist ein Missverständnis des Autors bezüglich der Existenz von verbesserten Schätzern beim Wolff-Algorithmus zu Beginn der Implementierung.

bzw. den Speicher wieder frei zu geben, nachdem die Simulation abgeschlossen ist. Damit lässt sich bei entsprechender Disziplin des Programmierers eine objektorientierte Benutzung des Moduls nachahmen, auch wenn die Programmiersprache C dafür keine syntaktischen Hilfsmittel anbietet. Genauso bieten beide Algorithmen je eine Funktion für eine Iteration des Algorithmus, also zur Erzeugung eines neuen Markov-Kettenglieds. Darüber hinaus gibt es Funktionen zur anschließenden Messung der Energie pro Spin und der Zeitscheibenkorrelationsfunktion. Die Messung der Energie erfolgt bei beiden Algorithmen mit der naiven Variante, dass einmal über das gesamte Gitter iteriert und dabei das Produkt des Spins mit allen noch nicht betrachteten Nachbarspins aufsummiert und mit dem Gittervolumen normiert wird. Da die Energie pro Spin nur zum Finden der nötigen Equilibrationszeit relevant ist, ist eine effizientere Implementierung auch nicht notwendig.

Ein wesentlicher Unterschied zwischen den Algorithmen ist, dass ein Update bei Swendsen-Wang immer jeden einzelnen Spin betrifft, also ein kompletter Sweep über das Gitter gemacht wird. Im Gegensatz dazu wird bei Wolff nur ein Cluster, dessen mittlere Größe von der Temperatur abhängt, behandelt, aber kein kompletter Sweep. Weiterhin ist die interne Repräsentation beider Algorithmen grundsätzlich verschieden.

Die Implementierung des Swendsen-Wang-Algorithmus mit dem Hoshen-Kopelman Cluster-Labeling ist vergleichbar mit dem Pseudocode in Algorithmus 2 auf Seite 20. Weil bei jedem Update das komplette Gitter durchlaufen werden muss, wird die gesamte Information in linearen, also eindimensionalen Arrays mit je genau der Anzahl an Elementen, wie Gitterstellen vorhanden sind, gehalten. Man kann sich das Gitter der Reihe nach durchnummeriert vorstellen und zum gleichen Index findet man alle Informationen wie Label, Spin etc. in den jeweiligen einzelnen Arrays. Dadurch, dass beim linearen Durchlaufen des Gitters hauptsächlich auf aufeinanderfolgende Elemente zugegriffen wird, kann man mit dieser Implementierung positive Cache-Effekte erwarten.

Für ein komplettes Update muss zweimal über das Gitter iteriert werden. Beim ersten Durchgang wird allen Gitterstellen ein Label zugewiesen, wodurch die Cluster erzeugt werden. Falls zwei Teilcluster vereinigt werden müssen, enthält ein Hilfsarray zu jedem Label einen Verweis auf das zugehörige eigentliche Label (signalisiert durch einen negativen Wert) oder die Größe des Clusters (signalisiert durch einen positiven Wert) mit diesem Label. Im zweiten Durchgang wird jeder Gitterstelle das eigentliche Label durch rekursives Auflösen der Referenzen im Hilfsarray, bis man bei einem nichtnegativen Wert anlangt, zugewiesen. Gleichzeitig wird dem Spin der neue Wert, der für das Cluster mit dem entsprechenden Label vorgesehen ist, zugewiesen.

Bei einem anschließenden Aufruf der Funktion zur Messung der Zeitscheibenkorrelationsfunktion für einen vorgegebenen Abstand  $t$  zwischen den Zeitscheiben wird für je zwei dieser Zeitscheiben ermittelt, wie viele Spins in der Zeitscheibe ein gemeinsames Label besitzen. Damit beim anschließenden Multiplizieren und Aufsummieren über die verschiedenen Label nicht allzu viele Nullen unnötigerweise aufaddiert werden<sup>15</sup>, wird das Hilfsarray diesmal benutzt um Buch zu führen, welche Labels überhaupt vertreten sind.

Diese Methode ist noch nicht optimal performant. Aber in Anbetracht der Tatsache, dass die Implementierung des Swendsen-Wang-Algorithmus sowieso nicht im Produktiveinsatz zur Anwendung kommt, sondern nur der Verifizierung der korrekten Implementierung des Wolff-Algorithmus dient, lohnt es sich kaum, an dieser Stelle mehr Aufwand zu betreiben.

<sup>15</sup>Schließlich ist die Anzahl der tatsächlich in einer Zeitscheibe vorkommenden, verschiedenen Labels wesentlich kleiner als die maximal mögliche Anzahl verschiedener Labels.

Aus dem Pseudocode in Algorithmus 3 auf Seite 22 lässt sich schon erkennen, dass eine Implementierung mit großen Arrays im Fall des Wolff-Algorithmus nicht das Mittel der Wahl ist. Stattdessen wird ein Ansatz in eher objektorientierter Manier verfolgt, bei dem jede Gitterstelle mit einer eigenen Datenstruktur, die alle relevanten Informationen enthält, repräsentiert wird. Das Cluster und die Randelemente, die das Cluster noch wachsen lassen können, werden als Listen repräsentiert, welche Verweise auf die eben erwähnten Strukturen enthalten. Aus Geschwindigkeitsgründen, und weil mit der Gittergröße eine maximale Elementzahl festliegt, bietet sich eine Implementierung als Arrayliste an.

Um beim Hinzufügen ein Durchsuchen der Liste, ob das neue Element nicht schon Teil des Clusters ist, zu ersparen, dient ein Flag, dass für jede Gitterstelle bei der erstmaligen Aufnahme in das Cluster gesetzt wird. Da insgesamt nur ein Cluster konstruiert wird, bietet es sich an, schon direkt beim Konstruktionsprozess die Anzahl der Spins in jeder Zeitscheibe direkt mit abzuspeichern<sup>16</sup>. Das ermöglicht eine sehr effiziente Auswertung der Zeitscheibenkorrelationsfunktion im Anschluss an das Umklappen der Spins im Cluster. Es muss dann nur noch das Produkt der Anzahl der Elemente in den jeweiligen Zeitscheiben mit vorgegebenem Abstand zueinander über die gesamte Gitterlänge gemittelt werden.

Die jeweilige Implementierung der beiden Algorithmen ist also grundsätzlich sehr unterschiedlich. Somit ist es relativ unwahrscheinlich, den gleichen Fehler bei der Implementierung zweimal zu machen. Dementsprechend können die Ergebnisse beider Algorithmen gegeneinander getestet werden. Dies wird jedoch nicht hier, sondern in Abschnitt 4.4 genauer erläutert.

Die Unabhängigkeit von der Dimensionalität des Systems wird in beiden Implementierungen der verschiedenen Algorithmen durch Nachbarschaftstabellen erreicht. Diese werden einmalig bei der Initialisierung des Gitters zu gegebenen Ausmaßen erzeugt. Ein weiterer Vorteil ist zunächst, dass man damit auch von der Repräsentation des Gitters, ob eindimensionale oder mehrdimensionale Array- oder Zeigerstruktur, unabhängig wird. Obendrein ist ein Nachschlagen in der Tabelle performanter als wiederholte Modulo-Operationen zur Behandlung der Gittergrenzen.

Nach dieser relativ erschöpfenden Behandlung des Moduls MONTE\_CARLO handelt der folgende Abschnitt davon, wie es mit der statistischen Auswertung der so gewonnenen Daten aussieht.

### 4.3. Statistik

Zu den mit Hilfe des Moduls MONTE\_CARLO gewonnenen Samples sind die statistischen Größen arithmetisches Mittel, Varianz und die aus letzterer abgeleiteten Größen Standardabweichung bzw. Standardfehler von Interesse. Das arithmetische Mittel, im Folgenden kurz Mittelwert genannt, ist der Erwartungswert der untersuchten Größe, wohingegen der Standardfehler ein Maß für die Qualität ist, mit der der Erwartungswert vorliegt, vergleiche dazu auch die Gleichungen (28) bis (30).

Die Lehrbuchformeln für den Mittelwert  $M$  und die Summe der quadratischen Abweichungen, kurz Quadratsumme  $S$ , eines Datensatzes  $x_1, \dots, x_N$  sind

$$M = \frac{1}{N} \sum_{i=1}^N x_i \text{ und } S = \sum_{i=1}^N (x_i - M)^2. \quad (58)$$

<sup>16</sup>Prinzipiell wäre das auch bei Swendsen-Wang möglich, der dort nötige Speicherbedarf im schlimmsten Fall von lauter Einzelclustern wäre aber mit statischer Allokierung (im Sinne von einmalig bei der Initialisierung der Simulation) nicht vertretbar.

Daraus ergeben sich Varianz  $\sigma^2$ , Standardabweichung  $\sigma$  und Standardfehler  $\sigma_N$  aus

$$\sigma^2 = \frac{S}{N-1} \text{ und } \sigma = \sqrt{\frac{S}{N-1}} \text{ sowie } \sigma_N = \sqrt{\frac{S}{N(N-1)}}. \quad (59)$$

Die Berechnung der Größe  $S$  nach Gleichung (58) erfordert ein zweimaliges Durchlaufen des gesamten Datensatzes, was besonders bei großen Datensätzen eine unnötige Verschwendung von Rechenzeit darstellt. Eine mathematisch äquivalente Methode zur Berechnung der Quadratsumme ist

$$S = \sum_{i=1}^N x_i^2 - \frac{1}{N} \left( \sum_{i=1}^N x_i \right)^2. \quad (60)$$

Diese hat es jedoch bei numerischer Auswertung in sich [37]. Das konkrete Problem ist, dass Subtrahend und Minuend im obigen Ausdruck beide extrem groß werden können. Wegen der endlichen Darstellung von Gleitkommazahlen im Rechner mit einer Genauigkeit relativ zur Größenordnung der Zahl kann die letztlich erhaltene Varianz aufgrund der Subtraktion einen inakzeptabel großen Fehler aufweisen<sup>17</sup>. Wenn die Streuung der Samples gering ist, kann es sogar soweit kommen, dass das gerundete Zwischenergebnis für den Subtrahend größer ist als das für den Minuend. Dann erhält man einen mathematisch natürlich unsinnigen, negativen Wert für die Varianz.

Knuth empfiehlt daher in seinem Standardwerk „The Art of Computer Programming“ eine andere Methode, bei der der Mittelwert und die Quadratsumme aus der Kombination von zwei rekursiven Formeln bestimmt werden [38]. Mit den Anfangsbedingungen  $M_1 = x_1$  und  $S_1 = 0$  lauten diese

$$M_i = M_{i-1} + \frac{x_i - M_{i-1}}{i}, \quad (61)$$

$$S_i = S_{i-1} + (x_i - M_{i-1})(x_i - M_i). \quad (62)$$

Diese Methode ist zwar wegen der Division bei jedem Update etwas ineffizienter<sup>18</sup>, hat aber den Vorteil, dass die Quadratsumme nie negativ werden kann, ohne dass ein zweifaches Iterieren über den Datensatz nötig ist. Ein weiterer Seiteneffekt ist, dass alle Werte quasi in Echtzeit zur Verfügung stehen. Und drittens ermöglicht es dieser Ansatz, weitaus mehr Samples zu nehmen, als man im Hauptspeicher halten kann<sup>19</sup>, ohne durch Festplattenzugriffe wertvolle Rechenzeit einzubüßen.

Die Methode stammt ursprünglich von Welford und die Formel für die Quadratsumme lautet in der Originalversion [39]

$$S_i = S_{i-1} + \frac{i-1}{i} (x_i - M_{i-1})^2. \quad (63)$$

<sup>17</sup>Dieses Phänomen einer massiven Zunahme des relativen Fehlers im Vergleich zum absoluten Fehler tritt bei numerischen Rechnungen mit endlicher Genauigkeit häufiger auf und wird als katastrophale Auslöschung bezeichnet.

<sup>18</sup>Im Allgemeinen ist eine Division die mit Abstand teuerste Grundrechenoperation in Bezug auf die Ausführungszeit, sodass normalerweise versucht wird, deren Anzahl möglichst gering zu halten.

<sup>19</sup>Auf der Zielplattform stehen pro Rechenknoten 24 GB Hauptspeicher für 12 Prozessorkerne zur Verfügung. Dies würde nach Abzug des Speicherbedarfs für das Betriebssystem und die Repräsentation der 12 Ising-Modelle bedeuten, dass man nur weniger als  $10^6$  Samples mit doppelter Gleitkommagenauigkeit pro Wert für die Zeitscheibenkorrelationsfunktion zu jedem Abstand der Zeitscheiben  $t = 0, 1, \dots, 63$  im untersuchten Fall des  $64 \times 64 \times T$ -Gitters im Hauptspeicher halten könnte. Dies wäre aber nur ein Bruchteil der für die gewünschte Präzision notwendigen Anzahl Samples, die in der Größenordnung  $10^8$  liegt.

Die von Knuth zur Ableitung seiner Formel benutzte Äquivalenz

$$x_i - M_i = \frac{i-1}{i}(x_i - M_{i-1}) \quad (64)$$

ist durch Einsetzen von Gleichung (61) auf der linken Seite leicht zu zeigen und kann noch ein weiteres Mal angewendet werden um den in der Implementierung benutzten Ausdruck

$$S_1 = 0, \quad S_i = S_{i-1} + \frac{i}{i-1}(x_i - M_i)^2 \quad (65)$$

zu erhalten. Der Vorteil dieser letzten Rekursionsgleichung wird klar, wenn man sich die in Abschnitt 3.5 behandelte Methode zur Abschätzung der Autokorrelation via Data Blocking zurück in Erinnerung ruft. Diese ist nämlich in die Implementierung des Moduls `DATA_ANALYSIS` mit eingearbeitet.

Der Konstruktor einer Statistik-Instanz verlangt eine maximale Blocklänge, an die die zurückgegebene Datenstruktur angepasst wird. Diese Struktur speichert den aktuellen Mittelwert  $M_i$  und die Quadratsummen  $S_i^j$  zu allen Blockgrößen  $j$  in Schritten von Zweierpotenzen, also  $j = 1, 2, 4, 8, \dots$ . Diese Datenstruktur wird der Funktion zum Update der bisherigen Statistik zusammen mit einem Satz von neu zur Statistik hinzuzufügenden Werten, deren Anzahl der maximalen Blockgröße  $j_{\max}$  entspricht, übergeben.

Beim Update der Statistik mit einem ganzen Block von neuen Daten liegt der Vorteil von Gleichung (65) gegenüber Gleichung (62) oder (63) auf der Hand: da immer nur der aktuelle Mittelwert  $M_i$  benötigt wird und dieser im Hinblick auf das Data Blocking invariant ist, erspart man sich das Hantieren mit zu jeder Blockgröße gehörigen alten Mittelwerten<sup>20</sup> und das neu zur Statistik hinzuzufügende Array von  $j_{\max}$  Zahlen muss nur einmal durchlaufen werden.

Nachdem schließlich sämtliche Daten nach Beendigung der Simulation verarbeitet wurden, kann man über durch das Modul bereitgestellte Funktionen die statistischen Größen in Gleichung (59) unter Angabe der gewünschten Blockgröße abfragen und danach den belegten Speicher wieder freigeben.

#### 4.4. Test

Das Testen von Software ist grundsätzlich bereits schon eine anspruchsvolle Aufgabe, für stochastische Algorithmen, wie sie bei Monte-Carlo-Methoden zur Anwendung kommen, handelt es sich jedoch um eine besondere Herausforderung. Der Grund ist, dass sich deterministische Komponententests nur auf wenige Teile des Quelltexts anwenden lassen. Die statistische Auswertung der Daten bezüglich Mittelwert und Varianz oder die korrekte Initialisierung der von den Clusteralgorithmen verwendeten Nachbarschaftstabellen sind Beispiele, die sich gut durch Unittests behandeln lassen.

Bei den Clusteralgorithmen selbst tritt das Problem auf, dass die zurückgegebenen Ergebnisse statistischen Schwankungen in einem gewissen Fehlerbereich unterliegen. Dadurch schlagen entweder Unittests quasi grundlos von Zeit zu Zeit fehl. Oder aber die Toleranz für die Akzeptanz des Tests muss so groß gewählt werden, dass subtile Abweichungen nicht erkannt werden können und dem Tester letztlich ein falsches Gefühl von Sicherheit vorgegaukelt wird. Auch das Initialisieren des Zufallszahlengenerators mit einem festen, vorgegebenen Wert, sodass immer die gleiche Sequenz an Zufallszahlen generiert wird,

<sup>20</sup>Schließlich ist das  $i$  in  $S_i^j$  abhängig von der Blockgröße  $j$ . Damit sind die  $M_{i-1}$  für verschiedene  $j$  unterschiedlich, das aktuelle  $M_i$  jedoch universell.

kann nur bedingt Abhilfe schaffen<sup>21</sup>. Das Problem bei dieser Vorgehensweise ist nämlich, dass selbst geringfügige Änderungen in der Implementierung die Konstruktion völlig unterschiedlicher, aber möglicherweise durchaus immer noch korrekter Cluster veranlassen können.

Aufgrund dieser Tatsachen ist ein defensiver Programmierstil, der die Wahrscheinlichkeit stark reduziert, unbeabsichtigt Fehler in das Programm einzubauen, von höchster Wichtigkeit. Dies bedeutet in erster Linie das Schreiben von übersichtlichem, klar strukturiertem und ausgiebig kommentiertem Code in allen relevanten Bereichen. Ein nicht zu unterschätzendes Hilfsmittel sind sogenannte Assertions. Dabei handelt es sich um Tests auf Programmzustände, die nach Meinung des Programmierers niemals eintreten dürfen<sup>22</sup>. Tritt ein solcher Zustand dennoch ein, liegt offensichtlich ein logischer Fehler vor und dieser wird mit einem Programmabbruch quittiert. Eine ausgiebige Verwendung dieser Assertions senkt das Fehlerrisiko ungemein, hat aber auch eine entsprechend längere Ausführungszeit zur Folge. Dies gilt besonders, wenn das Testen der Annahmen noch zusätzliche Berechnungen von Größen erfordert. Daher ist die allgemeine Vorgehensweise die, dass man, während ausgiebiger Testläufe des Programms, die Assertions auswerten lässt und sie dann im Produktiveinsatz der Software abschaltet. Dem liegt natürlich die Annahme zugrunde, dass ein Fehler, der in einem Produktivlauf auftritt, auch in einem entsprechenden Testlauf aufgetreten wäre. Dies erfordert durchdachte Testläufe, die möglichst alle Eventualitäten abdecken.

Auch wenn es, wie bereits erwähnt, aufgrund der stochastischen Natur der Monte-Carlo-Algorithmen keine objektive Möglichkeit gibt, die korrekte Arbeitsweise mit deterministischen Tests sicherzustellen, so besteht die Möglichkeit einer manuellen Verifikation mithilfe von speziellen Testläufen. Dazu kommen im vorliegenden Fall zwei Methoden zum Einsatz.

Die erste ist, für kleine Gitter Ergebnisse zu berechnen und diese mit aus exakten Methoden wie der Aufzählung aller Zustände gewonnenen Daten in der Literatur zu vergleichen. Konkret findet sich in [24] eine Reihe von Energieerwartungswerten für ein  $6 \times 6$ -Gitter für verschiedene Temperaturen. Bei dieser geringen Systemgröße lässt sich leicht eine riesige Anzahl Samples erzeugen, die innerhalb der dann kleinen statistischen Grenzen mit den Vergleichswerten übereinstimmen sollte.

Die andere Methode greift auf die beiden implementierten Algorithmen von Wolff sowie Swendsen und Wang zurück. In Abschnitt 4.2 wurde die grundsätzlich unterschiedliche Implementierung dieser bereits diskutiert. Nichtsdestotrotz müssen beide innerhalb ihrer jeweiligen statistischen Grenzen das gleiche Ergebnis für die Korrelationsfunktion vorherzusagen. Wenn das der Fall ist, kann man daraus schließen, dass wohl beide Algorithmen korrekt arbeiten, denn die Implementierung bei beiden ist so unterschiedlich, dass es sehr unwahrscheinlich ist, in beiden einen gleichartigen Fehler zu machen, ohne dass dieser auffällt. All das bisher Beschriebene ist im Modul `TEST` implementiert, das beim Aufruf von `make test` einen ausgiebigen Testlauf startet.

Zuletzt verbleibt noch ein Test des Zufallszahlengenerators. Tests auf Zufälligkeit einer Sequenz von Zahlen sind sehr anspruchsvoll und können aufwändig sein. Daher sind die als Bibliotheken eingebundenen Zufallsgeneratoren selbst nicht weiter explizit getestet, auch deswegen, weil es sich um weithin akzeptierte Standardgeneratoren handelt, für die

---

<sup>21</sup>Diese Aussage bezieht sich nur auf automatische Tests, die das unbemerkte Einführen von Fehlern bei Änderungen im Quelltext verhindern sollen. Bei der manuellen Analyse und Behebung von einmal entdeckten Programmfehlern ist eine vorhersagbare, reproduzierbare Sequenz von Zufallszahlen natürlich ein essentielles Hilfsmittel.

<sup>22</sup>Ein triviales Beispiel wäre, dass das Cluster niemals mehr Elemente enthalten darf als das Gitter.

ausgiebige Untersuchungen vorliegen [40].

Dennoch stehen die Qualität der Ergebnisse von Monte-Carlo-Simulationen mit der Qualität der verwendeten Pseudozufallszahlen in einem engen Zusammenhang. Insbesondere bereitet ein Bericht [41] über eine spezielle Empfindlichkeit des Wolffschen Algorithmus bezüglich bestimmter Mängel in Hinsicht auf die Autokorrelation der vom verwendeten Zufallszahlengenerator erzeugten Sequenz Anlass zur Sorge. Aus diesem Grund stehen zwei Generatoren, WELL und Mersenne-Twister, im Programm zur Auswahl. Bisher förderte jedoch kein Testlauf mit den verschiedenen Generatoren signifikant unterschiedliche Ergebnisse zu Tage.

#### 4.5. Programmbenutzung

Um den verschiedenen Bedürfnissen des Anwenders gerecht zu werden, lässt sich das Verhalten des Programms mit verschiedenen Parametern beeinflussen. Das kann auf zwei Wegen geschehen.

Zum einen lassen sich die Einstellungen, die sich zwischen mehreren Produktivläufen voraussichtlich nicht ändern werden, bequem an den dafür vorgesehenen Stellen im Makefile vor dem Übersetzen anpassen. Dabei handelt es sich im speziellen um die Wahl von Zufallszahlengenerator und zu verwendendem Algorithmus sowie die Dimensionalität des Systems. Letztere kann von eindimensional bis zu praktisch nicht mehr sinnvoll handhabbarer Größe gewählt werden.

Zum anderen können die Parameter, in denen sich mehrere parallel stattfindende Produktivläufe vermutlich unterscheiden werden, als Kommandozeilenargumente übergeben werden. Dies betrifft unter Anderem die Gitterausmaße selbst, die Temperatur, bei der die Simulation stattfinden soll, die Anzahl der zu nehmenden Samples nach einer zu spezifizierenden Equilibrationszeit sowie die bei der Analyse verwendete maximale Blocklänge. Der Schalter `-h` oder `--help` beim Programmaufruf gibt Hilfestellung dazu, in welchem Format die Optionen genau anzugeben sind. Zur Laufzeit hingegen ist eine Interaktion mit dem Programm nicht vorgesehen.

Das Hauptprogramm befindet sich im Modul `MAIN`. Es nimmt die per Kommandozeile spezifizierten Optionen entgegen und wertet diese aus. Dann wird gemäß den gewünschten Parametern die Simulation des Ising-Modells erzeugt, ausgeführt und der gewonnene Satz von Samples analysiert um im Anschluss das Ergebnis auszugeben. Es wird immer die Zeitscheibenkorrelationsfunktion über die gesamte Gitterlänge in der  $x_1$ -Richtung gemessen, so wie es in Abbildung 5 auf Seite 17 bildlich für ein 3D-Gitter dargestellt ist. Im Diagnosemodus, für den der Quelltext mit `make debug` übersetzt werden muss, werden noch weitere Informationen über das, was intern passiert, ausgegeben.

Außerdem existiert ein Kalibrationsmodus, mit dessen Hilfe die notwendige Equilibrationszeit abgeschätzt werden kann. Weil dies mit der Zeitscheibenkorrelationsfunktion eher schwierig ist, dient als Indikator die mittlere Energie pro Spin. Deren absoluter sowie gemittelter Wert werden zu jedem erfolgten Sweep ausgegeben. Die Auswertung muss nach Augenmaß erfolgen. Das folgende Kapitel enthält ein konkretes Beispiel, wie dabei vorzugehen ist.

## 5. Ergebnisse

Gegenstand der Arbeit ist die Untersuchung des Ising-Modells beim Übergang von einem zwei- zu einem dreidimensionalen Gitter mit periodischen Randbedingungen. Konkret wird das Verhalten des kritischen Exponenten  $\nu$  sowie der kritischen Temperatur  $\beta_c$  beobachtet, während man ein quasi-planares System schrittweise zu einem kubischen System verbreitert.

Dazu wurden mit dem im vorangegangenen Kapitel vorgestellten Programm auf dem universitätseigenen Hochleistungs-Rechnerverbund PALMA der WWU Münster Berechnungen mit einer Prozessorzeit von insgesamt rund 2600 Stunden ausgeführt. Die Präsentation der Endergebnisse und eine nachvollziehbare Beschreibung, auf welche Weise diese aus den Simulationsdaten konkret gewonnen wurden, ist Gegenstand dieses Kapitels.

Die nachfolgend präsentierten Fits sind mit der Implementierung des Marquardt-Levenberg-Algorithmus durch das Programm Gnuplot in der Version 4.6.3 erzeugt. Es handelt sich um die Methode der kleinsten Quadrate für nichtlineare Funktionen, wobei die Werte umgekehrt proportional zum Quadrat des angegebenen Fehlers gewichtet eingehen. Bei den im Ergebnis der Fits angegebenen Fehlern handelt es sich jeweils um den entsprechenden Standardfehler des gefitteten Parameters.

### 5.1. Validierung der Implementierung

Der erste Test für die Implementierung der Cluster-Algorithmen ist der Vergleich von Energiewerten mit Literaturwerten. Konkret werden als Vergleichswerte die mittleren Energien pro Spin im  $6 \times 6$ -Gitter zu acht verschiedenen Temperaturwerten herangezogen. Diese finden sich in [24] und wurden dort durch Aufzählung aller möglichen Zustände exakt bestimmt, auch wenn leider nur die ersten drei Nachkommastellen angegeben sind. Bei dieser geringen Systemgröße lassen sich um die  $5 \cdot 10^6$  Samples in wenigen Minuten erzeugen. Die Monte-Carlo-Methode reproduziert die Literaturwerte ohne Probleme mit beiden Clusteralgorithmen sogar mit noch mehr Dezimalstellen innerhalb des statistischen Fehlers.

Davon ermutigt, lässt sich die Energie pro Spin als Indikator für die nötige Equilibrierungszeit bei der Messung der Zeitscheibenkorrelationsfunktion benutzen. Dazu dient der im Programm eingebaute Kalibrierungsmodus, mit dem Abbildung 9 erzeugt wurde. Es wird exemplarisch die Abhängigkeit der absoluten Energiewerte sowie des Mittelwerts von der Anzahl der Sweeps beim Swendsen-Wang-Algorithmus für eines der untersuchten Gitter gezeigt. Es sind zwei Anfangskonfigurationen dargestellt, einmal der Cold Start, wo alle Spins zu Beginn parallel ausgerichtet sind, und einmal der Hot Start, wo alle Spins mit einem zufälligen Wert initialisiert werden. Die absoluten Energien fluktuieren schon nach wenigen Sweeps um dem langfristigen Mittelwert. Man kann sicher sagen, dass das System equilibriert ist, sobald sich die Mittelwerte zu den verschiedenen Startbedingungen schneiden. Dies ist bei ungefähr 2000 Sweeps der Fall.

Für andere Gittergrößen und Temperaturen als das hier untersuchte  $64 \times 64 \times 8$ -Gitter bei  $\beta = 0.222$  ergibt sich ein ähnliches Bild. Beim Wolff-Algorithmus konvergieren die Mittelwerte sogar noch schneller, da jeder Sweep aus mehreren Clusterupdates besteht, und nicht bei jedem Update ein Sample genommen wird.

Nun muss die Equilibrierungszeit bezüglich der Observablen Zeitscheibenkorrelationsfunktion zwar nicht exakt gleich sein, sollte aber in etwa in der gleichen Größenordnung liegen. Um ganz sicher zu gehen, wurde bei der Bestimmung der Korrelationslänge die 50-fache Equilibrierungszeit, also insgesamt  $10^5$  Sweeps angesetzt, was immer noch lediglich

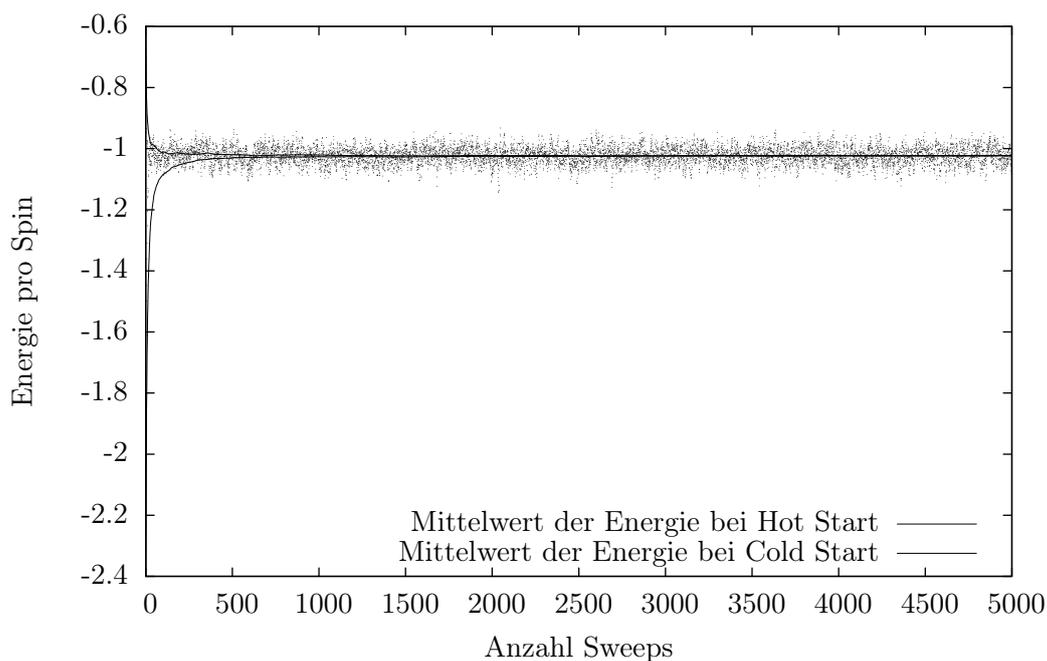


Abbildung 9: Die Equilibrierung des  $64 \times 64 \times 8$ -Ising-Modells bei  $\beta = 0.222$  für Hot bzw. Cold Start Anfangskonfigurationen in Abhängigkeit von der Monte-Carlo-Zeit beim Swendsen-Wang-Algorithmus. Die Punkte entsprechen den zur jeweiligen Zeit absolut gemessenen Energien.

einen verschwindenden Bruchteil der gesamten Rechenzeit ausmacht.

Die nächste Hürde ist der Vergleich der Zeitscheibenkorrelationsfunktionen  $c(t)$ , die jeweils mit der Implementierung des Wolffschen und des Algorithmus von Swendsen und Wang bestimmt wurden. Abbildung 10 zeigt die beiden Resultate für das gleiche System, an dem oben der Equilibrierungsvorgang diskutiert wurde. Ein Unterschied zwischen den Samples aus Wolff- und Swendsen-Wang-Algorithmus ist mit bloßem Auge nicht zu erkennen. Die verwendeten Symbole würden sich überdecken, wenn die beiden Kurven in der Darstellung nicht extra auseinander gezogen worden wären. In der halb-logarithmischen Darstellung sieht man schön, dass der funktionale Verlauf sich aus zwei verschobenen Exponentialfunktionen, die sich nur im Vorzeichen des Exponenten unterscheiden, zusammensetzt. Mit linearer Achsenskalierung würde sich das zu erwartende Bild eines Hyperbelcosinus ergeben.

Die symmetrische Form folgt mit den periodischen Randbedingungen aus der Translationsinvarianz der Zeitscheibenkorrelationsfunktion. Das Minimum liegt auch wie erwartet genau bei 32, der halben Seitenlänge. Um Korrelationen zwischen den Werten von  $c(t)$  für die einzelnen  $t$  zu minimieren und somit systematische Fehler zu vermeiden, wurde für jedes Sample von  $c(t)$  ein eigenes Cluster benutzt und  $t$  in der Reihenfolge  $t = 0, 2, 4, \dots, 60, 62, 63, 61, \dots, 3, 1$  durchlaufen.

Die numerischen Werte für  $c(t)$  lassen sich gemäß Gleichung (49) fiten. Man erhält  $\xi_{\text{Wolff}} = 10.116(1)$  und  $\xi_{\text{Swendsen-Wang}} = 10.111(5)$  aus den vorliegenden Daten. Die Ergebnisse stimmen also innerhalb ihrer statistischen Fehlergrenzen überein.

Gemäß Gleichung (48) setzt sich  $c(t)$  aus einer Reihe von Exponentialfunktionen

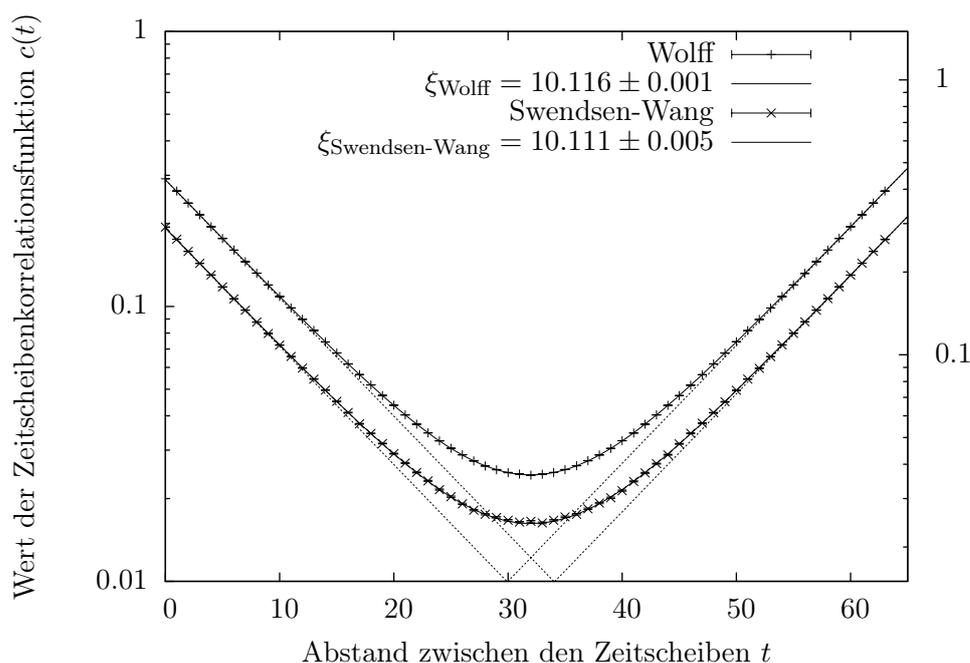


Abbildung 10: Die für das  $64 \times 64 \times 8$ -Ising-Modell bei  $\beta = 0.222$  gemessene Zeitscheibenkorrelationsfunktion in halblogarithmischer Darstellung mit Fit für die Korrelationslänge. Für den Wolff-Algorithmus gilt die linke y-Achse, für Swendsen-Wang die rechte. Beide sind etwas gegeneinander versetzt, da die beinahe perfekte Überdeckung die Übersichtlichkeit sonst deutlich reduziert. Im Fit sind die ersten vier und die letzten drei Werte nicht berücksichtigt. Bei den angedeuteten, geraden Linien handelt es sich um die Exponentialfunktionen, deren Summe jeweils die Fitkurve ergibt.

zusammen, im Fit ist aber nur der erste Summand berücksichtigt. Die höheren Ordnungen machen sich aber für sehr kleine oder sehr große  $t$  noch bemerkbar, wohingegen der Bereich dazwischen klar vom niedrigsten Reihenglied dominiert wird. Empirisch ergeben sich die geringsten Fehler für die Korrelationslänge, wenn  $0 \leq t \leq 3$  und  $61 \leq t \leq 63$  im Fit ausgelassen werden.

Die in Abbildung 10 angegebenen Fehlerbalken sind sehr klein, sodass man genau hinsehen muss, um sie zu erkennen. Sie wurden gemäß Gleichung (30) bestimmt, da die einzelnen Samples praktisch unkorreliert sind. Dies ist nämlich aus Abbildung 11 ersichtlich, die der Untersuchung mit Data Blocking auf Autokorrelationen hin dient. Gemäß der Beschreibung in Abschnitt 3.5 wäre bei korrelierten Samples ein Anstieg des Standardfehlers von Anfang an zu beobachten, was jedoch überhaupt nicht der Fall ist. Im Gegenteil, beim Swendsen-Wang-Algorithmus bleibt er konstant bis zu einer Blocklänge von  $\approx 2^5$ . Danach sinkt die Anzahl der noch verbleibenden Blöcke unter 5000 und der Fehler des Fehlers nimmt stark zu, was sich in verrauschten Werten für den Fehler bemerkbar macht. Für den Wolff-Algorithmus gilt genau das gleiche, nur dass die Schwelle für unzuverlässig große Fehler des Fehlers bei ungefähr  $2^{12}$  liegt. Somit können aufeinanderfolgende Samples für die  $c(t)$  zu festem  $t$  als unkorreliert betrachtet werden.

Für den Swendsen-Wang-Algorithmus liegen 131072 Samples für jedes  $c(t)$  nach einer

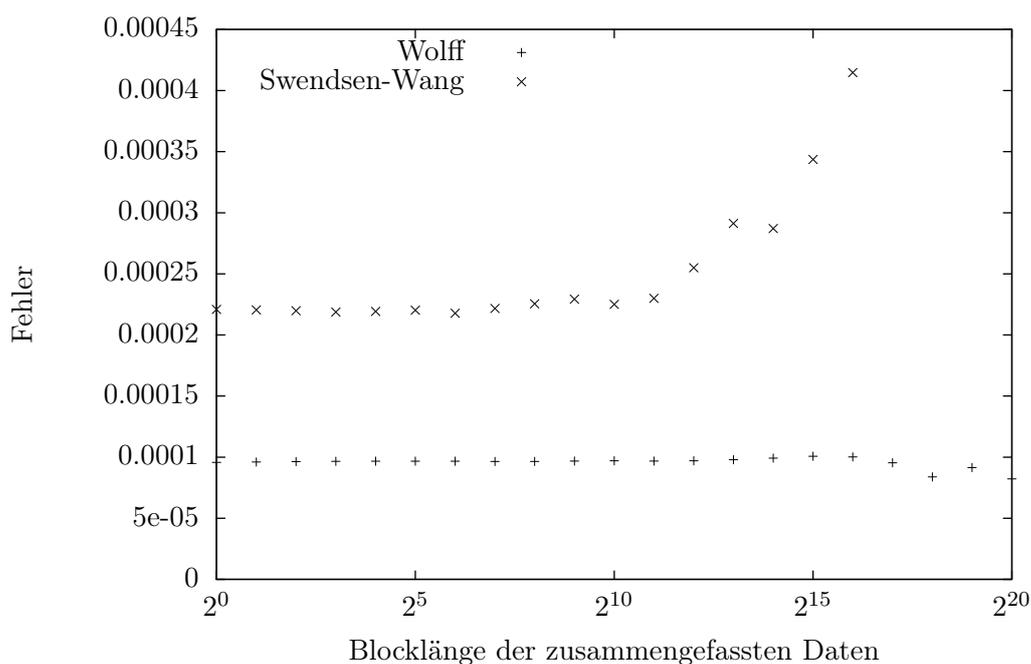


Abbildung 11: Abhängigkeit des Fehlers von der Blocklänge für  $c(4)$  beim  $64 \times 64 \times 8$ -Ising-Modell. Für den Wolff-Algorithmus wird die Anzahl der verbleibenden Blocks ab einer Blocklänge von  $2^{12}$  so gering, dass sich statistische Schwankungen in den Daten bemerkbar machen. Für Swendsen-Wang ist das aufgrund der viel geringeren Anzahl Samples sogar schon ab  $2^5$  der Fall. Nichtsdestrotz ist kein Anstieg des Fehlers mit Zunahme der Blocklänge von Anfang an erkennbar, sodass aufeinanderfolgende Samples scheinbar komplett unkorreliert sind!

Equilibrierungszeit von  $10^5$  Sweeps vor, während es beim Wolff-Algorithmus 16777216 Samples sind. Die krummen Zahlen erklären sich dadurch, dass es Potenzen von zwei sind, was das Zusammenfassen in Blöcke erheblich vereinfacht. Die Laufzeit dafür ist beim Wolffschen Algorithmus mit etwa 9 Stunden etwas geringer als die circa 11 Stunden, die Swendsen-Wang für weniger als ein Hundertstel der Anzahl an Samples entsprechend benötigt. Dass die Fehler bei beiden dennoch die gleiche Größenordnung haben, liegt daran, dass jedes Sample beim Swendsen Wang-Algorithmus bereits über viele Cluster gemittelt ist, während für den Wolff-Algorithmus immer nur ein einziges Cluster pro Sample betrachtet wird. Nichtsdestotrotz beträgt der Fehler, mit dem die Werte für die  $c(t)$  nicht nur für den in Abbildung 11 gezeigten Wert  $t = 4$ , sondern für alle  $t$  im Swendsen-Wang-Fall vorliegen, etwa das Doppelte gegenüber Wolff. Daraus resultiert letztlich der etwa vier mal so große Fehler in der Korrelationslänge im Fit in Abbildung 10. Gemäß Gleichung (30) wäre also für Ergebnisse von gleicher Qualität beim Swendsen-Wang-Algorithmus die vier- bis fünffache Laufzeit notwendig. Aus diesem Grund empfiehlt es sich, in Produktivläufen ausschließlich den Wolffschen Algorithmus zu nutzen.

## 5.2. Korrelationslängen

Es liegen Ergebnisse für insgesamt 240 Korrelationslängen  $\xi$  vor, deren numerische Bestimmung mit dem Wolff-Algorithmus analog zur eben beschriebenen Vorgehensweise verlief. Diese  $\xi$  verteilen sich auf 10 verschiedene Gitter der Größe  $64 \times 64 \times T$  mit  $T \in \{1, 2, 3, 4, 5, 6, 8, 16, 32, 64\}$  zu je acht verschiedenen Temperaturen mit nochmals je zwei sehr nah benachbarten Temperaturen im Abstand  $\Delta\beta$ , um die Ableitung der Korrelationslänge im jeweiligen Punkt numerisch zu bestimmen.

Die Temperaturen  $\beta$  sind jeweils so gewählt, dass die Korrelationslänge sich im Bereich  $5 \lesssim \xi(\beta) \lesssim 10$  befindet. Der Festlegung auf diesen Bereich liegt die Annahme zugrunde, dass einerseits die Korrelationslänge zwar klein genug ist, sodass Finite Size Effekte bzgl.  $L$  noch nicht auftreten, andererseits die Entfernung vom kritischen Punkt aber noch gering genug ist, damit das Potenzgesetz mit dem kritischen Exponenten in Gleichung (19) noch Gültigkeit behält.

Die genauen Werte, die numerisch bestimmt wurden, befinden sich in Tabellenform in Anhang B ab Seite 53 zusammen mit einigen weiteren Informationen. Die digitale Version dieser Arbeit enthält außerdem die Fits, mit denen die einzelnen Korrelationslängen bestimmt wurden im Anhang. Jeweils drei Korrelationslängen  $\xi(\beta)$  liegen zu sehr nah zusammen befindlichen Temperaturen  $\beta$  und  $\beta \pm \Delta\beta$  vor. Die aus den  $\xi(\beta \pm \Delta\beta)$  mit der zentralen Differenzmethode nach Gleichung (41) numerisch gewonnenen Ableitungen nach der Temperatur  $\beta$  zu jeder dieser acht Korrelationslängen  $\xi(\beta)$  pro Gitter befinden sich in der jeweiligen Tabelle direkt darunter. Es ist zu beachten, dass der zu jeder Ableitung angegebene Fehler sich nur auf den statistisch bedingten Fehler nach Gleichung (42) bezieht, wohingegen der exakte Fehler durch das Vernachlässigen von Ableitungen dritter und höherer Ordnung unbekannt ist!

## 5.3. Kritischer Exponent und kritische Temperatur

Die besagten Werte für  $\xi(\beta)$  und  $\xi'(\beta) = \frac{\partial \xi(\beta)}{\partial \beta}$  lassen sich zu  $\frac{\xi(\beta)}{\xi'(\beta)}$  zusammenfassen und gemäß Gleichung (40) fitten um daraus  $\nu$  und  $\beta_c$  zu den 10  $64 \times 64 \times T$ -Ising-Modellen zu bestimmen. Die Fits befinden sich in Anhang A ab Seite 48. Der in den dortigen Abbildungen angegebene Fehler berechnet sich nach Gaußscher Fehlerfortpflanzung aus dem statistischen Fehler  $\Delta\xi$  der Korrelationslänge und dem eben angesprochenen Fehler der Ableitung  $\Delta\xi'$  gemäß Gleichung (42) jeweils bei der Temperatur  $\beta$  gemäß

$$\Delta \left( \frac{\xi}{\xi'} \right) = \sqrt{\left( \frac{\Delta\xi}{\xi'} \right)^2 + \left( \frac{\xi \Delta\xi'}{\xi'^2} \right)^2}. \quad (66)$$

Umgekehrt proportional zum Quadrat dieses Fehlers sind die in die Fits eingehenden Werte gewichtet.

Die Resultate der Fits sind in Tabelle 1 zusammengefasst und in den Abbildungen 12 und 13 graphisch dargestellt. In den beiden Abbildungen sind zum Vergleich auch die nach Abschnitt 2.1.2 theoretisch vorhergesagten Werte im unendlich großen, zweidimensionalen System,  $\nu = 1$  und  $\beta_c \approx 0.440687$  eingetragen. Genauso sind für den dreidimensionalen Fall numerische Werte dabei, die mit Hilfe von Renormierungsgruppenrechnungen aus Monte-Carlo-Daten extrapoliert wurden. Diese Werte sind [42] entnommen und lauten  $\nu = 0.629(4)$  und  $\beta_c = 0.221654(6)$ .

Die hier bestimmten Werte sind in den Grenzfällen von planarem und kubischem Gitter jeweils unter Einbeziehung des Fehlers im Vergleich mit dem Erwarteten zu groß. So liegt

$T$	$\nu$	$\beta_c$
1	1.057(10)	0.4418(3)
2	1.026(14)	0.27633(21)
3	0.988(10)	0.24602(9)
4	0.975(8)	0.23606(6)
5	0.945(24)	0.23134(17)
6	0.904(11)	0.22854(7)
8	0.839(8)	0.22567(5)
16	0.784(8)	0.22291(3)
32	0.680(7)	0.22187(3)
64	0.657(9)	0.22171(4)

Tabelle 1: Die kritischen Exponenten  $\nu$  sowie die pseudokritischen Temperaturen  $\beta_c$ , die für das  $64 \times 64 \times T$  Ising-Modell bestimmt wurden, im Überblick.

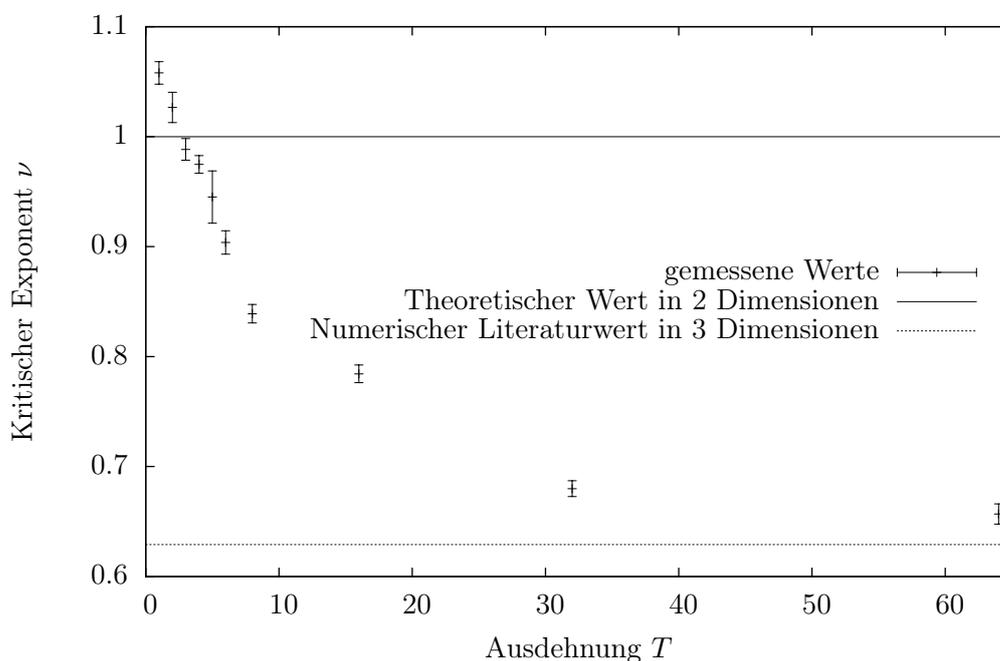


Abbildung 12: Der kritische Exponent  $\nu$  für das  $64 \times 64 \times T$  Ising-Modell in Abhängigkeit von  $T$ . Der angegebene, numerisch bestimmte Wert stammt aus [42].

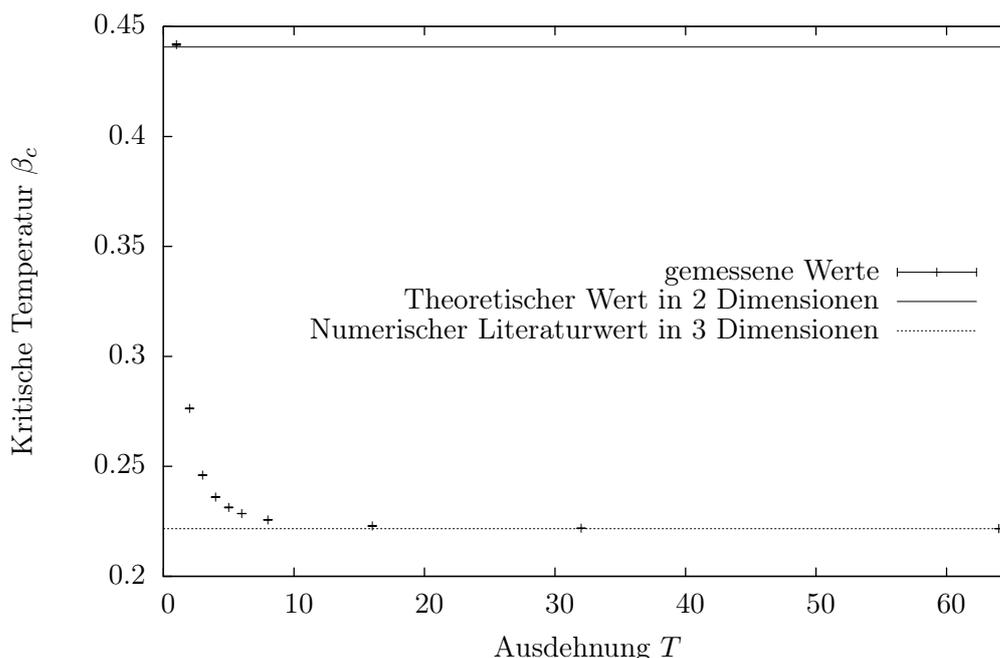


Abbildung 13: Die (pseudo-)kritische Temperatur  $\beta_c$  für das  $64 \times 64 \times T$  Ising-Modell in Abhängigkeit von  $T$ . Der angegebene, numerisch bestimmte Wert stammt aus [42].

beim planaren Gitter der numerisch bestimmte Wert für die kritische Temperatur mit  $0.4418(3)$  etwas höher als  $0.44046\dots$ , dem theoretisch erwarteten Wert. Genauso ist der kritische Exponent  $1.057(10)$  etwas zu groß im Vergleich zu 1.

Für das kubische Modell sieht die Sache ähnlich aus. Hier befindet sich die kritische Temperatur mit einem Wert von  $0.22171(4)$  vs.  $0.221654(6)$  mit den Fehlergrenzen etwas oberhalb dem Literaturwert. Gleiches gilt für den kritischen Exponenten im dreidimensionalen Fall, wo der eigene Wert mit  $0.657(9)$  gegenüber  $0.629(4)$  ebenfalls deutlich nicht mit dem Literaturwert kompatibel ist. Der Schluss liegt nahe, dass vermutlich auch der dimensionelle Übergang einem systematischen Fehler unterliegt.

Der Übergang des kritischen Exponenten vom zweidimensionalen Gitter bei  $T = 1$  zum kubischen Gitter bei  $T = 64$  verläuft an sich eher gemächlich. Der Übergang der kritischen Temperatur hingegen geht viel schneller vonstatten. Statistisch signifikante Ausreißer sind immerhin keine dabei.

Für das Verhalten der kritischen Temperatur beim dimensionellen Crossover gibt es sogar einen theoretischen Unterbau. Gemäß [14] gilt für den dimensionellen Übergang von einem  $(d - 1)$ -dimensionalen System zum entsprechenden  $d$ -dimensionalen System ein Zusammenhang ähnlich zu Gleichung (21) für die *relative Verschiebung* des pseudokritischen

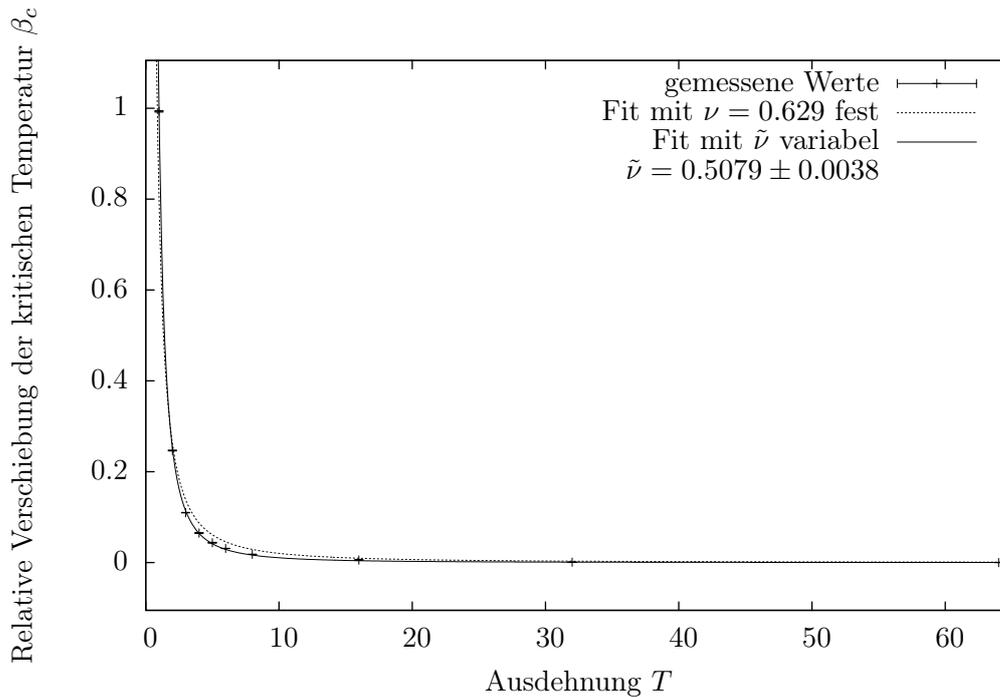


Abbildung 14: Relative Verschiebung der kritischen Temperatur für das  $64 \times 64 \times T$  Ising-Modell in Abhängigkeit von  $T$ . Der benutzte Wert für  $\beta_c(T = \infty)$  stammt wieder aus [42]

Punktes bei  $T_c$ <sup>23</sup> bzw.  $\beta_c$  in Abhängigkeit der Dicke  $T$  gemäß

$$\left| \frac{T_c(T = \infty) - T_c(T)}{T_c(T = \infty)} \right| \stackrel{(18)}{\approx} \left| \frac{\beta_c(T = \infty) - \beta_c(T)}{\beta_c(T = \infty)} \right| \propto T^{-\frac{1}{\tilde{\nu}}}. \quad (67)$$

Dabei ist bei der Herleitung der Äquivalenz im Bild von Temperatur und inverser Temperatur nach Gleichung (18)  $\beta_c$  durch  $\beta_c(T = \infty)$  und  $\beta$  durch  $\beta(T)$  sowie  $T_c$  entsprechend zu ersetzen. In Gleichung (67) entspricht  $\tilde{\nu}$  dem kritischen Exponenten im  $d$ -dimensionalen System, im vorliegenden Fall wäre also  $\tilde{\nu} = \nu \approx 0.629$  zu erwarten. Abbildung 14 beinhaltet einen Fit des rechten Teils von Gleichung (67) je einmal zur Bestimmung von  $\tilde{\nu}$  und einmal mit  $\nu$  gemäß dem erwarteten Wert fest. Der resultierende kritische Exponent  $\tilde{\nu} = 0.508(4)$  passt nicht zum erwarteten Literaturwert  $\nu = 0.629(4)$ , sondern liegt deutlich darunter. Auch zeigt sich, dass der Fit zu festem  $\nu$  nicht mit den Datenpunkten übereinstimmt.

Eine vergleichbare theoretische Vorhersage, wie der pseudokritische Exponent  $\nu(T)$  selbst sich beim dimensionellen Crossover verhält, ist dem Autor nicht bekannt.

Alles in allem zeigen sich also signifikante Diskrepanzen zwischen den eigenen numerischen Ergebnissen und den theoretischen Vorhersagen bzw. Ergebnissen aus der Literatur. Mögliche Gründe dafür werden im folgenden Abschnitt eingehend untersucht.

<sup>23</sup>In Kapitel 3 war angekündigt worden, dass nur noch  $\beta$  zur Darstellung der Temperatur benutzt werde. Dies lässt sich hier nicht mehr ganz einhalten, sodass auch  $T_c$  ab und zu für die Temperatur am kritischen Punkt benutzt wird. Dennoch bleibt  $T$  Seitenlänge des Gitters und ist nicht mit der Temperatur zu verwechseln!

## 5.4. Diskussion

Zuallererst ist festzuhalten, dass das theoretisch erwartete Verhalten beim dimensionellen Übergang des  $64 \times 64 \times T$ -Ising-Modells grundsätzlich eintritt. Gemäß [14] sollte ein gleichmäßiger Übergang zwischen den Grenzfällen dünne Scheibe und Kubus stattfinden. Dies ist durchaus der Fall. Für  $T \approx 1$  liegt ein quasi-zweidimensionales Gitter vor und die kritische Temperatur  $\beta_c$  sowie der kritische Exponent  $\nu$  liegen in der Größenordnung dessen, was für ein unendlich ausgedehntes, planares Gitter zu erwarten wäre. Gleiches gilt im Fall  $T = 64$ , bei dem das Gitter einen Kubus von Spins bildet. Auch hier entspricht die Größenordnung von  $\nu$  und  $\beta_c$  dem, was für ein dreidimensionales Ising-System im thermodynamischen Limes zu erwarten wäre. Dazwischen fallen die Werte für beide Größen monoton und ohne überraschende Sprünge. Auch der Bereich der Seitenlänge  $T$ , der im Bereich der untersuchten Korrelationslänge  $5 \lesssim \xi \lesssim 10$  liegt, zeigt keine offensichtlichen Besonderheiten, auch wenn gemäß [43] genau dort der Übergang besonders deutlich zu erwarten wäre.

Etwas überraschend ist, dass sich die kritische Temperatur wesentlich genauer bestimmen lässt als der kritische Exponent. Während sich die Ungenauigkeit des letzteren im Prozentbereich befindet, liegen die Werte für  $\beta_c$  um ein bis zwei Größenordnungen genauer vor. Mit der verwendeten Methode ist es also deutlich schwieriger, zu einem gegebenen System den kritischen Exponenten zu bestimmen als die kritische Temperatur.

Nichtsdestotrotz sind, wie bereits festgestellt, die numerisch bestimmten Größen  $\nu$  und  $\beta_c$  in Tabelle 1 in den beiden Extremfällen, für die Vergleichswerte existieren, durchgehend etwas zu groß und damit vermutlich auch die Werte dazwischen. Dass  $\nu$  zu groß ist, bedeutet  $\frac{1}{\nu}$  ist zu klein, und damit fallen die Geraden in den Plots in Anhang A, in denen Gleichung 40 gefittet wird, nicht schnell genug. Für die  $\beta_c$  gilt ähnliches. Dass die Schnittpunkte mit der  $x$ -Achse zu weit vom Ursprung entfernt liegen, muss bedeuten, dass die Geraden zu langsam fallen, oder dass die gemessenen Korrelationslängen einen gleichmäßigen Versatz nach oben haben.

Es ist in Anbetracht beider Aspekte wahrscheinlich, dass das Produkt  $\xi(\beta) \left( \frac{\partial \xi(\beta)}{\partial \beta} \right)^{-1}$  mit zunehmendem  $\beta$  schlicht zu große Werte annimmt bzw. nicht schnell genug fällt. Besonders problematisch ist, dass der untersuchte Temperaturbereich extrem schmal ist. Daher können kleinste systematische Abweichungen im besagten Produkt schon zu großen Fehlern in der Geradensteigung führen. Um diese zu ergründen, macht es Sinn, zuerst alle dem Autor bekannten Fehlerquellen der Übersicht halber ohne Gewichtung und Anspruch auf Vollständigkeit aufzulisten:

1. Statistische Fehler bedingt durch die Natur der Monte-Carlo-Algorithmen.
2. Eine fehlerhafte Implementierung oder Benutzung des Programms.
3. Die Qualität der Bestimmung der Korrelationslänge mit Hilfe der Zeitscheibenkorrelationsfunktion.
4. Die vernachlässigten Ableitungen höherer Ordnung bei der numerischen Berechnung der ersten Ableitung.
5. Finite Size Effekte in endlichen Systemen. Damit sind auch die folgenden zwei Punkte eng verbunden.
6. Die Gültigkeit der Beschreibung durch kritische Exponenten je nach Abstand vom kritischen Punkt an sich.

7. Die Gültigkeit der Äquivalenz der Beschreibung im Bild von Temperatur und inverser Temperatur nach Gleichung (18) in Abhängigkeit von der Entfernung zum kritischen Punkt.

Der erste Punkt, die statistischen Fehler, wurde so gut wie möglich bei der gesamten Prozedur berücksichtigt. Dies beginnt damit, dass jedes mit dem Wolff-Algorithmus erzeugte Cluster stets nur für ein Sample der Zeitscheibenkorrelation  $c(t)$  verwendet wurde, um Korrelationen bezüglich  $t$  im Verlauf von  $c(t)$  zu minimieren. Weiterhin wurden die Samples für  $c(t)$  nicht linear in  $t$  aufgenommen, sondern versetzt und dabei abwechselnd von kleinen  $t$  zu großen  $t$  und umgekehrt, sodass verbleibende Korrelationen bezüglich  $t$  sich zumindestens kompensieren sollten. Die Samples für  $c(t)$  mit  $t$  fest wurden per Data Blocking untersucht und es konnte kein Anwachsen des Fehlers für kleine Blockgrößen festgestellt werden, sodass die Autokorrelationszeit auf die Größenordnung  $\tau \approx 1$  geschätzt werden kann. Damit lässt sich der statistische Fehler der Samples von  $c(t)$  recht genau bestimmen. Diese Samples sind im Fit von Gleichung (40) zur Bestimmung der Korrelationslänge umkehrt proportional zum Quadrat ihres statistischen Fehlers gewichtet. Der statistische Fehler der resultierenden Korrelationslänge ist hauptsächlich abhängig von der bei großer Anzahl Samples gering werdenden Streuung der Mittelwerte von  $c(t)$ . Er geht mittels Gaußscher Fehlerfortpflanzung gemäß Gleichung (66) in den darauf folgenden Fit zur Bestimmung von  $\nu$  und  $\beta_c$  ein, wieder zum reziproken Fehlerquadrat gewichtet. Der Behandlung des statistischen Fehlers wurde also sehr viel Beachtung geschenkt und daher sollte er im Fehler des Endergebnisses angemessen repräsentiert sein.

Auch auf die Korrektheit der Implementierung des Programms sowie der anschließenden Skripts zur automatischen Auswertung der Daten bis einschließlich der Berechnung der einzelnen  $\nu$  und  $\beta_c$  und deren korrekte Benutzung wurde besonderes Augenmerk gelegt. Im Abschnitt 4.4 wurden die wesentlichen Aspekte beim Test der Implementierung bereits diskutiert und in Abschnitt 5.1 die Ergebnisse der Validierung der korrekten Berechnung der Korrelationslänge dargelegt. Dass dem Programm die korrekten Parameter übergeben wurden, wurde explizit überprüft. Um zuletzt die Korrektheit der Skripts zur automatischen Auswertung zu testen, wurden stichprobenartig Berechnungen von Hand nachgeprüft und keine Abweichungen gefunden. Dass die systematischen Abweichungen von  $\nu$  und  $\beta_c$  also auf einen Softwarefehler zurückzuführen sind, ist demnach unwahrscheinlich.

Sicherlich einen Einfluss auf das Endergebnis hat die Bestimmung der Korrelationslänge mittels der Zeitscheibenkorrelationsfunktion. Schließlich vernachlässigt ein Fit von Gleichung (40) die höheren Ordnungen. Um dies zu kompensieren, sind in den Fits von  $c(t)$  jeweils die Werte  $t = 0, 1$  und  $63$  nicht berücksichtigt, jedoch bei allen Korrelationslängen gleichermaßen. Dies stellt sich als optimal für geringste statistische Fehler in den Korrelationslängen insgesamt heraus. Wie das Beispiel in Abschnitt 5.1 jedoch zeigt, kann das Optimum eines individuellen Fits jedoch woanders liegen. Im Allgemeinen erweist es sich als besser, bei kleinen  $\beta$  und großen Gittern weniger Werte von  $t$  auszulassen, als bei kleineren Gittern und größeren  $\beta$ . Daher wäre anzuraten, anstatt einer globalen Vorgabe jeweils individuell zu prüfen, wie viele Werte bei der Bestimmung der Korrelationslänge optimalerweise nicht berücksichtigt werden. Die Werte für  $\nu$  und  $\beta$  ändern sich zwar geringfügig innerhalb der eigenen Fehlergrenzen zwischen dem Auslassen von gar keinen Werten und dem Auslassen der ersten vier und letzten drei Werte bei allen Korrelationslängen, aber eher zufällig als systematisch und bei Weitem nicht genug, um die systematische Abweichung zu erklären. Um den Einfluss der höheren Ordnungen genauer zu untersuchen, wäre es sicher sinnvoll, höhere Ordnungen in ausgewählten Fits mit einzubeziehen und die dann ermittelten Korrelationslängen zu untersuchen. Diese

Fits müssten jedoch mit einer größeren Anzahl Samples für  $c(t)$  durchgeführt werden, da mit den vorliegenden Datensätzen schon keine sinnvolle Konvergenz des Fits, der nur die zweite Ordnung mit einschließt, zu erzielen ist.

Das Vernachlässigen der höheren Ordnungen bei der numerischen Berechnung der Ableitung kann definitiv ein Faktor sein, der zum systematischen Fehler beiträgt. Wenn beispielsweise die dritte Ableitung von  $\xi(\beta)$  negativ ist, also die Krümmung des Verlaufs von  $\xi(\beta)$  mit zunehmendem  $\beta$  monoton abnimmt, so wären die numerisch bestimmten Werte der Ableitung  $\xi'(\beta)$  mit zunehmendem  $\beta$  gemäß Gleichung (41) systematisch zu klein. Da  $\xi'(\beta)$  reziprok in den Fit von Gleichung (40) eingeht, würde das genau ein zu flaches Abfallen der in Anhang A gefitteten Geraden bedeuten. Nun lässt sich mit Werten für  $\xi(\beta)$  und den Nachbarn  $\xi(\beta \pm \Delta\beta)$  nur maximal die zweite Ableitung berücksichtigen. Dennoch gibt es zu jedem Gitter acht Werte in gleichem Abstand voneinander, nämlich die, zu denen die Ableitungen jeweils bestimmt werden. Man kann jeweils zu den vier niedrigeren und den vier höheren  $\beta$  die dritte Ableitung numerisch bestimmen. Eine kurze Stichprobe mit den Werten in den Tabellen 3, 15 und 21 zeigt, dass bei durchgehend negativem Vorzeichen der Betrag der dritten Ableitung in diesen Fällen um einen Faktor zwischen 1,5 und 3 zunimmt. Dies trägt nach Gleichung (43) definitiv zum systematischen Fehler in  $\nu$  und  $\beta_c$  bei. Um diesen Beitrag zu reduzieren, wäre es anzuraten, die Simulation mit zwei zusätzlichen Nachbarn  $\xi(\beta \pm 2\Delta\beta)$  oder noch mehr zu jedem  $\xi(\beta)$  zu wiederholen und eine numerische Formel für die erste Ableitung zu benutzen, die die höheren Ableitungen entsprechend mit einbezieht. Davon wären genauere Ergebnisse für  $\nu$  und  $\beta_c$  zu erwarten.

Auch Finite Size Effekte spielen sicherlich eine Rolle, was den systematischen Fehler von  $\nu$  und  $\beta_c$  angeht, da sich bei endlichen Gittern die Position des kritischen Punktes gemäß Gleichung (21) verschiebt. Der genaue Fehler ist ohne weitere Simulationen bei anderen Gittergrößen jedoch nur schwer zu quantifizieren. Allerdings ist eine Seitenlänge von 64 schon relativ groß, sodass der zu erwartende Beitrag gering sein dürfte. Da Finite Size Effekte bei Zunahme der untersuchten Dimension im Allgemeinen abnehmen, sind für das kubische Gitter geringere Korrekturen zu erwarten als für die scheibenartigen Gitter. Genauere Aussagen lassen sich aber erst nach einer Wiederholung der Simulation mit Seitenlängen  $L$  von z. B. 32 oder 128 unter Betrachtung gleichartiger Verhältnisse  $\frac{T}{L}$  bei sonst gleichen Parametern treffen. Danach ließen sich die Beiträge zum systematischen Fehler gemäß Gleichung (21) einschätzen.

Ähnliches gilt für die Frage danach, inwieweit bei der betrachteten Entfernung  $|\beta_c - \beta|$  vom kritischen Punkt eine Beschreibung des Systems mittels kritischem Exponenten nach Gleichung (19) überhaupt noch gültig ist, bzw. wie groß der resultierende Näherungsfehler ist. Auch hier scheint das Einzige, was man tun kann um diese Frage zu beantworten, ein erneutes Simulieren zu sein, wobei diesmal  $\xi(\beta)$  aus einem anderen als dem hier benutzten Intervall stammen sollte. Ebenso ließe sich untersuchen, ob der systematische Fehler sich signifikant ändert, wenn man anstatt im Bild der inversen Temperatur  $\beta$  die Berechnungen direkt im Bild von Gleichung (16) durchführt. Erste Berechnungen scheinen darauf hin zu deuten, die rigorose Auswertung einer solchen Untersuchung würde jedoch noch mehr Zeit erfordern, als im Rahmen der Arbeit zur Verfügung steht.

Als letzter, bisher noch nicht erwähnter Punkt wäre noch der Einfluss der periodischen Randbedingungen beim  $64 \times 64 \times 1$ -Gitter zu nennen. Da das Programm unabhängig von der Dimensionalität des Gitters implementiert ist, würde sich zur Überprüfung eine Simulation des zweidimensionalen  $64 \times 64$ -Gitters anbieten. Signifikante Abweichungen in diesem Fall, genauso wie bei der Simulation von vierdimensionalen  $64 \times 64 \times T \times 1$ -Gittern, würden den Autor jedoch sehr überraschen und könnten auf einen bei den bisherigen

Tests nicht entdeckten Fehler in der Implementierung hinweisen.

Nachdem nun die Frage nach dem systematischen Fehler sehr ausführlich diskutiert wurde, verbleibt noch die Behandlung der Überprüfung von Gleichung (67) durch den Fit in Abbildung 14. Der erhaltene Wert für  $\tilde{\nu}$  sollte dem Wert des kritischen Exponenten  $\nu$  im dreidimensionalen Fall entsprechen, das Ergebnis  $\tilde{\nu} = 0.508(4)$  passt mit dem erwarteten Wert  $\nu = 0.629(4)$  jedoch überhaupt nicht zusammen.

In [44] wird das Verhalten der relativen Verschiebung des kritischen Punkts beim dimensionellen Crossover vom zwei- zu dreidimensionalen Ising-Modell ebenfalls untersucht, jedoch mit einer anderen Methode, nämlich der Reihenentwicklung in der Hochtemperaturphase. Interessanterweise ergibt sich dort speziell im Fall periodischer Randbedingungen ein Wert von  $\frac{1}{\tilde{\nu}} = 2.0(1)$ , was grundsätzlich gut zu dem hier gefundenen Resultat passen würde. Eine Untersuchung des gleichen Sachverhalts mit Monte-Carlo-Methoden in [45] kommt jedoch zu einem anderen Ergebnis, nämlich  $\tilde{\nu} = 0.628(3)$ , was auch dem von der Theorie her Erwarteten entspricht und das Ergebnis von [44] wird in [45] zu einem Artefakt der dort verwendeten Methode erklärt.

Da für die numerische Bestimmung in der vorliegenden Arbeit die in den Fit eingehenden Werte gemäß der vorangegangenen Diskussion bereits einem systematischen Fehler unterliegen, überrascht es auch nicht, dass das Resultat fehlerbehaftet ist. Desweiteren ist die Äquivalenz der Bilder in  $\beta_c$  und  $T_c$  in Gleichung (67) in Frage zu stellen. Denn in der Herleitung mittels Gleichung (18) sind Terme ab der Ordnung

$$\mathcal{O}\left(\left(\frac{\delta\beta_c(T)}{\beta_c(T=\infty)}\right)^2\right) \text{ mit } \delta\beta_c(T) = \beta_c(T) - \beta_c(T=\infty) \quad (68)$$

vernachlässigt. Für  $T = 1$  gilt jedoch  $\frac{\delta\beta_c(T=1)}{\beta_c(T=\infty)} \approx 1$ , sodass bei dieser Näherung mit Sicherheit ein beträchtlicher Fehler entsteht. In der Tat wächst der Wert von  $\tilde{\nu}$  kontinuierlich an, je mehr Punkte bei kleinen  $T$  man im Fit auslässt. Jedoch liegen ab  $T = 8$  nicht mehr genügend Punkte vor, sodass die Qualität des Fits ab dort rapide nachlässt. Für präzisere Resultate wären also eine Menge Werte von  $\beta_c(T)$  im Bereich  $32 \lesssim T \lesssim 64$  hilfreicher.

Die Äquivalenz der Bilder von Temperatur und inverser Temperatur mag bei der Bestimmung der kritischen Exponenten  $\nu$  und kritischen Temperaturen  $\beta_c$  in Abhängigkeit von  $T$  noch näherungsweise gültig sein. Bei der hier durchgeführten Untersuchung der relativen Verschiebung der kritischen Temperatur in Abhängigkeit der Systemdicke  $T$  gilt sie jedoch definitiv nicht mehr ohne Weiteres. In der Tat ergibt ein Fit der linken Seite von Gleichung (67) indem die  $\beta_c(T)$  zu  $T_c(T) = \frac{1}{\beta_c(T)}$  umgerechnet werden, einen Wert  $\tilde{\nu} = 0.647(14)$ , was viel eher dem theoretisch Erwarteten entspricht. Insbesondere ist dieses Ergebnis auch kompatibel zu dem kritischen Exponenten  $\nu = 0.657(9)$ , der im Grenzfall eines kubischen Gitters mit Seitenlänge 64 numerisch ermittelt wurde.

## 6. Zusammenfassung

Im Rückblick ist festzuhalten, dass das Ziel der Arbeit, die Untersuchung des dimensionellen Übergangs vom zwei- zum dreidimensionalen Ising-Modell, grundsätzlich erreicht wurde. Konkret wurden für ein  $64 \times 64 \times T$ -Gitter, bei dem eine Seitenlänge  $T$  im Bereich zwischen 1 und 64 variiert wurde, die pseudokritischen Temperaturen  $\beta_c$  und Exponenten der Korrelationslänge  $\nu$  in Abhängigkeit von der Seitenlänge  $T$  bestimmt.

Zur Bestimmung von  $\nu$  und  $\beta_c$  wurde das Potenzgesetz, das diese beiden Größen im thermodynamischen Limes verknüpft, ausgewertet. Es besagt, dass die Korrelationslänge in der Nähe des kritischen Punktes beim Ising-Modell proportional dem Abstand von der kritischen Temperatur hoch dem kritischen Exponenten ist. Mit Hilfe der Zeitscheibenkorrelationsfunktion wurden die Korrelationslänge sowie deren Ableitung in Abhängigkeit von der Temperatur auf numerischem Wege ermittelt.

Um die Zeitscheibenkorrelationsfunktion für verschiedene Gittergrößen und Temperaturen zu erhalten, wurde ein Programm entwickelt, das von der Monte-Carlo-Methode Gebrauch macht. Besonderes Augenmerk lag hier auf einer effizienten und korrekten Implementierung, beides sowohl aus algorithmischer als auch aus softwaretechnischer Sicht. Die Verwendung von Cluster-Algorithmen war dabei sehr zielführend.

Der Produktiveinsatz dieser Software auf dem universitätseigenen Hochleistungs-Rechnerverbund generierte umfangreiche Datensätze. Auf deren Basis konnten schließlich die gewünschten Größen bestimmt, und damit die Gültigkeit der theoretischen Vorhersagen für das untersuchte System überprüft werden.

Die Qualität der letztlich erhaltenen Ergebnisse ist im Grundsatz nicht schlecht. Der Übergang vom zwei- zum dreidimensionalen Grenzfall folgt dem erwarteten Verhalten, nämlich eines gleichmäßigen Crossover der kritischen Größen vom einen Grenzfall zum anderen. Dennoch scheinen sowohl  $\nu(T)$  als auch  $\beta_c(T)$  mit einem systematischen Fehler behaftet zu sein. Dessen genauer Ursprung konnte nicht abschließend geklärt werden. Als ein Verursacher wurde die numerische Bestimmung der Ableitung unter Vernachlässigung höherer Ordnungen identifiziert. Doch auch weitere Effekte, die letztlich alle auf der Endlichkeit des untersuchten Systems basieren, könnten Einfluss auf den systematischen Fehler haben. Eine endgültige Klärung können wohl nur weitere Untersuchungen bringen.

Dies gilt auch für die Überprüfung der theoretischen Vorhersagen für die relative Verschiebung des kritischen Punktes beim dimensionellen Crossover. Unglücklicherweise eignen sich die gewählten Werte der Gitterausdehnung  $T$  nicht wirklich dazu, belastbare Aussagen im betrachteten Bild der Beschreibung anhand der inversen Temperatur zu machen. Auch hier ist wohl noch weitere Arbeit nötig.

Abschließend möchte sich der Autor beim Betreuer der Arbeit, Herrn Prof. Dr. G. Münster recht herzlich bedanken. Dies gilt nicht nur für die Auswahl dieses hochinteressanten Themas, sondern auch für die umgehende Hilfestellung bei spontan auftretenden Problemen, was einen maßgeblichen Anteil zum Gelingen dieser Arbeit leistete.

### Anhang A. Fits zur Bestimmung von $\nu$ und $\beta_c$

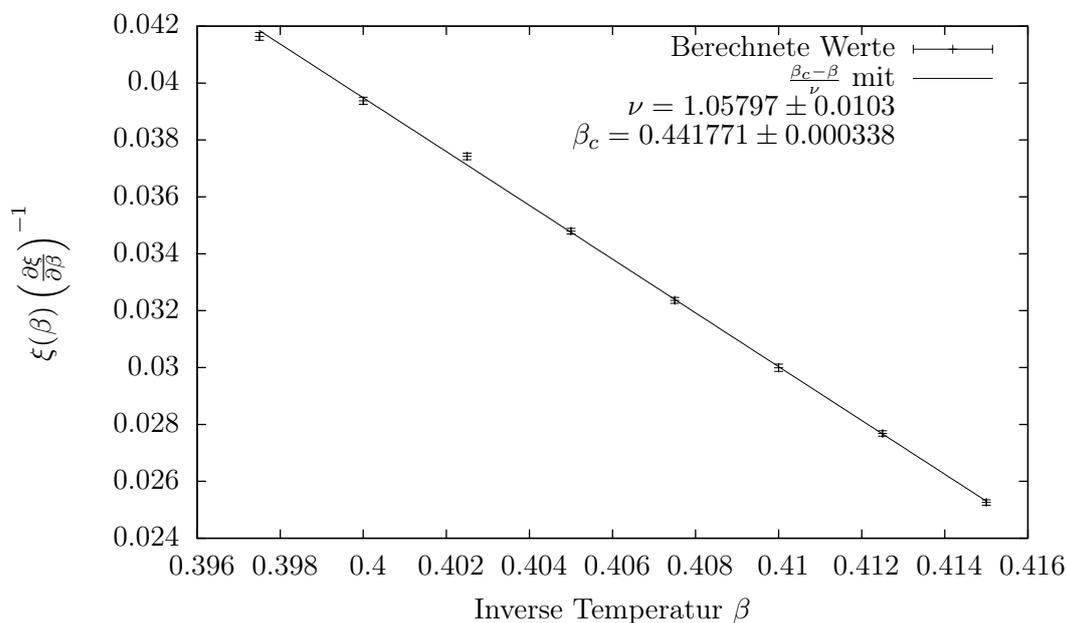


Abbildung 15: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 1$ .

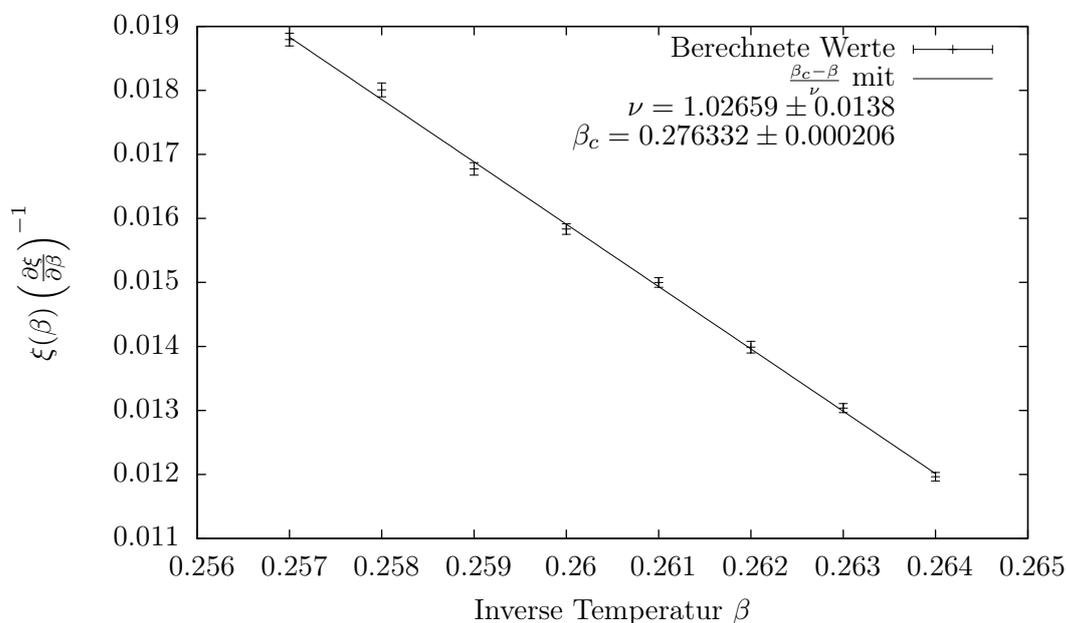


Abbildung 16: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 2$ .

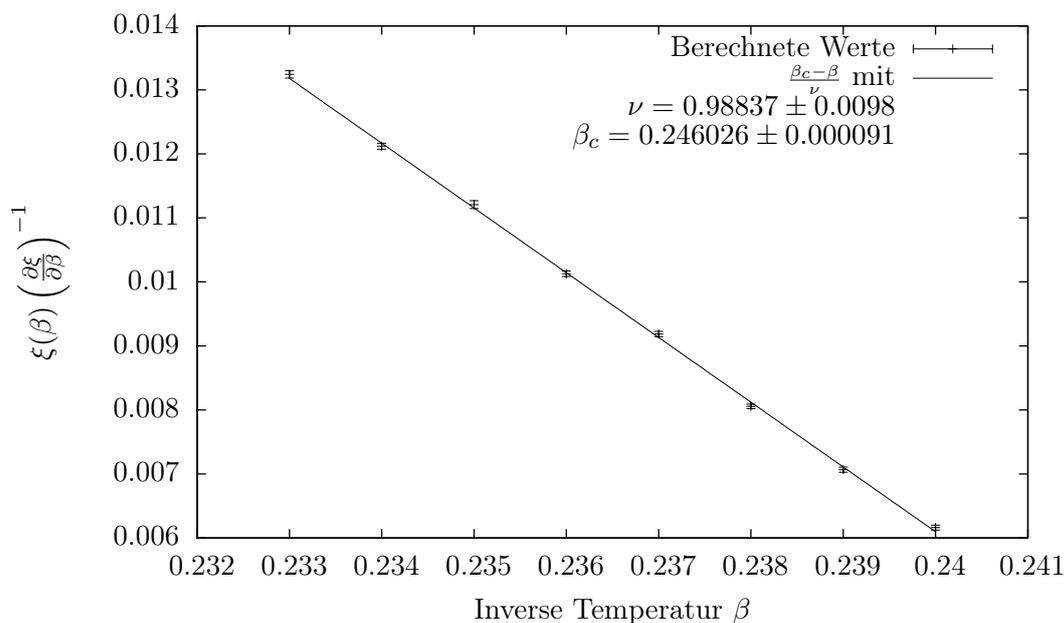


Abbildung 17: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 3$ .

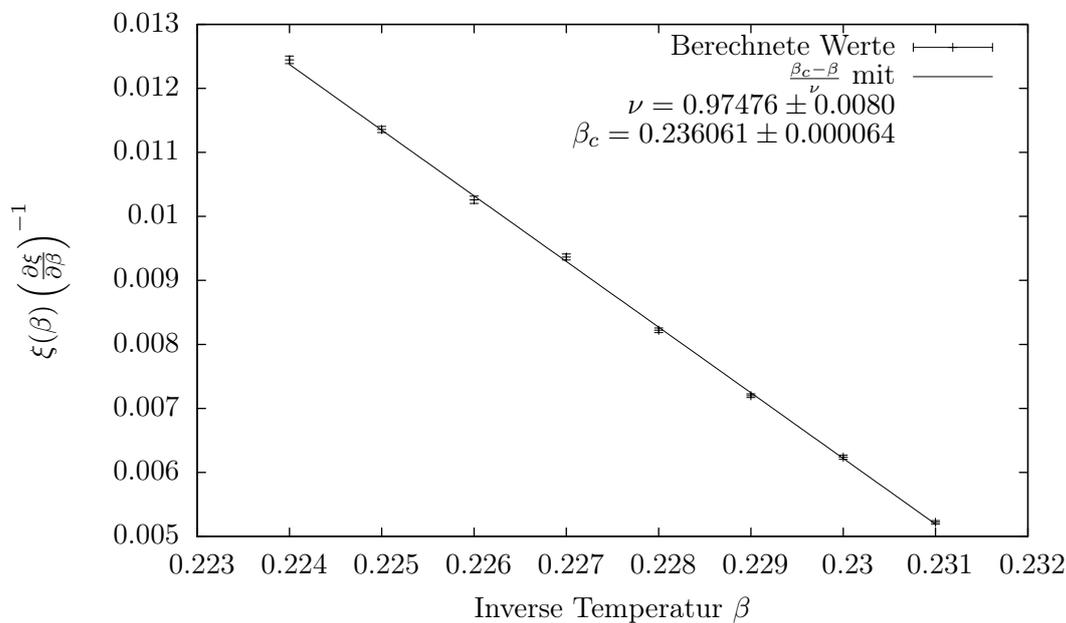


Abbildung 18: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 4$ .

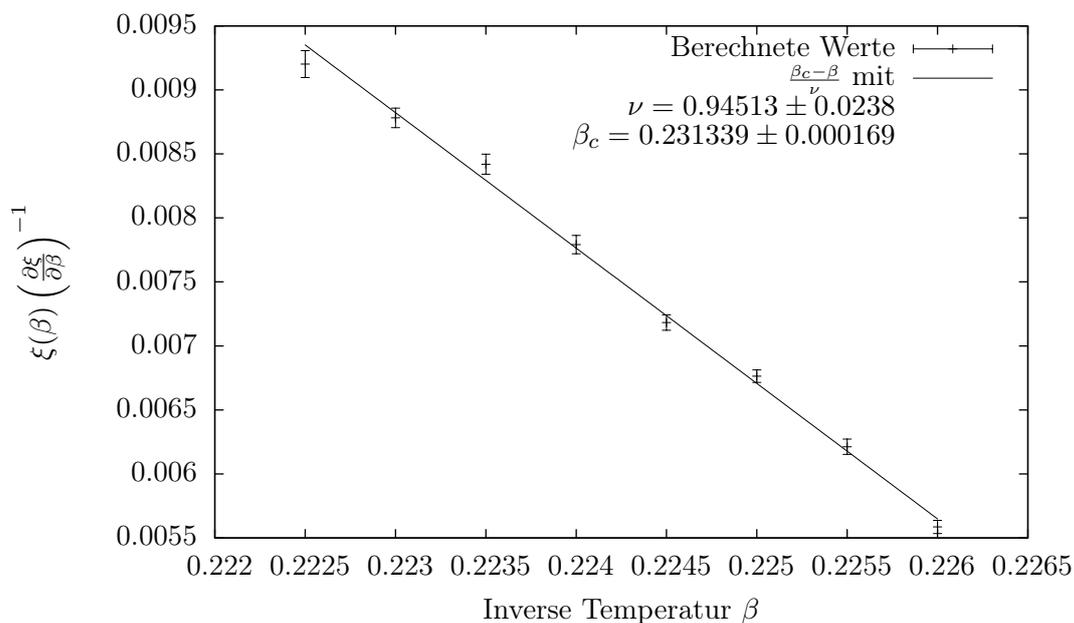


Abbildung 19: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 5$ .

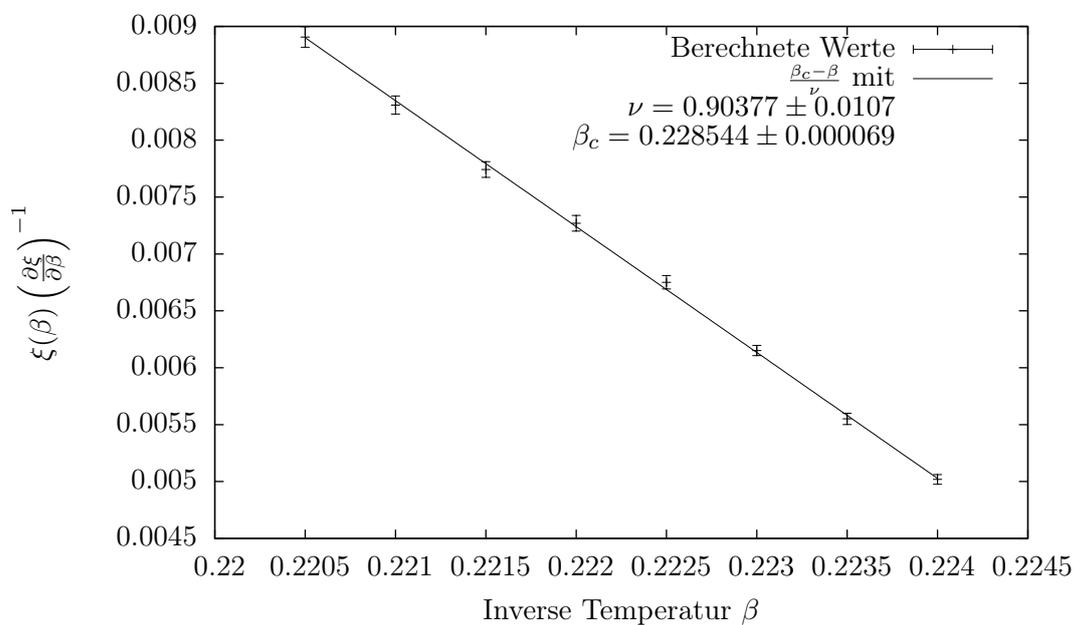


Abbildung 20: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 6$ .

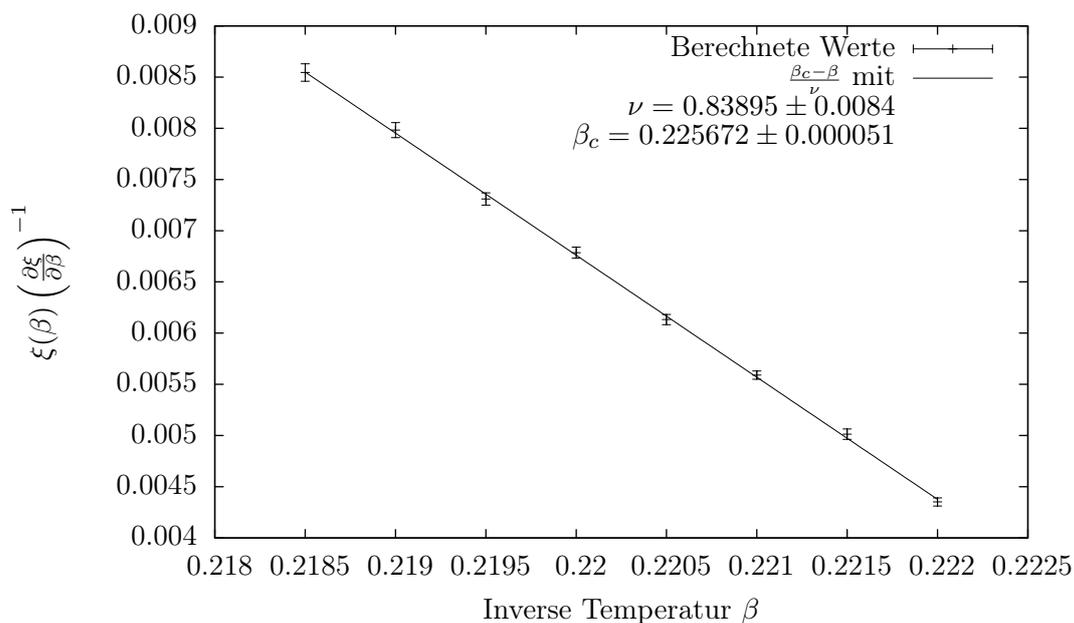


Abbildung 21: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 8$ .

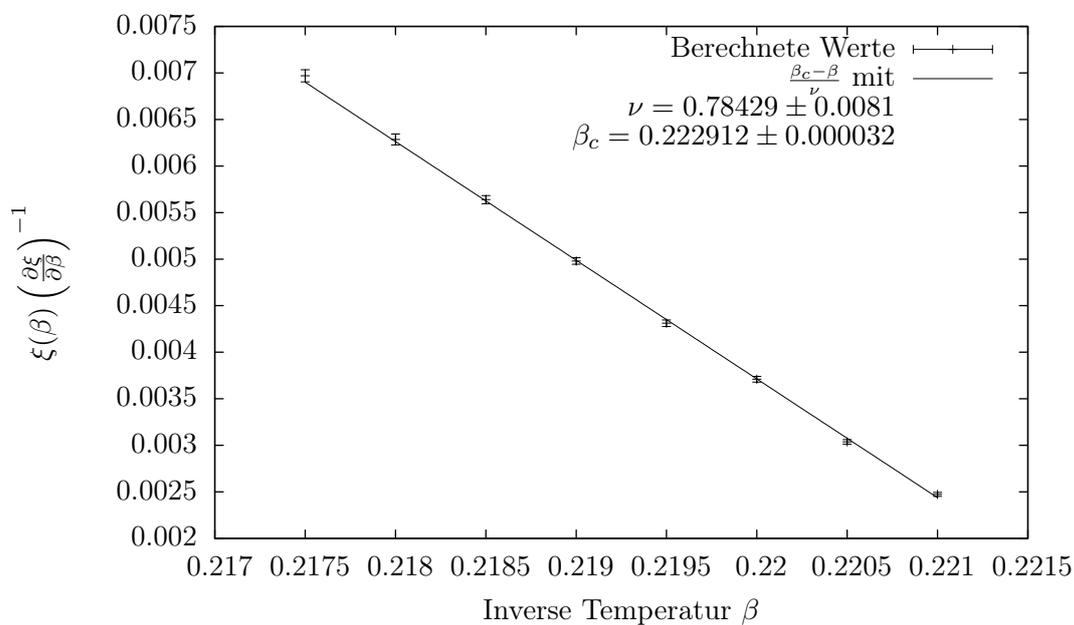


Abbildung 22: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 16$ .

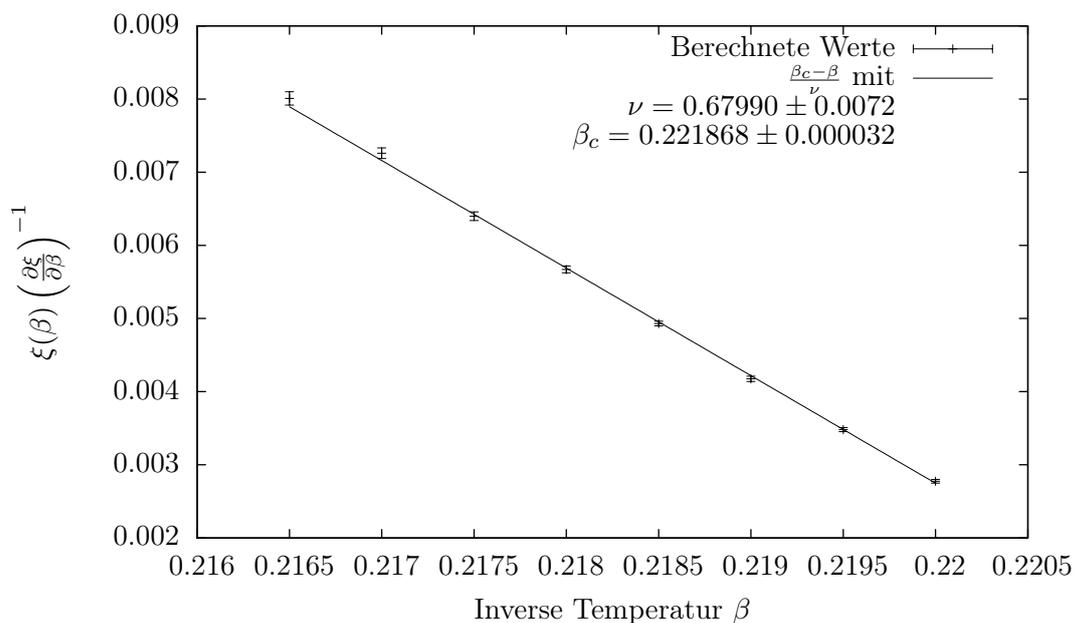


Abbildung 23: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 32$ .

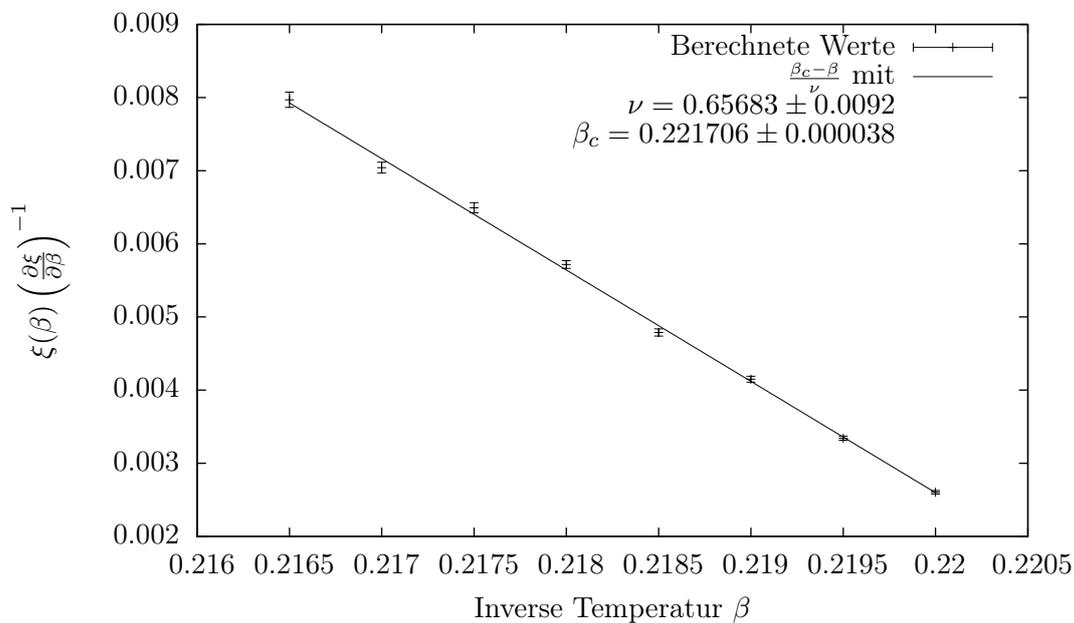


Abbildung 24: Fit der kritischen Temperatur  $\beta_c$  und des kritischen Exponenten  $\nu$  aus den Werten für  $\xi$  und  $\frac{\partial \xi}{\partial \beta}$  in Abhängigkeit von  $\beta$  bei  $T = 64$ .

## Anhang B. Gewonnene Daten in Tabellenform

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.39725	$5.57261 \pm 0.00014$	134217728	100000 Sweeps	$\approx 1$	31630 s
0.39750	$5.60582 \pm 0.00016$	134217728	100000 Sweeps	$\approx 1$	31951 s
0.39775	$5.63992 \pm 0.00017$	134217728	100000 Sweeps	$\approx 1$	32223 s
0.39975	$5.92451 \pm 0.00014$	134217728	100000 Sweeps	$\approx 1$	34764 s
0.40000	$5.96184 \pm 0.00021$	134217728	100000 Sweeps	$\approx 1$	35100 s
0.40025	$6.00021 \pm 0.00019$	134217728	100000 Sweeps	$\approx 1$	35448 s
0.40225	$6.32240 \pm 0.00016$	134217728	100000 Sweeps	$\approx 1$	38451 s
0.40250	$6.36450 \pm 0.00017$	134217728	100000 Sweeps	$\approx 1$	38856 s
0.40275	$6.40744 \pm 0.00021$	134217728	100000 Sweeps	$\approx 1$	39265 s
0.40475	$6.77451 \pm 0.00020$	134217728	100000 Sweeps	$\approx 1$	42862 s
0.40500	$6.82306 \pm 0.00021$	134217728	100000 Sweeps	$\approx 1$	43410 s
0.40525	$6.87256 \pm 0.00020$	134217728	100000 Sweeps	$\approx 1$	43846 s
0.40725	$7.29428 \pm 0.00023$	134217728	100000 Sweeps	$\approx 1$	48216 s
0.40750	$7.35052 \pm 0.00026$	134217728	100000 Sweeps	$\approx 1$	48800 s
0.40775	$7.40785 \pm 0.00027$	134217728	100000 Sweeps	$\approx 1$	49398 s
0.40975	$7.89783 \pm 0.00035$	67108864	100000 Sweeps	$\approx 1$	27392 s
0.41000	$7.96431 \pm 0.00034$	67108864	100000 Sweeps	$\approx 1$	27759 s
0.41025	$8.03060 \pm 0.00045$	67108864	100000 Sweeps	$\approx 1$	28147 s
0.41225	$8.60729 \pm 0.00039$	67108864	100000 Sweeps	$\approx 1$	31487 s
0.41250	$8.68529 \pm 0.00040$	67108864	100000 Sweeps	$\approx 1$	31953 s
0.41275	$8.76417 \pm 0.00039$	67108864	100000 Sweeps	$\approx 1$	32426 s
0.41475	$9.45066 \pm 0.00047$	67108864	100000 Sweeps	$\approx 1$	36684 s
0.41500	$9.54495 \pm 0.00048$	67108864	100000 Sweeps	$\approx 1$	37288 s
0.41525	$9.63962 \pm 0.00055$	67108864	100000 Sweeps	$\approx 1$	37907 s

Tabelle 2: Für das  $64 \times 64 \times 1$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.3975	$5.60582 \pm 0.00016$	$135 \pm 0.44$
0.4000	$5.96184 \pm 0.00021$	$151 \pm 0.48$
0.4025	$6.36450 \pm 0.00017$	$170 \pm 0.53$
0.4050	$6.82306 \pm 0.00021$	$196 \pm 0.56$
0.4075	$7.35052 \pm 0.00026$	$227 \pm 0.71$
0.4100	$7.96431 \pm 0.00034$	$266 \pm 1.1$
0.4125	$8.68529 \pm 0.00040$	$314 \pm 1.1$
0.4150	$9.54495 \pm 0.00048$	$378 \pm 1.5$

Tabelle 3: Für das  $64 \times 64 \times 1$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.25690	$5.78565 \pm 0.00023$	67108864	100000 Sweeps	$\approx 1$	34415 s
0.25700	$5.81665 \pm 0.00028$	67108864	100000 Sweeps	$\approx 1$	34640 s
0.25710	$5.84755 \pm 0.00023$	67108864	100000 Sweeps	$\approx 1$	34978 s
0.25790	$6.10958 \pm 0.00030$	67108864	100000 Sweeps	$\approx 1$	37596 s
0.25800	$6.14362 \pm 0.00029$	67108864	100000 Sweeps	$\approx 1$	37894 s
0.25810	$6.17781 \pm 0.00028$	67108864	100000 Sweeps	$\approx 1$	38385 s
0.25890	$6.47006 \pm 0.00036$	67108864	100000 Sweeps	$\approx 1$	41329 s
0.25900	$6.50914 \pm 0.00029$	67108864	100000 Sweeps	$\approx 1$	41712 s
0.25910	$6.54767 \pm 0.00028$	67108864	100000 Sweeps	$\approx 1$	42057 s
0.25990	$6.87634 \pm 0.00030$	67108864	100000 Sweeps	$\approx 1$	45693 s
0.26000	$6.91954 \pm 0.00028$	67108864	100000 Sweeps	$\approx 1$	46283 s
0.26010	$6.96374 \pm 0.00034$	67108864	100000 Sweeps	$\approx 1$	46578 s
0.26090	$7.33665 \pm 0.00033$	67108864	100000 Sweeps	$\approx 1$	50823 s
0.26100	$7.38506 \pm 0.00029$	67108864	100000 Sweeps	$\approx 1$	51440 s
0.26110	$7.43512 \pm 0.00037$	67108864	100000 Sweeps	$\approx 1$	52157 s
0.26190	$7.86004 \pm 0.00050$	33554432	100000 Sweeps	$\approx 1$	28618 s
0.26200	$7.91702 \pm 0.00048$	33554432	100000 Sweeps	$\approx 1$	28962 s
0.26210	$7.97323 \pm 0.00054$	33554432	100000 Sweeps	$\approx 1$	29259 s
0.26290	$8.46419 \pm 0.00049$	33554432	100000 Sweeps	$\approx 1$	32581 s
0.26300	$8.52938 \pm 0.00056$	33554432	100000 Sweeps	$\approx 1$	32837 s
0.26310	$8.59504 \pm 0.00052$	33554432	100000 Sweeps	$\approx 1$	33325 s
0.26390	$9.16526 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	37094 s
0.26400	$9.24183 \pm 0.00054$	33554432	100000 Sweeps	$\approx 1$	37559 s
0.26410	$9.31975 \pm 0.00063$	33554432	100000 Sweeps	$\approx 1$	38175 s

Tabelle 4: Für das  $64 \times 64 \times 2$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2570	$5.81665 \pm 0.00028$	$310 \pm 1.6$
0.2580	$6.14362 \pm 0.00029$	$341 \pm 2$
0.2590	$6.50914 \pm 0.00029$	$388 \pm 2.3$
0.2600	$6.91954 \pm 0.00028$	$437 \pm 2.3$
0.2610	$7.38506 \pm 0.00029$	$492 \pm 2.5$
0.2620	$7.91702 \pm 0.00048$	$566 \pm 3.7$
0.2630	$8.52938 \pm 0.00056$	$654 \pm 3.6$
0.2640	$9.24183 \pm 0.00054$	$772 \pm 4.3$

Tabelle 5: Für das  $64 \times 64 \times 2$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.23290	$5.95119 \pm 0.00027$	67108864	100000 Sweeps	$\approx 1$	45846 s
0.23300	$5.99554 \pm 0.00035$	67108864	100000 Sweeps	$\approx 1$	46474 s
0.23310	$6.04174 \pm 0.00029$	67108864	100000 Sweeps	$\approx 1$	47019 s
0.23390	$6.43691 \pm 0.00028$	67108864	100000 Sweeps	$\approx 1$	52467 s
0.23400	$6.49039 \pm 0.00035$	67108864	100000 Sweeps	$\approx 1$	53160 s
0.23410	$6.54401 \pm 0.00029$	67108864	100000 Sweeps	$\approx 1$	53991 s
0.23490	$7.01118 \pm 0.00043$	33554432	100000 Sweeps	$\approx 1$	30403 s
0.23500	$7.07397 \pm 0.00045$	33554432	100000 Sweeps	$\approx 1$	30823 s
0.23510	$7.13742 \pm 0.00054$	33554432	100000 Sweeps	$\approx 1$	31344 s
0.23590	$7.69675 \pm 0.00041$	33554432	100000 Sweeps	$\approx 1$	35618 s
0.23600	$7.77238 \pm 0.00045$	33554432	100000 Sweeps	$\approx 1$	36211 s
0.23610	$7.85024 \pm 0.00051$	33554432	100000 Sweeps	$\approx 1$	36928 s
0.23690	$8.53428 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	42609 s
0.23700	$8.62780 \pm 0.00080$	33554432	100000 Sweeps	$\approx 1$	43328 s
0.23710	$8.72213 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	44164 s
0.23790	$9.57223 \pm 0.00064$	33554432	100000 Sweeps	$\approx 1$	51860 s
0.23800	$9.69095 \pm 0.00076$	33554432	100000 Sweeps	$\approx 1$	53013 s
0.23810	$9.81282 \pm 0.00082$	33554432	100000 Sweeps	$\approx 1$	54135 s
0.23890	$10.8985 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	32501 s
0.23900	$11.0529 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	33315 s
0.23910	$11.2113 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	34216 s
0.23990	$12.6374 \pm 0.0019$	16777216	100000 Sweeps	$\approx 1$	41923 s
0.24000	$12.8407 \pm 0.0018$	16777216	100000 Sweeps	$\approx 1$	43098 s
0.24010	$13.0542 \pm 0.0018$	16777216	100000 Sweeps	$\approx 1$	44344 s

Tabelle 6: Für das  $64 \times 64 \times 3$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2330	$5.99554 \pm 0.00035$	$453 \pm 2$
0.2340	$6.49039 \pm 0.00035$	$536 \pm 2$
0.2350	$7.07397 \pm 0.00045$	$631 \pm 3.4$
0.2360	$7.77238 \pm 0.00045$	$767 \pm 3.3$
0.2370	$8.62780 \pm 0.00080$	$939 \pm 4.2$
0.2380	$9.69095 \pm 0.00076$	$1203 \pm 5.2$
0.2390	$11.0529 \pm 0.0013$	$1564 \pm 9.1$
0.2400	$12.8407 \pm 0.0018$	$2084 \pm 13$

Tabelle 7: Für das  $64 \times 64 \times 3$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.22390	$5.29330 \pm 0.00028$	67108864	100000 Sweeps	$\approx 1$	41477 s
0.22400	$5.33555 \pm 0.00026$	67108864	100000 Sweeps	$\approx 1$	42063 s
0.22410	$5.37904 \pm 0.00030$	67108864	100000 Sweeps	$\approx 1$	42649 s
0.22490	$5.75210 \pm 0.00030$	67108864	100000 Sweeps	$\approx 1$	47848 s
0.22500	$5.80299 \pm 0.00030$	67108864	100000 Sweeps	$\approx 1$	48593 s
0.22510	$5.85428 \pm 0.00030$	67108864	100000 Sweeps	$\approx 1$	49328 s
0.22590	$6.30012 \pm 0.00052$	33554432	100000 Sweeps	$\approx 1$	28086 s
0.22600	$6.36186 \pm 0.00044$	33554432	100000 Sweeps	$\approx 1$	28586 s
0.22610	$6.42414 \pm 0.00045$	33554432	100000 Sweeps	$\approx 1$	29053 s
0.22690	$6.96983 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	33464 s
0.22700	$7.04458 \pm 0.00050$	33554432	100000 Sweeps	$\approx 1$	34090 s
0.22710	$7.12027 \pm 0.00048$	33554432	100000 Sweeps	$\approx 1$	34722 s
0.22790	$7.80209 \pm 0.00060$	33554432	100000 Sweeps	$\approx 1$	40734 s
0.22800	$7.89639 \pm 0.00068$	33554432	100000 Sweeps	$\approx 1$	41602 s
0.22810	$7.99420 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	42490 s
0.22890	$8.86620 \pm 0.00073$	33554432	100000 Sweeps	$\approx 1$	50950 s
0.22900	$8.98931 \pm 0.00080$	33554432	100000 Sweeps	$\approx 1$	52216 s
0.22910	$9.11594 \pm 0.00066$	33554432	100000 Sweeps	$\approx 1$	53482 s
0.22990	$10.2743 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	33068 s
0.23000	$10.4392 \pm 0.0014$	16777216	100000 Sweeps	$\approx 1$	34061 s
0.23010	$10.6090 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	35014 s
0.23090	$12.2085 \pm 0.0017$	16777216	100000 Sweeps	$\approx 1$	44736 s
0.23100	$12.4409 \pm 0.0017$	16777216	100000 Sweeps	$\approx 1$	46231 s
0.23110	$12.6852 \pm 0.0019$	16777216	100000 Sweeps	$\approx 1$	47829 s

Tabelle 8: Für das  $64 \times 64 \times 4$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2240	$5.33555 \pm 0.00026$	$429 \pm 2.1$
0.2250	$5.80299 \pm 0.00030$	$511 \pm 2.1$
0.2260	$6.36186 \pm 0.00044$	$620 \pm 3.4$
0.2270	$7.04458 \pm 0.00050$	$752 \pm 3.8$
0.2280	$7.89639 \pm 0.00068$	$961 \pm 4.3$
0.2290	$8.98931 \pm 0.00080$	$1249 \pm 4.9$
0.2300	$10.4392 \pm 0.0014$	$1674 \pm 8.7$
0.2310	$12.4409 \pm 0.0017$	$2384 \pm 13$

Tabelle 9: Für das  $64 \times 64 \times 4$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.22245	$6.21849 \pm 0.00055$	33554432	100000 Sweeps	$\approx 1$	29141 s
0.22250	$6.25337 \pm 0.00041$	33554432	100000 Sweeps	$\approx 1$	29414 s
0.22255	$6.28645 \pm 0.00055$	33554432	100000 Sweeps	$\approx 1$	29703 s
0.22295	$6.56912 \pm 0.00046$	33554432	100000 Sweeps	$\approx 1$	32067 s
0.22300	$6.60716 \pm 0.00045$	33554432	100000 Sweeps	$\approx 1$	32403 s
0.22305	$6.64436 \pm 0.00047$	33554432	100000 Sweeps	$\approx 1$	32728 s
0.22345	$6.96244 \pm 0.00050$	33554432	100000 Sweeps	$\approx 1$	35531 s
0.22350	$7.00352 \pm 0.00055$	33554432	100000 Sweeps	$\approx 1$	35893 s
0.22355	$7.04562 \pm 0.00059$	33554432	100000 Sweeps	$\approx 1$	36306 s
0.22395	$7.40755 \pm 0.00070$	33554432	100000 Sweeps	$\approx 1$	39648 s
0.22400	$7.45551 \pm 0.00054$	33554432	100000 Sweeps	$\approx 1$	40098 s
0.22405	$7.50324 \pm 0.00053$	33554432	100000 Sweeps	$\approx 1$	40570 s
0.22445	$7.91667 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	44585 s
0.22450	$7.97148 \pm 0.00063$	33554432	100000 Sweeps	$\approx 1$	45147 s
0.22455	$8.02766 \pm 0.00069$	33554432	100000 Sweeps	$\approx 1$	45681 s
0.22495	$8.50478 \pm 0.00069$	33554432	100000 Sweeps	$\approx 1$	50636 s
0.22500	$8.56771 \pm 0.00068$	33554432	100000 Sweeps	$\approx 1$	51313 s
0.22505	$8.63145 \pm 0.00061$	33554432	100000 Sweeps	$\approx 1$	51997 s
0.22545	$9.18981 \pm 0.0010$	16777216	100000 Sweeps	$\approx 1$	29133 s
0.22550	$9.26309 \pm 0.00096$	16777216	100000 Sweeps	$\approx 1$	29642 s
0.22555	$9.33892 \pm 0.0010$	16777216	100000 Sweeps	$\approx 1$	30010 s
0.22595	$9.99657 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	33867 s
0.22600	$10.0841 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	34366 s
0.22605	$10.1771 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	34951 s

Tabelle 10: Für das  $64 \times 64 \times 5$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2225	$6.25337 \pm 0.00041$	$680 \pm 7.8$
0.2230	$6.60716 \pm 0.00045$	$752 \pm 6.6$
0.2235	$7.00352 \pm 0.00055$	$832 \pm 7.7$
0.2240	$7.45551 \pm 0.00054$	$957 \pm 8.8$
0.2245	$7.97148 \pm 0.00063$	$1110 \pm 9.3$
0.2250	$8.56771 \pm 0.00068$	$1267 \pm 9.2$
0.2255	$9.26309 \pm 0.00096$	$1491 \pm 14$
0.2260	$10.0841 \pm 0.0012$	$1805 \pm 17$

Tabelle 11: Für das  $64 \times 64 \times 5$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.22045	$6.02188 \pm 0.00043$	33554432	100000 Sweeps	$\approx 1$	28625 s
0.22050	$6.05578 \pm 0.00049$	33554432	100000 Sweeps	$\approx 1$	28932 s
0.22055	$6.08987 \pm 0.00055$	33554432	100000 Sweeps	$\approx 1$	29203 s
0.22095	$6.37646 \pm 0.00061$	33554432	100000 Sweeps	$\approx 1$	31693 s
0.22100	$6.41577 \pm 0.00053$	33554432	100000 Sweeps	$\approx 1$	32036 s
0.22105	$6.45368 \pm 0.00043$	33554432	100000 Sweeps	$\approx 1$	32368 s
0.22145	$6.78038 \pm 0.00048$	33554432	100000 Sweeps	$\approx 1$	35345 s
0.22150	$6.82511 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	35759 s
0.22155	$6.86854 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	36155 s
0.22195	$7.24344 \pm 0.00072$	33554432	100000 Sweeps	$\approx 1$	39778 s
0.22200	$7.29230 \pm 0.00063$	33554432	100000 Sweeps	$\approx 1$	40250 s
0.22205	$7.34375 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	40754 s
0.22245	$7.77759 \pm 0.00072$	33554432	100000 Sweeps	$\approx 1$	45156 s
0.22250	$7.83600 \pm 0.00077$	33554432	100000 Sweeps	$\approx 1$	45771 s
0.22255	$7.89366 \pm 0.00071$	33554432	100000 Sweeps	$\approx 1$	46374 s
0.22295	$8.40230 \pm 0.00068$	33554432	100000 Sweeps	$\approx 1$	51887 s
0.22300	$8.46984 \pm 0.00069$	33554432	100000 Sweeps	$\approx 1$	52648 s
0.22305	$8.53999 \pm 0.00076$	33554432	100000 Sweeps	$\approx 1$	53440 s
0.22345	$9.14093 \pm 0.0011$	16777216	100000 Sweeps	$\approx 1$	30299 s
0.22350	$9.22327 \pm 0.0010$	16777216	100000 Sweeps	$\approx 1$	30779 s
0.22355	$9.30710 \pm 0.0010$	16777216	100000 Sweeps	$\approx 1$	31289 s
0.22395	$10.0295 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	35810 s
0.22400	$10.1279 \pm 0.0011$	16777216	100000 Sweeps	$\approx 1$	36406 s
0.22405	$10.2313 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	37039 s

Tabelle 12: Für das  $64 \times 64 \times 6$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2205	$6.05578 \pm 0.00049$	$680 \pm 7$
0.2210	$6.41577 \pm 0.00053$	$772 \pm 7.5$
0.2215	$6.82511 \pm 0.00058$	$882 \pm 7.8$
0.2220	$7.29230 \pm 0.00063$	$1003 \pm 9.5$
0.2225	$7.83600 \pm 0.00077$	$1161 \pm 10$
0.2230	$8.46984 \pm 0.00069$	$1377 \pm 10$
0.2235	$9.22327 \pm 0.0010$	$1662 \pm 15$
0.2240	$10.1279 \pm 0.0011$	$2018 \pm 17$

Tabelle 13: Für das  $64 \times 64 \times 6$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.21845	$5.72701 \pm 0.00050$	33554432	100000 Sweeps	$\approx 1$	27506 s
0.21850	$5.76186 \pm 0.00053$	33554432	100000 Sweeps	$\approx 1$	27783 s
0.21855	$5.79444 \pm 0.00045$	33554432	100000 Sweeps	$\approx 1$	28095 s
0.21895	$6.08421 \pm 0.00049$	33554432	100000 Sweeps	$\approx 1$	30632 s
0.21900	$6.12156 \pm 0.00047$	33554432	100000 Sweeps	$\approx 1$	31008 s
0.21905	$6.16090 \pm 0.00051$	33554432	100000 Sweeps	$\approx 1$	31357 s
0.21945	$6.49290 \pm 0.00048$	33554432	100000 Sweeps	$\approx 1$	34482 s
0.21950	$6.53840 \pm 0.00064$	33554432	100000 Sweeps	$\approx 1$	34900 s
0.21955	$6.58234 \pm 0.00055$	33554432	100000 Sweeps	$\approx 1$	35344 s
0.21995	$6.96964 \pm 0.00063$	33554432	100000 Sweeps	$\approx 1$	39191 s
0.22000	$7.02110 \pm 0.00064$	33554432	100000 Sweeps	$\approx 1$	39709 s
0.22005	$7.07311 \pm 0.00052$	33554432	100000 Sweeps	$\approx 1$	40247 s
0.22045	$7.52849 \pm 0.00069$	33554432	100000 Sweeps	$\approx 1$	45063 s
0.22050	$7.58870 \pm 0.00067$	33554432	100000 Sweeps	$\approx 1$	45732 s
0.22055	$7.65223 \pm 0.00073$	33554432	100000 Sweeps	$\approx 1$	46404 s
0.22095	$8.19525 \pm 0.00075$	33554432	100000 Sweeps	$\approx 1$	52596 s
0.22100	$8.26917 \pm 0.00069$	33554432	100000 Sweeps	$\approx 1$	53481 s
0.22105	$8.34319 \pm 0.00077$	33554432	100000 Sweeps	$\approx 1$	54320 s
0.22145	$9.00324 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	31394 s
0.22150	$9.09410 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	31961 s
0.22155	$9.18464 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	32576 s
0.22195	$9.99872 \pm 0.0014$	16777216	100000 Sweeps	$\approx 1$	38005 s
0.22200	$10.1127 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	38812 s
0.22205	$10.2311 \pm 0.0016$	16777216	100000 Sweeps	$\approx 1$	39599 s

Tabelle 14: Für das  $64 \times 64 \times 8$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2185	$5.76186 \pm 0.00053$	$674 \pm 6.7$
0.2190	$6.12156 \pm 0.00047$	$767 \pm 7.1$
0.2195	$6.53840 \pm 0.00064$	$894 \pm 7.3$
0.2200	$7.02110 \pm 0.00064$	$1035 \pm 8.2$
0.2205	$7.58870 \pm 0.00067$	$1237 \pm 10$
0.2210	$8.26917 \pm 0.00069$	$1479 \pm 11$
0.2215	$9.09410 \pm 0.0013$	$1814 \pm 18$
0.2220	$10.1127 \pm 0.0013$	$2324 \pm 21$

Tabelle 15: Für das  $64 \times 64 \times 8$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.21745	$5.88858 \pm 0.00054$	33554432	100000 Sweeps	$\approx 1$	31894 s
0.21750	$5.93045 \pm 0.00055$	33554432	100000 Sweeps	$\approx 1$	32353 s
0.21755	$5.97367 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	32693 s
0.21795	$6.34314 \pm 0.00064$	33554432	100000 Sweeps	$\approx 1$	36748 s
0.21800	$6.39466 \pm 0.00068$	33554432	100000 Sweeps	$\approx 1$	36960 s
0.21805	$6.44488 \pm 0.00070$	33554432	100000 Sweeps	$\approx 1$	37655 s
0.21845	$6.89056 \pm 0.00069$	33554432	100000 Sweeps	$\approx 1$	42198 s
0.21850	$6.95342 \pm 0.00067$	33554432	100000 Sweeps	$\approx 1$	42929 s
0.21855	$7.01387 \pm 0.00067$	33554432	100000 Sweeps	$\approx 1$	43736 s
0.21895	$7.56753 \pm 0.00072$	33554432	100000 Sweeps	$\approx 1$	50447 s
0.21900	$7.64442 \pm 0.00072$	33554432	100000 Sweeps	$\approx 1$	51381 s
0.21905	$7.72104 \pm 0.00080$	33554432	100000 Sweeps	$\approx 1$	52320 s
0.21945	$8.41894 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	31418 s
0.21950	$8.51793 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	32008 s
0.21955	$8.61649 \pm 0.0011$	16777216	100000 Sweeps	$\approx 1$	32838 s
0.21995	$9.53430 \pm 0.0015$	16777216	100000 Sweeps	$\approx 1$	39507 s
0.22000	$9.66386 \pm 0.0016$	16777216	100000 Sweeps	$\approx 1$	40582 s
0.22005	$9.79480 \pm 0.0016$	16777216	100000 Sweeps	$\approx 1$	41460 s
0.22045	$11.0481 \pm 0.0022$	16777216	100000 Sweeps	$\approx 1$	51650 s
0.22050	$11.2321 \pm 0.0020$	16777216	100000 Sweeps	$\approx 1$	53214 s
0.22055	$11.4179 \pm 0.0021$	16777216	100000 Sweeps	$\approx 1$	54929 s
0.22095	$13.2284 \pm 0.0031$	8388608	100000 Sweeps	$\approx 1$	36890 s
0.22100	$13.4981 \pm 0.0032$	8388608	100000 Sweeps	$\approx 1$	38301 s
0.22105	$13.7743 \pm 0.0037$	8388608	100000 Sweeps	$\approx 1$	39725 s

Tabelle 16: Für das  $64 \times 64 \times 16$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2175	$5.93045 \pm 0.00055$	$851 \pm 7.9$
0.2180	$6.39466 \pm 0.00068$	$1017 \pm 9.5$
0.2185	$6.95342 \pm 0.00067$	$1233 \pm 9.6$
0.2190	$7.64442 \pm 0.00072$	$1535 \pm 11$
0.2195	$8.51793 \pm 0.0012$	$1976 \pm 16$
0.2200	$9.66386 \pm 0.0016$	$2605 \pm 22$
0.2205	$11.2321 \pm 0.0020$	$3698 \pm 30$
0.2210	$13.4981 \pm 0.0032$	$5459 \pm 48$

Tabelle 17: Für das  $64 \times 64 \times 16$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.21645	$5.24209 \pm 0.00048$	33554432	100000 Sweeps	$\approx 1$	30153 s
0.21650	$5.27473 \pm 0.00060$	33554432	100000 Sweeps	$\approx 1$	31455 s
0.21655	$5.30795 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	30588 s
0.21695	$5.59581 \pm 0.00050$	33554432	100000 Sweeps	$\approx 1$	34263 s
0.21700	$5.63241 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	34386 s
0.21705	$5.67339 \pm 0.00060$	33554432	100000 Sweeps	$\approx 1$	35339 s
0.21745	$6.01421 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	38580 s
0.21750	$6.05983 \pm 0.00052$	33554432	100000 Sweeps	$\approx 1$	39514 s
0.21755	$6.10892 \pm 0.00063$	33554432	100000 Sweeps	$\approx 1$	39013 s
0.21795	$6.52211 \pm 0.00061$	33554432	100000 Sweeps	$\approx 1$	43335 s
0.21800	$6.57967 \pm 0.00066$	33554432	100000 Sweeps	$\approx 1$	43603 s
0.21805	$6.63815 \pm 0.00073$	33554432	100000 Sweeps	$\approx 1$	44445 s
0.21845	$7.15235 \pm 0.00074$	33554432	100000 Sweeps	$\approx 1$	52208 s
0.21850	$7.22454 \pm 0.00096$	33554432	100000 Sweeps	$\approx 1$	51961 s
0.21855	$7.29888 \pm 0.00061$	33554432	100000 Sweeps	$\approx 1$	53164 s
0.21895	$7.96008 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	33323 s
0.21900	$8.05797 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	34067 s
0.21905	$8.15307 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	34897 s
0.21945	$9.04275 \pm 0.0015$	16777216	100000 Sweeps	$\approx 1$	41893 s
0.21950	$9.17497 \pm 0.0014$	16777216	100000 Sweeps	$\approx 1$	42329 s
0.21955	$9.30631 \pm 0.0015$	16777216	100000 Sweeps	$\approx 1$	43817 s
0.21995	$10.5802 \pm 0.0020$	8388608	100000 Sweeps	$\approx 1$	28812 s
0.22000	$10.7657 \pm 0.0022$	8388608	100000 Sweeps	$\approx 1$	30448 s
0.22005	$10.9682 \pm 0.0030$	8388608	100000 Sweeps	$\approx 1$	31813 s

Tabelle 18: Für das  $64 \times 64 \times 32$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2165	$5.27473 \pm 0.00060$	$659 \pm 7.5$
0.2170	$5.63241 \pm 0.00058$	$776 \pm 7.8$
0.2175	$6.05983 \pm 0.00052$	$947 \pm 8.8$
0.2180	$6.57967 \pm 0.00066$	$1160 \pm 9.5$
0.2185	$7.22454 \pm 0.00096$	$1465 \pm 9.5$
0.2190	$8.05797 \pm 0.0012$	$1930 \pm 17$
0.2195	$9.17497 \pm 0.0014$	$2636 \pm 21$
0.2200	$10.7657 \pm 0.0022$	$3880 \pm 36$

Tabelle 19: Für das  $64 \times 64 \times 32$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

$\beta$	$\xi$	# Samples	Equilibrierung	$\tau$	CPU-Zeit
0.21645	$5.24342 \pm 0.00058$	33554432	100000 Sweeps	$\approx 1$	34835 s
0.21650	$5.27647 \pm 0.00049$	33554432	100000 Sweeps	$\approx 1$	35899 s
0.21655	$5.30963 \pm 0.00064$	33554432	100000 Sweeps	$\approx 1$	35858 s
0.21695	$5.59660 \pm 0.00057$	33554432	100000 Sweeps	$\approx 1$	38243 s
0.21700	$5.63594 \pm 0.00066$	33554432	100000 Sweeps	$\approx 1$	40149 s
0.21705	$5.67664 \pm 0.00062$	33554432	100000 Sweeps	$\approx 1$	39896 s
0.21745	$6.01946 \pm 0.00073$	33554432	100000 Sweeps	$\approx 1$	42582 s
0.21750	$6.06542 \pm 0.00056$	33554432	100000 Sweeps	$\approx 1$	42848 s
0.21755	$6.11288 \pm 0.00070$	33554432	100000 Sweeps	$\approx 1$	44743 s
0.21795	$6.53088 \pm 0.00077$	33554432	100000 Sweeps	$\approx 1$	48843 s
0.21800	$6.58817 \pm 0.00067$	33554432	100000 Sweeps	$\approx 1$	48998 s
0.21805	$6.64611 \pm 0.00072$	33554432	100000 Sweeps	$\approx 1$	49655 s
0.21845	$7.16666 \pm 0.0012$	16777216	100000 Sweeps	$\approx 1$	31431 s
0.21850	$7.24008 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	31581 s
0.21855	$7.31787 \pm 0.00096$	16777216	100000 Sweeps	$\approx 1$	33013 s
0.21895	$7.99287 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	37919 s
0.21900	$8.08796 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	38721 s
0.21905	$8.18776 \pm 0.0013$	16777216	100000 Sweeps	$\approx 1$	39747 s
0.21945	$9.10880 \pm 0.0016$	16777216	100000 Sweeps	$\approx 1$	46390 s
0.21950	$9.24189 \pm 0.0015$	16777216	100000 Sweeps	$\approx 1$	48387 s
0.21955	$9.38530 \pm 0.0017$	16777216	100000 Sweeps	$\approx 1$	48825 s
0.21995	$10.7259 \pm 0.0028$	8388608	100000 Sweeps	$\approx 1$	34468 s
0.22000	$10.9282 \pm 0.0027$	8388608	100000 Sweeps	$\approx 1$	35672 s
0.22005	$11.1456 \pm 0.0028$	8388608	100000 Sweeps	$\approx 1$	36480 s

Tabelle 20: Für das  $64 \times 64 \times 64$ -System numerisch bestimmte Korrelationslängen  $\xi$  und weitere, dabei relevante Parameter.

$\beta$	$\xi$	$\frac{\partial \xi}{\partial \beta}$
0.2165	$5.27647 \pm 0.00049$	$662 \pm 8.7$
0.2170	$5.63594 \pm 0.00066$	$800 \pm 8.4$
0.2175	$6.06542 \pm 0.00056$	$934 \pm 10$
0.2180	$6.58817 \pm 0.00067$	$1152 \pm 11$
0.2185	$7.24008 \pm 0.0013$	$1512 \pm 16$
0.2190	$8.08796 \pm 0.0013$	$1949 \pm 19$
0.2195	$9.24189 \pm 0.0015$	$2765 \pm 23$
0.2200	$10.9282 \pm 0.0027$	$4197 \pm 40$

Tabelle 21: Für das  $64 \times 64 \times 64$ -System berechnete Korrelationslängen und deren erste Ableitungen in Abhängigkeit von der Temperatur  $\beta$ .

## Literatur

- [1] SOKAL, A. D.: *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms*. Lectures at the Cargese Summer School on Functional Integration: Basics and Application, 1996. – <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.4444>
- [2] BAAKE, E. ; BAAKE, M. ; WAGNER, H.: Ising Quantum Chain is Equivalent to a Model of Biological Evolution. In: *Phys. Rev. Lett.* 78 (1997), S. 559–562
- [3] MÜNSTER, G.: *Quantentheorie*. 2. Auflage. De Gruyter, 2010
- [4] ONSAGER, L.: Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition. In: *Phys. Rev.* 65 (1944), S. 117–149
- [5] CIPRA, B. A.: The Ising Model is NP-Complete. In: *SIAM News* 33 (2000), Nr. 6
- [6] ISING, E.: Beitrag zur Theorie des Ferromagnetismus. In: *Zeitschrift für Physik* 31 (1925), Nr. 1, S. 253–258
- [7] [https://en.wikipedia.org/w/index.php?title=File:Torus\\_cycles.png&oldid=518461935](https://en.wikipedia.org/w/index.php?title=File:Torus_cycles.png&oldid=518461935), Abgerufen am 20.11.2013
- [8] BAXTER, R. J.: *Exactly Solved Models in Statistical Mechanics*. London : Academic Press, 1982
- [9] KRAMERS, H. A. ; WANNIER, G. H.: Statistics of the Two-Dimensional Ferromagnet. Part I. In: *Phys. Rev.* 60 (1941), S. 252–262
- [10] <http://courses.physics.illinois.edu/phys466/2DIsing/Ising.htm>, Abgerufen am 20.11.2013
- [11] NOLTING, W.: *Grundkurs Theoretische Physik*. Bd. 6: *Statistische Mechanik*. 7. Auflage. Berlin Heidelberg : Springer Spektrum, 2002
- [12] GRIFFITHS, R. B.: Dependence of Critical Indices on a Parameter. In: *Phys. Rev. Lett.* 24 (1970), S. 1479–1482
- [13] CHANDLER, D.: *Introduction to Modern Statistical Mechanics*. Oxford, New York : Oxford University Press, 1987
- [14] BARBER, M. N.: Finite-size Scaling. In: DOMB, C. (Hrsg.) ; LEBOWITZ, J. L. (Hrsg.): *Phase Transitions and Critical Phenomena* Bd. 8. Academic Press, 1983, S. 145–266
- [15] HOURAHINE, B.: *Private Mitschrift der Vorlesung PH456 - Topics in Computational and Complex Systems in Physics an der University of Strathclyde Glasgow*. 2013
- [16] KLESSINGER, S.: *Numerische Untersuchung der Oberflächenspannung im dreidimensionalen Ising-Modell*. Diplomarbeit am Institut für Theoretische Physik, Westfälische Wilhelms-Universität Münster, 1992
- [17] METROPOLIS, N. ; ROSENBLUTH, A. W. ; ROSENBLUTH, M. N. ; TELLER, A. H. ; TELLER, E.: Equation of State Calculations by Fast Computing Machines. In: *The Journal of Chemical Physics* 21 (1953), Nr. 6, S. 1087–1092

- [18] WOOD, W. ; ERPENBECK, J.: Molecular Dynamics and Monte Carlo Calculations in Statistical Mechanics. In: *Annual Review of Physical Chemistry* 27 (1976), Nr. 1, S. 319–348
- [19] GATTRINGER, C. ; LANG, C. B.: *Quantum Chromodynamics on the Lattice: An Introductory Presentation*. Springer, 2010 (Lecture Notes in Physics)
- [20] WOLFF, U.: Critical slowing down. In: *Nuclear Physics B - Proceedings Supplements* 17 (1990), Nr. 0, S. 93 – 102
- [21] FRIEDRICH, H. ; PIETSCHMANN, F.: *Numerische Methoden: Ein Lehr- und Übungsbuch*. De Gruyter, 2010 (De Gruyter Studium)
- [22] MONTVAY, I. ; MÜNSTER, G.: *Quantum Fields on a Lattice*. Cambridge University Press, 1997 (Cambridge Monographs on Mathematical Physics)
- [23] Mündliche Mitteilung vom 20. 11. 2013
- [24] KRAUTH, W.: *Statistical Mechanics: Algorithms and Computations*. Oxford University Press, UK, 2006 (Oxford Master Series in Physics)
- [25] DENG, Y. ; GARONI, T. M. ; SOKAL, A. D.: Dynamic Critical Behavior of the Worm Algorithm for the Ising Model. In: *Phys. Rev. Lett.* 99 (2007), S. 110601
- [26] SWENDSEN, R. H. ; WANG, J.-S.: Nonuniversal Critical Dynamics in Monte Carlo Simulations. In: *Phys. Rev. Lett.* 58 (1987), S. 86–88
- [27] WOLFF, U.: Collective Monte Carlo Updating for Spin Systems. In: *Phys. Rev. Lett.* 62 (1989), S. 361–364
- [28] MURTHY, K.: *Monte Carlo Methods in Statistical Physics*. Universities Press, Hyderabad, 2004
- [29] HOSHEN, J. ; KOPELMAN, R.: Percolation and Cluster Distribution. I. Cluster Multiple Labeling Technique and Critical Concentration Algorithm. In: *Phys. Rev. B* 14 (1976), S. 3438–3445
- [30] WOLFF, U.: Asymptotic Freedom and Mass Generation in the O(3) Nonlinear  $\sigma$ -model. In: *Nuclear Physics B* 334 (1990), Nr. 3, S. 581 – 610
- [31] FLYVBJERG, H. ; PETERSEN, H. G.: Error Estimates on Averages of Correlated Data. In: *The Journal of Chemical Physics* 91 (1989), Nr. 1, S. 461–466
- [32] HAWICK, K. A. ; LEIST, A. ; PLAYNE, D. P.: Cluster and Fast Update Lattice Simulations Using Graphical Processing Units / Computer Science, Massey University. 2010 (CSTN-104). – Forschungsbericht
- [33] SAITO, M.: *An Application of Finite Field: Design and Implementation of 128-bit Instruction-Based Fast Pseudorandom Number Generator*, Hiroshima University, Diplomarbeit, 2007
- [34] MATSUMOTO, M. ; NISHIMURA, T.: Mersenne Twister: A 623-dimensionally Equi-distributed Uniform Pseudo-random Number Generator. In: *ACM Trans. Model. Comput. Simul.* 8 (1998), Nr. 1, S. 3–30

- [35] PANNETON, F. ; L'ECUYER, P. ; MATSUMOTO, M.: Improved Long-period Generators Based on Linear Recurrences Modulo 2. In: *ACM Trans. Math. Softw.* 32 (2006), Nr. 1, S. 1–16
- [36] JONES, D. ; UNIT, UCL B.: *Good Practice in (Pseudo) Random Number Generation for Bioinformatics Applications*. 2010. – <http://www.cs.ucl.ac.uk/staff/d.jones/GoodPracticeRNG.pdf>
- [37] CHAN, T. F. ; GOLUB, G. H. ; LEVEQUE, R. J.: Algorithms for Computing the Sample Variance: Analysis and Recommendations. In: *The American Statistician* 37 (1983), Nr. 3, S. 242–247
- [38] KNUTH, D. E.: *The Art of Computer Programming*. Bd. 2: *Seminumerical Algorithms*. 3. Auflage. Addison-Wesley Longman Publishing Co., Inc., 1997
- [39] WELFORD, B. P.: Note on a Method for Calculating Corrected Sums of Squares and Products. In: *Technometrics* 4 (1962), Nr. 3, S. 419–420
- [40] L'ECUYER, P. ; SIMARD, R.: TestU01: A C Library for Empirical Testing of Random Number Generators. In: *ACM Trans. Math. Softw.* 33 (2007), Nr. 4
- [41] FERRENBURG, A. M. ; LANDAU, D. P. ; WONG, Y. J.: Monte Carlo Simulations: Hidden Errors from “good” Random Number Generators. In: *Phys. Rev. Lett.* 69 (1992), S. 3382–3384
- [42] PAWLEY, G. S. ; SWENDSEN, R. H. ; WALLACE, D. J. ; WILSON, K. G.: Monte Carlo Renormalization-Group Calculations of Critical Behavior in the Simple-Cubic Ising Model. In: *Phys. Rev. B* 29 (1984), S. 4030–4040
- [43] ROUAULT, Y. ; BASCHNAGEL, J. ; BINDER, K.: Phase Separation of Symmetrical Polymer Mixtures in Thin-Film Geometry. In: *Journal of Statistical Physics* 80 (1995), Nr. 5-6, S. 1009–1031
- [44] CAPEHART, T. W. ; FISHER, M. E.: Susceptibility Scaling Functions for Ferromagnetic Ising Films. In: *Phys. Rev. B* 13 (1976), S. 5021–5038
- [45] CASELLE, M. ; HASENBUSCH, M.: Deconfinement Transition and Dimensional Crossover in the 3D Gauge Ising Model. In: *Nuclear Physics B* 470 (1996), Nr. 3, S. 435 – 453

## Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel

### **Dimensionelles Crossover im Ising-Modell**

selbständig verfasst habe, und dass ich keine anderen Quellen und Hilfsmittel als die angegebenen benutzt habe und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

---

Datum, Ort

Leopold Kellers