Numerische Tests von Vermutungen der konformen Feldtheorie für dreidimensionale Systeme

Diplomarbeit

zur Erlangung des Grades eines

Diplom-Physikers

vorgelegt von

Martin Weigel

Institut für Physik,
Johannes Gutenberg–Universität Mainz,
Staudinger Weg 7, D-55099 Mainz

Mainz, am 27. Februar 1998

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die Theorie der Phasenübergänge spielt eine zentrale Rolle in der Erforschung der Eigenschaften kondensierter Materie. Kritische Phasengrenzen kennzeichnen Singularitäten im Verhalten thermodynamischer Systeme beim Übergang zwischen Regionen der Stabilität mit essentiell verschiedenen makroskopischen Eigenschaften. Die Identität der mikroskopischen Basis der Systeme bietet den Ausgangspunkt zum Verständnis der Transformation der makroskopischen Beschreibungen beider Phasen ineinander. Am Punkte einer solchen Diskontinuität divergieren charakteristische Größen des Systems; anhand der Fundamentalität der divergierenden Größen unterscheidet man verschiedene Ordnungen von Phasenübergängen. Im Falle eines kontinuierlichen Phasenüberganges, oder Phasenübergangs zweiter Ordnung, entstehen aus der lokalen Wechselwirkung der Elemente des Systems makroskopische Ordnungsstrukturen, die Größen der Bereiche kohärenten Verhaltens der mikroskopischen Freiheitsgrade, die Korrelationslängen, divergieren. Damit gehen die Längenskalen des Systems verloren, das System wird selbstähnlich; es tritt hier spontan eine neue Symmetrieeigenschaft auf, die sich zur Analyse solcher Systeme nutzbar machen läßt.

Die Skaleninvarianz am Punkte eines Phasenübergangs zweiter Ordnung führt in der Theorie des (*Finite Size*) *Scaling*, wie sie aus einem Konzept der *Renormierung* folgt, zu einer quantitativen Beschreibung des Überganges einerseits zwischen den aneinander angrenzenden Phasen, andererseits zwischen dem kontinuierlichen Verhalten endlicher Systeme und der singulären Transformation ihrer Eigenschaften im thermodynamischen Limes. Der Verlust der Längenskalen führt dabei zu einer potenzartigen Form der Divergenz der charakteristischen Größen, deren Exponenten durch verschiedene universelle Relationen miteinander verknüpft sind und sich für einige zweidimensionale Modelle sogar exakt bestimmen lassen.

Die *Konforme Feldtheorie* erweitert das Konzept der Renormierung durch die Hinzunahme weiterer Symmetrietransformationen: die Skaleninvarianz des Systems wird ergänzt durch Homogenität und Isotropie, also Invarianz unter Translationen und Rotationen. Durch Ausnutzung dieser zusätzlichen Symmetrien gelingt es ihr, für zweidimensionale Systeme nicht nur die Exponenten der Skalengesetze exakt zu bestimmen, sondern zusätzlich sogar die zugehörigen Amplituden im Divergenzverhalten charakteristischer Größen. Diese Ergebnisse sind jedoch nur Nebenprodukte einer vollständigen algebraischen Klassifikation der Observablen von Systemen am Punkte eines Phasenübergangs zweiter Ordnung nach ihrem Skalenverhalten, die die konforme Feldtheorie liefert: ein solches Resultat kommt der analytischen Lösung des Modells — wie etwa des zweidimensionalen Ising–Modells mit Hilfe der Methode der Transfermatrix — sehr nahe.

Für Systeme in drei Raumdimensionen haben dagegen bisher, abgesehen von wenigen Grenzfällen, weder direkte Ansätze einer analytischen Behandlung wie die Transfermatrix–Rechnung noch auch feldtheoretische Methoden zu exakten Ergebnissen für die fundamentalen Größen geführt, was angesichts der größeren physikalischen Relevanz solcher Systeme gegenüber den zweidimensionalen Analoga besonders bedauerlich ist. Eine numerische Beobachtung im Zusammenhang mit dem Verhalten der Korrelationslängen von Modellen wechselwirkender Spin–Variablen auf bestimmten dreidimensionalen Geometrien durch M. Henkel [45, 46] nährt nun die Hoffnung, die Methoden der konformen Feldtheorie möglicherweise auch erfolgreich auf dreidimensionale Systeme anwenden zu können: die Theorie der konformen Invarianz hatte gezeigt, daß auf bestimmten zylindrischen Geometrien in *zwei* Dimensionen die Amplituden des Finite Size Scaling der Korrelationslängen in einer universellen Weise mit den zugehörigen Exponenten zusammenhängen; die numerische Beobachtung Henkels legt nun den Schluß nahe, daß eine solche Beziehung zwischen Amplituden und Exponenten mit einer kleinen Modifikation auch für analoge Systeme in *drei* Dimensionen Gültigkeit haben könnte. Sollte sich diese Vermutung bestätigen, so könnte man hoffen, mit den Ergebnissen auch die Methoden der konformen Feldtheorie auf dreidimensionale Systeme übertragen zu können und damit dem Ziel einer analytischen Lösung solcher Modelle entscheidend näher zu kommen.

Auf der Seite der physikalischen Methodik haben sich *Computer–Simulationen* in den letzten Jahrzehnten als gleichberechtigte dritte Säule neben dem Experiment und der analytischen Theorie mit Papier und Bleistift in der Physik etabliert. Die Möglichkeiten, die vereinfachten Modelle der analytischen Richtung direkt zu untersuchen oder Bedingungen zu realisieren, die sich im Experiment nur schwer herstellen lassen, machen sie oft zur ersten und favorisierten Instanz der Validierung theoretischer Vorstellungen von physikalischen oder modellhaften Zusammenhängen. Simulationen bieten in der statistischen Physik gegenüber nicht–statistischen Methoden der numerischen Approximation den Vorteil, daß hier eher selten vereinfachende Annahmen mit den daraus resultierenden unkontrollierbaren systematischen Fehlern notwendig sind und sich die stattdessen auftretenden statistischen Fehler genau bestimmen lassen. Durch die Unabhängigkeit von bestimmten Modellannahmen lassen sich darüber hinaus mit Simulationen oft Kenngrößen messen, die analytisch gar nicht zugänglich sind.

Ziel dieser Arbeit ist es, die Gültigkeit der Henkelschen Vermutung für das Ising–Modell mit einer größeren Genauigkeit zu überprüfen und die Möglichkeit ihrer Übertragbarkeit auf andere Spin–Modelle zu untersuchen. Wir werden uns dazu einer Monte–Carlo–Simulation unter Verwendung effizienter Cluster–Update–Algorithmen bedienen. Neben dem Vorteil der Unabhängigkeit dieser Methode von der Transfermatrix–Rechnung, die zur Formulierung der Hypothese führte, lassen sich mit ihrer Hilfe überdies universelle Amplituden bestimmen, die für die Methode der Transfermatrix nicht erreichbar sind. Nach einer Ausleuchtung des theoretischen Hintergrundes und einer genauen Formulierung der Problemstellung in Kap. 2 sowie einem kurzen Abriß des zur Durchführung der Simulation benötigten Instrumentariums in Kap. 3 werden wir in den Kapiteln 4 und 5 die Kenntnis der exakten Form der betrachteten Ausdrücke für den speziellen Fall des Ising–Modells in zwei Dimensionen dazu benutzen, die verwendeten Werkzeuge optimal abzustimmen, gewissermaßen die Instrumente zu „eichen“. Nach diesen Vorarbeiten geht es in den folgenden Kapiteln 6–9 dann um die Anwendung dieses recht elaborierten Instrumentariums auf einige Vertreter der Klasse der O(n)–Spin–Modelle in drei Dimensionen mit entsprechenden Antworten auf die Frage nach der Gültigkeit und Verallgemeinerbarkeit der Henkelschen Vermutung.

Kapitel 2

Theoretischer Hintergrund

2.1 O(n)–Spin–Modelle

Eine Verallgemeinerung des Ising–Modells hinsichtlich der Dimension des Parameterraumes stellt die Klasse der sog. O(n)–Spin–Modelle dar. Faßt man die beiden Einstellmöglichkeiten eines Ising–Spins als die Variation eines Vektors konstanter Länge in *einer* Dimension auf, so läßt sich dieses Modell in naheliegender Weise als Grenzfall einer allgemeineren Klasse von Modellen auffassen; man betrachtet dabei jetzt Spins $\mathbf{s}_i \in S^n$, d.h. $\mathbf{s}_i \in \mathbb{R}^n$ und $|\mathbf{s}_i| = 1$. Den Fall $n = 2$ bezeichnet man als XY–Modell und entsprechend die Wahl $n = 3$ (auch: $n \geq 3$) als Heisenberg–Modell. Die Dynamik des Systems entspricht dann gerade der sukzessiven Anwendung von Koordinatentransformationen der orthogonalen Gruppe O(n) auf die Zustandsvektoren, d.h. Drehungen und Spiegelungen im $\mathbb{R}^n$. Die Hamilton–Funktion eines solchen Modells ist in entsprechender Verallgemeinerung des Ising–Falls gegeben durch:

$$\mathcal{H} = -J \sum_{<ij>} \mathbf{s}_i \cdot \mathbf{s}_j - \mathbf{H} \sum_i \mathbf{s}_i, \quad \mathbf{s}_i \in S^n, \tag{2.1}$$

woraus man die Zustandssumme gemäß

$$Z \equiv Z(\beta) = \sum_{\{\mathbf{s}_i\}} \exp(-\beta \mathcal{H}(\{\mathbf{s}_i\})), \tag{2.2}$$

erhält. Die „Spins“ $\mathbf{s}_i$ sind dabei positioniert auf einem d–dimensionalen kubischen Gitter mit den Abmessungen $(L_1, \ldots, L_d)$ und dem Volumen $V = \prod_{i=1}^{d} L_i$. Die in J kodierte Wechselwirkung zwischen den Spins ist isotrop und wird ferromagnetisch angenommen, d.h. $J > 0$, so daß eine parallele Ausrichtung benachbarter Spins energetisch belohnt wird. Im einfachsten Fall, von dem im folgenden ausschließlich die Rede sein wird, erfolgt dabei die Summation über alle Paare von nächsten Nachbarn $\mathbf{s}_i$ und $\mathbf{s}_j$. Die Summation über $\{\mathbf{s}_i\}$ meint dabei eine Summe über die Gesamtzahl der aus den mikroskopischen Freiheitsgraden resultierenden Konfigurationen des Gesamtsystems. Desweiteren beschränken wir uns hier auf den feldfreien Fall $\mathbf{H} = 0$.

Im thermodynamischen Limes $L_i \to \infty \ \forall i \in \{1, \ldots, d\}$ zeigen einige Modelle in Abhängigkeit von der Temperatur einen Übergang von einer geordneten Tieftemperaturphase in eine ungeordnete Hochtemperaturphase, der durch Divergenzen in bestimmten Ableitungen der freien Energie

$$F = -\frac{1}{\beta}\ln(Z) \tag{2.3}$$

des Systems gekennzeichnet ist. In zwei Raumdimensionen haben nur zwei der O(n)–Modelle überhaupt einen temperaturgetriebenen Phasenübergang: während das zweidimensionale Ising–Modell ($n = 1$) bei $\beta_c = \frac{1}{2}\ln(1+\sqrt{2})$ einen kontinuierlichen Phasenübergang oder Phasenübergang zweiter Ordnung durchläuft, zeigt schon das XY–Modell ($n = 2$) nur noch einen extrem schwachen Phasenübergang, nämlich einen sogenannten *Kosterlitz–Thouless–Übergang* [61], der auch als Phasenübergang unendlicher Ordnung bezeichnet wurde. Alle Modelle höherer Ordnung ($n \geq 3$) lassen keine Hinweise auf einen temperaturgetriebenen Phasenübergang mehr erkennen[1]. Tatsächlich läßt sich für alle Modelle mit rotationssymmetrischem Ordnungsparameter, insbesondere also die O(n)–Spin–Modelle mit $n \geq 2$, das *Mermin–Wagner–Theorem* beweisen [63], das für diese Modelle in weniger als drei Raumdimensionen die Möglichkeit des Aufbaus einer langreichweitigen Ordnung verneint, so daß ein Phasenübergang endlicher Ordnung in diesen Fällen unmöglich ist.

In drei Dimensionen dagegen zeigen *alle* O(n)–Spin–Modelle einen kontinuierlichen Phasenübergang. Während sich für die Modelle in zwei Raumdimensionen eine Reihe von exakten Ergebnissen angeben läßt, sind für alle O(n)–Spin–Modelle für $d = 3$ lediglich numerische Näherungen (Transfermatrix–Rechnungen, Reihen–Entwicklungen, MC–Simulationen, feldtheoretische Methoden etc.) für wichtige Parameter verfügbar. In diesem Sinne kommt den hier untersuchten Vermutungen für

[1] So lautet zumindest die herrschende Meinung; für eine abweichende Auffassung im Falle des Heisenberg–Modells in zwei Dimensionen vgl. [72].

dreidimensionale Systeme eine Ausnahmestellung in der Theorie der kondensierten Materie zu; hierin liegt gerade der Reiz ihrer Untersuchung.

Es ist bekannt, daß gewisse Limites der $O(n)$–Spin–Modelle hinsichtlich ihrer kritischen Eigenschaften äquivalent sind zu zwei exakt lösbaren Modellen [80], nämlich der Fall $n \to \infty$ dem *sphärischen Modell* und der Grenzübergang $n \to 0$ dem *Gaußschen Modell* [8]. Beide beruhen auf der Idee, die mathematische Behandlung von Ising–Spins s_i zu vereinfachen, indem man die Restriktion $|s_i| = 1$ fallen läßt. Beim Gaußschen Modell fügt man dem Hamiltonian stattdessen einen Selbstenergie–Term $\frac{1}{2\beta} \sum_i s_i^2$ hinzu, der zum Hochtemperatur–Limes $s_i = 0 \; \forall i$ führt; am kritischen Punkt aber richten sich die Spins aufgrund des Wechselwirkungs–Terms nicht nur aus, sondern divergieren sogar. Um dies zu verhindern, ersetzt das sphärische Modell den Selbstenergie–Term durch die Restriktion $\sum_i s_i^2 = N$. Die für beide Modelle verfügbaren exakten Ergebnisse sind als Grenzfälle nützlich, wenn man die Abhängigkeit von Parametern der $O(n)$–Spin–Modelle von der Dimension n des Ordnungsparameters untersuchen will.

2.2 Observablen

Typische charakterisierende Größen in der mathematischen Statistik und somit auch der statistischen Mechanik sind Korrelationsfunktionen der elementaren Observablen; insofern nämlich jede stochastische Verteilungsfunktion durch ihre Momente eindeutig bestimmt wird, ist eine Untersuchung der Momente oder Korrelationsfunktionen einer Stichprobe als eine indirekte Methode der Schätzung der Verteilungsfunktion anzusehen. Die natürlichen Observablen der $O(n)$–Spin–Modelle sind einerseits die Spins $\mathbf{s}_i$ selbst, mithin die lokale Magnetisierung $\mathbf{m}(i)$, andererseits die lokale Energiedichte $\epsilon(i)$[2]; die entsprechenden Zweipunkt–Korrelationsfunktionen

$$\begin{aligned} G_\sigma(i,j) &= \langle \mathbf{m}(i) \cdot \mathbf{m}(j) \rangle, \\ G_\epsilon(i,j) &= \langle \epsilon(i)\, \epsilon(j) \rangle \end{aligned} \tag{2.4}$$

liefern dann die erste nichttriviale Ordnung der Momenten–Entwicklung. Abseits des kritischen Punktes erwartet man Korrelationen endlicher Reichweite, so daß die Größen $\mathbf{m}(i)$ und $\mathbf{m}(j)$ für $|i-j| \to \infty$ statistisch unabhängig werden. Um in

[2]Inwiefern diese beiden Observablen tatsächlich *die* elementaren Beobachtungsgrößen der $O(n)$–Spin–Modelle darstellen wird im Rahmen der konformen Feldtheorie klar. S.u. Kap. 2.5

diesem Limes verschwindende Korrelationsfunktionen zu erhalten, zieht man diesen Grenzwert ab und definiert die sog. *zusammenhängenden* Korrelationsfunktionen

$$\begin{aligned} G^c_\sigma(i,j) &= \langle \mathbf{m}(i)\cdot\mathbf{m}(j)\rangle - \langle\mathbf{m}\rangle\langle\mathbf{m}\rangle, \\ G^c_\epsilon(i,j) &= \langle \epsilon(i)\,\epsilon(j)\rangle - \langle\epsilon\rangle\langle\epsilon\rangle, \end{aligned} \tag{2.5}$$

wo $\langle\mathbf{m}(i)\rangle = \langle\mathbf{m}(j)\rangle = \langle\mathbf{m}\rangle$ und $\langle\epsilon(i)\rangle = \langle\epsilon(j)\rangle = \langle\epsilon\rangle \;\forall\{i,j\}$ vorauszusetzen ist. Aufgrund der endlichen Reichweite der ursprünglichen Wechselwirkung erwartet man abseits vom kritischen Punkt ein Verhalten gemäß:

$$G^c(i,j) \sim \exp\left(-\frac{|i-j|}{\xi}\right), \quad T \neq T_c, \quad |i-j| \gg 1, \tag{2.6}$$

was die *Korrelationslängen* ξ_s und ξ_ϵ der primären Observablen definiert. Sie kennzeichnen die typische Größe von Gebieten des Gitters, in denen die zugehörige Observable wenig schwankt. Während Phasenübergänge 1. Ordnung durch Phasenkoexistenz gekennzeichnet sind, Korrelationslängen also endlich bleiben, bilden sich im Verlaufe eines Phasenüberganges zweiter Ordnung wie etwa bei den O(n)–Spin–Modellen in drei Raumdimensionen Cluster, die am kritischen Punkt das gesamte Gitter überspannen; die Korrelationslängen lokaler Observablen divergieren: ein Indikator des Aufbaus einer langreichweitigen Ordnung. Der exponentielle Verlauf von $G^c(i,j)$ gemäß (2.6) geht in diesem Fall über in ein potenzartiges Verhalten entsprechend einer Korrelation unendlicher Reichweite. Außer der (im unendlich großen System irrelevanten) Kantenlänge a der Elementarzelle des Gitters gibt es dann im System keine charakteristische Längenskala mehr — das Spin–Gitter hat die Struktur eines selbstähnlichen, zufälligen Fraktals.

2.3 Finite Size Scaling (FSS)

Betrachtet man jetzt endliche Systeme, wie sie sich mit numerischen Methoden (etwa MC–Simulationen) ausschließlich untersuchen lassen, muß das Konzept des Phasenüberganges modifiziert werden. Eine echte Divergenz charakteristischer Größen ist in diesem Falle (zumindest für den betrachteten Fall der O(n)–Spin–Modelle) unmöglich: das zeigt sich besonders anschaulich bei den Korrelationslängen, für die im endlichen System offenbar immer $\xi \leq \max_{i=1}^{d} L_i$ gelten muß. Stattdessen erhält man eine mit wachsender Systemgröße zum Pol kontrahierende Folge von Maxima der entsprechenden Größen, wobei deren Positionen auf der Temperaturachse — die (pseudo–)kritischen Punkte der endlichen Systeme — gegenüber dem (echten) kritischen Punkt des unendlich großen Systems verschoben erscheinen. Da man für jedes endliche System bei der Temperatur T_c noch vom eigentlichen kritischen Verhalten entfernt ist, behält hier der exponentielle Verlauf der Korrelationsfunktionen (2.6) seine Gültigkeit. Das Konzept des *Finite Size Scaling* [5, 75] baut auf dieser „Dualität" zwischen der Annäherung $T \to T_c$ an den kritischen Punkt des unendlichen Systems und dem Vollzug des thermodynamischen Limes $L_i \to \infty\ \forall i = 1, \ldots, d$ aus dem endlichen System bei $T = T_c$ auf.

Das Verhalten vieler Gitter–Modelle ist im thermodynamischen Limes bei Vollzug des erstgenannten Grenzübergangs, d.h. einer Annäherung an den kritischen Punkt — sei es durch analytische Lösung des Modells, durch Renormierungsgruppen–Untersuchungen[3] oder numerische Methoden — in einer teilweise quantitativen Weise bekannt: die charakteristischen Größen wie die magnetische Suszeptibilität, die spezifische Wärme oder die Korrelationslänge können im allgemeinen gemäß einem Potenzgesetz divergieren[4]:

$$A(t, L = \infty) \approx A_{\pm}|t|^{-\omega},\ t = \frac{T - T_c}{T_c},\ \omega \in \mathrm{I\!R}, \tag{2.7}$$

wobei die kritischen Exponenten ω spezifische Funktionen des Modells und der betrachteten Größe A darstellen. Der Fall eines höchstens unstetigen Verhaltens ohne Pol — wie etwa die Korrelationslänge beim Phasenübergang 1. Ordnung — ist hierbei inbegriffen, insofern $\omega = 0$ eine zulässige Wahl darstellt[5]. Im allgemeinen Fall kann die Konstante A für $t < 0$ und für $t > 0$ unterschiedliche Werte annehmen. Das Zeichen „$\approx$" soll andeuten, daß höhere Ordnungen in $1/|t|$, d.h. Korrekturen zum Skalenverhalten vernachlässigt wurden.

Für ein endliches Gitter soll dann zusätzlich die Abhängigkeit von der linearen Systemgröße berücksichtigt, d.h. die zweite Möglichkeit des Grenzüberganges beschrieben werden. Da in eine potenzartige oder exponentielle Beziehung nur eine dimensionslose Variable eingehen darf und die einzige von L unabhängige charakteristische Längenskala des Systems durch die Korrelationslänge des Ordnungsparameters im unendlichen System gegeben ist, kommt als *Skalenvariable* nur die Kombination

[3] s.u. Kap. 2.4

[4] Die potenzartige Form hängt dabei mit der Forderung nach Invarianz des Systems unter Skalentransformationen zusammen, vgl. [30].

[5] Das ist zu unterscheiden von logarithmischen Divergenzen, die manchmal auch durch $\omega = 0$ charakterisiert werden.

$z = \frac{L}{\xi(t,L=\infty)}$ in Frage. ξ selbst wiederum skaliert gemäß (2.7) mit dem kritischen Exponenten $\omega = \nu$. Man macht daher den FSS–Ansatz:

$$A(t,L) \approx |t|^{-\omega} \mathcal{A}_\pm(L|t|^\nu), \tag{2.8}$$

wobei sich die FSS–Funktion $\mathcal{A}$ in den Limites $L \to \infty$ und $t \to 0$ singulär verhalten muß um einerseits die Abhängigkeit (2.7) zu reproduzieren, andererseits aber der Beobachtung Rechnung zu tragen, daß das System für jedes endliche L keinen echten Phasenübergang bei T_c zeigt, so daß $A(t,L)$ für jedes $t \in \mathbb{R}$ endlich bleibt. Um eine Skalenfunktion mit regulärem Verhalten in den relevanten Grenzfällen zu erhalten, kann man wie folgt transformieren:

$$\mathcal{A}(x) = x^{\frac{\omega}{\nu}} \tilde{\mathcal{A}}(x^{\frac{1}{\nu}}) \tag{2.9}$$

und erhält damit:

$$A(t,L) \approx L^{\frac{\omega}{\nu}} \tilde{\mathcal{A}}(L^{\frac{1}{\nu}} t). \tag{2.10}$$

Das ist die Grundform der FSS–Hypothese ohne Berücksichtigung des magnetischen Feldes. Mit Hilfe von (2.10) lassen sich die FSS–Exponenten von Observablen aus den zugehörigen kritischen Exponenten bestimmen. Umgekehrt werden solche Beziehungen ausgenutzt, um aus dem FSS–Verhalten endlicher Systeme in Simulationen oder Transfermatrix–Rechnungen kritische Exponenten sehr genau zu bestimmen.

Für $A = \xi$ erhält man am kritischen Punkt die wichtige Beziehung

$$\xi(0,L) \approx L\,\tilde{\mathcal{A}}(0), \tag{2.11}$$

im endlichen System skalieren die Korrelationslängen ξ_s und ξ_ϵ am kritischen Punkt linear mit der Systemgröße; die Konstante $\tilde{\mathcal{A}}(0)$ ist universell im Sinne der Theorie der Renormierungsgruppe, d.h. unabhängig von den mikroskopischen Details des Systems.

2.4 Renormierung und Universalität

Die phänomenologischen Skalenrelationen (2.7) und (2.10) lassen sich auch von einem analytischen Standpunkt aus erschließen, dem Konzept der sog. (Ortsraum–) *Renormierung*, das hier als Vorbereitung auf die konforme Feldtheorie kurz dargestellt werden soll. Man betrachte dazu ein System von N Spins auf einem d–dimensionalen kubischen Gitter der Kantenlänge a. Eine Renormierungsvorschrift ist dann gegeben durch das Zusammenfassen von Spins in Volumina der Größe $(ab)^d$, wo $b > 1$ einen Reskalierungsfaktor darstellt. Diesem *Block* von Spins ordnet man nun einen renormierten oder Blockspin zu, was eine Abbildung von einer Konfiguration $\{\mathbf{s}_i\}$ der N Ausgangsspins auf eine Konfiguration $\{\mathbf{s}'_i\}$ des renormierten Gitters der Größe $N' = N/b^d$ definiert. Für die Regel zur Zuordnung der Blockspins als Funktion der Spins im gegebenen Block gibt es eine ganze Reihe von unterschiedlichen Möglichkeiten [81], so etwa im Falle des Ising–Modells eine Mehrheitsregel gemäß:

$$s'_i = \begin{cases} +1, & \sum_{\text{Block}} s_i > 0 \\ -1, & \sum_{\text{Block}} s_i < 0 \end{cases} , \tag{2.12}$$

wobei für $\sum_{\text{Block}} s_i = 0$ der Spin s'_i gemäß einer symmetrischen Zweipunkt–Verteilung zufällig gezogen werden kann. Um die spezifischen Eigenschaften des Modells in der RG–Trafo zu erhalten, ist es lediglich notwendig, daß die Blocking–Transformation die neue Konfiguration in einem gewissen Sinne „sensibel“ aus der alten gewinnt, d.h. effektive Änderungen in der Konfiguration $\{\mathbf{s}_i\}$ auch auf die neue Konfiguration $\{\mathbf{s}'_i\}$ abbildet und dabei der Parameterraum der $\mathbf{s}'_i$ mit dem der $\mathbf{s}_i$ übereinstimmt. Es werden mithin alle Freiheitsgrade des Systems innerhalb der Blocks gemäß einer vorgegebenen Regel *ausintegriert*. Die Renormierungsgruppen–Abbildung sei mit R bezeichnet, d.h. es ist $R(\{\mathbf{s}_i\}) = \{\mathbf{s}'_i\}$.

Über die Verteilungsfunktionen der $\{\mathbf{s}_i\}$ bzw. $\{\mathbf{s}'_i\}$ läßt sich dann unter Zugrundelegung des kanonischen Ensembles ein *effektiver Hamiltonian* des transformierten Systems definieren. War die Verteilung vor der Transformation gegeben durch

$$P(\{\mathbf{s}_i\}) = \frac{1}{Z} \exp(-\mathcal{H}(\{\mathbf{s}_i\})), \tag{2.13}$$

wobei hier die inverse Temperatur β in die Definition von $\mathcal{H}$ absorbiert wurde, so ergibt sich die neue Verteilung durch Summation über die mit $\{\mathbf{s}'_i\}$ verträglichen Konfigurationen $\{\mathbf{s}_i\}$:

$$\begin{aligned} P(\{\mathbf{s}'_i\}) &= \sum_{R(\{\mathbf{s}_i\})=\{\mathbf{s}'_i\}} P(\{\mathbf{s}_i\}) \\ &= \sum_{\{\mathbf{s}_i\}} P(\{\mathbf{s}_i\}, \{\mathbf{s}'_i\}) P(\{\mathbf{s}_i\}) \\ &\equiv \frac{1}{Z'} \exp(-\mathcal{H}'(\{\mathbf{s}'_i\})) \end{aligned} \tag{2.14}$$

Diese Gleichung definiert $\mathcal{H}'(\{\mathbf{s}'_i\})$ bis auf eine additive Konstante. Diese wählen wir so, daß gerade die Zustandssumme invariant bleibt, denn dann ist:

$$F' = -\ln Z'$$

$$
\begin{aligned}
&\stackrel{!}{=} \; -\ln Z \\
&= \; F
\end{aligned}
\tag{2.15}
$$

Für die freie Energiedichte $f = F/N$ erhält man:

$$
\begin{aligned}
Nf[\mathcal{H}, N] &= N' f[\mathcal{H}', N'] \\
f[\mathcal{H}, N] &= \frac{N'}{N} f[\mathcal{H}', N'] \\
&= b^{-d} f[\mathcal{H}', N/b^d]
\end{aligned}
\tag{2.16}
$$

Im transformierten Hamiltonian $\mathcal{H}'$ tritt im allgemeinen noch ein Term auf, der selbst von N abhängt; für den singulären Teil $f^{\sin}$ der freien Energiedichte, der uns wegen seines Skalenverhaltens am kritischen Punkt interessiert, ist aber die N–Abhängigkeit durch (2.16) gegeben. Die Iteration dieses Prozesses entspricht einer Abbildung im Raum der Hamilton–Funktionen; da diese von den Parametern t und h abhängig seien, läßt sich das äquivalent als Abbildung $\mathcal{R}$ im Parameterraum darstellen:

$$
\begin{gathered}
\mathbf{H} = (t, h) \longmapsto \mathbf{H}' = (t', H') \\
\mathbf{H}^{(n+1)} = \mathcal{R}(\mathbf{H}^{(n)}), \; \mathbf{H}^{(0)} = (t, h).
\end{gathered}
\tag{2.17}
$$

Aus thermodynamischen Gründen hat diese Abbildung zwei triviale Fixpunkte, die, wie man zeigen kann, i.a. attraktiv sind; der eine entspricht dem Limes $T \to 0$[6]: ist das ursprüngliche System vollständig geordnet (z.B. $s_i = +1 \; \forall i$ im Falle des Ising–Modells), dann gilt das ebenso für das renormierte System, d.h. $\mathbf{H}' = \mathbf{H}$. Im Limes $T \to \infty$ hingegen überwiegt der entropische Teil der freien Energie alle anderen Beiträge: das System ist ungeordnet, so daß die Spins vollkommen zufällig und unkorreliert sind. Dann aber wird für jede sinnvolle Definition der Blockspins auch das renormierte System die gleichen Eigenschaften aufweisen, so daß wiederum $\mathbf{H}' = \mathbf{H}$.

Weist das System einen Phasenübergang auf, so hat die Abbildung $\mathcal{R}$ — wie man für viele Systeme explizit zeigen kann — einen weiteren Fixpunkt $\mathbf{H}_0$ entsprechend $T = T_c$ bzw. $t = 0$. Um das Verhalten von $\mathcal{H}$ bzw. $\mathbf{H}$ in seiner Umgebung zu analysieren, entwickelt man $\mathcal{R}$ linear um den Fixpunkt $\mathbf{H}_0$. Ist also $\mathbf{H} = \mathbf{H}_0 + \delta\mathbf{H}$, so hat man:

$$
\begin{aligned}
H'_i = \mathcal{R}_i(\mathbf{H}) &\approx \mathcal{R}_i(\mathbf{H}_0) + \sum_j \frac{\partial \mathcal{R}_i}{\partial H_j}(\mathbf{H}_0)\delta H_j \\
&= (\mathbf{H}_0)_i + \sum_j M_{ij}\delta H_j,
\end{aligned}
\tag{2.18}
$$

[6]Während die Temperaturabhängigkeit des ursprünglichen Hamiltonians trivial war, muß dies keineswegs auch für die transformierten Hamilton–Funktionen gelten.

wo

$$M_{ij} = \frac{\partial \mathcal{R}_i}{\partial H_j}(\mathbf{H}_0) \tag{2.19}$$

die Jacobi–Matrix von $\mathcal{R}$ am Fixpunkt ist. Macht man jetzt eine Koordinatentransformation im Parameterraum, so daß $\mathbf{M}$ diagonal wird und bezeichnet die Eigenwerte mit

$$\lambda_i = b^{y_i}, \tag{2.20}$$

so bezeichnen die zugehörigen Eigenvektoren v_i die Richtungen, in denen sich $\mathbf{H}$ in der Nähe von $\mathbf{H}_0$ transformiert wie

$$(v_t, v_h) \longmapsto (b^{y_t} v_t, b^{y_h} v_h). \tag{2.21}$$

Man kann sogar zeigen, daß zu jedem Eigenwert λ_i ein sog. *Skalenfeld* u_i existiert, nämlich die Integralkurve, deren Tangentenvektor in $\mathbf{H}_0$ dem zu λ_i gehörenden Eigenvektor entspricht und zwar so, daß diese dann für *beliebige Entfernungen* von $\mathbf{H}_0$ gemäß (2.21) transformieren. Der Fixpunkt $\mathbf{H}_0$ ist i.a. instabil. Die Skalenfelder, die zu repulsiven Richtungen gehören, d.h. $\lambda_i > 0$ bzw. $y_i > 0$, heißen *relevant*, da sie alleine das Skalenverhalten in der Nähe von $\mathbf{H}_0$ bestimmen. Die Felder mit $y_i < 0$ heißen *irrelevant*, die mit $y_i = 0$ heißen *marginal*. Dies gilt analog für entsprechende zusätzliche Felder, falls der Hamiltonian von weiteren Parametern abhängt. Gl. (2.16) schreibt sich dann in der Nähe von $\mathbf{H}_0$ als

$$f^{\sin}(u_t, u_h; N) = b^{-d} f^{\sin}(b^{y_t} u_t, b^{y_h} u_h; N/b^d). \tag{2.22}$$

Man erkennt, daß sich die inverse Systemgröße $1/N$ wie ein weiteres relevantes Skalenfeld mit Skalenexponent d verhält. Sukzessive Anwendungen der Renormierungsgruppen–Transformation ergeben nun:

$$f^{\sin}(u_t, u_h; 1/N) = b^{-dn} f^{\sin}(b^{ny_t} u_t, b^{ny_h} u_h; b^{nd} \cdot 1/N). \tag{2.23}$$

Zunächst betrachten wir den Limes $N \to \infty$, so daß die N–Abhängigkeit verschwindet. Soll die lineare Näherung (2.18) Gültigkeit behalten, kann n nicht beliebig groß werden ($\mathbf{H}_0$ ist ja repulsiv, falls $y_t > 0$), man stoppt den Prozeß also bei $b^{ny_t} u_t \equiv K$, $|K| \ll 1$. Auflösen nach n und Einsetzen in (2.23) ergibt:

$$\begin{aligned} f^{\sin}(u_t, u_h) &= K^{-d/y_t} u_t^{d/y_t} f^{\sin}(K, K^{y_h/y_t} u_t^{-y_h/y_t} u_h) \\ &\equiv \left|\frac{t}{t_o}\right|^{d/y_t} W_\pm\left(\left(\frac{h}{h_0}\right)\left(\frac{t}{t_0}\right)^{-y_h/y_t}\right), \end{aligned} \tag{2.24}$$

wobei gemäß

$$u_t = c_t t + O(t^2),\ u_h = c_h h + O(ht^2) \tag{2.25}$$

genähert wurde. Die FSS–Funktion W ist universell, alle systemabhängigen Informationen stecken in den beiden Konstanten h_0 und t_0. Das Skalenverhalten von Observablen ergibt sich dann durch entsprechendes Ableiten von f, z.B. für die spezifische Wärme:

$$c(h=0) \sim \left.\frac{\partial^2 f}{\partial t^2}\right|_{h=0} \sim |t|^{d/y_t-2} \tag{2.26}$$

in Übereinstimmung mit der Form (2.7).

Für endliches N ergibt sich entsprechend die Beschränkung $N^{-1}b^{nd} = N'^{-1} \equiv K,\ |K| \ll 1$, mit der Folgerung:

$$\begin{aligned} f^{\sin}(u_t, u_h; 1/N) &= K^{-1}N^{-1}f^{\sin}(K^{y_t/d}N^{y_t/d}u_t, K^{y_h/d}N^{y_h/d}u_h; K) \\ &= \left(\frac{N}{N_0}\right)^{-1} W\left(\left(\frac{N}{N_0}\right)^{y_t/d}\left(\frac{t}{t_0}\right), \left(\frac{N}{N_0}\right)^{y_h/d}\left(\frac{h}{h_0}\right)\right). \end{aligned} \tag{2.27}$$

Mit der linearen Systemgröße $L = N^{1/d}$ schreibt sich dies als

$$f^{\sin}(t, h : 1/N) = \left(\frac{L}{L_0}\right)^{-d} W\left(\left(\frac{L}{L_0}\right)^{y_t}\left(\frac{t}{t_0}\right), \left(\frac{L}{L_0}\right)^{y_h}\left(\frac{h}{h_0}\right)\right), \tag{2.28}$$

was gerade der FSS–Relation (2.10) entspricht; die Exponenten y_i bestimmen sich durch Vergleich dieses Ausdrucks mit den Gl. (2.7) entsprechenden Relationen für die betrachteten Observablen.

Beziehungen für die Korrelationsfunktionen lassen sich ebenfalls finden. Die in der Hamilton–Funktion zum Feld u_i kanonisch konjugierte Variable schreibt man als $\sum_{\mathbf{r}} \phi_i(\mathbf{r})$ und nennt die $\phi_i(\mathbf{r})$ *Skalenoperatoren* oder *Skalendichten* (obwohl sie natürlich keineswegs operatorwertig sind). Man addiert nun zur Hamilton–Funktion Terme gemäß

$$\mathcal{H} = \mathcal{H}_0 + \sum_{\mathbf{r}} u_i(\mathbf{r})\phi_i(\mathbf{r}), \tag{2.29}$$

wobei $u_i(\mathbf{r})$ eine Fortsetzung der u_i darstellt, die nur auf Längenskalen variiert, die groß sind gegenüber der Gitterkonstante a. Da die RG–Transformation lokal ist und ebenso gemäß einer wichtigen Annahme der Theorie der Renormierungsgruppe die $\phi_i(\mathbf{r})$ lokal sind, bleibt die Transformations–Eigenschaft (2.21) erhalten. Unter Ausnutzung der Beziehung

$$\langle \phi_1(\mathbf{r}_1) \cdots \phi_n(\mathbf{r}_n) \rangle = -\frac{\partial^{(n)} F}{\partial g_1(\mathbf{r}_1) \cdots \partial g_n(\mathbf{r}_n)}, \tag{2.30}$$

und Verwendung von (2.3) läßt sich dann zeigen, daß die n–Punkt–Korrelationsfunktionen am Fixpunkt $\mathbf{H}_0$ (d.h. dem kritischen Punkt des Systems) unter der RG–Abbildung transformieren wie

$$\langle \phi_1(\mathbf{r}_1) \cdots \phi_n(\mathbf{r}_n) \rangle = b^{x_1} \cdots b^{x_n} \langle \phi_1(\mathbf{r}'_1) \cdots \phi_n(\mathbf{r}'_n) \rangle. \tag{2.31}$$

Gl. (2.31) ist Ausdruck der *Kovarianz* der Korrelationsfunktion am Punkte des Phasenüberganges. Die Exponenten x_i nennt man *Skalendimensionen* der Operatoren ϕ_i; sie sind gegeben durch:

$$x_i = d - y_i. \tag{2.32}$$

Das Auftreten der Raumdimension d hat seinen Ursprung in der Transformation des Volumenelementes in der Summe in (2.29).

Ein Vergleich mit entsprechenden Aussagen der phänomenologischen Beschreibung des Skalenverhaltens erlaubt es nun, die Skalendimensionen durch die klassischen kritischen Exponenten auszudrücken. Für die spezifische Wärme etwa gilt:

$$C_{h=0} = \partial^2 f / \partial t^2|_{h=0} \sim |t|^{d/y_t - 2} \implies \alpha = 2 - d/y_t. \tag{2.33}$$

Ähnliche Betrachtungen für die Magnetisierung, die Suszeptibilität etc. ergeben insgesamt die Relationen:

$$\begin{aligned} \alpha &= 2 - d/y_t, \\ \beta &= (d - y_h)/y_t, \\ \gamma &= (2y_h - d)/y_t, \\ \delta &= y_h/(d - y_h), \\ \eta &= d + 2 - 2y_h, \\ \nu &= 1/y_t. \end{aligned} \tag{2.34}$$

Durch Elimination der Abhängigkeit von den RG–Eigenwerten y_i erhält man die *Skalenrelationen* [24]

$$\begin{aligned} \alpha + 2\beta + \gamma &= 2, \\ \alpha + \beta(1 + \delta) &= 2, \\ \nu(2 - \eta) &= \gamma, \end{aligned} \tag{2.35}$$

sowie die *Hyper–Skalenrelation*

$$\alpha = 2 - d\nu. \tag{2.36}$$

Mit Hilfe von (2.34) lassen sich nun auch die Skalendimensionen x_i durch die üblichen kritischen Exponenten ausdrücken. Die zum Temperaturfeld u_t kanonisch konjugierte Variable entspricht gerade der Energiedichte ϵ, die zu u_h konjugierte der lokalen

Magnetisierung $\mathbf{m}$. Man hat dann:

$$x_\epsilon \equiv x_t = \frac{1-\alpha}{\nu} \qquad x_\sigma \equiv x_h = \frac{\beta}{\nu}. \tag{2.37}$$

Erweitert man die Klasse der RG–Transformationen von den einfachen Streckungen bzw. Stauchungen um Translationen und Rotationen, so reicht die Restriktion (2.31) aus, die Zweipunkt–Funktion am kritischen Punkt festzulegen. Man setzt jedoch voraus, daß das System (mindestens am kritischen Punkt) translationsinvariant (d.h. unendlich groß) und rotationssymmetrisch ist (was für ein diskretes Gitter im strengen Sinne nicht erfüllt sein kann). Homogenität und Isotropie des Systems verlangen dann, daß die Korrelationen nur vom Abstand $|\mathbf{r}_1 - \mathbf{r}_2|$ abhängen; soll auch noch das Skalenverhalten (2.31) gültig sein, so ist die Zweipunkt–Funktion bis auf einen Normierungsfaktor eindeutig gegeben durch:

$$\langle \phi_1(\mathbf{r})\phi_2(\mathbf{r})\rangle = |\mathbf{r}_1 - \mathbf{r}_2|^{-2x}. \tag{2.38}$$

2.5 Konforme Feldtheorie

Die Renormierungsgruppen–Transformation ist gekennzeichnet durch eine einzige Konstante b: alle Teile des Gitters werden mit demselben Faktor reskaliert. Die zugrundeliegende Symmetrie ist die einer *globalen* Skaleninvarianz ($\mathbf{r}' = b\mathbf{r}$) eines thermodynamischen Systems am Punkte eines Phasenüberganges zweiter Ordnung. Die Idee der *konformen Feldtheorie* ist es, diese Transformation zu *lokalisieren*, d.h. gemäß

$$\mathbf{r}' = b(\mathbf{r})\mathbf{r} \tag{2.39}$$

zu transformieren [19]. In Abb. 2.1 ist dies anschaulich dargestellt. Dies ist natürlich nur zulässig, wenn $b(\mathbf{r})$ entsprechend langsam variiert und andererseits die Wechselwirkungen in der Hamilton–Funktion kurzreichweitig sind (was jedoch in den meisten Modellen der Fall ist). Die Transformation erfolge dabei so, daß lokal die Überlegungen aus der Renormierungsgruppen–Theorie gültig bleiben: in diesem Grenzfall soll (2.39) darstellbar sein als Summe von Translationen, Rotationen und einfachen (globalen) Skalentransformationen. Mit anderen Worten: die Transformation darf keine Scherkomponenten enthalten. Solche Abbildungen nennt man *konform*.

Die Beschreibung der Klasse solcher Transformationen ist dann ein genuin geometrisches Problem; nur mit geometrischen Methoden läßt sich auch die volle Information

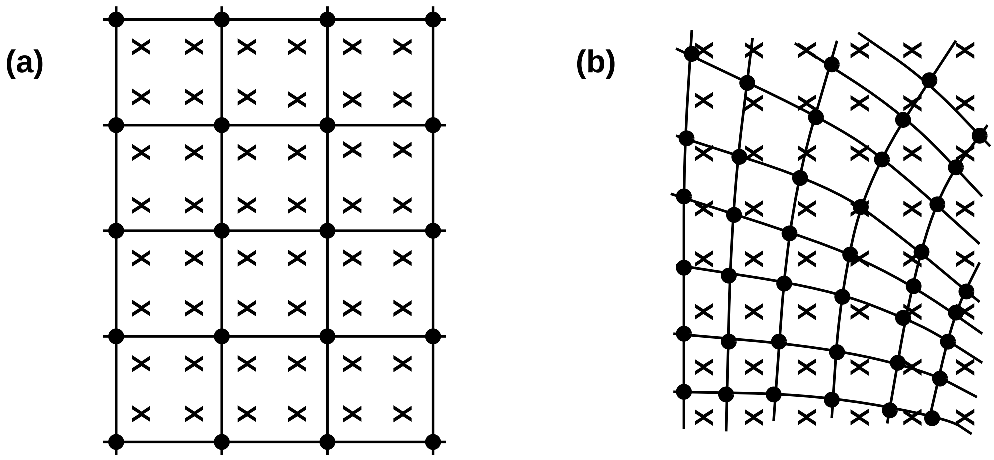


Abb. 2.1: Renormierungsgruppen–Transformation und konforme Transformation im Vergleich. Die Kreuze bezeichnen die Spins des ursprünglichen Gitters und die Punkte die transformierten Variablen. (a) Die Ortsraum–Renormierungsprozedur reskaliert das gesamte Gitter mit einem konstanten Faktor b; die Geometrie bleibt unverändert. (b) Eine konforme Transformation ist eine lokale Reskalierung $b(\mathbf{r})$, die das Gitter verzerrt; lediglich die Winkel zwischen den Gitterlinien bleiben unverändert.

extrahieren, die in der Eigenschaft der konformen Invarianz eines Systems der statistischen Mechanik enthalten ist. Um Ergebnisse von größtmöglicher Allgemeinheit zu gewinnen, müssen die apriorischen Voraussetzungen an die Topologie und Geometrie des betrachteten Systems minimal bleiben; diese Forderung läßt sich erfüllen, indem man alle Gleichungen im Kalkül der Differentialgeometrie für (differenzierbare) Mannigfaltigkeiten notiert. Ein solcher Aufwand an mathematischen Werkzeugen mag übertrieben erscheinen; es wird sich jedoch herausstellen, daß es dieses Aufwandes bedarf, um eine plausible Eingrenzung der Fälle zu ermöglichen, in denen das Konzept der konformen Invarianz in seiner vollen Tragweite anwendbar ist. Hinsichtlich der Notation und der zugrundeliegenden mathematischen Instrumente sei der Leser an die Darstellungen von Nakahara [67] und Spivak [79] verwiesen. Im Kalkül der Differentialgeometrie nimmt nun das Konzept einer konformen Transformation folgende Form an: ein Diffeomorphismus

$$f : M \longrightarrow N \tag{2.40}$$

zwischen zwei semi–riemannschen differenzierbaren Mannigfaltigkeiten M und N mit metrischen Tensoren g und $\tilde{g}$ heißt *konform*, wenn die von f auf M induzierte

Metrik bis auf eine skalare, positive Funktion mit g übereinstimmt:

$$f^*\tilde{g} = e^{2\sigma} g, \qquad \sigma \in \mathcal{F}(M), \tag{2.41}$$

wobei die Pullback–Abbildung f^* über das Differential $f_* : T_pM \to T_{f(p)}N$ von f definiert ist:

$$(f^*\tilde{g})(X,Y) \equiv \tilde{g}(f_*X, f_*Y), \qquad X, Y \in V(M). \tag{2.42}$$

Das heißt: Längen werden um einen lokalen Faktor $e^{2\sigma}$ gestreckt bzw. gestaucht, aber die Winkel zwischen Vektoren $X_p, Y_p \in T_pM$ gemäß

$$\cos(\alpha) = \frac{g_p(X_p, Y_p)}{[g_p(X_p, X_p) g_p(Y_p, Y_p)]^{1/2}} \tag{2.43}$$

bleiben erhalten.

Insbesondere betrachtet man den Spezialfall von konformen Automorphismen $f\colon M \to M$; sie bilden eine Gruppe von Transformationen, die sogenannte konforme Gruppe $\mathrm{Conf}(M)$. Die Struktur dieser Gruppe ist abhängig einerseits von der Metrik g der Mannigfaltigkeit M, andererseits, wie wir sehen werden, entscheidend von der Dimension $m = \dim(M)$. Dazu schreiben wir (2.41) für den Fall $N = M$ in lokalen Koordinaten; sind $\{x_i = \varphi_i(p)\}$ Koordinaten um p, $\{y_i = \psi_i(f(p)\}$ Koordinaten um $f(p)$, so hat man:

$$\frac{\partial y^\alpha}{\partial x^\mu}\frac{\partial y^\beta}{\partial x^\nu} g_{\alpha\beta}(f(p)) = e^{2\sigma(p)} g_{\mu\nu}(p), \tag{2.44}$$

wo $\frac{\partial y^\alpha}{\partial x^\mu} \equiv \frac{\partial(\psi\circ f\circ\varphi^{-1})^\alpha}{\partial x^\mu}$ die Jacobi–Matrix der konformen Abbildung f ist. Im trivialen Fall $m = 1$ ist offenbar jede Transformation $f \in \mathrm{Diff}(M)$ konform. Auch auf zweidimensionalen Mannigfaltigkeiten läßt sich der Gehalt von $\mathrm{Conf}(M)$ ohne spezielle Voraussetzungen an g bestimmen. Man kann zeigen, daß jede zweidimensionale riemannsche Mannigfaltigkeit (M, g) konform flach ist [67], d.h. für jeden Punkt $p \in M$ gibt es eine Karte, so daß g die Komponenten

$$g_{\mu\nu} = e^{2\sigma(p)} \delta_{\mu\nu} \tag{2.45}$$

hat. Das Linienelement auf M hat also die lokale Form

$$ds^2 = g(x,y)(dx^2 + dy^2) = h(z,\bar{z})\, dz\, d\bar{z}, \tag{2.46}$$

wenn man x und y als Real– und Imaginärteil einer komplexen Variablen z auffaßt. Macht man jetzt eine Transformation $z = z(w)$, die als Funktion von z komplex analytisch ist, d.h.

$$\frac{\partial z}{\partial \bar{w}} = 0, \tag{2.47}$$

so transformieren sich die Differentiale gemäß

$$dz = \left(\frac{\partial z}{\partial w}\right) dw \qquad d\bar{z} = \overline{\left(\frac{\partial z}{\partial w}\right)} d\bar{w} \tag{2.48}$$

und man hat:

$$ds^2 = h(z,\bar{z})\, dz\, d\bar{z} = h(z(w,\bar{w}), \overline{z(w,\bar{w})}) \left|\frac{\partial z}{\partial w}\right|^2 dw\, d\bar{w}, \tag{2.49}$$

was wiederum eine konforme Metrik darstellt. Jede holomorphe Transformation ist daher in zwei Dimensionen konform, sofern (vgl. (2.41))

$$\frac{\partial z}{\partial w} = z'(w) \neq 0. \tag{2.50}$$

Man kann zeigen [79], daß damit sogar *alle* konformen Transformationen in zwei Dimensionen charakterisiert sind. Die Gruppe $\mathrm{Conf}(M)$ ist in diesem Fall unendlichdimensional.

Anders ist die Lage im Fall dreier oder mehr Raumdimensionen: ohne Spezifikation der Metrik kann man hier die Menge der Transformationen nicht mehr explizit bestimmen. Eine wichtige strukturelle Aussage ist jedoch möglich [60]: ist M riemannsch und zusammenhängend, so hat $\mathrm{Conf}(M)$ die Eigenschaften einer *Lie*–Gruppe mit

$$\dim \mathrm{Conf}(M) \leq \frac{1}{2}(m+1)(m+2). \tag{2.51}$$

Unabhängig von der Metrik (bis auf den Index von 0) unterscheiden sich daher die konformen Gruppen auf zwei– und höherdimensionalen Mannigfaltigkeiten nach unendlicher und endlicher Dimension. Um die funktionale Form der konformen Transformationen für $m \geq 3$ zu erhalten, muß man die Metrik spezifizieren. Nach einem Satz von Liouville [29] ist jede konforme Transformation in einem euklidischen (bzw. pseudo–euklidischen) Raum mit einer Dimension größer als zwei die Komposition von Isometrien, d.h. euklidischen Transformationen (Translationen und Rotationen) bzw. Poincaré–Transformationen (Translationen und Lorentz–Transformationen), Dilatationen mit einem konstanten Faktor und Inversions–Abbildungen

$$x'^{\mu} = a^2 \frac{x^{\mu}}{x^2}. \tag{2.52}$$

Die Rolle der Inversionen kann äquivalent dazu beschrieben werden durch die sog. *speziellen konformen Transformationen*, die gegeben sind durch:

$$x'^{\mu} = \frac{x^{\mu} + a^{\mu} x}{1 + 2a_{\mu}x^{\mu} + a^2 x^2} \tag{2.53}$$

und gemäß

$$\frac{x'^{\mu}}{x'^{2}} = \frac{x^{\mu}}{x^{2}} + a^{\mu} \tag{2.54}$$

aus Inversionen und Translationen zusammengesetzt sind.

Dies sieht man am einfachsten, wenn man infinitesimale konforme Transformationen $x^{\mu} \mapsto x^{\mu} + \epsilon X^{\mu}$ (in lokalen Koordinaten) betrachtet, die von einem Vektorfeld $X = X^{\mu}\partial_{\mu} \in V(M)$ erzeugt werden. Gl. (2.44) schreibt sich dann als

$$\frac{\partial(x^{\kappa} + \epsilon X^{\kappa})}{\partial x^{\mu}} \frac{\partial(x^{\lambda} + \epsilon X^{\lambda})}{\partial x^{\nu}} g_{\kappa\lambda}(x + \epsilon X) = e^{2\sigma} g_{\mu\nu}(x), \tag{2.55}$$

oder mit $\sigma = \epsilon\psi/2$:

$$X^{\xi}\partial_{\xi} g_{\mu\nu} + \partial_{\mu} X^{\kappa} g_{\kappa\nu} + \partial_{\nu} X^{\lambda} g_{\mu\nu} = \psi g_{\mu\nu}. \tag{2.56}$$

ψ ergibt sich damit zu:

$$\psi = m^{-1}(X^{\xi} g^{\mu\nu}\partial_{\xi} g_{\mu\nu} + 2\partial_{\mu} X^{\mu}). \tag{2.57}$$

Solche Vektorfelder heißen *konforme Killing–Vektorfelder*. Gl. (2.56) wird in diesem Zusammenhang als konforme Killing–Gleichung bezeichnet und kann mit der Lie–Ableitung gemäß $\mathcal{L}_X g = \psi g$ geschrieben werden. Die Killing–Vektorfelder bilden bzgl. der Lie–Ableitung eine Lie–Algebra, die isomorph zur Lie–Algebra von $\mathrm{Conf}(M)$ ist, falls die Gruppe eine Lie–Gruppe ist [67]. Für eine euklidische Metrik $g_{\mu\nu} = \delta_{\mu\nu}$ erhält man:

$$\partial_{\mu} X_{\nu} + \partial_{\nu} X_{\mu} = \frac{2}{m}(\partial \cdot X)\delta_{\mu\nu}. \tag{2.58}$$

Nimmt man die Divergenz dieser Gleichung, so ergibt sich [58]:

$$\left[\delta_{\mu\nu}\Box + (m-2)\partial_{\mu}\partial_{\nu}\right](\partial \cdot X) = 0. \tag{2.59}$$

An Gleichung (2.59) läßt sich direkt ablesen, daß sich die Lösungen für $m = 2$ von denen für $m \geq 3$ fundamental unterscheiden. Wendet man den Operator $\partial^{\mu}\partial^{\nu}$ auf (2.58) an, so ergibt sich:

$$\begin{aligned} \partial^{\mu}\partial^{\nu}\partial_{\mu} X_{\nu} + \partial^{\mu}\partial^{\nu}\partial_{\nu} X_{\mu} &= 2\,\Box(\partial \cdot X) \\ = \frac{2}{m}\delta_{\mu\nu}\partial^{\mu}\partial^{\nu}(\partial \cdot X) &= \frac{2}{m}\Box(\partial \cdot X) \end{aligned} \tag{2.60}$$

Für $m \neq 2$ ist also $\Box(\partial \cdot X) = 0$. Mit (2.59) folgt $\partial_{\mu}\partial_{\nu}(\partial \cdot X) = 0$. Daher ist mit (2.58) allgemein:

$$\begin{aligned} \partial_{\mu}\partial_{\nu}\partial_{\alpha} X_{\beta} &= -\partial_{\mu}\partial_{\nu}\partial_{\beta} X_{\alpha} \\ \partial_{\mu}\partial_{\nu}\partial_{\alpha} X_{\beta} &= -\partial_{\beta}\partial_{\nu}\partial_{\alpha} X_{\mu} \\ &= \partial_{\mu}\partial_{\nu}\partial_{\beta} X_{\alpha}, \end{aligned} \tag{2.61}$$

so daß X höchstens von zweiter Ordnung in den x^μ sein kann; es erzeugt dann [57]:

$$
\begin{array}{rcll}
X^\mu &=& a^\mu & \text{Translationen,} \\
X^\mu &=& \omega^\mu{}_\nu x^\nu & \text{Lorentz–Transformationen,} \\
X^\mu &=& \lambda x^\mu & \text{Dilatationen und} \\
X^\mu &=& b^\mu x^2 - 2x^\mu b_\nu x^\nu & \text{spez. konforme Transformationen.}
\end{array}
$$

In $m = 2$ Dimensionen hingegen lautet (2.58):

$$\partial_1 X_1 = \partial_2 X_2 \qquad \partial_1 X_2 = -\partial_2 X_1. \tag{2.62}$$

Faßt man X_1 und X_2 wieder als Real– und Imaginärteil des Elements eines komplexifizierten Tangentialraumes auf, so entspricht (2.62) gerade den Cauchy–Riemannschen Differentialgleichungen der komplexen Analysis. Es erzeugen dann alle holomorphen Abbildungen in zwei Dimensionen zulässige konforme Transformationen, oder: es tragen hier *alle* Ordnungen in z bei. Die Lie–Algebra dieser Transformationen, die sog. *Witt–Algebra*, hat dann eine Basis aus Vektorfeldern

$$l_n = -z^{n+1}\partial_z \qquad \bar{l}_n = -\bar{z}^{n+1}\partial_{\bar{z}} \tag{2.63}$$

die den Vertauschungsrelationen

$$
\begin{array}{rcl}
[l_n, l_m] &=& (n-m)l_{m+n} \\
[\bar{l}_n, \bar{l}_m] &=& (n-m)\bar{l}_{m+n} \\
[l_n, \bar{l}_m] &=& 0
\end{array}
\tag{2.64}
$$

gehorcht. Die Algebra zerfällt in unabhängige holomorphe und antiholomorphe Anteile, die jedoch dieselbe Information enthalten; man kann sich daher auf die Betrachtung des holomorphen Teils beschränken. Da die l_n für $n < -1$ im Ursprung $z = 0$ singulär sind, hat man nur für $n \geq -1$ Vektorfelder im strengen Sinne: die Laurent–Reihe einer global holomorphen Funktion gemäß (2.47) enthält ja lediglich nichtnegative Potenzen von z. Kompaktifiziert man die komplexe Ebene zur Riemann–Sphäre $S^2 = \mathbb{C} \cup \{\infty\}$, so lassen sich diese Funktionen durch analytische Fortsetzung zumindest global definieren, was auch die „Singularität" der l_n für $n > 1$ im unendlich fernen Punkt $z = \infty$ beseitigt. Die Witt–Algebra läßt sich erst global definieren, wenn man die konformen Transformationen als *Meromorphismen* auf der Riemann–Sphäre auffaßt [58], d.h. Abbildungen f, bei denen an jeder Stelle entweder f oder $1/f$ holomorph ist, sofern man den unendlich fernen Punkt sowohl in die Definitions– als auch die Wertemenge mit einbezieht und festlegt, wie am Punkt $z = \infty$ differenziert werden soll. Im Sinne von invertierbaren Holomorphismen (Biholomorphismen) generieren lediglich die Operatoren l_{-1}, l_0 und l_1 global wohldefinierte Transformationen, nämlich gerade die projektiven Transformationen, aus denen die konforme Gruppe in höheren Raumdimensionen besteht, d.h. Translationen, Rotationen (bzw. Lorentz–Transformationen), Dilatationen und spezielle konforme Transformationen. Der Überhang der erzeugenden Algebra in einem zweidimensionalen euklidischen Raum gegenüber der in höherdimensionalen Räumen besteht also gerade aus den singulären, nicht global definierten Transformationen. Die Witt–Algebra ist dann im strengen Sinne nicht zu einer Gruppe global definierter Transformationen integrabel. Es ist jedoch der vollständige Satz lokaler Transformationen gemäß (2.63), der die Symmetrieeigenschaften eines konform invarianten Systems bestimmt, sofern (nach Voraussetzung) die Wechselwirkungen lokal sind; dann wird die Frage der Integrabilität zweitrangig. Man untersucht ja die Transformationseigenschaften der Skalenoperatoren bzw. –Felder, die *lokal* definiert sind.

Aufgrund der lokalen Eigenschaften der konformen Transformationen sollte die Kovarianz der Korrelationsfunktionen (2.31) erhalten bleiben, wenn man *lokale* Skalenfaktoren gemäß

$$b(\mathbf{r})^d = \det\left(\frac{\partial \mathbf{r}}{\partial \mathbf{r}'}\right) \tag{2.65}$$

verwendet, d.h.:

$$\langle \phi_1(\mathbf{r}_1) \cdots \phi_n(\mathbf{r}_n) \rangle = b(\mathbf{r}_1)^{x_1} \cdots b(\mathbf{r}_n)^{x_n} \langle \phi_1(\mathbf{r}'_1) \cdots \phi_n(\mathbf{r}'_n) \rangle. \tag{2.66}$$

Aufgrund der singulären Eigenschaften der Witt–Algebra gilt dies jedoch — wie sich herausstellt [6] — nicht in jedem Falle. Betrachtet man keine Meromorphismen, so hat man es ja sogar mit thermodynamischen Mittelungen bzgl. verschiedener Geometrien zu tun. Die Form (2.66) ist daher nicht für alle (skalaren) Operatoren ϕ_i und alle konformen Transformationen gültig. Man klassifiziert daher die Operatoren nach ihren Transformationseigenschaften; die sog. *quasi–primären* Operatoren sind kovariant lediglich bzgl. *projektiven* konformen Transformationen, die in der konformen Gruppe jeder Dimension enthalten sind und die Geometrie des Systems nicht ändern. Die sog. *primären* Operatoren sind bezüglich *jeder* konformen Transformation kovariant, insbesondere den finiten konformen Transformationen — entsprechend den Generatoren l_n mit $n < -1$ bzw. $n > 1$ —, die im allgemeinen die Geometrie eines Systems verzerren können. Diese Transformationen heißen finit, da ja nach dem

Satz von Liouville in der ganzen komplexen Ebene beschränkte analytische Funktionen konstant sind; nichttriviale Transformationen werden daher Singularitäten haben und nur in einem Teilbereich von $\mathbb{C}$ beschränkt sein. Die übrigen Operatoren heißen *sekundär*. Die spezielle Klasse der primären Operatoren besitzt jedoch nur in zwei Dimensionen nichttriviale Elemente, so daß für $d \geq 3$ Dimensionen die im allgemeinen unendlich große Menge der quasi–primären Operatoren die fundamentalen Operatoren enthält; in zwei Dimensionen reicht es dagegen aus, die in vielen Fällen endliche Klasse der primären Operatoren zu betrachten, denn aus deren Eigenschaften lassen sich, wie man zeigen kann, die Eigenschaften *aller* (primärer und sekundärer) Operatoren eines Systems ableiten [24]. Im Falle des Ising–Modells stellt sich heraus, daß der Gehalt an primären Operatoren sehr klein ist: danach sind nämlich die lokale Magnetisierung $\mathbf{m}(i)$ und die Energiedichte $\epsilon(i)$ die einzigen nichttrivialen primären Operatoren [57]; in diesem Sinne sind die beiden Größen (zunächst für das zweidimensionale Ising–Modell) ausgezeichnet und wurden daher oben als primäre Observablen eingeführt.

Die Analyse eines Modells der statistischen Mechanik mit Methoden der konformen Feldtheorie besteht daher zunächst in der Bestimmung seines Gehaltes an Skalenoperatoren (*operator content*) und der Klassifikation der Operatoren nach ihren Transformationseigenschaften. Dies reicht aus, um die Zweipunkt–Funktionen primärer Operatoren in verschiedenen Geometrien exakt, d.h. Skalenrelationen *inklusive der Amplituden* zu bestimmen. Die Darstellungstheorie der sog. *Virasoro–Algebra*, die die Erzeugenden der zu einer konformen Transformation des Raumes gehörenden Transformationen der Skalenoperatoren enthält, führt in zwei Dimensionen schließlich zu exakten Ergebnissen auch für die Vier–Punkt–Korrelationsfunktionen, die uns aber nicht weiter interessieren. Das weitere Vorgehen dabei ist durch die universelle Analogie zwischen statistischer und Quantenmechanik vorgezeichnet. Die Extraktion des Informationsgehaltes des Hamilton–Operators etwa in der Teilchenphysik geschieht mit gruppentheoretischen Methoden: danach ist die Bestimmung der maximalen Symmetriegruppe des Systems im wesentlichen äquivalent zur Berechnung des Spektrums der Hamilton–Funktion, da nach einem Korollar des Schurschen Lemmas Eigenräume von H mit irreduziblen Teilräumen einer Darstellung der Symmetriegruppe zusammenfallen. Die Bestimmung aller irreduziblen Darstellungen und den zugehörigen Dimensionen der Symmetriegruppe erlaubt daher eine vollständige Klassifikation der in der Theorie enthaltenen Teilchen. Mit Blick auf ein System der statistischen Mechanik, dessen Hamilton–Funktion (zumindest am

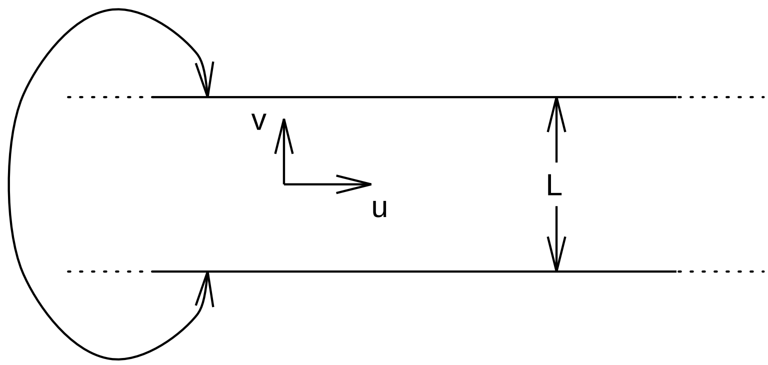


Abb. 2.2: Die logarithmische Abbildung (2.67) rollt die komplexe Ebene $\mathbb{C}$ auf einen unendlich langen Zylinder bzw. einen Streifen mit periodischen Randbedingungen auf.

kritischen Punkt) unter konformen Transformationen invariant ist, ist es dann die Darstellungstheorie der konformen Gruppe bzw. der sie erzeugenden Lie–Algebra, der Virasoro–Algebra, die den Gehalt des Systems liefert. Dem Energiespektrum des quantenmechanischen Hamiltonians entsprechen in der statistischen Mechanik gerade die Korrelationslängen der primären, also konform kovarianten, Operatoren und die Skalendimensionen den Gewichten der Darstellungen, die in der sog. *Kac–Tabelle* zusammengefaßt werden.

Am Beispiel einer speziellen Topologie, der sog. *Streifengeometrie*, soll nun die Anwendung der bisherigen Überlegungen demonstriert werden. Gegeben sei ein primärer Operator ϕ eines zweidimensionalen Systems mit Skalendimension x. Wir wollen den Effekt der *logarithmischen Transformation*

$$w = \frac{L}{2\pi} \ln z, \qquad z \in \mathbb{C} \tag{2.67}$$

auf die Zweipunkt–Korrelationsfunktion von ϕ untersuchen [15]. Diese rollt die komplexe Ebene $\{z\}$ auf einen unendlich langen Zylinder $S^1 \times \mathbb{R}$ vom Umfang L auf; ist $w = u + iv$, so mißt v den Polarwinkel entlang S^1 und u die longitudinale Richtung entlang der Zylinderachse $\mathbb{R}$, vgl. Abb. 2.2. Die Periodizität in der v–Richtung impliziert dabei *periodische Randbedingungen* in der transversalen Richtung, d.h.:

$$\phi(u, v + L) = \phi(u, v). \tag{2.68}$$

Wie oben gesehen, vgl. (2.38), ließ sich die Zweipunkt–Korrelationsfunktion in der Ebene schon allein durch Renormierungsgruppen–Argumente festlegen:

$$\langle \phi_1(z_1, \bar{z}_1) \phi_2(z_2, \bar{z}_2) \rangle = \frac{1}{(z_1 - z_2)^x (\bar{z}_1 - \bar{z}_2)^x}. \tag{2.69}$$

Mit der Umkehrung von (2.67) und der Kovarianzgleichung (2.66) ergibt sich dann für die Zylindergeometrie:

$$\begin{aligned}
\langle\phi(w_1,\bar{w}_1)\phi(w_2,\bar{w}_2)\rangle &= \left(\frac{2\pi}{L}\right)^{2x}\left(\frac{|z_1z_2|}{|z_1-z_2|^2}\right)^{x} \\
&= \left(\frac{2\pi}{L}\right)^{2x}\exp\left(\frac{2\pi}{L}(u_1+u_2)x\right)\times \\
&\times\left(e^{\frac{2\pi}{L}2u_1}+e^{\frac{2\pi}{L}2u_2}-e^{\frac{2\pi}{L}(u_1+u_2+iv_1-iv_2)}-e^{\frac{2\pi}{L}(u_1+u_2-iv_1+iv_2)}\right)^{-x} \\
&= \left(\frac{2\pi}{L}\right)^{2x}\left(2\cosh\frac{2\pi}{L}(u_1-u_2)-2\cos\frac{2\pi}{L}(v_1-v_2)\right)^{-x}. \qquad (2.70)
\end{aligned}$$

Man erhält die exakte Form der Zweipunkt–Funktion in der Streifengeometrie[7]. Im Limes großer Abstände, $|u_1-u_2|\gg L$, fallen damit die longitudinalen Korrelationen ab wie

$$\langle\phi(u_1,v_1)\phi(u_2,v_2)\rangle \approx \left(\frac{2\pi}{L}\right)\exp\left(-\frac{2\pi}{L}(u_1-u_2)x\right). \qquad (2.71)$$

Ein Vergleich mit der Definition der Korrelationslänge gemäß (2.6) ergibt:

$$\xi_{\|} = \frac{L}{2\pi x}. \qquad (2.72)$$

Die longitudinalen Korrelationslängen primärer Operatoren ϕ_i eines Systems auf einer Zylindergeometrie skalieren — wie aus der Theorie des FSS zu erwarten — linear mit der endlichen Systemausdehnung L. Die eigentlich wertvolle Aussage dieses Ergebnisses liegt jedoch in der exakten Bestimmung der *Amplitude* des Skalenverhaltens; diese ist, bis auf eine vom betrachteten Skalenoperator unabhängige Konstante $1/2\pi$, durch den Kehrwert der *Skalendimension* x_i des Operators gegeben. Die Beschränkung auf die reine Skaleninvarianz im Konzept der Renormierung, auf dem die klassischen Methoden des FSS aufbauen, reicht nicht aus, um die Amplituden in den Skalengesetzen, also die die universellen Konstanten, zu bestimmen; daher ist es eine Folge der verstärkten Symmetrieforderung in der Theorie der konformen Invarianz, daß jetzt ein solches Ergebnis möglich wird. In diesem Sinne ist es berechtigt mit Henkel [24] davon zu sprechen, im Gegensatz zur Theorie des FSS, die eher *qualitative* Aussagen über den Einfluß von Endlichkeitseffekten auf das Verhalten charakteristischer Größen thermodynamischer Systeme mache, ermögliche erst die Methode der konformen Feldtheorie eine *quantitative* Beschreibung dieser Phänomene im strengen Sinne (zumindest für zweidimensionale Systeme). Die Skalendimensionen x_i lassen sich in einigen Fällen exakt bestimmen, über Beziehungen von der

[7] Das Attribut *exakt* kommt ihr allerdings nur im Rahmen der gemachten Annahmen zu, d.h. dem Kontinuumslimes. S.u. Kap. 5.3

Form (2.37) sind sie jedoch auch mit für numerische Rechnungen leicht zugänglichen Größen, den üblichen kritischen Exponenten, verbunden, so daß die Amplituden des Skalenverhaltens der Korrelationslängen mit großer Genauigkeit bestimmt werden können. In der Sprache des FSS läßt sich nun umgekehrt die Skalenfunktion in der entsprechenden Gleichung,

$$\xi_i = LS_i(C_1 t L^{y_t}, C_2 h L^{y_h}), \tag{2.73}$$

mit Hilfe von (2.72) im Grenzfall $t, h \to 0$ exakt bestimmen:

$$S_i(0,0) = \frac{1}{2\pi x_i}. \tag{2.74}$$

Systeme auf der Streifengeometrie mit *antiperiodischen* Randbedingungen

$$\phi(u, v + L) = -\phi(u, v) \tag{2.75}$$

lassen sich ebenfalls mit den Methoden der konformen Invarianz analysieren [16, 56]. Für die O(n)–Spin–Modelle mit NN–Wechselwirkungen entspricht die Antiperiodizität gerade dem Einfügen eines Streifens von antiferromagnetischen Bindungen entlang der infiniten Richtung. Dies läßt sich wiederum im Rahmen der Feldtheorie durch das Einführen zweier zusätzlicher Operatoren ausdrücken, so daß sich Korrelationsfunktionen in der antiperiodischen Geometrie durch solche in der periodischen darstellen lassen. Für die Korrelationslängen von Spin und Energiedichte ergibt sich aus diesem Vorgehen:

$$\begin{aligned} \xi_s &= \frac{4}{3\pi} L, \\ \xi_\epsilon &= \frac{1}{4\pi} L. \end{aligned} \tag{2.76}$$

Das Skalenverhalten ist von derselben Art wie für Systeme auf Streifen mit periodischen Randbedingungen, aber die Amplituden A_i gemäß $\xi_i = A_i L$ stehen in diesem Fall nicht mehr in einem eindeutigen, insbesondere linearen, Zusammenhang mit den Skalendimensionen x_i der zugehörigen Operatoren. Dementsprechend lassen sich die Amplituden auch nur beispielhaft für Spin und Energiedichte angeben, nicht jedoch eine allgemeine Beziehung entsprechend (2.72).

2.6 Vermutungen für dreidimensionale Systeme

Da die konforme Gruppe in drei Raumdimensionen im Gegensatz zum zweidimensionalen Fall endlichdimensional ist, wird hier konforme Invarianz zu einer erheblich schwächeren Restriktion für ein System der statistischen Mechanik. Es ist daher von großem Interesse, Antworten auf die Frage zu finden, zu welchen Aussagen auch die reduzierte Menge an Symmetrieoperatoren für dreidimensionale Systeme noch ausreicht bzw. wie sich die Methoden der konformen Feldtheorie für solche Systeme erweitern lassen. Denn wie wir gesehen haben, ist der darstellungstheoretische Ansatz in der Lage, im wesentlichen dieselben Informationen zu liefern wie eine algebraische Integration des Systems. Eine Erweiterung der konformen Feldtheorie in dieser Richtung wäre daher mit der „exakten“ Lösung dreidimensionaler Systeme der statistischen Mechanik (im Kontinuumslimes) gleichzusetzen. Cardy [17] hat einen Weg vorgeschlagen, die Transformation (2.67) auf höhere Raumdimensionen zu verallgemeinern. Wählt man im $\mathbb{R}^2$ Polarkoordinaten mit Wegelement

$$ds^2 = dr^2 + r^2 d\phi^2 \tag{2.77}$$

und transformiert gemäß

$$\begin{array}{ccc} r & \longmapsto & u = R \ln r \\ \phi & \longmapsto & \phi, \end{array} \tag{2.78}$$

so schreibt sich ds^2 in den neuen Koordinaten als

$$ds^2 = R^{-2} e^{2u/R} (du^2 + R^2 d\phi^2). \tag{2.79}$$

Der Term in Klammern auf der rechten Seite entspricht gerade der natürlichen Metrik auf der Oberfläche eines Zylinders vom Radius R. Die von (2.78) (bzw. der Umkehrung) auf dem Zylinder induzierte Metrik stimmt mithin bis auf einen konformen Faktor $R^{-2}e^{2u/R}$ mit der ursprünglichen Metrik auf diesem Raum überein, so daß (2.78) eine konforme Transformation beschreibt. Nach einer zum Vorgehen in 1.5. analogen Rechnung erhält man auf diesem Wege das Ergebnis (2.72) zurück in der Form

$$\xi_i = \frac{R}{x_i}. \tag{2.80}$$

Die für eine Verallgemeinerung entscheidende Eigenschaft der Transformation (2.78) ist, daß sie den Winkelanteil ϕ invariant läßt; versieht man daher einen d–dimensionalen Raum $\mathbb{R}^d$ mit der Metrik

$$ds^2 = dr^2 + r^2 d\Omega^2, \tag{2.81}$$

wo $d\Omega^2$ das Winkelelement auf der Sphäre S^{d-1} symbolisiert, und transformiert

$$\begin{array}{ccc} r & \longmapsto & u = R \ln r \\ \vec{\Omega} & \longmapsto & \vec{\Omega}, \end{array} \tag{2.82}$$

so ist dies infolge

$$ds^2 = R^{-2} e^{2u/R}(du^2 + R^2 d\Omega^2) \tag{2.83}$$

eine konforme Abbildung $\mathbb{R}^d \to S^{d-1} \times \mathbb{R}$, die das Skalenverhalten (2.80) ergibt. Während eine solche Abbildung im zweidimensionalen Falle durch analytische Fortsetzung und Kompaktifikation noch als Meromorphismus aufgefaßt werden konnte, handelt es sich jetzt klar nicht mehr um einen Automorphismus $M \to M$, sondern um eine konforme Abbildung zwischen verschiedenen Mannigfaltigkeiten M und N, denn in der erzeugenden Algebra der konformen Gruppe von $(\mathbb{R}^d, \delta)$ sind ja die singulären Transformationen, die (2.82) erzeugen, gar nicht enthalten. Es bleibt also unklar, ob und in welchem Sinne die angegebene Verallgemeinerung zulässig ist. Cardy [17] zeigt, daß sie zumindest für das sphärische Modell gültig ist; einer numerischen Überprüfungen der Behauptung stellt sich die Tatsache entgegen, daß die Räume $S^{d-1} \times \mathbb{R}$ für $d > 2$ gekrümmt sind und sich die sphärischen Anteile nur mit einer endlichen Zahl von Punkten in regulärer Weise triangulieren lassen. Simulationen für das Ising–Modell im Falle $d = 3$ [3], wo die möglichen regelmäßigen Triangulierungen der $S^{d-1} = S^2$ gerade den platonischen Körpern entsprechen, d.h. maximal 20 Eckpunkte aufweisen, ließen keine Entscheidung in dieser Frage zu. Eine analoge Beziehung für die numerisch besser zugänglichen Geometrien vom Typ $T^{d-1} \times \mathbb{R} = S^1 \times \cdots \times S^1 \times \mathbb{R}$ zu gewinnen, scheitert daran, daß diese Flächen nicht konform auf die flachen Räume $(\mathbb{R}^d, \delta)$ abgebildet werden können.

Wenn sich hier auch unter Verwendung globaler Transformationen Aussagen für höherdimensionale Systeme gewinnen lassen, so scheint jedoch klar, daß eine vollständige Klassifikation des Gehaltes an Skalenoperatoren, wie sie im zweidimensionalen Fall möglich war, mit den Methoden der konformen Feldtheorie auf dreidimensionalen Räumen nicht zu erreichen ist, denn sie bedarf der Darstellungstheorie der unendlichdimensionalen Virasoro–Algebra, die in höheren Dimensionen nicht zur Verfügung steht. Ansätze in dieser Richtung liefert die Idee einer *modularen* Invarianz der Zustandssumme thermodynamischer Systeme. In einem zweidimensionalen System auf dem Torus $S^1 \times S^1$ reicht die Forderung nach modularer Invarianz, d.h. in diesem Fall der Invarianz unter Vertauschung der beiden Sphären und die Existenz einer konformen Abbildung zwischen den Generatoren von Translationen auf der S^1 und Skalentransformationen im $\mathbb{R}^2$ aus, um die Skalendimensionen etwa des Ising–Modells mit großer Genauigkeit zu bestimmen [20]. Während man hier lediglich eine globale konforme Transformation benötigt, erfordert die Gewinnung derselben Aussage mit Hilfe ausschließlich der konformen Gruppe die gesamte Information der lokalen Algebra. Da solche globalen Transformationen auch für höherdimensionale Systeme verfügbar sind, kann man hoffen, hier ähnliche Aussagen treffen zu können; Cardy [20] hat gezeigt, daß auf der Geometrie $S^2 \times S^1$ modulare Invarianz erreicht werden kann, wenn man *antiperiodische* Randbedingungen entlang der S^1 wählt, die Berechnung der Skalendimensionen gelang jedoch (außer für freie Felder) noch nicht.

Nun hat Henkel [45, 46] für die für Simulationen geeignete Geometrie $S^1 \times S^1 \times \mathbb{R}$, die *nicht* durch eine konforme Transformation auf $(\mathbb{R}^3, \delta)$ abgebildet werden kann, mit Hilfe einer Transfermatrix–Rechnung für das Ising–Modell die Vermutung erhärten können, daß am kritischen Punkt eine lineare Beziehung zwischen den Korrelationslängen der Operatoren σ und ϵ und den zugehörigen Skalendimensionen gemäß

$$\xi_\sigma \equiv A_\sigma L = \frac{A}{x_\sigma} L \qquad \xi_\epsilon \equiv A_\epsilon L = \frac{A}{x_\epsilon} L, \tag{2.84}$$

d.h. analog zum zweidimensionalen System $S^1 \times \mathbb{R}$, nur dann vorliegt, wenn man statt periodischen *antiperiodische* Randbedingungen auf dem Torus $S^1 \times S^1$ annimmt. Insbesondere gilt dann für das Verhältnis der Korrelationslängen:

$$\frac{\xi_\sigma}{\xi_\epsilon} = \frac{x_\epsilon}{x_\sigma}. \tag{2.85}$$

Er erhält die Werte:

$$\begin{aligned} A_\sigma/A_\epsilon &= 3.62(7) \qquad \text{für period. RB} \\ A_\sigma/A_\epsilon &= 2.76(4) \qquad \text{für antiperiod. RB,} \end{aligned} \tag{2.86}$$

was zu vergleichen ist mit dem Verhältnis der Skalendimensionen:

$$x_\epsilon/x_\sigma = 2.7326(16). \tag{2.87}$$

Dieser Wert wurde gewonnen aus dem gewichteten Mittel der in Tab. 2.1 aufgeführten Ergebnisse für die Exponenten ν und γ und der mit Hilfe von (2.35) und (2.36) aus (2.37) folgenden Beziehung:

$$\frac{x_\epsilon}{x_\sigma} = \frac{(1-\alpha)/\nu}{\beta/\nu} = \frac{2(\nu d - 1)}{\nu d - \gamma}. \tag{2.88}$$

Sollte sich diese Vermutung bestätigen und insbesondere auch auf andere Systeme wie etwa die O(n)–Spin–Modelle höherer Ordnung ($n > 1$) übertragbar sein, wäre hierin ein Hinweis auf die Möglichkeit der Übertragung der Methoden der konformen Feldtheorie und modularen Invarianz auf höhere Raumdimensionen zu sehen. Der

Methode	ν	γ	Quelle
g–Entwicklung	0.6300(15)	1.241(2)	[41, 42]
ϵ–Entwicklung	0.6310(15)	1.2390(25)	[43]
RG FD–Entwicklung	0.6301(10)	1.2378(25)	[66]
HT–Reihe	0.631(4)	1.239(3)	[1]
HT–Reihe bcc	0.632(1)	1.2395(4)	[22]
HT–Reihe bcc	0.6300(15)	1.237(2)	[68]
HT–Reihe sc	0.6315(8)	1.2388(10)	[14]
HT–Reihe bcc	0.6308(5)	1.2384(6)	[14]
MC	0.6289(8)	1.239(7)	[34]
MC	0.6301(8)	1.237(2)	[10]
Gewichtetes Mittel	0.63055(28)	1.23905(30)	

Tab. 2.1: Literaturwerte für die kritischen Exponenten ν und γ des dreidimensionalen Ising–Modells und varianzgewichtete Mittelwerte.

tiefere Grund für die Gültigkeit von (2.84) gerade für antiperiodische Randbedingungen ist dabei noch weitgehend unklar; während der Einfluß der Randbedingungen auf Systeme in zwei Dimensionen eingehend untersucht worden ist [18], liegt ihr Effekt in höheren Dimensionen — bis auf das genannte Ergebnis Cardys für die Torusgeometrie $S^2 \times S^1$ — noch weitgehend im dunkeln.

Eine Monte–Carlo–Simulation von Weston [85] konnte das Ergebnis mit einer unabhängigen Methode reproduzieren, jedoch die Genauigkeit nicht verbessern; er erhält für das Amplitudenverhältnis den Wert $A_\sigma/A_\epsilon \approx 2.6$. Darüberhinaus bestimmt er erstmals die universelle Konstante A in (2.84), was mit Henkels Methode nicht möglich war, da dieser im sog. Hamiltonschen Limes arbeitet, der die Hamilton–Funktion nur bis auf eine Normierungskonstante festlegt. Westons Simulationen liefern:

$$A = 0.117(3). \tag{2.89}$$

Der die Genauigkeit von Westons Simulationen begrenzende Faktor, das *critical slowing down* der Metropolis–Dynamik von Systemen in der Nähe eines kontinuierlichen Phasenüberganges, läßt sich heute durch die Verwendung von Cluster–Algorithmen minimieren: man kann also hoffen, auf diesem Wege wesentlich genauere Ergebnisse zu erzielen und somit eine sichere Entscheidung über die Gültigkeit von Henkels Vermutung treffen zu können. An erster Stelle steht jedoch die Frage, ob das Verhalten

(2.84) für die O(n)–Spin–Modelle höherer Ordnung reproduzierbar ist und damit universell im Sinne der Theorie der Renormierungsgruppe, oder ob es sich vielleicht um einen numerischen „Zufall“ handelt, der die analytischen Untersuchungen in die falsche Richtung lenkt. Schließlich ist es für die allgemeine Klassifikation der O(n)–Spin–Modelle auch von Interesse, wie sich die Konstante A über verschiedenen Werte von n hinweg verhält.

Exakte Rechnungen sind andererseits für die Grenzfälle des sphärischen ($n = \infty$) und des Gaußmodells ($n = 0$) möglich. Nach einer ersten Untersuchung von Henkel [47, 48] gilt für das sphärische Modell auf der dreidimensionalen Geometrie $T^2 \times \mathbb{R}$ eine lineare Beziehung zwischen den Amplituden des FSS der Korrelationslängen von lokaler Magnetisierung und Energiedichte gemäß (2.84) sowohl für periodische als auch für antiperiodische Randbedingungen. Die universelle Konstante A ist dabei gegeben durch:

$$\begin{aligned} &A \approx 3.0239 && \text{für periodische RB,} \\ &A = \tfrac{1}{2\pi\sqrt{2}}\left(1 - \tfrac{25}{8\pi^2}\right)^{-1/2} \approx 0.1361 && \text{für antiperiod. RB.} \end{aligned} \tag{2.90}$$

Da beim sphärischen Modell der kritische Exponent α der spezifischen Wärme, der in die Berechnung der Skalendimension der Energiedichte gemäß (2.37) eingeht, auf beiden Seiten des kritischen Punktes jedoch unterschiedliche Werte annimmt [8], ist zunächst unklar, ob dieses Ergebnis auch für den Limes $n \to \infty$ des O(n)–Spin–Modells gültig bleibt. Für das Gauß–Modell ergibt sich eine Amplitude von:

$$A = \frac{1}{2\pi\sqrt{2}} \approx 0.1125. \tag{2.91}$$

Da das Gauß–Modell jedoch keine geordnete Tieftemperaturphase besitzt und dementsprechend keinen echten Phasenübergang zweiter Ordnung aufweist, ist dieses Ergebnis für Vergleiche nur von begrenztem Wert. Eine Divergenz der Korrelationslängen im thermodynamischen Limes tritt nur auf, wenn alle Moden als masselos angenommen werden. Eine Übersicht über die bisherigen Kenntnisse über die Verhältnisse des FSS der Korrelationslängen in zwei– und dreidimensionalen Systemen der Geometrie $S^1 \times \mathbb{R}$ bzw. $T^2 \times \mathbb{R}$ gibt Tab. 2.2.

*

Es ist eine Spezialität zweidimensionaler Räume, daß sich auf ihnen eine komplexe Struktur definieren läßt; etwas Ähnliches ist dann erst wieder in vier Dimensionen

		$n=0$ **Gauß**	$n=1$ **Ising**	$n=2$ **XY**	$n=3$ **Heisenberg**	$\cdots$	$n=\infty$ **sphärisch**
2D	period. RB	($\checkmark$)	$\checkmark$	$\checkmark$	kein Phasenübergang		
	Amplitude		$1/2\pi$	$1/2\pi$			
	antiperiod. RB	$\times$	$\times$	$\times$			
3D	period. RB	$\times$	$\times$ (?)	?	?	?	$\checkmark$
	antiperiod. RB	$\checkmark$	$\checkmark$ (?)	?	?	?	$\checkmark$
	Amplitude	0.1125	≈ 0.12	?	?	?	0.1361

Tab. 2.2: Gültigkeit einer linearen Beziehung zwischen den Amplituden des FSS der Korrelationslängen und den zugehörigen Skalendimensionen gemäß $A_i = A/x_i$ für die O(n)–Spin–Modelle in zwei und drei Dimensionen.

möglich, nicht aber für „realistische“ Systeme mit drei Raumdimensionen. Die aus dieser Eigenart resultierende Struktur zweidimensionaler Räume ermöglicht es, die Klasse der global definierten konformen Transformationen des Raumes auf sich selbst (Automorphismen), die zunächst aus den einfachen projektiven Transformationen (Poincaré–Transformationen, Dilatationen und spezielle konforme Transformationen) besteht, zur unendlichdimensionalen Gruppe aller analytischen Transformationen zu erweitern. Die Darstellungstheorie dieser Gruppe bzw. ihrer Lie–Algebra liefert den gesamten Inhalt eines Modells an Skalenoperatoren und darüberhinaus (durch Bestimmung aller irreduziblen Darstellungen der Gruppe) eine Klassifikation aller möglichen zweidimensionalen Feldtheorien. Im Vergleich mit der klassischen Theorie des FSS, wie sie sich mit Methoden der Renormierung gewinnen läßt, liefert die konforme Feldtheorie zusätzlich zu den Exponenten der Skalengesetze auch die dort auftretenden Amplituden, also geschlossene, analytische Lösungen; in Form eine Störungsreihe lassen sich auch alle Korrekturen (Amplituden und Exponenten) zum eigentlichen FSS–Limes bestimmen.

In drei Dimensionen entsprechen die nichttrivialen (finiten) konformen Transformationen dagegen Abbildungen zwischen verschiedenen Räumen mit im allgemeinen unterschiedlicher Topologie und Geometrie (d.h. metrischem Tensor). Von einer Invarianz des Systems unter diesen Abbildungen kann im strengen Sinne nicht mehr gesprochen werden, so daß das gesamte Kalkül zweidimensionaler konformer Feldtheorien nicht mehr zur Anwendung kommen kann. Es lassen sich lediglich Aussagen für Systeme auf verschiedenen, konform ähnlichen Geometrien miteinander in

Verbindung setzen, was exakte Aussagen für die Bild–Geometrie erlaubt, falls die Urbild–Geometrie so einfach ist, daß sich dort mit anderen Methoden analytische Lösungen angeben lassen; es ist jedoch in diesem Fall nicht mehr ganz klar, für *welche* Operatoren solche Aussagen Geltung beanspruchen können [21]. Die Bemühungen der Feldtheoretiker gehen daher im Falle dreidimensionaler Systeme zunächst darauf aus, eine Klasse von Automorphismen für solche Räume zu finden, die groß genug ist, um auch hier mit Hilfe der entwickelten mathematischen Werkzeuge exakte Ergebnisse liefern zu können. Auf diesem Wege hat sich gezeigt, daß im Falle zweidimensionaler Systeme möglicherweise schon die wesentlich kleinere Klasse der modularen Transformationen, die auch in höheren Dimensionen uneingeschränkt zur Verfügung steht, für wesentliche Aussagen über die betrachteten Systeme ausreicht, so daß (zumindest für bestimmte Geometrien) die Hoffnung berechtigt ist, mit Hilfe dieser Transformationen für dreidimensionale Systeme mit den Methoden der Feldtheorie Ähnliches leisten zu können wie im zweidimensionalen Fall.

Kapitel 3

Methoden und Hilfsmittel

Die Untersuchung eines Modells der statistischen Mechanik mit Hilfe einer Methode der Computersimulation gliedert sich der Natur der Sache nach in zwei große Phasen. Diese Gliederung lehnt sich direkt an die Vorgehensweise der experimentellen Wissenschaften an: der Auswahl und Implementation der Methode der Informationsgewinnung (hier der Wahl der Simulationsmethode, dort der Versuchsaufbau) und dem eigentlichen Experiment folgt der gleichwertige zweite Schritt der Auswertung des gesammelten Datenmaterials und der Interpretation der so aufbereiteten Ergebnisse. Was die Methode der Computersimulation angeht, sind auf beiden Feldern in den letzten Jahren ausgefeilte Techniken entwickelt worden, die in dieser Arbeit zur Anwendung kommen sollen. Die Probleme bei der Gewinnung einer hinreichenden Menge unabhängiger Datensätze („Messungen") wurden für bestimmte Systeme durch die Verwendung von Cluster–Algorithmen gelöst. Eine Bewertung der statistischen Genauigkeit der gewonnen Ergebnisse andererseits — insbesondere unter Berücksichtigung der verbleibenden Korrelationen zwischen den Datensätzen — gelingt mit Hilfe der Binning–Technik und sog. Resampling–Verfahren wie das *Jackknifing*.

3.1 Simulationstechniken

3.1.1 Das Monte–Carlo–Verfahren

Die Methode der Monte–Carlo–Simulation dient allgemein der numerischen Approximation von Integralen (vornehmlich) über hochdimensionale Hyperflächen: und gerade ein solches Problem stellt die Berechnung der Zustandssumme (2.2) dar, wobei der diskrete Fall einer bloßen Summe (wie er beim Ising–Modell auftritt) durch entsprechende Wahl des Integrationsmaßes als spezielles Integral betrachtet werden kann. Wir schreiben hier für diskrete und kontinuierliche Fälle symbolisch ein Summenzeichen $\sum$. Eine ähnliche Aufgabe ist auch die Berechnung thermodynamischer Mittelwerte, etwa:

$$\langle A\rangle = \frac{1}{Z}\sum_{\{\mathbf{s}_i\}} A(\{\mathbf{s}_i\})\exp(-\beta\mathcal{H}(\{\mathbf{s}_i\})). \tag{3.1}$$

Die übliche Verwendung eines regelmäßigen Gitters von Stützpunkten liefert hier mit wachsender Dimension des Integrationsraumes immer schlechtere Ergebnisse, da in diesem Grenzübergang der Anteil der im Inneren des Integrationsvolumens gelegenen Punkte verschwindet; im Limes liegen daher alle Punkte auf einer Menge vom Raummaß Null (der Oberfläche), so daß die Verteilung der Stützpunkte im Integrationsvolumen alles andere als gleichmäßig ist [9]. Stattdessen wählt man die Stützpunkte $\mathbf{s}_i$ (d.h. Konfigurationen des betrachteten Systems im Falle der Integrale (2.2) bzw. (3.1)) der Integration zufällig und zwar im einfachsten Falle so, daß die zugehörige Verteilungsfunktion der einer Gleichverteilung auf dem Integrationsvolumen entspricht (*Simple Sampling*).

Von der Warte eines allgemeinen Algorithmus zur Wahl der Stützstellen aus besehen, ist dieser zunächst lediglich dadurch charakterisiert, daß er durch einen stochastischen Prozeß realisiert wird. Um unter bestimmten Voraussetzungen Existenz und Eindeutigkeit einer stationären Verteilung garantieren zu können, verwendet man für diesen zugrundeliegenden stochastischen Prozeß eine Markov–Kette, also einen Prozeß, dessen „Gedächtnis“ nur einen Schritt weit zurückreicht, so daß die Übergangswahrscheinlichkeit $T(\{\mathbf{s}'_i\}|\{\mathbf{s}_i\})$ lediglich von der jetzigen und der erreichten Konfiguration abhängt. Voraussetzung für eine Konvergenz dieses Verfahrens gegen das zu bestimmende Integral im Limes einer unendlich langen Kette ist, daß der Prozeß tatsächlich jeden Punkt des Integrationsgebietes mit im Grenzfall nichtverschwindender Wahrscheinlichkeit erreicht, die sog. *Ergodizität*. Das Vorgehen beim

Simple Sampling liefert jedoch in vielen Fällen immer noch unbefriedigende Ergebnisse; im allgemeinen ist nämlich der exponentielle Integrand des Integrals, der Boltzmann–Faktor, nur in einem schmalen Bereich des Integrationsvolumens merklich von Null verschieden — die Stützstellen sind jedoch nach Voraussetzungen im gesamten Volumen gleichverteilt. Beim *Importance Sampling* versucht man daher, die Stützpunkte direkt gemäß dem zugehörigen Boltzmann–Gewicht zu bestimmen, d.h.: wo der Integrand größere Beiträge liefert, werden auch mehr Stützstellen gewählt. Die stationäre Verteilung $W(\{\mathbf{s}_i\})$ der Markov–Kette muß demnach jetzt gerade der Gleichgewichtsverteilung des Systems, bei Simulationen im kanonischen Ensemble also gerade der Boltzmann–Verteilung, entsprechen. Eine hinreichende Bedingung für das Erreichen der gewünschten stationären Verteilung liefert die *detaillierte Balance*:

$$T(\{\mathbf{s}_i'\}|\{\mathbf{s}_i\})W(\{\mathbf{s}_i\}) = T(\{\mathbf{s}_i\}|\{\mathbf{s}_i'\})W(\{\mathbf{s}_i'\}). \tag{3.2}$$

Sie besagt, daß sich der lokale Austausch von Wahrscheinlichkeits–„Masse“ W zwischen zwei Zuständen im Gleichgewicht befindet; *notwendig* ist lediglich die schwächere Forderung einer Art Kontinuitätsgleichung, nach der die gesamte aus einem Zustand in alle anderen Zustände abfließende Wahrscheinlichkeit die einströmende Wahrscheinlichkeit ausgleichen muß. Metropolis et. al. [64] haben gezeigt, daß eine hierfür geeignete Wahl der Übergangswahrscheinlichkeit gegeben ist durch:

$$T(\{\mathbf{s}_i'\}|\{\mathbf{s}_i\}) = \min[1, W(\{\mathbf{s}_i'\})/W(\{\mathbf{s}_i\})], \tag{3.3}$$

wobei sich im Falle der O(n)–Spin–Modelle Ausgangs– und Endkonfiguration lediglich durch die Änderung *eines* Freiheitsgrades $\mathbf{s}_i$ unterscheiden sollen, so daß der Update–Algorithmus *lokal* ist. W entspricht hier der in (3.1) verwendeten Boltzmann–Verteilung. Ein entsprechendes Vorgehen bei der Erzeugung der Integrationsstützpunkte wird als *Metropolis–Algorithmus* bezeichnet.

Die Genauigkeit der mit Hilfe dieser Vorschrift gewonnenen Approximation ist abhängig von der Anzahl der im Verlaufe der Simulation erfolgten *unabhängigen* Messungen der jeweils betrachteten Größe; ein Maß für diese Unabhängigkeit ist gegeben durch die (integrierte) Autokorrelationszeit τ (s.u. Kap. 3.2.1). Die Varianz $\sigma^2(\bar{A})$ des Mittelwertes $\bar{A}$ aus n Messungen in Approximation von (3.1) ist in diesem Fall von der Größe

$$\sigma^2(\bar{A}) = \frac{\sigma^2(A)}{n/2\tau}, \tag{3.4}$$

wobei $\sigma^2(A)$ die Varianz einer Einzelmessung darstellt. Mit anderen Worten: die Anzahl der unabhängigen Messungen ist um den Faktor $1/2\tau$ kleiner als die Zahl der erfolgten Messungen. Die Effizienz eines gegeben Monte–Carlo–Algorithmus wird daher nicht nur bestimmt durch die zur Erzeugung der Konfigurationen und zur Messung der gewünschten Größen erforderlichen Rechenoperationen, sondern ebenso durch die der Dynamik inhärente Autokorrelationszeit. Aus Simulationen mit Augenmerk auf die dynamischen Qualitäten des Monte–Carlo–Verfahrens weiß man, daß die Autokorrelationszeiten gemäß einem Potenzgesetz mit der zugehörigen räumlichen Korrelationslänge einer gegebenen Observable anwächst:

$$\tau \propto \xi^z, \tag{3.5}$$

wobei der dynamische kritische Exponent z für den lokalen Metropolis–Algorithmus ungefähr den Wert $z \approx 2$ annimmt. Da jedoch die Korrelationslängen lokaler Observablen bei der Annäherung an den Punkt eines Phasenüberganges zweiter Ordnung divergieren, wachsen auch die Autokorrelationszeiten unbegrenzt an, so daß man im Grenzfall unendlich lange warten muß, bis eine neue unabhängige Messung erfolgen kann (*Critical Slowing Down*). In einem endlichen System können zwar die Korrelationslängen nicht divergieren und ihre Rolle übernimmt jetzt die lineare endliche Systemausdehnung L, so daß die Autokorrelationszeiten gemäß $\tau \propto L^z$ anwachsen, aber die kritische Verlangsamung der Dynamik macht hier Simulationen in der Nähe des kritischen Punktes zumindest sehr rechenaufwendig und zwar um so mehr, je grösser das betrachtete System und damit ξ ist. Im Falle etwa des ferromagnetischen Ising–Modells ist der Grund für dieses Verhalten anschaulich klar: bei einer Annäherung an den Punkt des Phasenüberganges bilden sich Cluster einheitlicher Spinorientierung und wachsender Größe aus, so daß ein Großteil der Spins im Innern solcher Bereiche liegt. Eine neue unabhängige Konfiguration wäre etwa gegeben durch die Umkehrung der Orientierungsrichtung der Spins eines Clusters; versucht man dazu sukzessive, einzelne Spins des Clusters gegen die vorherrschende Orientierung umzukehren, so wird jeder dieser Schritte gemäß (3.3) nur mit einer Wahrscheinlichkeit $\exp(-\beta\,\Delta E) < 1$ ausgeführt, so daß der gesamte Übergang mit wachsender Clustergröße (also mit wachsender Systemgröße oder Annäherung an den kritischen Punkt) immer unwahrscheinlicher, die Dynamik immer langsamer wird.

3.1.2 Cluster–Algorithmen

Die bisher beste Lösung dieses Problems im Falle der O(n)–Spin–Modelle beruht auf der Beobachtung der Clusterbildung am Punkt eines Phasenübergang zweiter

Ordnung. Statt lokale Veränderungen der Konfiguration zu versuchen, wie sie der Metropolis–Algorithmus erzeugt, wird hier für Bereiche von Spins in der Größenordnung der Gebiete einheitlicher Spinorientierung, d.h. der (geometrischen) Cluster, global eine neue Spinorientierung gewählt. Wie oben zu sehen war, ist es aufgrund der Symmetrien der Wechselwirkung in der Hamiltonfunktion (2.1) der O(n)–Spin–Modelle nicht die absolute Konfiguration der Spins selbst, die die Energie und damit das Verhalten bei einem MC–Update bestimmt, als vielmehr (im feldfreien Fall) die relative Orienterung benachbarter Spins zueinander. Dieser Sachverhalt erlaubt es, das Spin–Gitter–Modell auf ein Modell abzubilden, in dem nicht die Spins, sondern *Bindungen* zwischen den Spins die elementaren Größen sind und zwar mittels der sog. *Fortuin–Kasteleyn–Transformation.* Der Weg von dieser Überlegung zu einem funktionierenden Cluster–Algorithmus sei hier zunächst für das Ising–Modell nachgezeichnet [38]. Setzt man zwischen allen Spins gleicher Orienterung Bindungen, erhält man eine Zerlegung des Gitters in die sog. geometrischen Cluster; wollte man diese Zerlegung für einen Update–Algorithmus verwenden, bliebe jedoch die Cluster–Aufteilung eine unveränderliche Größe, Ergodizität wäre nicht zu erreichen. Daher dürfen Bindungen zwischen Spins gleicher Orientierung nur mit einer Wahrscheinlichkeit $p < 1$ gesetzt werden. Um p zu bestimmen, kann man die Forderung der detaillierten Balance (3.2) ausnutzen. Die Konfiguration der Bindungen sei repräsentiert durch einen Satz von Bindungsvariablen n_{ij}, so daß

$$n_{ij} = \begin{cases} 1 & \text{für } \mathbf{s}_i = \mathbf{s}_j \\ 0 & \text{für } \mathbf{s}_i \neq \mathbf{s}_j \end{cases} . \tag{3.6}$$

Die Übergangswahrscheinlichkeiten $T(\{\mathbf{s}'_i\}|\{\mathbf{s}_i\})$ setzen sich jetzt zusammen aus der (von p abhängigen) Wahrscheinlichkeit $P_{\text{cluster}}(\{n_{ij}\}|\{\mathbf{s}_i\})$ für eine bestimmte Aufteilung des Gitters bei der gegebenen Konfiguration in (stochastische) Cluster entsprechend einer Bindungskonfiguration $\{n_{ij}\}$, der Wahrscheinlichkeit $P_{\text{flip}}(\{n_{ij}\})$ für eine bestimmte Auswahl der zu flippenden Cluster und der Akzeptanzwahrscheinlichkeit $P_{\text{akz}}(\{n_{ij}\})$ für den vorgeschlagenden Update–Versuch; die Bedingung der detaillierten Balance nimmt daher folgende Form an:

$$\begin{aligned} W(\{\mathbf{s}_i\})P_{\text{cluster}}(\{n_{ij}\}|\{\mathbf{s}_i\})P_{\text{flip}}(\{n_{ij}\})P_{\text{akz}}(\{n_{ij}\}) &= \\ W(\{\mathbf{s}'_i\})P_{\text{cluster}}(\{n'_{ij}\}|\{\mathbf{s}'_i\})P_{\text{flip}}(\{n'_{ij}\})P_{\text{akz}}(\{n'_{ij}\}) & \end{aligned} \tag{3.7}$$

Da die Umkehrung der Spinorientierung bestimmter Cluster die Zerlegung des Gitters selbst nicht verändert, ist dabei $n'_{ij} = n_{ij}$ und daher auch $P_{\text{flip}}(\{n'_{ij}\}) =$

$P_{\text{flip}}(\{n_{ij}\})$; um die Akzeptanzrate zu maximieren setzen wir außerdem $P_{\text{akz}} \equiv 1$. Im kanonischen Ensemble ist daher

$$\frac{P_{\text{cluster}}(\{n_{ij}\}|\{\mathbf{s}_i\})}{P_{\text{cluster}}(\{n'_{ij}\}|\{\mathbf{s}'_i\})} = \frac{W(\{\mathbf{s}'_i\})}{W(\{\mathbf{s}_i\})} = \exp\left(-\beta\left[E(\{\mathbf{s}'_i\}) - E(\{\mathbf{s}_i\})\right]\right). \tag{3.8}$$

Enthält die Konfiguration $\{\mathbf{s}_i\}$ N_p parallele und N_a antiparallele Spinpaare, so ist ihre Energie gegeben durch $E = -J(N_p - N_a)$; ändert sich daher die Konfiguration gemäß $N_p \mapsto N_p + \Delta$, $N_a \mapsto N_a - \Delta$, so entspricht das einer Energiedifferenz $\Delta E = -2J\Delta$. Die Wahrscheinlichkeit für die Aufteilung des Gitters in Cluster ihrerseits hängt von der Wahrscheinlichkeit p ab, zwischen parallelen Spins eine Bindung zu aktivieren; hat man n_c verbundene und $n_b = N_p - n_c$ unverbundene parallele Spinpaare, so ist $P_{\text{cluster}}(\{n_{ij}\}|\{\mathbf{s}_i\}) = p^{n_c}(1-p)^{n_b}$. Mit dem Update ändert sich nicht die Clusterstruktur, d.h. es ist $n'_c = n_c$, aber es kommen Δ parallele Spinpaare hinzu, zwischen denen keine Bindungen aktiviert sind, d.h. $n'_b = n_b + \Delta$. Insgesamt hat man daher:

$$\begin{aligned} \frac{P_{\text{cluster}}(\{n_{ij}\}|\{\mathbf{s}_i\})}{P_{\text{cluster}}(\{n_{ij}\}|\{\mathbf{s}'_i\})} &= \frac{p^{n_c}(1-p)^{n_b}}{p^{n_c}(1-p)^{n_b+\Delta}} = (1-p)^{-\Delta} \\ \frac{W(\{\mathbf{s}'_i\})}{W(\{\mathbf{s}_i\})} &= \exp(2K\Delta), \end{aligned} \tag{3.9}$$

wo $K = \beta J$ die Koppelungskonstante ist. Daraus folgt schließlich:

$$p = 1 - \exp(-2K). \tag{3.10}$$

Der hiermit konstruierte *Swendsen–Wang–Cluster–Algorithmus* [82] setzt sich also aus folgenden Schritten zusammen:

1. Setze zwischen parallelen Spins des Gitters mit der Wahrscheinlichkeit $p = 1 - \exp(-2K)$ eine aktive Bindung.

2. Identifiziere die durch die aktiven Bindungen definierten zusammenhängenden Gebiete des Gitters als (stochastische) Cluster.

3. Wähle für jeden so konstruierten Cluster zufällig eine neue einheitliche Spin–Orientierung, im einfachsten Falle für $s = \pm 1$ je mit der Wahrscheinlichkeit $1/2$.

Daß es sich bei diesem Verfahren für das Ising–Modell um einen zulässigen Monte–Carlo–Update–Algorithmus handelt, ist klar: die detaillierte Balance ist qua Herleitung erfüllt; Ergodizität ergibt sich aus der Tatsache, daß die Wahrscheinlichkeit für die Orientierungsumkehr eines einzelnen Spins immer endlich ist, so daß jede beliebige Konfiguration in endlich vielen Schritten erreicht werden kann.

Wolff [86] hat eine Modifikation dieses Vorgehens vorgeschlagen, die die Effizienz des Algorithmus weiter verbessert: anstatt das gesamte Gitter in Cluster zu zerlegen und jeden davon mit der Wahrscheinlichkeit 1/2 zu flippen, kann man auch, ausgehend von einer zufällig gewählten Gitterposition, mit der Bindungs–Wahrscheinlichkeit (3.10) lediglich *einen* Cluster aufbauen, dessen Spinorientierung dann in jedem Falle invertiert wird. Dies läßt sich auch als spezielle Wahl der Wahrscheinlichkeit P_{flip} im Rahmen des Swendsen–Wang–Algorithmus interpretieren; der *Wolff–Algorithmus* ist sowohl programmtechnisch einfacher zu realisieren, als auch (zumindest für dreidimensionale Systeme) effizienter im Sinne der benötigten Zeit zur Erzeugung einer neuen unabhängigen Konfiguration und soll daher auch im folgenden vornehmlich zur Anwendung kommen. Der Grund für die größere Effizienz des Wolff–Algorithmus liegt in einer *durchschnittlich* größeren Ausdehnung der Cluster. Beide Cluster–Algorithmen reduzieren die Autokorrelationszeiten typischer Observablen drastisch. Die dynamischen kritischen Exponenten des zweidimensionalen Ising–Modells reduzieren sich im Falle des Swendsen–Wang–Algorithmus' (je nach Observable) auf $z \approx 0.20$-0.35 [53] und für den Wolff–Algorithmus auf $z \approx 0.13$-0.26; in drei Dimensionen schneidet dieser mit $z \approx 0.14$-0.30 gegenüber jenem mit $z \approx 0.50$-0.75 noch etwas besser ab, zumal die Amplituden im Skalengesetz (3.5) für den Wolff–Algorithmus im allgemeinen kleiner sind.

Eine Anwendung der Cluster–Algorithmen auf die O(n)–Spin–Modelle höherer Ordnung [86] erfordert die Isolation von isingartigen Freiheitsgraden in diesen Systemen, um das Konzept von parallelen und antiparallelen Spins übertragen zu können. Man zerlegt dazu die Spins $\mathbf{s}_i$ in Anteile parallel und senkrecht zu einer zufällig gewählten Richtung $\mathbf{r}$:

$$\mathbf{s}_i = \mathbf{s}_i^{\|} + \mathbf{s}_i^{\perp}, \qquad \mathbf{s}_i^{\|} = \epsilon_i |\mathbf{s}_i \cdot \mathbf{r}| \mathbf{r}, \qquad \epsilon_i = \operatorname{sign}(\mathbf{s}_i \cdot \mathbf{r}). \tag{3.11}$$

Einsetzen in die Hamiltonfunktion (2.1) des Systems ergibt:

$$\mathcal{H} = -\sum_{<ij>} J_{ij} \epsilon_i \epsilon_j + \text{const.}, \tag{3.12}$$

wobei

$$J_{ij} = J |\mathbf{s}_i \cdot \mathbf{r}| |\mathbf{s}_j \cdot \mathbf{r}|, \tag{3.13}$$

was dem Hamiltonian eines Ising–Modells mit zufälligen, positiven Wechselwirkungskonstanten J_{ij} entspricht. Der Swendsen–Wang–Algorithmus für die höheren $\mathrm{O}(n)$–Spin–Modelle sieht daher so aus:

1. Wähle (gemäß einer Gleichverteilung) eine zufällige Richtung $\mathbf{r}$.
2. Setze zwischen Spinpaaren $(\mathbf{s}_i, \mathbf{s}_j)$ mit $\epsilon_i \epsilon_j > 0$ mit der Wahrscheinlichkeit p gemäß (3.10) eine aktive Bindung.
3. Zerlege das Gitter anhand der Bindungsstruktur in Cluster.
4. Entscheide mit einer Wahrscheinlichkeit von $1/2$ für jeden Cluster, ob er geflippt wird. In den Clustern, die invertiert werden, spiegele alle Spins an der Ebene senkrecht zur Richtung $\mathbf{r}$.

Analog dazu wird der Wolff–Algorithmus implementiert. Es zeigt sich, daß beide Cluster–Algorithmen auch in diesem verallgemeinerten Falle bei Simulationen mit kritischer Verlangsamung wesentlich effektiver sind als der Metropolis–Algorithmus. In Abb. 3.1 ist zu erkennen, daß hier der Effizienzunterschied zwischen Swendsen–Wang– und Wolff–Algorithmus sogar noch größer ist als für das Ising–Modell.

3.2 Moderne Verfahren der Datenanalyse

Bei der Analyse von Zeitreihen aus Monte–Carlo–Simulationen ist man mit zwei großen Problembereichen konfrontiert. Sowohl für Metropolis–Daten, als auch in geringerem Maße für Messungen aus Cluster–Update–Simulationen, wird die Zeitreihe Autokorrelationen aufweisen, insofern τ erst im Limes unendlich seltener Messungen verschwindet. Da das Ausmaß der Autokorrelation einer Meßreihe apriori unbekannt ist, wird eine zuverlässige Schätzung des Meßfehlers problematisch, denn eine autokorrelierte Zeitreihe gaukelt kleinere Schwankungen vor, als die zu messende Größe sie aufweist. Eine korrekte Fehlerrechung ist möglich mit Hilfe der *Binning*–Technik, die die autokorrelative Verminderung der Varianzschätzung zu korrigieren vermag, indem sie eine nahezu unkorrelierte Ersatzreihe liefert. Selbst wenn zeitlich unkorrelierte Daten vorliegen, verbleibt das Problem, die Fehler von Größen zu bestimmen, die in der Simulation nicht direkt gemessen wurden, sondern nichtlineare Kombinationen der Observablen darstellen; in diesem Fall ist eine korrekte Fehlerbehandlung mit Hilfe der *Jackknife*–Methode möglich.

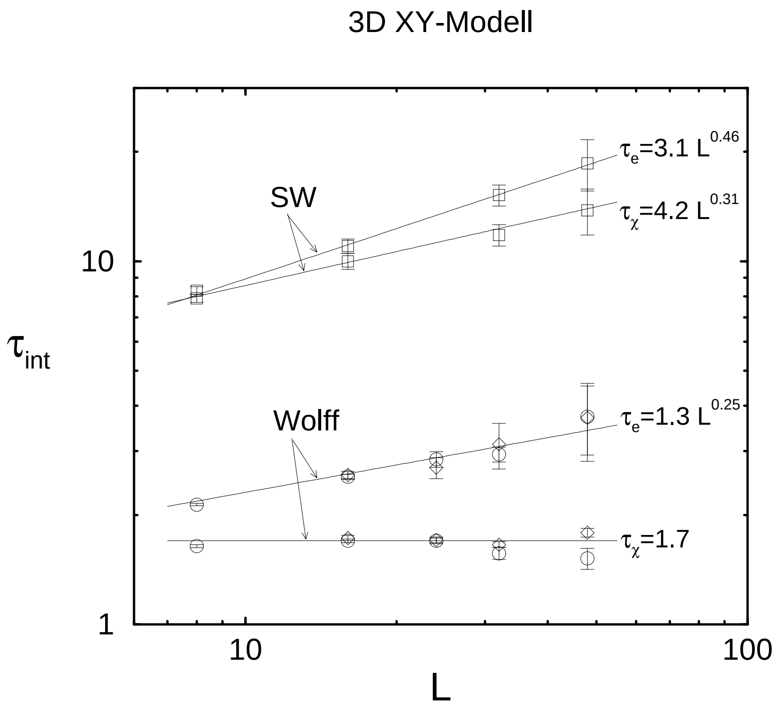


Abb. 3.1: Integrierte Autokorrelationszeiten des dreidimensionalen XY–Modells aus Simulationen mit dem Swendsen–Wang–Algorithmus (SW) im Vergleich zu solchen mit dem Wolff–Algorithmus (Wolff) in einer doppelt logarithmischen Auftragung. Gemessen wurden jeweils die magnetische Suszeptibilität (τ_χ) und die spezifische Wärme (τ_e). Die Abbildung ist der Arbeit [53] entnommen.

3.2.1 Binning

Zur näherungsweisen Bestimmung des Erwartungswertes $\langle A \rangle$ einer Observable A aus einer endlichen Meßreihe der Länge n betrachtet man den Mittelwert $\bar{A} = \sum_t A_t$, der einen erwartungstreuen Schätzer für $\langle A \rangle$ darstellt. Um eine Angabe über die Genauigkeit der Schätzung machen zu können, interessiert man sich für die Varianz dieser Größe:

$$\sigma^2(\bar{A}) = \langle \bar{A}^2 \rangle - \langle \bar{A} \rangle^2. \tag{3.14}$$

Einsetzen der Definition von $\bar{A}$ ergibt:

$$\sigma^2(\bar{A}) = \frac{1}{n^2} \sum_{s,t=1}^{n} \gamma_{s,t} = \frac{1}{n} \left[\gamma_0 + 2 \sum_{\Delta t=1}^{n-1} \left(1 - \frac{\Delta t}{n} \right) \gamma_{\Delta t} \right], \tag{3.15}$$

mit

$$\gamma_{s,t} \equiv \langle A_s A_t \rangle - \langle A_s \rangle \langle A_t \rangle, \qquad \gamma_{\Delta t} \equiv \gamma_{s,t}, \qquad \Delta t = |s - t|, \tag{3.16}$$

wobei die Invarianz der Reihe unter Zeittranslation ausgenutzt wurde. Definiert man nun die *integrierte Autokorrelationszeit*

$$\tau_{\rm int} \equiv \frac{1}{2} + \sum_{\Delta t=1}^{n-1} \frac{\gamma_{\Delta t}}{\gamma_0}\left(1 - \frac{\Delta t}{n}\right), \tag{3.17}$$

so erhält man mit $\gamma_0 = \sigma^2(A)$ gerade wieder die Beziehung (3.4). Für eine verläßliche Bestimmung von $\sigma^2(\bar{A})$ benötigt man daher einen Schätzer für $\gamma_{\Delta t}$ oder zumindest für $\tau_{\rm int}$. Der naheliegende Ansatz

$$c_{\Delta t} = \frac{1}{n - \Delta t} \sum_{t=1}^{n-\Delta t} (A_t - \bar{A})(A_{t+\Delta t} - \bar{A}) \tag{3.18}$$

liefert jedoch keine erwartungstreue Schätzfunktion, sondern weist einen *Bias* auf. Dieser läßt sich zwar für Korrelationen von endlicher Reichweite durch eine rekursive Prozedur näherungsweise beseitigen, aber es stellt sich heraus, daß der dazu nötige Rechenaufwand von der Größenordnung $O(nT_{\rm max})$ ist, wenn alle Korrelationen nach der Zeit $T_{\rm max}$ abgeklungen sind [37]; da in der Regel die Korrelationen nur im Rahmen einer gewünschten Genauigkeit von endlicher Reichweite sind, führt man durch diese Prozedur zusätzlich einen willkürlichen Abschneideradius ein.

Die Idee des *Binning* ist es nun, benachbarte Daten in Blöcke oder *Bins* zusammenzufassen, d.h. neue Variablen gemäß

$$A'_t = \frac{1}{2}(A_{2t-1} + A_{2t}), \qquad n' = \frac{1}{2}n \tag{3.19}$$

einzuführen. Der Mittelwert $\bar{A}$ und die Varianz $\sigma^2(\bar{A})$ bleiben von dieser Transformation unberührt; die Autokorrelationsfunktion ändert sich gemäß

$$\gamma'_{\Delta t} = \begin{cases} \frac{1}{2}\gamma_0 + \frac{1}{2}\gamma_1 & \text{für } \Delta t = 0 \\ \frac{1}{4}\gamma_{2\Delta t-1} + \frac{1}{2}\gamma_{2\Delta t} + \frac{1}{4}\gamma_{2\Delta t+1} & \text{für } \Delta t > 0 \end{cases}. \tag{3.20}$$

Der Effekt der Blocktransformation zeigt sich, wenn man ihre Wirkung auf die untere Schranke γ_0/n von $\sigma^2(\bar{A})$ betrachtet (vgl. (3.15)):

$$\sigma^2(\bar{A}) = \sigma^2(\bar{A}') \geq \frac{\gamma'_0}{n'} = \frac{\gamma_0}{n} + \frac{\gamma_1}{n}. \tag{3.21}$$

Die Werte für γ_0/n steigen also mit jeder Anwendung der Transformation — ausgehend von korrelierten Daten, d.h. von $\gamma_1 > 0$ — an, es sei denn $\gamma_1 = 0$. Man kann zeigen [37], daß $\gamma^*_{\Delta t}/n \propto \delta_{\Delta t\,0}$ einen attraktiven Fixpunkt der Block–Transformation darstellt, in dessen Einzugsbereich alle Ausgangskorrelationen $\gamma_{\Delta t}$ fallen, die schneller als $1/\Delta t$ nach außen hin abfallen. Die Transformation (3.19) vernichtet daher im Limes alle Autokorrelationen der Zeitreihe; anschaulich ist klar: wenn die Korrelationen schnell genug abfallen, werden zwei benachbarte Werte der transformierten Reihe schwächer korreliert sein als zwei Werte der ursprünglichen Reihe, da der „durchschnittliche" Abstand der in den transformierten Werten enthaltenen Elemente der Ausgangsreihe größer ist als Eins. Wie aus (3.20) ersichtlich ist, verdoppeln sich bis auf Korrekturen gerade die für die Korrelationen relevanten Abstände. Durch die Block–Transformation werden mehr und mehr der maßgeblichen Korrelationen auf kurze Distanzen in die (transformierten) Werte selbst absorbiert und erscheinen dort als Korrelationen der Werte mit sich selbst, d.h. in deren anwachsender Varianz.

Am Fixpunkt selbst läßt sich die Varianz $\sigma^2(\bar{A})$ aufgrund der fehlenden Korrelationen mit der naiven Methode bestimmen, d.h. mit dem nun erwartungstreuen Schätzer

$$\sigma^2(\bar{A}) = \langle \frac{1}{n^* - 1} c_0^* \rangle = \langle \frac{1}{n^*(n^* - 1)} \sum_{t=1}^{n^*} (A_t^* - \bar{A})^2 \rangle. \tag{3.22}$$

Die Bestimmung der Varianz des Mittelwertes einer autokorrelierten Zeitreihe mit der Methode des Binning nimmt daher folgende Form an:

1. Bestimme aus der ursprünglichen Zeitreihe $c_0/(n-1)$ als Schätzer für $\sigma^2(\bar{A})$.
2. Wende die Block–Transformation (3.19) auf die A_t an und bestimme $c_0'/(n'-1)$. Bis auf statistische Schwankungen gilt: $c_0'/(n'-1) \geq c_0/(n-1)$.
3. Iteriere diesen Prozeß so lange, bis die Werte $c_0^{(k)}/(n^{(k)} - 1)$ innerhalb der gewünschten Genauigkeit nicht mehr weiter wachsen, d.h. das System den Fixpunkt näherungsweise erreicht hat.
4. Dieser Plateau–Wert $c_0^*/(n^* - 1)$ entspricht der Binning–Schätzung für $\sigma^2(\bar{A})$.

Der Rechenaufwand für diesen Prozeß ist mit $O(2n)$ gegenüber $O(nT_{\max})$ für ein erwünschtes $T_{\max} \gg \tau$ in der meisten Fällen wesentlich geringer. Dabei ist natürlich die Schätzung $c_0^*/(n^* - 1)$ selbst wieder eine Zufallsvariable und daher mit einem Fehler behaftet. Da am Fixpunkt die A_t' unabhängige und aufgrund des zentralen Grenzwertsatzes annähernd gaußverteilte Zufallsgrößen sind, läßt sich die Varianz dieser Schätzung jedoch näherungsweise angeben; man erhält:

$$\sigma^2(c_0^*/(n^* - 1)) \approx \sqrt{\frac{2}{n^* - 1}} \frac{c_0^*}{n^* - 1}. \tag{3.23}$$

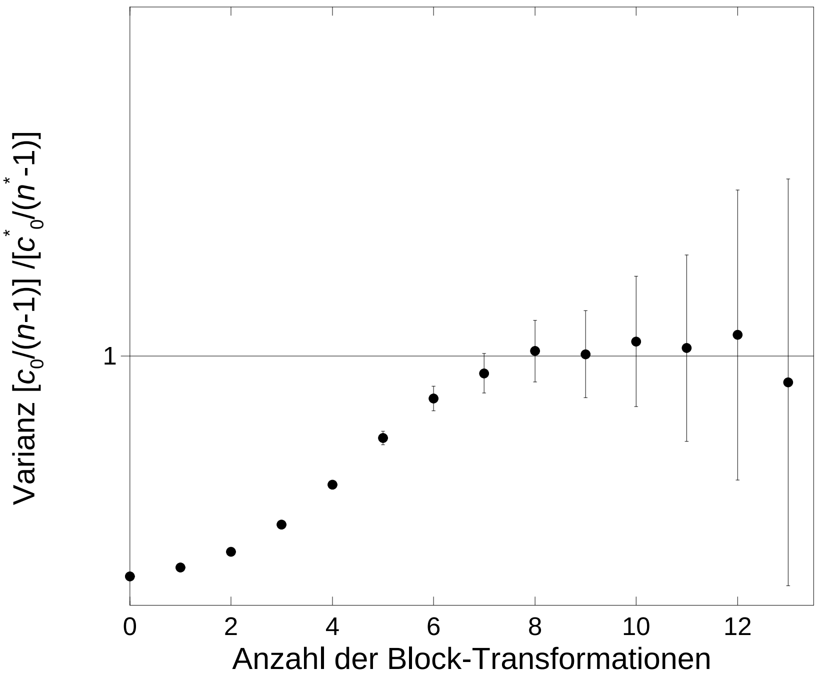


Abb. 3.2: Schematische Darstellung des Effektes sukzessiver Anwendung der Block–Transformation auf die Varianzschätzung $c_0/(n-1)$; die Fehlerbalken entsprechen der gaußschen Varianz gemäß (3.23). Die Daten stammen aus einer Simulation des dreidimensionalen XY–Modells.

Die Binning–Schätzung der Varianz des Mittelwertes wird daher in Abhängigkeit von der Anzahl der vollzogenen Transformationen typischerweise das in Abb. 3.2 dargestellte Verhalten zeigen: einem relativ glatten Anstieg auf den Plateau–Wert folgt eine stärker werdende erratische Schwankung, die dem Fehler der Schätzung gemäß (3.23) entspricht.

Mit derselben Methode läßt sich jetzt auch die Autokorrelationszeit bestimmen; aus (3.4) folgt:

$$\tau_{\text{int}} = \frac{1}{2}\frac{\sigma^2(\bar{A})}{\sigma^2(A)/n}. \tag{3.24}$$

Während eine Schätzfunktion für $\sigma^2(\bar{A})$ gerade durch das Binning geliefert wird, ist ein erwartungstreuer Schätzer für $\sigma^2(A)$ aus unabhängigen Größen nicht so leicht zu gewinnen. Der naive Ansatz liefert:

$$\begin{aligned} \langle\frac{1}{n-1}\sum_{t=1}^{n}(A_t-\bar{A})^2\rangle &= \sigma^2(A) - \frac{1}{n(n-1)}\sum_{s\neq t}\gamma_{s,t} \\ &= \sigma^2(A)\left[1-\frac{2}{n-1}(\tau_{\text{int}}-\frac{1}{2})\right], \end{aligned} \tag{3.25}$$

wobei in der letzten Zeile Gl. (3.17) verwendet wurde. Da n jedoch hier die ursprüngliche Länge der Zeitreihe symbolisiert, kann man für $n \gg \tau$, was man für die Bestimmung von $c_0^*/(n^*-1)$ ohnehin fordern muß, die Korrektur in den eckigen Klammern vernachlässigen und erhält:

$$\tau_{\rm int} \approx \frac{n(n-1)}{2(n^*-1)} \frac{c_0^*}{c_0}, \tag{3.26}$$

wobei c_0 aus der ursprünglichen Reihe und c_0^* nach hinreichend vielen Block–Transformationen zu bestimmen ist.

3.2.2 Jackknifing

Die Grundannahme statistischer Simulationen wie der Monte–Carlo–Methode behauptet die Möglichkeit, von der empirischen Verteilung $\hat{F}(A) = 1/n \sum_t \delta(A - A_t)$ einer betrachteten Zufallsvariable A auf die wahre Verteilung $F(A)$ schließen zu können. In der allgemeinsten Formulierung versucht man, einen Parameter $f(F)$ der Verteilung zu schätzen, indem man $\hat{f} \equiv f(\hat{F})$ bestimmt. Ist die Zeitreihe unendlich lang und die Dynamik ergodisch, so wird dies auch uneingeschränkt gelingen. Für eine endliche Zeitreihe muß dies nicht unbedingt gelten; betrachtet man etwa eine Funktion des Erwartungswertes von A, z.B. $f(F) = \langle A \rangle^2$, so erhält man im Mittel über unendlich viele Reihen der Länge n für die gemessene Verteilung:

$$\langle f(\hat{F}) \rangle = \langle \bar{A}^2 \rangle = \langle \left(\frac{1}{n} \sum_t A_t \right)^2 \rangle = \frac{n-1}{n} \langle A \rangle^2 + \frac{1}{n} \langle A^2 \rangle, \tag{3.27}$$

sofern die Messungen A_t zeitlich unkorreliert sind. Das liefert jedoch erst im Limes $n \to \infty$ das gewünschte Ergebnis. Die Abweichung zwischen Schätzer und wahrem Wert nennt man den *Bias* der Schätzfunktion:

$$\text{Bias} \equiv \langle f(\hat{F}) \rangle - f(F). \tag{3.28}$$

Im angegebenen Beispiel ist der Bias offenbar $O(1/n)$. Wir wollen uns im folgenden auf Funktionen des *Erwartungswertes* von A, wie die Beispielfunktion eine war, beschränken, d.h.

$$f(F) = f(\langle A \rangle), \tag{3.29}$$

da diese im Zusammenhang mit MC–Simulationen mit Abstand die wichtigste Rolle spielen. Wie kann man den Bias solcher Schätzfunktionen systematisch reduzieren?

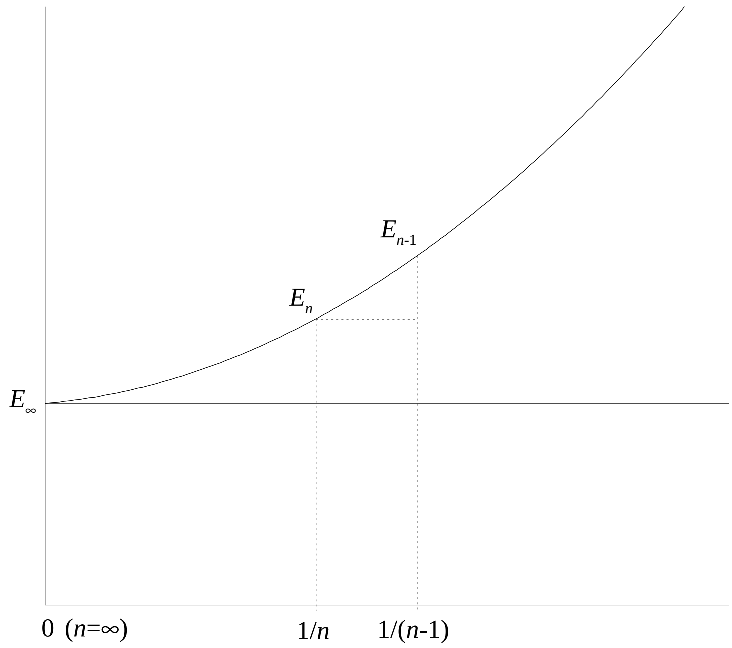


Abb. 3.3: Der Erwartungswert E_n als Funktion von $1/n$. In erster Ordnung verhält sich der Bias linear in $1/n$.

Für nicht allzu pathologische Funktionen f in (3.29) läßt sich die aus einer Meßreihe der Länge n gewonnene Schätzung wie folgt entwickeln:

$$E_n \equiv \langle f(\bar{A}) \rangle = f(\langle A \rangle) + \frac{a_1}{n} + \frac{a_2}{n^2} + \cdots, \tag{3.30}$$

wobei die a_i nicht von n abhängen [77]; der Bias ist also in erster Näherung proportional $(1/n)$. Dabei geht als Voraussetzung ein, daß die Elemente A_t der Zeitreihe unkorreliert sind, d.h. $\gamma_{i,j} = 0, i \neq j$; da dies wie gesehen bei den meisten Messungen nicht der Fall ist, wird man die Zeitreihe im allgemeinen zunächst mit der beschriebenen Binning–Methode in eine unkorrelierte Ersatzreihe transformieren müssen. Aus Abb. 3.3 liest man in dieser Näherung dann ab:

$$\frac{E_n - E_\infty}{E_{n-1} - E_n} = \frac{1/n}{1/(n-1) - 1/n}. \tag{3.31}$$

Daher lassen sich der wahre Wert $E_\infty \equiv f(\langle A \rangle)$ und der Bias $E_n - E_\infty$ aus den Schätzungen bestimmen zu:

$$\begin{aligned} E_\infty &= nE_n - (n-1)E_{n-1}, \\ E_n - E_\infty &= (n-1)(E_{n-1} - E_n). \end{aligned} \tag{3.32}$$

Ein Schätzer für E_n ist natürlichweise gegeben durch $\hat{f} = f(\bar{A})$. Man hat dann also die Aufgabe, aus der gegebenen Reihe der Länge n eine möglichst gute, unabhängige Schätzung für E_{n-1} zu gewinnen; da man keine neuen Daten aufnehmen (*samplen*) will, faßt man die gegebene Reihe ersatzweise als Grundgesamtheit auf und nimmt hieraus Reihen der Länge $n-1$ auf (*Resampling*). Offenbar lassen sich aus den vorliegenden Daten n Reihen der Länge $n-1$ bilden, nämlich indem man jeweils einen der n Meßwerte herausnimmt — gleichsam mit dem Taschenmesser (*jackknife*) herausschneidet. Man nimmt daher das arithmetische Mittel über diese aus den gegebenen Daten extrahierbaren n Schätzungen für E_{n-1}, d.h. man betrachet die Ersatzverteilungen $\hat{F}_{(s)}(A) = 1/(n-1)\sum_{t\neq s}\delta(A-A_t)$ und bildet:

$$\begin{aligned} \hat{f}_{(s)} &= f(\hat{F}_{(s)}) = f\left(\frac{1}{n-1}\sum_{t\neq s} A_t\right), \\ \hat{f}_{(\cdot)} &= \frac{1}{n}\sum_{s=1}^{n} f_{(s)}. \end{aligned} \tag{3.33}$$

Mit (3.32) ergibt sich daher als sog. Quenouillesche Schätzung des Bias [31]:

$$\widehat{\mathrm{BIAS}} = (n-1)(\hat{f}_{(\cdot)} - \hat{f}) \tag{3.34}$$

und daraus ein in erster Ordnung um den Bias bereinigter Schätzer zu:

$$\tilde{f} \equiv \hat{f} - \widehat{\mathrm{BIAS}} = n\hat{f} - (n-1)\hat{f}_{(\cdot)}. \tag{3.35}$$

Enwickelt man nun auch $\hat{f}_{(\cdot)}$ nach Potenzen von n:

$$\langle \hat{f}_{(\cdot)} \rangle = E_{n-1} = f(\langle A \rangle) + \frac{a_1}{n-1} + \frac{a_2}{(n-1)^2} + \cdots, \tag{3.36}$$

so zeigt sich, daß gemäß

$$\begin{aligned} \langle \tilde{f} \rangle &= nE_n - (n-1)E_{n-1} \\ &= f(\langle A \rangle) - \frac{a_2}{n(n-1)} + a_3\left(\frac{1}{n^2} + \frac{1}{(n-1)^2}\right) + \cdots \end{aligned} \tag{3.37}$$

der Bias von $\tilde{f}$ von der Ordnung $O(1/n^2)$ ist gegenüber $O(1/n)$ für $\hat{f}$.

Diese Vorgehensweise bildet nur eine von einer ganzen Reihe solcher Resampling–Methoden, ist jedoch rechentechnisch noch am ehesten zu bewältigen; das Bootstrapping etwa liefert im allgemeinen genauere Ergebnisse, erfordert aber auch die rechnerische Behandlung aller aus der ursprünglichen Reihe zu gewinnenden Teilmengen beliebiger Länge (der Potenzmenge) [31]. Eine solche Bias–Bereinigung von

Schätzfunktionen wird immer dann notwendig sein, wenn man *nichtlineare* Funktionen der Observablen betrachtet. Man könnte vermuten, daß diese Effekte in Monte–Carlo–Simulationen mit den aus den heutigen Rechenleistungen resultierenden langen Zeitreihen zu vernachlässigen sind, da sie doch mit $1/n$ abfallen. Die Anzahl von Einzelmessungen, die zum Erreichen unabhängiger Werte zusammengefaßt werden, ist jedoch oft sehr groß, d.h. die Anzahl der verwendeten Bins — auch aus Speicherplatzgründen — relativ klein (oft kleiner 10^2), so daß häufig ein relevanter Teil der in der Simulation erreichten Genauigkeit durch die Verwendung von nicht biasbereinigten Schätzern wieder verlorengeht. Da der Bias wie gesehen in führender Ordnung proportional zu $1/n$ mit wachsender Zahl von Messungen bzw. Bins abfällt, die Fehler der Messungen, d.h. die Standardabweichungen, jedoch typischerweise proportional $1/\sqrt{n}$, wird der Bias zwar asymptotisch für $n \to \infty$ gegenüber den statistischen Fehlern immer zu vernachlässigen sein; für endliche Laufzeiten n kann er jedoch je nach den zugehörigen Amplituden der beiden Terme gegenüber dem Fehler durchaus eine Rolle spielen. Eine Berücksichtigung der höheren Ordnungen in (3.30) andererseits ist zwar möglich [65], aber rechentechnisch sehr aufwendig und bei Simulationen in der statistischen Mechanik, wo relativ lange Zeitreihen auftreten, im allgemeinen nicht notwendig. Da es sich bei der Bias–Schätzung $\widehat{\mathrm{BIAS}}$ und deren Analoga höherer Ordnung im wesentlichen um Extrapolationsmethoden handelt, wird sich außerdem der *statistische* Fehler mit jeder Ordnung der Korrektur vergrößern, so daß man schon sehr lange (unkorrelierte) Zeitreihen benötigt, um eine Bias–Korrektur höherer Ordnung sinnvoll erscheinen zu lassen. Was die praktische Grenze solcher Korrekturverfahren angeht, so zeigt sich, daß ihre Anwendung den Fällen vorbehalten bleiben sollte, wo $\widehat{\mathrm{BIAS}}$ deutlich größer ist als die Varianz der betrachteten Größe $f(\bar{A})$ [31].

Die Jackknife–Methode zur Bestimmung des Bias liefert ein Werkzeug zur Verbesserung der Schätzung von nichtlinearen Funktionen von Observablen. Eine weitere nichttriviale Aufgabe ist die Bestimmung des *statistischen* Fehlers solcher Schätzungen. Zwar läßt sich die Varianz des einfachen Mittelwertes erwartungstreu mit der üblichen Formel

$$\hat{\sigma}^2(\bar{A}) = \frac{1}{n(n-1)} \sum_{t=1}^{n} (A_t - \bar{A})^2 \tag{3.38}$$

schätzen, aber dieser Ansatz kann nicht auf (nichtlineare) Funktionen $f(\bar{A})$ verallgemeinert werden. Auch für dieses Problem bietet der Jackknife–Ansatz eine Lösung an; sie macht sich die Tatsache zunutze, daß mit den Jackknife–Variablen $\hat{f}_{(s)}$ n verschiedene Schätzungen für $f(\langle A \rangle)$ vorliegen, so als hätte man n Zeitreihen der Länge

$n-1$ aufgenommen. Es liegt daher nahe, die mittlere quadratische Abweichung der $\hat{f}_{(s)}$ von ihrem Mittelwert $\hat{f}_{(\cdot)}$ als Maß für die Varianz der Schätzung zu verwenden. Der Jackknife–Schätzer für die Varianz von $f(\bar{A})$ ist gegeben durch:

$$\widehat{\mathrm{VAR}} = \frac{n-1}{n} \sum_{s=1}^{n} \left[\hat{f}_{(s)} - \hat{f}_{(\cdot)}\right]^2. \tag{3.39}$$

Er setzt sich aus zwei Teilen zusammen: die Summe entspricht dem eigentlichen Jackknife–Schätzer für die Varianz Var_{n-1} von $f_{n-1}(\bar{A})$, d.h. für Zeitreihen der Länge $n-1$; den Vorfaktor $(n-1)/n$ kann man als eine Umnormierung des Ergebnisses von Reihen der Länge $n-1$ auf die vorliegende Reihe der Länge n verstehen. Die Varianzen vieler Verteilungen zeigen folgende Abhängigkeit von der Sample–Länge n (vgl. z.B. (3.4)):

$$\mathrm{Var}_n = \frac{n-1}{n}\mathrm{Var}_{n-1} + O(\frac{1}{n^3}), \tag{3.40}$$

so daß der Vorfaktor $\frac{n-1}{n}$ in (3.39) bis auf Terme der Ordnung $O(1/n^3)$ die Schätzung korrekt auf die Reihen der Länge n umnormiert. Es wird zunächst verwundern, daß dem Schätzer für Var_{n-1}, d.h. der Summe in (3.39), gegenüber dem Ansatz (3.38) im linearen Falle der Vorfaktor $\frac{1}{n(n-1)}$ fehlt. Nun sind aber die Schätzungen $\hat{f}_{(s)}$ für $f(\langle A\rangle)$ keineswegs unabhängig, so daß sich der naive Ansatz (3.38) anwenden ließe; vielmehr sind sie hochgradig korreliert, da sie jeweils bis auf einen Wert aus denselben Messungen der A_t zusammengesetzt sind. Gerade weil es sich um dieselben Werte handelt, sind jedoch die Korrelationen trivial und das Fehlen des Vorfaktors korrigiert die Schätzung gerade um diese formellen Korrelationen. Anschaulich ist dies zu sehen, wenn man den trivialen Fall $f(\bar{A}) = \bar{A}$ des Mittelwertes selbst betrachtet; $\widehat{\mathrm{VAR}}$ liefert dann gerade den Ausdruck (3.38) zurück. Eine mathematisch stringente Begründung der Jackknife–Schätzung für die Varianz kann nur im Rahmen des allgemeinen Resampling–Konzepts erfolgen; der Leser sei hierzu auf die Arbeit von Efron [31] verwiesen. Was die Verläßlichkeit von $\widehat{\mathrm{VAR}}$ angeht, so kann man zeigen, daß zumindest

$$\langle \frac{n}{n-1}\widehat{\mathrm{VAR}}\rangle \geq \mathrm{Var}_{n-1} \tag{3.41}$$

gilt, d.h. abgesehen von der Umnormierung auf die Sample–Länge n ist die Jackknife–Schätzung der Varianz vorsichtig in dem Sinne, daß sie die Varianz eher überschätzt.

Ein spezielles Problem tritt auf, wenn man beide Jackknife–Verfahren, die Bias– und die Varianz–Schätzung, parallel verwenden will. Aus den obigen Erörterungen ist klar, daß $\widehat{\mathrm{BIAS}}$ selbst eine Zufallsvariable ist und dementsprechend auch die korrigierte Schätzung $\tilde{f}$ für $f(\langle A\rangle)$. $\widehat{\mathrm{VAR}}$ liefert zunächst jedoch nur ein Maß für die Schwankung von $\hat{f}$, nicht aber von $\tilde{f}$. Eine solche Angabe läßt sich gewinnen, indem man die Jackknife–Prozedur iteriert und ausgehend von der Verteilung $\hat{F}_{(s)}(A) = 1/(n-1)\sum_{t\neq s}\delta(A-A_t)$ in einem erneuten Resampling einzelne Werte der Reihe wegläßt, d.h. die Verteilung $\hat{F}_{(r)(s)}(A) = 1/(n-2)\sum_{t\neq r,s}\delta(A-A_t)$ verwendet. Mit den zusätzlichen Definitionen

$$\begin{aligned}
\hat{f}_{(r)(s)} &= f\left(\frac{1}{n-2}\sum_{t\neq r,s} A_t\right), \\
\hat{f}_{(\cdot)(s)} &= \frac{1}{n-1}\sum_{r\neq s}\hat{f}_{(r)(s)}, \\
\hat{f}_{(r)(\cdot)} &= \frac{1}{n-1}\sum_{s\neq r}\hat{f}_{(r)(s)},
\end{aligned} \tag{3.42}$$

erhält man ein Jackknife–Sample von biasbereinigten Schätzern:

$$\begin{aligned}
\tilde{f}_{(s)} &= (n-1)\hat{f}_{(s)} - (n-2)\hat{f}_{(s)(\cdot)}, \\
\tilde{f}_{(\cdot)} &= \frac{1}{n}\sum_{s}\tilde{f}_{(s)}.
\end{aligned} \tag{3.43}$$

Darauf läßt sich nun die ursprüngliche Jackknife–Varianzschätzung $\widehat{\mathrm{VAR}}$ (3.39) anwenden:

$$\widehat{\mathrm{VAR}}(\tilde{f}) = \frac{n-1}{n}\sum_{s}\left[\tilde{f}_{(s)} - \tilde{f}_{(\cdot)}\right]^2 \tag{3.44}$$

Mit einem solchen Jackknife–Verfahren zweiter Ordnung läßt sich somit eine in der ersten Ordnung um den Bias korrigierte Schätzfunktion für $f(\langle A\rangle)$ *und* eine Angabe über die Genauigkeit dieser Schätzung ermitteln [7].

Das Jackknife–Verfahren — und allgemeiner die Methoden des Resampling — liefern Schätzungen für die statistischen Fehler von gemessenen Größen *und* für den Bias der verwendeten Schätzfunktionen. Die Jackknife–Varianzschätzung sollte bei der Betrachtung nichtlinearer Größen von Observablen immer zur Anwendung kommen; ob eine Bias–Reduktion der Schätzer mit Hilfe des Jackknife–Verfahrens sinnvoll ist, muß im Einzelfall entschieden werden und hängt nicht zuletzt, angesichts der o.g. asymptotischen Dominanz der Fluktuationen gegenüber dem Bias, von der Länge der zur Verfügung stehenden Zeitreihen ab.

Kapitel 4

Abgleich der Methoden: das 2D Ising–Modell

Mit dem Wolff–Cluster–Algorithmus und der Binning– und Jackknife–Analyse stehen die grundlegenden Methoden zur Untersuchung des FSS der Korrelationslängen der $O(n)$–Spin–Modelle bereit. Die Anpassung dieser Methoden an die Erfordernisse des untersuchten Problems steht aber noch aus; dabei geht es einerseits um Parameter der Simulation und Methoden der Messung, etwa die Frage, wie sich die Korrelationslängen, die ja keine direkten Observablen darstellen, am besten bestimmen lassen oder das Problem der optimalen Dimensionierung der simulierten Systeme. Auf der anderen Seite soll unter Zuhilfenahme der Datenanalyse–Techniken der maximale Informationsgehalt der Meßreihen extrahiert werden; hier muß etwa die Frage beantwortet werden, wie sich aus den einzelnen Messungen der Korrelationslängen ein optimaler Mittelwert gewinnen läßt.

Der Verlauf dieses Abgleichsprozesses ist weitgehend unabhängig vom dabei betrachteten Modell (innerhalb der Klasse der $O(n)$–Spin–Modelle); im Grundsatz muß er daher nur einmal erfolgen. Da die Aussagen der konformen Feldtheorie für zweidimensionale Systeme (im Kontinuumslimes) exakt sind und dort das Ising–Modell als einziges Spin–Modell einen regulären Phasenübergang zweiter Ordnung aufweist, ist das zweidimensionale Ising–Modell die kanonische Wahl für einen solchen Abgleich. Hinzu kommt, daß dieses Modell auf anderem Wege, nämlich mit einer Transfermatrix–Rechnung auch im diskreten Bereich von Onsager exakt gelöst wurde [70, 71], so daß ein definitiver Vergleich der Ergebnisse mit den „wahren" Werten möglich ist. Da es sich bei dem Verfahren um einen rückgekoppelten Prozeß handelt, erscheint eine einfache lineare Darstellung nicht sinnvoll; wir werden daher die Methoden in einem Verfahren stetiger Verfeinerung zwischen Datenproduktion und –analyse pendelnd entwickeln.

4.1 Bestimmung der Korrelationslängen

4.1.1 Allgemeines, Bestimmung der Korrelationsfunktionen

Das zu untersuchende System besteht aus einem ferromagnetischen, feldfreien Ising–Modell auf der Geometrie $S^1 \times \mathbb{R}$; da ein solches semifinites Modell nicht simuliert werden kann, muß die $\mathbb{R}$–Richtung durch eine endliche Strecke ersetzt werden. Wir verwenden daher ein kubisches Gitter mit den Abmessungen (L_x, L_y), wo $L_y \gg L_x$ gesetzt wird. Wie eine vernünftige Relation L_y/L_x in der Praxis zu wählen ist, wird in Kap. 4.2 diskutiert. Um den Effekt der Endlichkeit des Gitters zu minimieren, werden in der y–Richtung darüber hinaus immer periodische Randbedingungen gewählt, d.h. für den Ising–Spin $s(x,y)$ an der Position (x,y) des Gitters gilt:

$$s(x, y + L_y) = s(x,y) \quad \forall \mathbf{x} = (x,y). \tag{4.1}$$

Die direkt zu messenden Größen sind die Zweipunkt–Korrelationsfunktionen $G^c_\sigma(\mathbf{x}_1, \mathbf{x}_2)$ und $G^c_\epsilon(\mathbf{x}_1, \mathbf{x}_2)$ gemäß (2.5); ein einfacher Schätzer für G^c_σ aus einer Meßreihe der Länge N ist gegeben durch

$$\hat{G}^c_\sigma(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{N}\sum_{t=1}^{N} s_t(\mathbf{x}_1)s_t(\mathbf{x}_2) - \left(\frac{1}{NV}\sum_{t=1}^{N}\sum_{\mathbf{x}} s_t(\mathbf{x})\right)^2, \tag{4.2}$$

wobei im zweiten Term schon die Identität der Erwartungswerte der Verteilungen der $s(\mathbf{x})$ verwendet wurde, um die Statistik zu verbessern. Der Schätzer ist allerdings nicht erwartungstreu, da, wie das Beispiel in Kap. 3.2.2 gezeigt hat, die zweite Summe als Schätzer des residualen Anteils $\langle s \rangle^2$ einen Bias aufweist; da die Anzahl der hier eingehenden Messungen (n in Gl. (3.27)) sehr groß ist, kann man diesen Effekt zunächst vernachlässigen. Weil die *longitudinalen* Korrelationslängen $\xi_{||}$ bestimmt werden sollen, sind nur solche Paare $\mathbf{x}_1$ und $\mathbf{x}_2$ relevant, für die $\mathbf{x}_2 - \mathbf{x}_1$ parallel zur y–Achse liegt. Insofern die periodischen Randbedingungen entlang der y–Achse Translationsinvarianz in dieser Richtung garantieren, kann man zudem über alle solchen Paare $\mathbf{x}_1$ und $\mathbf{x}_2$ mit dem gleichen Abstand $|\mathbf{x}_2 - \mathbf{x}_1|$ mitteln, was unter Berücksichtigung der periodischen Randbedingungen jeweils $L_x L_y = V$ Werte ergibt. Es stellt daher

$$\hat{G}_\sigma^{c,\|}(|\mathbf{x}_2-\mathbf{x}_1|) = \frac{1}{NV}\sum_{t=1}^{N}\sum_{\substack{|\mathbf{x}_2-\mathbf{x}_1|\\(\mathbf{x}_2-\mathbf{x}_1)\|\hat{e}_y}} s_t(\mathbf{x}_1)s_t(\mathbf{x}_2) - \left(\frac{1}{NV}\sum_{t=1}^{N}\sum_{\mathbf{x}} s_t(\mathbf{x})\right)^2 \tag{4.3}$$

einen varianzreduzierten Schätzer für die gesuchte Größe dar.

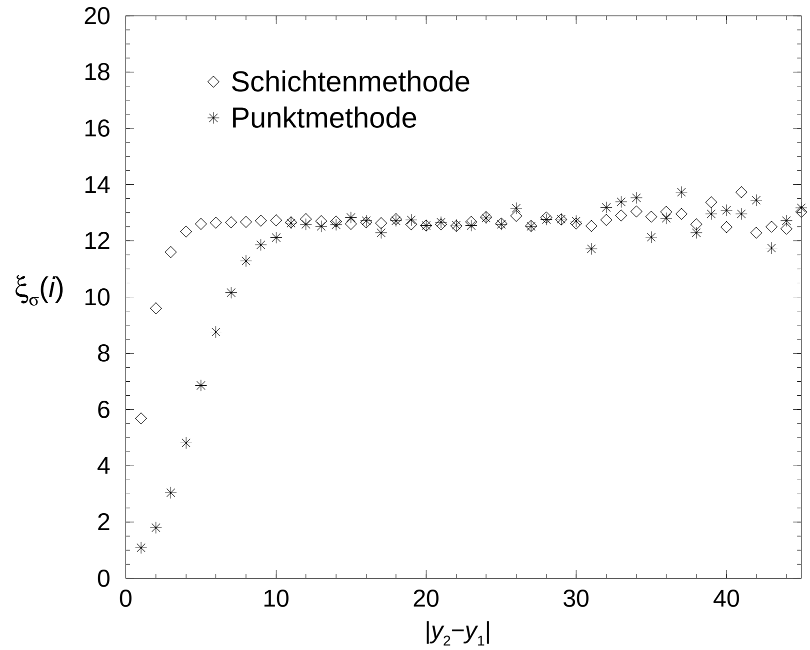
Neben diesem im folgenden als „Punktmethode“ bezeichneten Vorgehen bietet sich eine weitere Möglichkeit der Schätzung der Korrelationsfunktionen an, die sich in der Reihenfolge der Mittelung unterscheidet: man projiziert dazu alle Spins auf die longitudinale Achse $\mathbb{R}$ und betrachtet die Korrelationen zwischen diesen Ersatzgrößen, d.h. man bildet innerhalb der Schichten $\{(x,y)\,|\,y=\text{const}\}$ Mittelwerte

$$S_t(y) = \frac{1}{L_x}\sum_x s_t(x,y) \tag{4.4}$$

und erzeugt daraus den Schätzer

$$\tilde{G}_\sigma^{c,\|}(|y_2-y_1|) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{L_y}\sum_{|y_2-y_1|} S_t(y_1)S_t(y_2) - \left(\frac{1}{NL_y}\sum_{t=1}^{N}\sum_{y} S_t(y)\right)^2 . \tag{4.5}$$

Diese „Schichtenmethode“ ist zunächst nicht äquivalent zur Punktmethode, denn die Korrelationen innerhalb der Schichten, in der transversalen Richtung, gehen hierbei zusätzlich ein. Die Kontinuumsform (2.70) der Korrelationsfunktion zeigt jedoch, daß diese Effekte mit wachsendem Abstand $|y_2-y_1|$ verschwinden — und nur in diesem Limes ist ja auch die FSS–Relation (2.72) gültig. Eine solche Projektion bietet gegenüber der Punktmethode einige Vorteile; zunächst ist der Rechenaufwand hier deutlich geringer, da bei einem gegebenen Abstand pro Messung lediglich L_y Multiplikationen notwendig sind, während (4.3) L_xL_y Multiplikationen erfordert, wobei die Bildung der Mittelwerte $S_t(y)$ keinen zusätzlichen Rechenaufwand bedeuten, da sie zur Schätzung des residualen Terms in jedem Fall bestimmt werden müssen. Vor allem stellt die Projektionsmethode eine einfache Möglichkeit dar, die Statistik der Messung zu verbessern: da die Korrelationen von *Mittelwerten* betrachtet werden, die weniger schwanken als die in sie eingehenden Einzelwerte, wird auch der statistische Fehler der Korrelationsfunktion bei gleicher Länge der Zeitreihe bei der Schichtenmethode geringer sein als bei der Punktmethode [85, 55]. In Abb. 4.1 sind die sich aus dem Abfall von $\hat{G}_\sigma^{c,\|}$ bzw. $\tilde{G}_\sigma^{c,\|}$ zwischen benachbarten Abständen $|y_2-y_1|$ entlang der Zylinderachse ergebenden Schätzungen der Korrelationslänge ξ_σ für ein Beispielsystem mit den Abmessungen 10×120 aufgetragen. Aufgrund



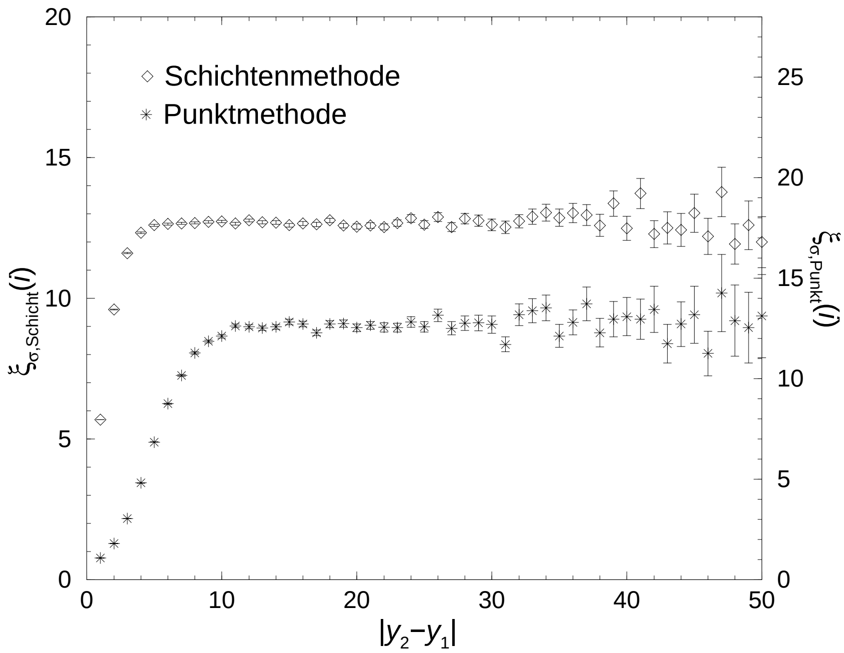


Abb. 4.1: Vergleich der aus der Punktmethode $\hat{G}_{\sigma}^{c,\parallel}$ und der Schichtenmethode $\tilde{G}_{\sigma}^{c,\parallel}$ gewonnenen Schätzungen für die Spin–Korrelationslänge für ein Beispielsystem mit den Abmessungen 10×192 und periodischen Randbedingungen in x–Richtung. Die Methode zur Bestimmung der ξ_i und der Fehler ist in Kap. 4.1.2 bzw. Kap. 4.4.1 beschrieben. **Oben:** Beide Methoden erzeugen im Bereich großer Abstände im Mittel identische Ergebnisse. **Unten:** Die Schichtenmethode liefert einen gegenüber der Punktmethode deutlich varianzreduzierten Schätzer.

der periodischen Randbedingungen in y–Richtung lassen sich mit beiden Methoden natürlich die Korrelationsfunktionen nur für Abstände $|y_2 - y_1| \leq L_y/2$ bestimmen; die Schätzungen für alle größeren Abstände $L_y/2 < |y_2 - y_1| < L_y$ sind identisch mit denen für die Abstände $L_y - |y_2 - y_1|$. Man erkennt sowohl die Konvergenz der beiden Methoden für große Abstände, d.h. für den Gültigkeitsbereich der FSS–Hypothese, als auch die deutlich geringeren statistischen Fehler der Ergebnisse der Schichtenmethode gegenüber denen aus der Punktmethode. Dieser Effekt wird aus folgender Überlegung verständlich: in einem Bereich, wo die transversalen Korrelationen zu vernachlässigen sind, reduziert die Schichtenbildung die Varianz der $s(\mathbf{x})$ um den Faktor $1/L_y$ und daher die des Produktes $s(\mathbf{x}_1)s(\mathbf{x}_2)$ um einen Faktor $1/L_y^2$; zusammen mit der Mittelung entlang der longitudinalen Richtung $\mathbb{R}$ ergibt sich ein Faktor von $1/(VL_y)$ — gegenüber der Varianzreduktion des Punktschätzers (4.3) von der Größenordnung $1/V$ ergibt sich damit eine um $1/L_y$ verkleinerte Varianz. Im folgenden werden wir daher ausschließlich die Schichtenmethode verwenden.

Was die Messung von $G_\epsilon^{c,\|}$ angeht, ist zunächst unklar, wie die Energiedichte definiert werden soll, d.h. zu welchem (evtl. richtungsabhängigen) Anteil die Energien der Bindungen eines Spins der Dichte an dem gegebenen Punkt zugerechnet werden sollen. Im Falle der Schichtenmethode ist zumindest klar, daß die Energien der Bindungen *innerhalb* der Schichten den Schichtvariablen $\mathcal{E}_t(y)$ zugerechnet werden; was die Bindungen in y–Richtungen angeht, so bieten sich die Möglichkeiten, jeweils beide Bindungen („vorwärts" und „rückwärts") ganz oder zur Hälfte dem betreffenden Spin der Schicht zuzurechnen, oder aber lediglich Bindungen in einer Richtung („vorwärts") zu berücksichtigen. Ob man die Energien im symmetrischen Fall ganz oder zur Hälfte eingehen läßt, ist jedoch offenbar für die Korrelationslängen ohne Belang, da sich lediglich der jeweilige absolute Wert der Korrelationsfunktion verdoppelt. Der symmetrische und der asymmetrische Ansatz sind zwar algebraisch verschieden, aber die Unterschiede verschwinden im Mittel, wie in Abb. 4.2 zu erkennen ist. Daher definieren wir die Schichtvariablen der Energiedichte als

$$\mathcal{E}_t(y) = \frac{1}{L_x} \sum_x s_t(x,y) \left[s_t(x+1,y) + s_t(x,y+1) \right] \tag{4.6}$$

und wenden den Schätzer $\tilde{G}^{c,\|}$ gemäß (4.5) analog auf die Energiedichte an, d.h.:

$$\tilde{G}_\epsilon^{c,\|}(|y_2 - y_1|) = \frac{1}{N} \sum_{t=1}^{N} \frac{1}{L_y} \sum_{|y_2 - y_1|} \mathcal{E}_t(y_1)\mathcal{E}_t(y_2) - \left(\frac{1}{NL_y} \sum_{t=1}^{N} \sum_y \mathcal{E}_t(y) \right)^2 . \tag{4.7}$$

Neben der Punkt– und der Schichtmethode bietet sich als dritte Möglichkeit der

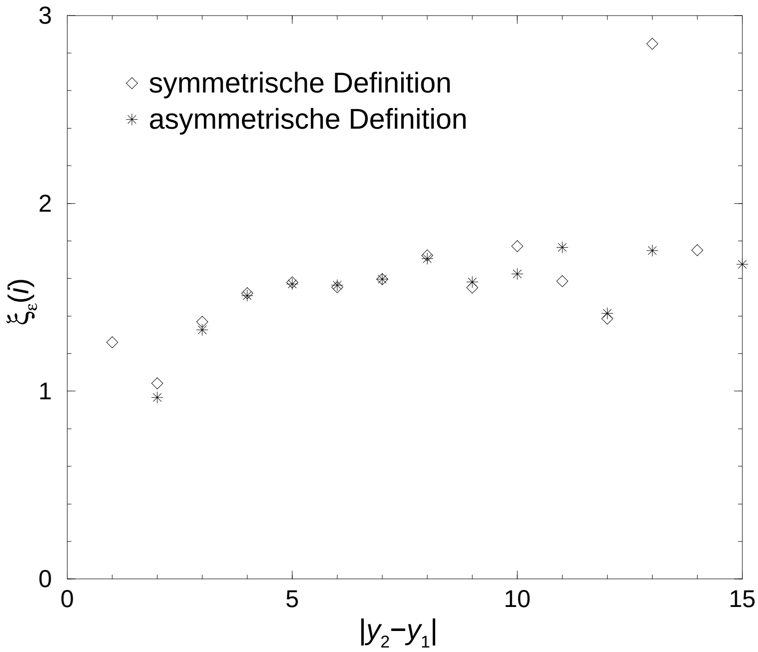


Abb. 4.2: Vergleich der symmetrischen („vorwärts“ und „rückwärts“) und der asymmetrischen („vorwärts“) Definition der Energiedichte für ein System mit den Abmessungen 10×192 und periodischen Randbedingungen. Als fehlergewichtete Mittelwerte der Plateau–Bereiche ergibt sich 1.5921(96) für die asymmetrische und 1.5903(83) für die symmetrische Methode.

Bestimmung der Korrelationsfunktionen die Verwendung von sog. *Improved Cluster Estimators* an, die in natürlicher Weise aus der Verwendung von Cluster–Algorithmen resultieren. Man kann zeigen [87], daß sich die Spin–Spin–Korrelation im Falle des Wolff–Cluster–Algorithmus alternativ durch Auszählen der in einem Cluster enthaltenen Spins gewinnen läßt. Es ist dann für ein allgemeines $O(n)$–Spin–Modell:

$$\hat{G}_\sigma^{\mathrm{imp}}(\mathbf{x}_1, \mathbf{x}_2) = \frac{n}{N\langle|C|\rangle} \sum_{t=1}^{N} \mathbf{r}_t \cdot \mathbf{s}_t(\mathbf{x}_1) \mathbf{r}_t \cdot \mathbf{s}_t(\mathbf{x}_2) \Theta_{C_t}(\mathbf{x}_1) \Theta_{C_t}(\mathbf{x}_2), \tag{4.8}$$

wobei $\mathbf{r}_t$ der zufällig gewählten Richtung gemäß (3.11) entspricht, die Normierungskonstante $\langle|C|\rangle$ durch die durchschnittliche Cluster–Größe gegeben ist und Θ_{C_t} die charakteristische Funktion auf dem aktuellen Cluster C_t darstellt, d.h.

$$\Theta_{C_t}(\mathbf{x}) = \begin{cases} 1 & \text{für } \mathbf{s}(\mathbf{x}) \in C_t \\ 0 & \text{für } \mathbf{s}(\mathbf{x}) \notin C_t \end{cases} \tag{4.9}$$

Der Vorteil dieser Schätzfunktion liegt in einer gegenüber dem üblichen Verfahren reduzierten Varianz (vgl. z.B. [55]), die anschaulich aus der Tatsache resultiert, daß zur Schätzung jeweils nur *positive* Beiträge addiert werden. Einen deutlichen Vorteil bringt diese Methode jedoch nur dann, wenn die durchschnittliche Cluster–Größe $\langle |C| \rangle$ im Vergleich zum Volumen V des Systems nicht zu groß wird, d.h. außerhalb des FSS–Gebietes. Zwar sind, wie wir sehen werden, für die Wahl antiperiodischer Randbedingungen in der transversalen Richtung die Cluster deutlich kleiner als für periodische Randbedingungen, so daß trotz der Simulation am kritischen Punkt die Verwendung der Cluster–Schätzer möglicherweise noch einen (wenn auch verkleinerten) Vorteil darstellen könnte; will man jedoch auch die benötigte Korrelationsfunktion der Energiedichte mit einer ähnlichen Methode bestimmen, so werden die entsprechenden Ausdrücke so kompliziert, daß nach unserer Einschätzung der Vorteil der Varianzreduktion durch den höheren Aufwand an Rechenzeit im allgemeinen wieder verlorengehen wird. Cluster–Schätzer werden daher in dieser Arbeit nicht zur Anwendung kommen.

4.1.2 Bestimmung der Korrelationslängen

Für große longitudinale Abstände erwartet man gemäß (2.6) bzw. (2.71) für die Korrelationsfunktionen $G_\sigma^{c,\parallel}$ und $G_\epsilon^{c,\parallel}$ bei Simulationen am kritischen Punkt ein exponentiell abfallendes Verhalten der Form

$$\left.\begin{array}{rcl} G_\sigma^{c,\parallel}(|y_2 - y_1|) & = & G_\sigma^0 \exp\left(-|y_2 - y_1|/\xi_\sigma\right) \\ G_\epsilon^{c,\parallel}(|y_2 - y_1|) & = & G_\epsilon^0 \exp\left(-|y_2 - y_1|/\xi_\epsilon\right) \end{array}\right. , \quad |y_2 - y_1| \gg 1. \tag{4.10}$$

Es gilt nun, aus den Schätzern $\tilde{G}$ der Korrelationsfunktionen Schätzungen für die Korrelationslängen ξ_σ und ξ_ϵ zu gewinnen. Logarithmiert man die Schätzfunktionen $\tilde{G}$, so lassen sich die Korrelationslängen als (inverse) Steigungen der resultierenden Geraden mit Hilfe eines linearen Fits (Ausgleichsgerade) bestimmen. Dabei muß man jedoch entscheiden, welcher Bereich von Abständen $|y_2 - y_1|$ in die Regression einbezogen werden soll; denn einerseits ist die exponentiell abfallende Form (4.10) nur für hinreichend große Abstände gültig, wie schon aus den Abbn. 4.1 und 4.2 hervorgeht, so daß die Werte von $\tilde{G}$ für kleine Abstände für den Fit nicht in Frage kommen. Für sehr große Abstände andererseits wird G so klein werden, daß die Schätzung $\tilde{G}$, deren relative Schwankung hier zusätzlich immer weiter anwächst, um den Wert Null fluktuieren und dementsprechend eine Logarithmierung nicht mehr möglich sein wird. Darüber hinaus wird sich in diesem Bereich die Endlichkeit

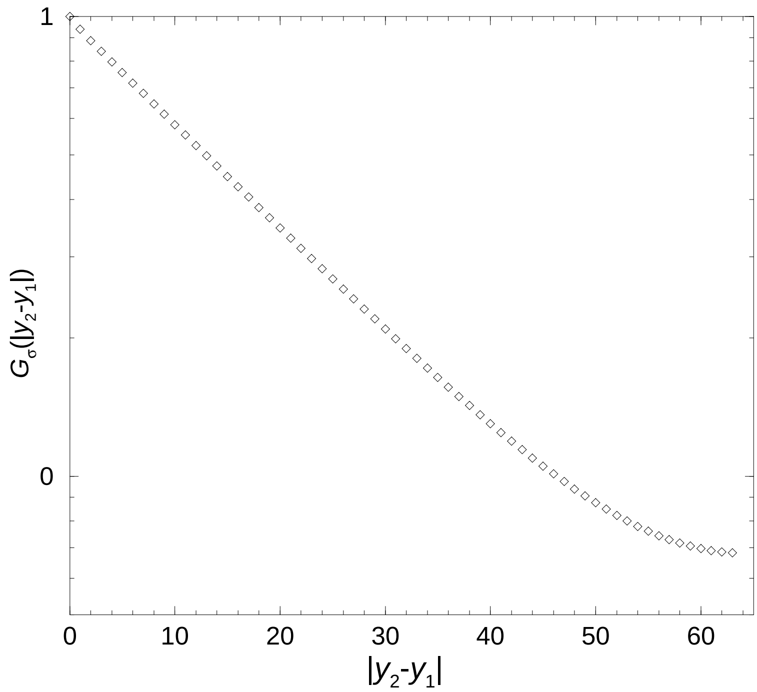


Abb. 4.3: Logarithmisch aufgetragener Verlauf der Schätzung $\tilde{G}_\sigma^{c,\|}$ der Spinkorrelationsfunktion für ein System mit den Abmessungen $L_x = 15$, $L_y = 128$ und periodischen Randbedingungen. Die Daten stammen aus einer Simulation mit 10^6 Messungen. Aufgetragen ist eine durch die Varianz $\tilde{G}_\sigma^{c,\|}(0)$ der Magnetisierung der Schichten normierte Korrelationsfunktion.

des Systems in der $\mathbb{R}$–Richtung bemerkbar und die Messungen zur Bestimmung der Korrelationslängen im Limes $L_y \to \infty$ unbrauchbar machen.

In Abb. 4.3 ist der Abfall der Schätzung $\tilde{G}_\sigma^{c,\|}$ für die Spinkorrelationsfunktion eines Systems mit periodischen Randbedingungen logarithmisch aufgetragen. Den linearen Bereich von $\tilde{G}$ in einem solchen Graphen mit bloßem Auge abzuschätzen, ist eine recht schwierige Aufgabe und daher eine etwas unsichere Methode, die jedoch durchaus verwendet wird (vgl. z.B. [85]). Man kann sich etwa behelfen, indem man aus den gemessenen Werten für $\ln \tilde{G}$ durch Interpolation eine differenzierbare Funktion erzeugt, die in etwa den bei dichter gesetzten Punkten auf dem Gitter zu erwartenden Verlauf der Korrelationsfunktion widerspiegelt. Die Ableitung dieser Funktion nach dem Abstand $|y_2 - y_1|$ sollte im Bereich des linearen Abfalls konstant sein, nämlich gerade gleich $-1/\xi$, so daß sich anhand der Ausdehnung dieses Plateaus das Gebiet des linearen Abfalls von $\tilde{G}$ abschätzen läßt. Abb. 4.4 zeigt das Ergebnis für ξ aus einer solchen Prozedur für das Beispielsystem aus Abb. 4.3: man erkennt deutlich die Ausbildung eines Plateaus im Bereich mittlerer Abstände; allerdings ist hier die

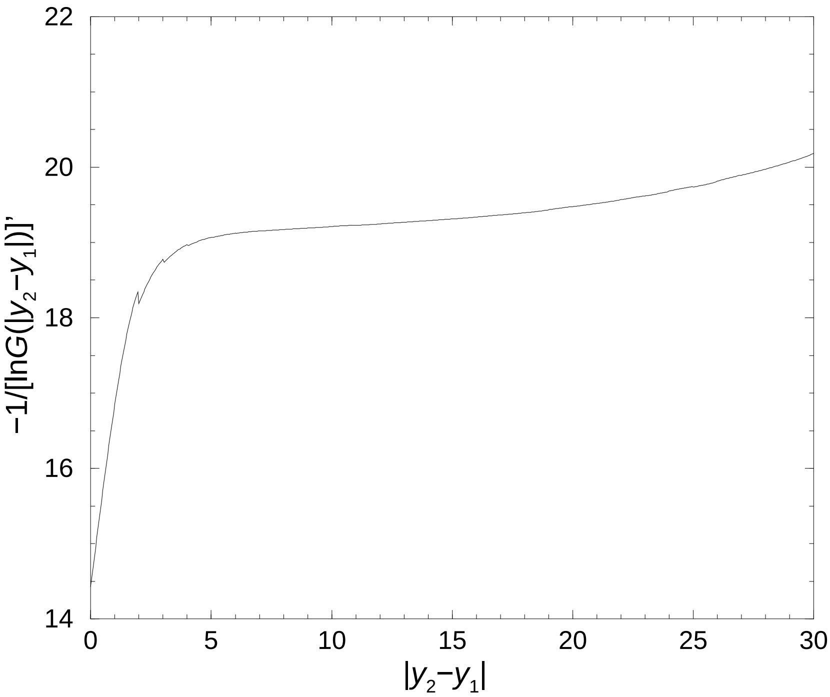


Abb. 4.4: Schätzung für die Korrelationslänge ξ_σ wie sie aus einer kubischen Interpolationsfunktion $\bar{G}$ für den Logarithmus der gemessenen Korrelationsfunktion $\tilde{G}_\sigma^{c,\|}$ des Systems aus Abb. 4.3 gemäß $\xi = -\left[\frac{d\ln\bar{G}(|y_2-y_1|)}{d|y_2-y_1|}\right]^{-1}$ folgt.

Steigung von $\ln\tilde{G}$ selbst unter Absehung von den statistischen Schwankungen nicht konstant, sondern steigt (in diesem Beispiel) leicht an. Der Grund hierfür liegt vermutlich in der Bildung der *zusammenhängenden* Korrelationsfunktionen durch Subtraktion eines residualen Terms (der Magnetisierung bzw. Energiedichte pro Spin): wir haben schon erwähnt, daß die verwendete Schätzung für diesen Term einen Bias aufweist; darüber hinaus aber wird der Schätzwert für diesen Term als statistische Größe immer etwas abseits vom „wahren“ Wert (dem Limes unendlich langer Laufzeiten) liegen. Daher muß man für den Abfall (4.10) von $\tilde{G}$ noch einen konstanten Term vorsehen, so daß nun die funktionale Form $\exp(-|y_2-y_1|/\xi)+\mathrm{const}$ vorliegt; die einfache Logarithmierung einer solchen Abhängigkeit führt jedoch gerade zu einer schwachen, aber systematischen Abweichung vom linearen Verhalten wie sie in Abb. 4.4 zu erkennen ist. Man müßte daher die zunächst angesetzte zweiparametrige, lineare Regression durch einen nun dreiparametrigen und zugleich nichtlinearen Fit ersetzen, was natürlich eine unverhältnismäßige Vergrößerung in der Unsicherheit der geschätzten Parameter bedeutete. Den Verlust an Genauigkeit gleicht zwar die zusätzliche Information über einen weiteren Parameter aus, da uns jedoch von den drei Parametern nur einer, nämlich die Konstante im Argument der Exponentialfunktion, interessiert, läßt sich aus dieser zusätzlichen Information kein Kapital schlagen und der dreiparametrige Ansatz bringt insofern nur Nachteile.

Wir suchen daher einen Ansatz, der aus den gegebenen Daten lediglich den Parameter extrahiert, der für unsere Untersuchung von Belang ist, diesen aber dann mit entsprechend größerer Genauigkeit. Im Falle des rein exponentiellen Verlaufs gemäß (4.10) lassen sich die Vorfaktoren G^0_σ bzw. G^0_ϵ durch Betrachtung von *Verhältnissen* von benachbarten Meßwerten eliminieren; für Abstände $i \equiv |y_2 - y_1|$ und $i+1$ hat man für die Korrelationsfunktion selbst:

$$\frac{G^{c,\|}(i)}{G^{c,\|}(i+1)} = \frac{\exp(-i/\xi)}{\exp(-(i+1)/\xi)} = \exp(1/\xi), \tag{4.11}$$

wobei i natürlich innerhalb des Geltungsbereichs des exponentiellen Abfalls von $G^{c,\|}$ zu wählen ist. Aus einer endlichen Zeitreihe kann man daher folgende Schätzwerte $\hat{\xi}_i$ für die zugehörige Korrelationslänge angeben:

$$\hat{\xi}_i = \left[\ln \frac{\tilde{G}^{c,\|}(i)}{\tilde{G}^{c,\|}(i+1)}\right]^{-1}. \tag{4.12}$$

Durch geeignete Mittelung über alle Abstände i innerhalb des Plateau–Bereichs erhält man damit eine Schätzung für ξ, ohne daß man den nicht benötigten Parameter G^0 mitbestimmen muß. Dieser Ansatz läßt sich auf den Fall des dreiparametrigen Problems erweitern; ließ sich die multiplikative Konstante durch Verhältnisbildung eliminieren, so leistet die Bildung von Differenzen dasselbe für eine additive Konstante. Um beide Konstanten gleichzeitig herauszufiltern, benötigt man jetzt allerdings *drei* benachbarte Werte von $\tilde{G}^{c,\|}$. Gemäß

$$\begin{aligned}\frac{G^{c,\|}(i) - G^{c,\|}(i-1)}{G^{c,\|}(i+1) - G^{c,\|}(i)} &= \frac{\exp\left(-\frac{i}{\xi}\right) - \exp\left(-\frac{i-1}{\xi}\right)}{\exp\left(-\frac{i+1}{\xi}\right) - \exp\left(-\frac{i}{\xi}\right)} = \frac{\exp\left(-\frac{i}{\xi}\right)\left[1 - \exp\left(\frac{1}{\xi}\right)\right]}{\exp\left(-\frac{i+1}{\xi}\right)\left[1 - \exp\left(\frac{1}{\xi}\right)\right]} \\ &= \exp\left(\frac{1}{\xi}\right)\end{aligned} \tag{4.13}$$

ergibt sich als Schätzung für ξ:

$$\hat{\xi}_i = \left[\ln \frac{\tilde{G}^{c,\|}(i) - \tilde{G}^{c,\|}(i-1)}{\tilde{G}^{c,\|}(i+1) - \tilde{G}^{c,\|}(i)}\right]^{-1}. \tag{4.14}$$

Abb. 4.5 zeigt die Abhängigkeit der Schätzwerte $\hat{\xi}_{\sigma,i}$ für die Spin–Korrelationslänge von der Distanz $i = |y_2 - y_1|$; die Abweichung von der Konstanz des Plateau–Wertes

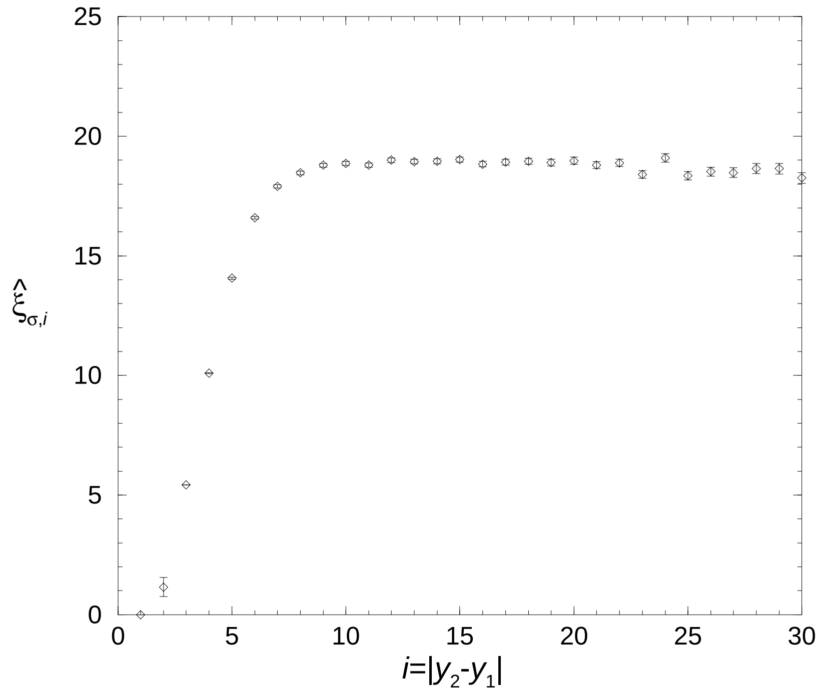


Abb. 4.5: Schätzungen $\hat{\xi}_{\sigma,i}$ für die Spin–Korrelationslänge des Systems aus den Abbn. 4.3 und 4.4 mit Hilfe der Differenzenmethode. Die Fehlerbalken stammen aus einer Jackknife–Analyse.

ist gegenüber Abb. 4.4 beseitigt. Erhalten bleibt dagegen der Vorteil einer solchen Darstellung im Vergleich zur direkten Auftragung der Korrelationsfunktion wie in Abb. 4.3 bei der Bestimmung des linearen Bereichs, der für die Berechnung der Korrelationslänge in Frage kommt. Diese kann hier relativ einfach als Abschätzung mit bloßem Auge erfolgen — in Kap. 4.4.5 stellen wir jedoch eine Methode vor, mit der sich der Vorgang teilweise automatisieren und auf eine solide mathematisch–statistische Grundlage stellen läßt. Ganz analog kann man dieses Vorgehen auch auf die Korrelationsfunktion der Energiedichte und den Fall antiperiodischer Randbedingungen übertragen: allerdings wird der Bereich verwendbarer Punkt kleiner ausfallen, da hier auch die Korrelationslängen wesentlich kleiner sind.

Die Bestimmung der Korrelationslängen nach der „Differenzenmethode“ (4.14) hat weiterhin rechentechnische Vorteile: da man direkt das exponentielle Verhalten untersucht, sind sowohl die Vorfaktoren G^0 der Exponentialfunktion als auch die residualen Terme $\langle s\rangle^2$ bzw. $\langle \epsilon\rangle^2$ ohne Belang, so daß man sich ihre Bestimmung und den damit verbundenen Rechenaufwand ersparen kann. Angesichts dieser Vorteile werden wir im folgenden die Korrelationslängen ausschließlich mit Hilfe der Differenzenmethode bestimmen. Es mag zunächst verwundern, daß diese Methode einen solch „glatten“ Verlauf für die Schätzung von ξ liefert, da sie doch die Bildung von Differenzen $G(i)-G(i-1)$ etc. beinhaltet — ein Vorgang, der für statistisch fluktuierende Größen im allgemeinen schwer zu kontrollierende Schwankungen hervorrufen wird. Von entscheidender Bedeutung ist hier die Art der Korrelationen zwischen den betrachteten Werten der Korrelationsfunktion G: eine Spektralanalyse zeigt, daß der Hauptbeitrag der Korrelationen vom *Zero Momentum Mode*, von der Mode mit unendlich großer Wellenlänge, geliefert wird; m.a.W.: die Korrelationsfunktion $G(i)$ verschiebt sich vornehmlich ohne Formänderung als ganze und diese Korrelationsmode ist auch absolut sehr stark ausgeprägt, so daß die Bildung von Differenzen zwischen einzelnen Werten von G nicht zu erratischen Schwankungen führt. Die höheren Moden der Korrelationen sind demgegenüber relativ schwach ausgeprägt; so erhält man für die Korrelationskoeffizienten benachbarter Werte der Schätzungen $\hat{\xi}(i)$ gemäß (4.14) im allgemeinen lediglich Werte zwischen 5% und 20%.

Die Differenzenmethode gemäß (4.12) und (4.14) läßt sich noch verallgemeinern: wurden bisher direkt benachbarte Werte der Korrelationsfunktion betrachtet, so kann man die Gleichungen ebenso gut für einen beliebigen Abstand $\Delta \neq 1$ zwischen den betrachteten Werten von $G^{c,\|}(i)$ und $G^{c,\|}(i \pm \Delta)$ formulieren. Man erhält:

$$\tilde{\xi}_i = \Delta \left[\ln \frac{\tilde{G}^{c,\|}(i) - \tilde{G}^{c,\|}(i-\Delta)}{\tilde{G}^{c,\|}(i+\Delta) - \tilde{G}^{c,\|}(i)} \right]^{-1} . \tag{4.15}$$

Die Wahl eines $\Delta > 1$ kann sich unter Umständen auf den Verlauf der Schätzungen $\hat{\xi}(i)$ positiv auswirken: Δ beeinflußt den Signal–Rausch–Abstand des Ergebnisses; sind die Varianzen der einzelnen Werte $G(i)$ der Korrelationsfunktionen näherungsweise vom Abstand i unabhängig, so steigt das Signal der Differenzen $G(i)-G(i-\Delta)$ etc. gegenüber den Fehlern mit wachsendem Δ an, der Verlauf der Schätzungen $\hat{\xi}(i)$ erscheint infolgedessen glatter. Dies ist vor allem für die Korrelationsfunktion der Energiedichte von Belang: hier sind die Korrelationslängen sehr klein, so daß die Differenzenmethode für wachsendes i schnell in einen Bereich gelangt, wo der Abfall von $G(i)$ zwischen benachbarten Werten in derselben Größenordnung liegt wie die statistischen Schwankungen, so daß der Logarithmus des Verhältnisses in (4.14) u.U. nicht mehr gebildet werden kann. Wir werden während des Abgleichsprozesses in diesem Kapitel zunächst die meisten Betrachtungen für $\Delta = 1$ durchführen, wobei eine Verallgemeinerung auf $\Delta > 1$ jederzeit sehr leicht möglich ist. Für die Produktionsläufe werden wir dann weiter unten in einigen Fällen ein $\Delta > 1$ wählen, um den Verlauf der Schätzungen $\hat{\xi}(i)$ zu glätten. Da die Schwankungen für $\Delta = 1$

am größten sind, erscheint ein solches Vorgehen von der Vorsicht diktiert: lassen sich alle verwendeten Werkzeuge für $\Delta = 1$ zufriedenstellend handhaben, dann gilt dies erst recht für den glatteren Verlauf der Schätzungen mit $\Delta > 1$.

4.2 Parameter der Simulation

Vor der Produktion sinnvoller Daten müssen einige Parameter der Simulation „vernünftig" gewählt werden. Welche Kombinationen von Parametern jeweils geeignet sind, läßt sich durch Vorsimulationen herausfinden, die die Abhängigkeit der eigentlichen Meßgrößen — also der Korrelationsfunktionen bzw. –längen — von den jeweiligen Kennzahlen explizieren.

Die Simulationen sollen am kritischen Punkt des jeweiligen Systems ausgeführt werden. Dieser ist der Punkt des Phasenübergangs im thermodynamischen Limes $L_i \to \infty \quad \forall i$ (die kritische Temperatur des „Bulk"–Systems). Im Falle des zweidimensionalen Ising–Modells ist er exakt bekannt, so daß alle Simulationen bei derselben (inversen) Temperatur

$$\beta_c = \frac{1}{2} \ln \left(1 + \sqrt{2}\right) \approx 0.440686793509771 \qquad (4.16)$$

erfolgen[1].

Äquilibrierung: Da wir nicht die Dynamik des Systems, sondern das Verhalten im thermodynamischen Gleichgewicht untersuchen wollen, müssen die Systeme vor Beginn der Messungen äquilibriert werden. Zeitreihen charakteristischer Observablen eines Systems im Gleichgewicht zeichnen sich durch Fluktuationen um ihren Mittelwert aus, zeigen jedoch definitionsgemäß keine Entwicklung, ihr Zustand ist stationär. Ausgehend von einer beliebigen Startkonfiguration der Spins wird die Zeitreihe einer solchen Observable sich asymptotisch einem Plateau–Wert nähern. Die Anzahl der zur Äquilibrierung notwendigen Update–Schritte kann dabei von Observable zu Observable variieren; wir werden ein System hier als im Gleichgewicht befindlich betrachten, wenn die Korrelationsfunktionen von Spin und Energie für

[1] Für die dreidimensionalen Systeme kennt man β_c zwar nicht exakt, jedoch mit großer Genauigkeit. Dort werden wir daher der Unsicherheit bzgl. dieses Parameters Rechnung tragen und zwar mit Hilfe der Reweighting–Technik, s.u. Kap. 6.2

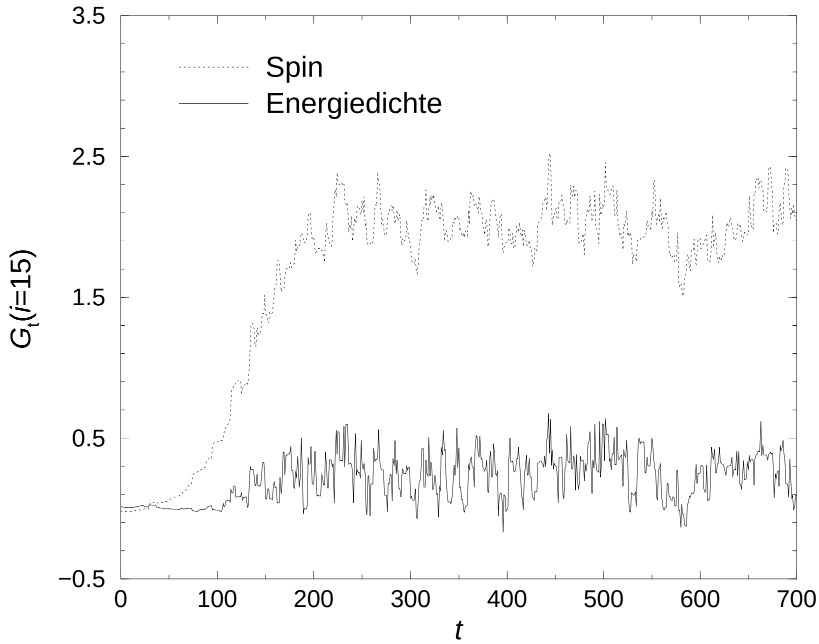


Abb. 4.6: Verlauf der Korrelationsfunktionen $\tilde{G}^{||}_{\sigma}(i)$ und $\tilde{G}^{||}_{\epsilon}(i)$ von lokaler Magnetisierung und Energiedichte für den Abstand $i = 15$ und die ersten 700 Wolff–Updates; es handelt sich wieder um ein System der Größe 15×128 mit periodischen Randbedingungen. An der Tatsache, daß beide Funktionen nahe Null beginnen, zeigt sich, daß das System mit einer gemäß einer Gleichverteilung zufällig gewählten Konfiguration initialisiert wurde. Da die Größe der Cluster des Wolff-Algorithmus variabel ist, hat man es gleichsam mit einer nichtlinearen Zeitskala zu tun.

beliebige Abstände stationär sind: Abb. 4.6 zeigt dies beispielhaft. Für alle Produktionsläufe wurde die Anzahl der Äquilibrierungsschritte gegenüber der anschaulich erkennbaren Äquilibrierungsdauer zur Sicherheit mindestens verdoppelt.

Autokorrelationen und Meßdauer: Wegen der höheren Effizienz vor allem für die eigentlich interessanten Systeme in drei Raumdimensionen wird für alle Simulationen der Wolff–Cluster–Algorithmus verwendet. Da dieser definitionsgemäß pro Update nur einen Teil des Gitters erfaßt, während ein Update–Schritt des Metropolis– oder des Swendsen–Wang–Algorithmus für jeden Spin des Gitters eine neue Orientierung vorschlägt, lassen sich Anzahl und Häufigkeit von Messungen aus Simulationen mit den verschiedenen Algorithmen schlecht zueinander in Beziehung setzen. Um eine gewisse Vergleichbarkeit zu gewährleisten, kann man für Angaben zur Länge der Läufe spezielle Einheiten verwenden, die einen vollständigen Update–

Versuch des Gitters (*Sweep*) durch die Anzahl von Cluster–Updates bestimmt, die bei durchschnittlicher Cluster–Größe das Gitter gerade einmal überstreicht, d.h.

$$1 \text{ Sweep} \doteq \frac{V}{\langle |C| \rangle} \text{ Wolff–Cluster–Updates.} \tag{4.17}$$

Natürlich überschneiden sich aufeinander folgende Cluster beim Wolff–Algorithmus, so daß nach einem Sweep i.a. keineswegs für *jeden* Spin eine neue Orientierung vorgeschlagen wurde, aber die Größenordnungen lassen sich so durchaus vergleichen. Die Anzahl der Sweeps kann man allerdings erst nach der Simulation angeben, wenn die durchschnittliche Cluster–Größe $\langle |C| \rangle$ bekannt ist.

Für die Messungen von größerer Bedeutung ist die Anzahl der Wolff–Updates, die zur Erzeugung einer neuen, unabhängigen Konfiguration nötig ist, also die integrierte Autokorrelationszeit τ_{int} der betrachteten Observablen. Welche Frequenz der Messungen unter Abwägung zwischen der dafür benötigten Rechenzeit und dem Gewinn an unabhängiger Information optimal ist, hängt von der genauen Form der Autokorrelationen und des Simulationsprogramms ab; wir halten uns hier an die Faustregel, nach jeder Messung ungefähr τ_{int} Wolff–Updates ohne Messung verstreichen zu lassen (wobei hier τ_{int} in den ursprünglichen Einheiten der Wolff–Updates anzugeben ist). Da die Autokorrelationszeit von der betrachteten Observable abhängt, es aber relativ umständlich ist, die verschiedenen Beobachtungsgrößen mit unterschiedlicher Frequenz zu messen, wurde für diese Regel immer das größte auftretende τ_{int} verwendet (in unserem Fall ist dabei die Korrelationsfunktion der Energiedichte maßgeblich). Bestimmt wurden die Autokorrelationszeiten mit Hilfe des Binning–Verfahrens wie es in Kap. 3.2.1 beschrieben wurde; eine direkte Bestimmung durch die Definition der normierten Autokorrelationsfunktion und die daraus ableitbare naive Schätzfunktion ist auch möglich und bestätigt die Ergebnisse der Binning–Analyse. Abb. 4.7 zeigt die Ergebnisse einer solchen Binning–Analyse für einige Systeme mit antiperiodischen Randbedingungen. Während die Autokorrelationszeiten für die lokale Magnetisierung mit wachsender Systemgröße monoton ansteigen, sind sie für die Energiedichte unter Berücksichtigung der Begrenztheit der aufgewandten Statistik konstant: der dynamische kritische Exponent z ist hier offenbar sehr klein. Was die Schwankungen in den Schätzungen für τ_{int} angeht, muß man berücksichtigen, daß hier die Autokorrelationen der Korrelationsfunktionen $G(i)$ selbst und nicht die der Magnetisierung bzw. der Energie selbst bestimmt wurden (denn $G(i)$ ist ja auch die später zu messende Observable); da es sich hierbei jedoch gegenüber den globalen Größen um höhere Momente handelt, sind hier auch die Fluktuationen größer. Da

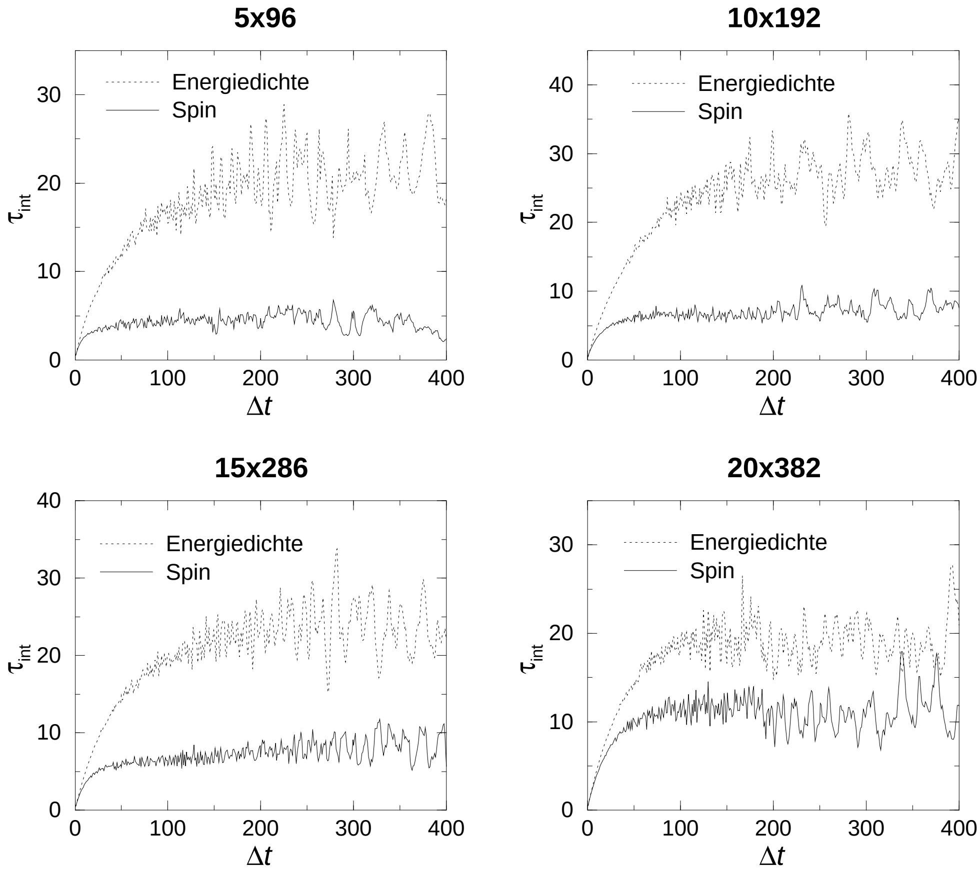


Abb. 4.7: Verlauf der aus einer Binning–Analyse gemäß (3.24) resultierenden Schätzung für die integrierten Autokorrelationszeiten von lokaler Magnetisierung und Energiedichte in Abhängigkeit von der Länge Δt der Bins. Es handelt sich um Systeme mit antiperiodischen Randbedingungen. Die Zeitreihen haben eine Länge von 8000 Wolff–Update–Schritten. Alle Systeme wurden vor der Messung sorgfältig äquilibriert.

für die Frequenz der Messungen das größte τ_{int} maßgeblich ist, kann man sich mit einer konstanten Frequenz von Messungen für alle Systeme zwischen $L_x = 5$ und $L_x = 20$ zufriedengeben; so wurden bei den Simulationen des 2D Ising–Modells mit antiperiodischen Randbedingungen nach jedem 40. Wolff–Update die Korrelationsfunktionen von Spin und Energiedichte bestimmt.

Streifenlänge: Um die Streifengeometrie $S^1 \times \mathbb{R}$ für Simulationen zugänglich zu machen, wurde sie durch ein endliches Gitter $L_x \times L_y$ mit der Bedingung $L_y \gg L_x$ ersetzt. Die Verwendung periodischer Randbedingungen in der langen Richtung führt zu einer Periodizität der Korrelationsfunktionen in y–Richtung mit der Periode L_y;

ist die $\mathbb{R}$–Richtung zu kurz, überlagert der Anstieg der Korrelationsfunktionen für $y > L_y/2$ den Abfall für den betrachteten Bereich $y \leq L_y/2$ und läßt die Korrelationslängen zu klein erscheinen. Die für den Vergleich von L_x und L_y relevante Längenskala ist durch die gemessenen Korrelationslängen von Spin und Energiedichte gegeben; zur Bestimmung eines adäquaten Richtwertes für das Verhältnis L_y/ξ wurde u.a. für ein Beispielsystem mit $L_x = 10$ und periodischen Randbedingungen die Länge L_y zwischen den Werten 50 und 250 durchgestimmt und die berechneten Korrelationslängen miteinander verglichen. Wenn innerhalb der statistischen Fehler eine Vergrößerung von L_y keinen spürbaren Effekt mehr zeigt, kann man das System als ausreichend „lang“ ansehen. Abb. 4.8 zeigt diesen Vergleich für die Spin–Korrelationslänge; da diese mit einer Amplitude $A_\sigma = \xi_\sigma/L_x$ von etwa 1.3 wesentlich größer ist als die Korrelationslänge der Energiedichte mit $A_\epsilon \approx 0.2$, ist sie hier die maßgebliche Größe. Spätestens für das 10×150–System ist offenbar kein Einfluß der Endlichkeit der y–Richtung mehr erkennbar; um auch für eine evtl. verbesserte Statistik bei Produktionsläufen noch auf der sicheren Seite zu sein, wurde die „lange“ Richtung der Systeme (auch später im dreidimensionalen Fall) regelmäßig so gewählt, daß $L_y/\xi_\sigma \approx 15$ ist (ein in vielen Simulationen verwendeter Wert ist hier 8). Um die vorhandene Rechenleistung optimal auszunutzen, wurden daher die Längen L_y der Systeme linear in der transversalen Ausdehnung L_x, die ihrerseits proportional zu den Korrelationslängen ist, skaliert:

$$\frac{L_y}{\xi} = \frac{L_y}{AL_x} \stackrel{!}{=} \text{const.} \Longrightarrow L_y \propto L_x. \tag{4.18}$$

Der hierfür notwendige Richtwert für die Amplitude A_σ im Falle periodischer Randbedingungen wurde mit 1.3 angenommen und stammt aus Vorsimulationen. Für antiperiodische Randbedingungen erhält man $A_\sigma \approx 0.4$ und $A_\epsilon \approx 0.08$; da sich die Rechenzeiten für zweidimensionale Systeme in Grenzen halten, wurden hier die Systeme für beide Arten von Randbedingungen gleich dimensioniert, so daß der Wert $A_\sigma \approx 1.3$ ausschlaggebend war. In drei Dimensionen wurde dann jedoch das in einer „Verkürzung“ der Systeme für antiperiodische Randbedingungen angelegte Potential zur Verbesserung der Statistik ausgenutzt und die Systeme für periodische und antiperiodische Randbedingungen unabhängig voneinander dimensioniert.

Insgesamt erfordert die Bestimmung der relevanten Parameter zur Simulation eines Systems die folgenden Schritte:

1. Die Simulation eines „hinreichend langen“ (überdimensionierten) Systems und die nachfolgende Bestimmung der Korrelationslängen liefert Schätzwerte für

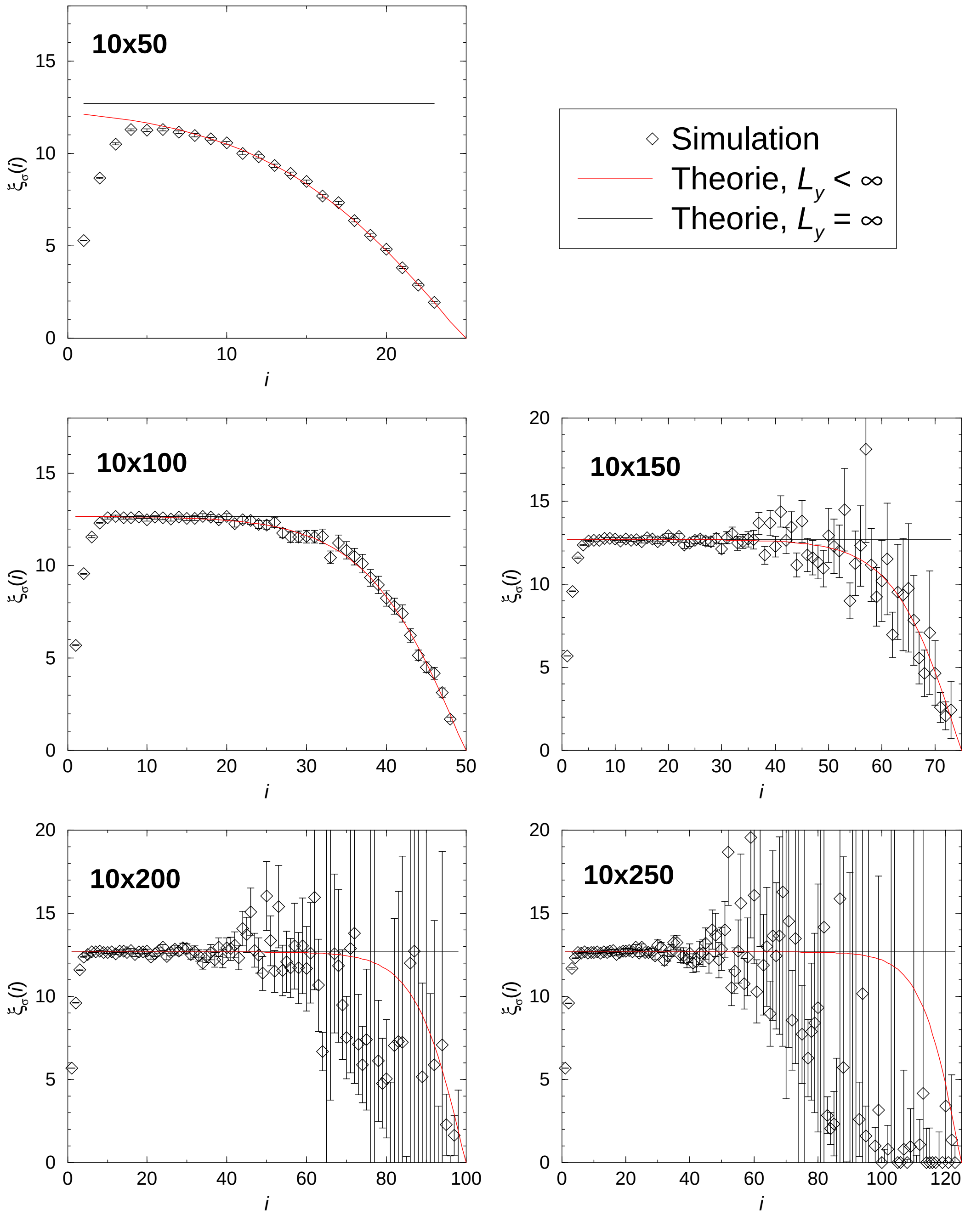


Abb. 4.8: Einfluß der endlichen Länge L_y der simulierten Streifen auf die gemessene Korrelationslänge (hier) der lokalen Magnetisierung. Die waagrechten Linien symbolisieren die für $L_y = \infty$ zu erwartenden Werte, wobei Korrekturen zum FSS wie sie in Kap. 5.3 beschrieben werden schon berücksichtigt wurden — es handelt sich um die exakten Lösungen des Problems für endliche Längen L_x. Die roten Kurven zeigen das Ergebnis der Anwendung der „Differenzenmethode“ auf eine Korrelationsfunktion der Form $G(i) = exp(-i/\xi) + exp(-(L-i)/\xi)$ wie sie die Verwendung periodischer Randbedingungen in einer endlichen y–Richtung unter Vernachlässigung des „cos“–Terms und des asymptotisch verschwindenden Anteils des „cosh“–Terms in Gl. (2.70) impliziert. Etwa ab der Länge $L_y = 150$ findet der so begründete Abfall der ξ–Schätzung in einem Bereich statt, der wegen der extremen Fluktuationen in den Schätzwerten für ξ für die Mittelung ohnehin nicht mehr taugt.

die Amplituden A_σ und A_ϵ in Abhängigkeit von der Wahl der Randbedingungen.

2. Wähle nun die Systemgrößen so, daß $L_y = 15 A_\sigma L_x$, wobei A_σ getrennt nach Randbedingungen zu verwenden ist.
3. Bestimme für jedes dieser so dimensionierten Systeme durch Simulation die notwendige Äquilibrierungszeit durch Analyse der Zeitreihen der Korrelationsfunktionen von lokaler Magnetisierung und Energiedichte.
4. Bestimme die Autokorrelationszeiten τ_{int} beider Observablen durch eine Binning–Analyse der Zeitreihen der äquilibrierten Systeme.
5. Dann können die Produktionsläufe der in 2. bestimmten Systeme mit einer Äquilibrierungszeit gemäß 3. und einer Frequenz der Messungen von $1/\tau_{\mathrm{int,max}}$ gemäß 4. gestartet werden.

4.3 Wahl des Zufallszahlen–Generators

Monte–Carlo–Simulationen sind sowohl bei der Generierung der bei einem Update–Versuch vorgeschlagenen neuen Konfiguration als auch bei der Entscheidung über die Annahme derselben auf eine Zufallsquelle angewiesen, die einigen Anforderungen genügen muß. Echte „Zufälligkeit" (z.B. im Sinne von prinzipieller Unvorhersagbarkeit) läßt sich von der Softwareseite her nicht generieren, ist jedoch auch für eine MC–Simulation ohne Belang: man muß zu sog. *Pseudo–Zufallszahlen–Generatoren* Zuflucht nehmen [52]. Für Simulationen relevante Eigenschaften der von diesen Algorithmen generierten Zahlenfolgen sind neben der Erzeugung der richtigen Verteilung (stochastische Konvergenz des Histogramms an die Verteilungsfunktion) vor allem Parameter, die die Verteilung der generierten Folgenglieder in ihrer zeitlichen Abfolge charakterisieren, insbesondere Zwei– oder n–Punkt–Autokorrelationsfunktionen. Zum Test solcher Zufallszahlen–Generatoren (ZZG) auf die gewünschten Eigenschaften sind verschiedene Verfahren entwickelt worden; einen Überblick gibt [84]. Neben den eigentlichen statistischen Tests werden noch Tests auf der Bitebene der Zahlen verwendet und sog. *Application Tests*, die die Güte des Generators anhand von Ergebnissen eines Simulationsprogramms beurteilen, das diesen verwendet. Generell stellt die Wahl des ZZG einen Kompromiß zwischen der Optimierung gewisser Kenngrößen der Güte der erzeugten Zufallszahlen und der Minimierung der zur Erzeugung der Zahlen notwendigen Rechenoperationen dar; die Aufwendigkeit des verwendeten Algorithmus ist relevant, da zur Erzeugung einer unabhängigen Konfiguration eines simulierten Systems in d Dimensionen mit dem Wolff–Cluster–Algorithmus etwa $d \langle |C| \rangle \tau_{\text{int}}$ Zufallszahlen benötigt werden — für große Systeme in drei Dimensionen können das durchaus 10^6 Werte sein. Da alle Algorithmen zur Erzeugung von (Pseudo–)Zufallszahlen in einem diskreten Zahlenraum operieren und deterministisch sind, wiederholen sich die generierten Zahlenfolgen nach einer bestimmten Zeit, der Periode des Generators. Für Simulationen, die sehr viele Zufallszahlen benötigen, ist daher die Länge der Periode ein wichtiges Entscheidungskriterium bei der Wahl des Zufallsgenerators.

Ein ZZG mit sehr guten statistischen Eigenschaften und akzeptabler Effizienz ist durch den sog. „R250“ gegeben [59]. Es handelt sich um einen *Lagged–Fibonacci*–Generator, bei denen ein neues Glied z_t der (Pseudo–)Zufallszahlen–Folge durch eine binäre Verknüpfung (mindestens) zweier zuvor in der Folge aufgetretener Glieder z_{t-r} und z_{t-s} mit festen Abständen (*Lag*) r und s erzeugt wird; im Falle des R250 ist die Verknüpfung als bitweise XOR–Operation realisiert, man spricht dann von einem *Shift Register* Generator:

$$z_t = z_{t-r} \oplus z_{t-s}. \tag{4.19}$$

Die XOR–Operation $\oplus$ hat den Vorteil, daß eine zeitaufwendige Modulo–Operation zur Rückfaltung des Ergebnisses in den Wertebereich entfallen kann. Im speziellen Fall des R250 ist $r = 250$ und $s = 103$, daher der Name. Die Periode eines solchen Generators ist gegeben durch $2^r - 1$ [59], hier also $2^{250} - 1 \approx 10^{75}$, so daß bei den heutigen Rechenleistungen kaum Gefahr besteht, diese zu erschöpfen. Der R250 schneidet bei den meisten der gängigen Gütetests hervorragend ab [84] und hat daher — und wegen seiner hohen Effizienz — im Bereich der Simulationen und in der statistischen Physik weite Verbreitung gefunden.

Bei einem *Application Test* in Form einer MC–Simulation des 2D Ising–Modells auf einem quadratischen Gitter der Größe 16×16 unter Verwendung des Wolff–Cluster–Algorithmus traten jedoch hoch signifikante Abweichungen in den Ergebnissen für die Energie und die spezifische Wärme des Modells gegenüber den bekannten exakten Werten für diese Größen auf [35]. Nachfolgende Untersuchungen ergaben, daß diese Abweichungen mit bestimmten, für Schiebe–Register–Generatoren typischen Triplet–Korrelationen in der Zeitreihe der Zufallszahlen zusammenhängen [25]; wie genau diese Korrelationen die beobachteten Abweichungen etwa von Energie und spezifischer Wärme erzeugen, ist jedoch nicht bekannt [49]. In einer systematischen Studie [78] des Verhaltens dieser Abweichungen über verschiedene (quadratische) Gittergrößen hinweg konnte inzwischen gezeigt werden, daß es bestimmte Gitter gibt, bei denen die Abweichungen maximal werden — nämlich dann, wenn die durchschnittliche Cluster–Größe $\langle |C| \rangle$ mit dem *Lag* r zusammenfällt — und daß die Abweichungen bei einer Annäherung an diese Maxima ein Skalenverhalten zeigen, wie man es aus der Theorie des FSS kennt. Für die vorliegende Arbeit von Interesse sind die offenen Fragen, inwieweit sich diese Ergebnisse auf die hier verwendeten anisotropen Geometrien, auf die Korrelationsfunktion der Energiedichte (die ja eine „lokale spezifische Wärme“ als Spezialfall enthält) als Observable und auf dreidimensionale Systeme übertragen lassen.

Sind die beobachteten Abweichungen Folgen der theoretisch verstandenen Drei–Punkt–Korrelationen des R250, so liegt der Versuch nahe, diese zu eliminieren, ohne die Vorteile solcher Schiebe–Register–Generatoren aufgeben zu müssen. Dafür sind verschiedene Vorschläge gemacht worden [49]; die effizienteste Methode besteht in einer Verknüpfung *zweier* solcher Generatoren mit unterschiedlichen *Lags*, indem die Folgenglieder z_t^1 und z_t^2 der Generatoren nochmals durch eine XOR–Operation verbunden werden und so die zu verwendende Zufallszahl $z_t^1 \otimes z_t^2$ erzeugen. Diese Prozedur vernichtet die Triplet–Korrelationen des Generators, es verbleiben lediglich Neun–Punkt–Korrelationen, die natürlich um Größenordnungen gegenüber den Drei–Punkt–Korrelationen des R250 unterdrückt sind. Der spezielle Fall $r_1 = 250$ und $r_2 = 521$ sei hier in Anlehnung an [49] mit „R250/521“ bezeichnet.

Als Referenz, gegen die die beiden Schiebe–Register–Generatoren getestet werden können, wurde als dritter ZZG der sog. „RANLUX“–Generator [62] implementiert. Er basiert auf einem sog. *Subtract–with–Borrow*–Algorithmus, einer Art modifiziertem Lagged–Fibonacci–Generator mit einer Subtraktion als Verknüpfung und der Addition eines zusätzlichen *Carry–Bit*, das durch den logischen Wahrheitswert einer Ungleichung gegeben ist. Man weiß, daß die Korrelationen dieses ursprünglichen Generators kurzreichweitig sind; daher, so die Idee, lassen sie sich eliminieren, indem nicht *jede* generierte Zufallszahl weiter verwendet wird, sondern lediglich jede p–te. Man kann zeigen [62], daß für $p \geq 389$ alle Bits der generierten Zahlen voneinander unabhängig chaotisch fluktuieren (im Sinne der nichtlinearen Dynamik), so daß die Zufallszahlen–Folge hervorragende statistische Eigenschaften aufweist. Diesen „Luxus“ muß man aber mit einem entsprechend hohen Rechenaufwand zur Erzeugung einer „verwendbaren“ Zufallszahl bezahlen. Im Vergleich mit dem schnellsten

Größe	exakt	R250	R250/521	RANLUX
E/V	1.4142136	1.414087(63)	1.414206(59)	1.414253(60)
Abw.		-2.0σ	-0.14σ	$+0.65\sigma$
c	1.3259279	1.3569(16)	1.3255(16)	1.3269(13)
Abw.		$+19.4\sigma$	-0.27σ	$+0.75\sigma$

Tab. 4.1: Vergleich von Ergebnissen für die innere Energie und die spezifische Wärme eines 10×192–Systems mit periodischen Randbedingungen aus Simulationen mit verschiedenen Zufallsgeneratoren.

der drei Generatoren, dem R250, braucht das gesamte Simulationsprogramm bei Verwendung des RANLUX–Generators etwa 250% der Rechenzeit, die das Simulationsprogramm mit dem R250 benötigt; die Modifikation des R250 zum R250/521 dagegen schlägt (wiederum bezogen auf die gesamte Simulation) lediglich mit einem um 25% erhöhten Zeitbedarf zu Buche. Daher ist der RANLUX–Generator bei dem hier vorliegenden Bedarf an Zufallszahlen lediglich für Testzwecke geeignet.

Für das Ising–Modell in zwei Dimensionen und mit periodischen Randbedingungen auf einem Rechteck–Gitter stehen zum Vergleich (auf Maschinen–Genauigkeit) exakte Ergebnisse für die innere Energie und die spezifische Wärme zur Verfügung [53]; sie wurden durch numerische Differentiation der von Ferdinand und Fisher [32] angegebenen Formel für die freie Energie eines solchen Systems gewonnen. Simulationen für die hier betrachtete Geometrie unter Verwendung des R250 bestätigen die auffälligen Abweichungen auch für anisotrope Systeme; für ein Beispielsystem mit $(L_x, L_y) = (10, 192)$ und $2 \cdot 10^6$ Messungen liegt die berechnete Energie etwa 2σ, die spezifische Wärme sogar 20σ neben dem exakten Wert, vgl. Tab. 4.1. Dabei ist zu beachten, daß in diesem Beispiel die durchschnittliche Cluster–Größe mit $\langle|C|\rangle \approx 159$ vom *Lag* $r = 250$ und damit dem Maximum der Abweichungen noch weit entfernt ist. Das gute Abschneiden des modifizierten Schiebe–Register–Generators R250/521 zeigt, daß mit den Triplet–Korrelationen die Ursache der systematischen Fehler effektiv beseitigt wurde. Der RANLUX–Generator liefert erwartungsgemäß ebenfalls konsistente Ergebnisse.

Für antiperiodische Randbedingungen liegen unseres Wissens keine exakten Daten für die freie Energie und damit die abgeleiteten Größen E/V und c vor; daher muß hier ein Vergleich der Generatoren untereinander genügen. Es wurde der gewichtete Mittelwert der Ergebnisse der Simulationen mit dem RANLUX–Generator und dem

Größe	Mittel	R250	R250/521	RANLUX
E/V	1.214247(34)	1.214173(47)	1.214277(47)	1.214215(48)
Abw.		-1.6σ	$+0.6\sigma$	-0.7σ
c	0.69808(82)	0.70452(85)	0.69817(86)	0.69808(82)
Abw.		$+7.5\sigma$	$+0.06\sigma$	-0.05σ

Tab. 4.2: Vergleich von Ergebnissen für die innere Energie und die spezifische Wärme eines 10×192–Systems mit antiperiodischen Randbedingungen aus Simulationen mit verschiedenen Zufallsgeneratoren. Die Werte in der Spalte „Mittel“ entsprechen dem varianzgewichteten Mittel der Ergebnisse des R250/521 und des RANLUX–Generators. Die Abweichungen sind auf diese Mittelwerte bezogen.

R250/521 als Vergleichswert angenommen, an dem sich der R250 zu messen hatte. Tab. 4.2 zeigt die Ergebnisse einer solchen Simulation für das 10×192–System; die sehr gute Übereinstimmung der Werte aus den RANLUX– und R250/521–Simulationen rechtfertigt ihre Verwendung als Referenz. Auch hier zeigen sich bei Verwendung des R250 signifikante Abweichungen in den Ergebnissen für die innere Energie und (vor allem) die spezifische Wärme; die gegenüber dem periodischen Fall kleineren Abweichungen sind angesichts der Beobachtungen in [78] vermutlich nicht auf Randbedingungen selbst zurückzuführen, sondern auf die hier um etwa einen Faktor 2 auf $\langle|C|\rangle \approx 83$ reduzierten durchschnittlichen Cluster–Größen, so daß man vom Maximum der Abweichungen bei $\langle|C|\rangle = 251$ noch weiter entfernt ist.

Wie verhalten sich andere Observablen? Die Ergebnisse für die Magnetisierung und die magnetische Suszeptibilität sind — in Übereinstimmung mit den früheren Beobachtungen — offenbar nicht betroffen. Die Korrelationsfunktionen der Energiedichte aus Simulationen mit dem R250 im Vergleich zu solchen mit den anderen beiden Generatoren erscheinen gegeneinander um einen weitgehend konstanten Offset verschoben — wohl ein Effekt der (wie oben gesehen) vom R250 falsch geschätzten inneren Energie des Systems. Die hier verwendete Analyse der Daten mit Hilfe der „Differenzenmethode“ eliminiert jedoch eine solche Verschiebung, so daß lediglich Abweichungen in der *Schwankung* der Energiedichte sichtbar sein sollten. Tab. 4.3 zeigt die Mittelwerte für ξ_σ und ξ_ϵ des 10×192–Systems mit beiden Arten von Randbedingungen aus Simulationen mit den verschiedenen Generatoren. Das Ergebnis ist uneinheitlich; immerhin zeigen sich bei zwei Größen in den Simulationen mit dem R250 signifikante Abweichungen, wobei eines davon die Spin–Korrelationslänge betrifft

RB	Größe	Mittel/exakt	R250	R250/521	RANLUX
period. RB	ξ_σ	12.678845	12.7092(92)	12.6787(74)	12.6789(85)
	Abw.		$+3.3\sigma$	-0.02σ	$+0.006\sigma$
	ξ_ϵ	1.5812(35)	1.5796(46)	1.5851(48)	1.5770(50)
	Abw.		-0.3σ	$+0.8\sigma$	-0.8σ
antiperiod. RB	ξ_σ	4.2906(28)	4.2878(43)	4.2925(39)	4.2887(40)
	Abw.		-0.7σ	$+0.5\sigma$	-0.5σ
	ξ_ϵ	0.8118(58)	0.7803(84)	0.8110(82)	0.8126(83)
	Abw.		-3.8σ	-0.1σ	$+0.1\sigma$

Tab. 4.3: Mittelwerte für die Korrelationslängen von Spin und Energiedichte eines 10×192–Systems aus Simulationen mit verschiedenen ZZG. Für die Bestimmung dieser Mittelwerte wurde das weiter unten in Kap. 4.4.3 beschriebene Verfahren mit einem konstanten Bereich $i_{\rm min},\ldots,i_{\rm max}$ verwendet. Für ξ_σ des Systems mit periodischen Randbedingungen ist der exakte Wert als Bezugsgröße angegeben, in den anderen Fällen dient das varianzgewichtete Mittel der Ergebnisse der Simulationen mit RANLUX und R250/521 als Referenz.

und aufgrund des hier vorliegenden exakten Wertes eine besondere Aussagekraft hat. Eine eindeutige Aussage über das Verhalten etwa auftretender Abweichungen in Abhängigkeit von den Randbedingungen und der beobachteten Größe erforderte eine systematische Analyse mit einer Mittelung über mehrere Simulationen desselben Systems mit unterschiedlichen Initialisierungen des Zufallsgenerators. Was die Abhängigkeit von der Größe des Systems angeht, so kommt es hier möglicherweise aufgrund der Projektion der „Schichtenmethode“ nicht mehr ausschließlich auf die gesamte durchschnittliche Cluster–Größe, sondern vielmehr auf die Ausdehnung der Cluster in der y–Richtung an. Immerhin läßt sich festhalten, daß ein Auftreten systematischer Abweichungen auch in den hier betrachteten Observablen bei Verwendung des R250 nicht unwahrscheinlich und daher von der Verwendung des R250 in einer solchen Simulation abzuraten ist. Andererseits stimmen die Ergebnisse der Simulationen mit dem modifizierten Schiebe–Register–Generator R250/521 mit denen des nachweislich besonders verläßlichen RANLUX innerhalb der Fehler sehr gut miteinander und im Falle der Spin–Korrelationslänge des Systems mit periodischen Randbedingungen auch hervorragend mit der exakten Lösung überein; daher kann man den R250/521 wohl guten Gewissens für unsere Zwecke verwenden, was wir auch ausschließlich tun werden.

Größe	Mittel	R250	R250/521	RANLUX
ξ_σ	7.781(74)	7.81(11)	7.88(11)	7.70(10)
Abw.		$+0.3\sigma$	$+0.9\sigma$	-0.8σ
ξ_ϵ	2.089(44)	2.093(57)	2.072(62)	2.108(64)
Abw.		$+0.2\sigma$	-0.3σ	$+0.3\sigma$

Tab. 4.4: Vergleich von Ergebnisse für die Korrelationslängen von Spin und Energiedichte eines 3D Ising–Systems mit $L_x = L_y = 10$, $L_z = 120$ und periodischen Randbedingungen aus Simulationen mit verschiedenen ZZG. Die Mittelwerte der ersten Spalte sind aus den Werten für R250/521 und RANLUX gebildet.

Eine Simulation für ein Ising–Modell auf der *dreidimensionalen* Geometrie $L_x \times L_y \times L_z$ mit $L_x = L_y$ und $L_z \gg L_x$ mit periodischen Randbedingungen schließlich zeigt innerhalb der Fehler keine signifikanten Unterschiede zwischen den drei Generatoren, vgl. Tab. 4.4. Man macht daher wohl zumindest keinen Fehler, wenn man den R250/521 als einen Kompromiß zwischen der Aufwendigkeit des Algorithmus und Güte der erzeugten Zufallszahlen auch für die Simulationen in drei Dimensionen verwendet.

4.4 Fehlerrechnung

Die Anwendung der „Differenzenmethode“ zur Bestimmung der Korrelationslängen aus den Korrelationsfunktionen ermöglicht eine zuverlässige Abschätzung des linearen Bereichs eines logarithmischen Graphen der Korrelationsfunktionen und korrigiert die Ergebnisse um den Einfluß schlecht geschätzter additiver und multiplikativer Konstanten in der Korrelationsfunktion. Der Preis hierfür besteht in den hohen Anforderungen an die Methode der Berechnung der *Fehler* der so gewonnenen Größen $\hat{\xi}(i)$: es müssen Varianzen von hochgradig nichtlinearen Funktionen von Observablen geschätzt werden. Sind die Fehler der $\hat{\xi}(i)$ bestimmt, so muß entschieden werden, wie aus aus diesen Einzelwerten ein endgültiger Mittelwert $\bar{\xi}$ zu gewinnen ist und auch für diesen sollte wieder eine verläßliche Fehlerangabe gemacht werden können.

4.4.1 Fehlerbestimmung der $\hat{\xi}(i)$

Für nichtlineare Funktionen von Observablen ist gem. Kap. 3.2.2 der „naive“ Ansatz der Fehlerrechnung (Gl. 3.38) ungeeignet. Stattdessen bietet sich das Jackknifing als leistungsfähiges Werkzeug für ein solches Problem an. Voraussetzung für eine Jackknife–Analyse ist zunächst das Vorliegen von Zeitreihen mit verschwindenden Autokorrelationen. Dies wird durch ein Binning der Messungen der Observablen während des Simulationslaufs erreicht. Zwar wurde die Frequenz der Messungen schon so gewählt, daß die einzelnen Messungen statistisch weitgehend unabhängig sein sollten; um die Korrelationen noch weiter zu reduzieren, vor allem aber um die anfallenden Datenmengen auf ein erträgliches Maß zu reduzieren, wurden jeweils zwischen 2048 und 8192 Messungen zu Blöcken zusammengefaßt, was ein Vielfaches des eigentlich notwendigen Wertes darstellt, insofern das Plateau einer Binning–Analyse schon nach wesentlich weniger Block–Transformationen augenscheinlich erreicht wird, vgl. Abb 4.9.

Die so gewonnenen, statistisch nahezu unabhängigen Ausgangswerte für die Jackknife–Analyse seien in diesem Abschnitt mit $G_t(i)$ bezeichnet, womit der t. Binning–Block für den Wert der Korrelationsfunktion G für den Abstand $i = |y_2 - y_1|$ gemeint ist; zusätzlich kann bei Bedarf durch Indizes σ oder ϵ zwischen lokaler Magnetisierung und Energiedichte unterschieden werden. Aus diesen $L_y/2$ Zeitreihen der $G(i)$ für die verschiedenen Distanzen $i = 0, \ldots, L_y/2 - 1$ werden nun zum Zwecke der Varianz–Analyse Jackknife–Blöcke gebildet (vgl. Gl. (3.33)):

$$\begin{aligned} G_{(s)}(i) &= \frac{1}{n-1} \sum_{t \neq s} G_t(i), \\ G_{(\cdot)}(i) &= \frac{1}{n} \sum_s G_{(s)}(i), \end{aligned} \tag{4.20}$$

wo n die Länge der Zeitreihen, d.h. hier die Anzahl der von der anfänglichen Binning–Analyse erzeugten Blöcke, bezeichnet. Bei Verwendung der „Differenzenmethode“ ergeben sich daraus entsprechende Jackknife–Block–Schätzungen für die Korrelationslängen:

$$\begin{aligned} \hat{\xi}_{(s)}(i) &= \left[\ln \frac{G_{(s)}(i) - G_{(s)}(i-1)}{G_{(s)}(i+1) - G_s(i)} \right]^{-1}, \\ \hat{\xi}_{(\cdot)}(i) &= \frac{1}{n} \sum_s \hat{\xi}_{(s)}(i). \end{aligned} \tag{4.21}$$

Unter Verwendung der Gleichungen aus Kap. 3.2.2 läßt sich daher für die Varianz

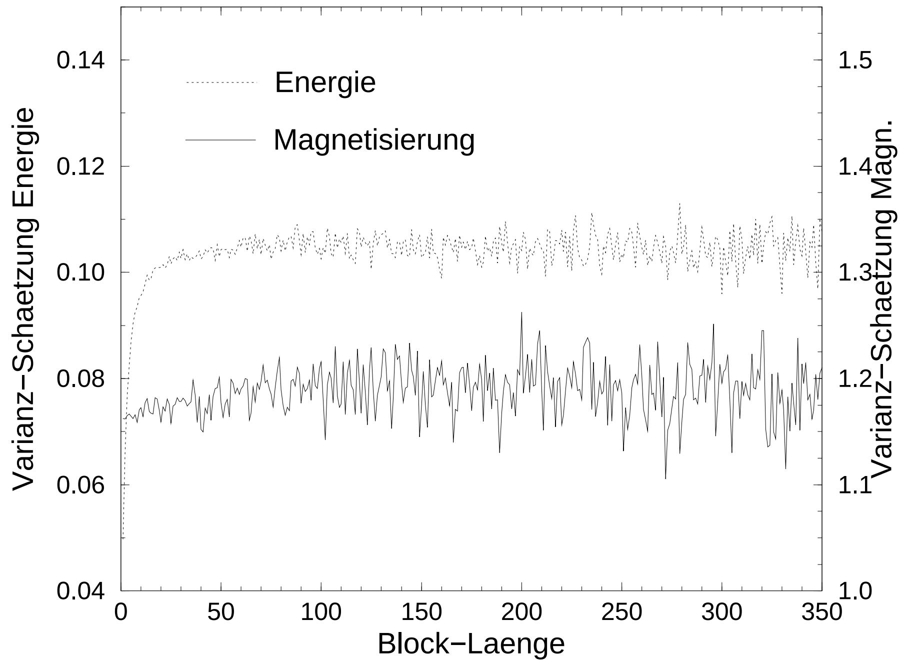


Abb. 4.9: Verlauf der Varianz–Schätzung für Energie und Magnetisierung eines Beispielmodells mit $L_x = 10, L_y = 192$ und periodischen Randbedingungen in Abhängigkeit von der Länge der verwendeten Bins oder Blöcke. Die Frequenz der ursprünglichen Messungen war dabei etwa $1/\tau_{\text{int}}$. Bei den Produktionsläufen wurde die Bin–Länge aus Speicherplatz–Gründen schon während des Runs auf einen Wert zwischen 2048 und 8192 festgelegt; Unabhängigkeit der Blöcke ist dabei offenbar mit hervorragendem „Sicherheitsabstand" gewährleistet.

der $\hat{\xi}(i)$ folgende Schätzung angeben:

$$\widehat{\mathrm{VAR}}(\hat{\xi}(i)) = \frac{n-1}{n} \sum_{s=1}^{n} \left(\hat{\xi}_{(s)}(i) - \hat{\xi}_{(\cdot)}(i) \right)^2. \tag{4.22}$$

Mit dieser Methode sind die Fehlerbalken aller vorstehenden Abbildungen von Ergebnissen der „Differenzenmethode" erzeugt worden; ein einfacher optischer Vergleich dieser Angaben mit dem Maß der augenscheinlichen Schwankungen der $\hat{\xi}(i)$ über die Abstände i hinweg zeigt zumindest, daß die Jackknife–Methode „vernünftige" Ergebnisse liefert — eine genauere Verifikation durch einen Vergleich mit einer anderen Methoden der Fehlerrechnung wird weiter unten in Kap. 4.4.4 erfolgen.

4.4.2 Bias–Reduktion der Schätzungen

Wie Kap. 3.2.2 gesehen, bietet die Jackknife–Methode nicht nur einen Schätzer für die Varianz nichtlinearer Funktionen von Observablen an, sondern auch einen Ansatz zur Bias–Reduktion solcher Schätzungen. Ist dieser — wie für alle nicht–pathologischen Verteilungen zu erwarten — von der Ordnung $O(1/n)$, so ist ein biasbereinigter Schätzer durch Gl. (3.35) gegeben. Im Sinne dieses Ansatzes lassen sich in erster Ordnung um den Bias korrigierte Schätzer für die Korrelationslängen $\hat{\xi}(i)$ angeben:

$$\tilde{\xi}(i) = n\hat{\xi}(i) - (n-1)\hat{\xi}_{(\cdot)}(i). \tag{4.23}$$

Ob eine solche Bias–Korrektur notwendig ist, hängt dabei in erster Linie von der Anzahl der verwendeten Bins und daher — bei einer festen Größe der Blöcke — von der Laufzeit der Simulationen ab; die beste „Bias–Reduktion“ liegt natürlich in einer möglichst langen Zeitreihe, da alle verwendeten Schätzer zumindest im Limes $n \to \infty$ erwartungstreu sind.

Abb. 4.10 zeigt einen Vergleich der Ergebnisse der „Differenzenmethode“ mit und ohne Anwendung einer solchen Bias–Reduktion. Die Schätzung (4.23) selbst stellt dabei keine wesentlichen zusätzlichen Anforderungen an den Analyseaufwand dar; problematisch ist lediglich die Bestimmung der Fehler solcher biasbereinigter Größen: da der Bias selbst eine fluktuierende Größe darstellt, muß er in die Fehlerbetrachtung mit einbezogen werden; dies läßt sich durch Verwendung von sog. „*Double Jackknife Bias Corrected Estimators*“ erreichen, wie sie in Kap. 3.2.2 beschrieben wurden. Mit diesem Problem werden wir uns weiter unten in Kap. 4.4.6 genauer beschäftigen. Die obige Abbildung zeigt daher nur für die ursprüngliche Reihe der $\hat{\xi}(i)$ Fehlerbalken, damit das Verhältnis der Bias–Reduktion zu den statistischen Fehlern beurteilt werden kann. Im Plateau–Bereich beträgt die Korrektur etwa $10^{-2}\%$ von $\hat{\xi}(i)$ selbst und ca. 1% von der Schätzung $\widehat{\mathrm{VAR}}$ für den Fehler von $\hat{\xi}(i)$.

4.4.3 Bestimmung von $\bar{\xi}$: Fehlerrechnung

Für jeden longitudinalen Abstand $i = |y_2 - y_1|$ im Plateau–Bereich liefert die Funktion $\hat{\xi}(i)$ eine Schätzung für die gesuchte Korrelationslänge ξ. Um den maximalen Informationsgehalt dieser Daten zu extrahieren, wird man daher über die verschiedenen Abstände i des Plateaus mitteln wollen. Die Frage, *welche* Abstände dies im Einzelfall sein sollten, d.h. wie der Plateau–Bereich abgeschätzt werden kann,

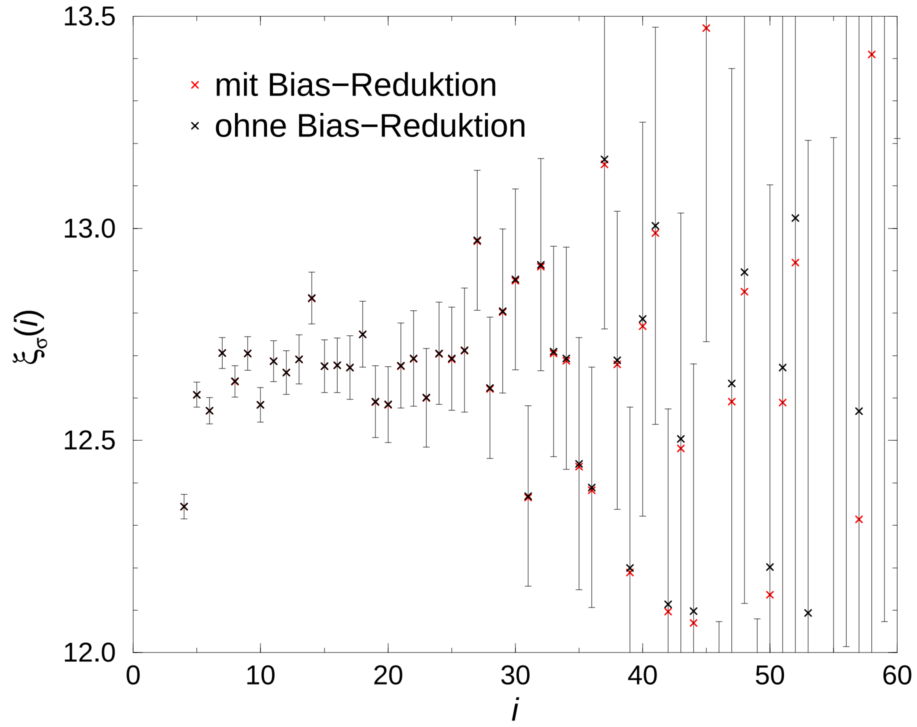


Abb. 4.10: Effekt einer Jackknife–Bias–Reduktion auf die Schätzungen $\hat{\xi}_\sigma(i)$ eines 10×192–Beispielsystems mit periodischen Randbedingungen. Die Ausgangsdaten bestanden aus 256 Bins à 8192 Messungen. Die Fehlerbalken entsprechen den aus der Jackknife–Analyse resultierenden Fehlern für die Schätzungen $\hat{\xi}$ ohne Bias–Korrektur. Die Bias–Korrektur entspricht für den Spin etwa $5 \cdot 10^{-3}\%$ von ξ_σ und ca. 0.5% der Fehlerbalken; für die Korrelationslänge der Energiedichte ist die relative Korrektur etwa doppelt so groß.

wird weiter unten in Kap. 4.4.5 abgehandelt; hier geht es um die Entscheidung, *wie* über eine gegebene Zahl von Punkten $\{\hat{\xi}(i)\,|\,i = i_{\rm min}, \ldots, i_{\rm max}\}$ gemittelt werden sollte. Hat der Mittelwert die allgemeine Form

$$\bar{\xi} = \sum_{i=i_{\rm min}}^{i_{\rm max}} \alpha_i \hat{\xi}(i), \qquad \sum_{i=i_{\rm min}}^{i_{\rm max}} \alpha_i = 1, \tag{4.24}$$

so geht es um die Frage, wie die Gewichte α_i der $\hat{\xi}(i)$ gewählt werden sollen. Das relevante Kriterium für eine optimale Wahl der α_i ist dabei augenscheinlich die theoretische Varianz des gebildeten Mittelwertes; ist diese minimal, so ist auch die Wahrscheinlichkeit klein, daß die vorliegende Schätzung weit vom theoretischen Erwartungswert entfernt ist. Es handelt sich um ein einfaches Variationsproblem mit

Nebenbedingung, den Satz von α_i zu finden, der die Varianz von $\bar{\xi}$ minimiert:

$$\begin{aligned} \sigma^2(\bar{\xi}) = \sigma^2(\sum_{i=i_{\min}}^{i_{\max}} \alpha_i \hat{\xi}(i)) &= \min, \\ \sum_{i=i_{\min}}^{i_{\max}} \alpha_i &= 1 \end{aligned} \tag{4.25}$$

Die Varianz von $\bar{\xi}$ hängt von den Verteilungen der $\xi(i)$ ab (hiermit seien im Gegensatz zu den Schätzungen $\hat{\xi}$ die zugehörigen Zufallsvariablen selbst bezeichnet). Da sich alle $\xi(i)$ auf dem Plateau befinden sollen, ist $\langle \xi(i) \rangle = \xi,\ i = i_{\min}, \ldots, i_{\max}$.

Betrachten wir zunächst den Fall von *unabhängigen* Zufallsvariablen $\xi(i)$ mit Varianzen $\sigma_i^2 \equiv \sigma^2(\xi(i))$:

$$\begin{aligned} \sigma^2(\bar{\xi}) &= \langle \left(\sum_i \alpha_i \hat{\xi}(i) \right)^2 \rangle - \langle \sum_i \alpha_i \hat{\xi}(i) \rangle^2 \\ &= \sum_{i,j} \alpha_i \alpha_j \left(\langle \hat{\xi}(i) \hat{\xi}(j) \rangle - \langle \hat{\xi}(i) \rangle \langle \hat{\xi}(j) \rangle \right) \\ &= \sum_i \alpha_i^2 {\sigma_i}^2, \end{aligned} \tag{4.26}$$

wobei in der letzten Zeile die Unabhängigkeit der $\xi(i)$ verwendet wurde. Die Lagrange–Gleichungen des Problems lauten daher:

$$\begin{aligned} \frac{\partial}{\partial \alpha_k} \left(\sum_i \alpha_i^2 {\sigma_i}^2 + \lambda \sum_i \alpha_i \right) &= \\ 2 \alpha_k {\sigma_k}^2 + \lambda &= 0, \end{aligned} \tag{4.27}$$

wobei der Lagrange–Multiplikator λ eingeführt wurde, um die Nebenbedingung zu erfüllen. Diese ergibt:

$$\begin{aligned} \sum_i \alpha_i &= -\frac{\lambda}{2} \sum_i \frac{1}{{\sigma_i}^2} \stackrel{!}{=} 1 \\ \lambda &= -\frac{2}{\sum_i 1/{\sigma_i}^2}. \end{aligned} \tag{4.28}$$

Daher erhält man als Lösung des Variationsproblems:

$$\alpha_k = \frac{1/{\sigma_k}^2}{\sum_i 1/{\sigma_i}^2}. \tag{4.29}$$

Nimmt man nun die Schätzer $\hat{\xi}$ als erwartungstreu an, d.h. $\langle \hat{\xi}(i) \rangle = \xi,\ i = i_{\min}, \ldots, i_{\max}$, sei es, weil eine Jackknife–Bias–Reduktion vorgenommen wurde, oder weil der Bias zu vernachlässigen ist und ersetzt die Varianzen ${\sigma_i}^2$ der $\xi(i)$ durch die Jackknife–Varianz–Schätzungen $\widehat{\mathrm{VAR}}$ für die $\hat{\xi}(i)$, so wird die Wahl

$$\hat{\alpha}_k = \frac{1/\widehat{\mathrm{VAR}}(\hat{\xi}(k))}{\sum_i 1/\widehat{\mathrm{VAR}}(\hat{\xi}(i))} \tag{4.30}$$

in (4.25) im Mittel die Abweichung des so gewonnenen Mittelwertes $\bar{\xi}$ vom Erwartungswert ξ minimieren.

Nun sind die Zufallsvariablen $\xi(i)$ jedoch keineswegs statistisch unabhängig; aufgrund ihrer Definition durch die „Differenzenmethode“ sind die Variablen $\xi(i)$ und $\xi(j)$ vielmehr hochgradig korreliert, wenn $|i-j| \leq 2$, da hier teilweise dieselben Werte der Korrelationsfunktion $G(i)$ eingehen. Aber auch über diese formal aus der Definition resultierenden Korrelationen (und damit über den Abstand $|i-j| = 2$) hinaus sind die $\xi(i)$ nicht unabhängig voneinander: schon die einzelnen Werte der Korrelationsfunktion $G(i)$ sind keine unabhängigen Zufallsvariablen, sondern selbst wiederum korrelierte Größen; statistische Schwankungen bewirken nämlich Fluktuationen der Kurve G insgesamt, m.a.W.: es existieren Moden dieser Fluktuation, deren Wellenlängen wesentlich größer sind als der Abstand 1 zwischen benachbarten Punkten der Kurve. Anschaulich ist das aus der Cluster–Struktur des zugrundeliegenden Modells verständlich: liegt etwa im Sinne einer Fluktuation eine Konfiguration mit einem außergewöhnlich großen Gebiet einheitlicher Spin–Orientierung (geometrischer Cluster) vor, so erscheinen hier die Spins für *alle* Abstände, die kleiner sind als die Cluster–Ausdehnung, hochgradig korreliert; der Wert der Korrelationsfunktion $G(i)$ wird für alle diese Abstände größer sein als im Mittel. Gl. (4.26) muß daher entsprechend modifiziert werden:

$$\begin{aligned} \sigma^2(\bar{\xi}) &= \sum_{i,j} \alpha_i \alpha_j \left(\langle \hat{\xi}(i) \hat{\xi}(j) \rangle - \langle \hat{\xi}(i) \rangle \langle \hat{\xi}(j) \rangle \right) \\ &= \sum_{i,j} \alpha_i \alpha_j \Gamma_{ij}, \end{aligned} \tag{4.31}$$

wo Γ die Kovarianzmatrix der $\xi(i)$ bezeichnet. Eine analoge Variationsrechnung ergibt jetzt für die optimale Wahl der α_i:

$$\alpha_k = \frac{\sum_i (\Gamma^{-1})_{ik}}{\sum_{i,j} (\Gamma^{-1})_{ij}}. \tag{4.32}$$

Man muß also die gesamte Kovarianzmatrix der $\xi(i)$ kennen, um sie bei der Mittelwertbildung optimal gewichten zu können. Für die praktische Rechnung läßt sich dabei eine naheliegende Verallgemeinerung des Jackknife–Schätzers $\widehat{\mathrm{VAR}}$ auf nichtdiagonale Elemente der Korrelationsmatrix verwenden:

$$\widehat{\mathrm{CORR}}_{ij} \equiv \widehat{\mathrm{CORR}}(\hat{\xi}(i), \hat{\xi}(j)) = \frac{n-1}{n} \sum_{s=1}^{n} \left(\hat{\xi}_{(s)}(i) - \hat{\xi}_{(\cdot)}(i)\right)\left(\hat{\xi}_{(s)}(j) - \hat{\xi}_{(\cdot)}(j)\right). \tag{4.33}$$

Eine im Mittel optimale Wahl der Gewichte α_i unter Berücksichtigung der Korrelationen der $\hat{\xi}(i)$ ist damit gegeben durch:

$$\alpha_k = \frac{\sum_i (\widehat{\mathrm{CORR}}^{-1})_{ik}}{\sum_{i,j} (\widehat{\mathrm{CORR}}^{-1})_{ij}}. \tag{4.34}$$

Die Varianz des solchermaßen gewichteten Mittelwertes selbst ist dann, wie man aus Gl. (4.31) abliest, gerade durch den Normierungsfaktor der Gewichte gegeben:

$$\sigma^2(\bar{\xi}) = \frac{1}{\sum_{i,j} (\Gamma^{-1})_{ij}} \tag{4.35}$$

und entsprechend ein Schätzer dafür durch:

$$\hat{\sigma}^2(\bar{\xi}) = \frac{1}{\sum_{i,j} (\widehat{\mathrm{CORR}}^{-1})_{ij}}. \tag{4.36}$$

Alternativ dazu könnte man die Varianz des gewichteten Mittelwertes auch durch eine weitere Jackknife–Analyse abschätzen, d.h. indem man Jackknife–Block–Schätzer $\bar{\xi}_{(s)}$ und $\bar{\xi}_{(\cdot)}$ in der üblichen Weise definiert und in die Definition des Jackknife–Varianz–Schätzers $\widehat{\mathrm{VAR}}$ einsetzt; ein Vergleich der beiden so gewonnenen Schätzungen für die Varianz des Mittelwertes kann als Test auf Konsistenz des Verfahrens verstanden werden. Für die Analyse aller in dieser Arbeit untersuchten Systeme wurden beide Verfahren verwendet: in vielen Fällen stimmen die Ergebnisse nahezu auf Maschinengenauigkeit überein, immer jedoch auf mindestens 8-10 Stellen genau.

Aus den Ausführungen dieses Abschnittes wird klar, warum auf dem Weg hierhin ein so großer Wert auf die Fehleranalyse gelegt wurde: während in vielen Fällen die berechneten Fehler (etwa) einer MC–Simulation lediglich als Genauigkeitsangabe in den Endergebnissen auftauchen, gehen hier die geschätzten Varianzen und Korrelationen durch die gewichtete Mittelung direkt in den Wert des Endergebnisses für die Korrelationslängen ein, so daß in unserem Fall die Varianzanalyse kein schmückendes Beiwerk darstellt, sondern an der Gewinnung der endgültigen Ergebnisse essentiellen Anteil hat.

4.4.4 Fehler–Fortpflanzung vs. Jackknife–Analyse

Die klassische Methode, Fehler von nichtlinearen Funktionen von Observablen zu bestimmen, ist die Verwendung von *Fehler–Fortpflanzungs*–Gesetzen; dabei werden unter Voraussetzung hinreichend kleiner Fehler die Varianzen solcher Funktionen von Zufallsvariablen durch eine Taylor–Entwicklung der Funktionen abgeschätzt, die nach der ersten Ordnung abgebrochen wird. Sind $z_1, \ldots, z_m$ unabhängige Zufallsvariable mit Varianzen ${\sigma_1}^2, \ldots, {\sigma_m}^2$ und sind $f_1(z_1, \ldots, z_m), \ldots, f_n(z_1, \ldots, z_m)$ differenzierbare Funktionen der Zufallsvariablen, so lassen sich die Varianzen der f_i abschätzen durch:

$$\sigma^2(f_k) \approx \sum_{i=1}^{m} \left(\frac{\partial f_k}{\partial z_i} \right)^2 {\sigma_i}^2. \tag{4.37}$$

Gibt man die Voraussetzung der Unabhängigkeit der z_i auf, so muß man auch hier die gesamte Kovarianzmatrix Γ der z_i berücksichtigen [11]:

$$\Gamma^f{}_{kl} \approx \sum_{i,j=1}^{m} T_{ki} \Gamma^z{}_{ij} T_{lj}, \tag{4.38}$$

wobei $\Gamma^z \in \mathbb{R}_{m \times m}$ die Kovarianzmatrix der ursprünglichen Variablen z_i und $\Gamma^f \in \mathbb{R}_{n \times n}$ die Matrix der transformierten Variablen f_i bezeichnet; $T \in \mathbb{R}_{n \times m}$ ist die Jacobi–Matrix der Transformation, d.h.:

$$T_{ij} = \left(\frac{\partial f_i(z_1, \ldots, z_m)}{\partial z_j} \right). \tag{4.39}$$

Eine Fehlerrechnung für die Korrelationslängen $\hat{\xi}(i)$ mit Hilfe „klassischer“ Methoden sieht daher folgendermaßen aus:

1. Die Fehler–Fortpflanzung muß nur für die nichtlinearen Schritte der Transformation von den Werten der Korrelationsfunktion zu den Korrelationslängen bei der Verwendung der Differenzenmethode gemäß (4.14) durchgeführt werden. Daher kann der erste Schritt, die Differenzenbildung, direkt ausgeführt werden. Die Variablen, von denen hier auszugehen ist, sind damit gegeben durch die Differenzen

$$d(i) = G(i) - G(i-1), \quad i = 1, \ldots, L_z/2, \tag{4.40}$$

so daß die Korrelationslängen aus der Transformation

$$\hat{\xi}(i) = \left[\ln \left(\frac{d(i)}{d(i+1)} \right) \right]^{-1}, \quad i = 1, \ldots, L_z/2 - 1 \tag{4.41}$$

hervorgehen.

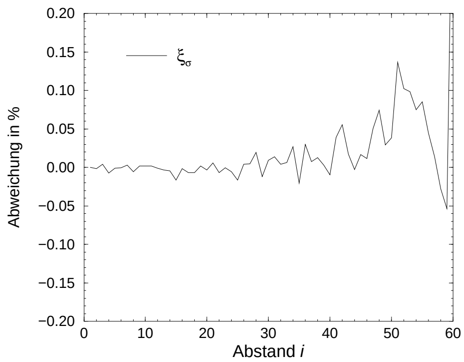
2. Die Varianzen der Differenz–Variablen $d(i)$ als Linearkombinationen der Observablen lassen sich gemäß dem naiven Ansatz der Fehlerrechnung abschätzen:
$$\hat{\sigma}^2(d(i)) = \frac{1}{n(n-1)} \sum_{t=1}^{n} (d_t(i) - \bar{d}(i))^2, \tag{4.42}$$
wobei $d_t(i)$ die, durch eine vorgängige Binning–Analyse weitgehend unabhängigen, Elemente der gemessenen Zeitreihe der Länge n bezeichnet und $\bar{d}(i)$ den zugehörigen Zeitmittelwert.

3. Analog lassen sich die Korrelationen der $d(i)$ durch eine entsprechende Verallgemeinerung dieses Ausdrucks abschätzen:
$$\hat{\Gamma}(d(i), d(j)) = \frac{1}{n(n-1)} \sum_{t=1}^{n} (d_t(i) - \bar{d}(i))(d_t(j) - \bar{d}(j)). \tag{4.43}$$

4. Durch Anwendung des Gesetzes der Fehler–Fortpflanzung (4.38) läßt sich nun aus dieser Information ein Schätzer für die Kovarianzmatrix der abgeleiteten Größen $\hat{\xi}(i)$ gewinnen:
$$\hat{\Gamma}^{\xi}_{kl} \approx \sum_{i,j=1}^{L_z/2} \left(\frac{\partial \hat{\xi}(k)}{\partial d(i)} \right) \hat{\Gamma}^{d}_{ij} \left(\frac{\partial \hat{\xi}(l)}{\partial d(j)} \right), \qquad 1 \le k, l \le L_z/2 - 1. \tag{4.44}$$

Zur Bestimmung von $\hat{\Gamma}^{\xi}$ muß man noch die Jacobi–Matrix der Transformation (4.41) kennen:
$$\begin{aligned} \frac{\partial \hat{\xi}(i)}{\partial d(i)} &= -\frac{1/d(i)}{\left[\ln\left(\frac{d(i)}{d(i+1)}\right)\right]^2} \equiv a_i, \\ \frac{\partial \hat{\xi}(i)}{\partial d(i+1)} &= +\frac{1/d(i+1)}{\left[\ln\left(\frac{d(i)}{d(i+1)}\right)\right]^2} \equiv b_i, \\ \frac{\partial \hat{\xi}(i)}{\partial d(j)} &= 0, \qquad j \notin \{i, i+1\}. \end{aligned} \tag{4.45}$$
Ausführen des Matrizenproduktes in (4.44) ergibt dann:
$$\hat{\Gamma}^{\xi}_{ij} \approx a_i \hat{\Gamma}^{d}_{ij} a_j + a_i \hat{\Gamma}^{d}_{i\,j+1} b_j + b_i \hat{\Gamma}^{d}_{i+1\,j} a_j + b_i \hat{\Gamma}^{d}_{i+1\,j+1} b_j. \tag{4.46}$$

Für die Diagonalelemente, also die Varianzen der $\hat{\xi}(i)$, schließlich hat man daher:
$$\hat{\sigma}^2(\hat{\xi}(i)) \equiv \hat{\Gamma}^{\xi}_{ii} = {a_i}^2 \hat{\sigma}^2(d(i)) + {b_i}^2 \hat{\sigma}^2(d(i+1)) + 2 a_i b_i \hat{\Gamma}^{d}_{i\,i+1}. \tag{4.47}$$



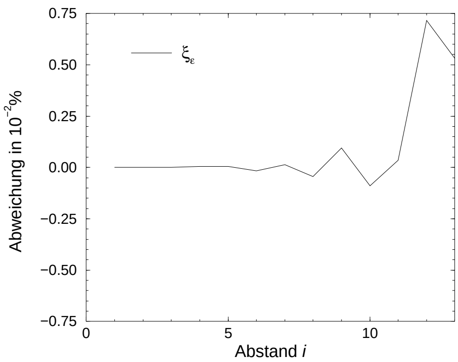


Abb. 4.11: Abweichungen zwischen den geschätzten Varianzen der Korrelationslängen $\hat{\xi}_\sigma(i)$ (links) und $\hat{\xi}_\epsilon(i)$ (rechts) aus der Jackknife– und der Fehler–Fortpflanzungs–Analyse, bezogen auf die Schätzung der Fehler–Fortpflanzungs–Rechnung. Es handelt sich um ein System mit $L_x = 10, L_y = 192$ und periodischen Randbedingungen. Für $\hat{\xi}_\epsilon$ sind wesentlich weniger Punkte aufgetragen, da hier aufgrund der deutlich kürzeren Reichweite der Korrelationen die statistischen Schwankungen schon für kleinere Abstände i (als im Falle der Spin–Korrelationslänge) den exponentiellen Abfall überdecken und die Anwendung der Differenzenmethode unmöglich machen (das Argument des Logarithmus wird negativ).

Zur Validierung der verwendeten, in Kap. 4.4.1 beschriebenen Jackknife–Methode wurden für einige Systeme beide Methoden der Fehler–Analyse parallel angewandt, um einen Vergleich zu ermöglichen. Die Differenzen der Ergebnisse beider Methoden sind generell erstaunlich gering: obwohl die Ansätze vollkommen verschieden sind und die Fehler–Fortpflanzungs–Rechnung nur die erste (nicht verschwindende) Ordnung der Taylor–Entwicklung berücksichtigt, liegen die relativen Abweichungen der mit beiden Methoden abgeschätzten Fehler generell unter etwa 0.5%; im Bereich kleiner Schwankungen, d.h. zu Beginn des Plateaus in den Werten für $\hat{\xi}$ sind die Unterschiede sogar regelmäßig noch um eine Größenordnung geringer. Abb. 4.11 zeigt dies für ein Beispielsystem mit periodischen Randbedingungen. Mit diesen Ergebnissen ist der Jackknife–Ansatz allgemein in dieser Anwendung hervorragend gerechtfertigt. Man kann hoffen, daß die verbleibenden Abweichungen — insbesondere bei großen Abständen i, also großen Schwankungen der $G(i)$ und damit $\hat{\xi}(i)$ — die Berücksichtigung der höheren Ordnungen in der Taylor–Reihe der Momenten–Entwicklung (4.38) durch das Jackknife–Verfahren gegenüber deren systematischer Vernachlässigung durch die Fehler–Fortpflanzungs–Methode anzeigen.

4.4.5 Abschätzung des Plateau–Bereichs

Die Differenzenmethode bietet gegenüber der einfachen Verwendung eines logarithmischen Graphen der Korrelationsfunktionen den Vorteil, daß der Bereich konstanter Korrelationslängen — der lineare Abschnitt des logarithmischen Graphen — allein mit „Augenmaß“ verhältnismäßig leicht abgeschätzt werden kann. Da solches Augenmaß niemals frei von Subjektivität sein kann, erscheint es aus methodologischer Sicht wünschenswert, diese Verantwortung von sich auf die Maschine abzuwälzen und den Prozeß der Bestimmung des Plateau–Bereichs, dessen Werte $\hat{\xi}(i)$ in die Bestimmung des Mittelwertes $\bar{\xi}$ gemäß Kap. 4.4.3 eingehen sollen, zu automatisieren. Zu diesem Zweck benötigt man ein quantifiziertes Maß für die Konstanz der Schätzungen $\hat{\xi}(i)$ auf dem Plateau gegenüber ihrem Abfall zu den Extremen sehr kurzer und sehr langer Abstände i hin; dies ist nichts anderes als ein Spezialfall eines allgemeinen Maßes für die Güte der Anpassung (*Fit*) eines Satzes von Messungen von Zufallsvariablen y_i an eine vorgegebene Funktion $y = f(x)$, d.h. eines Gütemaßes der Regressions–Rechnung: die Funktion f ist hier eben eine Konstante, nämlich $\bar{\xi}$. Die klassische Kennzahl zur Bewertung der Güte eines solchen Fits ist durch die χ^2–Verteilung gegeben [36]:

$$\chi^2 = \sum_{i,j} (y_i - f(x_i))(\Gamma^{-1})_{ij}(y_j - f(x_j)), \tag{4.48}$$

wobei Γ wiederum die Korrelationsmatrix der y_i symbolisiert. In Anwendung auf das Problem der Abschätzung des Plateau–Bereichs ist eine Schätzfunktion für diese Größe gegeben durch:

$$\chi^2 = \sum_{i,j=i_{\min}}^{i_{\max}} (\hat{\xi}(i) - \bar{\xi})(\hat{\Gamma}^{-1})_{ij}(\hat{\xi}(j) - \bar{\xi}). \tag{4.49}$$

Der Fit wird dann als um so besser beurteilt, je weniger die Größe χ^2/g von Eins abweicht; dabei ist g durch die Anzahl der Freiheitsgrade des Problems gegeben und entspricht in unserem Fall $i_{\max} - i_{\min}$ (eigentlich würde man $i_{\max} - i_{\min} + 1$ erwarten, aber die Vorgabe des Mittelwertes $\bar{\xi}$ eliminiert einen Freiheitsgrad). Anschaulich bedeutet die Aussage „$\chi^2/g = 1$“ gerade, daß die tatsächlichen Schwankungen der Schätzungen $\hat{\xi}(i)$ um den Mittelwert $\bar{\xi}$ (und die entsprechenden Korrelationen) gerade dem gemäß ihrer berechneten Varianzen (bzw. Kovarianzen für die Nichtdiagonal–Terme) zu erwartenden Maß entsprechen.

Das Problem der Bestimmung des optimalen Bereichs $i_{\min}, \ldots, i_{\max}$ läßt sich daher als Variationsproblem auffassen: bestimme jenen Bereich, für den $|\chi^2/g - 1|$ gemäß

(4.49) minimal wird. Prinzipiell ist es auch möglich, in die Menge der in Frage kommenden Abstände neben geschlossenen Bereichen Kombinationen isolierter Punkte mit aufzunehmen; das scheint aber wenig sinnvoll: ganz abgesehen vom explodierenden Rechenaufwand bei der Berücksichtigung *aller* möglichen Kombinationen von Punkten, besteht die Gefahr, daß aufgrund der Asymmetrie der Verteilung der Schätzungen $\hat{\xi}(i)$ um den theoretischen Wert ξ (die sich auch in der Struktur der Kovarianzmatrix widerspiegelt) systematische Fehler entstehen. Weiterhin ist unklar, ob als Vergleichspunkt der Optimierung der Wert $\chi^2/g = 1$ dienen *muß*, oder ob nicht etwa $\chi^2/g = 0$ einen maßgeblichen Richtwert darstellt; dies umso mehr, als (wie in Kap. 3.2.2 erwähnt) das Jackknife–Verfahren bewiesenermaßen Varianzen und Kovarianzen eher überschätzt als unterschätzt. In der praktischen Anwendung zeigt sich jedoch, daß die Vorschrift $\chi^2/g \to 0$ dazu neigt, einige wenige Punkte auszuwählen, die zufällig sehr wenig voneinander abweichen, anstatt den gesamten sichtbaren Plateau–Bereich als Gebiet der Mittelung vorzuschlagen (was auch hinsichtlich einer Minimierung der Varianz von $\bar{\xi}$ wünschenswert ist). Generell muß natürlich eine gewisse Mindestlänge des Bereichs $i_{\min}, \ldots, i_{\max}$ vorgegeben werden, um eine vernünftige Schätzung des Plateaus zu erhalten; diese sollte jedoch deutlich kleiner sein als die tatsächliche sichtbare Ausdehnung des Plateaus, um das Verfahren nicht sinnlos zu machen.

Das beschriebene Verfahren funktioniert für die Spin–Korrelationslängen insgesamt sehr gut. Nur in einigen wenigen Fällen sind hier Korrekturen von Hand vonnöten, da der vorgeschlagene Bereich gegenüber der sichtbaren Plateau–Ausdehnung extrem klein ist; in diesen Fällen könnte man zu einer Alternative übergehen, die ebenfalls einen akzeptablen Wert für χ^2/g aufweist, etwa indem nicht nur der (im Sinne von $|\chi^2/g - 1| \to \min$) optimale Bereich bestimmt wird, sondern auch noch einige der nächstgelegenen suboptimalen Vorschläge in Erwägung gezogen werden. Für die Korrelationslängen der Energiedichte gestaltet sich die Lage generell etwas schwieriger: da für alle Modelle und Randbedingungen ξ_ϵ wesentlich kleiner ist als ξ_σ, stehen hier weniger Punkte $\hat{\xi}(i)$ zur Mittelung zur Verfügung, so daß hier einer Optimierungsrechnung unter Berücksichtigung einer Mindestlänge deutlich weniger Spielraum zur Verfügung steht. Abb. 4.12 zeigt einige Beispiele für das Ergebnis des Optimierungsverfahrens.

Insbesondere angesichts der Unklarheit über eine eventuelle systematische Überschätzung der Fehler durch das Jackknife–Verfahren ist die Minimierung der Abweichung $|\chi^2/g - 1|$ aus Sicht der mathematischen Statistik von zweifelhaftem

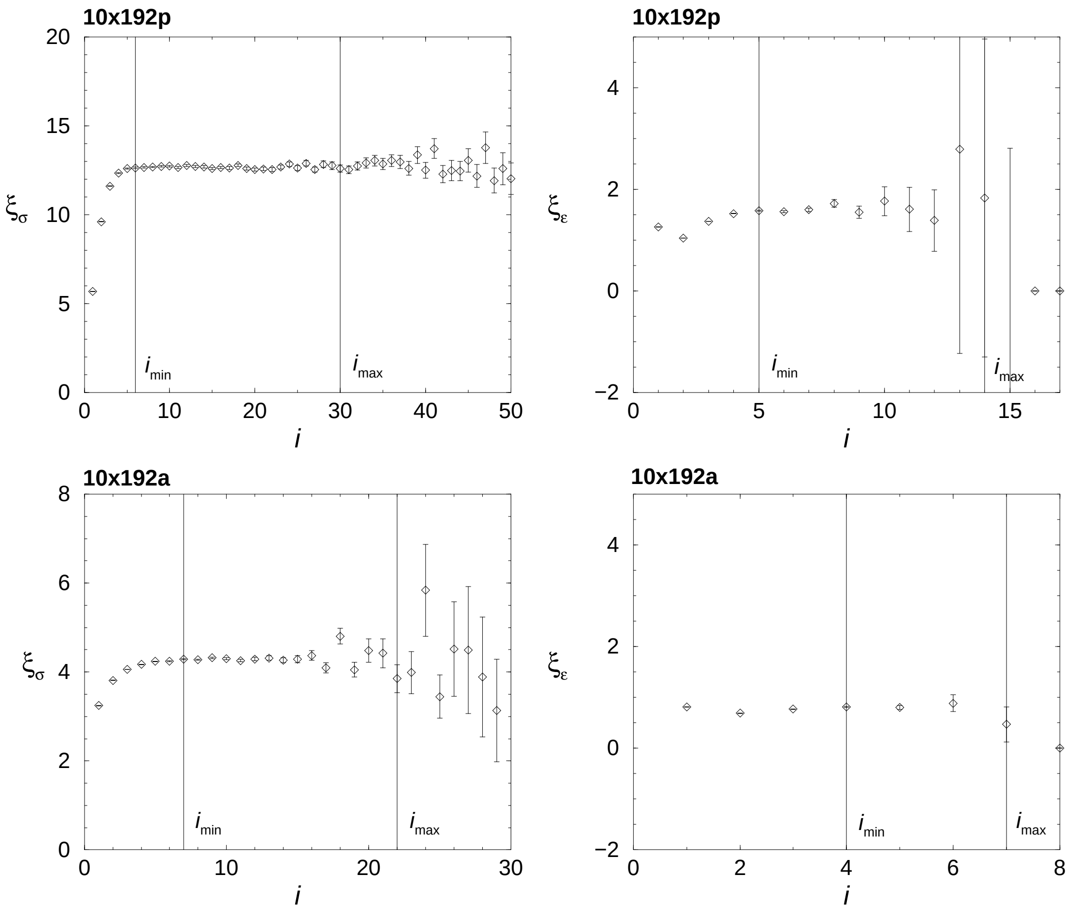


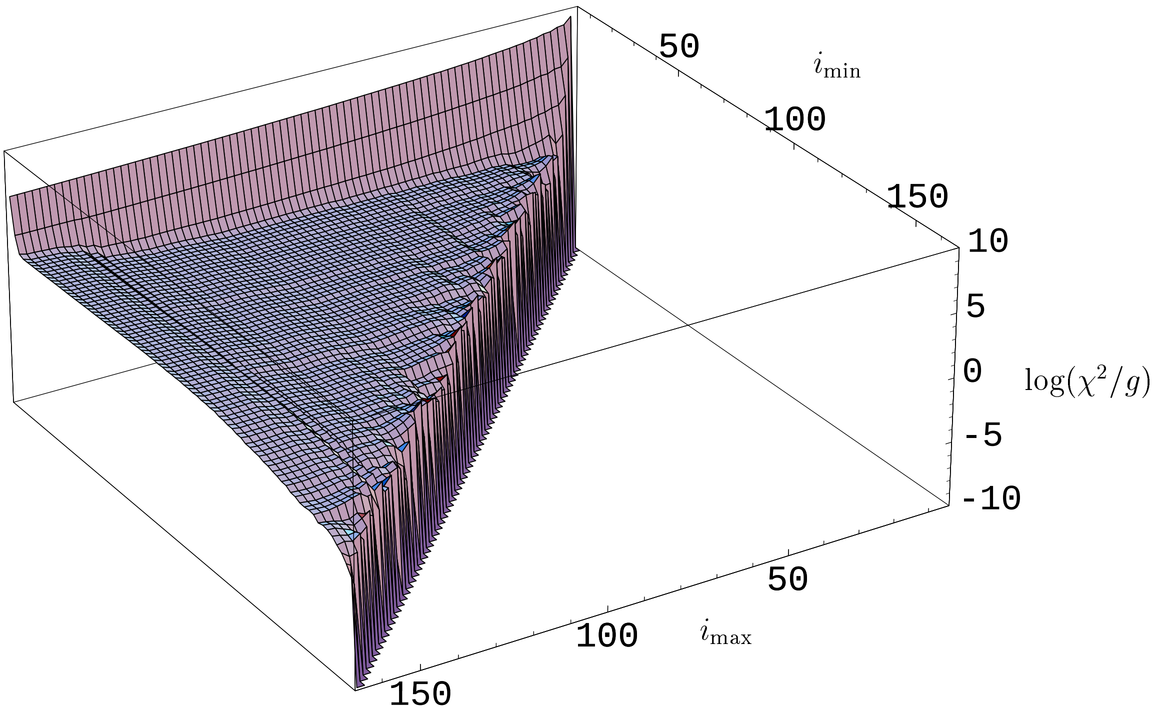
Abb. 4.12: Ergebnisse des Ansatzes $|\chi^2/g - 1| \to \min$ zur Bestimmung des Plateau–Bereichs der Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen von Spin und Energiedichte. Die Grenzen $i_{\min}$ und $i_{\max}$ sind die Resultate des Optimierungsprozesses; für ξ_σ galt die Restriktion $i_{\max} - i_{\min} \geq 10$, für ξ_ϵ lediglich $i_{\max} - i_{\min} \geq 3$. Die Fit–Güte ergab sich für das System mit periodischen Randbedingungen (oben) zu $\chi^2/g = 0.99994$ (ξ_σ, $i = 6, \ldots, 30$) und $\chi^2/g = 0.957$ (ξ_ϵ, $i = 7, \ldots, 14$); für die antiperiodischen Randbedingungen (unten) ergab sich: $\chi^2/g = 1.008$ (ξ_σ, $i = 7, \ldots, 22$) und $\chi^2/g = 0.361$ (ξ_ϵ, $i = 4, \ldots, 7$).

Wert: erhält man etwa (infolge des Jackknifings) für den sichtbaren Plateau–Bereich eine Wert von χ^2/g z.B. von 0.5, so wird ein solches Optimierungsverfahren Punkte aus dem systematischen Abfall der Schätzung $\hat{\xi}(i)$ zu kleinen i hin zur Mittelung hinzunehmen, um das geforderte $\chi^2/g = 1$ zu erreichen; das ergibt jedoch eine systematisch zu kleine Schätzung für $\bar{\xi}$. Die — aus mathematischer Sicht vielleicht eher akzeptable — Vorschrift $|\chi^2/g| \to \min$ erzeugt jedoch wie gesehen ebenfalls keine zufriedenstellenden Schätzungen für den Bereich $i_{\min}, \ldots, i_{\max}$. Grundsätzlich haftet einem solchen Optimierungsverfahren das Manko an, daß es bei einer Anwendung nicht auf eine einfache Funktion, sondern auf einen Satz von Schätzern für Zufallsvariablen, zunächst nicht zwischen „echten“ Minima, die auch im Limes unendlich langer Messungen auftreten würden, und einfachen „Ausreißern“ der fluktuierenden Größe unterscheiden kann.

Um genauer zu verstehen, was bei Anwendung des Optimierungsverfahrens geschieht und vielleicht eine befriedigendere Lösung des Problems zu finden, kann man die Verteilung der Größe χ^2/g in Abhängigkeit von der Wahl des Plateau–Bereiches $i_{\min}, \ldots, i_{\max}$ betrachten. In Abhängigkeit von *beiden* Parametern $i_{\min}$ und $i_{\max}$ läßt sich χ^2/g lediglich in einem dreidimensionalen Graphen darstellen. Abb. 4.13 zeigt Beispiele für diese Abhängigkeiten für die Korrelationslängen von Spin und Energiedichte eines Systems. Wegen der Bedingung $i_{\min} \leq i_{\max}$ liegen die relevanten Werte über der Grundfläche eines gleichseitigen Dreiecks. Ausgehend von einem klar ausgebildeten Plateau–Bereich erkennt man deutlich den extrem steilen Anstieg von χ^2/g für (zu) kleine Werte von $i_{\min}$: hier machen sich die systematischen Abweichungen vom rein exponentiellen Verhalten der Korrelationsfunktion für kleine Abstände i bemerkbar, die überdies durch die besonders kleinen Fehler der Schätzungen $\hat{G}(i)$ in diesem Bereich einen überproportionalen Einfluß auf den Wert von χ^2/g nehmen. Am anderen Ende des Bereiches zeigt sich für große $i_{\max}$ wiederum ein (allerdings flacherer) Anstieg, der vom Einfluß der endlichen Länge L_y der Streifen gemäß Kap. 4.2 herrührt; in der Nähe der Hypotenuse des Dreiecks schließlich zeigt sich ein „Ausfransen“ des Plateaus durch den Abfall der Breite des Bereiches $i_{\min}, \ldots, i_{\max}$. Während für ξ_σ das Plateau etwa auf der Höhe $\chi^2/g = 1$ liegt, was sinnvolle Ergebnisse für das Optimierungsverfahren verspricht, liegt es für ξ_ϵ deutlich niedriger (etwa bei 0.5), was zu den oben erläuterten Problemen bei der Optimierung führt.

Weitere Informationen liefern Schnitte durch dieses Gebirge wie sie in Abb. 4.14 für ξ_σ gezeigt sind: hier zeichnet sich die Möglichkeit eines verbesserten Verfahrens zur Bestimmung des optimalen Bereiches anhand der hinzugewonnnen Information ab. Setzt man sich als Ziel die Maximierung der Bereichsausdehnung $i_{\max} - i_{\min} + 1$, was angesichts des übergeordneten Ziels, mit den gegebenen Daten eine möglichst hohe Genauigkeit der Endergebnisse zu erreichen, sinnvoll erscheint und beobachtet dabei das Verhalten von χ^2/g im Sinne einer Nebenbedingung, etwa $\chi^2/g < 1.5$, so kann man hierbei folgendermaßen vorgehen:

1. Zunächst wird die Grenze $i_{\max}$ bestimmt, da sie im Sinne des o.g. schwächeren



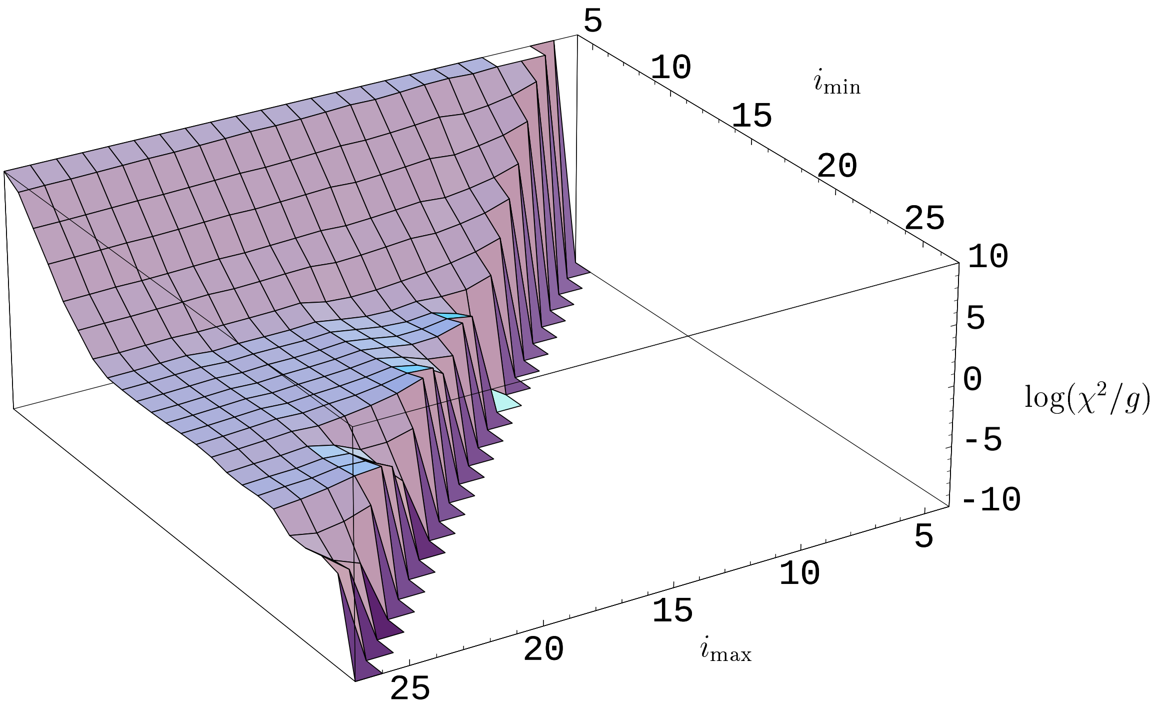


Abb. 4.13: χ^2/g für die Bildung der Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin (oben) und Energiedichte (unten) in Abhängigkeit von den Plateau–Grenzen i_{min} und i_{max} für ein 18×344–System mit periodischen Randbedingungen. Im Falle von $\bar{\xi}_\sigma$ sind aus reproduktionstechnischen Gründen nur die Punkte mit geraden Werten von i_{min} und i_{max} eingezeichnet. Im Falle von $\bar{\xi}_\epsilon$ ist nur der Parameterbereich der i berücksichtigt, für den $\hat{\xi}$ nach der Differenzenmethode wohldefiniert ist.

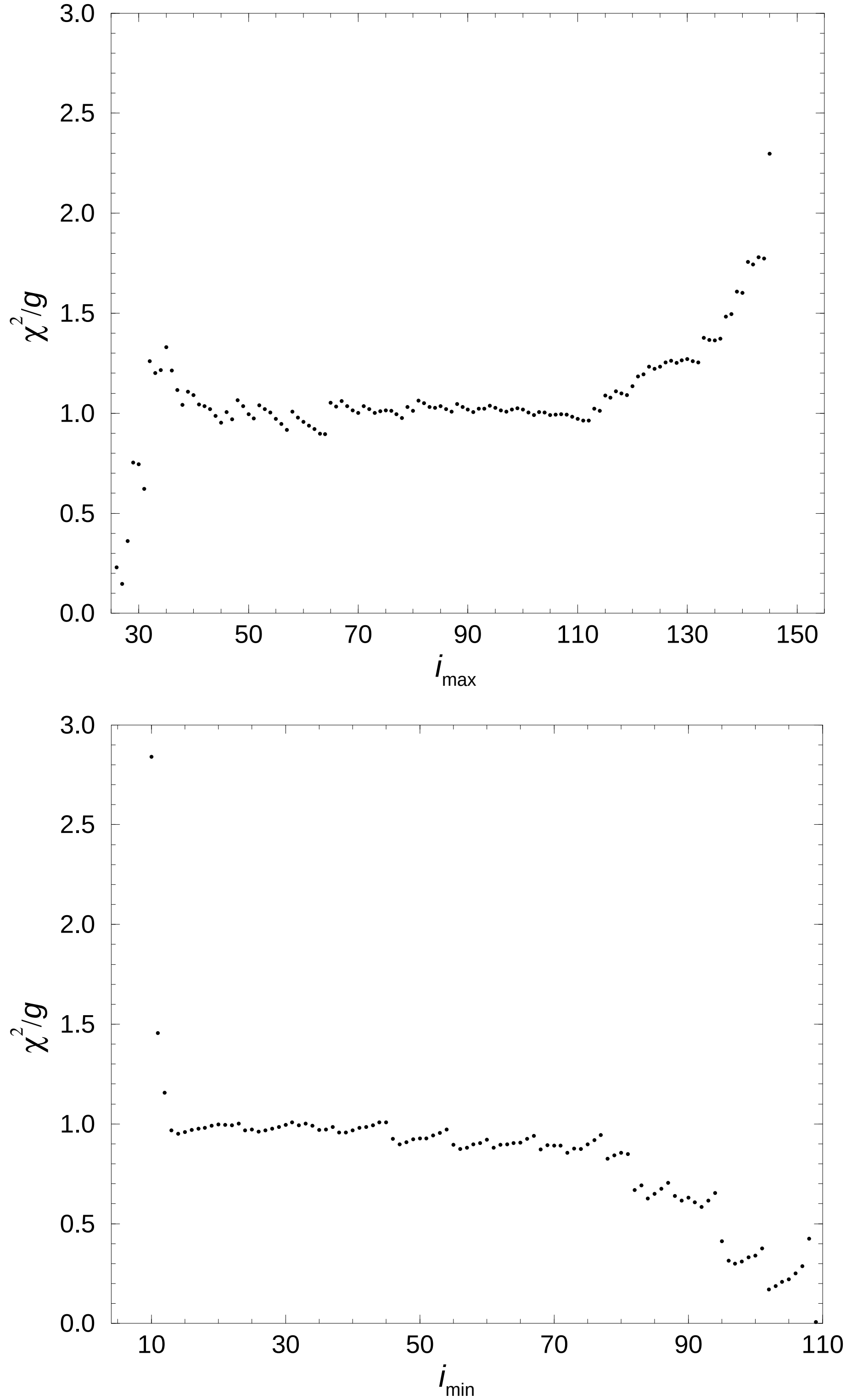


Abb. 4.14: Schnitte durch den Graphen von $\chi^2/g(i_{\text{min}}, i_{\text{max}})$ gemäß Abb. 4.13 für $\bar{\xi}_\sigma$. Zunächst wurde als deutlich auf dem Plateau gelegener Wert $i_{\text{min}} = 25$ gewählt. Im Schnitt $\{\chi^2/g \,|\, i_{\text{min}} = 25\}$ (oben) erkennt man den Anstieg von χ^2/g zu sehr großen Abständen i_{max} hin, der von der endlichen Länge L_y der Streifen herrührt. Hieraus wurde $i_{\text{max}} = 110$ bestimmt. Der Schnitt $\{\chi^2/g \,|\, i_{\text{max}} = 110\}$ (unten) zeigt den explosionsartigen Anstieg von χ^2/g für etwa $i_{\text{min}} < 15$. Mit einem gewissen „Sicherheitsabstand" wurde hier $i_{\text{min}} = 20$ gewählt. Die langreichweitigen Korrelationen haben ihre Ursache in der Wahl $\Delta = 4$ in (4.15).

Anstieges (d.h. der größeren Fehler der dortigen $\hat{\xi}(i)$) die unproblematischere Beschränkung darstellt. Dazu wählt man aus der dreidimensionalen Darstellung $\chi^2/g(i_{\rm min}, i_{\rm max})$ ein $i^+_{\rm min}$ aus, das deutlich innerhalb des Plateaus liegt (aber noch nicht optimal ist). Im Schnitt $\{\chi^2/g \,|\, i_{\rm min} = i^+_{\rm min}\}$ bestimmt man nun die rechte Grenze $i^*_{\rm max}$ des dortigen Plateaus von χ^2/g.

2. In einem zweiten Schritt läßt sich dann anhand des Schnittes $\{\chi^2/g \,|\, i_{\rm max} = i^*_{\rm max}\}$ die wichtigere Grenze $i^*_{\rm min}$ sehr leicht bestimmen, da der Plateau–Bereich sehr scharf vom Anstieg von χ^2/g für zu kleine $i_{\rm min}$ abgegrenzt ist.

Während die Methode der Minimierung von $|\chi^2/g - 1|$ selbst im Falle der Spin–Korrelationslänge aufgrund statistischer Schwankungen dazu neigt, für verschiedene Systeme stark schwankende Bereichslängen $i_{\rm max} - i_{\rm min} + 1$ vorzuschlagen, was zu einer ungerechtfertigten Schwankung der Gewichte dieser Ergebnisse führt, lassen sich mit dem hier angegebenen Verfahren weitgehend konstante, bzw. mit wachsender Länge L_y der Streifen ansteigende, Bereiche der Mittelung auswählen. Wir haben daher die endgültige Analyse aller Produktionsläufe mit Hilfe dieser Methode vorgenommen.

4.4.6 Spezielle Verfahren: *Double Jackknife Bias Corrected Estimators*

Oben in Kap. 4.4.2 wurde die Anwendung der Jackknife–Methode zur Bias–Reduktion der Schätzer $\hat{\xi}(i)$ für die Korrelationslängen beschrieben. Wendet man diese an, so wird jedoch die Fehler–Rechnung gemäß Kap. 4.4.1 ungültig: die Bias–Schätzung addiert zum ursprünglichen Schätzer einen Term, der selbst wiederum eine fluktuierende Zufallsvariable darstellt und somit zur Varianz des korrigierten Schätzers beiträgt. Für solche korrigierten Schätzer ist eine konsistente Fehlerrechnung durch das in Kap. 3.2.2 beschriebene Verfahren des doppelten Jackknifing (*Double Jackknife Bias Corrected Estimators*) möglich. Dazu bildet man Jackknife–Blöcke zweiter Ordnung und erhält einen Satz von biaskorrigierten Jackknife–Schätzern für die Korrelationslängen:

$$\begin{aligned} \tilde{\xi}_{(s)}(i) &= (n-1)\hat{\xi}_{(s)}(i) - (n-2)\hat{\xi}_{(s)(\cdot)}(i), \\ \tilde{\xi}_{(\cdot)}(i) &= \frac{1}{n}\sum_s \tilde{\xi}_{(s)}. \end{aligned} \tag{4.50}$$

Mit ihrer Hilfe läßt sich dann eine Jackknife–Varianz–Schätzung in der üblichen Weise angeben:

$$\widehat{\mathrm{VAR}}(\tilde{\xi}(i)) = \frac{n-1}{n} \sum_s \left[\tilde{\xi}_{(s)}(i) - \tilde{\xi}_{(\cdot)}(i)\right]^2. \tag{4.51}$$

Ebenso lassen sich analog die Korrelationen der korrigierten Schätzer $\tilde{\xi}(i)$ untereinander bestimmen. Mit diesem Werkzeug kann man nun die gesamte in diesem Kapitel beschriebene Fehleranalyse — die gewichtete Mittelung zur Bestimmung von $\bar{\xi}$ und die χ^2–Methode der Plateau–Abschätzung — auch unter Verwendung von biaskorrigierten Schätzern vornehmen. Oben hatten wir gesehen, daß der Effekt der Bias–Korrektur eher gering ist und damit die Notwendigkeit ihrer Anwendung fraglich. Eine erneute Untersuchung lohnt hier vor allem deshalb, weil die Mittelung der $\hat{\xi}(i)$ über das Plateau die rein statistischen Fehler (unter Vernachlässigung der Korrelationen) um einen Faktor $1/\sqrt{\Delta i}$, $\Delta i = i_{\max} - i_{\min} + 1$ reduziert, die durch einen Bias der Schätzer hervorgerufenen systematischen Abweichungen jedoch a priori unberührt läßt; daher kann sich das Verhältnis der Bias–Korrektur zu den statistischen Fehlern zugunsten der ersteren noch verschieben.

Wie Tab. 4.5 zeigt, fällt die Bias–Korrektur in der Summe der Mittelwerte $\bar{\xi}$ tatsächlich deutlich stärker ins Gewicht als bei ihrer Anwendung auf die Einzelwerte $\hat{\xi}(i)$ in Kap. 4.4.2. Besonders gilt dies für den Fall der Energie–Korrelationslänge: hier werden die Schwankungen der Schätzungen $\hat{\xi}(i)$ für wachsende Abstände schnell sehr groß und gleichzeitig liefert das Jackknife–Verfahren Bias–Korrekturen in der Größenordnung der Varianzen selbst. Der allgemeinen Erfahrung mit der Jackknife–Methode nach [31] ist es etwa ab einem Verhältnis $\widehat{\mathrm{BIAS}}/\sqrt{\widehat{\mathrm{VAR}}} \gtrsim 1/4$ sinnvoll, ein Bias–Reduktions–Verfahren anzuwenden. Auf die geschätzten Varianzen hat das *Double Jackknife* Verfahren dagegen erstaunlicherweise nur einen geringen Einfluß; im Sinne einer gewissen Konstanz der in einem Sample enthaltenen Informationsmenge würde man erwarten, daß der vermehrte Einsatz dieser Information zur möglichst erwartungstreuen, „genauen" Bestimmung der Meßgrößen in einem entsprechenden Ansteigen der Schwankungen der korrigierten Schätzer resultiert. In dieser Hinsicht ist die Anwendung der Jackknife–Bias–Korrektur hier offenbar noch durchaus unbedenklich. Eine Ausnahme bildet hier wiederum der Rand des Plateaus der Schätzungen für die Korrelationslänge der Energiedichte, wo die Bias–Korrektur der *einzelnen* Punkte die Varianzschätzung des *Double Jackknife* Verfahrens gegenüber der ursprünglichen Schätzung deutlich in die Höhe treibt. Andererseits muß man den erheblichen Rechenaufwand für eine solche Jackknife–Analyse zweiter Ordnung als Kosten mit in die Erwägung einbeziehen: ist die Fehler–Analyse insgesamt

Größe	unkorrigiert	korrigiert	Abweichung/$\bar{\xi}$	Abweichung/$\sqrt{\widehat{\mathrm{VAR}}}$
$\bar{\xi}_\sigma$	12.673083	12.672610	0.03%	6.5%
$\sqrt{\widehat{\mathrm{VAR}}(\bar{\xi}_\sigma)}$	0.00589421	0.00589406		0.003%
$\bar{\xi}_\epsilon$	1.581975	1.581245	0.05%	13%
$\sqrt{\widehat{\mathrm{VAR}}(\bar{\xi}_\epsilon)}$	0.00558219	0.00558628		4%

Tab. 4.5: Effekt einer Jackknife–Bias–Korrektur auf Mittelwert und Varianz der Korrelationslängen des 10×192–Beispielsystems mit periodischen Randbedingungen. Um einen sinnvollen Vergleich der Werte zu ermöglichen, wurden die mit der χ^2–Methode ermittelten Grenzen für die Mittelung der $\hat{\xi}(i)$ für das unkorrigierte Sample auch für die biaskorrigierten Schätzungen verwandt. Da — insbesondere im Falle der Spin–Korrelationslänge — die Abweichungen $|\chi^2/g - 1|$ sehr klein sind, kann die Verwendung einer Bias–Korrektur nämlich durchaus auch die Ergebnisse des Optimierungsverfahrens aus Kap. 4.4.5 berühren.

hier schon recht aufwendig, so multipliziert sich der Aufwand durch die Verwendung zweier Jackknife–Transformationen nochmals mit der Anzahl der Blöcke (hier bis zu 2000). Zudem verschwindet der Bias der betrachteten Größen mit dem Verhalten $1/n$ mit einer Verlängerung der Zeitreihen (Anzahl der Blöcke n) wesentlich schneller als die statistischen Schwankungen, die sich gemäß $1/\sqrt{n}$ reduzieren; abseits vom asymptotische Bereich können jedoch die Amplituden der beiden Terme so unterschiedlich sein, daß der Bias durchaus eine Rolle spielt. Wir werden angesichts dieses uneinheitlichen Bildes für die Simulationen der dreidimensionalen Systeme, für die vielleicht nur kürzere Laufzeiten möglich sein werden, im Einzelfalle nochmals abzuwägen haben, ob eine kombinierte Bias–/Varianz–Analyse mit der *Double Jackknife* Methode sinnvoll ist.

4.5 Der Einfluß der Randbedingungen

Bis auf wenige Ausnahmen war in den bisherigen Beispielen ausschließlich von Systemen mit *periodischen* Randbedingungen die Rede; zumindest für die dreidimensionalen Systeme sind es jedoch gerade die *antiperiodischen* Randbedingungen, für die man die interessanteren Ergebnisse erwartet (s.o. Kap. 2.6). Neben den eigentlichen Effekten auf die Behauptungen für die Verhältnisse von Korrelationslängen, die wir hier untersuchen, ist auch der Einfluß dieses Wechsels vom üblichen periodischen zum antiperiodischen Fall auf typische Konfigurationen der Systeme von Interesse. Daher scheinen hier einige Bemerkungen über charakteristische Unterschiede zwischen Systemen mit diesen unterschiedlichen Randbedingungen angebracht.

Zunächst stellt sich das technische Problem, den hier verwendeten Wolff–Cluster–Algorithmus von seiner üblichen Implementation auf den Spezialfall antiperiodischer Randbedingungen zu adaptieren. Die Berücksichtigung der Randbedingungen wird in unserem Fall erleichtert durch die Beschränkung der Betrachtung auf Nächste–Nachbar–Wechselwirkungen; daher muß der Vorzeichenwechsel über den Systemrand hinaus gemäß

$$\mathbf{s}(x + L_x, y) = -\mathbf{s}(x, y) \tag{4.52}$$

lediglich für die diesem Rand direkt benachbarten Spin–Positionen berücksichtigt werden: weiter reicht der „Blick“ der Wechselwirkung nicht hinaus. Daher beschränkt sich der Einfluß der antiperiodischen Randbedingungen auf die Existenz einer Naht von *anti*ferromagnetischen Bindungen entlang $\{(x, y)|x = 0\}$, denn aufgrund der skalarproduktartigen Wechselwirkung der $O(n)$–Spin–Modelle ist eine Umkehr der Orientierung eines der beiden an der Wechselwirkung beteiligten Spins äquivalent zu einem Vorzeichenwechsel der Wechselwirkungskonstante J vom ferromagnetischen ($J > 0$) zum antiferromagnetischen ($J < 0$) Fall. Für die Implementation eines Cluster–Update–Algorithmus ergibt sich entsprechend der Fortuin–Kasteleyn–Darstellung eine Modifikation der Wahrscheinlichkeit, mit der Bindungen über die Systemgrenzen hinweg gesetzt werden. Werden Bindungen im Innern des Systems mit der der Swendsen–Wang–Regel entsprechenden Wahrscheinlichkeit

$$p = 1 - \exp(-2K) \tag{4.53}$$

gesetzt, so wird für eine für das System fest gegebene Wechselwirkungskonstante $J = K/\beta > 0$ eine Bindung über eine mit antiperiodischen Randbedingungen versehene Systemgrenze hinweg mit der Wahrscheinlichkeit

$$p' = 1 - \exp(2K) \tag{4.54}$$

aktiviert. Es ist leicht einzusehen, daß es sich bei dieser Regel auch weiterhin um einen zulässigen MC–Update–Algorithmus handelt: Ergodizität ist trivialerweise gegeben, da auch in diesem Fall der Update *einzelner* Spins möglich bleibt und somit jede Konfiguration in endlicher Zeit erreicht werden kann; die Bedingung der detaillierten Balance ist, wie man an (3.7) abliest, erfüllt, da die angegebene Modifikation der Swendsen–Wang–Regel gerade der Änderung des Boltzmann–Faktors durch das Einführen einer antiferromagnetischen Bindung entspricht.

Wie wirken sich die Randbedingungen auf die Konfigurationen der Systeme aus? Abb. 4.15 zeigt einige typische Zustände der hier simulierten Systeme für die beiden Arten von Randbedingungen. Grundsätzlich bedingt die stark anisotrope Geometrie der Systeme eine entsprechende Anisotropie der Cluster–Verteilung: zumindest für periodische Randbedingungen dehnen sich die (geometrischen) Cluster in der transversalen, kurzen Richtung in der Regel nahezu über die gesamte Breite des Systems aus, während sie nur jeweils einen Bruchteil der Größe des Systems in longitudinaler Richtung ausmachen; es ergibt sich daher in Längsrichtung ein Muster von Querstreifen von entgegengesetzt orientierten Clustern. Während sich die periodischen Randbedingungen in einem häufigen Hinausreichen der sichtbaren Cluster–Strukturen über die transversalen Systemgrenzen hinweg manifestieren, erzeugen die antiperiodischen Randbedingungen eine Präferenz zur Spinumkehr an den Systemgrenzen und damit im Mittel eine Grenzfläche im System; in diesem Fall sind die stochastischen Cluster, wie sie durch die Bindungsstruktur definiert sind, entgegen dem üblichen Fall u.U. sogar größer als die sichtbaren, geometrischen Cluster. Da für jedes endliche System am kritischen Punkt die Magnetisierung im Gleichgewicht verschwindet und die Verwendung antiperiodischer Randbedingungen eine ortsfeste Grenzfläche am Rand des System definiert, entsteht im Inneren des Systems eine zweite, fluktuierende Grenzfläche entlang der y–Richtung, in der sich die Spinorientierung ein zweites Mal umkehrt.

Die durchschnittliche Cluster–Größe $\langle |C| \rangle$ skaliert nicht nur erwartungsgemäß mit der Größe der Systeme, sondern ist auch — wie in den Abbildungen zu erahnen — für die Systeme mit antiperiodischen Randbedingungen systematisch kleiner als für solche mit periodischen Randbedingungen. Für die hier zu messenden Korrelationen (diese anschauliche Erläuterung betrifft natürlich in erster Linie die lokale Magnetisierung) erklären sich hieraus die (auch später für die dreidimensionalen Systeme) um einen Faktor $2-3$ reduzierten Korrelationslängen und die damit einhergehende Schwierigkeit, diese aus einem deutlich reduzierten Plateau–Bereich der Schätzungen $\hat{\xi}(i)$ zu bestimmen.

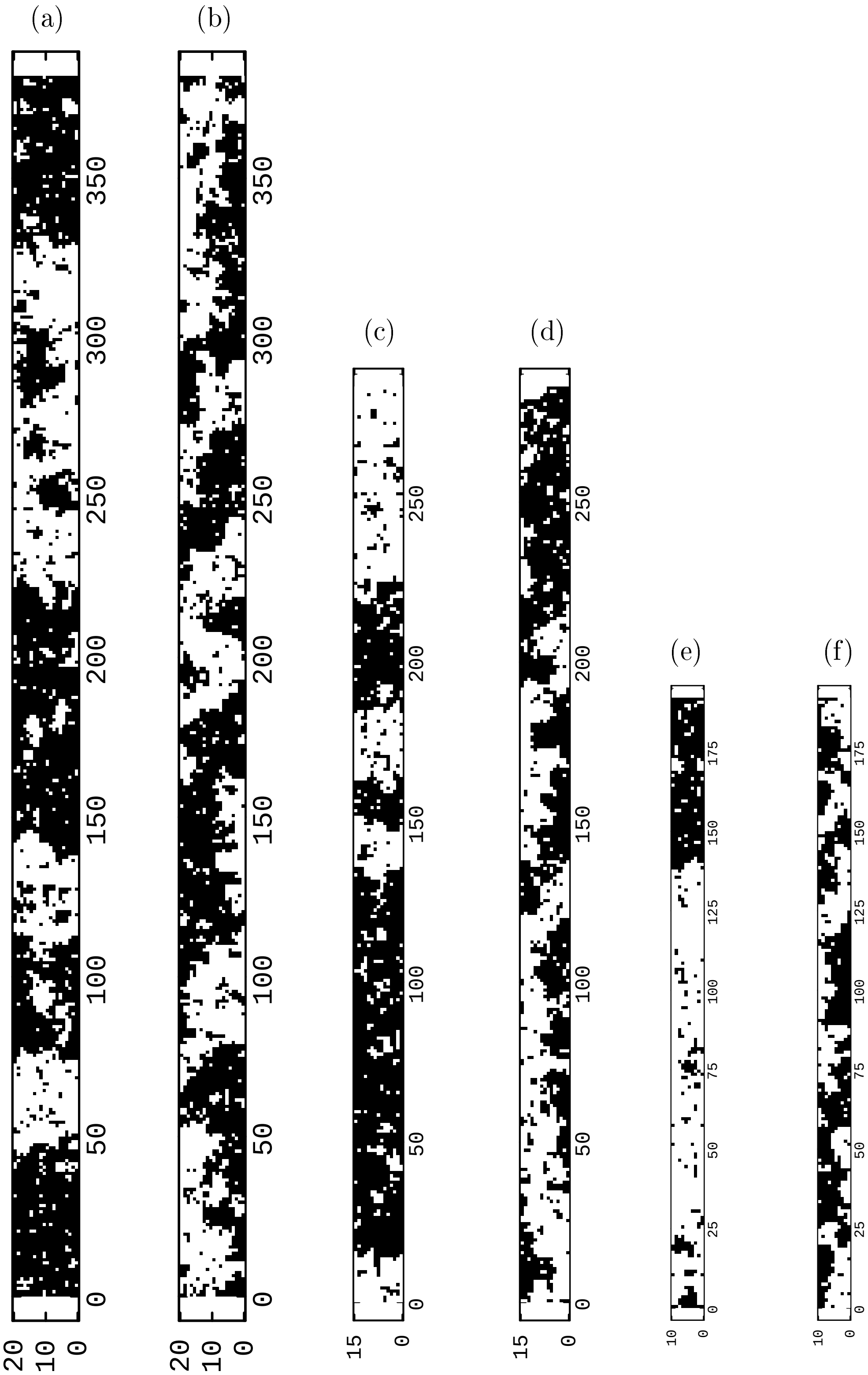


Abb. 4.15: Einige typische Konfigurationen der Streifensysteme für das 2D Ising–Modell mit periodischen ((a), (c), (e)) und antiperiodischen ((b), (d), (f)) Randbedingungen. Die Systemgrößen betrugen: 20×382 ((a), (b)), 15×286 ((c), (d)) und 10×192 ((e), (f)). Die schwarzen Bereiche entsprechen Gebieten mit $s = -1$, die weißen solchen mit $s = +1$. Die Systeme wurden mit einer gleichverteilt–zufälligen Konfiguration gestartet und äquilibriert.

Kapitel 5

2D Ising–Modell: Simulationen und Ergebnisse

Mit den bisher entwickelten Werkzeugen lassen sich die Korrelationslängen von lokaler Magnetisierung und Energiedichte in einzelnen Systemen zuverlässig und weitgehend „automatisiert“, also frei von subjektiven Unwägbarkeiten, bestimmen. Um zu einem möglichst genauen Endergebnis für die Amplituden A gemäß $\xi = AL$ und das Verhältnis ξ_σ/ξ_ϵ zu gelangen, bieten sich zwei verschiedene Wege an: verwendet man die verfügbare Rechenzeit vollständig auf die Simulation eines Systems (oder weniger Systeme) gegebener Größe, so läßt sich für dieses eine sehr hohe statistische Genauigkeit erzielen (dieser Weg wurde in der Arbeit von Weston [85] beschritten); zugleich ist diese Vorgehensweise jedoch unempfindlich gegenüber systematischen Fehlern, die mit der linearen Systemgröße L_x skalieren. Daher werden wir hier einen zweiten Weg wählen und einen gewissen Bereich von Systemgrößen L_x durch Simulationen systematisch durchlaufen. Auf diese Weise wird zwar die Länge der Zeitreihen der einzelnen Systeme (bei einer gegebenen zur Verfügung stehenden Rechenzeit) erheblich kleiner ausfallen, nur so läßt sich jedoch das prognostizierte Skalenverhalten der Korrelationslängen direkt überprüfen und gegebenenfalls Abweichungen von diesem Verhalten erkennen.

5.1 Simulationsparameter

Für das Ising–Modell in zwei Dimensionen wurden Systeme bis zur Größe 20×382 simuliert; die Systemgrößen in der y–Richtung wurden dabei entsprechend der aus den Vorsimulationen gewonnen Schätzungen für A_σ skaliert. Aus den in Kap. 4.2 angegebenen Gründen haben wir hier auf eine Differenzierung der Systemgrößen in Abhängigkeit von der Wahl der Randbedingungen verzichtet. Die Tabellen 5.1 und 5.2 geben einen Überblick über die für die Produktionsläufe relevanten Parameter der Systeme. Die integrierten Autokorrelationszeiten wurden mit der in Kap. 4.2 beschriebenen Binning–Methode bestimmt, ohne jedoch hierauf eine bemerkenswerte Rechenzeit zu verwenden, da die Werte nur zur überschlägigen Bestimmung einer sinnvollen Frequenz von Messungen zu dienen hatten (entsprechend wurde hier auch auf eine Fehlerrechung verzichtet); die Meßfrequenz wurde zu $1/\tau_{\rm int}$ festgelegt. Bei der Festlegung der Block–Länge für die Produktionsläufe war der bestimmende Faktor weniger durch die für die Jackknife–Analyse erforderliche statistische Unabhängigkeit der Block–Werte gegeben — entsprechend der gewählten Frequenz von Messungen sind ja schon die Einzel–Meßwerte nahezu unabhängig —, sondern vielmehr durch Restriktionen des verfügbaren Plattenspeicherplatzes.

Die Systeme mit periodischen Randbedingungen wurden als erste untersucht; daher sind hier einige Parameter im Vergleich mit den vorstehenden diesbezüglichen Bemerkungen noch etwas „übervorsichtig“ gewählt. Zudem wurde auf einige Anpassungen verzichtet, die bei den dreidimensionalen Systemen zur Anwendung kommen werden und daher oben erwähnt wurden; so ist die (systematische) Variation der gemessenen Autokorrelationszeiten für die Systeme mit antiperiodischen Randbedingungen so gering, daß hier eine einheitliche Meßfrequenz gewählt wurde. Mit Blick auf die Gültigkeit der zu überprüfenden Aussagen im FSS–Limes wurden zunächst, d.h. im Falle der periodischen Randbedingungen, die großen Systeme mit erhöhter Genauigkeit untersucht (etwa $3 \cdot 10^6$ Messungen). Im weiteren Verlaufe der Untersuchung wird sich herausstellen, daß es günstiger ist, die Rechenzeit auf die Analyse möglichst vieler Systemgrößen breit zu verteilen; daher wurde für die Systeme mit antiperiodischen Randbedingungen eine einheitliche Laufzeit aller Simulationen gewählt. Die Laufzeit der Simulationen skaliert nicht einfach mit der Anzahl der vorhandenen Spin–Positionen: die durchschnittliche Cluster–Größe, die die Laufzeit der Update–Schritte entscheidend bestimmt, hängt nämlich sowohl vom Verhältnis L_y/L_x als auch von der Wahl der Randbedingungen ab. Zudem benötigen die

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	periodische Randbedingungen							
5 × 96	100	200	352000	5	10	44	2048	512
6 × 116	150	200	352000	6	13	44	2048	512
7 × 134	200	250	352000	7	15	44	2048	512
8 × 154	200	250	352000	7	15	44	2048	512
9 × 172	250	300	384000	8	15	48	2048	512
10 × 192	300	300	30000	10	15	20	4096	512
11 × 210	300	350	416000	10	15	52	2048	512
12 × 230	350	350	416000	11	17	52	2048	512
13 × 250	400	450	416000	11	18	52	2048	512
14 × 268	400	450	448000	11	20	56	2048	512
15 × 286	500	500	84000	11	22	56	2048	1536
16 × 306	550	600	84000	11	22	56	2048	1536
17 × 324	600	650	84000	11	22	56	2048	1536
18 × 344	600	650	84000	13	22	56	2048	1536
19 × 362	650	700	84000	13	22	56	2048	1024
20 × 382	700	750	84000	14	23	56	2048	1536

Tab. 5.1: Simulationsparameter für die Produktionsläufe der 2D Ising–Systeme mit periodischen Randbedingungen. $t_{\text{Äquil}}(\sigma)$ bzw. $t_{\text{Äquil}}(\epsilon)$ entspricht der gemessenen zum Erreichen des Gleichgewichts benötigten Dauer von Wolff–Update–Schritten für die Korrelationsfunktionen $G(i)$ von Spin bzw. Energiedichte und einen Abstand i am Beginn des Plateaus. Bei den Produktionsläufen wurde mit den Messungen nach n_0 Schritten begonnen; sie erfolgten nach jeweils $1/f$ Wolff–Updates. Die n Messungen wurden zu n/μ Blöcken der Länge μ zusammengefaßt.

Messungen einen erheblichen Teil der gesamten Rechenzeit, da zur Bestimmung der Korrelationsfunktionen viele zeitaufwendige Multiplikationen notwendig sind. Um einen Anhaltspunkt zu geben: die gesamte Laufzeit des 20×382–Systems mit antiperiodischen Randbedingungen auf einem PentiumPro–System (180 MHz) betrug ca. 28 CPU–Stunden.

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	antiperiodische Randbedingungen							
5 × 96	200	300	20000	5	22	40	8192	512
6 × 116	200	300	20000	6	13	40	8192	512
7 × 134	200	350	20000	7	32	40	8192	512
8 × 154	250	350	20000	4	22	40	8192	512
9 × 172	250	350	20000	5	30	40	8192	512
10 × 192	250	400	20000	7	30	40	8192	512
11 × 210	300	450	20000	7	23	40	8192	512
12 × 230	300	500	20000	9	28	40	8192	512
13 × 250	350	500	20000	10	23	40	8192	512
14 × 268	350	600	20000	9	24	40	8192	512
15 × 286	350	600	20000	9	26	40	8192	512
16 × 306	450	650	20000	8	30	40	8192	512
17 × 324	500	650	20000	13	32	40	8192	512
18 × 344	600	800	20000	12	35	40	8192	512
19 × 362	700	1000	20000	15	30	40	8192	512
20 × 382	800	1100	20000	13	20	40	8192	512

Tab. 5.2: Simulationsparameter für die Produktionsläufe der 2D Ising–Systeme mit antiperiodischen Randbedingungen.

5.2 Ergebnisse: FSS der Korrelationslängen

Auf die Zeitreihen der Produktionsläufe wurden die im vorangegangenen Kapitel beschriebenen Analysemethoden angewandt: zunächst wurden mit Hilfe der Differenzenmethode die Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen von Spin und Energiedichte generiert. Mittels einer Jackknifing–Analyse wurde eine Schätzung für die Kovarianzmatrix der $\hat{\xi}(i)$ bestimmt. Auf eine Jackknife–Bias–Reduktion wurde hier aus den oben in Kap. 4.4.6 erwähnten Gründen verzichtet.

Zur Bestimmung des Plateau–Bereichs $i_{\min}, \ldots, i_{\max}$ der Schätzungen $\hat{\xi}(i)$ zur Mittelung wurde eine systematische Analyse der Funktion $\chi^2/g(i_{\min}, i_{\max})$ vorgenommen wie sie in Kap. 4.4.5 beschrieben wurde. Um als Voraussetzung hierfür eine Glättung des Plateau–Bereiches zu erreichen, wurde hier erstmals von der Wahl $\Delta = 1$ in (4.15) abgewichen; da der Abfall der Korrelationsfunktionen von den zugehörigen

Korrelationslängen bestimmt wird und diese ihrerseits mit der transversalen Systemgröße L_x skalieren, scheint es sinnvoll auch die verwendeten Abstände Δ entsprechend mit zu vergrößern: hält man ξ/Δ über die verschiedenen Systeme hinweg konstant, so vergleicht man für alle Systeme die gleichen Differenzen $G(i)-G(i-\Delta)$, so daß auch der Signal–Rausch–Abstand, abgesehen vom Einfluß der absoluten Systemgröße selbst, d.h. etwa vom Effekt der Projektion der Schichtenmethode etc., konstant bleibt. Da die Schwankungen der $G(i)$ in erster Linie im Falle der Korrelationsfuntkion der Energiedichte eine Rolle spielen, wurde für den Vergleich von Δ und ξ die Korrelationslänge ξ_ϵ verwendet und auf eine Differenzierung der Abstände Δ zwischen den Untersuchungen von ξ_σ und ξ_ϵ verzichtet. Was die Amplitude der Skalierung dieser Abstände Δ anbelangt, so ist zu beachten, daß eine Vergrößerung des Abstandes die Differenzenmethode auch immer weiter in den systematisch abweichenden Bereich kleiner Abstände i eingreifen läßt und so den dortigen Abfall der Schätzungen $\hat{\xi}$ glättet, was die exakte Bestimmung des minimal zulässigen $i_{\min}$ durch Beobachtung von χ^2/g erschwert. Als Kompromiß zwischen beiden Effekten wurde hier der Abstand Δ zu $\approx 2\xi_\epsilon$ gewählt, was zu den in den Tabellen 5.3 und 5.4 verzeichneten Werten führt.

Zum Vergleich mit den Aussagen aus Kap. 4.5 sind dort neben den Ergebnissen für die Korrelationslängen von lokaler Magnetisierung und Energiedichte in Abhängigkeit von der transversalen Systemgröße L_x auch die Ergebnisse für die durchschnittlichen Cluster–Größen $\langle |C| \rangle$ aufgeführt: in den Systemen mit antiperiodischen Randbedingungen sind die (stochastischen) Cluster des Wolff–Algorithmus im Durchschnitt nur etwa halb so groß wie in den entsprechenden Systemen mit periodischen Randbedingungen. Für die Korrelationslängen der Energiedichte verbleiben auch nach der Anwendung einer genaueren Analyse von χ^2/g noch relativ große Schwankungen der zugehörigen geschätzten Varianzen. Hier stehen jedoch insgesamt (wie in den Tabellen 5.3 und 5.4 zu erkennen ist) so wenige Werte–Tripel von $G(i)$ zur Verfügung, auf die sich die Differenzenmethode anwenden läßt, daß solche Schwankungen nicht zu vermeiden sind, denn diese Fehler fallen ja (im wesentlichen) mit der Wurzel des Kehrwertes der Länge $i_{\max} - i_{\min} + 1$ des eingehenden Bereiches ab. Um das Finite Size Scaling der Korrelationslängen zu veranschaulichen, kann man diese nun gegen die transversalen Systemgrößen L_x auftragen. Diese Graphen zeigen die Abbn. 5.1 und 5.2 für die beiden untersuchten Arten von Randbedingungen.

Man erkennt in allen Fällen klar ein lineares Skalenverhalten wie es der Voraussage entspricht. An der Größe der Fehlerbalken zeigt sich der schon erwähnte grundsätz-

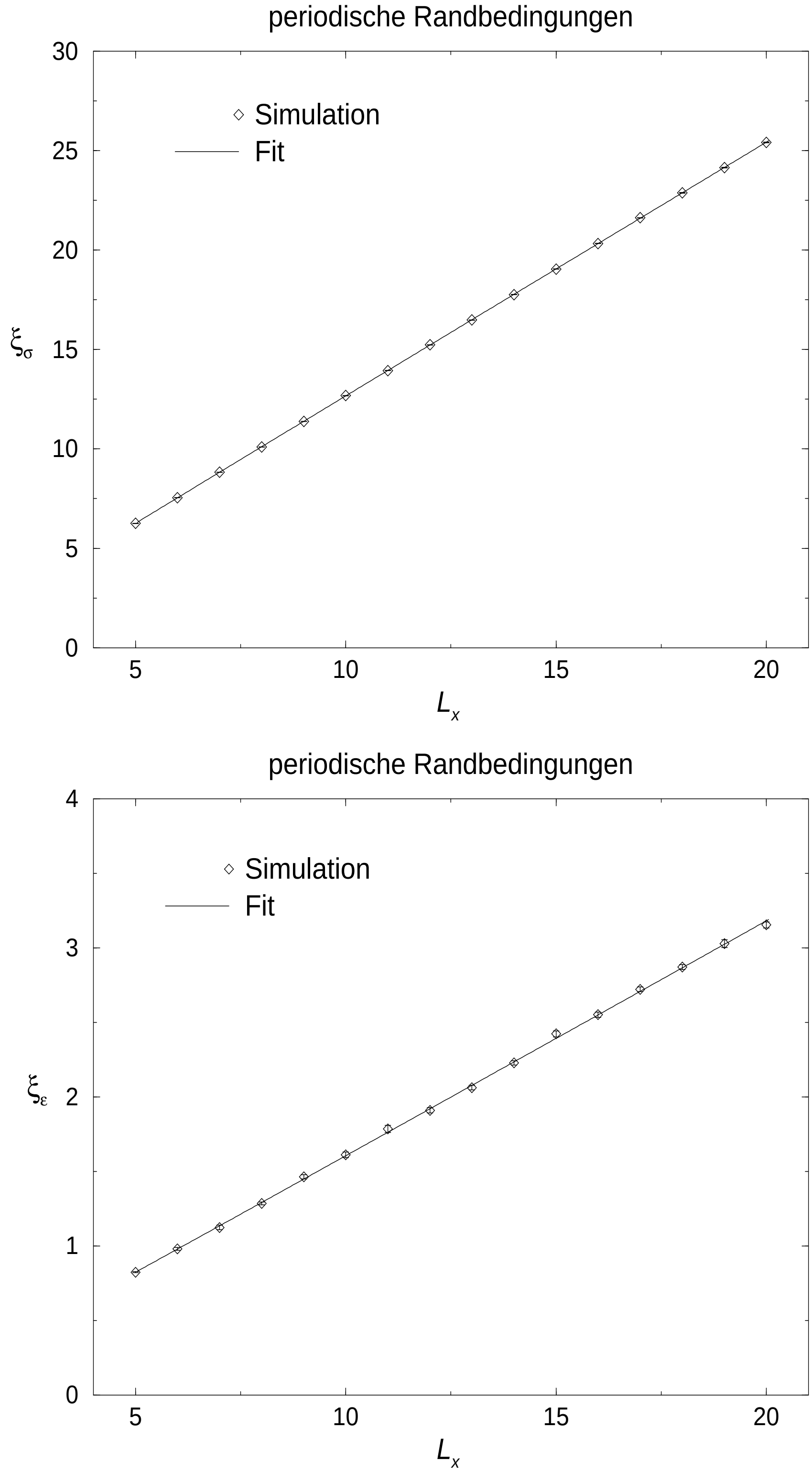
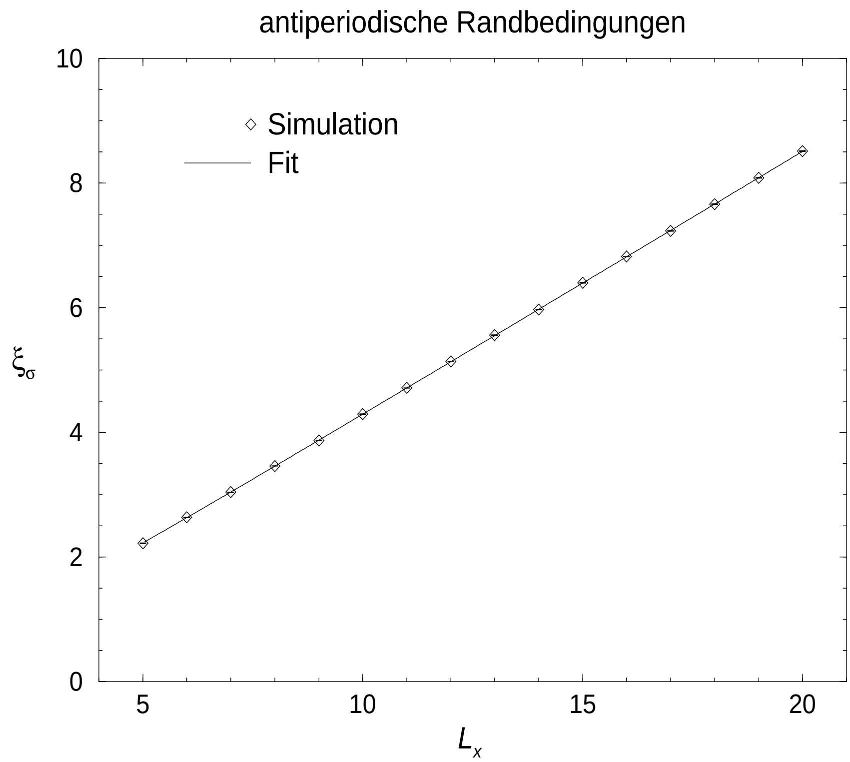


Abb. 5.1: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 2D Ising–Modells mit periodischen Randbedingungen für Systeme mit $L_x = 5$ bis $L_x = 20$. Die einzelnen Punkte entsprechen den Mittelwerten $\bar{\xi}$ über den Plateau–Bereich; die Fehler sind die entsprechenden Endergebnisse der Jackknife–Analyse.



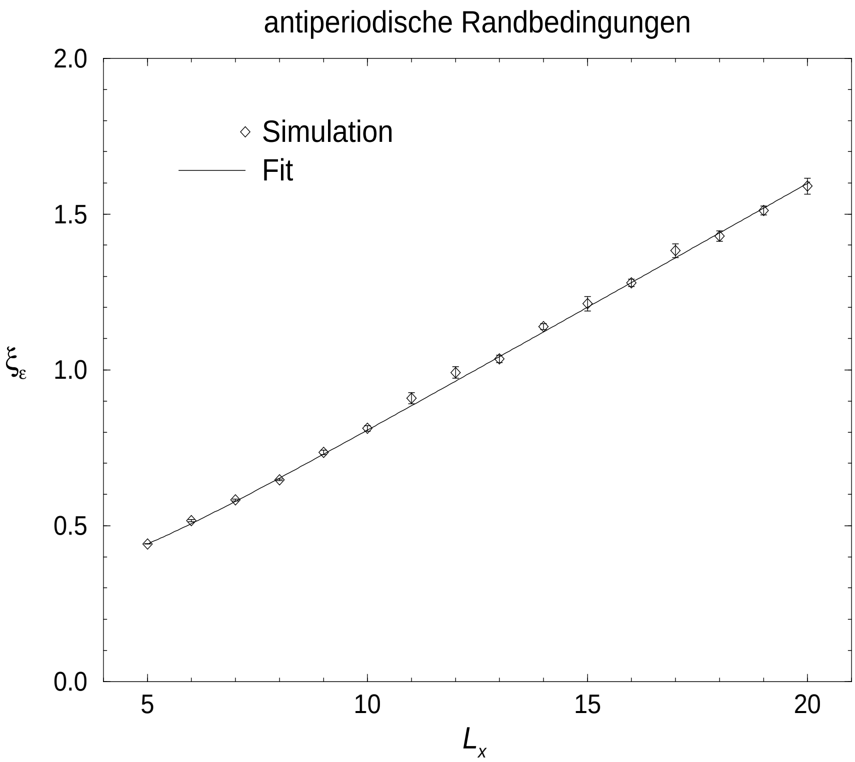


Abb. 5.2: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 2D Ising–Modells mit antiperiodischen Randbedingungen für Systeme mit $L_x = 5$ bis $L_x = 20$.

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle\vert C\vert\rangle}{V}$
	periodische Randbedingungen							
5 × 96	6.2539	0.0043	5 – 31	0.8236	0.0045	4 – 9	2	9.7%
6 × 116	7.5416	0.0048	5 – 48	0.9794	0.0089	5 – 9	2	9.2%
7 × 134	8.8334	0.0058	6 – 51	1.1234	0.0157	6 – 10	2	9.0%
8 × 154	10.1010	0.0059	6 – 55	1.2843	0.0088	6 – 10	3	8.7%
9 × 172	11.3825	0.0070	7 – 53	1.4632	0.0139	7 – 11	3	8.5%
10 × 192	12.6786	0.0076	8 – 80	1.6108	0.0198	8 – 15	3	8.3%
11 × 210	13.9429	0.0096	10 – 80	1.7848	0.0229	9 – 16	4	8.1%
12 × 230	15.2353	0.0094	11 – 82	1.9086	0.0176	9 – 18	4	7.9%
13 × 250	16.4869	0.0125	14 – 80	2.0612	0.0151	9 – 18	4	7.7%
14 × 268	17.7641	0.0133	14 – 82	2.2281	0.0138	9 – 19	4	7.7%
15 × 286	19.0516	0.0092	17 –118	2.4222	0.0202	12 – 23	5	7.6%
16 × 306	20.3316	0.0095	17 –105	2.5501	0.0179	12 – 24	5	7.4%
17 × 324	21.6315	0.0097	18 –115	2.7206	0.0153	12 – 28	5	7.3%
18 × 344	22.8831	0.0106	20 –110	2.8704	0.0159	13 – 26	6	7.2%
19 × 362	24.1491	0.0148	22 –112	3.0272	0.0260	14 – 26	6	7.1%
20 × 382	25.4183	0.0122	22 –120	3.1524	0.0192	14 – 27	6	7.0%

Tab. 5.3: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des zweidimensionalen Ising–Modells mit periodischen Randbedingungen. Die Größe $\langle|C|\rangle/V$ gibt den durchschnittlichen Anteil der Spins an, deren Orientierung bei einem Wolff–Cluster–Update umgekehrt wird. Mit diesem Faktor ist die Gesamtzahl der Wolff–Updates zu multiplizieren, um eine Angabe in Metropolis–Äquivalenten (Sweeps) zu erhalten.

liche Anstieg der Anforderungen an die Simulation beim Übergang von ξ_σ zu ξ_ϵ und analog beim Wechsel von periodischen zu antiperiodischen Randbedingungen. Auch wenn die Linearität in dieser Darstellung — abgesehen von statistischen Schwankungen — nahezu perfekt zu sein scheint, so täuscht dieser Eindruck dennoch: eine Auftragung nicht von $\xi(L_x)$ selbst, sondern der Amplitude $A = \xi/L_x$ wie sie in den Abbn. 5.3 und 5.4 gezeigt ist, offenbart für alle vier Korrelationslängen einen Verlauf, der erst asymptotisch für $L_x \to \infty$ konstant wird. Bei einer Bewertung dieser Graphen sollte man immer die eigentlichen FSS–Darstellungen Abbn. 5.1 und 5.2 im Auge behalten: es handelt sich bei den Korrektur–Plots doch um sehr sensible Darstellungen und es ist apriori keineswegs klar, daß sich Korrekturen zum führenden

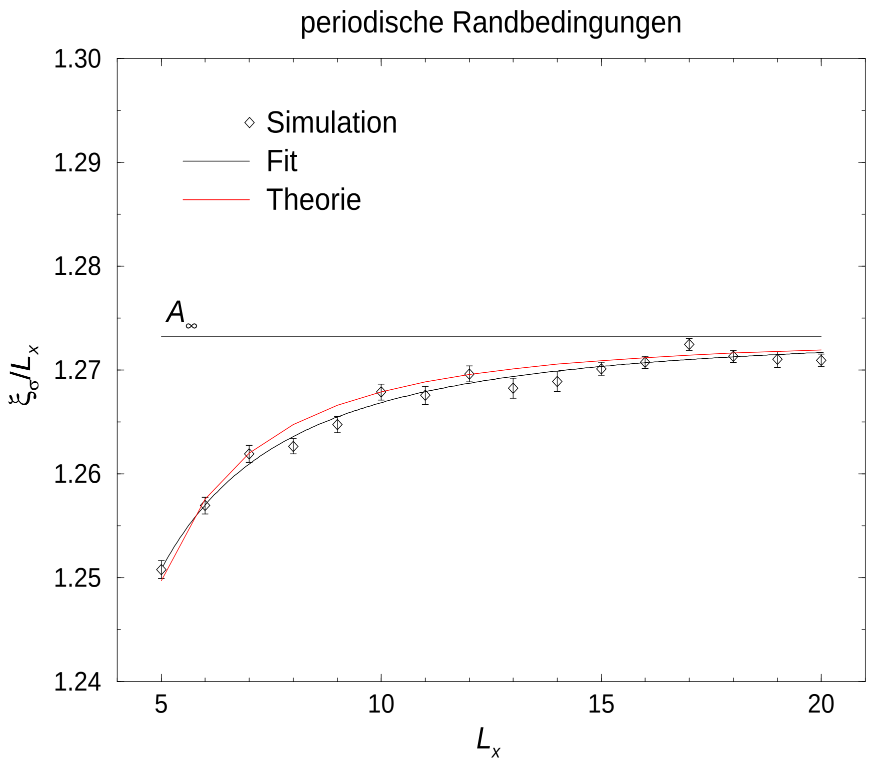
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	Δ	$\frac{\langle\vert C\vert\rangle}{V}$
	antiperiodische Randbedingungen							
5 × 96	2.2197	0.0054	7 – 21	0.4414	0.0008	2 – 4	1	5.4%
6 × 116	2.6318	0.0026	6 – 22	0.5161	0.0038	3 – 5	1	5.1%
7 × 134	3.0373	0.0034	7 – 20	0.5823	0.0028	3 – 5	1	4.8%
8 × 154	3.4576	0.0049	9 – 30	0.6468	0.0021	3 – 6	1	4.6%
9 × 172	3.8672	0.0025	7 – 31	0.7348	0.0071	4 – 6	1	4.5%
10 × 192	4.2889	0.0032	9 – 34	0.8123	0.0090	5 – 9	2	4.3%
11 × 210	4.7100	0.0038	10 – 33	0.9090	0.0183	6 – 9	2	4.2%
12 × 230	5.1347	0.0041	11 – 36	0.9914	0.0181	6 – 9	2	4.1%
13 × 250	5.5548	0.0049	12 – 43	1.0350	0.0121	6 – 11	2	4.0%
14 × 268	5.9680	0.0050	12 – 29	1.1385	0.0095	6 – 11	2	3.9%
15 × 286	6.3958	0.0051	13 – 40	1.2124	0.0233	7 – 10	2	3.8%
16 × 306	6.8158	0.0049	14 – 53	1.2789	0.0121	7 – 12	3	3.8%
17 × 324	7.2287	0.0044	14 – 52	1.3825	0.0222	8 – 13	3	3.7%
18 × 344	7.6570	0.0043	14 – 56	1.4290	0.0170	8 – 15	3	3.6%
19 × 362	8.0821	0.0054	16 – 55	1.5112	0.0144	8 – 16	3	3.6%
20 × 382	8.5087	0.0054	16 – 56	1.5894	0.0253	9 – 16	3	3.5%

Tab. 5.4: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des zweidimensionalen Ising–Modells mit antiperiodischen Randbedingungen.

FSS–Verhalten gegenüber den Fehlern der Datenpunkte noch auflösen lassen.

5.3 Korrekturen zum FSS

Es ist bekannt [75], daß die Theorie des FSS wie sie oben in Kap. 2 dargestellt wurde, für Systeme auf diskreten Gittern nicht exakt gültig ist, sondern daß vielmehr Korrekturen zum hieraus erhaltenen führenden Verhalten zu erwarten sind. Eine Abweichung vom rein linearen FSS–Verhalten wie sie die Abbn. 5.3 und 5.4 aufzeigen, ist mithin von dieser Warte aus nicht verwunderlich. Eine Analyse der Voraussetzungen der oben angegebenen Ableitungen zeigt den Ursprung solcher Korrekturen auf. Zunächst soll daher dieses Phänomen im Rahmen der genannten Theorien diskutiert werden, um dann im folgenden Abschnitt überlegen zu können, wie



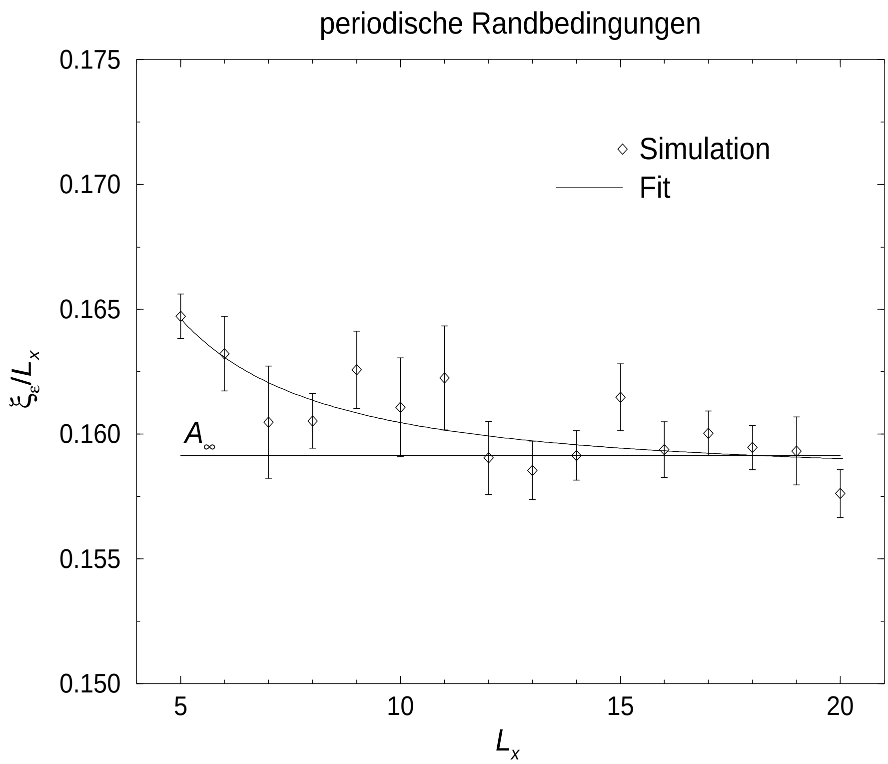


Abb. 5.3: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für periodische Randbedingungen. Die waagrechten Linien zeigen die führenden Amplituden $A_{\sigma,\infty} = 4/\pi \approx 1.27324$ bzw. $A_{\epsilon,\infty} = 1/(2\pi) \approx 0.15915$. Die schwarzen Kurven zeigen Fits an die Funktion $\xi = AL_x + BL_x^\alpha$, die Fitparameter sind in Kap. 5.3.2 angegeben. Die rote Kurve im oberen Bild zeigt die exakte Gitterlösung gemäß Tab. 5.5.

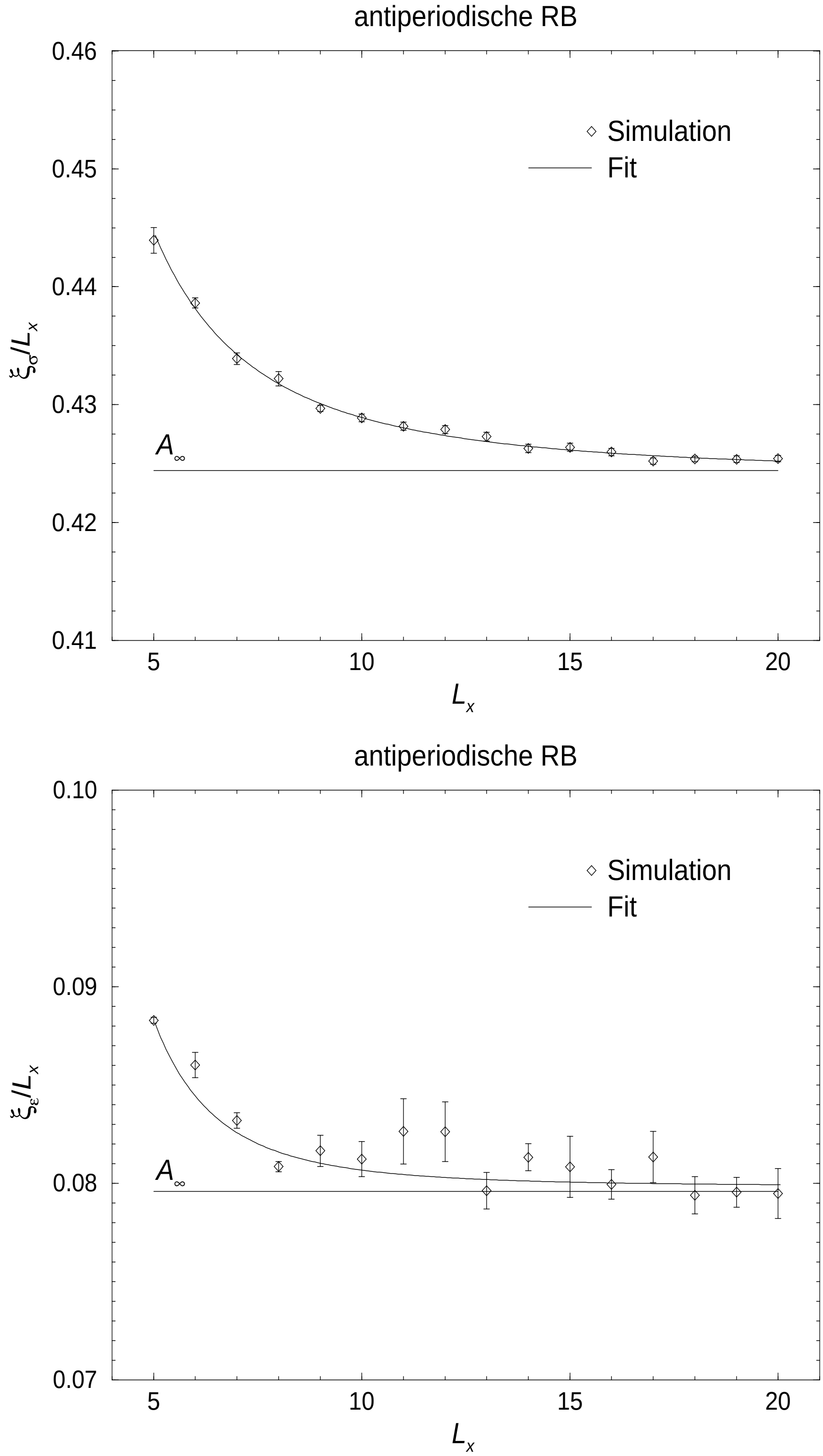


Abb. 5.4: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für antiperiodische Randbedingungen. Die waagrechten Linien zeigen die führenden Amplituden $A_{\sigma,\infty} = 4/(3\pi) \approx 0.42441$ bzw. $A_{\epsilon,\infty} = 1/(4\pi) \approx 0.07958$.

mit diesen Abweichungen im Hinblick auf das gesteckte Ziel einer Bestimmung der FSS–Amplituden umgegangen werden soll.

5.3.1 Korrekturen zum Kontinuumslimes

In der Ableitung der FSS–Gesetze in Kap. 2.4 wurden an zwei Stellen Näherungen gemacht, die exakte Gültigkeit der gewonnenen Aussagen lediglich für den FSS–Limes, der für $T = T_c$ durch den thermodynamischen Limes $L \to \infty$ gegeben ist, garantieren. Ein Fallenlassen dieser Vereinfachungen führt dann zu entsprechenden Korrekturen zum führenden FSS–Verhalten, die sich für endliche Systeme bemerkbar machen:

1. Das Verhalten der Skalenfelder bei einer Annäherung an den kritischen Punkt (bzw. dual dazu den thermodynamischen Limes) wurde als *linear* angenommen (vgl. Gl. (2.25)). In manchen Fällen ist es jedoch notwendig, das volle, u.U. nichtlineare, Skalenverhalten der Felder zu berücksichtigen; man spricht dann auch von *nichtlinearen Skalenfeldern*. Eine Berücksichtigung dieser Nichtlinearität erzeugt *analytische* Korrekturterme zum führenden FSS–Verhalten. In der Terminologie der konformen Feldtheorie sind es gerade die sekundären Operatoren eines Modells, die für solche Korrekturen verantwortlich sind.

2. Im FSS–Limes tragen nur die relevanten Skalenfelder, d.h. solche mit RG–Eigenwerten $y_i > 0$, zum FSS-Verhalten bei. Der Einfluß der *irrelevanten Skalenfelder* mit $y_i < 0$ nimmt mit jeder Anwendung der RG–Transformation gegenüber dem der relevanten Felder mit einem Potenzgesetz ab und verschwindet daher im Limes unendlich vieler Iterationen der Transformation wie man sie auf ein unendlich großes System anwenden kann. Für ein endliches System muß man jedoch die Reihe der Block–Transformationen spätestens dann abbrechen, wenn (zumindest in einer Raumrichtung) nur noch ein Block–Spin vorhanden ist, also nach endlich vielen Transformationen. Daher können in diesem Fall auch irrelevante Skalenfelder einen Einfluß auf das Skalenverhalten haben. Sie führen zu *nicht–analytischen* (singulären) Korrekturen des Skalenverhaltens, die durch Korrekturexponenten beschrieben werden.

Diese Korrekturen lassen sich im Kalkül der konformen Feldtheorie durch eine sog. *konforme Störungsrechung* berücksichtigen [76, 24]. Kennt man den Gehalt eines

Modells an primären Operatoren, so läßt sich das FSS–Verhalten etwa der Korrelationslängen durch eine Störungsreihe darstellen; bis zur ersten Ordnung erhält man:

$$\xi_\sigma{}^{-1}(L) = \frac{2\pi}{L}\left[x_n + 2\pi a_k \mathbf{C}_{nkn}\left(\frac{2\pi}{L}\right)^{x_k-2}\right], \tag{5.1}$$

wobei die x_n die Skalendimensionen der Störoperatoren darstellen und die Koeffizienten $\mathbf{C}_{nkn}$ aus einer sog. Operator–Produkt–Entwicklung resultieren und mit den Methoden der konformen Feldtheorie bestimmt werden können.

Die Herleitung des geschlossenen Ausdrucks (2.70) für die Korrelationsfunktionen eines primären Operators auf der Streifengeometrie $S^1 \times \mathbb{R}$ in Kap. 2.5 nahm jedoch keinerlei Bezug auf ein endliches System; sie verwendete vielmehr eine Transformation, die die gesamte komplexe Ebene, d.h. das unendlich große System in zwei Dimensionen, auf den Zylinder $S^1 \times \mathbb{R}$ „aufrollte". Dennoch wurden auch hier implizit die o.g. Näherungen gemacht, sie treten hier lediglich in einer zur obigen Formulierung komplementären Form auf: hier ist es nicht die Endlichkeit, sondern vielmehr die *Diskretizität* des betrachteten Gitters, die Anlaß zu Korrekturen gibt. Die Form (2.70) für die Korrelationsfunktion ist lediglich im *Kontinuumslimes* des Modells exakt, nicht jedoch für das Gittermodell. Der Einfluß einer solchen Diskretisierung läßt sich recht anschaulich anhand einer Fourier–Darstellung der betrachteten Größe (d.h. der Korrelationsfunktion) verstehen. Man geht dabei von der Kontinuumslösung des Problems aus und diskretisiert nicht die Funktion selbst, sondern ihre Fourier–Transformierte und nimmt die Rücktransformation des Ergebnisses als Ausdruck der Korrelationsfunktion des diskreten Modells. Dabei treten zunächst zwei Effekte auf:

1. Die Endlichkeit des Systems in y–Richtung in Verbindung mit den in dieser Richtung verwendeten periodischen Randbedingungen führt zu einer Diskretisierung der erlaubten Moden: nur solche „Schwingungen" sind möglich, die der Periodizität des Systems in y–Richtung genügen, d.h. in der Sprache der Wellenlehre gerade jene Wellenlängen, die die Ausbildung einer stehenden Welle in y–Richtung erlauben, also $\lambda = L_y/n, \quad n \in \mathbb{N}$. Dies führt gerade zu den Effekten endlicher Längen L_y, die in Kap. 4.2 diskutiert wurden und kann daher für die beobachteten Korrekturen zum FSS nicht verantwortlich sein, da die Streifenlänge L_y so groß gewählt wurde ($L_y \approx 15\xi_\sigma$), daß der Effekt zu vernachlässigen ist.

2. Die Diskretizität des Gittermodells spiegelt sich in der Existenz eines unteren

Abschneideradius für die erlaubten Moden wider: extrem kurzwellige Moden mit $\lambda < 2a$, wo a die Gitterkonstante ist, können auf dem diskreten Gitter nicht realisiert werden und sind daher unphysikalisch. Der Wegfall solcher kurzreichweitigen Fluktuationen des betrachteten Skalenoperators im Gittermodell sollte jedoch die zugehörigen Korrelationslängen gegenüber dem Kontinuums–Fall eher zu klein erscheinen lassen, wohingegen etwa ξ_σ im Falle periodischer Randbedingungen gemäß Abb. 5.3 für kleine Systeme zu größeren Werten hin abweicht. Eine quantitative Abschätzung in Anlehnung an [55] zeigt überdies, daß die hieraus resultierenden Korrekturen wesentlich kleiner sind als die beobachteten Abweichungen.

Die Durchführung der konformen Störungsrechnung für das Ising–Modell zeigt den Grund für die Abweichungen auf: in der ersten Ordnung der Störungsreihe (5.1) trägt als einziger Störoperator die Kombination

$$L_{-2}^2 - \frac{3}{5}L_{-4} + \bar{L}_{-2}^2 - \frac{3}{5}\bar{L}_{-4} \tag{5.2}$$

bei [24], d.h. nur dieser Operator hat in dieser Ordnung eine nichtverschwindende Koppelung $\mathbf{C}_{nkn}$. Dabei stellen die L_n Elemente der Virasoro–Algebra analog zu den Elementen l_n der Witt–Algebra gemäß (2.63) dar. Dieser Operator beschreibt die *Brechung der Rotationssymmetrie* durch das quadratische Gitter des diskreten Modells! Betrachtet man die Voraussetzungen für die Anwendung der Methoden der konformen Feldtheorie auf ein Modell der statistischen Mechanik, so ist für ein Gittermodell eben nicht nur die Homogenität (Invarianz unter Translationen) verletzt — diese Verletzung führt zu den oben im Rahmen der Fourieranalyse diskutierten Korrekturen —, sondern auch die Isotropie (Invarianz unter Rotationen). Da beide Bedingungen nur näherungsweise erfüllt sind, kann daher die Anwendung einer konformen Transformation auf ein solches System nur im FSS–Limes exakte Lösungen liefern, für endliche Gittergrößen müssen Korrekturen hinzukommen. Die Skalendimension des Operators (5.2) ist $x = 4$; daher sind die führenden Korrekturen im FSS der Korrelationslänge ξ_σ des Ising–Modells von der Ordnung L^{-2}.

Der spezielle Fall des zweidimensionalen Ising–Modells hat den Vorzug, daß zumindest für bestimmte Observablen und Randbedingungen nicht nur die exakte Lösung des Kontinuumsmodells, sondern sogar analytische Ausdrücke für die Gittermodelle verfügbar sind, so daß hier ein Vergleich der mit anderen Methoden gewonnenen Ergebnisse mit der exakten Lösung möglich ist. Insbesondere läßt sich für die Systeme auf der Streifengeometrie $S^1 \times \mathbb{R}$ mit periodischen Randbedingungen die Korrelationslänge der lokalen Magnetisierung auch für endliche Gitterlängen L_x exakt angeben [26, 28, 69]:

$$\begin{aligned} \frac{1}{\xi_\sigma(L_x, K)} &= \frac{1}{2}\left[(\gamma_1 + \gamma_3 + \cdots + \gamma_{2L_x-1}) - (\gamma_0 + \gamma_2 + \cdots + \gamma_{2L_x-2})\right], \\ \cosh\gamma_r &= \frac{\cosh^2 2K}{\sinh 2K} - \cos\frac{r\pi}{L_x}, \end{aligned} \tag{5.3}$$

wobei K der Koppelung βJ entspricht. Der Effekt endlicher Längen L_y ist hierin allerdings nicht berücksichtigt, d.h. die Ausdrücke gelten für $L_y = \infty$. Für die kritische Koppelung $K_c = \frac{1}{2}\ln(1+\sqrt{2})$ ergibt eine Entwicklung in erster Ordnung der Korrekturen nach $1/L_x$:

$$\xi_\sigma(L_x) = \frac{4}{\pi} L_x \left(1 - \frac{\pi^2}{24 L_x^2} + \cdots\right), \tag{5.4}$$

was das Ergebnis der konformen Störungstheorie für den Korrekturexponenten bestätigt[1]. Zum Vergleich mit den Ergebnissen dieser Simulation und zwischen der exakten Lösung (5.3) und der Berücksichtigung lediglich der Korrekturen der ersten Ordnung gemäß (5.4) sind die numerischen Werte für diese Größen und den für Simulationen heute etwa zugänglichen Bereich der L_y in Tab. 5.5 zusammengestellt. Zum Vergleich mit den Ergebnissen der Simulation ist der Verlauf der exakten Lösung $\xi_\sigma(L_x)$ in Abb. 5.3 zusätzlich zum Ergebnis des Fits gemäß Kap. 5.3.2 eingetragen.

Wahrscheinlich läßt sich auch im Falle antiperiodischer Randbedingungen mit Hilfe einer Transfermatrix–Rechnung die exakte Lösung für die Spin–Korrelationslänge angeben und eventuell ist dies auch für die Energie–Korrelationslängen möglich. Wir werden diese Fragen jedoch hier nicht weiter verfolgen, denn für die schließlich zu untersuchenden dreidimensionalen Modelle lassen sich solche Lösungen natürlich nicht angeben, so daß hier ein Vergleich nicht möglich sein wird.

5.3.2 Bestimmung der führenden Amplituden

Da im Falle der dreidimensionalen O(n)–Spin–Modelle der genaue funktionale Zusammenhang $\xi(L_x)$ nicht bekannt ist, muß spätestens dort versucht werden, den asymptotischen Grenzwert der Amplituden $A = \xi/L_x$ aus den gegebenen Daten für

[1] Die Korrekturamplitude läßt sich mit den Methoden der konformen Feldtheorie ebenfalls berechnen und auch dieses Ergebnis ist identisch mit der angegebenen exakten Lösung.

L_x	$\xi_\sigma^{\text{exakt}}(L_x)$	$\xi_\sigma^{\text{exakt}}/L_x$	$\xi_\sigma^{\text{2.Ordg.}}(L_x)$	$\xi_\sigma^{\text{2.Ordg.}}/L_x$
5	6.24855	1.24971	6.26148	1.25230
6	7.54523	1.25754	7.55217	1.25870
7	8.83387	1.26198	8.83788	1.26255
8	10.11796	1.26474	10.12047	1.26506
9	11.39929	1.26659	11.40098	1.26678
10	12.67884	1.26788	12.68004	1.26800
11	13.95716	1.26883	13.95804	1.26891
12	15.23458	1.26955	15.23524	1.26960
13	16.51132	1.27010	16.51184	1.27014
14	17.78754	1.27054	17.78795	1.27057
15	19.06335	1.27089	19.06369	1.27091
16	20.33884	1.27118	20.33911	1.27119
17	21.61405	1.27141	21.61427	1.27143
18	22.88903	1.27161	22.88922	1.27162
19	24.16383	1.27178	24.16399	1.27179
20	25.43847	1.27192	25.43861	1.27193
21	26.71298	1.27205	26.71310	1.27205
22	27.98737	1.27215	27.98747	1.27216
23	29.26165	1.27225	29.26174	1.27225
24	30.53585	1.27233	30.53593	1.27233
25	31.80997	1.27240	31.81004	1.27240
26	33.08403	1.27246	33.08409	1.27246
27	34.35802	1.27252	34.35808	1.27252
28	35.63196	1.27257	35.63201	1.27257
29	36.90585	1.27262	36.90589	1.27262
30	38.17969	1.27266	38.17973	1.27266
31	39.45350	1.27269	39.45354	1.27269
32	40.72727	1.27273	40.72730	1.27273
33	42.00101	1.27276	42.00104	1.27276
34	43.27472	1.27279	43.27474	1.27279
35	44.54840	1.27281	44.54842	1.27281

Tab. 5.5: Exakte Werte für die Spin–Korrelationslänge des 2D Ising–Modells auf der Geometrie $S^1 \times \mathbb{R}$ mit periodischen Randbedingungen. Die Spalten 2 und 3 zeigen die Korrelationslängen bzw. Amplituden gemäß der vollen Lösung (5.3), die Spalten 4 und 5 die Entwicklung bis zur zweiten Ordnung (d.h. bis zur ersten Ordnung der Korrektur) gemäß (5.4). Der Verlauf der Amplituden ist zu vergleichen mit der Amplitude $A_\infty = 4/\pi \approx 1.27324$.

endliche Längen L_x abzuschätzen. Von der Aufgabenstellung her wäre hierzu eine Methode der numerischen Extrapolation geeignet, da diese keine speziellen Voraussetzungen an die genaue funktionale Form des Verlaufs der $\xi(L_x)$ machen, sondern lediglich allgemeine Vorbedingungen wie etwa die Darstellbarkeit der Folge durch eine Potenzreihe erfordern. Solche Extrapolationsmethoden werden zu eben diesem Zweck etwa auf Ergebnisse aus Transfermatrix–Rechnungen angewandt [46]; dort handelt es sich um den Ausgleich von systematischen, glatt verlaufenden Abweichungen von einem intendierten Verhalten. In unserem Falle liegen jedoch statistisch fluktuierende Daten vor, die eine solche einfache Fortschreibung des beobachteten Verhaltens über den Rand des untersuchten Parameterbereiches hinaus nicht gestatten. Daher muß man hier Zuflucht nehmen zu entsprechenden statistisch fundierten Werkzeugen, d.h. Methoden der Regressionsrechnung.

Diese Vorgehensweise hat den Nachteil, daß die vorhandenen Daten nur an einen Verlauf von vorgegebener funktionaler Form angepaßt werden können (*Fit*); da die genaue funktionale Abhängigkeit $\xi(L_x)$ (abgesehen von ξ_σ im Falle periodischer Randbedingungen) nicht bekannt ist, muß man sich mit der allgemeinen Form einer potenzartigen Korrektur zum rein linearen FSS–Verhalten begnügen, man paßt also die Daten an eine Funktion der Form

$$\xi(L_x) = AL_x + BL_x^\alpha \tag{5.5}$$

an, indem im Raum der Parameter (A, B, α) diejenige Kombination gesucht wird, die die Abweichungsquadrate der zugehörigen Funktionswerte von den Datenpunkten minimiert (*Least–Square–Fit*). Für den Exponenten α der Korrektur könnte man den sich aus der exakten Lösung ergebenden festen Wert $\alpha = -1$ einsetzen[2]; da man aufgrund des Vorhandenseins höherer Ordnungen der Korrektur bei einem Fit der Form (5.5) jedoch einen *effektiven* Exponenten α sehen wird, der vom Exponenten der ersten Korrekturordnung abweicht, erzeugt ein festes α einen systematischen Fehler. Hier im zweidimensionalen Fall werden wir auf eine feste Vorgabe von α verzichten.

Mit feldtheoretischen Methoden lassen sich auch in drei Raumdimensionen die Exponenten der ersten Korrekturordnung in den FSS–Gesetzen gemäß (5.4) für die O(n)–Spin–Modelle näherungsweise angeben [88]; ihre Verwendung als feste Parameter α für den Fit–Prozeß ist jedoch auch dort von fraglichem Wert, da für jeden

[2]Die *Exponenten* der ersten Korrektur zum FSS sind unabhängig von der betrachteten Größe, nicht jedoch die zugehörigen *Amplituden*.

Bereich von endlichen Längen L_x, der als Ausgangsmaterial zur Verfügung steht, auch höhere Ordnungen der Korrektur eine Rolle spielen, so daß eine erzwungene Anpassung an ein Gesetz der Form (5.5) nicht für den theoretisch zu erwartenden Exponenten der ersten Korrekturordnung optimal wird, sondern für einen effektiven Exponenten α, der seinerseits vom betrachteten Parameterbereich der L_x abhängig ist. Darüber hinaus sind diese feldtheoretischen Werte für die Korrekturexponenten selbst noch mit einem Fehler behaftet, der in diesem Fall zu berücksichtigen wäre. Ob die Festlegung des Korrekturexponenten α auf den entsprechenden feldtheoretischen Wert dennoch von Vorteil ist, muß sich bei der Analyse der Daten für die dreidimensionalen Systeme unten zeigen.

Die erzwungene Mangelhaftigkeit dieser Vorgehensweise wird in unserem Falle etwas gemildert durch den Umstand, daß von dem so bestimmten Parametersatz (A, B, α) hier nur die führende Amplitude A von Interesse ist, da für sie die zu untersuchende lineare Amplituden–Exponenten–Relation gilt. Mit anderen Worten: es geht hier lediglich um eine *Korrektur*, ein Herausrechnen der Abweichungen vom rein linearen FSS–Verhalten, nicht um eine quantitative Bestimmung der Art (Amplitude und Exponent) der Korrekturen zum FSS, die hier weder gewünscht noch anhand des vorliegenden Datenmaterials zufriedenstellend möglich ist. Vor diesem Hintergrund wurde versucht, den verbleibenden Fehler durch Nichtbeachtung der höheren Korrekturen durch Beobachtung der Güteparameter des Fits zu minimieren. Dazu wurden für den Fit sukzessive Punkte am unteren Ende des L_x–Bereichs, wo höhere Korrekturen naturgemäß die größte Rolle spielen, entfernt und die Entwicklung der Fit–Parameter über die einzelnen Fits hinweg untersucht. Betrachtet wurden dabei die Größe χ^2/g und der daraus abgeleitete Güteparameter Q, der die Wahrscheinlichkeit angibt, daß man den beobachteten oder einen größeren Wert für χ^2/g erhält, wenn das verwendete Modell richtig ist (d.h. die resultierenden Parameter stimmen); mit anderen Worten: anhand von Q läßt sich entscheiden, ob die beobachteten Schwankungen der Meßwerte um die Fit–Kurve tatsächlich rein zufälliger Natur sind oder ihre Ursache in einem Fehler in der unterstellten funktionalen Abhängigkeit $\hat{\xi}(L_x)$ haben. Ein gebräuchlicher unterer Grenzwert für eine gerade noch akzeptable Güte liegt bei $Q = 0.001$ [74].

Bei diesem Prozeß galt es abzuwägen zwischen dem Ziel einer Maximierung der Güte Q bzw. Minimierung von χ^2/g und einer Minimierung des statistischen Fehlers für den geschätzten Parameter A; denn: läßt man zunehmend mehr Punkte weg, so wird es immer „leichter“, die gegebene Funktion an die verbleibenden Punkte anzupassen — bis zum Extremfall von zwei oder drei Punkten, die sich mit einer solchen Funktion schlicht interpolieren lassen. Das alleinige Ziel einer Optimierung der Fit–Güte ist daher nicht sinnvoll. Für die vier hier betrachteten Korrelationslängen lag dabei jedoch das Optimum bei der Berücksichtigung des vollen Parameterbereichs der L_x; dies muß jedoch für die unten zu untersuchenden dreidimensionalen Modelle nicht mehr notwendigerweise der Fall sein.

Die Ergebnisse der Fit–Prozedur sind als durchgezogene Kurven in den Abbn. 5.3 und 5.4 eingetragen. Die Ausdehnung dieser Kurven auf der L_x–Achse kennzeichnet das Gebiet der L_x, das in den Fit einging. Als Ergebnisse der Fits erhält man für periodische Randbedingungen:

$$\begin{array}{|rrl|} A_\sigma = & 1.27374(81) & (L_x = 5-20) \\ B_\sigma = & -0.37(11) & \\ \alpha_\sigma = & -0.73(19) & \\ \chi^2/g = & 1.4 & \\ Q = & 0.15 & \\ \hline A_\epsilon = & 0.1583(17) & (L_x = 5-20) \\ B_\epsilon = & 0.07 \pm 0.11 & \\ \alpha_\epsilon = & -0.53 \pm 1.06 & \\ \chi^2/g = & 0.79 & \\ Q = & 0.67 & \end{array} \tag{5.6}$$

Diese Ergebnisse für die Amplituden A sind zu vergleichen mit den theoretisch zu erwartenden Werten $A_\sigma = 4/\pi \approx 1.27324$ bzw. $A_\epsilon = 1/(2\pi) \approx 0.15915$ und stimmen mit diesen sehr gut überein. Der gegenüber der ersten Korrektur $\alpha = -1$ in (5.4) nach oben abweichende Schätzwert für den Exponenten der Korrektur zum FSS von ξ_σ zeigt den wachsenden Einfluß der höheren Korrekturordnungen für kleiner werdende Längen L_x: gemäß Tab. 5.5 ergibt hier die Berücksichtigung aller Ordnungen der Korrektur gegenüber der ersten Korrekturordnung kleinere Werte für ξ_σ, so daß man bei einer Festlegung auf ein Gesetz der Form (5.5) einen Exponenten $|\alpha| < 1$ erwartet. Die großen Fehler in der Schätzung von Amplitude und Exponent der Korrektur vor allem der Energie–Korrelationslänge zeigt, daß die Genauigkeit der Daten hier für eine quantitative Analyse nicht ausreicht (was allerdings auch nicht Ziel der Arbeit ist). Für ξ_ϵ sind die Abweichungen vom linearen Verhalten gegenüber den Schwankungen der einzelnen Punkte sogar so gering, daß es sinnvoll erscheint, als Alternative zum angegebenen Fit eine Anpassung an eine Gerade gemäß

$$\xi_\epsilon = A' L_x \tag{5.7}$$

vorzunehmen und dabei wiederum die Werte für kleine L_x, wo sich in Abb. 5.4 noch Abweichungen erkennen lassen, sukzessive auszulassen. Es ergibt sich folgendes Bild:

$$\begin{aligned} A'_\epsilon &= 0.15927(36) \quad (L_x = 12 - 20) \\ \chi^2/g &= 0.86 \\ Q &= 0.55 \end{aligned} \tag{5.8}$$

Man erhält einen innerhalb des (nun wesentlich kleineren) Fehlers weiterhin sehr gut mit dem theoretischen Wert verträgliches Ergebnis, hat nun aber einen möglichen systematischen Fehler nicht mehr unter Kontrolle; da der Wert etwas zu hoch liegt, kann man vermuten, daß sich hierin angesichts des Verhaltens gemäß Abb. 5.3 die Vernachlässigung der Korrekturen zum FSS bemerkbar macht.

Analog erhält man für antiperiodische Randbedingungen:

$$\begin{array}{|rrl|} \hline A_\sigma = & 0.42410(30) & (L_x = 5 - 20) \\ B_\sigma = & 0.60(14) & \\ \alpha_\sigma = & -1.10(13) & \\ \chi^2/g = & 1.04 & \\ Q = & 0.41 & \\ \hline A_\epsilon = & 0.07984(38) & (L_x = 5 - 20) \\ B_\epsilon = & 1.91 \pm 1.29 & \\ \alpha_\epsilon = & -2.37(44) & \\ \chi^2/g = & 2.1 & \\ Q = & 0.01 & \\ \hline \end{array} \tag{5.9}$$

Hier sind die theoretischen Amplituden gegeben durch $A_\sigma = 4/(3\pi) \approx 0.42441$ und $A_\epsilon = 1/(4\pi) \approx 0.07958$, wiederum in guter Übereinstimmung mit den Werten der Simulation. Die vergleichsweise schlechten Parameter der Fit–Güte im Falle der Energie–Korrelationslänge sind fast ausschließlich auf den Einfluß des zu niedrig gelegenen Punktes mit $L_x = 8$ zurückzuführen (vgl. Abb. 5.4), der aufgrund seines besonders kleinen statistischen Fehlers ein großes Gewicht hat. Versieht man ihn probehalber mit einem mit seinen Nachbarpunkten vergleichbaren Gewicht ($\sigma = 0.005$), so verbessern sich die Fit–Parameter sofort auf $\chi^2/g = 1.17$ bzw. $Q = 0.29$, während sich der Wert für die Amplitude A gerade mal um 15% des Fehlers verändert. Das Auftreten eines solchen „Ausreißers“ ist angesichts des kleinen Plateau–Bereichs für $G_\epsilon(i)$ im Falle antiperiodischer Randbedingungen wohl auch nicht ungewöhnlich. Im Vergleich zum Verhalten der Energie–Korrelationslänge für die Systeme mit periodischen Randbedingungen zeigt sich hier eine so deutliche Abweichung vom linearen FSS–Verhalten, daß ein Fit an eine Funktion der Form (5.7) nicht sinnvoll erscheint.

Für die Amplitudenverhältnisse erhält man schließlich:

$$\boxed{\begin{array}{ll} A_\sigma/A_\epsilon = 8.048(87) & \text{für periodische RB} \\ A_\sigma/A_\epsilon = 5.312(25) & \text{für antiperiodische RB,} \end{array}} \tag{5.10}$$

in guter Übereinstimmung mit den theoretisch zu erwartenden Werten A_σ/A_ϵ von 8 für periodische Randbedingungen und $5\frac{1}{3}$ für antiperiodische Randbedingungen. Die Skalendimensionen für die beiden primären Operatoren des zweidimensionalen Ising–Modells sind gegeben durch

$$x_\sigma = \frac{\beta}{\nu} = \frac{1/8}{1} = 1/8 \quad x_\epsilon = \frac{1-\alpha}{\nu} = \frac{1-0}{1} = 1, \tag{5.11}$$

so daß sich wie erwartet eine lineare Beziehung zwischen Amplituden und Skalendimensionen gemäß

$$A_\sigma/A_\epsilon = 8 = x_\epsilon/x_\sigma \tag{5.12}$$

für die Systeme mit periodischen Randbedingungen ergibt, nicht jedoch für die Systeme mit antiperiodischen Randbedingungen.

Nimmt man nun diese lineare Amplituden–Exponenten–Relation für den Fall periodischer Randbedingungen als gegeben an, so läßt sich die universelle Amplitude A gemäß

$$\xi_{\sigma/\epsilon} = A/x_{\sigma/\epsilon} \tag{5.13}$$

ebenfalls aus den Simulationsergebnissen bestimmen. Zieht man dazu die (genauer zu bestimmende) Amplitude der Spin–Korrelationslänge heran, so erhält man:

$$A = 0.1592(1), \tag{5.14}$$

in sehr guter Übereinstimmung mit dem exakten Wert $A = 1/(2\pi) \approx 0.15915$.

Kapitel 6

Das 3D Ising–Modell

Nachdem sich die in den Kapiteln 3 und 4 beschriebenen Simulations– und Analysemethoden bei der Untersuchung des FSS der Korrelationslängen des zweidimensionalen Ising–Modells bewährt haben, indem die so erhaltenen Ergebnisse gegenüber den hier verfügbaren exakten Ergebnissen bestehen konnten, läßt sich nun mit entsprechendem Vertrauen in die zu erhaltenden Resultate die Untersuchung der $\mathrm{O}(n)$–Spin–Modelle in drei Dimensionen beginnen, um die in Kap. 2.6 beschriebenen Vermutungen für dreidimensionale Systeme bestätigen oder widerlegen zu können.

Wir werden hier zunächst notwendige Änderungen der Simulations– und Analysewerkzeuge zur Anpassung an die dreidimensionale Geometrie beschreiben sowie zusätzlich notwendige Bemühungen anführen, den Einfluß der hier auftretenden Unsicherheit über die Position des kritischen Punktes auf die Endergebnisse der Korrelationslängen zu berücksichtigen, um dann die Anwendung des solchermaßen modifizierten Instrumentariums auf das dreidimensionale Ising–Modell und die hieraus resultierenden Ergebnisse vorzustellen.

6.1 Modifikationen für dreidimensionale Systeme

Die zu den zweidimensionalen Zylindern $S^1 \times \mathbb{R}$ analoge dreidimensionale Zylindergeometrie mit Torusquerschnitt gemäß $S^1 \times S^1 \times \mathbb{R}$ wird für die Simulation wiederum als rechteckiges Gitter realisiert, jetzt mit Längen (L_x, L_y, L_z), wobei nun die „lange“ Richtung L_z die Rolle der Länge L_y in den zweidimensionalen Streifensystemen übernimmt. Das System ist in den beiden transversalen Richtungen x und

y isotrop, d.h. $L_x = L_y$, die in diesen Richtungen verwendeten periodischen bzw. antiperiodischen Randbedingungen treten nun paarig auf, d.h.

$$\begin{aligned} \mathbf{s}(x + L_x, y, z) &= \pm\mathbf{s}(x, y, z), \\ \mathbf{s}(x, y + L_y, z) &= \pm\mathbf{s}(x, y, z) \end{aligned} \tag{6.1}$$

mit für beide Richtungen gleichen Vorzeichen + oder −; in z–Richtung werden wiederum in allen Fällen periodische Randbedingungen verwendet. Um die Forderung $L_z \gg L_x$ zu erfüllen, wurde erneut das Verhältnis L_z/ξ_σ betrachtet (vgl. Kap. 4.2), da wiederum die Korrelationslänge der lokalen Magnetisierung (wie Vorsimulationen ergaben) als die deutlich größere für die Dimensionierung der Systeme in der longitudinalen Richtung maßgeblich ist. Der für die Dimensionierung notwendige Richtwert für die Amplitude A_σ wurde für periodische Randbedingungen zu $A_\sigma \approx 0.8$ und für antiperiodische Randbedingungen zu $A_\sigma \approx 0.25$ bestimmt. Die hieraus resultierenden Systemlängen L_z sind weiter unten im Zusammenhang mit den übrigen Simulationsparametern aufgeführt, vgl. Tabn. 6.2 und 6.3. Es wurden hierbei entsprechend den Amplituden für die beiden Arten von Randbedingungen verschiedene Systemgrößen L_z gewählt, um für die Systeme mit antiperiodischen Randbedingungen, die hier ja von besonderem Interesse sind, längere Produktionsläufe zu ermöglichen. Für die Systeme mit antiperiodischen Randbedingungen wurde die Skalierung der Längen L_z für die Systeme mit kleinen transversalen Längen L_x unterlassen, da hier das aus der Amplitude A_σ folgende L_z so klein ausfallen würde, daß die Ausbildung eines Plateaus–Bereiches der Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen nicht mehr zu beobachten wäre; stattdessen wurde hier eine konstante Länge $L_z = 50$ angenommen bis die aus der Bedingung $L_z/\xi_\sigma \approx 15$ resultierenden Längen diesen Wert eingeholt hatten.

Zur Bestimmung der Korrelationsfunktionen verwenden wir auch hier wieder die in Kap. 4.1.1 beschriebene „Schichtenmethode“; die Bildung der Schichtvariablen, d.h. die Projektion der Spins bzw. Energiedichten auf die z–Achse, umfaßt hier in entsprechender Verallgemeinerung die Summation über die zweidimensionalen Flächen $\{(x, y, z) \,|\, z = \mathrm{const}\}$, es erfolgt damit eine Transformation auf ein effektiv eindimensionales Problem. Zur Bestimmung der Korrelationslängen kommt die Differenzenmethode zur Anwendung; die charakteristischen Abstände Δ wählen wir analog zum zweidimensionalen Fall so, daß für alle Systeme etwa $\Delta = 2\xi\epsilon$ erfüllt ist. Sind die Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen einmal bestimmt, so ist der weitere Fortgang der Analyse, d.h. vor allem die Fehlerrechnung und Bestimmung des Plateau–Bereichs, vollkommen unabhängig von der Dimension der zugrundeliegenden Geometrie (und im übrigen auch innerhalb der Klasse der O(n)–Spin–Modelle vom betrachteten Modell), so daß diese vollkommen analog zum Fall des zweidimensionalen Ising–Modells erfolgen kann. Zum „Beweis“, daß unsere Methode der Bestimmung der Korrelationslängen auch im dreidimensionalen Fall sinnvolle Ergebnisse erbringt, sind in Abb. 6.1 einige Beispiele für die Schätzungen $\hat{\xi}(i)$ der Korrelationslängen zusammengefaßt.

6.2 Die kritische Temperatur: Reweighting

Für das zweidimensionale Ising–Modell ist die Temperatur des Phasenüberganges exakt bekannt. Dies konnten sich die Simulationen zunutze machen, indem sie in die Entscheidungsregeln für die Cluster–Bildung diese Temperatur mit Maschinengenauigkeit eingehen ließen. Für dasselbe Modell in drei Dimensionen, wie auch die O(n)–Spin–Modelle mit $n > 1$ auf dieser Geometrie, ist der Punkt des Phasenübergangs aus Simulationen und Reihenentwicklungen lediglich numerisch mit einer begrenzten Genauigkeit verfügbar. Da in der Nähe des Phasenüberganges viele Observablen sehr empfindlich auf Temperaturänderungen reagieren (etwa da sie am kritischen Punkt eine Polstelle aufweisen), kann diese Unsicherheit über den genauen Wert β_c u.U. einen deutlichen Einfluß auf die resultierenden Schätzungen für das Verhalten von solchen Observablen am kritischen Punkt, etwa in einer FSS–Analyse, haben. Dies gilt auch für den Fall der hier betrachteten Korrelationslängen: da die Annäherung an den kritischen Punkt und die an den thermodynamischen Limes im Sinne der Renormierungstheorie komplementäre Prozesse darstellen, wirkt die Durchführung einer FSS–Analyse für eine gegenüber dem kritischen Punkt des Systems leicht verschobene Temperatur wie eine Verschiebung der verwendeten Systemgrößen L_x zu effektiven Größen L'_x, was zu entsprechenden Fehlern in den Schätzungen für die Amplituden $A_{\sigma/\epsilon}$ des FSS der Korrelationslängen führt.

Die Berücksichtigung eines solchen Effektes ist mit Hilfe der sog. *Reweighting*–Technik möglich [33]. Sie macht sich die Tatsache zunutze, daß die Temperaturabhängigkeit einer makroskopischen Observable im kanonischen Ensemble gemäß (3.1) vollständig durch die Boltzmann–Verteilung gegeben ist. Für die (inverse) Temperatur β_0 sind die zugehörigen Wahrscheinlichkeiten gegeben durch

$$p_{\beta_0}(E) = \frac{\Omega(E)e^{-\beta_0 E}}{\sum_E \Omega(E)e^{-\beta_0 E}}, \tag{6.2}$$

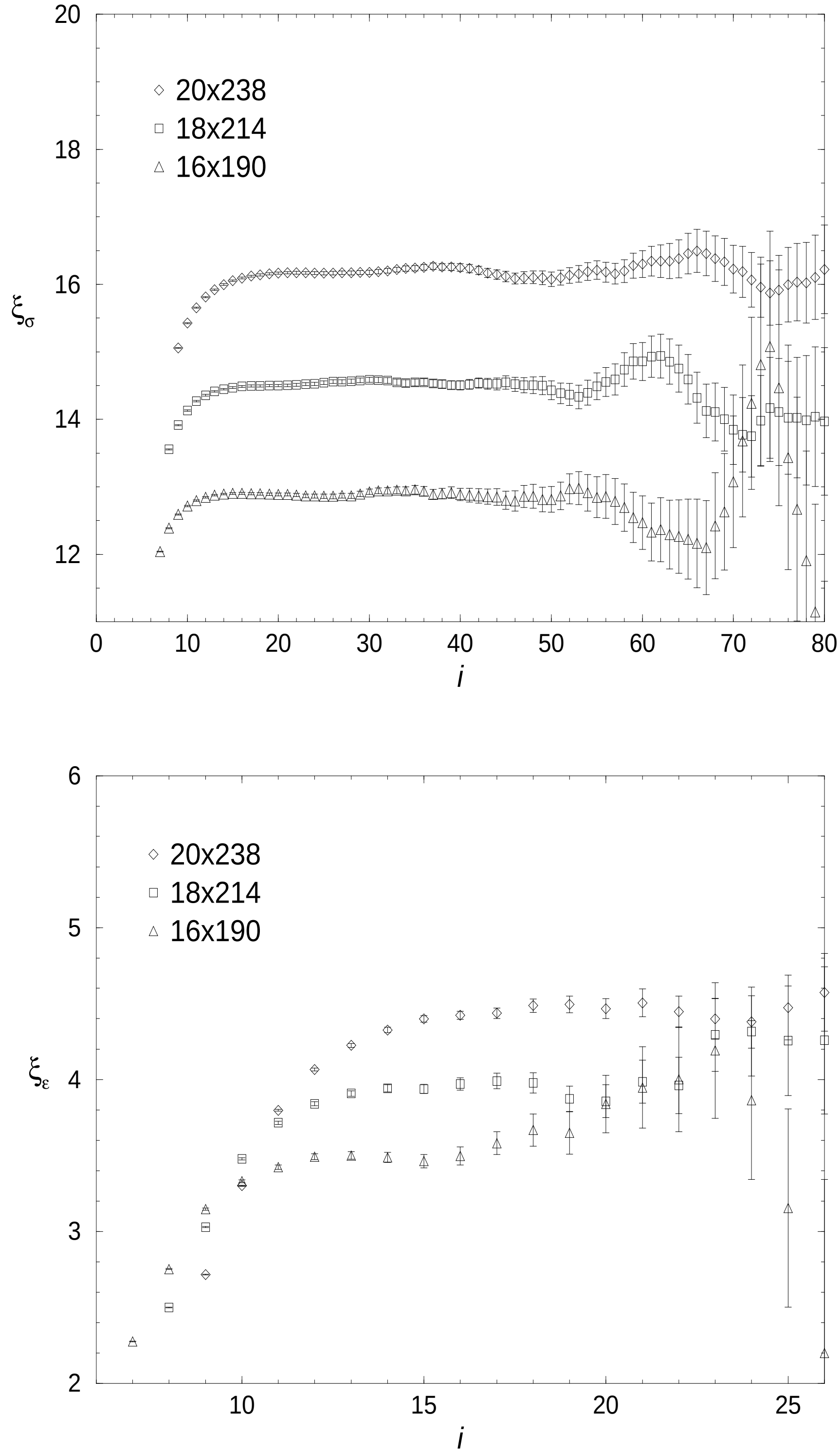


Abb. 6.1: Beispiele für den Verlauf der Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen von Spin (oben) und Energiedichte (unten) für dreidimensionale Ising–Systeme mit periodischen Randbedingungen und den angegebenen Abmessungen $L_x^2 \times L_z$. Gemäß der Regel $\Delta \approx 2\xi_\epsilon$ wurden bei der Anwendung der Differenzenmethode variable Abstände verwendet: $\Delta = 7$ für $L_x = 16$, $\Delta = 8$ für $L_x = 18$ und $\Delta = 9$ für $L_x = 20$.

wobei $\Omega(E)$ die Anzahl der Konfigurationen $\{\mathbf{s}_i\}$ des Systems mit der Energie E (die Zustandsdichte) bezeichnet. Durch eine Umgewichtung (*Reweighting*) läßt sich daraus die Verteilung für jede andere Temperatur β gewinnen, da die Temperaturabhängigkeit in (6.2) explizit ist:

$$p_\beta(E) = c\, p_{\beta_0}(E)\, e^{-(\beta-\beta_0)E}, \tag{6.3}$$

wobei sich die Konstante c durch Normierung der umgewichteten Verteilung ergibt. Den Erwartungswert einer Observable $A(E)$ für die Temperatur β kann man daher schreiben als

$$\langle A\rangle_\beta = c\sum_E A(E) p_{\beta_0}(E) e^{-(\beta-\beta_0)E}. \tag{6.4}$$

Kennt man daher die Verteilung $p_{\beta_0}(E)$, so lassen sich die Erwartungswerte $\langle A\rangle_\beta$ für alle Temperaturen bestimmen. Diese Verteilung wird jedoch gerade durch eine Monte–Carlo–Simulation geliefert: beim Importance Sampling werden die Zustände mit den korrekten Boltzmann–Gewichten erzeugt, so daß das Histogramm der gemessenen Zeitreihe der Energien, $\{E_t\}$, eine Schätzfunktion für die Verteilung $p_{\beta_0}(E)$ darstellt, wenn die Simulation für $\beta = \beta_0$ durchgeführt wurde. Die Eignung des umgewichteten Histogramms zur Approximation der Verteilung bei der Temperatur β hängt neben der Statistik des ursprünglichen Histogramms vom Abstand $|\beta - \beta_0|$ ab: ist dieser zu groß, so erhält man die umgewichtete Verteilung vornehmlich aus den Randbereichen kleiner Häufigkeiten des ursprünglichen Histogramms; dort sind aber aufgrund weniger Ereignisse die statistischen Fehler groß.

Die hier interessierenden Korrelationsfunktionen $G(i)$ hängen jedoch nicht nur von der Energie der Konfiguration ab, zu einer Energie gehören mehrere Werte von $G(i)$ (für ein festes i); in diesem Fall muß man entsprechend die zweidimensionale Dichte $p_{\beta_0}(E, G(i))$ behandeln. Da es jedoch vollkommen unpraktikabel ist, dieses zweidimensionale Histogramm (dazu noch für alle Werte von i) während der Simulation aufzunehmen und zu speichern, kann man die Mittelung über die Werte von $G(i)$ bei fester Energie vorziehen, also schon während des Laufs durchführen, und behält die „mikrokanonischen Mittelwerte“

$$\langle\langle G(i)\rangle\rangle_{\beta_0}(E) = c\sum_{G(i)} \Omega(E, G(i)) G(i) e^{-\beta_0 E}. \tag{6.5}$$

Diese kann man darauf direkt als Spezialfall der Observable $A(E)$ in Gl. (6.4) umwichten. Angesichts einer maximalen Länge L_z der Systeme von etwa 350 Einheiten und den etwa $3V = 3L_X L_y L_z$ möglichen Werten für die Energie (hier maximal ca.

System	Größe	β_0	$\beta_0 - \Delta\beta$	$\beta_0 + \Delta\beta$	Änderung
10×120 periodisch	$\bar{\xi}_\sigma$	8.02147165	8.02114488	8.02179846	$1 \cdot 10^{-5} =$
	$\sqrt{\widehat{\mathrm{VAR}}}$	0.00510784	0.00510758	0.00510810	$6\% \cdot \sigma$
	$\bar{\xi}_\epsilon$	2.16436970	2.16434286	2.16439655	$1 \cdot 10^{-5} =$
	$\sqrt{\widehat{\mathrm{VAR}}}$	0.00656571	0.00656588	0.00656555	$6\% \cdot \sigma$
10×50 antiperiodisch	$\bar{\xi}_\sigma$	2.42878966	2.42877132	2.42880799	$2 \cdot 10^{-5} =$
	$\sqrt{\widehat{\mathrm{VAR}}}$	0.00846211	0.00846213	0.00846207	$0.2\% \cdot \sigma$
	$\bar{\xi}_\epsilon$	0.87591228	0.87590907	0.87591549	$4 \cdot 10^{-6} =$
	$\sqrt{\widehat{\mathrm{VAR}}}$	0.01311917	0.01311928	0.01311906	$0.02\% \cdot \sigma$

Tab. 6.1: Einfluß der Unsicherheit in der kritischen Temperatur auf die Ergebnisse für die Korrelationslängen von zwei beispielhaften Systemen mit unterschiedlichen Randbedingungen. Es ist $\beta_0 = 0.2216544$ und $\Delta\beta = 0.0000003$. Die Testläufe bestanden aus ca. $4 \cdot 10^6$ Messungen. Die relativen Angaben der letzten Spalte sind bezogen auf die Werte für β_0; die Abweichungen sind für $\pm\Delta\beta$ mit großer Genauigkeit symmetrisch.

10^6) ist jedoch auch diese Vorgehensweise hinsichtlich des temporären und permanenten Speicherplatzbedarfes nur schwer praktikabel. Um den Speicherplatzaufwand für die Histogramme kommt man jedoch herum, wenn man Aussagen lediglich für einige wenige, vorher bekannte Temperaturen $\beta \neq \beta_0$ benötigt: dann läßt sich die Umwichtung bei jedem Schritt der Messung während des Laufes selbst vornehmen und es müssen nur die zusätzlichen Werte für die umgewichteten Schätzungen von $G(i)$ selbst gesichert werden. Da wir hier lediglich ein Gefühl dafür bekommen wollen, in welcher Größenordnung sich eine gegebene Unsicherheit in β_c auf die Endergebnisse für die Korrelationslängen auswirkt, reichen aber solche Informationen über einzelnen Temperaturen vollkommen aus; daher werden wir das Reweighting der Korrelationsfunktionen jeweils während des Laufes für vorgegebene Temperaturen vornehmen.

Als Simulationstemperatur wurde der genaueste derzeit verfügbare Schätzwert für die kritische Temperatur des dreidimensionalen Ising–Modells verwendet, der aus einer Monte–Carlo–Simulation stammt [83]:

$$\beta_c = 0.2216544(3). \tag{6.6}$$

Der durch den statistischen Fehler $\Delta\beta = 0.0000003$ dieses Wertes markierte Bereich $[\beta_0 - \Delta\beta, \beta_0 + \Delta\beta]$ wurde als Gebiet betrachtet, in dem mit hoher Wahrscheinlichkeit die wahre kritische Temperatur β_c liegt. Aufgrund der hohen Genauigkeit des angegebenen Wertes ist ein großer Einfluß der Unsicherheit in der kritischen Temperatur auf die Endergebnisse für die Korrelationslängen nicht zu erwarten. Umgewichtete Zeitreihen wurden daher lediglich für die Grenztemperaturen $\beta_0 - \Delta\beta$ und $\beta_0 + \Delta\beta$ erstellt. Da diese sehr nahe an der Simulationstemperatur liegen, stellen die umgewichteten Histogramme gute Schätzer für die entsprechenden Verteilungen dar. In Tab. 6.1 sind die endgültigen Schätzungen für die Korrelationslängen aus zwei Simulation mit periodischen und antiperiodischen Randbedingungen für die o.g. Temperaturen angeführt. Tatsächlich beträgt der Unterschied in den Ergebnissen noch nicht einmal ein Zehntel des statistischen Fehlers der Schätzungen. Damit kann man den Einfluß der Unsicherheit über den genauen Wert der kritischen Temperatur auf die Meßwerte für die Korrelationslängen im Falle des dreidimensionalen Ising–Modells ohne Bedenken vernachlässigen. Für die O(n)–Spin–Modelle mit $n > 1$ sind jedoch die verfügbaren Werte für β_c ungenauer, so daß dort erneut über die Notwendigkeit einer Reweighting–Analyse entschieden werden muß.

6.3 Parameter der Simulation

Für die gemäß der Regel $L_z/\xi_\sigma \approx 15$ dimensionierten Systeme wurden die zur Äquilibrierung notwendigen Update–Schritte durch Beobachtung des Verhaltens der Korrelationsfunktionen $G(i)$ ausgehend von einer zufällig–gleichverteilten Startkonfiguration bestimmt. Hierbei ist offenbar die Korrelationsfunktion der Energiedichte die maßgebliche Größe; für die Produktionsläufe wurden die Systeme daraufhin sicherheitshalber deutlich länger äquilibriert, bevor mit den Messungen begonnen wurde, vgl. die Tabellen 6.2 und 6.3.

Die Autokorrelationszeiten wurden wie im zweidimensionalen Fall mit Hilfe einer Binning–Analyse, d.h. unter Ausnutzung von Gl. (3.24), ermittelt. Im Gegensatz zur gängigen Methode wurden hier jedoch nicht die Autokorrelationszeiten der globalen Magnetisierung bzw. Gesamtenergie bestimmt, die oft gleichsam als pauschale Anhaltspunkte für die Autokorrelationen des Systems überhaupt dienen, sondern direkt die der Korrelationsfunktionen $G(i)$ von lokaler Magnetisierung bzw. Energiedichte, da diese ja auch die eigentlichen Observablen darstellen. Dabei wurde deren Abhängigkeit vom Abstand i vernachlässigt, indem ein fester Abstand am Beginn des Plateaus der Schätzungen $\hat{\xi}(i)$ betrachtet wurde. Eine solche Vereinfachung ist hier wohl zulässig, da nicht die Autokorrelationszeiten als solche — etwa

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	periodische Randbedingungen							
4 × 48	300	300	60000	3	14	60	8192	320
5 × 60	400	400	60000	3	15	60	8192	320
6 × 72	400	400	60000	5	17	60	8192	320
7 × 84	500	500	60000	5	17	60	8192	320
8 × 96	600	700	60000	7	25	60	8192	320
9 × 108	800	800	60000	12	35	60	8192	320
10 × 120	900	1000	60000	16	40	60	8192	320
11 × 132	1100	1200	60000	18	40	60	8192	320
12 × 144	1400	1600	60000	20	42	60	8192	320
13 × 156	1500	1800	60000	28	50	60	8192	320
14 × 168	2000	2000	60000	25	68	60	8192	320
15 × 178	2200	2400	60000	25	70	60	8192	320
16 × 190	2500	3000	60000	40	75	60	8192	320
17 × 202	3200	3500	60000	45	80	60	8192	320
18 × 214	3600	4500	60000	45	80	60	8192	320
19 × 226	4500	5000	70000	50	82	70	8192	992
20 × 238	5500	6000	80000	55	86	80	8192	315
25 × 296	10000	12000	80000	90	180	80	8192	876
30 × 356	16000	18000	120000	110	220	100	8192	432

Tab. 6.2: Simulationsparameter für die Produktionsläufe der 3D Ising–Systeme mit periodischen Randbedingungen. Die Angaben für $t_{\text{Äquil}}$ und τ_{int} sind in Einheiten von Wolff–Updates zu verstehen und beziehen sich jeweils auf das Verhalten der Korrelationsfunktion $G(i)$ selbst, nicht auf die (globale) Magnetisierung bzw. die Gesamtenergie. Für i wurde ein „typischer“ Abstand zu Beginn des Plateaus gewählt; die (schwache) Variation (insbesondere) der Autokorrelationszeiten über den Abstand i wurde vernachlässigt.

zur Bestimmung des dynamischen kritischen Exponenten — für uns von Interesse sind, sondern lediglich ein Anhaltspunkt für eine sinnvolle Frequenz der Messungen gewonnen werden soll.

Die Zusammenfassung vieler Messungen in den Bins der Produktionsläufe sorgt dafür, daß die trotz einer „vernünftigen“ Wahl der Häufigkeit der Messungen verbleibenden Autokorrelationen eliminiert werden. Die Meßfrequenz wurde in der Größen-

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	antiperiodische Randbedingungen							
4 × 50	100	200	40000	6	20	80	8192	1024
5 × 50	300	300	40000	9	25	80	8192	1024
6 × 50	250	400	40000	10	25	80	8192	1024
7 × 50	300	400	40000	10	30	80	8192	1024
8 × 50	300	500	40000	10	25	80	8192	1024
9 × 50	350	500	40000	10	30	80	8192	1024
10 × 50	400	600	40000	10	30	80	8192	1024
11 × 50	500	600	40000	10	30	80	8192	1024
12 × 50	500	600	40000	10	35	80	8192	1024
13 × 50	600	700	40000	11	35	80	8192	1088
14 × 54	800	900	40000	10	35	80	8192	1000
15 × 56	800	1000	40000	10	32	80	8192	1000
16 × 60	900	1200	40000	10	40	80	8192	1000
17 × 64	1000	1400	40000	10	45	80	8192	1024
18 × 68	1200	1800	45000	12	45	90	8192	1080
19 × 72	1300	1800	45000	15	50	90	8192	670
20 × 76	1500	1900	45000	16	55	90	8192	910
25 × 96	3000	3500	60000	20	60	120	8192	1286
30 × 114	6000	6500	70000	25	85	140	8192	516

Tab. 6.3: Simulationsparameter für die Produktionsläufe der 3D Ising–Systeme mit antiperiodischen Randbedingungen.

ordnung der inversen Autokorrelationszeit angenommen, für die kleinen Systeme noch etwas kleiner. Für die Systeme mit antiperiodischen Randbedingungen, für die man sich hier ja die interessanteren Ergebnisse erwartet, wurde dabei noch ein gewisser „Sicherheitsabstand“ aufgeschlagen.

6.4 Ergebnisse

Aus den Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen von Spin und Energiedichte wurden unter Beobachtung des Verhaltens der Größe $\chi^2/g(i_{\min}, i_{\max})$ die Mittel-

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle\lvert C\rvert\rangle}{V}$
	periodische Randbedingungen							
4 × 48	3.0537	0.0054	7 – 21	0.8281	0.0038	4 – 11	2	4.49%
5 × 60	3.9098	0.0034	6 – 26	1.0430	0.0028	4 – 10	2	3.68%
6 × 72	4.7375	0.0035	7 – 27	1.2600	0.0032	5 – 11	3	3.11%
7 × 84	5.5473	0.0039	8 – 36	1.4852	0.0057	6 – 12	3	2.68%
8 × 96	6.3751	0.0072	11 – 38	1.7255	0.0082	7 – 14	3	2.35%
9 × 108	7.1897	0.0058	11 – 46	1.9410	0.0088	8 – 16	4	2.09%
10 × 120	8.0094	0.0064	12 – 55	2.1587	0.0074	8 – 16	4	1.89%
11 × 132	8.8276	0.0056	12 – 53	2.4176	0.0198	11 – 24	5	1.72%
12 × 144	9.6460	0.0066	13 – 55	2.6283	0.0186	11 – 22	5	1.57%
13 × 156	10.4665	0.0066	14 – 51	2.8603	0.0174	12 – 25	6	1.45%
14 × 168	11.2603	0.0091	13 – 46	3.0222	0.0195	11 – 19	6	1.34%
15 × 178	12.0812	0.0083	15 – 70	3.2835	0.0186	12 – 25	6	1.27%
16 × 190	12.8984	0.0069	15 – 79	3.4979	0.0150	12 – 25	7	1.19%
17 × 202	13.7285	0.0090	17 – 77	3.7569	0.0224	14 – 31	7	1.12%
18 × 214	14.5151	0.0093	18 – 85	3.9543	0.0229	15 – 31	8	1.05%
19 × 226	15.3479	0.0065	21 – 84	4.1905	0.0145	16 – 37	8	0.99%
20 × 238	16.1730	0.0097	21 – 86	4.4256	0.0200	16 – 35	9	0.94%
25 × 296	20.2645	0.0097	30 –100	5.5358	0.0226	21 – 45	11	0.75%
30 × 356	24.3313	0.0154	35 –130	6.6438	0.0506	26 – 54	13	0.63%

Tab. 6.4: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des dreidimensionalen Ising–Modells mit periodischen Randbedingungen. Man beachte den gegenüber dem 2D Ising–Modell gesunkenen durchschnittlichen Anteil $\langle|C|\rangle/V$ von Spins des Gitters, die bei einem Wolff–Update–Schritt ihre Orientierung umkehren.

werte $\bar{\xi}$ als endgültige Schätzungen für die gesuchten Korrelationslängen gewonnen. Somit erhält man wiederum Wertetabellen von Schätzungen für ξ in Abhängigkeit von der transversalen Systemgröße $L_x = L_y$ wie sie in den Tabellen 6.4 und 6.5 für die beiden Arten von Randbedingungen zu sehen sind. Der nur schwach von Fluktuationen überlagerte, kontinuierliche Anstieg der Bereichsgröße $i_{\max} - i_{\min} + 1$ mit der Systemgröße belegt hierbei die Zuverlässigkeit der angegebenen Methode der Plateau–Bestimmung. Die für antiperiodische Randbedingungen deutlich kleineren durchschnittliche Größen $\langle|C|\rangle$ der stochastischen Cluster des Wolff–Algorithmus

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle\lvert C\rvert\rangle}{V}$
	antiperiodische Randbedingungen							
4×50	1.0326	0.0078	5 – 11	0.3715	0.0011	2 – 3	1	2.20%
5×50	1.2575	0.0021	4 – 12	0.4610	0.0065	3 – 5	1	2.10%
6×50	1.4869	0.0036	5 – 11	0.5410	0.0040	3 – 6	1	2.03%
7×50	1.7240	0.0052	6 – 16	0.6169	0.0031	3 – 5	1	1.98%
8×50	1.9527	0.0044	6 – 13	0.7091	0.0104	4 – 7	1	1.93%
9×50	2.1813	0.0041	7 – 20	0.7818	0.0043	4 – 7	2	1.89%
10×50	2.4253	0.0085	9 – 22	0.8810	0.0109	5 – 8	2	1.86%
11×50	2.6648	0.0074	9 – 22	0.9526	0.0094	5 – 8	2	1.84%
12×50	2.8824	0.0070	9 – 22	1.0362	0.0075	5 – 10	2	1.82%
13×50	3.1390	0.0086	10 – 20	1.1462	0.0156	6 – 11	2	1.79%
14×54	3.3682	0.0105	11 – 23	1.2146	0.0131	6 – 12	2	1.64%
15×56	3.6124	0.0097	11 – 20	1.2879	0.0119	6 – 11	2	1.57%
16×60	3.8431	0.0125	12 – 27	1.4441	0.0285	7 – 10	2	1.46%
17×64	4.0732	0.0126	13 – 29	1.4608	0.0171	7 – 15	2	1.35%
18×68	4.3143	0.0089	13 – 27	1.5789	0.0116	7 – 11	3	1.27%
19×72	4.5396	0.0106	13 – 28	1.6397	0.0251	8 – 13	3	1.19%
20×76	4.7676	0.0110	14 – 27	1.7260	0.0188	8 – 15	3	1.12%
25×96	5.9639	0.0092	17 – 37	2.1638	0.0193	10 – 18	4	0.87%
30×114	7.1335	0.0157	20 – 47	2.5853	0.0374	12 – 23	5	0.72%

Tab. 6.5: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des dreidimensionalen Ising–Modells mit antiperiodischen Randbedingungen.

bestätigen die in diesem Zusammenhang für die zweidimensionalen Systeme gemachten Beobachtungen des Einflusses der Randbedingungen auf typische Konfigurationen des Systems. Dies gilt allerdings nur für Systeme mit gleichem Aspektverhältnis: so ist der Anteil der durchschnittlich pro Cluster–Update erfaßten Spins für die kleinsten Systeme, die für beide Arten von Randbedingungen (etwa) gleich dimensioniert wurden, für antiperiodische Randbedingungen nur etwa halb so groß wie für periodische Randbedingungen. Mit Beginn der Skalierung der Längen L_z mit der Spin–Korrelationslänge ξ_σ auch für antiperiodische Randbedingungen werden die Anteile für beide Randbedingungen zunehmend vergleichbar. Wie anschaulich klar ist, verkleinert eine große Anisotropie der Systeme die relative Größe typischer Cluster. Das Aspektverhältnis L_x/L_z ist jedoch für die (größeren) Systeme mit periodischen Randbedingungen kleiner als für die Systeme mit antiperiodischen Randbedingungen.

Die Abbildungen 6.2 und 6.3 zeigen die FSS–Graphen für die beiden Arten von Randbedingungen. Der Verlauf $\xi(L_x)$ sieht wiederum in recht großer Genauigkeit linear aus; die Auftragung der Amplituden ξ/L_x in den Abbildungen 6.4 und 6.5 zeigt jedoch, daß auch hier Korrekturen zum FSS mit der vorhandenen Simulationsgenauigkeit aufgelöst werden können. Im zweidimensionalen Fall hatten wir den Bias der Schätzer $\hat{\xi}(i)$ bzw. daraus folgend der Mittelwerte $\bar{\xi}(i)$ aufgrund der Ergebnisse einer Jackknife–Bias–Analyse als vernachlässigbar eingeschätzt (vgl. Kap. 4.4.2). Um auch für die jetzt zu untersuchenden Simulationsergebnisse diesbezüglich sicher zu gehen, wurde auch hier eine Bias–Analyse mit Hilfe von *Double Jackknife Bias Corrected Estimators* gemäß Kap. 4.4.6 durchgeführt. Da der mögliche Einfluß eines solchen Bias generell als gering einzuschätzen ist, wurde dabei die Bestimmung des Plateau–Bereichs durch Beobachtung der Funktion $\chi^2/g(i_{\min}, i_{\max})$ zunächst weiterhin unter Verwendung der unkorrigierten Schätzer $\hat{\xi}(i)$ durchgeführt, um abzuschätzen, ob eine Bias–Korrektur notwendig ist. Dies hat auch einen Grund im großen Rechenaufwand, der mit eine Double–Jackknife–Analyse verbunden ist: gegenüber der einfachen Jackknife–Analyse multipliziert sich dieser nochmals mit der Anzahl der Bins. Abb. 6.4 zeigt den Vergleich der so gewonnenen Schätzungen mit denen aus der einfachen Jackknife–Analyse für die Energie–Korrelationslänge; für ξ_σ sind die Korrekturen so klein, daß sie selbst in der Auftragung der Amplituden ξ/L_x sind erkennbar sind. Da im Falle der Systeme mit antiperiodischen Randbedingungen deutlich längere Zeitreihen und daher (bei konstanter Größe der Blöcke) auch mehr Bins vorliegen und der Bias mit der inversen Anzahl der Blöcke abfällt, sind in Abb. 6.4 die stärksten auftretenden Abweichungen gezeigt. Insgesamt ist der Effekt der Bias–Korrektur offenbar minimal und im Vergleich zu den statistischen Fehlern zu vernachlässigen. Auch die etwas stärkeren Abweichungen bei zwei oder drei Punkten ändern nichts an diesem Bild: eine genauere Analyse zeigt, daß hier die Bias–Korrektur am Ende des verwendeten Bereichs $i_{\min}, \ldots, i_{\max}$, der ja ohne Verwendung der Bias–Korrektur bestimmt wurde, also dort, wo das Rauschen schon vollkommen in der Größenordnung des Signals liegt, nicht mehr sinnvoll angewendet werden kann, da die statistischen Fehler zu groß sind. Dies schlägt sich auch in entsprechend vergrößerten Werten für χ^2/g für diese Punkte nieder; m.a.W.: würde man auch die Analyse der Funktion $\chi^2/g(i_{\min}, i_{\max})$ mit Hilfe der Double–Jackknife–

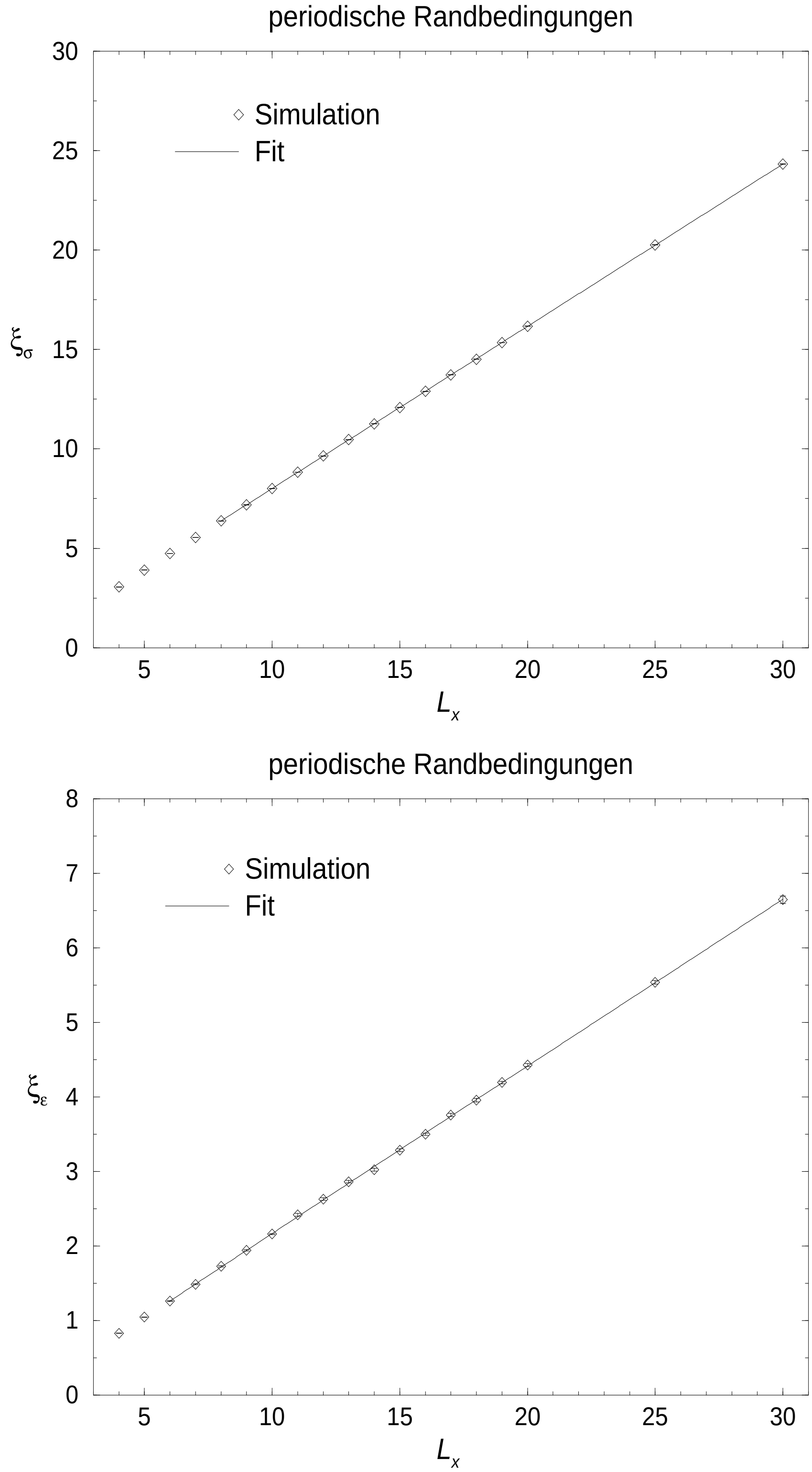
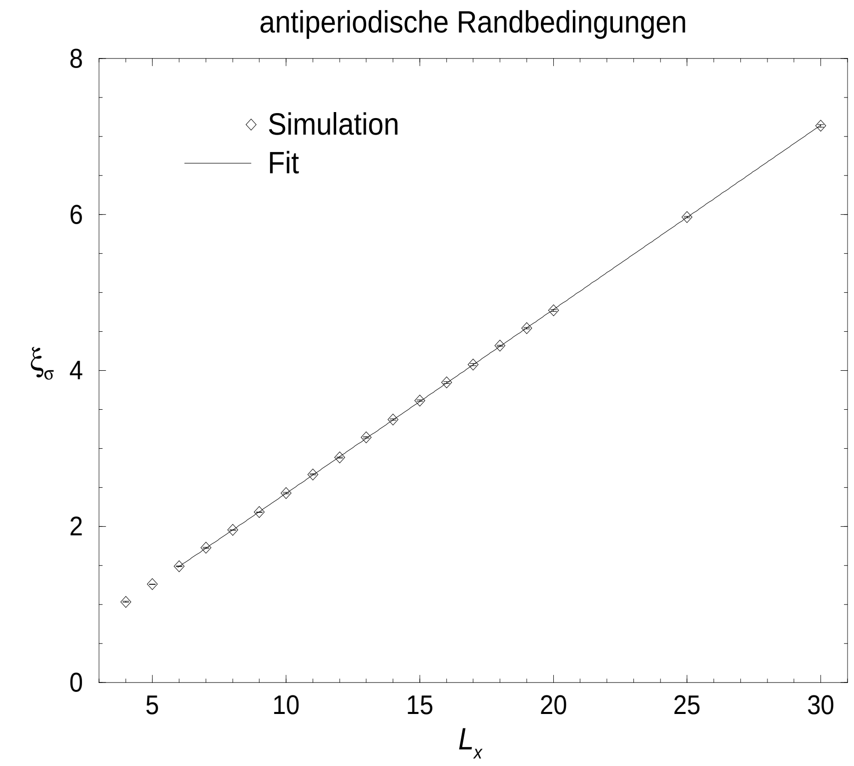


Abb. 6.2: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 3D Ising–Modells mit periodischen Randbedingungen für Systeme mit $L_x = 4$ bis $L_x = 30$. Die einzelnen Punkte entsprechen den Mittelwerten $\bar{\xi}$ über den Plateau–Bereich; die Fehler sind die entsprechenden Endergebnisse der Jackknife–Analyse.



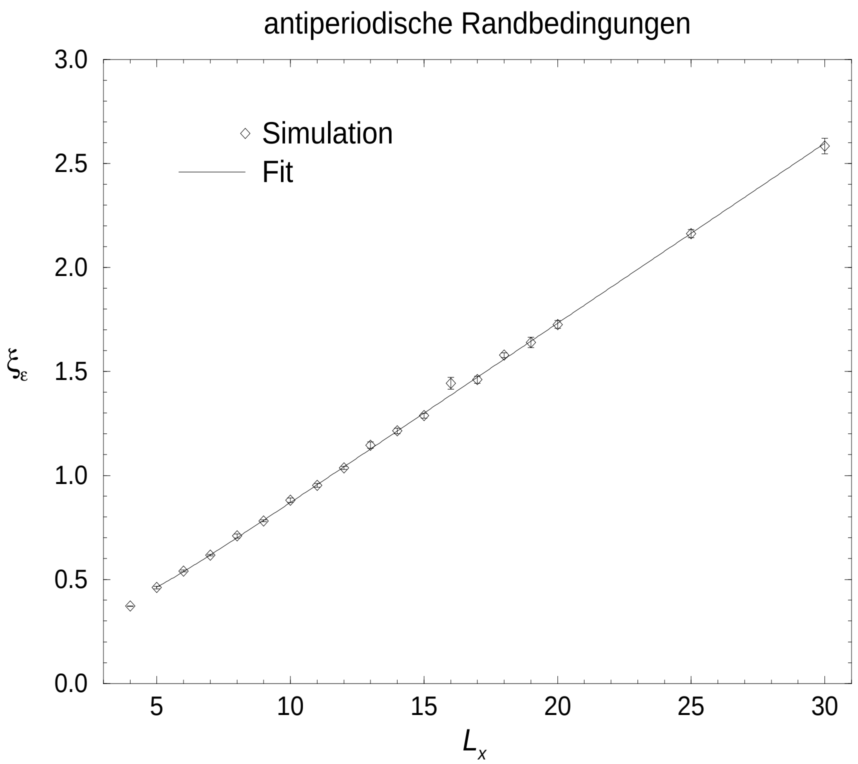
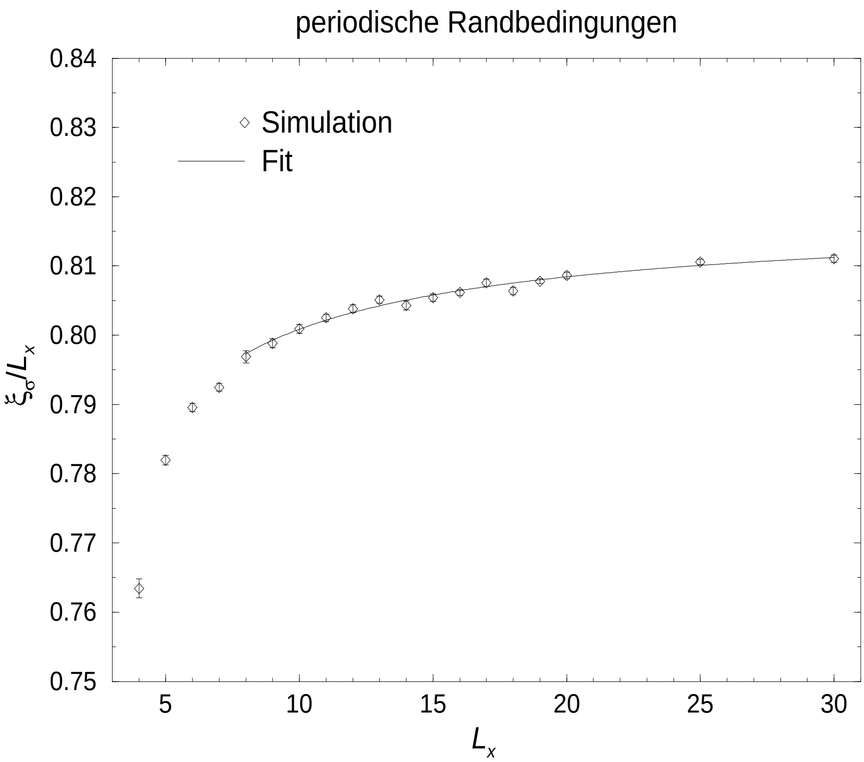


Abb. 6.3: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 3D Ising–Modells mit antiperiodischen Randbedingungen für Systeme mit $L_x = 4$ bis $L_x = 30$.



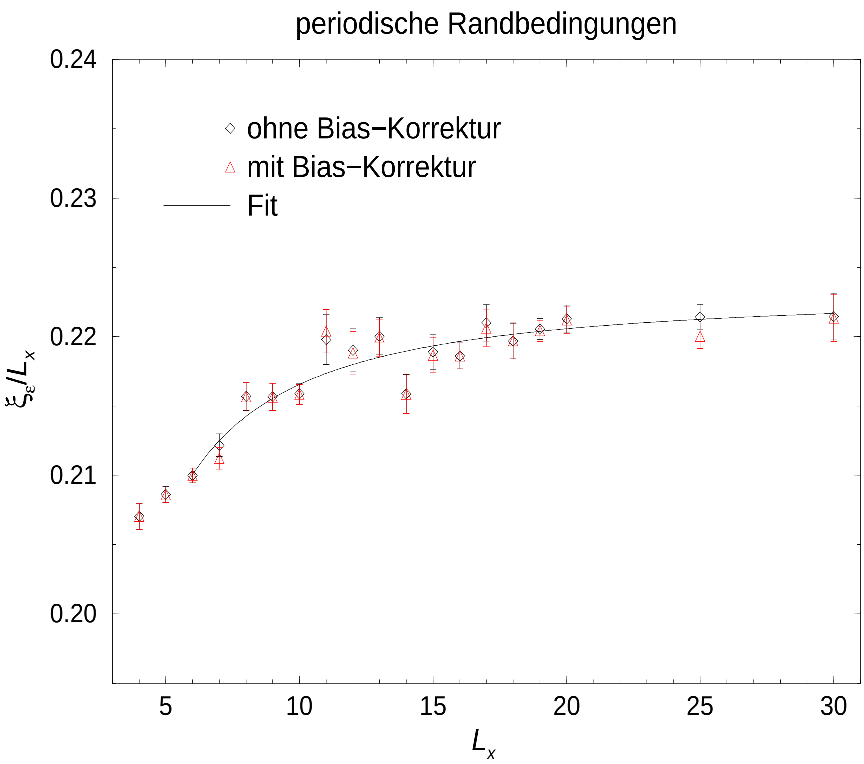
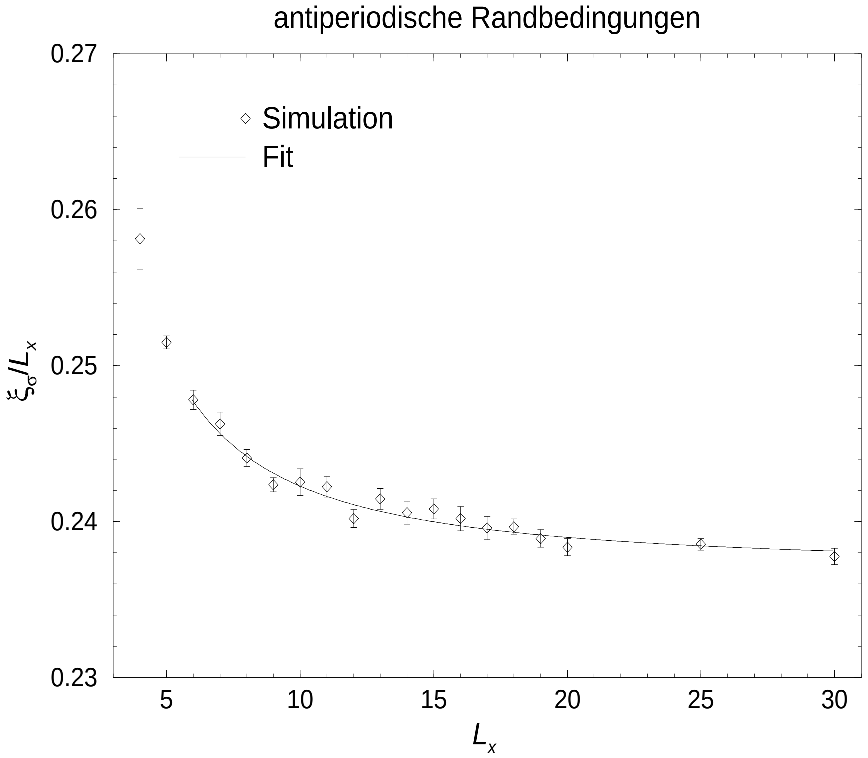


Abb. 6.4: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für periodische Randbedingungen. Die roten Symbole zeigen zusätzlich die Ergebnisse der Fehler– und Bias–Analyse mit Hilfe von *Double Jackknife Bias Corrected Estimators*. Die schwarzen Kurven zeigen Fits an die Funktion $\xi = AL_x + BL_x^\alpha$,



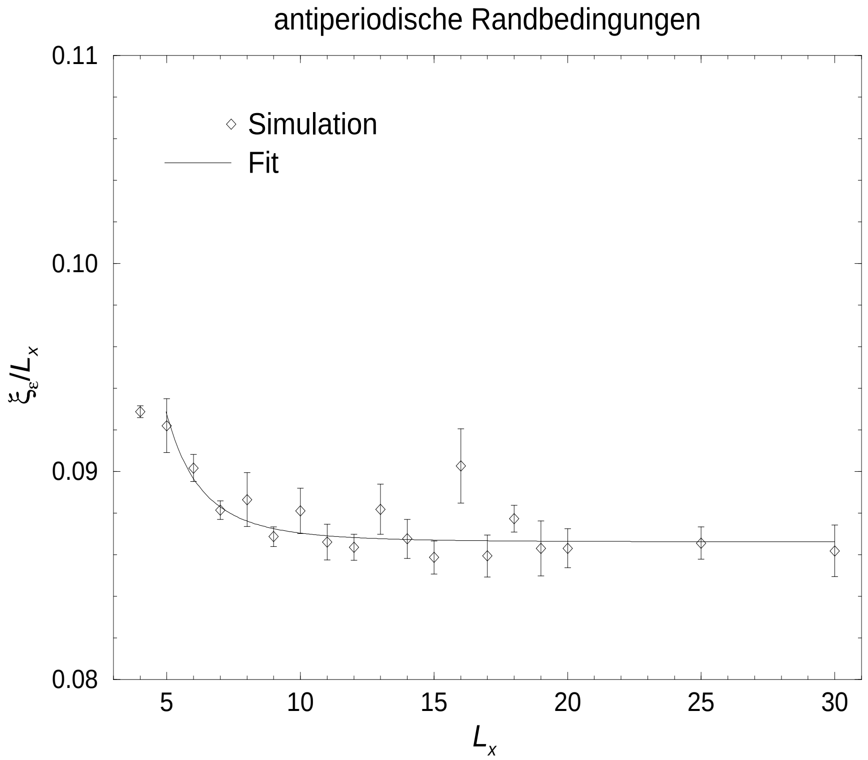


Abb. 6.5: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für antiperiodische Randbedingungen.

Analyse durchführen, würde man die entsprechenden Punkte $\hat{\xi}(i)$ gar nicht mehr dem Plateau zurechnen, so daß der Effekt der Bias–Korrektur auch in diesem Fall auf das bei den übrigen Punkten zu beobachtende Maß zurückginge.

Zur Berücksichtigung der Korrekturen zum linearen Hauptterm des FSS der Korrelationslängen wurden die Ergebnisse für $\xi(L_x)$ wiederum per Regression an ein Gesetz der Form

$$\xi(L_x) = AL_x + BL_x^\alpha \tag{6.7}$$

angepaßt. Dem systematischen Fehler dieses Ansatzes durch Vernachlässigung höherer Ordnung in den Korrekturen wurde dabei durch sukzessives Weglassen von Werten ausgehend vom kleinsten L_x und Beobachtung des Verhaltens der Fit–Parameter Rechnung getragen. Für die Systeme mit periodischen Randbedingungen ergibt sich auf diesem Wege:

$$\begin{array}{|rrl|} \hline A_\sigma = & 0.8183(32) & (L_x = 8-30) \\ B_\sigma = & -0.116(35) & \\ \alpha_\sigma = & 0.18(21) & \\ \chi^2/g = & 1.31 & \\ Q = & 0.20 & \\ \hline A_\epsilon = & 0.2232(16) & (L_x = 6-30) \\ B_\epsilon = & -0.144(79) & \\ \alpha_\epsilon = & -0.33(36) & \\ \chi^2/g = & 1.02 & \\ Q = & 0.43 & \\ \hline \end{array} \tag{6.8}$$

Für antiperiodische Randbedingungen erhält man:

$$\begin{array}{|rrl|} \hline A_\sigma = & 0.23694(80) & (L_x = 6-30) \\ B_\sigma = & 0.126(54) & \\ \alpha_\sigma = & -0.37(26) & \\ \chi^2/g = & 0.92 & \\ Q = & 0.53 & \\ \hline A_\epsilon = & 0.08661(31) & (L_x = 5-30) \\ B_\epsilon = & 3.0 \pm 5.4 & \\ \alpha_\epsilon = & -2.9 \pm 1.1 & \\ \chi^2/g = & 0.93 & \\ Q = & 0.53 & \\ \hline \end{array} \tag{6.9}$$

	α	0.18	0.29	0.07
period. Randb.	A_σ	0.81832(49)	0.82024(57)	0.81685(44)
	B_σ	-0.1154(45)	-0.0992(39)	-0.1364(54)
	χ^2/g	1.21	1.23	1.23
	Q	0.26	0.25	0.25
	A_ϵ	0.22556(97)	0.2266(11)	0.22475(87)
	B_ϵ	-0.0587(80)	-0.0511(70)	-0.0686(93)
	χ^2/g	0.98	0.97	0.99
	Q	0.47	0.47	0.45
antiperiod. Randb.	A_σ	0.23494(47)	0.23412(53)	0.23557(42)
	B_σ	0.0481(40)	0.0416(35)	0.0565(47)
	χ^2/g	0.80	0.80	0.80
	Q	0.67	0.66	0.66
	A_ϵ	0.08619(82)	0.08610(93)	0.08626(74)
	B_ϵ	0.0055(65)	0.0048(57)	0.0065(77)
	χ^2/g	1.02	1.02	1.02
	Q	0.43	0.43	0.43

Tab. 6.6: Parameter von Fits der Form (6.7) an die Schätzungen für die Korrelationslängen $\bar{\xi}(L_x)$ mit *festen* Korrekturexponenten α. Die Fits für $\alpha = 0.18$ beinhalten (zufällig) alle den Bereich $L_x = 8 - 30$; die Fits für die anderen Exponenten wurden dann zwecks Vergleichbarkeit der Ergebnisse mit denselben Bereichen durchgeführt.

Die großen Fehler für Amplitude und Exponent der Korrektur für ξ_ϵ im Falle antiperiodischer Randbedingungen bestätigen den aus Abb. 6.5 zu gewinnenden Eindruck, daß hier die Genauigkeit der Daten nicht ausreicht, um den genauen Verlauf der Korrekturen abzuschätzen; lediglich ihr Vorhandensein läßt sich im strengen Sinne hieraus noch ableiten. Das beeinträchtigt jedoch nicht das Ergebnis für die hier interessierende führende Amplitude; der systematische Fehler durch nicht aufgelöste Korrekturen zum FSS ist durch den statistischen Fehler vollkommen abgedeckt.

Die Hauptschwierigkeit in der Fit–Prozedur liegt in der Nichtlinearität der Form (6.7): legt man den Exponenten α fest, erhöht sich sofort die Stabilität der Fits. Wie oben erwähnt, existieren für die Exponenten der ersten Korrektur–Ordnung des FSS auch im Falle der O(n)–Spin–Modelle in drei Dimensionen feldtheoretische Schätzwerte, die man hier probehalber als feste Werte für α einsetzen kann. Dies muß jedoch nicht notwendigerweise genauere Ergebnisse liefern: für endliche L_x sind immer auch Korrekturen höherer Ordnung von Bedeutung, so daß ein freier Fit gemäß (6.7) einen effektiven Exponenten α liefern wird, der nicht mit dem Exponenten der ersten Ordnung der Korrektur übereinstimmt. Da sich jedoch gemäß den Abbn. 6.4 und 6.5 die Schätzwerte für die Amplituden ξ/L_x innerhalb der statistischen Fehler für die größten Systeme nicht mehr verändern, kann man hoffen, daß zumindest für die größeren Systeme das FSS–Verhalten mit hinreichender Genauigkeit neben dem linearen Term durch die führende Ordnung der Korrektur beschrieben wird. Daher wollen wir hier auch die so gewonnenen Werte für die führenden Amplituden angeben. Zinn–Justin [88] nennt als mittleren Wert für den Korrekturexponenten für das Skalenverhalten von Observablen in der Temperaturvariable für das 3D Ising–Modell auf dem Gitter den Wert $\Delta_1 = 0.52(7)$. Für Korrekturen zum FSS ergibt sich daraus mit dem kritischen Exponenten $\nu = 0.63055(28)$ gemäß Tab. 2.1 ein Exponent von $\omega = \Delta_1/\nu = 0.82(11)$. In der Schreibweise (6.7), die die führende Ordnung nicht faktorisiert, ergibt sich daraus $\alpha = 1 - \omega = 0.18(11)$. Der Fehler im Korrekturexponenten wurde durch zusätzliche Fits mit $\alpha \pm \Delta\alpha$ berücksichtigt, vgl. Tab. 6.6. Zwar gehen die Schwankungen der Fit–Parameter durch ein solches Festlegen des Korrekturexponenten zurück, aber durch den großen Fehler im Exponenten lassen sich mit dieser Methode keine genaueren Schätzwerte für die Amplituden A gewinnen als durch den 3–Parameter–Fit.

Für das Verhältnis der Korrelationslängen im FSS–Limes ergibt sich aus den Fits mit freiem Korrekturexponenten schließlich:

$$\boxed{\begin{array}{ll} A_\sigma/A_\epsilon = 3.666(30) & \text{für periodische RB} \\ A_\sigma/A_\epsilon = 2.736(13) & \text{für antiperiodische RB,} \end{array}} \tag{6.10}$$

Vergleicht man mit dem Literaturmittelwert der Skalendimensionen (vgl. Kap. 2.6):

$$x_\epsilon/x_\sigma = 2.7326(16), \tag{6.11}$$

so sieht man die Behauptung einer linearen Beziehung zwischen den Amplituden des FSS der Korrelationslängen und den zugehörigen Skalendimensionen im Fall antiperiodischer Randbedingungen mit hervorragender Genauigkeit bestätigt, während sich die Gültigkeit einer solchen Gesetzmäßigkeit für Systeme mit periodischen Randbedingungen ausschließen läßt. Die Genauigkeit des Ergebnisses für antiperiodische Randbedingungen konnte dabei gegenüber Henkels Transfermatrix–Rechnung um einen Faktor 3, im Vergleich zu dem Wert aus Westons MC–Simulation von 2.6(2) sogar um einen Faktor 15 gesteigert werden.

Kapitel 7

Das 3D XY–Modell

Wenn die lineare Relation zwischen Amplituden und Skalenexponenten des FSS der Korrelationslängen gemäß

$$\xi_i = \frac{A}{x_i} L_x \tag{7.1}$$

eine universelle Beziehung darstellt, muß ihre Gültigkeit vom betrachteten Modell unabhängig sein (sofern es einen kontinuierlichen Phasenübergang aufweist). Es ist daher von Interesse, diesen Zusammenhang für eine Reihe verschiedener Modelle zu untersuchen, um etwas über die Reichweite jener Relation zu erfahren. In dieser Arbeit betrachten wir dazu die in der Reihe der O(n)–Spin–Modelle dem Ising–Fall ($n = 1$) folgenden Modelle, das XY– ($n = 2$) und das Heisenberg–Modell ($n = 3$). In diesem Kapitel betrachten wir zunächst das XY–Modell: nach einer Diskussion der notwendigen Änderungen am verwendeten Instrumentarium präsentieren wir die Ergebnisse einer FSS–Analyse der Korrelationslängen dieses Modells.

7.1 Änderungen gegenüber dem Ising–Modell

Zunächst muß der Update–Algorithmus der Simulation an die vergrößerte Dimension des Ordnungsparameters angepaßt werden. Wir wollen weiterhin einen Cluster–Algorithmus im Wolffschen Sinne verwenden. Dieser ist zunächst jedoch nur für den Update isingartiger Freiheitsgrade geeignet, da nur hier das diskrete Konzept von gleich oder entgegengesetzt ausgerichteten Nachbarspins sinnvoll ist. Gemäß den Ausführungen in Kap. 3.1.2 läßt sich jedoch die Hamilton–Funktion der

Methode	β_c	ν	γ	Quelle
MC	0.45420(2)	0.662(7)	1.308(16)	[40]
MC	0.4542(1)	0.670(2)	1.319(2)	[54]
MC	0.45408(8)	—	—	[54]
MC	0.454165(4)	0.672(1)	1.319(2)	[4]
MC	0.45421(8)	—	1.327(8)	[44]
MC	0.45420(2)	—	—	[39]
MC	0.454170(7)	—	—	[39]
MC	0.454148(15)	—	—	[40]
HT–Reihe	0.45406(5)	—	—	[12]
HT–Reihe	0.45414(7)	—	—	[2]
HT–Reihe	0.45420(6)	0.679(3)	1.328(6)	[13]
HT–Reihe	0.45419(3)	0.677(3)	1.327(4)	[14]
HT–Reihe	—	0.675(2)	1.325(3)	[14]
Gewichtetes Mittel	0.4541670(32)	0.6728(8)	1.3217(14)	

Tab. 7.1: Literaturwerte für die Temperatur des Phasenübergangs und die kritischen Exponenten ν und γ des dreidimensionalen XY–Modells.

Modelle mit kontinuierlichen Freiheitsgraden durch Einführung eines Satzes zufällig–gleichverteilter Richtungen im Spin–Raum in eine Summe von Freiheitsgraden vom Ising–Typ zerlegen, wobei die Koppelungen jetzt zufällig werden. Der Aufbau eines Clusters erfolgt dann durch Wahl einer zufälligen Richtung, durch die eine Ebene definiert wird, die die symbolischen Spinorientierungen voneinander trennt, und Anwendung der Swendsen–Wang–Regel des Ising–Modells. Der Update des Clusters entspricht dann einer Spiegelung der Spins des Clusters an jener Ebene [86].

Die Bestimmung der Korrelationsfunktionen erfolgt durch entsprechende Verallgemeinerung der für den Ising–Fall angegebenen Gleichungen: die $\mathbb{R}$–Multiplikation der Ising–Spins muß lediglich durch Skalarprodukte zwischen Vektorspins ersetzt werden. Sind die Schätzungen $\hat{G}(i)$ für die Korrelationsfunktionen bekannt, kann die weitere Analyse vollkommen analog zu der des Ising–Modells erfolgen, da $G(i)$ wieder skalar ist.

Wie sehen typische Konfigurationen der Systeme aus? In einer vektoriellen Darstellung zeigen sich Cluster–Strukturen ähnlich den Weiss'schen Bezirken eines Ferro-

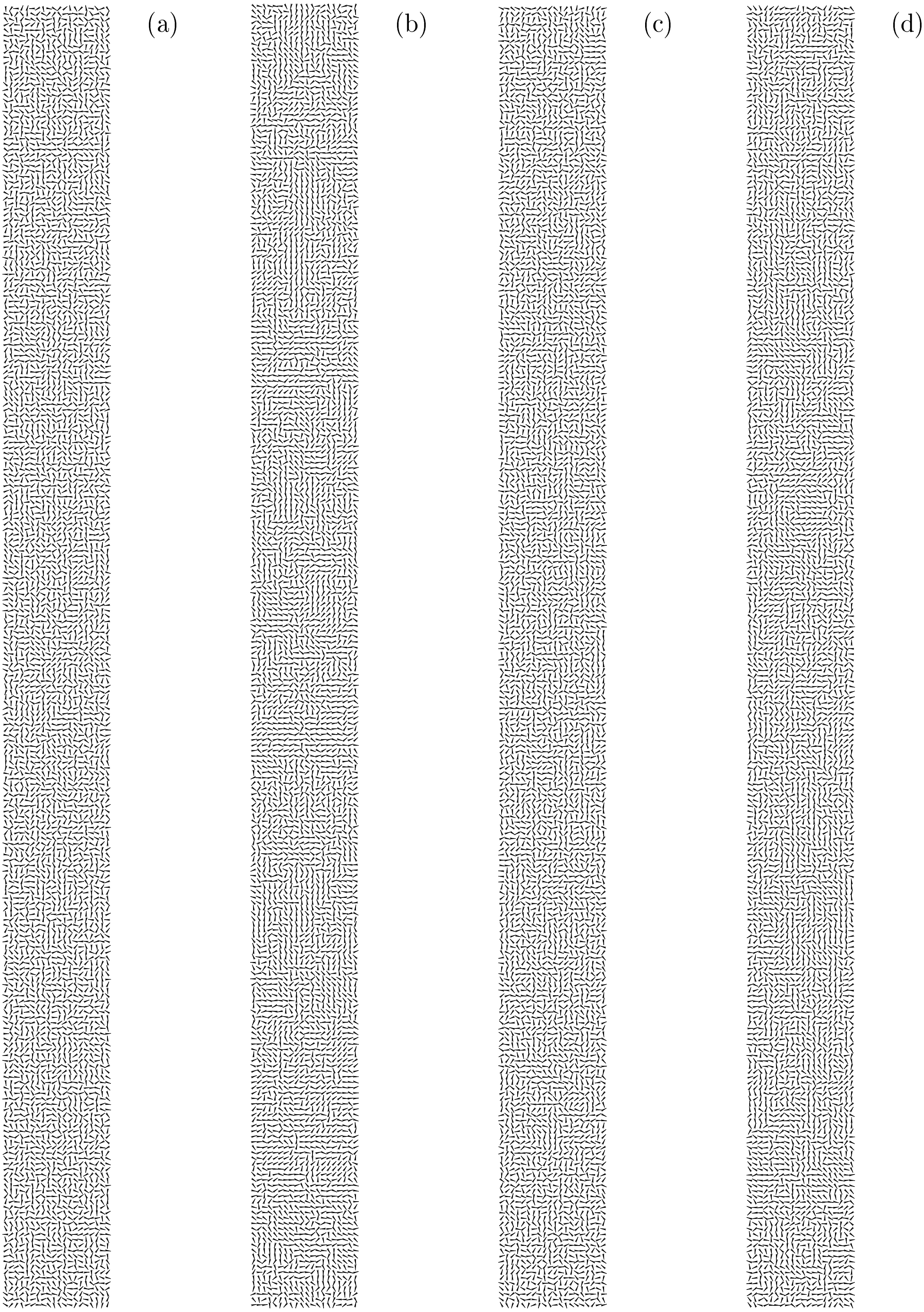


Abb. 7.1: Typische Konfigurationen des XY–Modells für ein System mit $L_x = L_y = 20$, $L_z = 240$. Die Abbildung zeigt Schnitte $\{(x, y, z) \,|\, x = \text{const.}\}$ ((a) und (c)) bzw. Projektionen auf die Schnittflächen ((b) und (d)) für periodische ((a) und (b)) und antiperiodische ((c) und (d)) Randbedingungen. Bei der Darstellung der Projektionen wurden unterschiedliche Längen der resultierenden Vektoren nicht berücksichtigt.

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	periodische Randbedingungen							
4 × 48			42500			85	8192	1024
5 × 60			42500			85	8192	1024
6 × 72	2500	3000	42500	10	85	85	8192	1152
7 × 84			70000			140	8192	1152
8 × 96	3000	4000	70000	19	140	140	8192	1000
9 × 108			82500			165	8192	1040
10 × 120	3500	5000	82500	23	165	165	8192	1024
11 × 132			125000			250	8192	960
12 × 144	4500	6000	125000	24	250	250	8192	1000
13 × 156			130000			260	8192	508
14 × 168	6000	10000	130000	38	260	260	8192	507
15 × 180			145000			290	8192	500
16 × 192	9000	14000	145000	48	290	290	8192	514
17 × 204			215000			430	8192	522
18 × 216	12000	16000	215000	90	430	430	8192	255
19 × 228			240000			480	8192	279
20 × 240	15000	20000	240000	100	480	480	8192	263
25 × 300	32000	35000	280000	170	560	560	8192	130
30 × 360	60000	60000	325000	200	650	650	8192	112

Tab. 7.2: Simulationsparameter für die Produktionsläufe der 3D XY–Systeme mit periodischen Randbedingungen. Für die Einheiten und die Bezugsgrößen der Autokorrelationszeiten gelten die Ausführungen in Kap. 6.

magneten, vgl. Abb. 7.1. Da eine dreidimensionale Darstellung unübersichtlich wäre, wurden zunächst Schnitte $\{(x, y, z)\,|\,x = \text{const.}\}$ durch die Systeme betrachtet. Hier fällt es jedoch schwer, die Cluster–Strukturen zu erkennen, da eben durch den Schnitt eine Dimension des Systems verlorengeht. Daher wurden alternativ *Projektionen* der Spins auf eine solche Schnittfläche erstellt, die die Cluster–Struktur deutlicher hervortreten lassen. Wie auch im Ising–Fall sind aufgrund der starken Anisotropie der Systeme die Cluster wesentlich kleiner als etwa in kubischen Systemen (und daher auch schwerer zu erkennen). Eine optische Unterscheidung der Cluster–Strukturen von Systemen mit antiperiodischen Randbedingungen von solchen mit periodischen

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	antiperiodische Randbedingungen							
4 × 50	1500	2500	31000	21	62	62	8192	2176
5 × 50			37500			75	8192	2176
6 × 50	1500	3000	37500	17	75	75	8192	2176
7 × 50			40000			80	8192	2176
8 × 50	2000	3000	40000	15	80	80	8192	2176
9 × 50			40000			80	8192	2240
10 × 50	2500	3500	40000	13	78	80	8192	2120
11 × 50			45000			90	8192	2046
12 × 50	3000	4000	45000	13	90	90	8192	2010
13 × 50			50000			100	8192	2030
14 × 54	3500	4000	50000	16	100	100	8192	2044
15 × 56			55000			110	8192	2048
16 × 60	3500	4500	55000	18	110	110	8192	1076
17 × 64			60000			120	8192	1000
18 × 68	4000	5500	60000	18	120	120	8192	1318
19 × 72			80000			160	8192	1290
20 × 76	5000	6500	80000	25	160	160	8192	1055
25 × 96	15000	15000	120000	33	240	240	8192	522
30 × 114	18000	20000	135000	43	270	270	8192	503

Tab. 7.3: Simulationsparameter für die Produktionsläufe der 3D XY–Systeme mit antiperiodischen Randbedingungen.

ist nur schwer möglich; relativ gut zu erkennen ist lediglich das Auftreten einer festen Grenzfläche (antiferromagnetische Naht) am entsprechenden Systemrand. Die Ergebnisse für die durchschnittlichen Größen der Cluster des Update–Algorithmus' weiter unten zeigen jedoch auch für das XY–Modell (bei gleicher Systemgröße) für antiperiodische Randbedingungen deutlich kleinere Cluster. Ein Vergleich mit den Ergebnissen für das 3D Ising–Modell im vorangegangenen Kapitel zeigt auch, daß die Cluster des Wolff–Algorithmus (stochastische Cluster) für das XY–Modell bei gleicher Systemgröße für beide Arten von Randbedingungen kleiner sind als für das Ising–Modell.

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle\|C\|\rangle}{V}$
	periodische Randbedingungen							
4×48	2.8902	0.0010	5 – 19	0.7283	0.0038	4 – 7	2	3.31%
5×60	3.6659	0.0024	8 – 21	0.9173	0.0066	5 – 10	2	2.69%
6×72	4.4273	0.0048	12 – 28	1.1247	0.0099	6 – 11	2	2.26%
7×84	5.1886	0.0023	10 – 28	1.3010	0.0083	7 – 16	3	1.94%
8×96	5.9500	0.0024	10 – 30	1.5065	0.0134	8 – 16	3	1.69%
9×108	6.7121	0.0031	12 – 37	1.6972	0.0103	8 – 16	3	1.51%
10×120	7.4692	0.0025	12 – 48	1.8696	0.0187	10 – 18	4	1.36%
11×132	8.2263	0.0027	13 – 49	2.0493	0.0136	10 – 24	4	1.23%
12×144	8.9854	0.0040	16 – 60	2.2636	0.0114	10 – 21	4	1.13%
13×156	9.7387	0.0043	15 – 56	2.4347	0.0115	10 – 19	5	1.04%
14×168	10.5009	0.0043	15 – 51	2.6507	0.0157	11 – 22	5	0.96%
15×180	11.2581	0.0047	17 – 62	2.8367	0.0240	13 – 27	6	0.90%
16×192	12.0052	0.0043	17 – 84	3.0003	0.0222	13 – 24	6	0.83%
17×204	12.7803	0.0048	18 – 78	3.2044	0.0189	13 – 23	6	0.79%
18×216	13.5260	0.0077	21 – 81	3.3860	0.0367	15 – 30	7	0.74%
19×228	14.2564	0.0081	23 – 85	3.5621	0.0311	15 – 29	7	0.70%
20×240	15.0424	0.0089	24 – 93	3.7681	0.0418	16 – 32	7	0.66%
25×300	18.8121	0.0180	35 – 97	4.7430	0.0439	18 – 39	9	0.53%
30×360	22.5836	0.0208	40 –105	5.7542	0.0452	20 – 40	11	0.45%

Tab. 7.4: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des dreidimensionalen XY–Modells mit periodischen Randbedingungen.

7.2 Simulationsparameter

Um den Einfluß der Unsicherheit hinsichtlich des genauen Wertes der kritischen Temperatur des Modells zu minimieren, wurden die Simulationen bei einer Temperatur durchgeführt, die dem fehlergewichteten Mittelwert einiger jüngerer Werte der Literatur entspricht, vgl. Tab. 7.1:

$$\beta_0 = 0.4541670(32) \tag{7.2}$$

Durch Temperaturumwichtung wurden zusätzlich Zeitreihen für die Temperaturen $\beta_0 \pm \Delta\beta$ sowie die größte und die kleinste in den Mittelwert eingehende Schätzung

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle \lvert C \rvert \rangle}{V}$
	antiperiodische Randbedingungen							
4×50	1.0560	0.0015	4 – 10	0.3724	0.0009	2 – 3	1	1.69%
5×50	1.2832	0.0024	5 – 13	0.4556	0.0054	3 – 5	1	1.63%
6×50	1.5137	0.0018	5 – 14	0.5252	0.0032	3 – 6	1	1.59%
7×50	1.7699	0.0079	8 – 15	0.6323	0.0132	4 – 6	1	1.55%
8×50	2.0105	0.0094	9 – 20	0.6962	0.0090	4 – 7	1	1.52%
9×50	2.2334	0.0050	9 – 21	0.7943	0.0130	5 – 9	2	1.50%
10×50	2.4540	0.0098	11 – 22	0.8504	0.0109	5 – 8	2	1.49%
11×50	2.7122	0.0056	10 – 20	0.9442	0.0081	5 – 10	2	1.47%
12×50	2.9259	0.0100	12 – 20	1.0073	0.0067	5 – 11	2	1.45%
13×50	3.1864	0.0064	11 – 20	1.1036	0.0153	6 – 11	2	1.44%
14×54	3.4222	0.0071	12 – 22	1.1942	0.0132	6 – 10	2	1.32%
15×56	3.6657	0.0064	12 – 21	1.2614	0.0110	6 – 11	2	1.27%
16×60	3.9139	0.0098	13 – 26	1.3409	0.0138	6 – 11	2	1.18%
17×64	4.1362	0.0069	12 – 26	1.4139	0.0130	6 – 10	2	1.10%
18×68	4.3709	0.0060	13 – 26	1.5317	0.0292	8 – 12	3	1.03%
19×72	4.6307	0.0067	14 – 30	1.5877	0.0217	8 – 14	3	0.97%
20×76	4.8634	0.0107	16 – 27	1.6577	0.0246	8 – 13	3	0.92%
25×96	6.0604	0.0127	18 – 39	2.0582	0.0265	9 – 16	4	0.71%
30×114	7.2542	0.0087	18 – 42	2.4768	0.0615	12 – 18	5	0.59%

Tab. 7.5: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des dreidimensionalen XY–Modells mit antiperiodischen Randbedingungen.

für β_0, $\beta_{\min} = 0.45406$ und $\beta_{\max} = 0.45421$, erzeugt.

Die Wahl der longitudinalen Größe L_z der Systeme bei gegebenem $L_x = L_y$ erfolgte wiederum durch Betrachtung des Verhältnisses L_z/ξ_σ, das etwa 15 betragen sollte. Als Richtwerte für die Amplituden des FSS der Spin–Korrelationslänge wurden dazu $A_\sigma = 0.8$ für periodische bzw. $A_\sigma = 0.25$ für antiperiodische Randbedingungen angenommen. Für die kleineren Systeme mit antiperiodischen Randbedingungen wurde die Skalierung von L_z wie auch für die Ising–Systeme durch die Einführung einer konstanten Länge L_z ersetzt. Autokorrelations– und Äquilibrierungszeiten wurden analog zum Ising–Fall durch eine Binning– bzw. Zeitreihen–Analyse bestimmt und zur Festlegung der Simulationsparameter verwendet. Die relevanten Parameter für die Produktionsläufe für das XY–Modell sind in den Tabellen 7.2 und 7.3 zusammengestellt.

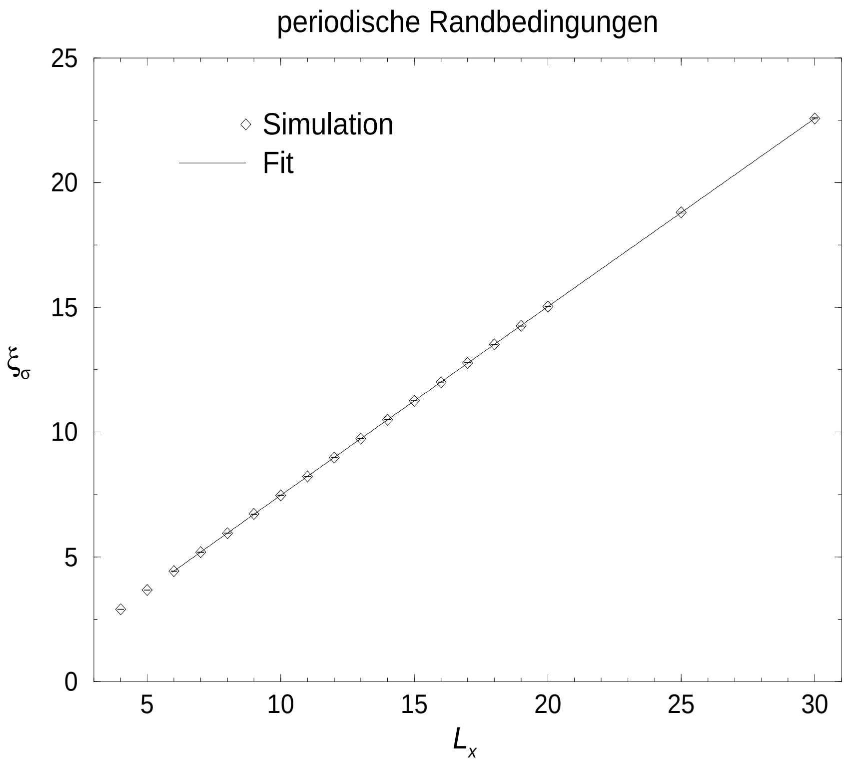
Bei der Analyse der Messungen für die Korrelationsfunktionen durch die Differenzenmethode wurde der Abstand Δ gemäß der Regel $\Delta \approx 2\xi_\epsilon$ gewählt.

7.3 Ergebnisse

Die Anwendung des in den vorangegangenen Kapiteln vorgestellten Apparates an Analyse–Methoden liefert wiederum eine Liste von Schätzungen $\bar{\xi}$ für die Korrelationslängen von lokaler Magnetisierung und Energiedichte in Abhängigkeit von den transversalen Systemgrößen L_x; nach den gewählten Randbedingungen geordnet sind diese in den Tabellen 7.4 und 7.5 zusammengestellt. Bei der Jackknife–Analyse wurde hier infolge der Erkenntnisse aus der Untersuchung des 3D Ising–Modells oben auf eine Bias–Reduktion unter Verwendung von *Double Jackknife Bias Corrected Estimators* verzichtet.

Das FSS–Verhalten der Korrelationslängen ist in den Abbn. 7.2 und 7.3 dargestellt. Zur Bestimmung der Amplituden des FSS der Korrelationslängen wurden Fits an Funktionen der Form (6.7) vorgenommen, d.h. unter Berücksichtigung einer effektiven Korrektur zum führenden Skalenverhalten. Daß solche Korrekturen mit der Länge der vorhandenen Zeitreihen deutlich aufgelöst werden können, zeigt die Auftragung der Amplituden ξ/L_x in den Abbildungen 7.4 und 7.5. Für die Systeme mit periodischen Randbedingungen ergibt die Fit–Prozedur folgende Schätzungen für die Amplituden des FSS der Korrelationslängen:

$$\begin{array}{|rrl|}
\hline
A_\sigma = & 0.75409(59) & (L_x = 6-30) \\
B_\sigma = & -0.298(79) & \\
\alpha_\sigma = & -0.62(15) & \\
\chi^2/g = & 1.39 & \\
Q = & 0.14 & \\
\hline
A_\epsilon = & 0.1899(15) & (L_x = 4-30) \\
B_\epsilon = & -0.050(41) & \\
\alpha_\epsilon = & -0.35(65) & \\
\chi^2/g = & 0.79 & \\
Q = & 0.70 & \\
\hline
\end{array} \qquad (7.3)$$



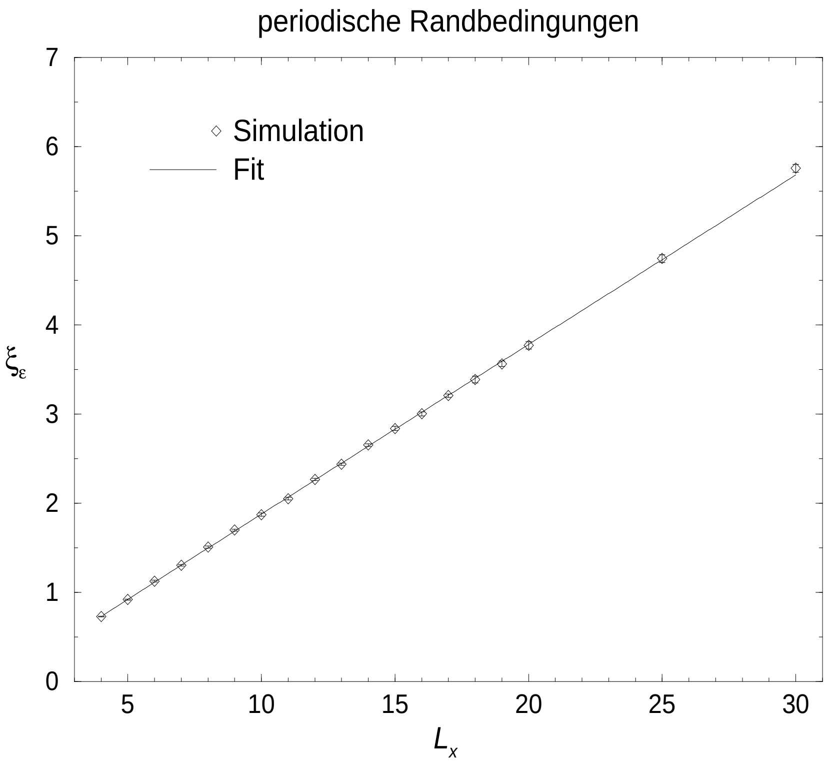
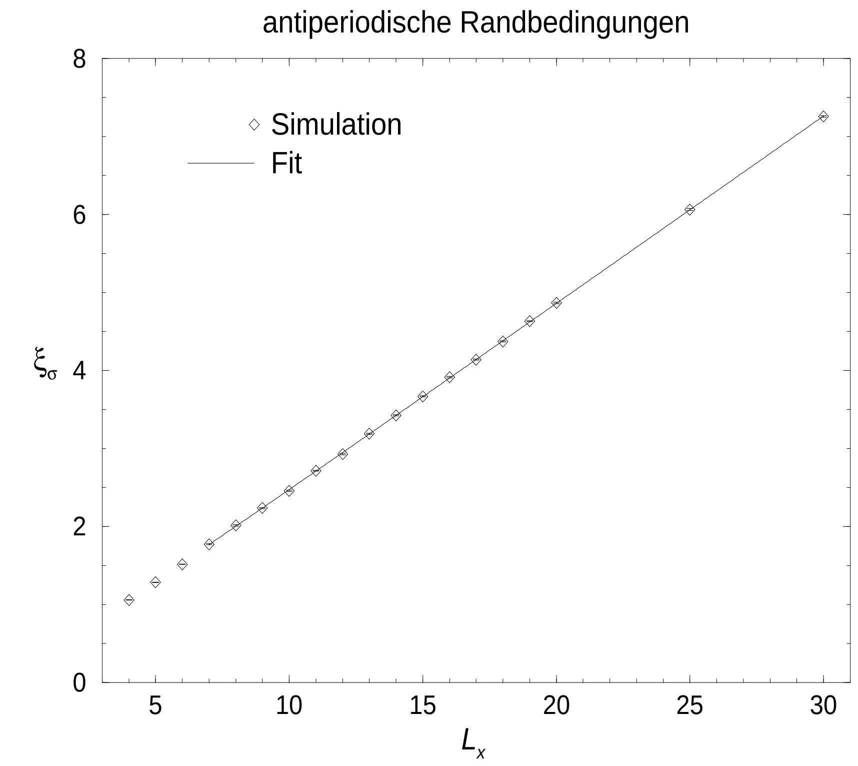


Abb. 7.2: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 3D XY–Modells mit periodischen Randbedingungen für Systeme mit $L_x = 4$ bis $L_x = 30$.



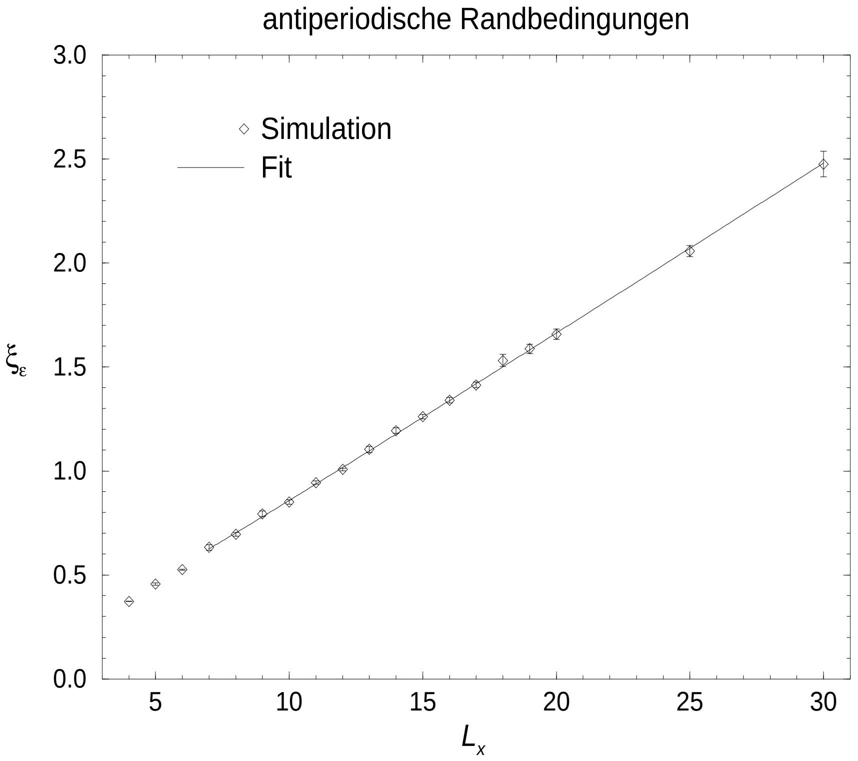
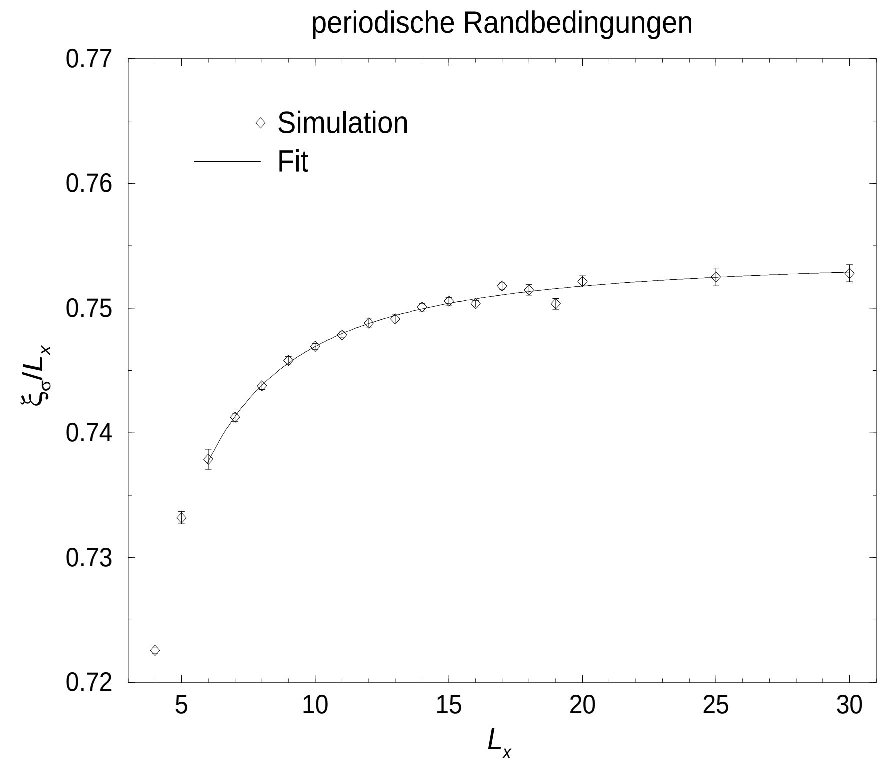


Abb. 7.3: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 3D XY–Modells mit antiperiodischen Randbedingungen für Systeme mit $L_x = 4$ bis $L_x = 30$.



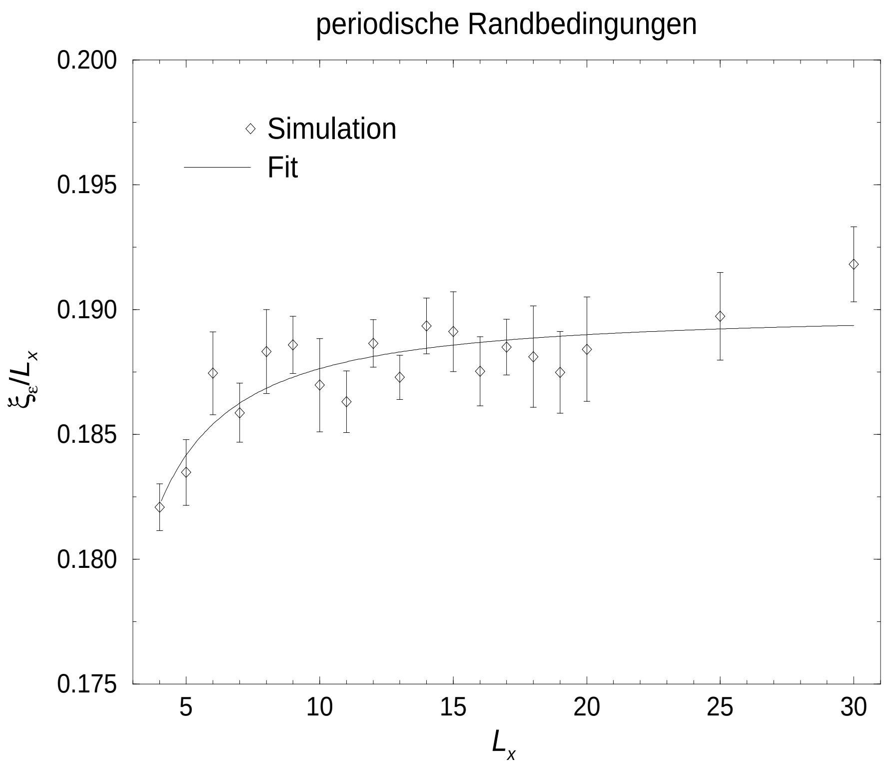
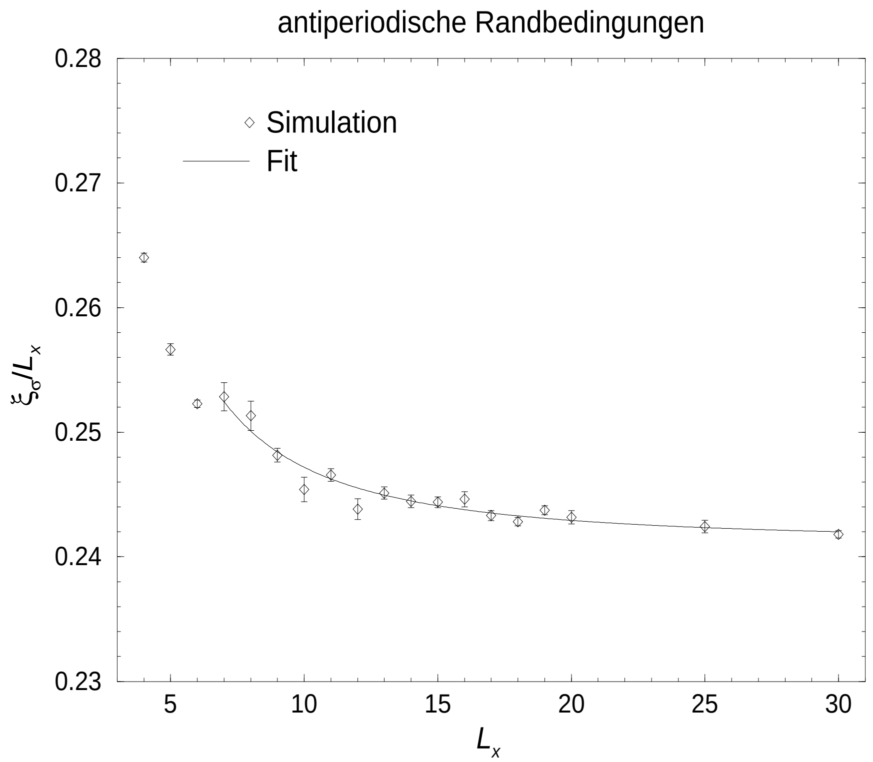


Abb. 7.4: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für periodische Randbedingungen. Die Kurven zeigen Fits an die Funktion $\xi = AL_x + BL_x^\alpha$; die Ausdehnung der Kurven entspricht dem in den Fit einbezogenen Bereich der L_x.



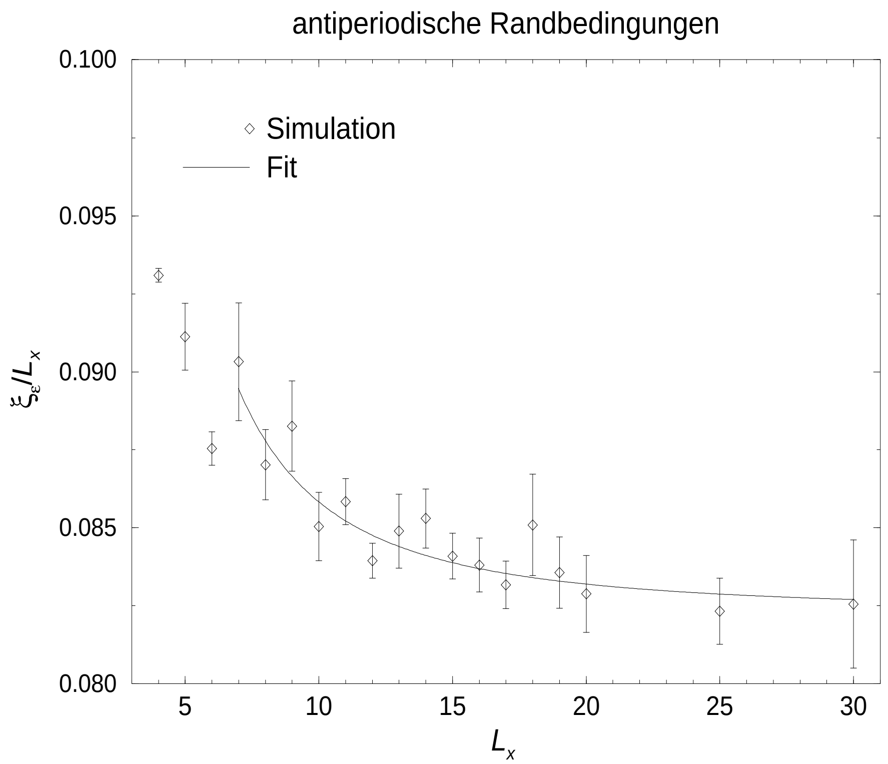


Abb. 7.5: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für antiperiodische Randbedingungen.

		$\beta_0 - \Delta\beta$	$\beta_0 + \Delta\beta$	β_{min}	β_{max}
periodisch	A_σ	0.75350(54)	0.75471(65)	—	0.7685(27)
	A_ϵ	0.1898(15)	0.1899(15)	0.1892(12)	0.1903(17)
	A_σ/A_ϵ	3.967(32)	3.974(32)	—	4.038(39)
antiperiodisch	A_σ	0.24106(58)	0.24120(56)	0.23846(102)	0.24198(46)
	A_ϵ	0.0823(13)	0.0823(13)	0.0822(13)	0.0823(13)
	A_σ/A_ϵ	2.929(47)	2.931(47)	2.901(48)	2.940(47)

Tab. 7.6: Ergebnisse für die Amplituden A_σ bzw. A_ϵ der Fits gemäß (6.7) für die Temperaturen $\beta_0 \pm \Delta\beta$ mit $\beta_0 = 0.4541670$ und $\Delta\beta = 0.0000032$ sowie die Temperaturen $\beta_{\text{min}} = 0.45406$ und $\beta_{\text{max}} = 0.45421$.

Für antiperiodische Randbedingungen erhält man:

$$\begin{array}{|rrl|}
\hline
A_\sigma = & 0.24113(57) & (L_x = 7 - 30) \\
B_\sigma = & 0.35(20) & \\
\alpha_\sigma = & -0.76(29) & \\
\chi^2/g = & 1.37 & \\
Q = & 0.17 & \\
\hline
A_\epsilon = & 0.0823(13) & (L_x = 7 - 30) \\
B_\epsilon = & 0.34 \pm 0.66 & \\
\alpha_\epsilon = & -0.98 \pm 1.02 & \\
\chi^2/g = & 0.68 & \\
Q = & 0.79 & \\
\hline
\end{array} \tag{7.4}$$

Da eine Festlegung des Korrekturexponenten α auf feldtheoretisch gewonnene Schätzwerte für die Fits beim Ising–Modell keinen Vorteil erbrachte, wurde hier auf eine solche Untersuchung verzichtet. Analysiert wurde jedoch der Einfluß etwaiger Abweichungen der Simulationstemperatur von der wahren kritischen Temperatur des Modells: durch Reweighting ergeben sich für die Temperaturen $\beta_0 \pm \Delta\beta$ gemäß Kap. 7.2 als veränderte Schätzungen für die Amplituden die in Tab. 7.6 zusammengestellten Werte. Im Vergleich mit den statistischen Fehlern ist hier offenbar der Einfluß der Unsicherheit hinsichtlich β_c nicht bedeutend. Anders verhält es sich mit den Ergebnissen für die zusätzlich berücksichtigten Temperaturen β_{min} und β_{max}; hier sind teilweise erhebliche Abweichungen zu beobachten. Abb. 7.6 zeigt dies für die Spin–Korrelationslänge der Systeme mit periodischen Randbedingungen, für die

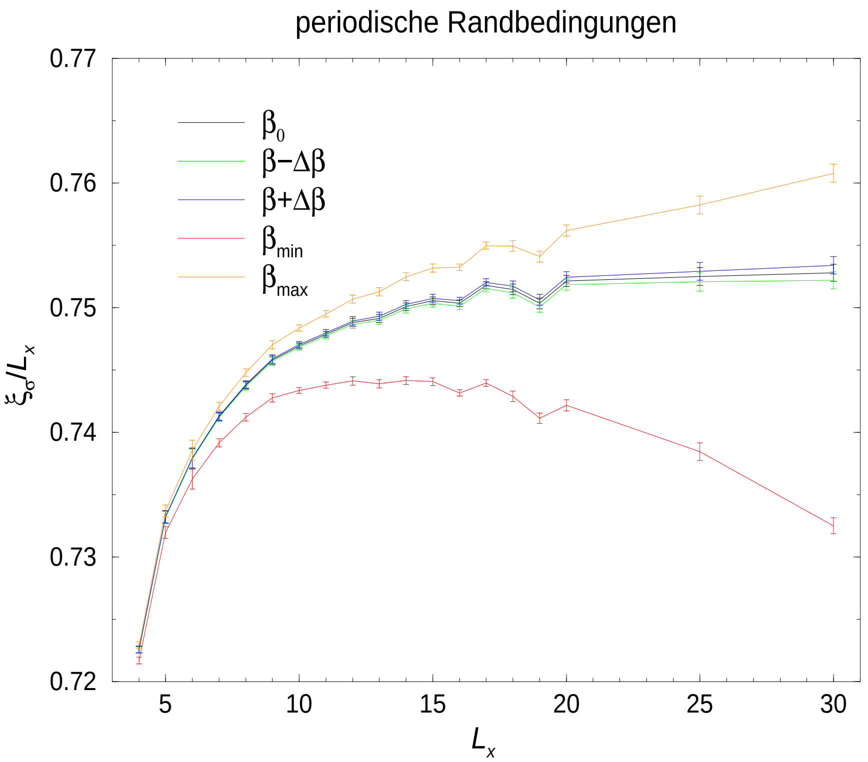


Abb. 7.6: Durch Reweighting gewonnene Schätzungen für den Verlauf der Amplitude A_σ (periodische Randbedingungen) für verschiedene Temperaturen um den (mutmaßlichen) kritischen Punkt des 3D XY–Modells. Die Temperaturen entsprechen den Bezeichnungen der Tab. 7.6.

der Effekt in Relation zu den statistischen Fehlern am stärksten ausgeprägt ist. Für die Temperatur $\beta_{\rm min}$ ist hier nicht einmal mehr ein Fit der Form (6.7) möglich, da die Amplituden sich nicht länger monoton verhalten. Aufgrund der vorhandenen Genauigkeit der Simulationsdaten könnte man sogar umgekehrt die Aussage treffen, daß $\beta_{\rm min}$ offenbar klar unterhalb der wahren kritischen Temperatur des 3D XY–Modells liegt. Aufgrund der höheren Genauigkeit ist der Einfluß des Reweighting im Vergleich zu den statistischen Fehlern für die Spin–Korrelationslängen generell größer als für die Korrelationslängen der Energiedichte. Der Einfluß auf die Amplituden–Verhältnisse A_σ/A_ϵ ist dabei jeweils deutlich geringer als der auf die Amplituden, da eine Änderung der Temperatur jeweils *beide* Korrelationslängen in die gleiche Richtung verschiebt. Die Schätzungen $\bar{\xi}(L_x)$ für die einzelnen Korrelationslängen aus den umgewichteten Zeitreihen sind im Anhang zusammengestellt.

Für die Simulationstemperatur β_0 erhält man für das Verhältnis der Amplituden des FSS der Korrelationslängen schließlich:

$$\boxed{\begin{array}{ll} A_\sigma/A_\epsilon = 3.971(32) & \text{für periodische RB} \\ A_\sigma/A_\epsilon = 2.930(47) & \text{für antiperiodische RB.} \end{array}} \tag{7.5}$$

Aus den Literaturwerten für ν und γ gemäß Tab. 7.1 ergibt sich für das Verhältnis der Skalendimensionen die Schätzung (vgl. Gl. (2.88)):

$$x_\epsilon/x_\sigma = 2.923(7). \tag{7.6}$$

Es zeigt sich somit eine sehr gute Übereinstimmung des Amplitudenverhältnisses mit dem Verhältnis der Skalendimensionen für die Systeme mit antiperiodischen Randbedingungen, keinerlei Übereinstimmung dagegen für die Systeme mit periodischen Randbedingungen. Eine lineare Beziehung zwischen den Amplituden des FSS der Korrelationslängen und den zugehörigen Skalendimensionen gemäß (7.1) ist daher auch für das XY–Modell in drei Dimensionen nur erfüllt, wenn man antiperiodische Randbedingungen entlang der T^2–Richtungen wählt.

Kapitel 8

Das 3D Heisenberg–Modell

Nachdem die Frage nach der Gültigkeit einer linearen Amplituden–Exponenten–Relation für das Ising– und das XY–Modell in ermutigend eindeutiger Weise beantwortet werden konnte, wollen wir mit demselben Methoden nun auch das FSS der Korrelationslängen des dreidimensionalen Heisenberg–Modells untersuchen.

Die methodischen Änderungen gegenüber der Analyse des XY–Modells beschränken sich auf die Anpassung des Simulationsprogramms von zweidimensionalen auf dreidimensionale Spin–Variablen. Die Analyse der Daten verläuft dagegen vollkommen analog, da die Korrelationsfunktionen selbst wieder Skalare sind. Die Parameter für die Simulations–Läufe wurden nach den in den vorangegangenen Kapiteln entwickelten Kriterien bestimmt und sind in den Tabellen 8.2 und 8.3 zusammengestellt. Gegenüber den Simulationen zum 3D Ising– und XY–Modell wurden jeweils nur halb so viele Werte der Zeitreihe zu einem Block zusammengefaßt, da für das Heisenberg–Modell aufgrund des höheren Rechenaufwandes eine etwas geringere Länge der Zeitreihen zu erwarten ist und dennoch — um den Einfluß des Bias zu minimieren — möglichst viele Bins zur Verfügung stehen sollen; die Blöcke sind jedoch weiterhin nahezu unkorreliert, wie es für die Jackknife–Analyse erforderlich ist. Die Größe der Systeme in der longitudinalen Richtung wurde unter Verwendung folgender Schätzwerte für die Amplituden des FSS der Spin–Korrelationslängen bestimmt:

$$A_\sigma = \begin{cases} 0.71 & \text{für periodische Randbedingungen} \\ 0.25 & \text{für antiperiodische Randbedingungen} \end{cases} . \tag{8.1}$$

Dabei sollte die Bedingung $\xi_\sigma / L_z \approx 15$ erfüllt sein.

Alle Simulationen wurden bei einer Temperatur durchgeführt, die einem Mittelwert

Methode	β_c	ν	γ	Quelle
HT–Reihe	0.69303(3)	0.715(3)	1.404(4)	[14]
HT–Reihe	0.69305(4)	0.716(2)	1.406(3)	[14]
HT–Reihe	0.6929(1)	0.712(10)	1.400(10)	[2]
MC	0.693002(12)	0.7128(14)	1.399(2)	[4]
MC	0.6930(1)	0.704(6)	1.389(14)	[50, 51]
MC	0.693035(37)	0.7036(23)	1.3896(70	[23]
MC	0.6929(1)	0.706(9)	1.390(23)	[73]
MC	0.6930(2)	—	—	[27]
Gewichtetes Mittel	0.693004(7)	0.71187(95)	1.4008(15)	

Tab. 8.1: Literaturwerte für die Temperatur des Phasenübergangs und die kritischen Exponenten ν und γ des dreidimensionalen Heisenberg–Modells.

einiger jüngerer Werte der Literatur für die kritische Temperatur des Heisenberg–Modells entspricht, vgl. Tab. 8.1:

$$\beta_0 = 0.693004(7). \tag{8.2}$$

Durch Verwendung der Reweighting–Technik wurden zusätzlich Zeitreihen für die Temperaturen $\beta_0 \pm \Delta\beta$ mit $\Delta\beta = 0.000007$ sowie die beiden Extremwerte $\beta_{\rm min} = 0.6929$ und $\beta_{\rm max} = 0.69305$ erzeugt.

Aus den Zeitreihen für die Korrelationsfunktionen wurden mit den in den vorangegangenen Kapiteln beschriebenen Methoden die Schätzungen $\bar{\xi}(L_x)$ für die Korrelationslängen von lokaler Magnetisierung und Energiedichte gewonnen. Der in die Bestimmung der Korrelationslängen mit Hilfe der Differenzenmethode eingehende Abstand wurde so gewählt, daß $\Delta \approx 2\xi_\epsilon$ erfüllt ist. Die Werte–Tabellen 8.4 und 8.5 zeigt die Ergebnisse dieser Analyse in Abhängigkeit von den gewählten Randbedingungen und den transversalen Gittergrößen L_x. Die zusätzlich aufgeführten Werte für die durchschnittlichen Größen der stochastischen Wolff–Cluster zeigen, daß diese offenbar für die Klasse der O(n)–Spin–Modelle mit wachsendem n bei gleicher Systemgröße kleiner werden. Die Beobachtung durchschnittlich kleinerer Cluster für die Systeme mit antiperiodischen Randbedingungen läßt sich auch im Falle des Heisenberg–Modells machen.

Die Abbn. 8.1 und 8.2 zeigen das FSS–Verhalten der Korrelationslängen des Heisenberg–Modells. Die Auftragung der Amplituden ξ/L_x in den Abbildungen 8.3

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	periodische Randbedingungen							
4×50	600	1500	42500	20	85	85	4096	1024
5×54			60000			120	4096	1024
6×64	1000	2500	60000	45	150	120	4096	1024
7×76			80000			160	4096	1024
8×86	2000	3000	80000	50	120	160	4096	1024
9×96			90000			180	4096	1024
10×108	4000	4500	90000	80	200	180	4096	1088
11×118			95000			190	4096	1024
12×128	6000	7000	95000	80	180	190	4096	1000
13×140			100000			200	4096	1002
14×150	9000	10000	100000	100	190	200	4096	1012
15×160			125000			250	4096	992
16×170	16000	18000	125000	160	250	250	4096	550
17×182			150000			300	4096	564
18×192	18000	20000	150000	240	300	300	4096	518
19×204			180000			360	4096	538
20×213	22000	26000	180000	250	360	360	4096	526
25×266	45000	45000	240000	260	480	480	4096	247
30×320	60000	60000	300000	320	600	600	4096	133

Tab. 8.2: Simulationsparameter für die Produktionsläufe der 3D Heisenberg–Systeme mit periodischen Randbedingungen.

und 8.4 zeigt, daß die Korrelationslängen nicht vollkommen linear mit der Systemgröße L_x skalieren und daß die Korrekturen zum FSS aufgelöst werden können. Bei den Schätzungen für die Energie–Korrelationslängen der Systeme mit periodischen Randbedingungen sind die Korrekturen vor dem Hintergrund der statistischen Fehler jedoch nur noch sehr schwach zu sehen. Hier wurde daher auf eine explizite Berücksichtigung der Korrekturen verzichtet und unter Auslassung des Punktes $L_x = 4$ ein linearer Fit vorgenommen.

In den übrigen Fällen wurde zur Berücksichtigung der Korrekturen zum FSS der

Systemgröße	$t_{\text{Äquil}}(\sigma)$	$t_{\text{Äquil}}(\epsilon)$	n_0	$\tau_{\text{int}}(\sigma)$	$\tau_{\text{int}}(\epsilon)$	$1/f$	μ	n/μ
	antiperiodische Randbedingungen							
4 × 50	300	500	42500	35	85	85	4096	2048
5 × 50			50000			100	4096	2048
6 × 50	1000	2000	50000	35	100	100	4096	2048
7 × 50			55000			110	4096	2048
8 × 50	1200	2000	55000	32	110	110	4096	2048
9 × 50			70000			140	4096	2048
10 × 50	2000	3000	70000	27	140	140	4096	2176
11 × 50			70000			140	4096	2100
12 × 50	2500	3000	70000	40	130	140	4096	1960
13 × 50			70000			140	4096	2000
14 × 54	3000	4000	70000	50	140	140	4096	2100
15 × 56			75000			150	4096	2040
16 × 60	4000	5000	75000	60	150	150	4096	1070
17 × 64			115000			230	4096	1090
18 × 68	5000	7000	115000	70	230	230	4096	1065
19 × 72			115000			230	4096	1047
20 × 76	9000	9000	115000	50	220	230	4096	1071
25 × 94	16000	20000	200000	100	400	400	4096	652
30 × 114	28000	32000	300000	140	600	600	4096	483

Tab. 8.3: Simulationsparameter für die Produktionsläufe der 3D Heisenberg–Systeme mit antiperiodischen Randbedingungen.

Verlauf $\bar{\xi}(L_x)$ der Korrelationslängen an Funktionen der Form

$$\xi(L_x) = AL_x + BL_x^{\alpha} \tag{8.3}$$

gefittet; dabei wurde auf eine Festlegung der Korrekturexponenten auf vorhandene feldtheoretische Schätzwerte verzichtet.

Für die eigentliche Simulationstemperatur β_0 ergaben sich so für die führenden Am-

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle\lvert C\rvert\rangle}{V}$
	periodische Randbedingungen							
4×50	2.7857	0.0048	8 – 20	0.6529	0.0134	4 – 6	1	2.81%
5×54	3.5339	0.0030	8 – 21	0.8482	0.0038	4 – 8	2	2.62%
6×64	4.2654	0.0020	7 – 25	1.0183	0.0080	5 – 11	2	2.22%
7×76	4.9870	0.0019	7 – 25	1.2023	0.0146	6 – 11	2	1.86%
8×86	5.7182	0.0028	10 – 32	1.3592	0.0154	7 – 13	3	1.65%
9×96	6.4408	0.0028	10 – 35	1.5596	0.0237	8 – 16	3	1.47%
10×108	7.1656	0.0030	11 – 39	1.7040	0.0098	7 – 17	3	1.30%
11×118	7.8862	0.0035	13 – 45	1.8676	0.0127	8 – 15	4	1.20%
12×128	8.6024	0.0036	14 – 52	2.0118	0.0188	9 – 16	4	1.09%
13×140	9.3366	0.0038	14 – 58	2.1719	0.0173	9 – 16	4	1.00%
14×150	10.0579	0.0042	16 – 52	2.4106	0.0295	11 – 20	5	0.93%
15×160	10.7784	0.0038	15 – 69	2.6005	0.0366	12 – 23	5	0.87%
16×170	11.4961	0.0050	15 – 70	2.6957	0.0313	11 – 20	5	0.82%
17×182	12.2187	0.0048	16 – 73	2.9047	0.0328	12 – 23	6	0.76%
18×192	12.9476	0.0054	17 – 70	3.0203	0.0419	13 – 28	6	0.72%
19×204	13.6669	0.0062	18 – 70	3.1934	0.0377	13 – 26	6	0.69%
20×213	14.3883	0.0064	20 – 78	3.4003	0.0301	13 – 27	7	0.65%
25×266	17.9859	0.0096	23 – 96	4.1676	0.0749	16 – 25	8	0.52%
30×320	21.6020	0.0127	24 – 75	5.1375	0.1204	18 – 26	10	0.42%

Tab. 8.4: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des dreidimensionalen Heisenberg–Modells mit periodischen Randbedingungen.

plituden als Schätzungen für die Systeme mit periodischen Randbedingungen:

$$\boxed{\begin{array}{rrl} A_\sigma = & 0.72068(34) & (L_x = 5-30) \\ B_\sigma = & -0.203(45) & \\ \alpha_\sigma = & -0.67(14) & \\ \chi^2/g = & 0.89 & \\ Q = & 0.57 & \\ \hline A_\epsilon = & 0.16966(36) & (L_x = 5-30) \\ \chi^2/g = & 0.94 & \\ Q = & 0.53 & \end{array}} \tag{8.4}$$

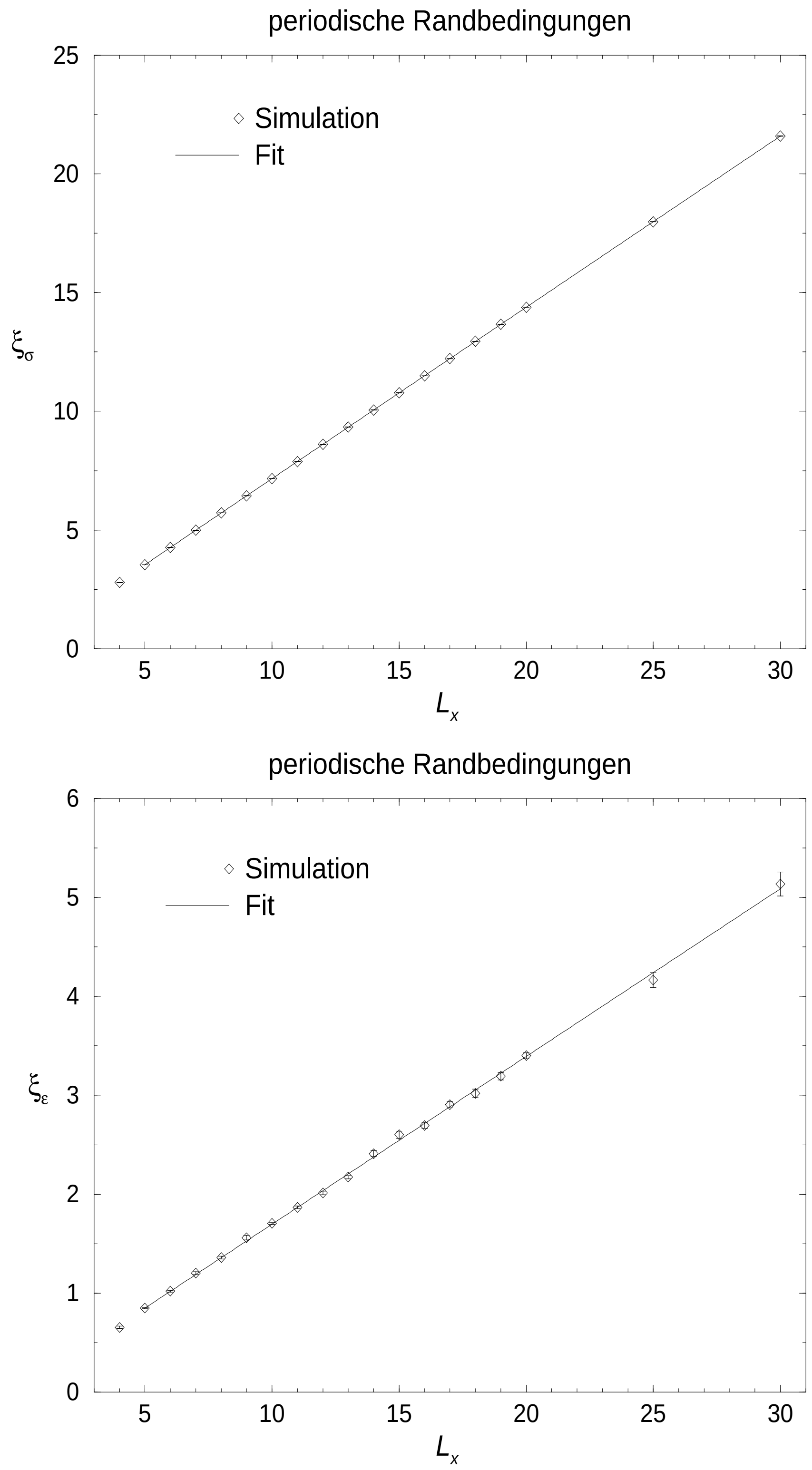
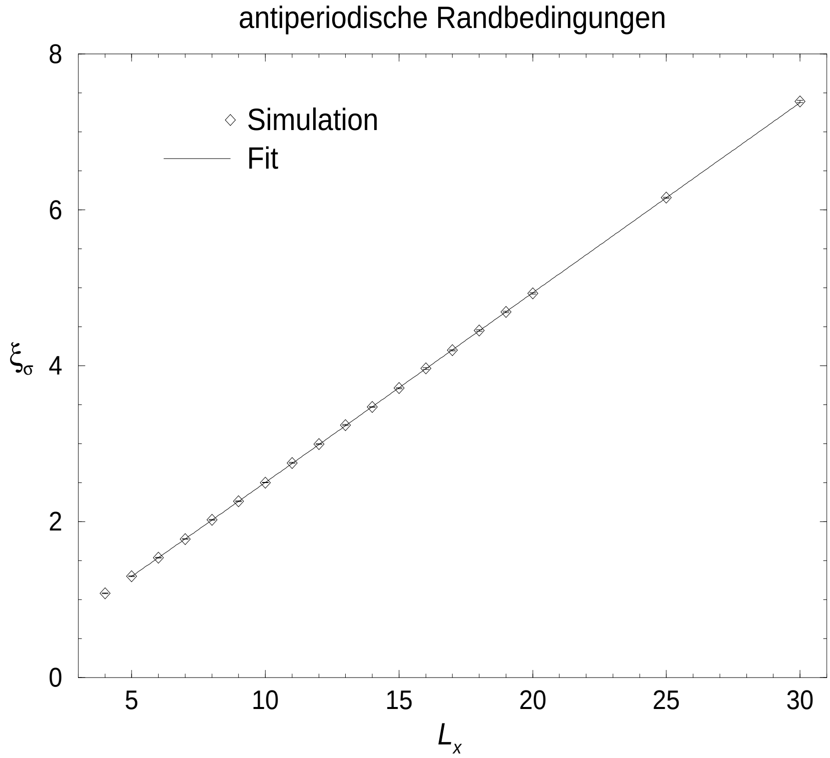


Abb. 8.1: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 3D Heisenberg–Modells mit periodischen Randbedingungen für Systeme mit $L_x = 4$ bis $L_x = 30$. Die Kurve im oberen Bild zeigt einen Fit an die Funktion $\xi = AL_x + BL_x^\alpha$, die im unteren einen linearen Fit gemäß $\xi_\epsilon = AL_x$; die Ausdehnung der Kurven entspricht dem in den Fit einbezogenen Bereich der L_x.



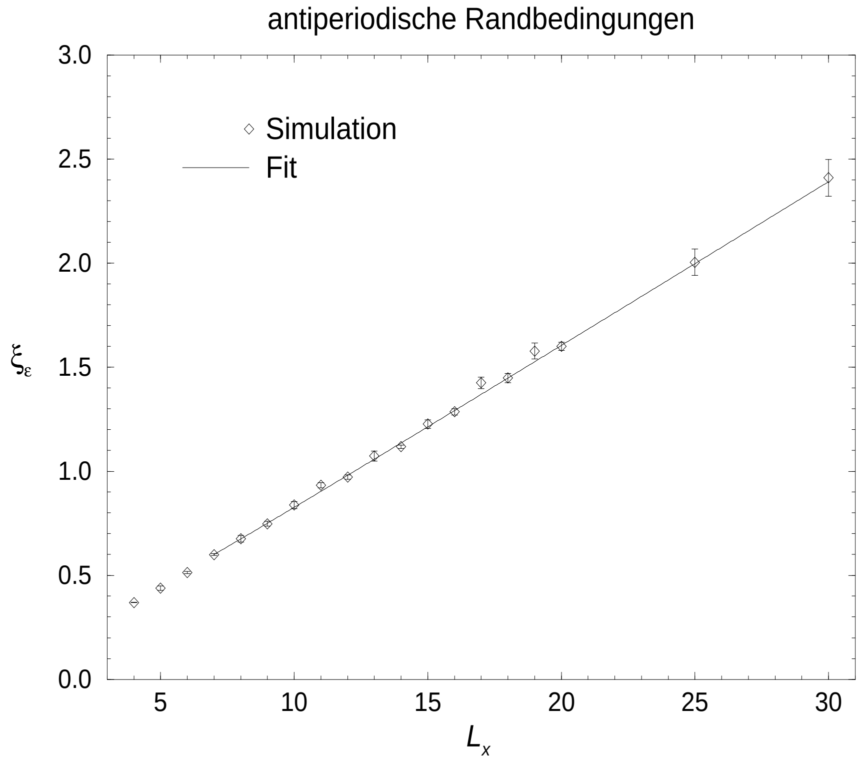
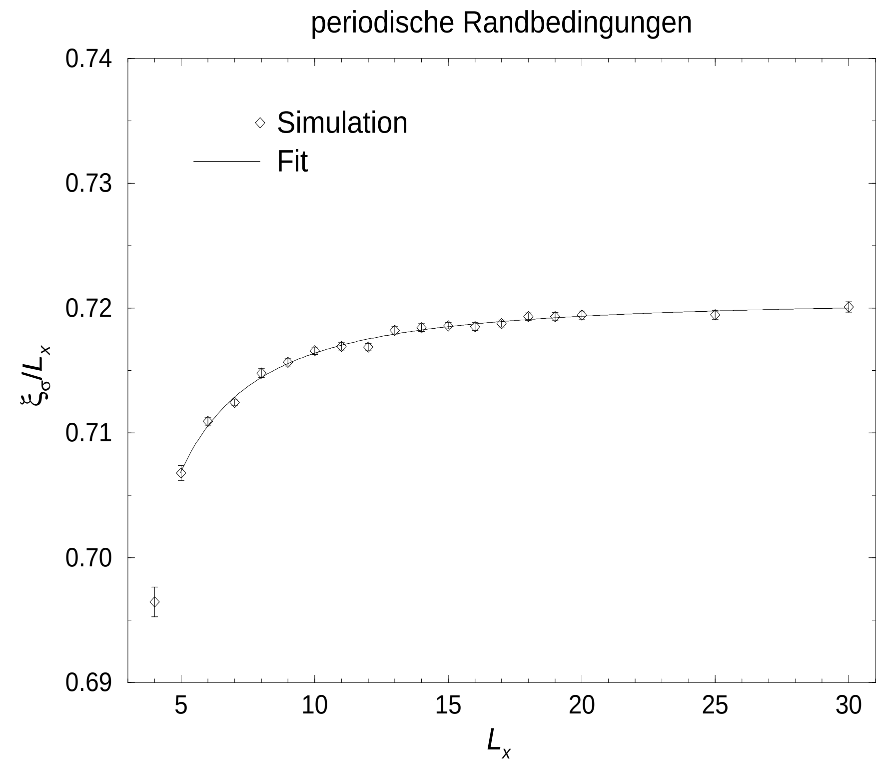


Abb. 8.2: FSS–Graphen für die Korrelationslängen von lokaler Magnetisierung (oben) und Energiedichte (unten) des 3D Heisenberg–Modells mit antiperiodischen Randbedingungen für Systeme mit $L_x = 4$ bis $L_x = 30$. Die Kurven zeigen Fits an die Funktion $\xi = AL_x + BL_x^\alpha$.



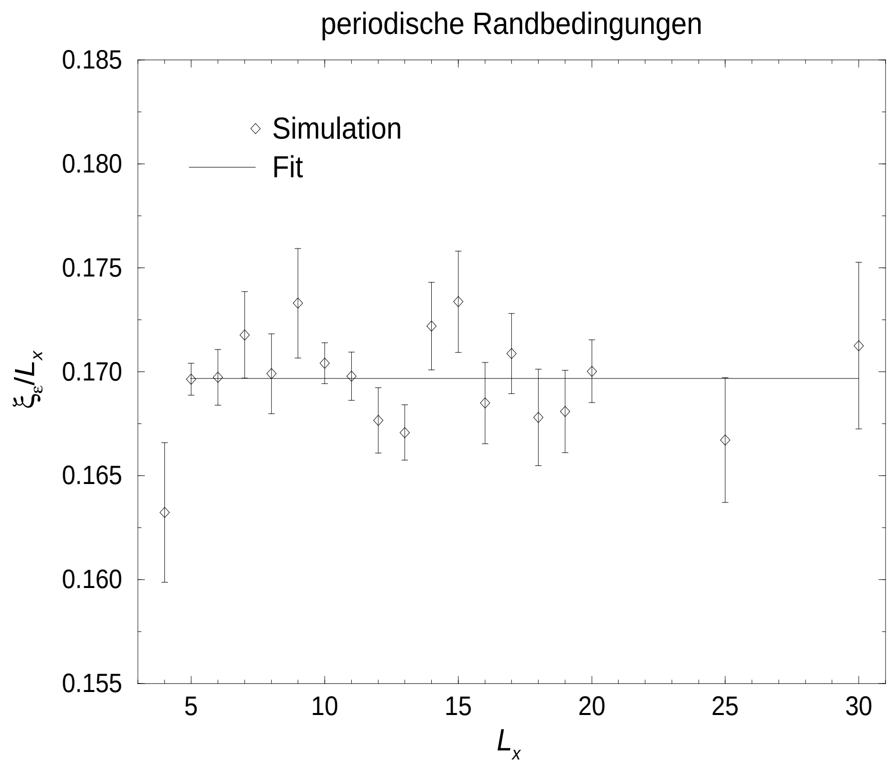
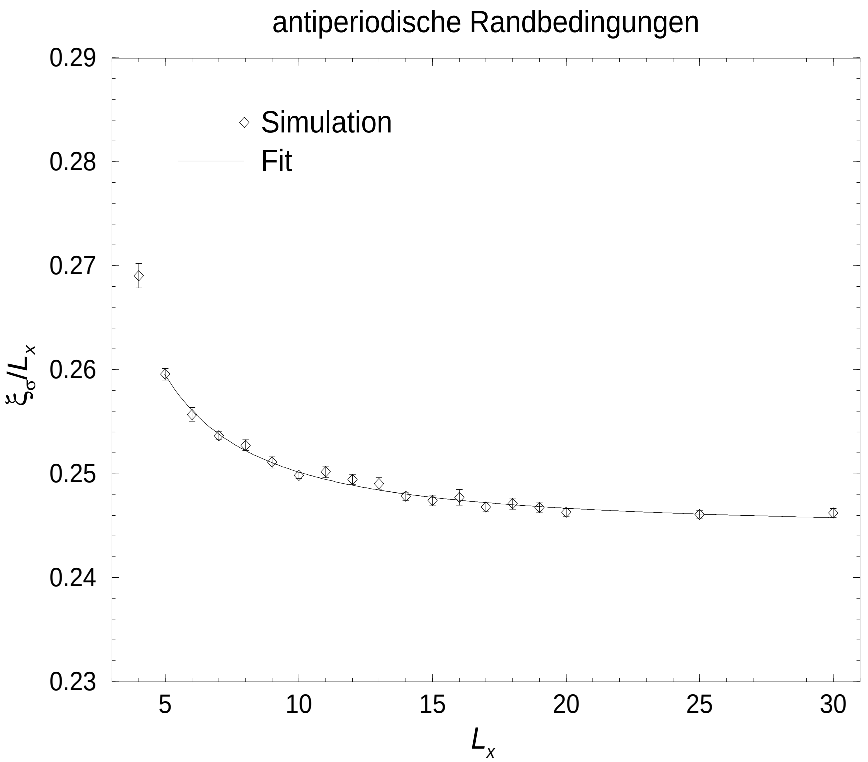


Abb. 8.3: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für periodische Randbedingungen.



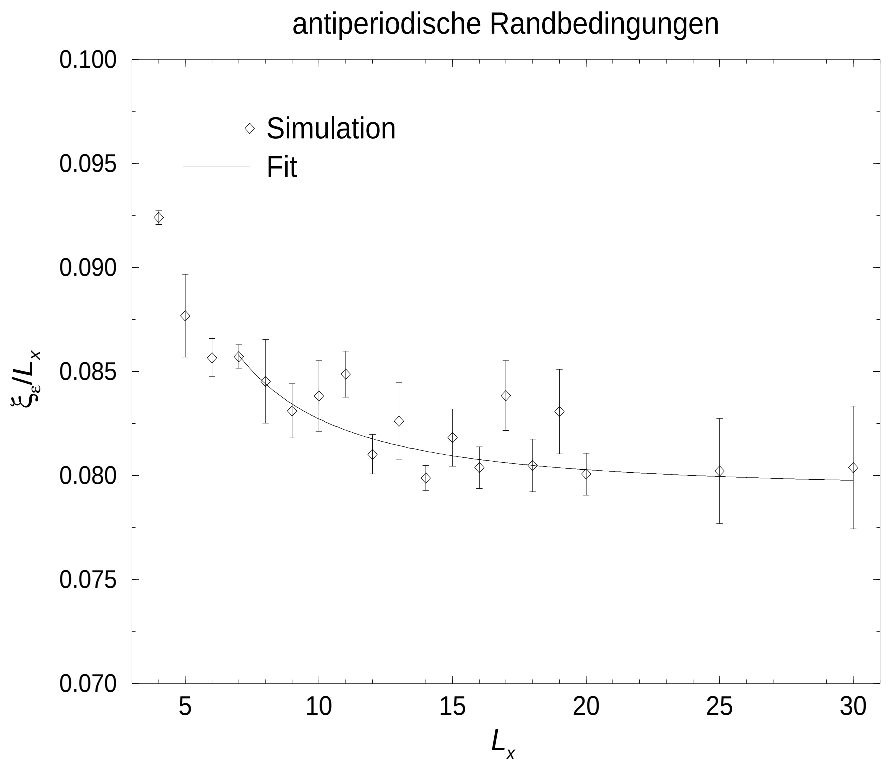


Abb. 8.4: Verlauf der Schätzungen für die Amplituden $\hat{A} = \bar{\xi}/L_x$ des FSS der Korrelationslängen von Spin und Energiedichte für antiperiodische Randbedingungen.

System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ	$\frac{\langle\lvert C\rvert\rangle}{V}$
	antiperiodische Randbedingungen							
4×50	1.0763	0.0047	5 – 9	0.3696	0.0013	2 – 4	1	1.56%
5×50	1.2979	0.0027	5 – 12	0.4384	0.0100	3 – 4	1	1.51%
6×50	1.5343	0.0040	6 – 13	0.5140	0.0056	3 – 5	1	1.47%
7×50	1.7757	0.0029	6 – 18	0.6000	0.0039	3 – 6	1	1.44%
8×50	2.0219	0.0041	7 – 16	0.6761	0.0161	4 – 6	1	1.42%
9×50	2.2602	0.0051	8 – 22	0.7479	0.0117	4 – 8	1	1.40%
10×50	2.4984	0.0030	8 – 21	0.8381	0.0170	5 – 8	2	1.39%
11×50	2.7521	0.0061	10 – 20	0.9335	0.0123	5 – 9	2	1.38%
12×50	2.9934	0.0056	10 – 22	0.9722	0.0114	5 – 9	2	1.37%
13×50	3.2379	0.0072	11 – 21	1.0739	0.0243	6 – 13	2	1.36%
14×54	3.4695	0.0059	11 – 22	1.1181	0.0086	5 – 9	2	1.25%
15×56	3.7118	0.0072	12 – 24	1.2272	0.0206	6 – 9	2	1.20%
16×60	3.9637	0.0122	14 – 26	1.2857	0.0160	6 – 11	3	1.11%
17×64	4.1955	0.0078	13 – 28	1.4250	0.0285	7 – 10	3	1.04%
18×68	4.4482	0.0094	14 – 26	1.4484	0.0229	7 – 13	3	0.97%
19×72	4.6885	0.0084	14 – 29	1.5782	0.0388	8 – 16	3	0.94%
20×76	4.9258	0.0077	14 – 32	1.6012	0.0202	7 – 14	3	0.87%
25×94	6.1524	0.0090	16 – 34	2.0051	0.0630	10 – 15	4	0.67%
30×114	7.3874	0.0128	20 – 45	2.4112	0.0886	10 – 13	5	0.57%

Tab. 8.5: Mittelwerte $\bar{\xi}$ für die Korrelationslängen von Spin und Energiedichte des dreidimensionalen Heisenberg–Modells mit antiperiodischen Randbedingungen.

Für die Systeme mit antiperiodischen Randbedingungen erhält man:

$$
\begin{array}{|rrl|}
\hline
A_\sigma = & 0.24462(51) & (L_x = 5-30) \\
B_\sigma = & 0.147(31) & \\
\alpha_\sigma = & -0.42(14) & \\
\chi^2/g = & 0.74 & \\
Q = & 0.75 & \\
\hline
A_\epsilon = & 0.0793(20) & (L_x = 7-30) \\
B_\epsilon = & 0.21(42) & \\
\alpha_\epsilon = & -0.79 \pm 1.15 & \\
\chi^2/g = & 1.38 & \\
Q = & 0.16 & \\
\hline
\end{array}
\qquad (8.5)
$$

		$\beta_0 - \Delta\beta$	$\beta_0 + \Delta\beta$	β_{min}	β_{max}
periodisch	A_σ	0.72034(36)	0.72124(38)	—	0.72542(77)
	A_ϵ	0.16972(41)	0.16967(36)	0.16964(36)	0.16967(36)
	A_σ/A_ϵ	4.2443(105)	4.2508(93)	—	4.2755(101)
antiperiodisch	A_σ	0.24455(51)	0.24469(50)	0.24351(61)	0.24507(47)
	A_ϵ	0.0793(20)	0.0793(20)	0.0792(19)	0.0790(22)
	A_σ/A_ϵ	3.084(78)	3.086(78)	3.075(74)	3.102(87)

Tab. 8.6: Ergebnisse für die Amplituden A_σ bzw. A_ϵ der Fits gemäß (8.3) für die Temperaturen $\beta_0 - \Delta\beta = 0.692997$, $\beta_0 + \Delta\beta = 0.693011$, $\beta_{\text{min}} = 0.6929$ und $\beta_{\text{max}} = 0.69305$.

Den Einfluß der Unsicherheit hinsichtlich der kritischen Temperatur des Heisenberg–Modells auf die Schätzungen für die Korrelationslängen verdeutlichen die in Tab. 8.6 aufgeführten Ergebnisse für die Amplituden bei den Temperaturen $\beta_0 \pm \Delta\beta$ sowie β_{min} und β_{max}. Für die Temperatur β_{min} zeigt sich ein ähnlicher Effekt wie im Falle des 3D XY–Modells: der Verlauf $\bar{\xi}_\sigma/L_x$ der Amplitude der Spin–Korrelationslänge für die Systeme mit periodischen Randbedingungen ist für diese Temperatur nicht mehr monoton, so daß kein sinnvoller Fit an eine Funktion der Form (8.3) mehr möglich ist. Daß sich die Amplituden mit der Temperatur β nicht immer monoton verhalten wie man es etwa aus Abb. 7.6 erwarten würde, ist eine Folge teilweise notwendiger Veränderungen des Bereiches $i_{\text{min}}, \ldots, i_{\text{max}}$ der in $\bar{\xi}$ eingehenden Werte. Diese Bereiche sowie die einzelnen Werte $\bar{\xi}(L_x)$ aus den umgewichteten Zeitreihen sind wiederum im Anhang aufgeführt. Die Korrekturen aus der Umwichtung auf $\beta_0 \pm \Delta\beta$ sind offenbar gegenüber den statistischen Fehlern zu vernachlässigen.

Für die Verhältnisse der führenden Amplituden des FSS der Korrelationslängen ergeben sich die Schätzungen:

$$\boxed{\begin{array}{ll} A_\sigma/A_\epsilon = 4.2478(92) & \text{für periodische RB} \\ A_\sigma/A_\epsilon = 3.085(78) & \text{für antiperiodische RB.} \end{array}} \tag{8.6}$$

Ein Vergleich mit dem Verhältnis x_ϵ/x_σ der Skalendimensionen wie es aus den in Tab. 8.1 aufgeführten Werten für die kritischen Exponenten ν und γ des Heisenberg–Modells folgt:

$$x_\epsilon/x_\sigma = 3.091(8), \tag{8.7}$$

zeigt eine hervorragende Übereinstimmung der Ergebnisse mit der Behauptung einer linearen Beziehung zwischen Amplituden und Exponenten des FSS der Korrelationslängen für antiperiodische Randbedingungen. Für die Systeme mit periodischen Randbedingungen läßt sich die Gültigkeit einer solchen Beziehung mit großer Sicherheit ausschließen.

Kapitel 9

Universelle Amplituden

Die Ergebnisse der bisherigen Untersuchungen zeigen, daß eine lineare Beziehung zwischen den Amplituden des FSS der Korrelationslängen von Spin und Energiedichte und den zugehörigen Skalendimensionen gemäß

$$\xi_i = \frac{A}{x_i} L \tag{9.1}$$

für die drei ersten Vertreter der Klasse der $\mathrm{O}(n)$–Spin–Modelle dann Gültigkeit hat, wenn man antiperiodische Randbedingungen in der Torusrichtung der Geometrie $T^2 \times \mathbb{R}$ wählt. Man kann vermuten, daß eine solche Beziehung zumindest innerhalb der Klasse der $\mathrm{O}(n)$–Spin–Modelle universelle Gültigkeit hat. Es ist dann von Interesse, den Charakter der „Meta–Amplitude" A in obiger Gleichung zu untersuchen. Nimmt man — aufgrund der Ergebnisse der Arbeit mit Berechtigung — an, daß diese Relation gültig ist, so läßt sich im Rückschluß die Amplitude A aus den gemessenen Amplituden des FSS der Korrelationslängen und Literaturwerten für die Skalendimensionen bestimmen. Diese Möglichkeit stellt einen direkten Vorteil der Untersuchung des Problems mit Hilfe einer MC–Simulation dar, da eine Analyse etwa mittels einer Transfermatrix–Rechnung die Hamiltonsche Formulierung der betrachteten Modelle verwenden muß, in der die Hamilton–Funktion nur bis auf einen Normierungsfaktor bestimmt ist, so daß sich die Amplitude A hieraus nicht erschließen läßt. Insbesondere soll hier die Frage beantwortet werden, ob die Amplitude A vom betrachteten Modell, d.h. dem Parameter n, abhängt, oder einen universellen Charakter hat.

Wir wollen hier die Amplituden A der untersuchten Modelle unter Annahme der Gültigkeit der linearen Amplituden–Exponenten–Relation für Systeme mit antiperi-

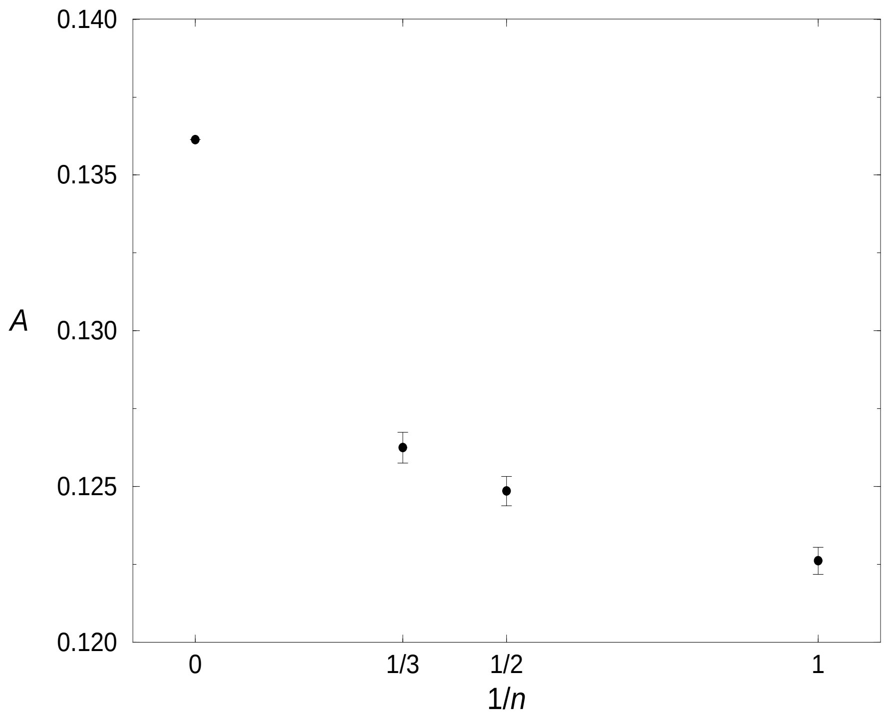


Abb. 9.1: Schätzungen für die Amplituden A der linearen Amplituden–Exponenten–Relation (9.1) in Abhängigkeit von $1/n$. $n = 1$ entspricht dem Ising–, $n = 2$ dem XY– und $n = 3$ dem Heisenberg–Modell. Die Werte wurden aus den gemessenen Amplituden des FSS der Spin–Korrelationslängen der Modelle gewonnen. Der Wert für den Limes $n \to \infty$ folgt aus der Äquivalenz des sphärischen Modells und des O(n)–Spin–Modells in diesem Limes [80, 48].

odischen Randbedingungen aus den Skalen–Amplituden der Spin–Korrelationslänge bestimmen, da diese in allen Fällen genauer bestimmt werden konnten als die Amplituden der Energie–Korrelationslänge. Aus den üblichen kritischen Exponenten läßt sich die Skalendimension x_σ über folgende Beziehung bestimmen (vgl. Gln. (2.34), (2.35) und (2.37)):

$$x_\sigma = \frac{\beta}{\nu} = \frac{d - \gamma/\nu}{2}. \tag{9.2}$$

Mit den in den Kapiteln 2, 7 und 8 angegebenen Werten für die Exponenten γ und ν erhält man daraus:

$$x_\sigma = \begin{cases} 0.5175(5) & \text{für das Ising–Modell} \\ 0.5178(15) & \text{für das XY–Modell} \\ 0.5161(17) & \text{für das Heisenberg–Modell} \end{cases} . \tag{9.3}$$

Aus den Schätzwerten für die führenden Amplituden des FSS der Spin–Korrelationslängen der Modelle mit antiperiodischen Randbedingungen ergeben sich damit für die Amplituden A nach (9.1) die Werte:

$$A = A_\sigma x_\sigma = \begin{cases} 0.12262(43) & \text{für das Ising–Modell } (n=1) \\ 0.12486(47) & \text{für das XY–Modell } (n=2) \\ 0.12625(49) & \text{für das Heisenberg–Modell } (n=3) \end{cases} \quad . \tag{9.4}$$

Während die entsprechenden Werte für höhere n zunächst nicht vorliegen, läßt sich ein solcher jedoch für den Grenzfall $n \to \infty$ angeben, der ja dem sphärischen Modell entspricht. Wie in Kap. 2 erwähnt, ergibt sich hier der Wert

$$A = \frac{1}{2\pi\sqrt{2}}\left(1 - \frac{25}{8\pi^2}\right)^{-1/2} \approx 0.13614. \tag{9.5}$$

Zur Veranschaulichung ist in Abb. 9.1 der Verlauf von A gegen $1/n$ aufgetragen. Unter Berücksichtigung der Genauigkeit der vorliegenden Werte läßt sich klar feststellen, daß die Amplitude A als Funktion von n nicht konstant ist. Sie ist daher nicht, wie vermutet wurde [24], universell im Sinne einer Unabhängigkeit vom betrachteten Modell, so daß die einzige Abhängigkeit der Amplituden des FSS der Korrelationslängen in ihrer linearen Abhängigkeit von den Skalendimensionen kondensiert wäre. Die vorliegenden Daten stellen einen guten Ausgangspunkt dar, um eine detailliertere Aussage über das Verhalten von A mit der Ordnung n zu machen; geeignet wäre hierfür etwa eine $1/n$–Entwicklung, die die Lücke zwischen $n = 3$ und $n = \infty$ verkleinert, aber gerade für kleine n keine Aussagen machen kann, da sie stark divergent ist.

Kapitel 10

Zusammenfassung

In einer Monte–Carlo–Simulation unter Verwendung des Wolffschen Single–Cluster–Algorithmus untersuchten wir das Finite Size Scaling der Korrelationslängen von lokaler Magnetisierung und Energiedichte von Gittermodellen. Dabei betrachteten wir die ersten drei Vertreter der Klasse der $O(n)$–Spin–Modelle, das Ising–, das XY– und das Heisenberg–Modell auf dreidimensionalen kubischen Gittern in Approximation der Geometrie $S^1 \times S^1 \times \mathbb{R}$. In den transversalen Richtungen S^1 wurden dabei periodische und antiperiodische Randbedingungen analysiert.

Zur Bestimmung der Korrelationslängen der betrachteten Observablen wurde eine Reihe von Methoden entwickelt oder angepaßt, die zunächst durch Anwendung auf das zweidimensionale Ising–Modell auf der analogen Geometrie $S^1 \times \mathbb{R}$, für das exakte Ausdrücke für die Amplituden des FSS bekannt sind, getestet und optimal abgestimmt wurden. Aus den gemessenen Korrelationsfunktionen wurden Schätzungen für die Korrelationslängen durch eine kombinierte Bildung von Differenzen und Verhältnissen zur Elimination von additiven und multiplikativen Konstanten direkt bestimmt, ohne daß auf nichtlineare Fits zurückgegriffen werden mußte. Der Bestimmung jenes Bereichs der Korrelationsfunktion, wo ihr Abfall durch eine einzelne Exponentialfunktion beschrieben wird, wurde dabei große Aufmerksamkeit gewidmet, um systematische Fehler durch das zugrundeliegende diskrete Gitter zu unterdrücken. Zur Fehlerrechnung wurde extensiver Gebrauch von Jackknife–Verfahren gemacht, die der Nichtlinearität der Transformation von den Korrelationsfunktionen zu den Korrelationslängen Rechnung tragen.

Ziel der Arbeit war es, für die betrachteten Geometrien die Gültigkeit einer linearen Beziehung zwischen den Amplituden des FSS der Korrelationslängen und den

Skalendimensionen der zugehörigen Operatoren gemäß

$$\xi_i = A_i L, \quad A_i = A/x_i \tag{10.1}$$

in Abhängigkeit von den Randbedingungen der Systeme in der transversalen Torusrichtung T^2 zu untersuchen. Eine solche Beziehung läßt sich für die analogen zweidimensionalen Systeme mit *periodischen* Randbedingungen mit Mitteln der konformen Feldtheorie analytisch ableiten. Unter Berücksichtigung der auftretenden Korrekturen zum FSS erhielten wir für das dreidimensionale Ising–Modell die Amplituden:

$$\begin{aligned} &\left.\begin{aligned} A_\sigma &= 0.8183(32) \\ A_\epsilon &= 0.2232(16) \end{aligned}\right\} \quad \text{für periodische Randbedingungen} \\ &\left.\begin{aligned} A_\sigma &= 0.23694(80) \\ A_\epsilon &= 0.08661(31) \end{aligned}\right\} \quad \text{für antiperiodische Randbedingungen} \end{aligned} \tag{10.2}$$

Zur Überprüfung der Relation (10.1) ergab sich daraus für das Verhältnis:

$$\begin{aligned} A_\sigma/A_\epsilon &= 3.666(30) \quad \text{für periodische Randbedingungen} \\ A_\sigma/A_\epsilon &= 2.736(13) \quad \text{für antiperiodische Randbedingungen} \end{aligned} \tag{10.3}$$

Im Vergleich mit dem varianzgewichteten Mittelwert einer Reihe von neueren Literaturwerten für das Verhältnis der Skalendimensionen:

$$x_\epsilon/x_\sigma = 2.7326(16) \tag{10.4}$$

zeigt sich eine hervorragende Übereinstimmung mit dem Verhältnis der Amplituden des FSS der Korrelationslängen für den Fall *antiperiodischer* Randbedingungen, aber eine eindeutige Abweichung für periodische Randbedingungen. Gegenüber ersten Simulationen für das Ising–Modell von Weston [85] konnte die Genauigkeit um einen Faktor 15 soweit verbessert werden, daß die Gültigkeit der Beziehung (10.1) für das dreidimensionale Ising–Modelle als erwiesen gelten kann.

Ein Hauptanliegen der Arbeit lag in der Beantwortung der Frage, ob die angegebene lineare Amplituden–Exponenten–Relation von universeller Gültigkeit ist, oder ob es sich etwa um eine Art numerischen Zufalls für das Ising–Modell handelt. Daher wurden erstmals weitere Vertreter der Klasse der O(n)–Spin–Modelle mit ähnlichen Methoden auf das FSS der Korrelationslängen in der betrachteten hyperzylindrischen Geometrie hin untersucht. Für das XY–Modell erhielten wir für die führenden

Amplituden des FSS und deren Verhältnis die Schätzungen

$$\begin{aligned}
&A_\sigma &&= 0.75409(59) \\
&A_\epsilon &&= 0.1899(15) \qquad \text{für periodische Randbedingungen} \\
&A_\sigma/A_\epsilon &&= 3.971(32) \\
& && \\
&A_\sigma &&= 0.24113(57) \\
&A_\epsilon &&= 0.0823(13) \qquad \text{für antiperiodische Randbedingungen} \\
&A_\sigma/A_\epsilon &&= 2.930(47)
\end{aligned} \tag{10.5}$$

Der Vergleich mit dem Literaturwert für das Verhältnis der Skalendimensionen von

$$x_\epsilon/x_\sigma = 2.923(7) \tag{10.6}$$

bestätigt die Gültigkeit der linearen Amplituden–Exponenten–Relation auch für das XY–Modell mit antiperiodischen Randbedingungen mit hoher Genauigkeit und widerlegt es im Falle periodischer Randbedingungen.

Schließlich wurde auch für das Heisenberg–Modell eine FSS–Analyse durchgeführt. Es ergaben sich folgende Amplituden und Amplitudenverhältnisse:

$$\begin{aligned}
&A_\sigma &&= 0.72068(34) \\
&A_\epsilon &&= 0.16966(36) \qquad \text{für periodische Randbedingungen} \\
&A_\sigma/A_\epsilon &&= 4.2478(92) \\
& && \\
&A_\sigma &&= 0.24462(51) \\
&A_\epsilon &&= 0.0793(20) \qquad \text{für antiperiodische Randbedingungen} \\
&A_\sigma/A_\epsilon &&= 3.085(78)
\end{aligned} \tag{10.7}$$

Das Verhältnis der Skalendimensionen von lokaler Magnetisierung und Energiedichte ist hier gegeben durch:

$$x_\epsilon/x_\sigma = 3.091(8). \tag{10.8}$$

Daher ergibt sich auch im Falle des Heisenberg–Modells eine sehr gute Verträglichkeit der Ergebnisse mit einer linearen Beziehung zwischen Amplituden und Exponenten des FSS der Korrelationslängen im Falle antiperiodischer Randbedingungen, wohingegen die Ergebnisse für periodische Randbedingungen mit der Gültigkeit einer solchen Beziehung vollkommen unverträglich sind.

In Verbindung mit den analytischen Ergebnissen für das Gauß–Modell ($n = 0$) und das sphärische Modell ($n = \infty$), für die (10.1) im Falle antiperiodischer Randbedingungen direkt bewiesen werden kann, wird somit klar, daß eine lineare Beziehung zwischen den Amplituden des FSS der Korrelationslängen und den zugehörigen Skalendimensionen auf der Geometrie $T^2 \times \mathbb{R}$ mit großer Wahrscheinlichkeit universelle Gültigkeit hat. Wie in Kap. 2 diskutiert, sind hierbei zwei Tatsachen von besonderer Bedeutung:

1. Die analoge Aussage für zweidimensionale Systeme $S^1 \times \mathbb{R}$ läßt sich mit Hilfe der konformen Feldtheorie allgemein beweisen. Die hier betrachtete Geometrie $T^2 \times \mathbb{R}$ läßt sich jedoch gerade *nicht* mittels einer konformen Abbildung auf den flachen Raum $\mathbb{R}^3$ abbilden wie dies für $S^1 \times \mathbb{R}$ der Fall ist. Dies gibt einen Hinweis darauf, daß sich Aussagen der konformen Feldtheorie möglicherweise von einer allgemeineren Theorie, also ausgehend von einer größeren Gruppe von Transformationen, reproduzieren lassen, die dreidimensionale Geometrien wie die betrachtete hyperzylindrische Form $T^2 \times \mathbb{R}$ einschließt.

2. Während eine lineare Amplituden–Exponenten–Relation auf zweidimensionalen Streifen $S^2 \times \mathbb{R}$ für periodische Randbedingungen gültig ist, gilt dies für analoge dreidimensionale Systeme offenbar im Falle antiperiodischer Randbedingungen. Der Einfluß der Randbedingungen auf den Gehalt eines Modells an Skalenoperatoren ist in zwei Dimensionen für eine Reihe von Modellen genau untersucht worden [18]. Im Falle des Ising–Modells führt der Wechsel zu antiperiodischen Randbedingungen zur Einführung von Störoperatoren, so daß die lokaler Magnetisierung und die Energiedichte ihren Charakter als primäre Operatoren verlieren. Eine verallgemeinerte Analyse des Einflusses von antiperiodischen Randbedingungen auf die hier betrachteten Systeme bietet möglicherweise den Ansatzpunkt für ein theoretisches Verständnis des Zustandekommens der beobachteten linearen Amplituden–Exponenten–Relation auf dreidimensionalen Geometrien.

Ist eine lineare Relation zwischen Amplituden und Skalendimensionen gültig, so sind die Amplituden A gemäß (10.1) unabhängig vom betrachteten Skalenoperator und in diesem Sinne universell. Für die Systeme mit antiperiodischen Randbedingungen erhalten wir:

$$\begin{array}{ll} A{=}0.12262(43) & \text{für das Ising–Modell} \\ A{=}0.12486(47) & \text{für das XY–Modell} \\ A{=}0.12625(49) & \text{für das Heisenberg–Modell} \end{array} \tag{10.9}$$

Es läßt sich daher innerhalb der Fehler mit großer Sicherheit sagen, daß diese Amplituden *nicht* universell sind im Sinne einer Unabhängigkeit vom betrachteten *Modell* (innerhalb der Klasse der O(n)–Spin–Modelle); stattdessen sind sie — wenn auch schwach — von der Dimension n des Ordnungsparameters abhängig.

Literaturverzeichnis

[1] J. Adler, J. Phys. **A16**, 2585 (1983).

[2] J. Adler, C. Holm, W. Janke, Physica **A201**, 581 (1993).

[3] F.C. Alcaraz, H.J. Herrmann, J. Phys. **A20**, 5735 (1987).

[4] H.G. Ballesteros, L.A. Fernández, V. Martín-Mayor, A. Muñoz Sudupe, Phys. Lett. **B387**, 125 (1996).

[5] M.E. Barber: *Finite–Size Scaling*, in: *Phase Transitions and Critical Phenomena*, Vol 8, eds. C. Domb and J.L Lebowitz (Academic Press, London, 1983), S. 146.

[6] A.A. Belavin, A.M. Polyakov, A.B. Zamolodchikov, Nucl. Phys. **B241**, 333 (1984).

[7] B.A. Berg, *Double Jackknife Bias Corrected Estimators*, Preprint FSU–SCRI–90–100 (1990).

[8] T.H. Berlin, M. Kac, Phys. Rev. **86**, 821 (1952).

[9] K. Binder, D.W. Heermann, *Monte Carlo Simulation in Statistical Physics — An Introduction* (Springer, Berlin/Heidelberg/New York, 1988) [Springer Series in Solid–State Sciences, 80].

[10] H.W.J. Blöte, E. Luijten, J.R. Heringa, J. Phys. **A28**, 6289 (1995).

[11] S. Brandt, *Statistical and Computational Methods in Data Analysis* (North–Holland, Amsterdam/New York/Oxford, 1976).

[12] P. Butera, M. Comi, A.J. Guttmann, Phys. Rev. **B48**, 13987 (1993).

[13] P. Butera. M. Comi, Phys. Rev **B52**, 6185 (1995).

[14] P. Butera, M. Comi, Phys. Rev. **B56**, 8212 (1997).

[15] J.L. Cardy, J. Phys. **A17**, L385 (1984).

[16] J.L. Cardy, J. Phys. **A17**, L961 (1984).

[17] J.L. Cardy, J. Phys. **A18**, L757 (1985).

[18] J.L. Cardy, Nucl. Phys. **B275**, 200 (1986).

[19] J.L. Cardy, *Conformal Invariance*, in: *Phase Transitions and Critical Phenomena*, Vol 11, eds. C. Domb and J.L Lebowitz (Academic Press, London, 1987), S. 55.

[20] J.L. Cardy, Nucl. Phys. **B366**, 403 (1991).

[21] J.L. Cardy, private Mitteilung.

[22] J.H. Chen, M.E. Fisher, J. Physique **46**, 1645 (1985).

[23] K. Chen, A.M. Ferrenberg, D.P. Landau, Phys. Rev. **B48**, 3249 (1993).

[24] P. Christe, M. Henkel, *Introduction to Conformal Invariance and Its Applications to Critical Phenomena* (Springer, Berlin/Heidelberg/New York, 1993) [New Series m: Monographs, Lecture Notes in Physics, m 16].

[25] A. Compagner, Phys. Rev. **E52**, 5634 (1995).

[26] B. Derrida, L. De Seze, J. Physique **43**, 475 (1982).

[27] I. Dimitrović, P. Hasenfratz, J. Nager, F. Niedermayer, Nucl. Phys. **B350**, 893 (1991).

[28] C. Domb, Adv. in Phys. **9** (1960).

[29] B.A. Dubrovin, A.T. Fomenko, S.P. Novikov, *Modern Geometry — Methods and Applications, Part I* (Springer, New York, 1992) [Graduate Texts in Mathematics, 93].

[30] B. Dünweg, *Simulation of phase transitions: critical phenomena*, in: *Monte Carlo and Molecular Dynamics of Condensed Matter Systems*, eds. K. Binder, C. Ciccotti (Italian Physical Society, Bologna, 1995) [Proceedings of the Euroconference on Computer Simulation in Condensed Matter Physics and Chemistry, Como, 1995].

[31] B. Efron, *The Jackknife, the Bootstrap and Other Resampling Plans* (Society for Industrial and Applied Mathematics [SIAM], Philadelphia, 1982) [CBMS–NSF Regional Conference Series in Applied Mathematics].

[32] A.E. Ferdinand, M.E. Fisher, Phys. Rev. **185**, 832 (1969).

[33] A.M. Ferrenberg, A.M. Swendsen, Phys. Rev. Lett. **63**, 1658(E) (1989).

[34] A.M. Ferrenberg, D.P. Landau, Phys. Rev. **B44**, 5081 (1991).

[35] A.M. Ferrenberg, D.P. Landau, Y.J. Wong, Phys. Rev. Lett. **69**, 3382 (1992).

[36] M. Fisz, *Wahrscheinlichkeitsrechnung und mathematische Statistik* (VEB, Berlin, 1989).

[37] H. Flyvbjerg, H.G. Petersen, J. Chem. Phys. **91**, 461 (1989).

[38] D. Frenkel, *Advanced Monte Carlo Techniques*, in: *Computer Simulation in Chemical Physics*, ed. M.P. Allen, D.J. Tildesley (Kluwer Academic Publishers, Dordrecht/Boston/London, 1993), S. 93.

[39] A.P. Gottlob, M. Hasenbusch, Physica **A201**, 593 (1993).

[40] A.P. Gottlob, cond-mat/9406092 (1994).

[41] J.C. Le Guillou, J. Zinn–Justin, Phys. Rev. Lett. **39**, 95 (1977).

[42] J.C. Le Guillou, J. Zinn–Justin, Phys. Rev. **B21**, 3976 (1980).

[43] J.C. Le Guillou, J. Zinn–Justin, J. Physique Lett. **46**, L317 (1985).

[44] M. Hasenbusch, S. Meyer, Phys. Lett. **B241**, 238 (1990).

[45] M. Henkel, J. Phys. **A19**, L247 (1986).

[46] M. Henkel, J. Phys. **A20**, L769 (1987).

[47] M. Henkel, J. Phys. **A21**, L227 (1988).

[48] M. Henkel, R.A. Weston, **A25**, L207 (1992).

[49] A. Heuer, B. Dünweg, A.M. Ferrenberg, Comput. Phys. Commun. **103**, 1 (1997).

[50] C. Holm, W. Janke, Phys. Rev. **B48**, 936 (1993).

[51] C. Holm, W. Janke, Phys. Lett. **A173**, 8 (1993).

[52] F. James, Comput. Phys. Commun. **60**, 329 (1990).

[53] W. Janke, *Monte Carlo Simulations of Spin Systems*, in: *Computational Physics: Selected Methods — Simple Excercises — Serious Applications*, ed. K.H. Hoffmann, M. Schreiber (Springer, Berlin, 1996), S.10.

[54] W. Janke, Phys. Lett. **A148**, 306 (1990).

[55] W. Janke, K. Nather, Phys. Rev. **B48**, 7419 (1993).

[56] L.P. Kadanoff, H. Ceva, Phys. Rev. **B3**, 3918 (1971).

[57] M. Kaku, *Strings, Conformal Fields, and Topology — An Introduction* (Springer, New York, 1991) [Graduate Texts in Contemporary Physics].

[58] S.V. Ketov, *Conformal Field Theory* (World Scientific, Singapore, 1995).

[59] S. Kirkpatrick, E.P. Stoll, Journ. Comp. Phys. **40**, 517 (1981).

[60] S. Kobayashi, K. Nomitzu, *Foundations of Differential Geometry* (Interscience Publishers, New York/London/Sydney, 1969).

[61] J.M. Kosterlitz, D.J. Thouless, J. Phys. **C6**, 1181 (1973).

[62] M. Lüscher, Comput. Phys. Commun. **79**, 100 (1994).

[63] N.D. Mermin, H. Wagner, Phys. Rev. Lett. **17**, 1133 (1966).

[64] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[65] R. G. Miller, Biometrika **61**, 1 (1974).

[66] D.B. Murray, B.G. Nickel, *Revised estimates for critical exponents for the continuum N–vector model in 3 dimensions*, Unpublished Guelph University report (1991).

[67] M. Nakahara, *Geometry, Topology and Physics* (Institute of Physics Publishing, Bristol/Philadelphia, 1990) [Graduate Student Series in Physics].

[68] B.G. Nickel, J.J. Rehr, J. Stat. Phys. **61**, 1 (1990).

[69] M.P. Nightingale, Physica **83A**, 561 (1976).

[70] L. Onsager, Phys. Rev. **65**, 117 (1944).

[71] L. Onsager, B. Kaufmann, Phys. Rev. **76**, 1232, 1244 (1949).

[72] A. Patrascioiu, E. Seiler, Munich preprint MPI–PhT/95–73, hep–lat/9508014 (1995).

[73] P. Peczak, A.M. Ferrenberg, D.P. Landau, Phys. Rev. **B43**, 6087 (1991).

[74] W.H. Press et. al., *Numerical Recipes in C — The Art of Scientific Computing*, Second Edition (Cambridge University Press, Cambridge, 1992).

[75] V. Privman (ed.): *Finite Size Scaling and Numerical Simulation of Statistical Systems* (World Scientific, Singapore, 1990).

[76] P. Reinicke, J. Phys. **A20**, 4501 und 5325 (1987).

[77] W. Schucany, H. Gray, O. Owen, J. Amer. Statist. Assoc. **66**, 524 (1971).

[78] L.N. Shchur, H.W.J. Blöte, Phys. Rev. **E55**, R4905 (1997).

[79] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, Second Edition, 5 Vols. (Publish or Perish, Wilmington, 1979).

[80] H.E. Stanley, Phys. Rev. **176**, 718 (1968).

[81] R.H. Swendsen, Phys. Rev. **B20**, 2080 (1979).

[82] R.H. Swendsen, J.–S. Wang, Phys. Rev. Lett. **58**, 86 (1987).

[83] A.L. Talapov, H.W.J. Blöte, J. Phys. **A29**, 5727 (1996).

[84] I. Vattulainen, K. Kankaala, J. Saarinen, T. Ala-Nissila, Comput. Phys. Commun. **86**, 209 (1994).

[85] R.A. Weston, Phys. Lett. **B248**, 340 (1990).

[86] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

[87] U. Wolff, Nucl. Phys. **B334**, 581 (1990).

[88] J. Zinn-Justin, *Quantum Field Theory and Critical Phenomena* (Clarendon Press, Oxford, 1989).

Anhang A

Daten

In den folgenden Tabellen sind die Schätzungen $\bar{\xi}(L_x)$ für die Korrelationslängen von Spin und Energiedichte des XY– und des Heisenberg–Modells für die neben der eigentlichen Simulations–Temperatur durch Reweighting zusätzlich berücksichtigten Temperaturen aufgeführt. Diese sind:

- für das XY–Modell:

$$\begin{aligned} \beta_0 + \Delta\beta &= 0.4541638 \\ \beta_0 - \Delta\beta &= 0.4541702 \\ \beta_{\min} &= 0.45406 \\ \beta_{\max} &= 0.45421 \end{aligned} \tag{A.1}$$

- für das Heisenberg–Modell:

$$\begin{aligned} \beta_0 + \Delta\beta &= 0.693011 \\ \beta_0 - \Delta\beta &= 0.692997 \\ \beta_{\min} &= 0.6929 \\ \beta_{\max} &= 0.69305 \end{aligned} \tag{A.2}$$

Die Bereiche $i_{\min}, \ldots, i_{\max}$ wurden für die umgewichteten Schätzungen gegenüber der Schätzung für β_0 nur dann geändert, wenn mit den ursprünglichen Bereichen die Größen $\bar{\xi}$ gemäß der Differenzenmethode aufgrund negativer Argumente im Logarithmus nicht definiert waren. In diesem Fall mußte die obere Grenze $i_{\max}$ etwas verkleinert werden.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	periodische Randbedingungen, $\beta = \beta_0 - \Delta\beta$						
4×48	2.8901	0.0010	5 – 19	0.7283	0.0038	4 – 7	2
5×60	3.6657	0.0024	8 – 21	0.9173	0.0066	5 – 10	2
6×72	4.4270	0.0048	12 – 28	1.1247	0.0099	6 – 11	2
7×84	5.1882	0.0023	10 – 28	1.3010	0.0083	7 – 16	3
8×96	5.9494	0.0024	10 – 30	1.5065	0.0134	8 – 16	3
9×108	6.7112	0.0031	12 – 37	1.6972	0.0103	8 – 16	3
10×120	7.4681	0.0025	12 – 48	1.8696	0.0187	10 – 18	4
11×132	8.2250	0.0027	13 – 49	2.0492	0.0136	10 – 24	4
12×144	8.9837	0.0040	16 – 60	2.2635	0.0114	10 – 21	4
13×156	9.7367	0.0043	15 – 56	2.4346	0.0115	10 – 19	5
14×168	10.4984	0.0043	15 – 51	2.6506	0.0157	11 – 22	5
15×180	11.2552	0.0047	17 – 62	2.8366	0.0240	13 – 27	6
16×192	12.0018	0.0043	17 – 84	3.0002	0.0222	13 – 24	6
17×204	12.7763	0.0048	18 – 78	3.2043	0.0189	13 – 23	6
18×216	13.5214	0.0077	21 – 81	3.3857	0.0367	15 – 30	7
19×228	14.2511	0.0081	23 – 85	3.5617	0.0311	15 – 29	7
20×240	15.0364	0.0089	24 – 93	3.7676	0.0417	16 – 32	7
25×300	18.8014	0.0180	35 – 97	4.7428	0.0439	18 – 39	9
30×360	22.5657	0.0208	40 –105	5.7536	0.0452	20 – 40	11

Tab. A.1: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit periodischen Randbedingungen für die Temperatur $\beta_0 - \Delta\beta = 0.4541638$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	Δ
	periodische Randbedingungen, $\beta = \beta_0 + \Delta\beta$						
4×48	2.8903	0.0010	5 – 19	0.7283	0.0038	4 – 7	2
5×60	3.6661	0.0024	8 – 21	0.9174	0.0066	5 – 10	2
6×72	4.4276	0.0048	12 – 28	1.1247	0.0099	6 – 11	2
7×84	5.1891	0.0023	10 – 28	1.3011	0.0082	7 – 16	3
8×96	5.9506	0.0024	10 – 30	1.5066	0.0134	8 – 16	3
9×108	6.7129	0.0031	12 – 37	1.6972	0.0103	8 – 16	3
10×120	7.4703	0.0025	12 – 48	1.8697	0.0187	10 – 18	4
11×132	8.2276	0.0027	13 – 49	2.0494	0.0136	10 – 24	4
12×144	8.9871	0.0040	16 – 60	2.2636	0.0114	10 – 21	4
13×156	9.7408	0.0043	15 – 56	2.4347	0.0115	10 – 19	5
14×168	10.5034	0.0043	15 – 51	2.6508	0.0157	11 – 22	5
15×180	11.2610	0.0047	17 – 62	2.8368	0.0240	13 – 27	6
16×192	12.0087	0.0043	17 – 84	3.0005	0.0222	13 – 24	6
17×204	12.7843	0.0048	18 – 78	3.2044	0.0189	13 – 23	6
18×216	13.5307	0.0077	21 – 81	3.3862	0.0367	15 – 30	7
19×228	14.2617	0.0081	23 – 85	3.5625	0.0311	15 – 29	7
20×240	15.0484	0.0089	24 – 93	3.7695	0.0417	16 – 32	7
25×300	18.8227	0.0180	35 – 97	4.7431	0.0439	18 – 39	9
30×360	22.6014	0.0209	40 –105	5.7548	0.0451	20 – 40	11

Tab. A.2: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit periodischen Randbedingungen für die Temperatur $\beta_0 + \Delta\beta = 0.4541702$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	periodische Randbedingungen, $\beta = \beta_{\min}$						
4×48	2.8867	0.0010	5 – 19	0.7282	0.0038	4 – 7	2
5×60	3.6597	0.0024	8 – 21	0.9170	0.0066	5 – 10	2
6×72	4.4174	0.0048	12 – 28	1.1244	0.0100	6 – 11	2
7×84	5.1741	0.0023	10 – 28	1.2999	0.0083	7 – 16	3
8×96	5.9297	0.0024	10 – 30	1.5056	0.0135	8 – 16	3
9×108	6.6848	0.0030	12 – 37	1.6966	0.0103	8 – 16	3
10×120	7.4336	0.0025	12 – 48	1.8678	0.0188	10 – 18	4
11×132	8.1815	0.0027	13 – 49	2.0468	0.0136	10 – 24	4
12×144	8.9293	0.0040	16 – 60	2.2615	0.0115	10 – 21	4
13×156	9.6703	0.0042	15 – 56	2.4321	0.0116	10 – 19	5
14×168	10.4180	0.0043	15 – 51	2.6488	0.0158	11 – 22	5
15×180	11.1608	0.0047	17 – 62	2.8322	0.0240	13 – 27	6
16×192	11.8905	0.0042	17 – 84	2.9957	0.0225	13 – 24	6
17×204	12.6471	0.0048	18 – 78	3.2029	0.0192	13 – 23	6
18×216	13.3717	0.0076	21 – 81	3.3768	0.0370	15 – 30	7
19×228	14.0814	0.0080	23 – 85	3.5488	0.0317	15 – 29	7
20×240	14.8432	0.0088	24 – 93	3.7470	0.0415	16 – 32	7
25×300	18.4608	0.0180	35 – 97	4.7374	0.0435	18 – 39	9
30×360	21.9748	0.0192	40 –105	5.7561	0.0499	20 – 40	11

Tab. A.3: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit periodischen Randbedingungen für die Temperatur $\beta_{\min} = 0.45406$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	periodische Randbedingungen, $\beta = \beta_{\max}$						
4 × 48	2.8916	0.0010	5 – 19	0.7283	0.0038	4 – 7	2
5 × 60	3.6684	0.0024	8 – 21	0.9175	0.0066	5 – 10	2
6 × 72	4.4313	0.0048	12 – 28	1.1248	0.0099	6 – 11	2
7 × 84	5.1945	0.0023	10 – 28	1.3008	0.0087	7 – 14	3
8 × 96	5.9582	0.0024	10 – 30	1.5069	0.0134	8 – 16	3
9 × 108	6.7231	0.0031	12 – 37	1.6974	0.0103	8 – 16	3
10 × 120	7.4836	0.0025	12 – 48	1.8704	0.0187	10 – 18	4
11 × 132	8.2444	0.0027	13 – 49	2.0503	0.0136	10 – 24	4
12 × 144	9.0081	0.0040	16 – 60	2.2644	0.0114	10 – 21	4
13 × 156	9.7664	0.0043	15 – 56	2.4357	0.0115	10 – 19	5
14 × 168	10.5345	0.0044	15 – 51	2.6514	0.0156	11 – 22	5
15 × 180	11.2975	0.0047	17 – 62	2.8384	0.0240	13 – 27	6
16 × 192	12.0518	0.0043	17 – 84	3.0022	0.0221	13 – 24	6
17 × 204	12.8344	0.0049	18 – 78	3.2049	0.0189	13 – 23	6
18 × 216	13.5888	0.0078	21 – 81	3.3896	0.0367	15 – 30	7
19 × 228	14.3276	0.0082	23 – 85	3.5680	0.0309	15 – 29	7
20 × 240	15.1234	0.0090	24 – 93	3.7725	0.0415	16 – 32	7
25 × 300	18.9559	0.0180	35 – 97	4.7449	0.0443	18 – 39	9
30 × 360	22.8228	0.0219	40 –105	5.7622	0.0449	20 – 40	11

Tab. A.4: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit periodischen Randbedingungen für die Temperatur $\beta_{\max} = 0.45421$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	antiperiodische Randbedingungen, $\beta = \beta_0 - \Delta\beta$						
4 × 50	1.0560	0.0015	4 – 10	0.3724	0.0009	2 – 3	1
5 × 50	1.2832	0.0024	5 – 13	0.4556	0.0054	3 – 5	1
6 × 50	1.5137	0.0018	5 – 14	0.5252	0.0032	3 – 6	1
7 × 50	1.7699	0.0079	8 – 15	0.6322	0.0132	4 – 6	1
8 × 50	2.0104	0.0094	9 – 20	0.6962	0.0090	4 – 7	1
9 × 50	2.2333	0.0050	9 – 21	0.7943	0.0130	5 – 9	2
10 × 50	2.4539	0.0098	11 – 22	0.8504	0.0109	5 – 8	2
11 × 50	2.7121	0.0056	10 – 20	0.9442	0.0081	5 – 10	2
12 × 50	2.9257	0.0100	12 – 20	1.0073	0.0067	5 – 11	2
13 × 50	3.1862	0.0064	11 – 20	1.1036	0.0153	6 – 11	2
14 × 54	3.4220	0.0071	12 – 22	1.1942	0.0132	6 – 10	2
15 × 56	3.6654	0.0064	12 – 21	1.2614	0.0110	6 – 11	2
16 × 60	3.9136	0.0098	13 – 26	1.3409	0.0138	6 – 11	2
17 × 64	4.1358	0.0069	12 – 26	1.4139	0.0130	6 – 10	2
18 × 68	4.3705	0.0060	13 – 26	1.5316	0.0293	8 – 12	3
19 × 72	4.6303	0.0068	14 – 30	1.5876	0.0217	8 – 14	3
20 × 76	4.8629	0.0107	16 – 27	1.6576	0.0246	8 – 13	3
25 × 96	6.0596	0.0127	18 – 39	2.0581	0.0265	9 – 16	4
30 × 114	7.2528	0.0087	18 – 42	2.4767	0.0616	12 – 18	5

Tab. A.5: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_0 - \Delta\beta = 0.4541638$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	Δ
	antiperiodische Randbedingungen, $\beta = \beta_0 + \Delta\beta$						
4×50	1.0560	0.0015	4 – 10	0.3724	0.0009	2 – 3	1
5×50	1.2832	0.0024	5 – 13	0.4556	0.0054	3 – 5	1
6×50	1.5137	0.0018	5 – 14	0.5253	0.0032	3 – 6	1
7×50	1.7700	0.0079	8 – 15	0.6323	0.0132	4 – 6	1
8×50	2.0105	0.0094	9 – 20	0.6962	0.0090	4 – 7	1
9×50	2.2334	0.0050	9 – 21	0.7944	0.0130	5 – 9	2
10×50	2.4541	0.0098	11 – 22	0.8504	0.0109	5 – 8	2
11×50	2.7123	0.0056	10 – 20	0.9442	0.0081	5 – 10	2
12×50	2.9260	0.0100	12 – 20	1.0073	0.0067	5 – 11	2
13×50	3.1865	0.0064	11 – 20	1.1037	0.0153	6 – 11	2
14×54	3.4224	0.0071	12 – 22	1.1942	0.0132	6 – 10	2
15×56	3.6659	0.0064	12 – 21	1.2614	0.0110	6 – 11	2
16×60	3.9142	0.0098	13 – 26	1.3410	0.0138	6 – 11	2
17×64	4.1365	0.0069	12 – 26	1.4140	0.0130	6 – 10	2
18×68	4.3712	0.0060	13 – 26	1.5317	0.0292	8 – 12	3
19×72	4.6312	0.0068	14 – 30	1.5877	0.0217	8 – 14	3
20×76	4.8639	0.0107	16 – 27	1.6577	0.0246	8 – 13	3
25×96	6.0612	0.0127	18 – 39	2.0582	0.0265	9 – 16	4
30×114	7.2555	0.0087	18 – 42	2.4768	0.0615	12 – 18	5

Tab. A.6: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_0 + \Delta\beta = 0.4541702$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	antiperiodische Randbedingungen, $\beta = \beta_{\min}$						
4×50	1.0557	0.0015	4 – 10	0.3723	0.0009	2 – 3	1
5×50	1.2830	0.0024	5 – 12	0.4555	0.0054	3 – 5	1
6×50	1.5128	0.0018	5 – 14	0.5251	0.0032	3 – 6	1
7×50	1.7686	0.0079	8 – 15	0.6321	0.0132	4 – 6	1
8×50	2.0088	0.0094	9 – 20	0.6960	0.0090	4 – 7	1
9×50	2.2311	0.0050	9 – 21	0.7940	0.0130	5 – 9	2
10×50	2.4509	0.0098	11 – 22	0.8499	0.0110	5 – 8	2
11×50	2.7084	0.0056	10 – 20	0.9438	0.0082	5 – 10	2
12×50	2.9210	0.0100	12 – 20	1.0064	0.0067	5 – 11	2
13×50	3.1810	0.0064	11 – 20	1.1024	0.0154	6 – 11	2
14×54	3.4153	0.0071	12 – 22	1.1930	0.0133	6 – 10	2
15×56	3.6576	0.0065	12 – 21	1.2603	0.0110	6 – 11	2
16×60	3.9044	0.0099	13 – 26	1.3393	0.0139	6 – 11	2
17×64	4.1252	0.0069	12 – 26	1.4123	0.0131	6 – 10	2
18×68	4.3585	0.0060	13 – 26	1.5300	0.0295	8 – 12	3
19×72	4.6168	0.0068	14 – 30	1.5870	0.0218	8 – 14	3
20×76	4.8467	0.0107	16 – 27	1.6563	0.0249	8 – 13	3
25×96	6.0332	0.0128	18 – 39	2.0551	0.0268	9 – 16	4
30×114	7.2106	0.0088	18 – 42	2.4752	0.0625	12 – 18	5

Tab. A.7: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_{\min} = 0.45406$.

XY–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	Δ
	antiperiodische Randbedingungen, $\beta = \beta_{\mathrm{max}}$						
4×50	1.0562	0.0015	4 – 10	0.3724	0.0009	2 – 3	1
5×50	1.2834	0.0024	5 – 13	0.4557	0.0054	3 – 5	1
6×50	1.5141	0.0018	5 – 14	0.5253	0.0032	3 – 6	1
7×50	1.7705	0.0079	8 – 15	0.6323	0.0132	4 – 6	1
8×50	2.0112	0.0094	9 – 20	0.6962	0.0090	4 – 7	1
9×50	2.2343	0.0050	9 – 21	0.7945	0.0130	5 – 9	2
10×50	2.4553	0.0098	11 – 22	0.8506	0.0109	5 – 8	2
11×50	2.7137	0.0056	10 – 20	0.9444	0.0081	5 – 10	2
12×50	2.9278	0.0100	12 – 20	1.0076	0.0067	5 – 11	2
13×50	3.1885	0.0064	11 – 20	1.1041	0.0153	6 – 11	2
14×54	3.4250	0.0071	12 – 22	1.1947	0.0132	6 – 10	2
15×56	3.6689	0.0064	12 – 21	1.2618	0.0109	6 – 11	2
16×60	3.9177	0.0098	13 – 26	1.3416	0.0138	6 – 11	2
17×64	4.1406	0.0069	12 – 26	1.4146	0.0130	6 – 10	2
18×68	4.3758	0.0060	13 – 26	1.5323	0.0292	8 – 12	3
19×72	4.6364	0.0068	14 – 30	1.5879	0.0217	8 – 14	3
20×76	4.8702	0.0107	16 – 27	1.6582	0.0246	8 – 13	3
25×96	6.0714	0.0126	18 – 39	2.0594	0.0265	9 – 16	4
30×114	7.2718	0.0087	18 – 42	2.4776	0.0613	12 – 18	5

Tab. A.8: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D XY–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_{\mathrm{max}} = 0.45421$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	Δ
	periodische Randbedingungen, $\beta = \beta_0 - \Delta\beta$						
4×50	2.7856	0.0048	8 – 20	0.6529	0.0134	4 – 6	1
5×54	3.5405	0.0039	8 – 21	0.8573	0.0070	4 – 8	2
6×64	4.2641	0.0025	7 – 25	1.0092	0.0146	5 – 10	2
7×76	4.9856	0.0023	7 – 25	1.1991	0.0249	6 – 11	2
8×86	5.7175	0.0028	10 – 32	1.3592	0.0154	7 – 13	3
9×96	6.4400	0.0028	10 – 35	1.5595	0.0237	8 – 16	3
10×108	7.1644	0.0030	11 – 39	1.7041	0.0098	7 – 17	3
11×118	7.8848	0.0035	13 – 45	1.8676	0.0127	8 – 15	4
12×128	8.6007	0.0036	14 – 52	2.0117	0.0188	9 – 16	4
13×140	9.3344	0.0038	14 – 58	2.1718	0.0173	9 – 16	4
14×150	10.0554	0.0042	16 – 52	2.4106	0.0295	11 – 20	5
15×160	10.7754	0.0038	15 – 69	2.6007	0.0366	12 – 23	5
16×170	11.4926	0.0050	15 – 70	2.6958	0.0313	11 – 20	5
17×182	12.2147	0.0048	16 – 73	2.9047	0.0328	12 – 23	6
18×192	12.9429	0.0054	17 – 70	3.0201	0.0419	13 – 28	6
19×204	13.6616	0.0062	18 – 70	3.1930	0.0377	13 – 26	6
20×213	14.3823	0.0064	20 – 78	3.4003	0.0301	13 – 27	7
25×266	17.9756	0.0096	23 – 96	4.1672	0.0749	16 – 25	8
30×320	21.5860	0.0126	24 – 75	5.1385	0.1202	18 – 26	10

Tab. A.9: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit periodischen Randbedingungen für die Temperatur $\beta_0 - \Delta\beta = 0.692997$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	periodische Randbedingungen, $\beta = \beta_0 + \Delta\beta$						
4 × 50	2.7859	0.0048	8 – 20	0.6529	0.0134	4 – 6	1
5 × 54	3.5341	0.0030	8 – 21	0.8482	0.0038	4 – 8	2
6 × 64	4.2657	0.0020	7 – 25	1.0183	0.0080	5 – 11	2
7 × 76	4.9874	0.0019	7 – 25	1.2023	0.0146	6 – 11	2
8 × 86	5.7188	0.0028	10 – 32	1.3591	0.0154	7 – 13	3
9 × 96	6.4417	0.0028	10 – 35	1.5597	0.0237	8 – 16	3
10 × 108	7.1667	0.0030	11 – 39	1.7040	0.0098	7 – 17	3
11 × 118	7.8876	0.0035	13 – 45	1.8676	0.0127	8 – 15	4
12 × 128	8.6041	0.0036	14 – 52	2.0118	0.0188	9 – 16	4
13 × 140	9.3387	0.0038	14 – 58	2.1719	0.0173	9 – 16	4
14 × 150	10.0604	0.0042	16 – 52	2.4106	0.0295	11 – 20	5
15 × 160	10.7814	0.0038	15 – 69	2.6002	0.0366	12 – 23	5
16 × 170	11.4996	0.0050	15 – 70	2.6957	0.0313	11 – 20	5
17 × 182	12.2228	0.0048	16 – 73	2.9048	0.0328	12 – 23	6
18 × 192	12.9522	0.0055	17 – 70	3.0204	0.0419	13 – 28	6
19 × 204	13.6722	0.0062	18 – 70	3.1939	0.0377	13 – 26	6
20 × 213	14.3943	0.0064	20 – 78	3.4003	0.0301	13 – 27	7
25 × 266	17.9963	0.0096	23 – 96	4.1680	0.0748	16 – 25	8
30 × 320	21.6180	0.0127	24 – 75	5.1365	0.1205	18 – 26	10

Tab. A.10: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit periodischen Randbedingungen für die Temperatur $\beta_0 + \Delta\beta = 0.693011$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	periodische Randbedingungen, $\beta = \beta_{\min}$						
4×50	2.7839	0.0048	8 – 20	0.6528	0.0134	4 – 6	1
5×54	3.5307	0.0030	8 – 21	0.8482	0.0038	4 – 8	2
6×64	4.2605	0.0020	7 – 25	1.0183	0.0080	5 – 11	2
7×76	4.9799	0.0019	7 – 25	1.2024	0.0146	6 – 11	2
8×86	5.7085	0.0028	10 – 32	1.3592	0.0154	7 – 13	3
9×96	6.4278	0.0028	10 – 35	1.5587	0.0238	8 – 16	3
10×108	7.1488	0.0029	11 – 39	1.7042	0.0099	7 – 17	3
11×118	7.8654	0.0035	13 – 45	1.8673	0.0128	8 – 15	4
12×128	8.5768	0.0036	14 – 52	2.0111	0.0188	9 – 16	4
13×140	9.3050	0.0038	14 – 58	2.1711	0.0174	9 – 16	4
14×150	10.0204	0.0042	16 – 52	2.4105	0.0296	11 – 20	5
15×160	10.7341	0.0038	15 – 69	2.6039	0.0368	12 – 23	5
16×170	11.4442	0.0050	15 – 70	2.6968	0.0314	11 – 20	5
17×182	12.1590	0.0047	16 – 73	2.9041	0.0329	12 – 23	6
18×192	12.8788	0.0054	17 – 70	3.0182	0.0419	13 – 28	6
19×204	13.5883	0.0062	18 – 70	3.1866	0.0380	13 – 26	6
20×213	14.2998	0.0063	20 – 78	3.4001	0.0302	13 – 27	7
25×266	17.8333	0.0095	23 – 96	4.1620	0.0758	16 – 25	8
30×320	—	—	—	5.1414	0.1588	18 – 26	10

Tab. A.11: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit periodischen Randbedingungen für die Temperatur $\beta_{\min} = 0.6929$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	periodische Randbedingungen, $\beta = \beta_{\max}$						
4×50	2.7865	0.0048	8 – 20	0.6529	0.0134	4 – 6	1
5×54	3.5352	0.0030	8 – 21	0.8482	0.0038	4 – 8	2
6×64	4.2675	0.0020	7 – 25	1.0183	0.0080	5 – 11	2
7×76	4.9901	0.0019	7 – 25	1.2023	0.0146	6 – 11	2
8×86	5.7225	0.0028	10 – 32	1.3591	0.0154	7 – 13	3
9×96	6.4466	0.0028	10 – 35	1.5600	0.0237	8 – 16	3
10×108	7.1730	0.0030	11 – 39	1.7040	0.0098	7 – 17	3
11×118	7.8954	0.0035	13 – 45	1.8677	0.0127	8 – 15	4
12×128	8.6137	0.0036	14 – 52	2.0121	0.0188	9 – 16	4
13×140	9.3506	0.0038	14 – 58	2.1722	0.0173	9 – 16	4
14×150	10.0745	0.0042	16 – 52	2.4107	0.0294	11 – 20	5
15×160	10.7980	0.0038	15 – 69	2.5988	0.0365	12 – 23	5
16×170	11.5191	0.0051	15 – 70	2.6953	0.0313	11 – 20	5
17×182	12.2452	0.0048	16 – 73	2.9051	0.0327	12 – 23	6
18×192	12.9781	0.0055	17 – 70	3.0210	0.0419	13 – 28	6
19×204	13.7018	0.0062	18 – 70	3.1963	0.0376	13 – 26	6
20×213	14.4277	0.0064	20 – 78	3.4004	0.0301	13 – 27	7
25×266	18.0540	0.0096	23 – 96	4.1704	0.0746	16 – 25	8
30×320	21.7078	0.0129	24 – 75	5.1307	0.1215	18 – 26	10

Tab. A.12: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit periodischen Randbedingungen für die Temperatur $\beta_{\max} = 0.69305$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	antiperiodische Randbedingungen, $\beta = \beta_0 - \Delta\beta$						
4×50	1.0762	0.0047	5 – 9	0.3696	0.0013	2 – 4	1
5×50	1.2978	0.0027	5 – 12	0.4384	0.0100	3 – 4	1
6×50	1.5342	0.0040	6 – 13	0.5139	0.0056	3 – 5	1
7×50	1.7756	0.0029	6 – 18	0.6000	0.0039	3 – 6	1
8×50	2.0218	0.0041	7 – 16	0.6761	0.0161	4 – 6	1
9×50	2.2601	0.0051	8 – 22	0.7478	0.0117	4 – 8	1
10×50	2.4983	0.0030	8 – 21	0.8381	0.0170	5 – 8	2
11×50	2.7520	0.0061	10 – 20	0.9335	0.0123	5 – 9	2
12×50	2.9932	0.0056	10 – 22	0.9721	0.0114	5 – 9	2
13×50	3.2377	0.0072	11 – 21	1.0739	0.0243	6 – 13	2
14×54	3.4692	0.0059	11 – 22	1.1180	0.0086	5 – 9	2
15×56	3.7115	0.0072	12 – 24	1.2271	0.0206	6 – 9	2
16×60	3.9633	0.0122	14 – 26	1.2857	0.0160	6 – 11	3
17×64	4.1950	0.0078	13 – 28	1.4250	0.0285	7 – 10	3
18×68	4.4477	0.0094	14 – 26	1.4484	0.0229	7 – 13	3
19×72	4.6879	0.0084	14 – 29	1.5783	0.0388	8 – 16	3
20×76	4.9252	0.0077	14 – 32	1.6011	0.0202	7 – 14	3
25×94	6.1513	0.0090	16 – 34	2.0049	0.0630	10 – 15	4
30×114	7.3857	0.0128	20 – 45	2.4109	0.0887	10 – 13	5

Tab. A.13: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_0 - \Delta\beta = 0.692997$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	antiperiodische Randbedingungen, $\beta = \beta_0 + \Delta\beta$						
4 × 50	1.0763	0.0047	5 – 9	0.3696	0.0013	2 – 4	1
5 × 50	1.2979	0.0027	5 – 12	0.4384	0.0100	3 – 4	1
6 × 50	1.5343	0.0040	6 – 13	0.5140	0.0055	3 – 5	1
7 × 50	1.7757	0.0029	6 – 18	0.6000	0.0039	3 – 6	1
8 × 50	2.0220	0.0041	7 – 16	0.6761	0.0161	4 – 6	1
9 × 50	2.2603	0.0051	8 – 22	0.7480	0.0117	4 – 8	1
10 × 50	2.4986	0.0030	8 – 21	0.8382	0.0170	5 – 8	2
11 × 50	2.7523	0.0061	10 – 20	0.9335	0.0123	5 – 9	2
12 × 50	2.9936	0.0056	10 – 22	0.9722	0.0114	5 – 9	2
13 × 50	3.2381	0.0072	11 – 21	1.0738	0.0243	6 – 13	2
14 × 54	3.4698	0.0059	11 – 22	1.1181	0.0086	5 – 9	2
15 × 56	3.7121	0.0072	12 – 24	1.2272	0.0206	6 – 9	2
16 × 60	3.9640	0.0122	14 – 26	1.2857	0.0160	6 – 11	3
17 × 64	4.1959	0.0078	13 – 28	1.4251	0.0285	7 – 10	3
18 × 68	4.4487	0.0094	14 – 26	1.4485	0.0229	7 – 13	3
19 × 72	4.6890	0.0084	14 – 29	1.5782	0.0388	8 – 16	3
20 × 76	4.9264	0.0077	14 – 32	1.6013	0.0202	7 – 14	3
25 × 94	6.1534	0.0090	16 – 34	2.0052	0.0630	10 – 15	4
30 × 114	7.3891	0.0128	20 – 45	2.4115	0.0886	10 – 13	5

Tab. A.14: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_0 + \Delta\beta = 0.693011$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	i_{min}–i_{max}	Δ
	antiperiodische Randbedingungen, $\beta = \beta_{\mathrm{min}}$						
4×50	1.0761	0.0047	5 – 9	0.3696	0.0013	2 – 4	1
5×50	1.2975	0.0027	5 – 12	0.4383	0.0100	3 – 4	1
6×50	1.5337	0.0040	6 – 13	0.5139	0.0056	3 – 5	1
7×50	1.7749	0.0029	6 – 18	0.5999	0.0039	3 – 6	1
8×50	2.0208	0.0041	7 – 16	0.6757	0.0161	4 – 6	1
9×50	2.2588	0.0051	8 – 22	0.7462	0.0117	4 – 8	1
10×50	2.4966	0.0030	8 – 21	0.8376	0.0171	5 – 8	2
11×50	2.7500	0.0061	10 – 20	0.9334	0.0123	5 – 9	2
12×50	2.9908	0.0056	10 – 22	0.9719	0.0114	5 – 9	2
13×50	3.2346	0.0072	11 – 21	1.0733	0.0243	6 – 13	2
14×54	3.4655	0.0059	11 – 22	1.1174	0.0086	5 – 9	2
15×56	3.7073	0.0072	12 – 24	1.2262	0.0206	6 – 9	2
16×60	3.9583	0.0123	14 – 26	1.2855	0.0160	6 – 11	3
17×64	4.1893	0.0078	13 – 28	1.4248	0.0286	7 – 10	3
18×68	4.4411	0.0094	14 – 26	1.4474	0.0230	7 – 13	3
19×72	4.6803	0.0084	14 – 29	1.5789	0.0389	8 – 16	3
20×76	4.9166	0.0077	14 – 32	1.5999	0.0203	7 – 14	3
25×94	6.1366	0.0091	16 – 34	2.0026	0.0634	10 – 15	4
30×114	7.3623	0.0129	20 – 45	2.4070	0.0892	10 – 13	5

Tab. A.15: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_{\mathrm{min}} = 0.6929$.

Heisenberg–Modell							
System	$\bar{\xi}_\sigma$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	$\bar{\xi}_\epsilon$	$\sqrt{\widehat{\mathrm{VAR}}}$	$i_{\min}$–$i_{\max}$	Δ
	antiperiodische Randbedingungen, $\beta = \beta_{\max}$						
4 × 50	1.0763	0.0047	5 – 9	0.3696	0.0013	2 – 4	1
5 × 50	1.2980	0.0027	5 – 12	0.4384	0.0100	3 – 4	1
6 × 50	1.5345	0.0040	6 – 13	0.5140	0.0055	3 – 5	1
7 × 50	1.7761	0.0029	6 – 18	0.6000	0.0039	3 – 6	1
8 × 50	2.0224	0.0041	7 – 16	0.6763	0.0161	4 – 6	1
9 × 50	2.2608	0.0051	8 – 22	0.7487	0.0117	4 – 8	1
10 × 50	2.4992	0.0030	8 – 21	0.8384	0.0170	5 – 8	2
11 × 50	2.7531	0.0061	10 – 20	0.9336	0.0122	5 – 9	2
12 × 50	2.9946	0.0056	10 – 22	0.9723	0.0114	5 – 9	2
13 × 50	3.2394	0.0072	11 – 21	1.0738	0.0242	6 – 13	2
14 × 54	3.4713	0.0059	11 – 22	1.1184	0.0086	5 – 9	2
15 × 56	3.7138	0.0072	12 – 24	1.2276	0.0205	6 – 9	2
16 × 60	3.9661	0.0122	14 – 26	1.2858	0.0160	6 – 11	3
17 × 64	4.1982	0.0078	13 – 28	1.4252	0.0285	7 – 10	3
18 × 68	4.4513	0.0094	14 – 26	1.4331	0.0252	7 – 11	3
19 × 72	4.6921	0.0084	14 – 29	1.5779	0.0387	8 – 16	3
20 × 76	4.9299	0.0077	14 – 32	1.6018	0.0202	7 – 14	3
25 × 94	6.1594	0.0090	16 – 34	2.0062	0.0629	10 – 15	4
30 × 114	7.3986	0.0128	20 – 45	2.4134	0.0884	10 – 13	5

Tab. A.16: Mittelwerte $\bar{\xi}(L_x)$ für die Korrelationslängen des 3D Heisenberg–Modells mit antiperiodischen Randbedingungen für die Temperatur $\beta_{\max} = 0.69305$.

Anhang B

Programme

Zur Vervollständigung der Dokumentation sind im folgenden die C-Quellcodes der wichtigsten in der Arbeit verwendeten Computerprogramme aufgeführt. Im einzelnen handelt es sich um folgende Routinen:

- **ising.c:** Das Simulationsprogramm für die zwei- und dreidimensionalen Ising–Systeme. Die Konfiguration erfolgt über die im Kopf des Programms zusammengestellten „`#define`“–Anweisungen. Es lassen sich so Systeme der allgemeinen Geometrie $T^{n-1} \times \mathbb{R}$ simulieren. Zu Testzwecken sind neben dem Wolff–Single–Cluster–Algorithmus auch der Swendsen–Wang– sowie lokale Metropolis– und Heatbath–Algorithmen implementiert. Über den „`CORR`“–Parameter lassen sich verschiedene Methoden zur Bestimmung der Korrelationsfunktionen aktivieren.

- **xy.c:** Das Simulationsprogramm für $(n-1) \times 1$–dimensionale XY–Systeme.

- **heisen.c:** Das Simulationsprogramm für $(n-1) \times 1$–dimensionale Heisenberg–Systeme.

- **ranvec.c:** Implementation des verbesserten Schiebe–Register–Generators „R250/521“ [49] von B. Dünweg.

- **ranlux.c:** Der „RANLUX“–Generator nach M. Lüscher [62] in einer eigenen Implementation.

- **analyse.c:** Programm zur Bestimmung der Korrelationslängen aus den von den Simulationsprogrammen gelieferten Messungen für die Korrelationsfunktionen. In Abhängigkeit von der Zahl der übergebenen Parameter erfüllt es verschiedene Aufgaben:

  - `analyse` *file* bestimmt aus den Simulationsdaten in den mit *file* beginnenden Dateien die Schätzungen $\hat{\xi}(i)$ für die Korrelationslängen sowie deren Fehler nach der Jackknife–Methode mit und ohne Bias–Reduktion und schreibt die Ergebnisse in eine Datei.
  - `analyse` *file* `auto` bestimmt für alle möglichen Bereiche $i_{\min}, \ldots, i_{\max}$ der kovarianzgewichteten Mittelung der $\hat{\xi}(i)$ die zugehörigen Werte von χ^2/g und schreibt die Matrix in eine Datei.
  - `analyse` *file* i^{σ}_{min} i^{σ}_{max} i^{ϵ}_{min} i^{ϵ}_{max} ermittelt schließlich die endgültigen Schätzwerte $\bar{\xi}_{\sigma}$ und $\bar{\xi}_{\epsilon}$ sowie das Verhältnis $\bar{\xi}_{\sigma}/\bar{\xi}_{\epsilon}$ und bestimmt mit verschiedenen Methoden die zugehörigen Fehler. Wenn statt der Routine `error()` die Funktion `error_dj()` aufgerufen wird, erfolgt die Berechnung der Mittelwerte mit Hilfe von *Double Jackknife Bias Corrected Estimators.*

Die im Programm `analyse.c` aufgerufenen Routinen `lubksb()` und `ludcmp()` dienen zur Invertierung der Kovarianzmatrix und sind vollständig [74] entnommen.

```
/***********/
/* ising.c */
/***********/

/*         (n+1)d Ising-Modell        */
/* Berechnung von Korrelationslaengen */
/*    fuer Spin und Energiedichte     */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <sys/times.h>
#include <signal.h>
#include <string.h>

#define BC_3D    0.2216544
#define DBC_3D   0.0000003
#define BC_2D    0.440686793509771
#define TEST     0.4406

#define DIM      3
#define VOL      30*30*114
#define LMAX     114             /* Muss dem letzten Eintrag in l entsprechen und gerade
 sein*/
#define INIT     1               /* 1: random, 0: ordered  */
#define BETA     BC_3D
#define DBETA    DBC_3D
#define UPDATE   'w'             /* Swendsen-Wang       = 's'  */
                                 /* Single-Cluster      = 'w'  */
                                 /* Metropolis          = 'm'  */
                                 /* Heatbath            = 'h'  */
                                 /* Hybrid (Wolf/Metro) = 'y'  */
#define EQUI     500
#define MEAS     BINS*8192
#define EMPTY    1
#define WOLFF    140
#define SHORT    1
#define RAN      get_r250()
#define SEED     32415627
#define NRAND    10000
#define BINS     32              /* Muss Teiler von MEAS sein */
#define EMSERIES 0
#define MAGHIST  0
#define CORR     1               /* 1: gemittelt, 2: direkt, 3: improved estimators */
#define STATUS   0
#define APPEND   0
#define REWEIGHT 1               /* nur fuer CORR==1 */

short l[DIM]  = {30,30,114};                       /* letzter Eintrag: "lange z-Richtung
" */
short rb[DIM] = {-1,-1,1};                         /* Randbedingungen: +1 periodisch,
  */
                                                   /*                  -1 antiperiodisch
, */
                                                   /*                   0 frei
  */


char *path="/scratch/anderson_1/weigel/diplom/3DIsing/3D30x114a_2";
char file[100];
char str[10];
double boltz, boltz_me[2*DIM+1], boltz_hb[4*DIM+1];
double *rannum;

void init_vector_random_generator(int iseed,int nrand);
void vector_random_generator(int nrand,double random_numbers[]);
void write_random_generator();
void read_random_generator();
double get_r250();

main()
{
  extern short l[DIM];
  short *s, **bound;
  int dummy;
  char **bond, *in_clus;
  int **nn, *next;
  int *iehis, *imhis;
  int sk[LMAX], ek[LMAX];
  double skorr[LMAX], ekorr[LMAX];
  double skorrm[LMAX], ekorrm[LMAX];
  double skorrp[LMAX], ekorrp[LMAX];
  double snorm, enorm;
  double rewm, rewp;
  double *xmlist, xmi;
  double acc_av, nc_tot, av_clus, av_clus2, av_e, av_e2, av_e4, av_ma, av_m, av_m2, av_m4
, e, xm, est_ave2;
  double av_e_old,av_e2_old,av_e4_old,av_m_old,av_ma_old,av_m2_old,av_m4_old,av_nc_old,av
_clus_old,av_clus2_old,nc_tot_old;
  int bins_old,lmax_old,meas_old;
  int i, k, kk, j, ie, m, s_time;
  int nc, ic, ic2;
  int sh, copy;
  short nn_sum, *ee;
  double c, sus, chi, binder, av_nc, chi_imp;
  double metro();
  struct tms u_start, u_end;
  FILE *fp1,*fp2,*fp3,*fp4,*fp5,*fp6,*fp7,*fp;

/* Initialisierungen */

  s_time=time(NULL);
  times(&u_start);
  init_vector_random_generator(SEED,NRAND);
  rannum=calloc(NRAND,sizeof(double));
  if(rannum==NULL) {
    printf("\nising: couldn't allocate memory\n");
    exit(1);
  }
  if(APPEND) {
    strcat(strcpy(file,path),"ran.ini");
    read_random_generator(file);
  }
  vector_random_generator(NRAND,rannum);

  sh=(UPDATE=='y') ? SHORT : 1;

  for(j=0; j<LMAX; j++) {
    skorr[j]=ekorr[j]=0.0;
    skorrm[j]=ekorrm[j]=0.0;
    skorrp[j]=ekorrp[j]=0.0;
  }

  if((nn=calloc(VOL, sizeof(int *)))==NULL) {
    printf("Couldn't allocate memory!");
    exit(1);
  }
  for(i=0; i<VOL; i++)
    if((nn[i]=calloc(2*DIM, sizeof(int)))==NULL) {
        printf("Couldn't allocate memory!");
        exit(1);
    }
  if((bound=calloc(VOL, sizeof(short *)))==NULL) {
```

```
    printf("Couldn’t allocate memory!");
    exit(1);
  }
  for(i=0; i<VOL; i++)
    if((bound[i]=calloc(2*DIM, sizeof(short)))==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }
  s=calloc(VOL, sizeof(short));

  if(MAGHIST) {
    iehis=calloc(2*DIM*VOL+1, sizeof(int));
    imhis=calloc(2*VOL+1, sizeof(int));
    xmlist=calloc((2*DIM*VOL+i)*5, sizeof(double));
    if(iehis==NULL || imhis==NULL || xmlist==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }
  }

  av_e_old=av_e2_old=av_e4_old=av_ma_old=av_m_old=av_m2_old=av_m4_old=av_nc_old=av_clus_
old=av_clus2_old=nc_tot_old=0.0;
  bins_old=lmax_old=meas_old=0.0;
  copy=0;

  if(APPEND) {
    strcat(strcpy(file,path),"conf.ini");
    fp=fopen(file,"r");
    fscanf(fp,"%d\t%d\t%d\t%d\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\n",
&lmax_old,&bins_old,&copy,&meas_old,&av_e_old,&av_e2_old,&av_e4_old,&av_ma_old,&av_m_old
,&av_m2_old,&av_m4_old,&av_nc_old,&av_clus_old,&av_clus2_old,&nc_tot_old);
      for(k=0; k<VOL; k++) {
        fscanf(fp,"%d\n",&dummy);
        s[k]=(short)(dummy);
      }
    fclose(fp);
  }
  else {
    for(k=0; k<VOL; k++)
      *(s+k)=(INIT) ? 2*(RAN<.5)-1 : 1;
  }

  action();

  if(UPDATE==’w’ || UPDATE==’y’) nc=1;
  if(UPDATE==’s’ || UPDATE==’w’ || UPDATE==’y’) {
    in_clus=calloc(VOL, sizeof(char));
    next=calloc(VOL, sizeof(int));
    if((bond=calloc(VOL, sizeof(char *)))==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }
    for(i=0; i<VOL; i++)
      if((bond[i]=calloc(2*DIM, sizeof(short)))==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    for(i=0; i<VOL; i++)
      in_clus[i]=0;
  }

  if(s==NULL || ((in_clus==NULL || next==NULL) && (UPDATE==’s’ || UPDATE==’w’ || UPDATE=
=’y’))) {
    printf("Couldn’t allocate memory!");
    exit(1);
  }
```

```
if(MAGHIST)
  clear_his(iehis, imhis, xmlist);

ic=0;
acc_av=nc_tot=av_clus=av_clus2=av_e=av_e2=av_e4=av_ma=av_m=av_m2=av_m4=0.0;

nn_table(nn,bound);

if(EMSERIES) {
  strcat(strcpy(file,path),"e_series.plo");
  fp1=fopen(file,"w");
  fprintf(fp1,"%d\t%d\n",MEAS,EMPTY);
}

if(CORR) {
  ee=calloc(VOL,sizeof(short));
  if(ee==NULL) {
    printf("Couldn’t allocate memory!");
    exit(1);
  }
  sprintf(str,"%d",copy);
  strcat(strcpy(file,path),str);
  strcat(file,"s_korr.ser");
  fp2=fopen(file,"w");
  sprintf(str,"%d",copy);
  strcat(strcpy(file,path),str);
  strcat(file,"e_korr.ser");
  fp3=fopen(file,"w");
  if(REWEIGHT) {
    sprintf(str,"%d",copy);
    strcat(strcpy(file,path),str);
    strcat(file,"ms_korr.ser");
    fp4=fopen(file,"w");
    sprintf(str,"%d",copy);
    strcat(strcpy(file,path),str);
    strcat(file,"me_korr.ser");
    fp5=fopen(file,"w");
    sprintf(str,"%d",copy);
    strcat(strcpy(file,path),str);
    strcat(file,"ps_korr.ser");
    fp6=fopen(file,"w");
    sprintf(str,"%d",copy);
    strcat(strcpy(file,path),str);
    strcat(file,"pe_korr.ser");
    fp7=fopen(file,"w");
  }
}

printf("\n\n***   Ising-Modell auf Geometrie S1xS1x...xR   ***\n");
printf("\n\nUpdate:                  %c\n", UPDATE);
printf("Dimension:               %d\nAbmessungen:              ", DIM);
for(i=0; i<DIM; i++)
  printf("(x%d=%4d)    ", i+1, l[i]);
printf("\nRandbedingungen:          ");
for(i=0; i<DIM; i++)
  printf("  %2d         ", rb[i]);
printf("\nTemperatur Beta:         %.15lf\n", BETA);
printf("Aequilibrierungs-Sweeps: %d/%d\n", EQUI*(1-APPEND), EQUI);
printf("Produktions-Sweeps:      %d/%d\n", MEAS, MEAS+meas_old);
printf("’Leere’ Sweeps:          %d\n", EMPTY);
if (UPDATE==’w’)
  printf("1 Sweep entspricht:      %d Wolff Updates\n",WOLFF);
if (UPDATE==’y’)
  printf("1 Metropolis auf:        %d Wolf Updates\n", SHORT);
printf("Startkonfiguration:      %d\t\t(1: random, 0: ordered)\n", INIT);
printf("Zufallsgenerator:        modifizierter R250/103\n");
if(CORR) {
```

```
    if(CORR==1)
      printf("Korrelationsmethode:     gemittelt\n");
    else if(CORR==2)
      printf("Korrelationsmethode:     direkt\n");
    else if(CORR==3)
      printf("Korrelationsmethode:     improved estimators\n");
    printf("Anzahl der Bins:         %d/%d\n", BINS, BINS+bins_old);
    printf("Werte pro Bin:           %d\n\n", MEAS/BINS);
  }
  else
    printf("\n");

  if(STATUS)
    printf("\n\nMessung #00000000");

/* Aequilibrierung */

  for(i=0; i<EQUI*EMPTY*WOLFF*(1-APPEND); i++) {
#if UPDATE=='m'
    metro(s,nn,bound);
#elif UPDATE=='h'
    heat(s,nn,bound);
#elif UPDATE=='s'
    swendsen(s,nn,bound,bond,in_clus,next,&nc,&ic,&ic2);
#elif UPDATE=='w'
    single(s,nn,bound,bond,in_clus,next,&ic);
#elif UPDATE=='y'
    for(j=0; j<SHORT; j++)
        single(s,nn,bound,bond,in_clus,next,&ic);
      metro(s,nn,bound);
#endif
  }

/* Messungs-Sweeps */

  for(i=0; i<MEAS; i++) {
    for(k=0; k<EMPTY*WOLFF; k++) {
#if UPDATE=='m'
      acc_av+=metro(s,nn,bound);
#elif UPDATE=='h'
      heat(s,nn,bound);
#elif UPDATE=='s'
      swendsen(s,nn,bound,bond,in_clus,next,&nc,&ic,&ic2);
#elif UPDATE=='w'
      single(s,nn,bound,bond,in_clus,next,&ic);
#elif UPDATE=='y'
      for(j=0; j<SHORT; j++)
        single(s,nn,bound,bond,in_clus,next,&ic);
      acc_av+=metro(s,nn,bound);
#endif
    }

/* Messungen */

#if CORR
#if CORR==2
    ie=e_korr2(s,nn,bound,ek,ee);
    m=s_korr2(s,sk);
    for(j=0; j<LMAX; j++) {
      skorr[j]+=sk[j]/((double)((LMAX-j)*(VOL/LMAX)));
      ekorr[j]+=ek[j]/((double)((LMAX-j)*(VOL/LMAX)));
    }
#elif CORR==1
    ie=e_korr(s,nn,bound,ek);
    m=s_korr(s,sk);
#if REWEIGHT
    rewm=exp(-DBETA*ie);
    rewp=exp(DBETA*ie);
#endif
    for(j=0; j<LMAX; j++) {
      snorm=sk[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));
      enorm=ek[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));
      skorr[j]+=snorm;
      ekorr[j]+=enorm;
#if REWEIGHT
      skorrm[j]+=snorm*rewm;
      ekorrm[j]+=enorm*rewm;
      skorrp[j]+=snorm*rewp;
      ekorrp[j]+=enorm*rewp;
#endif
    }
#elif CORR==3
    m=s_korr3(s,skorr,next,&ic);
    ie=e_korr(s,nn,bound,ek);
    for(j=0; j<LMAX; j++)
      ekorr[j]+=ek[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));

#endif
#if (CORR==1 || CORR==2)
    if(!((i+1)%(MEAS/BINS))) {
      fprintf(fp2,"%.15lf\t",skorr[0]/(MEAS/BINS));
      fprintf(fp3,"%.15lf\t",ekorr[0]/(MEAS/BINS));
      for(j=1; j<LMAX/2; j++) {
        fprintf(fp2,"%.15lf\t",(skorr[j]*(LMAX-j)+skorr[LMAX-j]*j)/(MEAS/BINS*LMAX));
        fprintf(fp3,"%.15lf\t",(ekorr[j]*(LMAX-j)+ekorr[LMAX-j]*j)/(MEAS/BINS*LMAX));
      }
      for(j=0; j<LMAX; j++) {
        skorr[j]=0.0;
        ekorr[j]=0.0;
      }
      fprintf(fp2,"\n");
      fprintf(fp3,"\n");
#if REWEIGHT
      fprintf(fp4,"%.15lf\t",skorrm[0]/(MEAS/BINS));
      fprintf(fp5,"%.15lf\t",ekorrm[0]/(MEAS/BINS));
      for(j=1; j<LMAX/2; j++) {
        fprintf(fp4,"%.15lf\t",(skorrm[j]*(LMAX-j)+skorrm[LMAX-j]*j)/(MEAS/BINS*LMAX));
        fprintf(fp5,"%.15lf\t",(ekorrm[j]*(LMAX-j)+ekorrm[LMAX-j]*j)/(MEAS/BINS*LMAX));
      }
      for(j=0; j<LMAX; j++) {
        skorrm[j]=0.0;
        ekorrm[j]=0.0;
      }
      fprintf(fp4,"\n");
      fprintf(fp5,"\n");
      fprintf(fp6,"%.15lf\t",skorrp[0]/(MEAS/BINS));
      fprintf(fp7,"%.15lf\t",ekorrp[0]/(MEAS/BINS));
      for(j=1; j<LMAX/2; j++) {
        fprintf(fp6,"%.15lf\t",(skorrp[j]*(LMAX-j)+skorrp[LMAX-j]*j)/(MEAS/BINS*LMAX));
        fprintf(fp7,"%.15lf\t",(ekorrp[j]*(LMAX-j)+ekorrp[LMAX-j]*j)/(MEAS/BINS*LMAX));
      }
      for(j=0; j<LMAX; j++) {
        skorrp[j]=0.0;
        ekorrp[j]=0.0;
      }
      fprintf(fp6,"\n");
      fprintf(fp7,"\n");
#endif
    }
#elif CORR==3
    if(!((i+1)%(MEAS/BINS))) {
      fprintf(fp2,"%.15lf\t",skorr[0]/(MEAS/BINS));
      fprintf(fp3,"%.15lf\t",ekorr[0]/(MEAS/BINS));
      for(j=1; j<LMAX/2; j++) {
```

```
        fprintf(fp2,"%.15lf\t",skorr[j]/(MEAS/BINS));
        fprintf(fp3,"%.15lf\t",(ekorr[j]*(LMAX-j)+ekorr[LMAX-j]*j)/(MEAS/BINS*LMAX));
      }
      for(j=0; j<LMAX; j++) {
        skorr[j]=0.0;
        ekorr[j]=0.0;
      }
      fprintf(fp2,"\n");
      fprintf(fp3,"\n");
    }
#endif
#else
    ie=m=0;
    for(kk=0; kk<VOL; kk++) {
      m+=*(s+kk);
      nn_sum=0;
      for(j=0; j<2*DIM; j+=2)
        nn_sum+=bound[kk][j]**(s+nn[kk][j]);
      ie+=*(s+kk)*nn_sum;
    }
#endif

    xm=m/((double)VOL);
    e=ie/((double)VOL);

#if STATUS
    printf("\b\b\b\b\b\b\b\b%8d",i);
#endif

/* Histogramm-Update */

#if MAGHIST
    ++iehis[ie+VOL*DIM];
    ++imhis[m+VOL];
    xmlist[ie+VOL*DIM][0]+=fabs(xm);
    xmi=1;
    for(j=1; j<5; j++) {
      xmi*=xm;
      xmlist[ie+VOL*DIM][j]+=xmi;
    }
#endif

#if EMSERIES
    fprintf(fp1, "%d\t%d\t%d\n", (i+1)*EMPTY*sh, ie, m);
#endif

    av_e+=e;
    av_e2+=e*e;
    av_e4+=e*e*e*e;

    av_ma+=fabs(xm);
    av_m+=xm;
    av_m2+=xm*xm;
    av_m4+=xm*xm*xm*xm;

    nc_tot+=nc;
    av_clus+=ic;
    av_clus2+=ic2;

  }

/* Bildschirmausgabe der Ergebnisse */

  if(UPDATE==’m’ || UPDATE==’y’)
    printf("\nacc     = %10.10lf\n", acc_av/(MEAS*EMPTY*SHORT));
  av_e=(av_e+av_e_old*meas_old)/(MEAS+meas_old);
  av_e2=(av_e2+av_e2_old*meas_old)/(MEAS+meas_old);
  av_e4=(av_e4+av_e4_old*meas_old)/(MEAS+meas_old);
  c=BETA*BETA*VOL*(av_e2-av_e*av_e);

  av_ma=(av_ma+av_ma_old*meas_old)/(MEAS+meas_old);
  av_m=(av_m+av_m_old*meas_old)/(MEAS+meas_old);
  av_m2=(av_m2+av_m2_old*meas_old)/(MEAS+meas_old);
  av_m4=(av_m4+av_m4_old*meas_old)/(MEAS+meas_old);

  sus=VOL*(av_m2-av_ma*av_ma);
  chi=VOL*av_m2;
  binder=1-av_m4/(3.0*av_m2*av_m2);

  av_nc=(nc_tot+av_nc_old*meas_old)/(MEAS+meas_old);
  av_clus=(av_clus+av_clus_old*nc_tot_old)/(nc_tot+nc_tot_old);
  av_clus2=(av_clus2+av_clus2_old*nc_tot_old)/(nc_tot+nc_tot_old);
  switch(UPDATE) {
    case ’s’: chi_imp=av_clus2/av_clus;
              break;
    case ’w’: chi_imp=av_clus;
              break;
    case ’y’: chi_imp=av_clus;
              break;
    default:  chi_imp=0.0;
              break;
  }

  printf("nc       = %10.10lf\n", av_nc);
  printf("clus     = %10.10lf\n", av_clus);
  printf("vol/clus = %10.10lf\n", VOL/av_clus);
  printf("-e       = %10.10lf\n", av_e);
  printf("e+d      = %10.10lf\n", DIM-av_e);
  printf("c        = %10.10lf\n", c);
  printf("abs(m)   = %10.10lf\n", av_ma);
  printf("m        = %10.10lf\n", av_m);
  printf("m2       = %10.10lf\n", av_m2);
  printf("sus      = %10.10lf\n", sus);
  printf("chi      = %10.10lf\n", chi);
  printf("chi_imp  = %10.10lf\n", chi_imp);
  printf("binder   = %10.10lf\n", binder);

/* Histogramme in Dateien schreiben */

  if(MAGHIST)
    put_his(iehis, imhis, xmlist);

  if (CORR) {
    fclose(fp2);
    fclose(fp3);
    if(REWEIGHT) {
      fclose(fp4);
      fclose(fp5);
      fclose(fp6);
      fclose(fp7);
    }
  }

  times(&u_end);
  printf("\nSystemzeit: %ds\n", time(NULL)-s_time);
  printf("CPU-Zeit:   %ds\n", (int)((u_end.tms_utime-u_start.tms_utime))/100);

/* Zustands-dump */

  copy++;
  strcat(strcpy(file,path),"conf.ini");
  fp=fopen(file,"w");
  fprintf(fp,"%d\t%d\t%d\t%d\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.1
5lf\t%.15lf\t%.15lf\t%.15lf\n",LMAX,BINS+bins_old,copy,MEAS+meas_old,av_e,av_e2,av_e4,av_
```

```
ma,av_m,av_m2,av_m4,av_nc,av_clus,av_clus2,nc_tot);
  for(k=0; k<VOL; k++)
    fprintf(fp,"%d\n",(int)(s[k]));
  fclose(fp);
  strcat(strcpy(file,path),"ran.ini");
  write_random_generator(file);

/* Aufraeumen */

  free(rannum);
  free(s);
  for(i=0; i<VOL; i++) {
    free(bound[i]);
    free(nn[i]);
  }
  free(bound);
  free(nn);
  if(UPDATE==’s’ || UPDATE==’w’ || UPDATE==’y’) {
    for(i=0; i<VOL; i++)
      free(bond[i]);
    free(bond);
    free(in_clus);
    free(next);
  }

  if(EMSERIES)
    fclose(fp1);

  if(MAGHIST) {
    free(iehis);
    free(imhis);
    free(xmlist);
  }

  return(0);
}

nn_table(nn,bound)
int **nn;
short **bound;
{
  int i, id, isf[DIM][LMAX], isb[DIM][LMAX], loff[DIM], k, kk, ind;
  short sigf[DIM][LMAX], sigb[DIM][LMAX];
  extern short l[DIM], rb[DIM];

  for(id=0; id<DIM; id++) {
    for(i=0; i<l[id]; i++) {
      isf[id][i]=1;
      isb[id][i]=-1;
      sigf[id][i]=1;
      sigb[id][i]=1;
    }
    isf[id][l[id]-1]=1-l[id];               /* Randbedingungen */
    isb[id][0]=l[id]-1;
    sigf[id][l[id]-1]=rb[id];
    sigb[id][0]=rb[id];
  }

  for(id=loff[0]=1; id<DIM; id++)         /* Offsets fuer die hoeheren Raumdimensionen *
/
    loff[id]=loff[id-1]*l[id-1];

  for(k=0; k<VOL; k++) {
    kk=k;
    for(id=DIM-1; id>=0; id--) {
      ind=kk/loff[id];
      kk-=ind*loff[id];
      nn[k][2*id]             = k + loff[id]*isf[id][ind];   /* positive Koordinatenricht
ungen */
      nn[k][2*id+1]           = k + loff[id]*isb[id][ind];   /* negative Koordinatenricht
ungen */
      bound[k][2*id]          = sigf[id][ind];
      bound[k][2*id+1]        = sigb[id][ind];
    }
  }
}

clear_his(iehis, imhis, xmlist)
int iehis[], imhis[];
double xmlist[][5];
{
  int i, j;

  for(i=0; i<2*VOL+1; imhis[i++]=0);
  for(i=0; i<2*DIM*VOL+1; i++) {
    iehis[i]=0;
    for(j=0; j<5; xmlist[i][j++]=0);
  }
}

action()
{
  extern double boltz, boltz_me[2*DIM+1], boltz_hb[4*DIM+1];
  int i;

  boltz=exp(-2*BETA);
  for(i=1; i<=2*DIM; i++)
    boltz_me[i]=exp(-2*BETA*i);
  for(i=0; i<=4*DIM; i++)
    boltz_hb[i]=1.0/(1.0+exp(2*BETA*(i-2*DIM)));
}

double metro(s,nn,bound)
int **nn;
short *s, **bound;
{
  int acc, i, k;
  short nn_sum, ide;

  acc=0;
  for(k=0; k<VOL; k++) {
    nn_sum=0;
    for(i=0; i<2*DIM; i++)
      nn_sum+=bound[k][i]**(s+nn[k][i]);
    ide=*(s+k)*nn_sum;
    if(ide <= 0) {
      *(s+k)*=-1;
      acc++;
    }
    else if(RAN < boltz_me[ide]) {
      *(s+k)*=-1;
      acc++;
    }
  }
  return(acc/((double)VOL));
}

heat(s,nn,bound)
int **nn;
short *s, **bound;
{
  int i, k;
  short nn_sum;
```

```
  for(k=0; k<VOL; k++) {
    nn_sum=0;
    for(i=0; i<2*DIM; i++)
      nn_sum+=bound[k][i]**(s+nn[k][i]);
    if(RAN < boltz_hb[nn_sum+2*DIM])
      *(s+k)=-1;
    else
      *(s+k)=+1;
  }
}

swendsen(s,nn,bound,bond,in_clus,next,nc,ic_tot,ic2_tot)
int **nn, *next, *nc, *ic_tot, *ic2_tot;
short *s, **bound;
char **bond, *in_clus;
{
  int i, k1, k0, clus_leng, tested, k, flip;

/* Aktive bonds markieren */

  for(k0=0; k0<VOL; k0++) {
    *(in_clus+k0)=0;
    for(i=0; i<2*DIM; i+=2) {
      k1=nn[k0][i];
      if(*(s+k0) != bound[k0][i]**(s+k1))
        bond[k0][i]=0;
      else if(RAN<boltz)
        bond[k0][i]=0;
      else
        bond[k0][i]=1;
    bond[k1][i+1]=bond[k0][i];
    }
  }

  *nc=*ic_tot=*ic2_tot=0;

  for(k=0; k<VOL; k++) {
    if(!*(in_clus+k)) {
      flip=2*(RAN<.5)-1;
      (*nc)++;
      clus_leng=tested=0;
      *next=k;
      *(in_clus+k)=1;

      while(tested<=clus_leng) {
        k0=*(next+tested);
        *(s+k0)*=flip;
        for(i=0; i<2*DIM; i++) {
          k1=nn[k0][i];
          if((!*(in_clus+k1)) && bond[k0][i]) {
            clus_leng++;
            *(next+clus_leng)=k1;
            *(in_clus+k1)=1;
          }
        }
        tested++;
      }

      clus_leng++;
      *ic_tot+=clus_leng;
      *ic2_tot+=clus_leng*clus_leng;
    }
  }
}

single(s,nn,bound,bond,in_clus,next,clus_leng)
int **nn, *next, *clus_leng;
short *s, **bound;
char **bond, *in_clus;
{
  int i, k0, k1, tested, length;

  for(i=0; i<*clus_leng; i++)
    *(in_clus+*(next+i))=0;

  k0=(int)(RAN*VOL);
  if(k0>=VOL)
    k0=VOL-1;

  *clus_leng=tested=0;
  *next=k0;
  *(in_clus+k0)=1;

  while(tested<=*clus_leng) {
    k0=*(next+tested);
    *(s+k0)*=-1;
    for(i=0; i<2*DIM; i++) {
      k1=nn[k0][i];
      if((!(*(in_clus+k1))) && ((*(s+k1))!=(bound[k0][i]**(s+k0))) && (RAN>boltz)) {
        (*clus_leng)++;
        *(in_clus+k1)=1;
        *(next+*clus_leng)=k1;
      }
    }
    tested++;
  }
  (*clus_leng)++;
}

put_his(iehis, imhis, xmlist)
int iehis[], imhis[];
double xmlist[][5];
{
  int j, k, ifirst, ilast, his[2*DIM*VOL+1];
  FILE *fp1, *fp2, *fp3;

/* Ersten und letzten Eintrag bestimmen */

  for(ifirst=0; (! iehis[ifirst]) && (ifirst <= 2*VOL*DIM); ifirst++);
  for(ilast=2*VOL*DIM; (! iehis[ilast]) && (ilast >= 0); ilast--);


/* Normierung des Histogramms */

  for(k=ifirst; k<=ilast; k++)
    his[k]=VOL*iehis[k]/MEAS;

/* Histogramm schreiben */

  strcat(strcpy(file,path),"ehis_b0.plo");
  fp1=(APPEND) ? fopen(file,"a") : fopen(file,"w");
  strcat(strcpy(file,path),"malis_b0.plo");
  fp2=(APPEND) ? fopen(file,"a") : fopen(file,"w");
  strcat(strcpy(file,path),"m2lis_b0.plo");
  fp3=(APPEND) ? fopen(file,"a") : fopen(file,"w");

  for(k=ifirst; k<=ilast; k++)
    if(iehis[k]) {
      fprintf(fp1, "%lf\t%lf\n", k/((double)VOL), his[k]);
      fprintf(fp2, "%lf\t%lf\n", k/((double)VOL), xmlist[k][0]/iehis[k]);
      fprintf(fp3, "%lf\t%lf\n", k/((double)VOL), xmlist[k][0]/iehis[k]);
    }
  fclose(fp1);
  fclose(fp2);
```

```
{
  int sk[LMAX];
  int i, k0, k1;
  int m;

  m=0;
  for(i=0; i<LMAX; i++) {
    sk[i]=0;
    for(k0=0; k0<VOL/LMAX; k0++)
      m+=s[i*VOL/LMAX+k0];
  }

  for(k0=0; k0<*clus_leng; k0++)
    for(k1=0; k1<=k0; k1++)
      /* if((next[k0]%(VOL/LMAX))==(next[k1]%(VOL/LMAX))) */
        sk[abs(next[k0]/(VOL/LMAX)-next[k1]/(VOL/LMAX))]++;

  for(i=0; i<LMAX; i++)
    skorr[i]+=sk[i]/((double)(*clus_leng));

  return(m);
}

double get_r250()
{
  static int count;

  if(count>=NRAND) {
    vector_random_generator(NRAND,rannum);
    count=0;
  }

  return(*(rannum+count++));
}
```

```c
/**********/
/*  xy.c  */
/**********/


/*         (n+1)d xy-Modell        */
/* Berechnung von Korrelationslaengen */
/*    fuer Spin und Energiedichte     */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <sys/times.h>
#include <signal.h>
#include <string.h>

#define BC_3D    0.4541670
#define TEST     0.40
#define BETANUM  4

#define DIM      3
#define VOL      17*17*204
#define LMAX     204            /* Muss dem letzten Eintrag in l entsprechen und gerade sei
n*/
#define INIT     1              /* 1: random, 0: ordered  */
#define BETA     BC_3D
#define UPDATE   'w'            /* Swendsen-Wang       = 's'  */
                                /* Single-Cluster      = 'w'  */
                                /* Metropolis          = 'm'  */
                                /* Hybrid (Wolf/Metro) = 'y'  */
#define EQUI     500
#define MEAS     BINS*8192
#define EMPTY    1
#define WOLFF    430
#define SHORT    1
#define RAN      get_r250()
#define SEED     01415627
#define NRAND    10000
#define BINS     10             /* Muss Teiler von MEAS sein */
#define EMSERIES 0
#define CORR     1              /* 1: gemittelt, 2: direkt, 3: improved estimators */
#define STATUS   0
#define APPEND   0
#define REWEIGHT 1              /* nur fuer CORR==1 */
#define CONFPLO  0              /* 1: 2D Schnitt y=0; 2: Mittelung ueber x un y; f"ur DIM=3
 */


short l[DIM]  = {17,17,204};                      /* letzter Eintrag: "lange z-Richtung" */

short rb[DIM] = {1,1,1};                          /* Randbedingungen: +1 periodisch,     */
                                                  /*                  -1 antiperiodisch, */
                                                  /*                   0 frei            */

double beta_list[BETANUM] = {0.4541638,0.4541702,0.45406,0.45421};

char *path="/home/weigel/xy/data/3D17x204p";
char file[100];
char str1[10], str2[10];
double *rannum;

void init_vector_random_generator(int iseed,int nrand);
void vector_random_generator(int nrand,double random_numbers[]);
void write_random_generator();
void read_random_generator();
double get_r250();

main()
{
  extern short l[DIM];
  short **bound;
  double *s;
  double dummy_x,dummy_y,dummy;
  char **bond, *in_clus;
  int **nn, *next;
  double sk[LMAX], ek[LMAX];
  double skorr[LMAX], ekorr[LMAX];
  double skorr_rew[BETANUM][LMAX], ekorr_rew[BETANUM][LMAX];
  double snorm, enorm;
  double rew[BETANUM];
  double acc_av, nc_tot, av_clus, av_clus2, av_e, av_e2, av_e4;
  double av_ma_x, av_ma_y, av_m_x, av_m_y, av_m2, av_m4, est_ave2;
  double av_e_old,av_e2_old,av_e4_old;
  double av_m_x_old,av_m_y_old,av_ma_x_old,av_ma_y_old,av_m2_old,av_m4_old,av_nc_old,av_clus
_old,av_clus2_old,nc_tot_old;
  int bins_old,lmax_old,meas_old;
  double e, m_x, m_y;
  int i, k, kk, j, s_time;
  int nc, ic, ic2;
  int sh, copy;
  double *ee;
  double nn_sum_x, nn_sum_y, phi;
  double c, sus, chi, binder, av_nc, chi_imp;
  double metro(), e_korr(), ekorr2();
  struct tms u_start, u_end;
  FILE *fp1,*fp2,*fp3,*fp;
  FILE *fprs[BETANUM], *fpre[BETANUM];

/* Initialisierungen */

  s_time=time(NULL);
  times(&u_start);
  init_vector_random_generator(SEED,NRAND);
  rannum=calloc(NRAND,sizeof(double));
  if(rannum==NULL) {
    printf("\nising: couldn't allocate memory\n");
    exit(1);
  }
  if(APPEND) {
    strcat(strcpy(file,path),"ran.ini");
    read_random_generator(file);
  }
  vector_random_generator(NRAND,rannum);

  sh=(UPDATE=='y') ? SHORT : 1;

  for(j=0; j<LMAX; j++) {
    skorr[j]=ekorr[j]=0.0;
    for(kk=0; kk<BETANUM; kk++)
      skorr_rew[kk][j]=ekorr_rew[kk][j]=0.0;
  }

  if((nn=calloc(VOL, sizeof(int *)))==NULL) {
    printf("Couldn't allocate memory!");
    exit(1);
  }
  for(i=0; i<VOL; i++)
    if((nn[i]=calloc(2*DIM, sizeof(int)))==NULL) {
        printf("Couldn't allocate memory!");
        exit(1);
    }
  if((bound=calloc(VOL, sizeof(short *)))==NULL) {
```

```
    printf("Couldn't allocate memory!");
    exit(1);
  }
  for(i=0; i<VOL; i++)
    if((bound[i]=calloc(2*DIM, sizeof(short)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
  s=calloc(2*VOL, sizeof(double));

  av_e_old=av_e2_old=av_e4_old=av_ma_x_old=av_ma_y_old=av_m_x_old=av_m_y_old=0.0;
  av_m2_old=av_m4_old=av_nc_old=av_clus_old=av_clus2_old=nc_tot_old=0.0;
  bins_old=lmax_old=meas_old=0;
  copy=0;

  if(APPEND) {
    strcat(strcpy(file,path),"conf.ini");
    fp=fopen(file,"r");
    fscanf(fp,"%d\t%d\t%d\t%d\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%
lf\n",&lmax_old,&bins_old,&copy,&meas_old,&av_e_old,&av_e2_old,&av_e4_old,&av_ma_x_old,&av_m
a_y_old,&av_m_x_old,&av_m_y_old,&av_m2_old,&av_m4_old,&av_nc_old,&av_clus_old,&av_clus2_old,
&nc_tot_old);
      for(k=0; k<VOL; k++) {
        fscanf(fp,"%lf\t%lf\n",&dummy_x,&dummy_y);
        s[2*k]=dummy_x;
        s[2*k+1]=dummy_y;
      }
      fclose(fp);

      for(k=0; k<VOL; k++) {
        dummy_x=sqrt(s[2*k]*s[2*k]+s[2*k+1]*s[2*k+1]);
        s[2*k]/=dummy_x;
        s[2*k+1]/=dummy_x;
      }
  }
  else {
    for(k=0; k<VOL; k++) {
      phi=2*M_PI*RAN;
      s[2*k]=(INIT) ? cos(phi) : 1;
      s[2*k+1]=(INIT) ? sin(phi) : 0;
    }
  }

  if(UPDATE=='w' || UPDATE=='y') nc=1;
  if(UPDATE=='s' || UPDATE=='w' || UPDATE=='y') {
    in_clus=calloc(VOL, sizeof(char));
    next=calloc(VOL, sizeof(int));
    if((bond=calloc(VOL, sizeof(char *)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    for(i=0; i<VOL; i++)
      if((bond[i]=calloc(2*DIM, sizeof(short)))==NULL) {
        printf("Couldn't allocate memory!");
        exit(1);
      }
    for(i=0; i<VOL; i++)
      in_clus[i]=0;
  }

  if(s==NULL || ((in_clus==NULL || next==NULL) && (UPDATE=='s' || UPDATE=='w' || UPDATE=='y'
))) {
    printf("Couldn't allocate memory!");
    exit(1);
  }

  ic=0;
```

```
  acc_av=nc_tot=av_clus=av_clus2=av_e=av_e2=av_e4=av_ma_x=av_ma_y=av_m_x=av_m_y=av_m2=av_m4=
0.0;

  nn_table(nn,bound);

  if(EMSERIES) {
    strcat(strcpy(file,path),"e_series.plo");
    fp1=fopen(file,"w");
    fprintf(fp1,"%d\t%d\n",MEAS,EMPTY);
  }

  if(CORR) {
    if(CORR==2) {
      ee=calloc(VOL,sizeof(double));
      if(ee==NULL) {
        printf("Couldn't allocate memory!");
        exit(1);
      }
    }
    sprintf(str1,"0.");
    sprintf(str2,"%d",copy);
    strcat(strcpy(file,path),str1);
    strcat(file,str2);
    strcat(file,"s_korr.ser");
    fp2=fopen(file,"w");
    dummy_x=BETA;
    fwrite(&dummy_x,sizeof(double),1,fp2);
    strcat(strcpy(file,path),str1);
    strcat(file,str2);
    strcat(file,"e_korr.ser");
    fp3=fopen(file,"w");
    fwrite(&dummy_x,sizeof(double),1,fp3);
    if(REWEIGHT) {
      for(i=0; i<BETANUM; i++) {
        sprintf(str1,"%d.",i+1);
        strcat(strcpy(file,path),str1);
        strcat(file,str2);
        strcat(file,"s_korr.ser");
        fprs[i]=fopen(file,"w");
        fwrite(beta_list+i,sizeof(double),1,fprs[i]);
        strcat(strcpy(file,path),str1);
        strcat(file,str2);
        strcat(file,"e_korr.ser");
        fpre[i]=fopen(file,"w");
        fwrite(beta_list+i,sizeof(double),1,fpre[i]);
      }
    }
  }

  printf("\n\n***   XY-Modell auf Geometrie S1xS1x...xR   ***\n");
  printf("\n\nUpdate:                  %c\n", UPDATE);
  printf("Dimension:               %d\nAbmessungen:              ", DIM);
  for(i=0; i<DIM; i++)
    printf("(x%d=%4d)    ", i+1, l[i]);
  printf("\nRandbedingungen:          ");
  for(i=0; i<DIM; i++)
    printf("  %2d         ", rb[i]);
  printf("\nTemperatur Beta:         %.15lf\n", BETA);
  printf("Aequilibrierungs-Sweeps: %d/%d\n", EQUI*(1-APPEND), EQUI);
  printf("Produktions-Sweeps:      %d/%d\n", MEAS, MEAS+meas_old);
  printf("'Leere' Sweeps:          %d\n", EMPTY);
  if (UPDATE=='w')
    printf("1 Sweep entspricht:      %d Wolff Updates\n",WOLFF);
  if (UPDATE=='y')
    printf("1 Metropolis auf:        %d Wolf Updates\n", SHORT);
  printf("Startkonfiguration:      %d\t\t(1: random, 0: ordered)\n", INIT);
  printf("Zufallsgenerator:        modifizierter R250/103\n");
```

```
  if(CORR) {
    if(CORR==1)
      printf("Korrelationsmethode:     gemittelt\n");
    else if(CORR==2)
      printf("Korrelationsmethode:     direkt\n");
    else if(CORR==3)
      printf("Korrelationsmethode:     improved estimators\n");
    printf("Anzahl der Bins:         %d/%d\n", BINS, BINS+bins_old);
    printf("Werte pro Bin:           %d\n\n", MEAS/BINS);
  }
  else
    printf("\n");

  if(STATUS)
    printf("\n\nMessung #00000000");

/* Aequilibrierung */

  for(i=0; i<EQUI*EMPTY*WOLFF*(1-APPEND); i++) {
#if UPDATE=='m'
    metro(s,nn,bound);
#elif UPDATE=='s'
    swendsen(s,nn,bound,bond,in_clus,next,&nc,&ic,&ic2);
#elif UPDATE=='w'
    single(s,nn,bound,bond,in_clus,next,&ic);
#elif UPDATE=='y'
    for(j=0; j<SHORT; j++)
        single(s,nn,bound,bond,in_clus,next,&ic);
      metro(s,nn,bound);
#endif
  }

/* Messungs-Sweeps */

  for(i=0; i<MEAS; i++) {
    for(k=0; k<EMPTY*WOLFF; k++) {
#if UPDATE=='m'
      acc_av+=metro(s,nn,bound);
#elif UPDATE=='s'
      swendsen(s,nn,bound,bond,in_clus,next,&nc,&ic,&ic2);
#elif UPDATE=='w'
      single(s,nn,bound,bond,in_clus,next,&ic);
#elif UPDATE=='y'
      for(j=0; j<SHORT; j++)
        single(s,nn,bound,bond,in_clus,next,&ic);
      acc_av+=metro(s,nn,bound);
#endif
    }

/* Messungen */

#if CORR
#if CORR==2
    e=e_korr2(s,nn,bound,ek,ee);
    s_korr2(s,sk,&m_x,&m_y);
    for(j=0; j<LMAX; j++) {
      skorr[j]+=sk[j]/((double)((LMAX-j)*(VOL/LMAX)));
      ekorr[j]+=ek[j]/((double)((LMAX-j)*(VOL/LMAX)));
    }
#elif CORR==1
    e=e_korr(s,nn,bound,ek);
    s_korr(s,sk,&m_x,&m_y);
#if REWEIGHT
    for(j=0; j<BETANUM; j++)
      rew[j]=exp((beta_list[j]-BETA)*e);
#endif
    for(j=0; j<LMAX; j++) {
      snorm=sk[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));
      enorm=ek[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));
      skorr[j]+=snorm;
      ekorr[j]+=enorm;
#if REWEIGHT
      for(kk=0; kk<BETANUM; kk++) {
        skorr_rew[kk][j]+=snorm*rew[kk];
        ekorr_rew[kk][j]+=enorm*rew[kk];
      }
#endif
    }
/* #elif CORR==3 */
/*     m=s_korr3(s,skorr,next,&ic); */
/*     e=e_korr(s,nn,bound,ek); */
/*     for(j=0; j<LMAX; j++) */
/*       ekorr[j]+=ek[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX))); */

#endif
#if (CORR==1 || CORR==2)
    if(!((i+1)%(MEAS/BINS))) {
      skorr[0]/=(MEAS/BINS);
      ekorr[0]/=(MEAS/BINS);
      for(j=1; j<LMAX/2; j++) {
        skorr[j]=(skorr[j]*(LMAX-j)+skorr[LMAX-j]*j)/(MEAS/BINS*LMAX);
        ekorr[j]=(ekorr[j]*(LMAX-j)+ekorr[LMAX-j]*j)/(MEAS/BINS*LMAX);
      }
      fwrite(skorr,sizeof(double),LMAX/2,fp2);
      fwrite(ekorr,sizeof(double),LMAX/2,fp3);
      for(j=0; j<LMAX; j++) {
        skorr[j]=0.0;
        ekorr[j]=0.0;
      }
#if REWEIGHT
      for(kk=0; kk<BETANUM; kk++) {
        skorr_rew[kk][0]/=(MEAS/BINS);
        ekorr_rew[kk][0]/=(MEAS/BINS);
        for(j=1; j<LMAX/2; j++) {
          skorr_rew[kk][j]=(skorr_rew[kk][j]*(LMAX-j)+skorr_rew[kk][LMAX-j]*j)/(MEAS/BINS*LM
AX);
          ekorr_rew[kk][j]=(ekorr_rew[kk][j]*(LMAX-j)+ekorr_rew[kk][LMAX-j]*j)/(MEAS/BINS*LM
AX);
        }
        fwrite(skorr_rew[kk],sizeof(double),LMAX/2,fprs[kk]);
        fwrite(ekorr_rew[kk],sizeof(double),LMAX/2,fpre[kk]);
        for(j=0; j<LMAX; j++)
          skorr_rew[kk][j]=ekorr_rew[kk][j]=0.0;
      }
#endif
    }
/* #elif CORR==3 */
/*     if(!((i+1)%(MEAS/BINS))) { */
/*       fprintf(fp2,"%.15lf\t",skorr[0]/(MEAS/BINS)); */
/*       fprintf(fp3,"%.15lf\t",ekorr[0]/(MEAS/BINS)); */
/*       for(j=1; j<LMAX/2; j++) { */
/*      fprintf(fp2,"%.15lf\t",skorr[j]/(MEAS/BINS)); */
/*      fprintf(fp3,"%.15lf\t",(ekorr[j]*(LMAX-j)+ekorr[LMAX-j]*j)/(MEAS/BINS*LMAX)); */
/*       } */
/*       for(j=0; j<LMAX; j++) { */
/*      skorr[j]=0.0; */
/*      ekorr[j]=0.0; */
/*       } */
/*       fprintf(fp2,"\n"); */
/*       fprintf(fp3,"\n"); */
/*     } */
#endif
#else
    e=m_x=m_y=0.0;
```

```
    for(kk=0; kk<VOL; kk++) {
      m_x+=s[2*kk];
      m_y+=s[2*kk+1];
      nn_sum_x=nn_sum_y=0.0;
      for(j=0; j<2*DIM; j+=2) {
        nn_sum_x+=bound[kk][j]*s[2*nn[kk][j]];
        nn_sum_y+=bound[kk][j]*s[2*nn[kk][j]+1];
      }
      e+=s[2*kk]*nn_sum_x+s[2*kk+1]*nn_sum_y;
    }
#endif

    m_x/=(double)VOL;
    m_y/=(double)VOL;
    e/=(double)VOL;

#if STATUS
    printf("\b\b\b\b\b\b\b\b%8d",i);
#endif

#if EMSERIES
    fprintf(fp1, "%d\t%.15lf\t%.15lf\t%.15lf\n", (i+1)*EMPTY*sh, e, m_x, m_y);
#endif

    av_e+=e;
    av_e2+=e*e;
    av_e4+=e*e*e*e;

    av_ma_x+=fabs(m_x);
    av_ma_y+=fabs(m_y);
    av_m_x+=m_x;
    av_m_y+=m_y;
    av_m2+=m_x*m_x+m_y*m_y;
    av_m4+=(m_x*m_x+m_y*m_y)*(m_x*m_x+m_y*m_y);

    nc_tot+=nc;
    av_clus+=ic;
    av_clus2+=ic2;

  }

/* Bildschirmausgabe der Ergebnisse */

  if(UPDATE=='m' || UPDATE=='y')
    printf("\nacc     = %10.10lf\n", acc_av/(MEAS*EMPTY*SHORT));
  av_e=(av_e+av_e_old*meas_old)/(MEAS+meas_old);
  av_e2=(av_e2+av_e2_old*meas_old)/(MEAS+meas_old);
  av_e4=(av_e4+av_e4_old*meas_old)/(MEAS+meas_old);
  c=BETA*BETA*VOL*(av_e2-av_e*av_e);

  av_ma_x=(av_ma_x+av_ma_x_old*meas_old)/(MEAS+meas_old);
  av_ma_y=(av_ma_y+av_ma_y_old*meas_old)/(MEAS+meas_old);
  av_m_x=(av_m_x+av_m_x_old*meas_old)/(MEAS+meas_old);
  av_m_y=(av_m_y+av_m_y_old*meas_old)/(MEAS+meas_old);
  av_m2=(av_m2+av_m2_old*meas_old)/(MEAS+meas_old);
  av_m4=(av_m4+av_m4_old*meas_old)/(MEAS+meas_old);

  sus=VOL*(av_m2-(av_ma_x*av_ma_x+av_ma_y*av_ma_y));
  chi=VOL*av_m2;
  binder=1-av_m4/(3.0*av_m2*av_m2);

  av_nc=(nc_tot+av_nc_old*meas_old)/(MEAS+meas_old);
  av_clus=(av_clus+av_clus_old*nc_tot_old)/(nc_tot+nc_tot_old);
  av_clus2=(av_clus2+av_clus2_old*nc_tot_old)/(nc_tot+nc_tot_old);
  switch(UPDATE) {
    case 's': chi_imp=av_clus2/av_clus;
              break;
    case 'w': chi_imp=av_clus;
              break;
    case 'y': chi_imp=av_clus;
              break;
    default:  chi_imp=0.0;
              break;
  }

  if(UPDATE!='m') {
    printf("nc       = %10.10lf\n", av_nc);
    printf("clus     = %10.10lf\n", av_clus);
    printf("clus/vol = %10.10lf\n", av_clus/((double)VOL));
  }
  printf("-e       = %10.10lf\n", av_e);
  printf("e+d      = %10.10lf\n", DIM-av_e);
  printf("c        = %10.10lf\n", c);
  printf("abs(m)_x = %10.10lf\n", av_ma_x);
  printf("abs(m)_y = %10.10lf\n", av_ma_y);
  printf("m_x      = %10.10lf\n", av_m_x);
  printf("m_y      = %10.10lf\n", av_m_y);
  printf("m2       = %10.10lf\n", av_m2);
  printf("sus      = %10.10lf\n", sus);
  printf("chi      = %10.10lf\n", chi);
  if(UPDATE!='m')
    printf("chi_imp  = %10.10lf\n", chi_imp);
  printf("binder   = %10.10lf\n", binder);

  if (CORR) {
    fclose(fp2);
    fclose(fp3);
    if(REWEIGHT)
      for(i=0; i<BETANUM; i++) {
        fclose(fprs[i]);
        fclose(fpre[i]);
      }
  }

  times(&u_end);
  printf("\nSystemzeit: %ds\n", time(NULL)-s_time);
  printf("CPU-Zeit:   %ds\n", (int)((u_end.tms_utime-u_start.tms_utime))/100);

/* Zustands-dump */

  copy++;
  strcat(strcpy(file,path),"conf.ini");
  fp=fopen(file,"w");
  fprintf(fp,"%d\t%d\t%d\t%d\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf
\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\n",LMAX,BINS+bins_old,copy,MEAS+meas_old,av_e,av_e2
,av_e4,av_ma_x,av_ma_y,av_m_x,av_m_y,av_m2,av_m4,av_nc,av_clus,av_clus2,nc_tot);
  for(k=0; k<VOL; k++)
    fprintf(fp,"%.15lf\t%.15lf\n",s[2*k],s[2*k+1]);
  fclose(fp);
  strcat(strcpy(file,path),"ran.ini");
  write_random_generator(file);
#if CONFPLO==1
  strcat(strcpy(file,path),"conf.plo");
  fp=fopen(file,"w");
  for(k=0; k<l[0]; k++) {
    for(i=0; i<l[2]; i++)
      fprintf(fp,"%.15lf\t%.15lf\t",s[2*(i*l[0]*l[1]+k)],s[2*(i*l[0]*l[1]+k)+1]);
    fprintf(fp,"\n");
  }
  fclose(fp);
#endif
#if CONFPLO==2
  strcat(strcpy(file,path),"conf.plo");
  fp=fopen(file,"w");
```

```
  for(k=0; k<l[0]; k++) {
    for(i=0; i<l[2]; i++) {
      dummy_x=dummy_y=0;
      for(j=0; j<l[1]; j++) {
        dummy_x+=s[2*(i*l[0]*l[1]+j*l[0]+k)]/l[1];
        dummy_y+=s[2*(i*l[0]*l[1]+j*l[0]+k)+1]/l[1];
      }
      dummy=sqrt(dummy_x*dummy_x+dummy_y*dummy_y);
      dummy_x/=dummy;
      dummy_y/=dummy;
      fprintf(fp,"%.15lf\t%.15lf\t",dummy_x,dummy_y);
    }
    fprintf(fp,"\n");
  }
  fclose(fp);
#endif

/* Aufraeumen */

  if(CORR==2)
    free(ee);
  free(rannum);
  free(s);
  for(i=0; i<VOL; i++) {
    free(bound[i]);
    free(nn[i]);
  }
  free(bound);
  free(nn);
  if(UPDATE==’s’ || UPDATE==’w’ || UPDATE==’y’) {
    for(i=0; i<VOL; i++)
      free(bond[i]);
    free(bond);
    free(in_clus);
    free(next);
  }

  if(EMSERIES)
    fclose(fp1);

  return(0);
}

nn_table(nn,bound)
int **nn;
short **bound;
{
  int i, id, isf[DIM][LMAX], isb[DIM][LMAX], loff[DIM], k, kk, ind;
  short sigf[DIM][LMAX], sigb[DIM][LMAX];
  extern short l[DIM], rb[DIM];

  for(id=0; id<DIM; id++) {
    for(i=0; i<l[id]; i++) {
      isf[id][i]=1;
      isb[id][i]=-1;
      sigf[id][i]=1;
      sigb[id][i]=1;
    }
    isf[id][l[id]-1]=1-l[id];               /* Randbedingungen */
    isb[id][0]=l[id]-1;
    sigf[id][l[id]-1]=rb[id];
    sigb[id][0]=rb[id];
  }

  for(id=loff[0]=1; id<DIM; id++)         /* Offsets fuer die hoeheren Raumdimensionen */
    loff[id]=loff[id-1]*l[id-1];

  for(k=0; k<VOL; k++) {
    kk=k;
    for(id=DIM-1; id>=0; id--) {
      ind=kk/loff[id];
      kk-=ind*loff[id];
      nn[k][2*id]             = k + loff[id]*isf[id][ind];   /* positive Koordinatenrichtung
en */
      nn[k][2*id+1]           = k + loff[id]*isb[id][ind];   /* negative Koordinatenrichtung
en */
      bound[k][2*id]          = sigf[id][ind];
      bound[k][2*id+1]        = sigb[id][ind];
    }
  }
}

double metro(s,nn,bound)
int **nn;
short **bound;
double *s;
{
  int acc, i, k;
  double ide, phi;
  double nn_sum_x,nn_sum_y, s_neu_x,s_neu_y;

  acc=0;
  for(k=0; k<VOL; k++) {
    nn_sum_x=nn_sum_y=0;
    for(i=0; i<2*DIM; i++) {
      nn_sum_x+=bound[k][i]*s[2*nn[k][i]];
      nn_sum_y+=bound[k][i]*s[2*nn[k][i]+1];
    }
    phi=RAN*2*M_PI;
    s_neu_x=cos(phi);
    s_neu_y=sin(phi);
    ide=(s[2*k]-s_neu_x)*nn_sum_x+(s[2*k+1]-s_neu_y)*nn_sum_y;
    if(ide <= 0) {
      s[2*k]=s_neu_x;
      s[2*k+1]=s_neu_y;
      acc++;
    }
    else if(RAN < exp(-BETA*ide)) {
      s[2*k]=s_neu_x;
      s[2*k+1]=s_neu_y;
      acc++;
    }
  }
  return(acc/((double)VOL));
}

swendsen(s,nn,bound,bond,in_clus,next,nc,ic_tot,ic2_tot)
int **nn, *next, *nc, *ic_tot, *ic2_tot;
double *s;
short **bound;
char **bond, *in_clus;
{
  int i, k1, k0, clus_leng, tested, k, flip;
  double r_x, r_y, prod, phi;

  phi=2*M_PI*RAN;
  r_x=cos(phi);
  r_y=sin(phi);

  for(k0=0; k0<VOL; k0++) {
    *(in_clus+k0)=0;
    for(i=0; i<2*DIM; i+=2) {
      k1=nn[k0][i];
      prod=bound[k0][i]*(s[2*k0]*r_x+s[2*k0+1]*r_y)*(s[2*k1]*r_x+s[2*k1+1]*r_y);
```

```
      if(prod<=0)
        bond[k0][i]=0;
      else if(RAN<exp(-2*BETA*prod))
        bond[k0][i]=0;
      else
        bond[k0][i]=1;
    bond[k1][i+1]=bond[k0][i];
    }
  }

  *nc=*ic_tot=*ic2_tot=0;

  for(k=0; k<VOL; k++) {
    if(!*(in_clus+k)) {
      flip=(RAN<.5);
      (*nc)++;
      clus_leng=tested=0;
      *next=k;
      *(in_clus+k)=1;

      while(tested<=clus_leng) {
        k0=*(next+tested);
        if(flip) {
          prod=s[2*k0]*r_x+s[2*k0+1]*r_y;
          s[2*k0]-=2*prod*r_x;
          s[2*k0+1]-=2*prod*r_y;
        }
        for(i=0; i<2*DIM; i++) {
          k1=nn[k0][i];
          if((!*(in_clus+k1)) && bond[k0][i]) {
            clus_leng++;
            *(next+clus_leng)=k1;
            *(in_clus+k1)=1;
          }
        }
        tested++;
      }

      clus_leng++;
      *ic_tot+=clus_leng;
      *ic2_tot+=clus_leng*clus_leng;
    }
  }
}

single(s,nn,bound,bond,in_clus,next,clus_leng)
int **nn, *next, *clus_leng;
double *s;
short **bound;
char **bond, *in_clus;
{
  int i, k0, k1, tested, length;
  double r_x, r_y, prod1, prod2, phi;

  for(i=0; i<*clus_leng; i++)
    *(in_clus+*(next+i))=0;

  k0=(int)(RAN*VOL);
  if(k0>=VOL)
    k0=VOL-1;
  phi=2*M_PI*RAN;
  r_x=cos(phi);
  r_y=sin(phi);

  *clus_leng=tested=0;
  *next=k0;
  *(in_clus+k0)=1;

  while(tested<=*clus_leng) {
    k0=*(next+tested);
    prod1=s[2*k0]*r_x+s[2*k0+1]*r_y;
    for(i=0; i<2*DIM; i++) {
      k1=nn[k0][i];
      if(!(in_clus[k1])) {
        prod2=bound[k0][i]*prod1*(s[2*k1]*r_x+s[2*k1+1]*r_y);
        if(prod2>=0)
          if(RAN>exp(-2*BETA*prod2))
            {
              (*clus_leng)++;
              *(in_clus+k1)=1;
              *(next+*clus_leng)=k1;
            }
      }
    }
    s[2*k0]-=2*prod1*r_x;
    s[2*k0+1]-=2*prod1*r_y;
    tested++;
  }
  (*clus_leng)++;
}

double e_korr(s,nn,bound,ek)
double *s;
int **nn;
short **bound;
double *ek;
{
  int i, j, k;
  int factor;
  double nn_sum_x, nn_sum_y;
  double ee[LMAX], e;

  e=0.0;
   for(i=0; i<LMAX; i++) {
     ee[i]=0.0;
     factor=VOL/LMAX*i;
     for(j=0; j<VOL/LMAX; j++) {
       nn_sum_x=0.0;
       nn_sum_y=0.0;
       for(k=0; k<2*DIM-2; k+=2) {
        nn_sum_x+=bound[factor+j][k]*s[2*nn[factor+j][k]];
        nn_sum_y+=bound[factor+j][k]*s[2*nn[factor+j][k]+1];
       }
      nn_sum_x+=bound[factor+j][2*DIM-2]*s[2*nn[factor+j][2*DIM-2]];
      nn_sum_y+=bound[factor+j][2*DIM-2]*s[2*nn[factor+j][2*DIM-2]+1];
      ee[i]+=s[2*(factor+j)]*nn_sum_x+s[2*(factor+j)+1]*nn_sum_y;
     }
     e+=ee[i];
   }

   for(i=0; i<LMAX; i++) {
     ek[i]=0;
     for(j=0; j<LMAX-i; j++)
       ek[i]+=ee[j]*ee[j+i];
   }

   return(e);
}

double e_korr2(s,nn,bound,ek,ee)
double *s;
int **nn;
short **bound;
double *ek, *ee;
```

```
{
  int i, j, k, n;
  double nn_sum_x, nn_sum_y;
  double e;

  e=0.0;
  for(i=0; i<VOL; i++) {
    nn_sum_x=0.0;
    nn_sum_y=0.0;
    for(k=0; k<2*DIM; k++) {
      nn_sum_x+=bound[i][k]*s[2*nn[i][k]];
      nn_sum_y+=bound[i][k]*s[2*nn[i][k]+1];
    }
    ee[i]=(s[2*i]*nn_sum_x+s[2*i+1]*nn_sum_y)/2;
    e+=ee[i];
  }

  for(i=0; i<LMAX; i++) {
    ek[i]=0.0;
    for(j=0; j<LMAX-i; j++)
      for(n=0; n<VOL/LMAX; n++)
        ek[i]+=ee[j*VOL/LMAX+n]*ee[(j+i)*VOL/LMAX+n];
  }

  return(e);
}

s_korr(s,sk,m_x,m_y)
double *s;
double *sk, *m_x, *m_y;
{
  int i,j;
  double ss[2*LMAX];
  int factor;

  *m_x=*m_y=0.0;
  for(i=0; i<2*LMAX; i+=2) {
    ss[i]=ss[i+1]=0.0;
    factor=VOL/LMAX*i;
    for(j=0; j<2*VOL/LMAX; j+=2) {
      ss[i]+=s[factor+j];
      ss[i+1]+=s[factor+j+1];
    }
    *m_x+=ss[i];
    *m_y+=ss[i+1];
  }

  for(i=0; i<LMAX; i++) {
    sk[i]=0;
    for(j=0; j<LMAX-i; j++)
      sk[i]+=(ss[2*j]*ss[2*(j+i)])+(ss[2*j+1]*ss[2*(j+i)+1]);
  }
}

s_korr2(s,sk,m_x,m_y)
double *s;
double *sk;
double *m_x, *m_y;
{
  int i, j, n;

  *m_x=*m_y=0;
  for(n=0; n<VOL; n++) {
    *m_x+=s[2*n];
    *m_y+=s[2*n+1];
  }

  for(i=0; i<LMAX; i++) {
    sk[i]=0.0;
    for(j=0; j<LMAX-i; j++)
      for(n=0; n<VOL/LMAX; n++)
        sk[i]+=s[2*(j*VOL/LMAX+n)]*s[2*((j+i)*VOL/LMAX+n)+1];
  }
}

/* s_korr3(s,skorr,next,clus_leng,m_x,m_y) */
/* double *s; */
/* int *clus_leng, *next; */
/* double *skorr; */
/* double *m_x, *m_y; */
/* { */
/*   double sk[2*LMAX]; */
/*   int i, k0, k1; */

/*   *m_x=*m_y=0.0; */
/*   for(i=0; i<LMAX; i++) { */
/*     sk[i]=0.0; */
/*     for(k0=0; k0<VOL/LMAX; k0++) { */
/*       m_x+=s[2*(i*VOL/LMAX+k0)]; */
/*       m_y+=s[2*(i*VOL/LMAX+k0)+1]; */
/*     } */
/*   } */

/*   for (k0=0; k0<*clus_leng; k0++) */
/*     for(k1=0; k1<=k0; k1++) */
/*       if((next[k0]%(VOL/LMAX))==(next[k1]%(VOL/LMAX))) */
/*      sk[abs(next[k0]/(VOL/LMAX)-next[k1]/(VOL/LMAX))]++; */

/*   for(i=0; i<LMAX; i++) */
/*     skorr[i]+=sk[i]/((double)(*clus_leng)); */

/*   return(m); */
/* } */

double get_r250()
{
  static int count;

  if(count>=NRAND) {
    vector_random_generator(NRAND,rannum);
    count=0;
  }

  return(*(rannum+count++));
}
```

```
/************/
/* heisen.c */
/************/

/*      (n+1)d Heisenberg-Modell        */
/* Berechnung von Korrelationslaengen   */
/*    fuer Spin und Energiedichte       */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <sys/times.h>
#include <signal.h>
#include <string.h>

#include <ranvec.c>

#define BC_3D    0.693004
#define TEST     0.660
#define BETANUM  4

#define DIM      3
#define VOL      12*12*50
#define LMAX     50              /* Muss dem letzten Eintrag in l entsprechen und gerade
 sein*/
#define INIT     1               /* 1: random, 0: ordered  */
#define BETA     BC_3D
#define UPDATE   'w'             /* Swendsen-Wang       = 's'  */
                                 /* Single-Cluster      = 'w'  */
                                 /* Metropolis          = 'm'  */
                                 /* Hybrid (Wolf/Metro) = 'y'  */
#define EQUI     500
#define MEAS     BINS*4096
#define EMPTY    1
#define WOLFF    140
#define SHORT    1
#define RAN      get_r250()
#define SEED     31415627
#define NRAND    10000
#define BINS     160             /* Muss Teiler von MEAS sein */
#define EMSERIES 0
#define CORR     1               /* 1: gemittelt, 2: direkt, 3: improved estimators */
#define STATUS   0
#define APPEND   1
#define REWEIGHT 1               /* nur fuer CORR==1 */


short l[DIM]  = {12,12,50};                        /* letzter Eintrag: "lange z-Richtung
" */
short rb[DIM] = {-1,-1,1};                          /* Randbedingungen: +1 periodisch,
   */
                                                    /*                  -1 antiperiodisc
h, */
                                                    /*                   0 frei
   */

double beta_list[BETANUM] = {0.693011,0.692997,0.6929,0.69305};

char *path="/scratch/anderson_1/weigel/diplom/3Dheisen/3D12x50a";
char file[100];
char str1[10], str2[10];
double *rannum;

void init_vector_random_generator(int iseed,int nrand);
void vector_random_generator(int nrand,double random_numbers[]);
void write_random_generator();
void read_random_generator();
double get_r250();

main()
{
  extern short l[DIM];
  short **bound;
  double *s;
  double dummy_x,dummy_y,dummy_z;
  char **bond, *in_clus;
  int **nn, *next;
  double sk[LMAX], ek[LMAX];
  double skorr[LMAX], ekorr[LMAX];
  double skorr_rew[BETANUM][LMAX], ekorr_rew[BETANUM][LMAX];
  double snorm, enorm;
  double rew[BETANUM];
  double acc_av, nc_tot, av_clus, av_clus2, av_e, av_e2, av_e4;
  double av_ma_x, av_ma_y, av_ma_z, av_m_x, av_m_y, av_m_z, av_m2, av_m4, est_ave2;
  double av_e_old,av_e2_old,av_e4_old;
  double av_m_x_old,av_m_y_old,av_m_z_old,av_ma_x_old,av_ma_y_old,av_ma_z_old,av_m2_old,a
v_m4_old;
  double av_nc_old,av_clus_old,av_clus2_old,nc_tot_old;
  int bins_old,lmax_old,meas_old;
  double e, m_x, m_y, m_z;
  int i, k, kk, j, s_time;
  int nc, ic, ic2;
  int sh, copy;
  double *ee;
  double nn_sum_x, nn_sum_y, nn_sum_z, phi, theta;
  double c, sus, chi, binder, av_nc, chi_imp;
  double metro(), e_korr(), ekorr2();
  struct tms u_start, u_end;
  FILE *fp1,*fp2,*fp3,*fp;
  FILE *fprs[BETANUM], *fpre[BETANUM];

/* Initialisierungen */

  s_time=time(NULL);
  times(&u_start);
  init_vector_random_generator(SEED,NRAND);
  rannum=calloc(NRAND,sizeof(double));
  if(rannum==NULL) {
    printf("\nising: couldn't allocate memory\n");
    exit(1);
  }
  if(APPEND) {
    strcat(strcpy(file,path),"ran.ini");
    read_random_generator(file);
  }
  vector_random_generator(NRAND,rannum);

  sh=(UPDATE=='y') ? SHORT : 1;

  for(j=0; j<LMAX; j++) {
    skorr[j]=ekorr[j]=0.0;
    for(kk=0; kk<BETANUM; kk++)
      skorr_rew[kk][j]=ekorr_rew[kk][j]=0.0;
  }

  if((nn=calloc(VOL, sizeof(int *)))==NULL) {
    printf("Couldn't allocate memory!");
    exit(1);
  }
  for(i=0; i<VOL; i++)
    if((nn[i]=calloc(2*DIM, sizeof(int)))==NULL) {
        printf("Couldn't allocate memory!");
        exit(1);
```

```
    }
  if((bound=calloc(VOL, sizeof(short *)))==NULL) {
    printf("Couldn’t allocate memory!");
    exit(1);
  }
  for(i=0; i<VOL; i++)
    if((bound[i]=calloc(2*DIM, sizeof(short)))==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }
  s=calloc(3*VOL, sizeof(double));

  av_e_old=av_e2_old=av_e4_old=av_ma_x_old=av_ma_y_old=av_ma_z_old=av_m_x_old=av_m_y_old
=av_m_z_old=0.0;
  av_m2_old=av_m4_old=av_nc_old=av_clus_old=av_clus2_old=nc_tot_old=0.0;
  bins_old=lmax_old=meas_old=0;
  copy=0;

  if(APPEND) {
    strcat(strcpy(file,path),"conf.ini");
    fp=fopen(file,"r");
    fscanf(fp,"%d\t%d\t%d\t%d\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%l
f\t%lf\t%lf\t%lf\n",&lmax_old,&bins_old,&copy,&meas_old,&av_e_old,&av_e2_old,&av_e4_old,
&av_ma_x_old,&av_ma_y_old,&av_ma_z_old,&av_m_x_old,&av_m_y_old,&av_m_z_old,&av_m2_old,&a
v_m4_old,&av_nc_old,&av_clus_old,&av_clus2_old,&nc_tot_old);
      for(k=0; k<VOL; k++) {
        fscanf(fp,"%lf\t%lf\t%lf\n",&dummy_x,&dummy_y,&dummy_z);
        s[3*k]=dummy_x;
        s[3*k+1]=dummy_y;
        s[3*k+2]=dummy_z;
      }
      fclose(fp);

      for(k=0; k<VOL; k++) {
        dummy_x=sqrt(s[3*k]*s[3*k]+s[3*k+1]*s[3*k+1]+s[3*k+2]*s[3*k+2]);
        s[3*k]/=dummy_x;
        s[3*k+1]/=dummy_x;
        s[3*k+2]/=dummy_x;
      }
  }
  else {
    for(k=0; k<VOL; k++) {
      phi=2*M_PI*RAN;
      theta=acos(1-2*RAN);
      s[3*k]=(INIT)   ? cos(phi)*sin(theta) : 1;
      s[3*k+1]=(INIT) ? sin(phi)*sin(theta) : 0;
      s[3*k+2]=(INIT) ? cos(theta)          : 0;
    }
  }

  if(UPDATE==’w’ || UPDATE==’y’) nc=1;
  if(UPDATE==’s’ || UPDATE==’w’ || UPDATE==’y’) {
    in_clus=calloc(VOL, sizeof(char));
    next=calloc(VOL, sizeof(int));
    if((bond=calloc(VOL, sizeof(char *)))==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }
    for(i=0; i<VOL; i++)
      if((bond[i]=calloc(2*DIM, sizeof(short)))==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    for(i=0; i<VOL; i++)
      in_clus[i]=0;
  }
  if(s==NULL || ((in_clus==NULL || next==NULL) && (UPDATE==’s’ || UPDATE==’w’ || UPDATE==
’y’))) {
    printf("Couldn’t allocate memory!");
    exit(1);
  }

  ic=0;
  acc_av=nc_tot=av_clus=av_clus2=av_e=av_e2=av_e4=av_ma_x=av_ma_y=av_ma_z=av_m_x=av_m_y=a
v_m_z=av_m2=av_m4=0.0;

  nn_table(nn,bound);

  if(EMSERIES) {
    strcat(strcpy(file,path),"e_series.plo");
    fp1=fopen(file,"w");
    fprintf(fp1,"%d\t%d\n",MEAS,EMPTY);
  }

  if(CORR) {
    if(CORR==2) {
      ee=calloc(VOL,sizeof(double));
      if(ee==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    }
    sprintf(str1,"0.");
    sprintf(str2,"%d",copy);
    strcat(strcpy(file,path),str1);
    strcat(file,str2);
    strcat(file,"s_korr.ser");
    fp2=fopen(file,"w");
    dummy_x=BETA;
    fwrite(&dummy_x,sizeof(double),1,fp2);
    strcat(strcpy(file,path),str1);
    strcat(file,str2);
    strcat(file,"e_korr.ser");
    fp3=fopen(file,"w");
    fwrite(&dummy_x,sizeof(double),1,fp3);
    if(REWEIGHT) {
      for(i=0; i<BETANUM; i++) {
        sprintf(str1,"%d.",i+1);
        strcat(strcpy(file,path),str1);
        strcat(file,str2);
        strcat(file,"s_korr.ser");
        fprs[i]=fopen(file,"w");
        fwrite(beta_list+i,sizeof(double),1,fprs[i]);
        strcat(strcpy(file,path),str1);
        strcat(file,str2);
        strcat(file,"e_korr.ser");
        fpre[i]=fopen(file,"w");
        fwrite(beta_list+i,sizeof(double),1,fpre[i]);
      }
    }
  }

  printf("\n\n***   Heisenberg-Modell auf Geometrie S1xS1x...xR   ***\n");
  printf("\n\nUpdate:                  %c\n", UPDATE);
  printf("Dimension:              %d\nAbmessungen:              ", DIM);
  for(i=0; i<DIM; i++)
    printf("(x%d=%4d)    ", i+1, l[i]);
  printf("\nRandbedingungen:          ");
  for(i=0; i<DIM; i++)
    printf("  %2d         ", rb[i]);
  printf("\nTemperatur Beta:         %.15lf\n", BETA);
  printf("Aequilibrierungs-Sweeps: %d/%d\n", EQUI*(1-APPEND), EQUI);
  printf("Produktions-Sweeps:      %d/%d\n", MEAS, MEAS+meas_old);
```

```
  printf("’Leere’ Sweeps:          %d\n", EMPTY);
  if (UPDATE==’w’)
    printf("1 Sweep entspricht:      %d Wolff Updates\n",WOLFF);
  if (UPDATE==’y’)
    printf("1 Metropolis auf:        %d Wolf Updates\n", SHORT);
  printf("Startkonfiguration:      %d\t\t(1: random, 0: ordered)\n", INIT);
  printf("Zufallsgenerator:        modifizierter R250/103\n");
  if(CORR) {
    if(CORR==1)
      printf("Korrelationsmethode:     gemittelt\n");
    else if(CORR==2)
      printf("Korrelationsmethode:     direkt\n");
    else if(CORR==3)
      printf("Korrelationsmethode:     improved estimators\n");
    printf("Anzahl der Bins:         %d/%d\n", BINS, BINS+bins_old);
    printf("Werte pro Bin:           %d\n\n", MEAS/BINS);
  }
  else
    printf("\n");

  if(STATUS)
    printf("\n\nMessung #00000000");

/* Aequilibrierung */

  for(i=0; i<EQUI*EMPTY*WOLFF*(1-APPEND); i++) {
#if UPDATE==’m’
    metro(s,nn,bound);
#elif UPDATE==’s’
    swendsen(s,nn,bound,bond,in_clus,next,&nc,&ic,&ic2);
#elif UPDATE==’w’
    single(s,nn,bound,bond,in_clus,next,&ic);
#elif UPDATE==’y’
    for(j=0; j<SHORT; j++)
        single(s,nn,bound,bond,in_clus,next,&ic);
      metro(s,nn,bound);
#endif
  }

/* Messungs-Sweeps */

  for(i=0; i<MEAS; i++) {
    for(k=0; k<EMPTY*WOLFF; k++) {
#if UPDATE==’m’
      acc_av+=metro(s,nn,bound);
#elif UPDATE==’s’
      swendsen(s,nn,bound,bond,in_clus,next,&nc,&ic,&ic2);
#elif UPDATE==’w’
      single(s,nn,bound,bond,in_clus,next,&ic);
#elif UPDATE==’y’
      for(j=0; j<SHORT; j++)
        single(s,nn,bound,bond,in_clus,next,&ic);
      acc_av+=metro(s,nn,bound);
#endif
    }

/* Messungen */

#if CORR
#if CORR==2
    e=e_korr2(s,nn,bound,ek,ee);
    s_korr2(s,sk,&m_x,&m_y,&m_z);
    for(j=0; j<LMAX; j++) {
      skorr[j]+=sk[j]/((double)((LMAX-j)*(VOL/LMAX)));
      ekorr[j]+=ek[j]/((double)((LMAX-j)*(VOL/LMAX)));
    }
#elif CORR==1
    e=e_korr(s,nn,bound,ek);
    s_korr(s,sk,&m_x,&m_y,&m_z);
#if REWEIGHT
    for(j=0; j<BETANUM; j++)
      rew[j]=exp((beta_list[j]-BETA)*e);
#endif
    for(j=0; j<LMAX; j++) {
      snorm=sk[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));
      enorm=ek[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX)));
      skorr[j]+=snorm;
      ekorr[j]+=enorm;
#if REWEIGHT
      for(kk=0; kk<BETANUM; kk++) {
        skorr_rew[kk][j]+=snorm*rew[kk];
        ekorr_rew[kk][j]+=enorm*rew[kk];
      }
#endif
    }
/* #elif CORR==3 */
/*     m=s_korr3(s,skorr,next,&ic); */
/*     e=e_korr(s,nn,bound,ek); */
/*     for(j=0; j<LMAX; j++) */
/*       ekorr[j]+=ek[j]/((double)((LMAX-j)*(VOL/LMAX)*(VOL/LMAX))); */

#endif
#if (CORR==1 || CORR==2)
    if(!((i+1)%(MEAS/BINS))) {
      skorr[0]/=(MEAS/BINS);
      ekorr[0]/=(MEAS/BINS);
      for(j=1; j<LMAX/2; j++) {
        skorr[j]=(skorr[j]*(LMAX-j)+skorr[LMAX-j]*j)/(MEAS/BINS*LMAX);
        ekorr[j]=(ekorr[j]*(LMAX-j)+ekorr[LMAX-j]*j)/(MEAS/BINS*LMAX);
      }
      fwrite(skorr,sizeof(double),LMAX/2,fp2);
      fwrite(ekorr,sizeof(double),LMAX/2,fp3);
      for(j=0; j<LMAX; j++) {
        skorr[j]=0.0;
        ekorr[j]=0.0;
      }
#if REWEIGHT
      for(kk=0; kk<BETANUM; kk++) {
        skorr_rew[kk][0]/=(MEAS/BINS);
        ekorr_rew[kk][0]/=(MEAS/BINS);
        for(j=1; j<LMAX/2; j++) {
          skorr_rew[kk][j]=(skorr_rew[kk][j]*(LMAX-j)+skorr_rew[kk][LMAX-j]*j)/(MEAS/BINS
*LMAX);
          ekorr_rew[kk][j]=(ekorr_rew[kk][j]*(LMAX-j)+ekorr_rew[kk][LMAX-j]*j)/(MEAS/BINS
*LMAX);
        }
        fwrite(skorr_rew[kk],sizeof(double),LMAX/2,fprs[kk]);
        fwrite(ekorr_rew[kk],sizeof(double),LMAX/2,fpre[kk]);
        for(j=0; j<LMAX; j++)
          skorr_rew[kk][j]=ekorr_rew[kk][j]=0.0;
      }
#endif
    }
/* #elif CORR==3 */
/*     if(!((i+1)%(MEAS/BINS))) { */
/*       fprintf(fp2,"%.15lf\t",skorr[0]/(MEAS/BINS)); */
/*       fprintf(fp3,"%.15lf\t",ekorr[0]/(MEAS/BINS)); */
/*       for(j=1; j<LMAX/2; j++) { */
/*      fprintf(fp2,"%.15lf\t",skorr[j]/(MEAS/BINS)); */
/*      fprintf(fp3,"%.15lf\t",(ekorr[j]*(LMAX-j)+ekorr[LMAX-j]*j)/(MEAS/BINS*LMAX)); */
/*       } */
/*       for(j=0; j<LMAX; j++) { */
/*      skorr[j]=0.0; */
/*      ekorr[j]=0.0; */
```

```
/*       } */
/*       fprintf(fp2,"\n"); */
/*       fprintf(fp3,"\n"); */
/*     } */
#endif
#else
    e=m_x=m_y=m_z=0.0;
    for(kk=0; kk<VOL; kk++) {
      m_x+=s[3*kk];
      m_y+=s[3*kk+1];
      m_z+=s[3*kk+2];
      nn_sum_x=nn_sum_y=nn_sum_z=0.0;
      for(j=0; j<2*DIM; j+=2) {
        nn_sum_x+=bound[kk][j]*s[3*nn[kk][j]];
        nn_sum_y+=bound[kk][j]*s[3*nn[kk][j]+1];
        nn_sum_z+=bound[kk][j]*s[3*nn[kk][j]+2];
      }
      e+=s[3*kk]*nn_sum_x+s[3*kk+1]*nn_sum_y+s[3*kk+2]*nn_sum_z;
    }
#endif

    m_x/=(double)VOL;
    m_y/=(double)VOL;
    m_z/=(double)VOL;
    e/=(double)VOL;

#if STATUS
    printf("\b\b\b\b\b\b\b\b%8d",i);
#endif

#if EMSERIES
    fprintf(fp1, "%d\t%.15lf\t%.15lf\t%.15lf\t%.15lf\n", (i+1)*EMPTY*sh, e, m_x, m_y, m_
z);
#endif

    av_e+=e;
    av_e2+=e*e;
    av_e4+=e*e*e*e;

    av_ma_x+=fabs(m_x);
    av_ma_y+=fabs(m_y);
    av_ma_z+=fabs(m_z);
    av_m_x+=m_x;
    av_m_y+=m_y;
    av_m_z+=m_z;
    av_m2+=m_x*m_x+m_y*m_y+m_z*m_z;
    av_m4+=(m_x*m_x+m_y*m_y+m_z*m_z)*(m_x*m_x+m_y*m_y+m_z*m_z);

    nc_tot+=nc;
    av_clus+=ic;
    av_clus2+=ic2;

  }

/* Bildschirmausgabe der Ergebnisse */

  if(UPDATE==’m’ || UPDATE==’y’)
    printf("\nacc     = %10.10lf\n", acc_av/(MEAS*EMPTY*SHORT));
  av_e=(av_e+av_e_old*meas_old)/(MEAS+meas_old);
  av_e2=(av_e2+av_e2_old*meas_old)/(MEAS+meas_old);
  av_e4=(av_e4+av_e4_old*meas_old)/(MEAS+meas_old);
  c=BETA*BETA*VOL*(av_e2-av_e*av_e);

  av_ma_x=(av_ma_x+av_ma_x_old*meas_old)/(MEAS+meas_old);
  av_ma_y=(av_ma_y+av_ma_y_old*meas_old)/(MEAS+meas_old);
  av_ma_z=(av_ma_z+av_ma_z_old*meas_old)/(MEAS+meas_old);
  av_m_x=(av_m_x+av_m_x_old*meas_old)/(MEAS+meas_old);
  av_m_y=(av_m_y+av_m_y_old*meas_old)/(MEAS+meas_old);
  av_m_z=(av_m_z+av_m_z_old*meas_old)/(MEAS+meas_old);
  av_m2=(av_m2+av_m2_old*meas_old)/(MEAS+meas_old);
  av_m4=(av_m4+av_m4_old*meas_old)/(MEAS+meas_old);

  sus=VOL*(av_m2-(av_ma_x*av_ma_x+av_ma_y*av_ma_y+av_ma_z*av_ma_z));
  chi=VOL*av_m2;
  binder=1-av_m4/(3.0*av_m2*av_m2);

  av_nc=(nc_tot+av_nc_old*meas_old)/(MEAS+meas_old);
  av_clus=(av_clus+av_clus_old*nc_tot_old)/(nc_tot+nc_tot_old);
  av_clus2=(av_clus2+av_clus2_old*nc_tot_old)/(nc_tot+nc_tot_old);
  switch(UPDATE) {
    case ’s’: chi_imp=av_clus2/av_clus;
              break;
    case ’w’: chi_imp=av_clus;
              break;
    case ’y’: chi_imp=av_clus;
              break;
    default:  chi_imp=0.0;
              break;
  }

  if(UPDATE!=’m’) {
    printf("nc       = %10.10lf\n", av_nc);
    printf("clus     = %10.10lf\n", av_clus);
    printf("clus/vol = %10.10lf\n", av_clus/((double)VOL));
  }
  printf("-e       = %10.10lf\n", av_e);
  printf("e+d      = %10.10lf\n", DIM-av_e);
  printf("c        = %10.10lf\n", c);
  printf("abs(m)_x = %10.10lf\n", av_ma_x);
  printf("abs(m)_y = %10.10lf\n", av_ma_y);
  printf("abs(m)_z = %10.10lf\n", av_ma_z);
  printf("m_x      = %10.10lf\n", av_m_x);
  printf("m_y      = %10.10lf\n", av_m_y);
  printf("m_z      = %10.10lf\n", av_m_z);
  printf("m2       = %10.10lf\n", av_m2);
  printf("sus      = %10.10lf\n", sus);
  printf("chi      = %10.10lf\n", chi);
  if(UPDATE!=’m’)
    printf("chi_imp  = %10.10lf\n", chi_imp);
  printf("binder   = %10.10lf\n", binder);

  if (CORR) {
    fclose(fp2);
    fclose(fp3);
    if(REWEIGHT)
      for(i=0; i<BETANUM; i++) {
        fclose(fprs[i]);
        fclose(fpre[i]);
      }
  }

  times(&u_end);
  printf("\nSystemzeit: %ds\n", time(NULL)-s_time);
  printf("CPU-Zeit:   %ds\n", (int)((u_end.tms_utime-u_start.tms_utime))/100);

/* Zustands-dump */

  copy++;
  strcat(strcpy(file,path),"conf.ini");
  fp=fopen(file,"w");
  fprintf(fp,"%d\t%d\t%d\t%d\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.1
5lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\t%.15lf\n",LMAX,BINS+bins_old,copy,ME
AS+meas_old,av_e,av_e2,av_e4,av_ma_x,av_ma_y,av_ma_z,av_m_x,av_m_y,av_m_z,av_m2,av_m4,av_
nc,av_clus,av_clus2,nc_tot);
```

```
  for(k=0; k<VOL; k++)
    fprintf(fp,"%.15lf\t%.15lf\t%.15lf\n",s[3*k],s[3*k+1],s[3*k+2]);
  fclose(fp);
  strcat(strcpy(file,path),"ran.ini");
  write_random_generator(file);

/* Aufraeumen */

  if(CORR==2)
    free(ee);
  free(rannum);
  free(s);
  for(i=0; i<VOL; i++) {
    free(bound[i]);
    free(nn[i]);
  }
  free(bound);
  free(nn);
  if(UPDATE=='s' || UPDATE=='w' || UPDATE=='y') {
    for(i=0; i<VOL; i++)
      free(bond[i]);
    free(bond);
    free(in_clus);
    free(next);
  }

  if(EMSERIES)
    fclose(fp1);

  return(0);
}

nn_table(nn,bound)
int **nn;
short **bound;
{
  int i, id, isf[DIM][LMAX], isb[DIM][LMAX], loff[DIM], k, kk, ind;
  short sigf[DIM][LMAX], sigb[DIM][LMAX];
  extern short l[DIM], rb[DIM];

  for(id=0; id<DIM; id++) {
    for(i=0; i<l[id]; i++) {
      isf[id][i]=1;
      isb[id][i]=-1;
      sigf[id][i]=1;
      sigb[id][i]=1;
    }
    isf[id][l[id]-1]=1-l[id];               /* Randbedingungen */
    isb[id][0]=l[id]-1;
    sigf[id][l[id]-1]=rb[id];
    sigb[id][0]=rb[id];
  }

  for(id=loff[0]=1; id<DIM; id++)         /* Offsets fuer die hoeheren Raumdimensionen *
/
    loff[id]=loff[id-1]*l[id-1];

  for(k=0; k<VOL; k++) {
    kk=k;
    for(id=DIM-1; id>=0; id--) {
      ind=kk/loff[id];
      kk-=ind*loff[id];
      nn[k][2*id]             = k + loff[id]*isf[id][ind];   /* positive Koordinatenrich
tungen */
      nn[k][2*id+1]           = k + loff[id]*isb[id][ind];   /* negative Koordinatenrich
tungen */
      bound[k][2*id]          = sigf[id][ind];
      bound[k][2*id+1]        = sigb[id][ind];
    }
  }
}

double metro(s,nn,bound)
int **nn;
short **bound;
double *s;
{
  int acc, i, k;
  double ide, phi, theta;
  double nn_sum_x,nn_sum_y,nn_sum_z,s_neu_x,s_neu_y,s_neu_z;

  acc=0;
  for(k=0; k<VOL; k++) {
    nn_sum_x=nn_sum_y=nn_sum_z=0;
    for(i=0; i<2*DIM; i++) {
      nn_sum_x+=bound[k][i]*s[3*nn[k][i]];
      nn_sum_y+=bound[k][i]*s[3*nn[k][i]+1];
      nn_sum_z+=bound[k][i]*s[3*nn[k][i]+2];
    }
    phi=2*M_PI*RAN;
    theta=acos(1-2*RAN);
    s_neu_x=cos(phi)*sin(theta);
    s_neu_y=sin(phi)*sin(theta);
    s_neu_z=cos(theta);
    ide=(s[3*k]-s_neu_x)*nn_sum_x+(s[3*k+1]-s_neu_y)*nn_sum_y+(s[3*k+2]-s_neu_z)*nn_sum_z
;
    if(ide <= 0) {
      s[3*k]=s_neu_x;
      s[3*k+1]=s_neu_y;
      s[3*k+2]=s_neu_z;
      acc++;
    }
    else if(RAN < exp(-BETA*ide)) {
      s[3*k]=s_neu_x;
      s[3*k+1]=s_neu_y;
      s[3*k+2]=s_neu_z;
      acc++;
    }
  }
  return(acc/((double)VOL));
}

swendsen(s,nn,bound,bond,in_clus,next,nc,ic_tot,ic2_tot)
int **nn, *next, *nc, *ic_tot, *ic2_tot;
double *s;
short **bound;
char **bond, *in_clus;
{
  int i, k1, k0, clus_leng, tested, k, flip;
  double r_x, r_y, r_z, prod, phi, theta;


  phi=2*M_PI*RAN;
  theta=acos(1-2*RAN);
  r_x=cos(phi)*sin(theta);
  r_y=sin(phi)*sin(theta);
  r_z=cos(theta);

  for(k0=0; k0<VOL; k0++) {
    *(in_clus+k0)=0;
    for(i=0; i<2*DIM; i+=2) {
      k1=nn[k0][i];
      prod=bound[k0][i]*(s[3*k0]*r_x+s[3*k0+1]*r_y+s[3*k0+2]*r_z)*(s[3*k1]*r_x+s[3*k1+1]*
r_y+s[3*k1+2]*r_z);
```

```
      if(prod<=0)
        bond[k0][i]=0;
      else if(RAN<exp(-2*BETA*prod))
        bond[k0][i]=0;
      else
        bond[k0][i]=1;
    bond[k1][i+1]=bond[k0][i];
    }
  }

  *nc=*ic_tot=*ic2_tot=0;

  for(k=0; k<VOL; k++) {
    if(!*(in_clus+k)) {
      flip=(RAN<.5);
      (*nc)++;
      clus_leng=tested=0;
      *next=k;
      *(in_clus+k)=1;

      while(tested<=clus_leng) {
        k0=*(next+tested);
        if(flip) {
          prod=s[3*k0]*r_x+s[3*k0+1]*r_y+s[3*k0+2]*r_z;
          s[3*k0]-=2*prod*r_x;
          s[3*k0+1]-=2*prod*r_y;
          s[3*k0+2]-=2*prod*r_z;
        }
        for(i=0; i<2*DIM; i++) {
          k1=nn[k0][i];
          if((!*(in_clus+k1)) && bond[k0][i]) {
            clus_leng++;
            *(next+clus_leng)=k1;
            *(in_clus+k1)=1;
          }
        }
        tested++;
      }

      clus_leng++;
      *ic_tot+=clus_leng;
      *ic2_tot+=clus_leng*clus_leng;
    }
  }
}

single(s,nn,bound,bond,in_clus,next,clus_leng)
int **nn, *next, *clus_leng;
double *s;
short **bound;
char **bond, *in_clus;
{
  int i, k0, k1, tested, length;
  double r_x, r_y, r_z, prod1, prod2, phi, theta;

  for(i=0; i<*clus_leng; i++)
    *(in_clus+*(next+i))=0;

  k0=(int)(RAN*VOL);
  if(k0>=VOL)
    k0=VOL-1;

  phi=2*M_PI*RAN;
  theta=acos(1-2*RAN);
  r_x=cos(phi)*sin(theta);
  r_y=sin(phi)*sin(theta);
  r_z=cos(theta);

  *clus_leng=tested=0;
  *next=k0;
  *(in_clus+k0)=1;

  while(tested<=*clus_leng) {
    k0=*(next+tested);
    prod1=s[3*k0]*r_x+s[3*k0+1]*r_y+s[3*k0+2]*r_z;
    for(i=0; i<2*DIM; i++) {
      k1=nn[k0][i];
      if(!(in_clus[k1])) {
        prod2=bound[k0][i]*prod1*(s[3*k1]*r_x+s[3*k1+1]*r_y+s[3*k1+2]*r_z);
        if(prod2>=0)
          if(RAN>exp(-2*BETA*prod2))
            {
              (*clus_leng)++;
              *(in_clus+k1)=1;
              *(next+*clus_leng)=k1;
            }
      }
    }
    s[3*k0]-=2*prod1*r_x;
    s[3*k0+1]-=2*prod1*r_y;
    s[3*k0+2]-=2*prod1*r_z;
    tested++;
  }
  (*clus_leng)++;
}

double e_korr(s,nn,bound,ek)
double *s;
int **nn;
short **bound;
double *ek;
{
  int i, j, k;
  int factor;
  double nn_sum_x, nn_sum_y, nn_sum_z;
  double ee[LMAX], e;

  e=0.0;
   for(i=0; i<LMAX; i++) {
     ee[i]=0.0;
     factor=VOL/LMAX*i;
     for(j=0; j<VOL/LMAX; j++) {
       nn_sum_x=0.0;
       nn_sum_y=0.0;
       nn_sum_z=0.0;
       for(k=0; k<2*DIM-2; k+=2) {
        nn_sum_x+=bound[factor+j][k]*s[3*nn[factor+j][k]];
        nn_sum_y+=bound[factor+j][k]*s[3*nn[factor+j][k]+1];
        nn_sum_z+=bound[factor+j][k]*s[3*nn[factor+j][k]+2];
       }
      nn_sum_x+=bound[factor+j][2*DIM-2]*s[3*nn[factor+j][2*DIM-2]];
      nn_sum_y+=bound[factor+j][2*DIM-2]*s[3*nn[factor+j][2*DIM-2]+1];
      nn_sum_z+=bound[factor+j][2*DIM-2]*s[3*nn[factor+j][2*DIM-2]+2];
      ee[i]+=s[3*(factor+j)]*nn_sum_x+s[3*(factor+j)+1]*nn_sum_y+s[3*(factor+j)+2]*nn_sum
_z;
     }
     e+=ee[i];
   }

   for(i=0; i<LMAX; i++) {
     ek[i]=0;
     for(j=0; j<LMAX-i; j++)
       ek[i]+=ee[j]*ee[j+i];
   }
```

```
   return(e);
}

double e_korr2(s,nn,bound,ek,ee)
double *s;
int **nn;
short **bound;
double *ek, *ee;
{
  int i, j, k, n;
  double nn_sum_x, nn_sum_y, nn_sum_z;
  double e;

  e=0.0;
  for(i=0; i<VOL; i++) {
    nn_sum_x=0.0;
    nn_sum_y=0.0;
    nn_sum_z=0.0;
    for(k=0; k<2*DIM; k++) {
      nn_sum_x+=bound[i][k]*s[3*nn[i][k]];
      nn_sum_y+=bound[i][k]*s[3*nn[i][k]+1];
      nn_sum_z+=bound[i][k]*s[3*nn[i][k]+2];
    }
    ee[i]=(s[3*i]*nn_sum_x+s[3*i+1]*nn_sum_y+s[3*i+2]*nn_sum_z)/2;
    e+=ee[i];
  }

  for(i=0; i<LMAX; i++) {
    ek[i]=0.0;
    for(j=0; j<LMAX-i; j++)
      for(n=0; n<VOL/LMAX; n++)
        ek[i]+=ee[j*VOL/LMAX+n]*ee[(j+i)*VOL/LMAX+n];
  }

  return(e);
}

s_korr(s,sk,m_x,m_y,m_z)
double *s;
double *sk, *m_x, *m_y, *m_z;
{
  int i,j;
  double ss[3*LMAX];
  int factor;

  *m_x=*m_y=*m_z=0.0;
  for(i=0; i<3*LMAX; i+=3) {
    ss[i]=ss[i+1]=ss[i+2]=0.0;
    factor=VOL/LMAX*i;
    for(j=0; j<3*VOL/LMAX; j+=3) {
      ss[i]+=s[factor+j];
      ss[i+1]+=s[factor+j+1];
      ss[i+2]+=s[factor+j+2];
    }
    *m_x+=ss[i];
    *m_y+=ss[i+1];
    *m_z+=ss[i+2];
  }

  for(i=0; i<LMAX; i++) {
    sk[i]=0;
    for(j=0; j<LMAX-i; j++)
      sk[i]+=ss[3*j]*ss[3*(j+i)]+ss[3*j+1]*ss[3*(j+i)+1]+ss[3*j+2]*ss[3*(j+i)+2];
  }
}

s_korr2(s,sk,m_x,m_y,m_z)
double *s;
double *sk;
double *m_x, *m_y, *m_z;
{
  int i, j, n;

  *m_x=*m_y=*m_z=0.0;
  for(n=0; n<VOL; n++) {
    *m_x+=s[3*n];
    *m_y+=s[3*n+1];
    *m_z+=s[3*n+2];
  }

  for(i=0; i<LMAX; i++) {
    sk[i]=0.0;
    for(j=0; j<LMAX-i; j++)
      for(n=0; n<VOL/LMAX; n++)
        sk[i]+=s[3*(j*VOL/LMAX+n)]*s[3*((j+i)*VOL/LMAX+n)]+s[3*(j*VOL/LMAX+n)+1]*s[3*((j+
i)*VOL/LMAX+n)+1]+s[3*(j*VOL/LMAX+n)+2]*s[3*((j+i)*VOL/LMAX+n)+2];
  }
}

/* s_korr3(s,skorr,next,clus_leng,m_x,m_y) */
/* double *s; */
/* int *clus_leng, *next; */
/* double *skorr; */
/* double *m_x, *m_y; */
/* { */
/*   double sk[2*LMAX]; */
/*   int i, k0, k1; */

/*   *m_x=*m_y=0.0; */
/*   for(i=0; i<LMAX; i++) { */
/*     sk[i]=0.0; */
/*     for(k0=0; k0<VOL/LMAX; k0++) { */
/*       m_x+=s[2*(i*VOL/LMAX+k0)]; */
/*       m_y+=s[2*(i*VOL/LMAX+k0)+1]; */
/*     } */
/*   } */

/*   for (k0=0; k0<*clus_leng; k0++) */
/*     for(k1=0; k1<=k0; k1++) */
/*       if((next[k0]%(VOL/LMAX))==(next[k1]%(VOL/LMAX))) */
/*      sk[abs(next[k0]/(VOL/LMAX)-next[k1]/(VOL/LMAX))]++; */

/*   for(i=0; i<LMAX; i++) */
/*     skorr[i]+=sk[i]/((double)(*clus_leng)); */

/*   return(m); */
/* } */

double get_r250()
{
  static int count;

  if(count>=NRAND) {
    vector_random_generator(NRAND,rannum);
    count=0;
  }

  return(*(rannum+count++));
}
```

```
/************/
/* ranvec.c */
/************/


/*****************************************************************/
/*                                                               */
/* Vectorized version of a (hopefully) improved shift-register   */
/* pseudorandom number generator.                                */
/* Based on two independent generators with different pairs      */
/* of "magic numbers", here chosen as (250,103) and (521,168).   */
/* The output of those is XORed together and yields the final    */
/* random number. For theoretical reasons, this generator should */
/* be significantly less hampered by correlations than the simple */
/* good old R250.                                                */
/*                                                               */
/* Disclaimer of warranty:                                       */
/* I will not be held responsible for any problems whatsoever    */
/* which this program may cause.                                 */
/*                                                               */
/* B. Duenweg, July 9, 1996.                                     */
/*                                                               */
/* Some (not all!) other possible magic umbers are (see also     */
/* Kirkpatrick and Stoll, Journ. Comp. Phys. 40, 517 (1981), and */
/* N. Zierler, Information and Control 15, p. 67 (1969)):        */
/* (98, 27)                                                      */
/* (521, 32)                                                     */
/* (521, 48)                                                     */
/* (521, 158)                                                    */
/* (607, 105)                                                    */
/* (607, 147)                                                    */
/* (607, 273)                                                    */
/* (1279, 216)                                                   */
/* (1279, 418)                                                   */
/* (2281, 715)                                                   */
/* (2281, 915)                                                   */
/* (2281, 1029)                                                  */
/* (9689, 4187)                                                  */
/*                                                               */
/* The program can be changed very easily to other magic numbers */
/* by just changing the parameters                               */
/* BIGMAGIC1, BIGMAGIC2, SMALLMAGIC1 and SMALLMAGIC2.            */
/* Since it is explicitly written for 31 bit integers, it        */
/* should produce the SAME sequence on any architecture.         */
/*                                                               */
/* This is how it works:                                         */
/*                                                               */
/* - Specify a "seed" value iseed.                               */
/*                                                               */
/* - Specify how many random numbers should be generated         */
/*   on ONE call of the generator. Call this number, say,        */
/*   nrand. A typical value is, say, 100000.                     */
/*                                                               */
/* - Invoke the function                                         */
/*   init_vector_random_generator(iseed,nrand).                  */
/*   This does the following:                                    */
/*   - A simple congruential generator is run for NWARM          */
/*     (in our case 10000) times.                                */
/*   - Two integer working arrays of size BIGMAGIC1 + nrand,     */
/*     BIGMAGIC2 + nrand, named rand_w_array1 and rand_w_array2  */
/*     are created (i.e. memory is allocated).                   */
/*   - The first BIGMAGICX array elements are                    */
/*     filled with (congruential) random numbers.                */
/*   - The bit columns are treated for linear independence.      */
/*   - The shift-register generators are run for nrand times     */
/*     (in order to also warm them up) and are then ready to go. */
/*                                                               */
/* - Invoke the function                                         */
/*   vector_random_generator(nrand, random_numbers).             */
/*   Result: Normalized double precision random numbers in [0:1] */
/*   are written on the array random_numbers.                    */
/*   The current "state" of the generator is "coded" in the first */
/*   BIGMAGICX elements of rand_w_arrayX.                        */
/*                                                               */
/* - There are also routines provided to save this status to a   */
/*   file, and read it from there. Have care: In case you read   */
/*   the status, make sure to first run the function             */
/*   init_vector_random_generator in order to assure proper      */
/*   memory allocation!                                          */
/*                                                               */
/*****************************************************************/


#include <stdio.h>
#include <stdlib.h>

#define BIGMAGIC1 250                       /* magic numbers for the      */
#define SMALLMAGIC1 103                     /*   first generator          */
#define BIGMAGIC2 521                       /* magic numbers for the      */
#define SMALLMAGIC2 168                     /*   second generator         */
#define NBIT 32                             /* use only (NBIT - 1) bits   */
#define BIGINTEGER 2147483647               /* = largest integer          */
#define BIGFLOAT 2147483647.                /* same in float              */
#define FACTOR 4.6566128752457969e-10       /* = 1. / (largest integer)   */
#define MULTIPLY 16807.                     /* for congruential generator */
#define NWARM 10000                         /* number of empty runs, c.g. */
#define WORKFILE "ran250.dat"               /* to store the status        */

/* The working arrays are declared as static so they need not be passed */
/* to the main program */

int *rand_w_array1 = NULL;
int *rand_w_array2 = NULL;  /* Waren vorher STATIC deklariert! */

void init_vector_random_generator(int iseed,int nrand)
{
  extern int *rand_w_array1;
  extern int *rand_w_array2;
  double rmod;
  int i, ihlp, imask1, imask2;
  int icyc, ncyc, nrest, ibas1, ibas2, ibas3;

  if(iseed <=0 || iseed >= BIGINTEGER)
    {
      printf("Message from random number initialization:\n");
      printf("Please specify a seed smaller than %d\n",BIGINTEGER);
      exit(0);
    }
  if(nrand <=0)
    {
      printf("Message from random number initialization:\n");
      printf("Please specify a positive number of random numbers\n");
      exit(0);
    }

  rmod = (double) (iseed);

/* Warm up the congruential generator */

  for(i = 0; i < NWARM; ++i)
    {
      rmod = MULTIPLY * rmod;
      rmod = rmod - ( (double) ( (int) (rmod * FACTOR) ) ) * BIGFLOAT;
      ihlp = (int) (rmod + 0.1);      /* This is done to get rid of */
```

```
      rmod = (double) (ihlp);         /* possible roundoff errors   */
    }

/* Allocate memory for the working arrays */

  rand_w_array1 = (int *) calloc(BIGMAGIC1 + nrand,sizeof(int));
  rand_w_array2 = (int *) calloc(BIGMAGIC2 + nrand,sizeof(int));

/* Put congruential random numbers onto the working arrays */

  for(i = 0; i < BIGMAGIC1; ++i)
    {
      rmod = MULTIPLY * rmod;
      rmod = rmod - ( (double) ( (int) (rmod * FACTOR) ) ) * BIGFLOAT;
      ihlp = (int) (rmod + 0.1);
      rmod = (double) (ihlp);
      rand_w_array1[i] = ihlp;
    }

  for(i = 0; i < BIGMAGIC2; ++i)
    {
      rmod = MULTIPLY * rmod;
      rmod = rmod - ( (double) ( (int) (rmod * FACTOR) ) ) * BIGFLOAT;
      ihlp = (int) (rmod + 0.1);
      rmod = (double) (ihlp);
      rand_w_array2[i] = ihlp;
    }

/* Linear independence of the bit columns for both generators. */
/* Put ones on the main diagonal, and zeroes above.            */
/* & is the bitwise AND                                        */
/* | is the bitwise OR                                         */
/* ^ is the bitwise XOR                                        */

  imask1 = 1;
  imask2 = BIGINTEGER;
  for(i = NBIT - 2; i > 0; --i)
    {
      rand_w_array1[i] = ( rand_w_array1[i] | imask1 ) & imask2;
      rand_w_array2[i] = ( rand_w_array2[i] | imask1 ) & imask2;
      imask2 = imask2 ^ imask1;
      imask1 = imask1 * 2;
    }
  rand_w_array1[0] = imask1;    /* This last element is treated separately */
  rand_w_array2[0] = imask1;    /* in order to avoid overflow in imask1    */

/* Warm up. Same structure as in vector_random_generator.      */
/* Double loop structure to enable vectorization of inner loop */

/* First, generator one */

  ncyc  = nrand / SMALLMAGIC1;
  nrest = nrand - SMALLMAGIC1 * ncyc;

  ibas3 = BIGMAGIC1;                /* position of first new random number */
  ibas2 = BIGMAGIC1 - SMALLMAGIC1;  /* position of first input for this    */
  ibas1 = 0;                        /* position of second input for this   */

  for(icyc = 0; icyc < ncyc; ++icyc)
    {
#pragma ivdep
      for(i = 0; i < SMALLMAGIC1; ++i)
        {
          rand_w_array1[ibas3 + i] = rand_w_array1[ibas1 + i]
                                   ^ rand_w_array1[ibas2 + i];
        }
      ibas1 = ibas1 + SMALLMAGIC1;
      ibas2 = ibas2 + SMALLMAGIC1;
      ibas3 = ibas3 + SMALLMAGIC1;
    }

  if(nrest > 0)
    {
#pragma ivdep
      for(i = 0; i < nrest; ++i)
        {
          rand_w_array1[ibas3 + i] = rand_w_array1[ibas1 + i]
                                   ^ rand_w_array1[ibas2 + i];
        }
    }

/* Put last elements to the beginning */

#pragma ivdep
  for(i = 0; i < BIGMAGIC1; ++i)
    {
      rand_w_array1[i] = rand_w_array1[nrand + i];
    }

/* Now the same for the second generator */

  ncyc  = nrand / SMALLMAGIC2;
  nrest = nrand - SMALLMAGIC2 * ncyc;

  ibas3 = BIGMAGIC2;                /* position of first new random number */
  ibas2 = BIGMAGIC2 - SMALLMAGIC2;  /* position of first input for this    */
  ibas1 = 0;                        /* position of second input for this   */

  for(icyc = 0; icyc < ncyc; ++icyc)
    {
#pragma ivdep
      for(i = 0; i < SMALLMAGIC2; ++i)
        {
          rand_w_array2[ibas3 + i] = rand_w_array2[ibas1 + i]
                                   ^ rand_w_array2[ibas2 + i];
        }
      ibas1 = ibas1 + SMALLMAGIC2;
      ibas2 = ibas2 + SMALLMAGIC2;
      ibas3 = ibas3 + SMALLMAGIC2;
    }

  if(nrest > 0)
    {
#pragma ivdep
      for(i = 0; i < nrest; ++i)
        {
          rand_w_array2[ibas3 + i] = rand_w_array2[ibas1 + i]
                                   ^ rand_w_array2[ibas2 + i];
        }
    }

/* Put last elements to the beginning */

#pragma ivdep
  for(i = 0; i < BIGMAGIC2; ++i)
    {
      rand_w_array2[i] = rand_w_array2[nrand + i];
    }

/* Initialization complete */

  return;
}
```

```c
void vector_random_generator(int nrand, double *random_numbers)
{
  extern int *rand_w_array1;
  extern int *rand_w_array2;
  int i, icyc, ncyc, nrest, ibas1, ibas2, ibas3;

/* First, run generator one */

  ncyc  = nrand / SMALLMAGIC1;
  nrest = nrand - SMALLMAGIC1 * ncyc;

  ibas3 = BIGMAGIC1;                 /* position of first new random number */
  ibas2 = BIGMAGIC1 - SMALLMAGIC1;   /* position of first input for this    */
  ibas1 = 0;                         /* position of second input for this   */

  for(icyc = 0; icyc < ncyc; ++icyc)
    {
#pragma ivdep
      for(i = 0; i < SMALLMAGIC1; ++i)
        {
          rand_w_array1[ibas3 + i] = rand_w_array1[ibas1 + i]
                                   ^ rand_w_array1[ibas2 + i];
        }
      ibas1 = ibas1 + SMALLMAGIC1;
      ibas2 = ibas2 + SMALLMAGIC1;
      ibas3 = ibas3 + SMALLMAGIC1;
    }

  if(nrest > 0)
    {
#pragma ivdep
      for(i = 0; i < nrest; ++i)
        {
          rand_w_array1[ibas3 + i] = rand_w_array1[ibas1 + i]
                                   ^ rand_w_array1[ibas2 + i];
        }
    }

/* Put last elements to the beginning */

#pragma ivdep
  for(i = 0; i < BIGMAGIC1; ++i)
    {
      rand_w_array1[i] = rand_w_array1[nrand + i];
    }

/* Now the same for the second generator */

  ncyc  = nrand / SMALLMAGIC2;
  nrest = nrand - SMALLMAGIC2 * ncyc;

  ibas3 = BIGMAGIC2;                 /* position of first new random number */
  ibas2 = BIGMAGIC2 - SMALLMAGIC2;   /* position of first input for this    */
  ibas1 = 0;                         /* position of second input for this   */

  for(icyc = 0; icyc < ncyc; ++icyc)
    {
#pragma ivdep
      for(i = 0; i < SMALLMAGIC2; ++i)
        {
          rand_w_array2[ibas3 + i] = rand_w_array2[ibas1 + i]
                                   ^ rand_w_array2[ibas2 + i];
        }
      ibas1 = ibas1 + SMALLMAGIC2;
      ibas2 = ibas2 + SMALLMAGIC2;
      ibas3 = ibas3 + SMALLMAGIC2;
    }

  if(nrest > 0)
    {
#pragma ivdep
      for(i = 0; i < nrest; ++i)
        {
          rand_w_array2[ibas3 + i] = rand_w_array2[ibas1 + i]
                                   ^ rand_w_array2[ibas2 + i];
        }
    }

/* Put last elements to the beginning */

#pragma ivdep
  for(i = 0; i < BIGMAGIC2; ++i)
    {
      rand_w_array2[i] = rand_w_array2[nrand + i];
    }

/* Generate normalized random numbers:                        */
/* Take output from generator one and combine it with         */
/* that from generator two, via a simple XOR                  */

#pragma ivdep
  for(i = 0; i < nrand; ++i)
    {
      random_numbers[i] = FACTOR *
        (rand_w_array1[i + BIGMAGIC1] ^ rand_w_array2[i + BIGMAGIC2]);
    }

  return;
}

void write_random_generator(char *dateiname)
{
  extern int *rand_w_array1;
  extern int *rand_w_array2;
  FILE *fp;

  fp = fopen(dateiname, "w");
  fwrite(rand_w_array1,sizeof(int),BIGMAGIC1,fp);
  fwrite(rand_w_array2,sizeof(int),BIGMAGIC2,fp);
  fclose(fp);
  return;
}

void read_random_generator(char *dateiname)
{
  extern int *rand_w_array1;
  extern int *rand_w_array2;
  FILE *fp;

  fp = fopen(dateiname,"r");
  fread(rand_w_array1,sizeof(int),BIGMAGIC1,fp);
  fread(rand_w_array2,sizeof(int),BIGMAGIC2,fp);
  fclose(fp);
  return;
}
```

```
/************/
/* ranlux.c */
/************/

#define R      24
#define S      10
#define P      223
#define ICONS  2147483563

#include <stdio.h>

const float leastbit=1.0/16777216;
const float halfleast=1.0/4096;
const int   int24=16777216;
static char next[R];
static float seeds[R], carry;
static int si, ri, inr;

rluxgo(seed)
int seed;
{
  extern char next[];
  extern float seeds[], carry;
  extern int si, ri, inr;
  int i, jseed, k;

  for(i=1; i<R; i++)
    next[i]=i-1;
  next[0]=R-1;
  si=S-1;
  ri=R-1;
  inr=0;
  jseed=seed;
  for(i=0; i<R; i++) {
    k=jseed/53668;
    jseed=40014*(jseed-k*53668)-k*12211;
    jseed=(jseed<0) ? jseed+ICONS : jseed;
    seeds[i]=((float)(jseed%int24))*leastbit;
  }
  carry=(seeds[R-1]==0) ? leastbit : 0;
}

ranlux(ranvec,length)
float *ranvec;
int length;
{
  extern char next[];
  extern float seeds[], carry;
  extern int si, ri, inr;
  int i, j;
  float new;

  for(i=0; i<length; i++) {
    new=seeds[si]-seeds[ri]-carry;
    if(new<0) {
      new+=1;
      carry=leastbit;
    }
    else
      carry=0;
    seeds[ri]=new;
    ri=next[ri];
    si=next[si];
    ranvec[i]=new;
    inr++;
    if(inr==R) {
      inr=0;
      for(j=0; j<P-R; j++) {
        new=seeds[si]-seeds[ri]-carry;
        if(new<0) {
          new+=1;
          carry=leastbit;
        }
        else
          carry=0;
        seeds[ri]=new;
        ri=next[ri];
        si=next[si];
      }
    }
  }
}

write_ranlux(file)
char *file;
{
  extern float seeds[];
  FILE *fp;

  fp=fopen(file,"w");
  fwrite(seeds,sizeof(float),R,fp);
  fclose(fp);
}

read_ranlux(file)
char *file;
{
  extern float seeds[];
  FILE *fp;

  fp=fopen(file,"r");
  fread(seeds,sizeof(float),R,fp);
  fclose(fp);
}
```

```
/*************/
/* analyse.c */
/*************/


/* Fehler-Analyse per */
/* jack knife         */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <nrutil.h>

#define LMAXMAX2 300
#define DELTA      2

main(argc,argv)
int argc;
char *argv[];
{
  FILE *fpe, *fps, *fperg, *fpres, *fp;
  int lmax, bins, sstart, send, estart, eend;
  int i, j, n;
  int dummy2;
  double *sk, *sk_av, *thetas, *thetasi, *thetasdot, *thetastild, *vars;
  double *ek, *ek_av, *thetae, *thetaei, *thetaedot, *thetaetild, *vare;
  double v;
  double *corrs[LMAXMAX2], *corre[LMAXMAX2];
  double dummy, dummys, dummye;
  char file[100];


  if(argc==2 || argc==4 || argc==6 || argc==3) {
    strcat(strcpy(file,argv[1]),"s_korr.ser");
    fps=fopen(file,"r");
    strcat(strcpy(file,argv[1]),"e_korr.ser");
    fpe=fopen(file,"r");
    strcat(strcpy(file,argv[1]),"conf.ini");
    fp=fopen(file,"r");
    fscanf(fp,"%d\t%d\t",&lmax,&bins);
    fclose(fp);

    sk=calloc(lmax/2*bins,sizeof(double));
    ek=calloc(lmax/2*bins,sizeof(double));
    if(ek==NULL || sk==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }

    for(i=0; i<bins; i++) {
      for(j=0; j<lmax/2; j++) {
        fscanf(fps,"%lf\t",sk+i*lmax/2+j);
        fscanf(fpe,"%lf\t",ek+i*lmax/2+j);
      }
      fscanf(fps,"\n");
      fscanf(fpe,"\n");
    }

    fclose(fps);
    fclose(fpe);
  }
  else {
    printf("\nanalyse: wrong number of arguments");
    exit(1);
  }

  if(argc==2) {
    vars=calloc(lmax/2,sizeof(double));
    thetas=calloc(lmax/2,sizeof(double));
    thetasi=calloc(lmax/2*bins,sizeof(double));
    thetasdot=calloc(lmax/2,sizeof(double));
    thetastild=calloc(lmax/2,sizeof(double));
    sk_av=calloc(lmax/2,sizeof(double));
    if(sk_av==NULL||thetasi==NULL||thetas==NULL||thetasdot==NULL||thetastild==NULL||vars==NU
LL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }

    vare=calloc(lmax/2,sizeof(double));
    thetae=calloc(lmax/2,sizeof(double));
    thetaei=calloc(lmax/2*bins,sizeof(double));
    thetaedot=calloc(lmax/2,sizeof(double));
    thetaetild=calloc(lmax/2,sizeof(double));
    ek_av=calloc(lmax/2,sizeof(double));
    if(ek_av==NULL||thetaei==NULL||thetae==NULL||thetaedot==NULL||thetaetild==NULL||vare==NU
LL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }

    strcat(strcpy(file,argv[1]),"korr.erg");
    fperg=fopen(file,"w");

    process(fperg,lmax,bins,vars,thetas,thetasi,thetasdot,thetastild,sk,sk_av,vare,thetae,th
etaei,thetaedot,thetaetild,ek,ek_av);

    fclose(fperg);

    free(sk);
    free(sk_av);
    free(vars);
    free(thetas);
    free(thetasi);
    free(thetasdot);
    free(thetastild);
    free(ek);
    free(ek_av);
    free(vare);
    free(thetae);
    free(thetaei);
    free(thetaedot);
    free(thetaetild);
  }
  else if (argc==6) {
    sstart=atoi(argv[2]);
    send=atoi(argv[3]);
    estart=atoi(argv[4]);
    eend=atoi(argv[5]);

    for(j=1; j<=send-sstart+1; j++)
      if((corrs[j]=calloc(send-sstart+2,sizeof(double)))==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    for(j=1; j<=eend-estart+1; j++)
      if((corre[j]=calloc(eend-estart+2,sizeof(double)))==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    thetastild=calloc(lmax/2,sizeof(double));
    thetaetild=calloc(lmax/2,sizeof(double));
    if(thetastild==NULL || thetaetild==NULL) {
```

```
      printf("Couldn’t allocate memory!");
      exit(1);
    }

    strcat(strcpy(file,argv[1]),"korr.erg");
    fperg=fopen(file,"r");

    for(j=DELTA; j<lmax/2-DELTA; j++) {
      fscanf(fperg,"%d\t",&dummy2);
      fscanf(fperg,"%lf\t%lf\t%lf\t%lf\t",thetastild+j,&dummy,&v,&dummy);
      fscanf(fperg,"%lf\t%lf\t%lf\t%lf\n",thetaetild+j,&dummy,&v,&dummy);
    }

    fclose(fperg);

    error(argv[1],lmax,bins,sstart,send,estart,eend,sk,corrs,ek,corre,thetastild,thetaetild)
;

    for(j=1; j<=send-sstart+1; j++)
      free(corrs[j]);
    for(j=1; j<=eend-estart+1; j++)
      free(corre[j]);

    free(sk);
    free(ek);
    free(thetastild);
    free(thetaetild);
  }
  else if(argc==4 || argc==3) {

    if(argc==4) {
      send=atoi(argv[2]);
      eend=atoi(argv[3]);
    }

    if(argc==3) {
      send=eend=lmax/2-DELTA-1;
      strcat(strcpy(file,argv[1]),"korr.erg");
      fperg=fopen(file,"r");
      for(j=DELTA; j<lmax/2-DELTA; j++) {
        fscanf(fperg,"%d\t",&dummy2);
        fscanf(fperg,"%lf\t%lf\t%lf\t%lf\t",&dummys,&dummy,&v,&dummy);
        fscanf(fperg,"%lf\t%lf\t%lf\t%lf\n",&dummye,&dummy,&v,&dummy);
        if(dummys<=0) {
          send=j-1;
          break;
        }
      }

      fclose(fperg);
      strcat(strcpy(file,argv[1]),"korr.erg");
      fperg=fopen(file,"r");
      for(j=DELTA; j<lmax/2-DELTA; j++) {
        fscanf(fperg,"%d\t",&dummy2);
        fscanf(fperg,"%lf\t%lf\t%lf\t%lf\t",&dummys,&dummy,&v,&dummy);
        fscanf(fperg,"%lf\t%lf\t%lf\t%lf\n",&dummye,&dummy,&v,&dummy);
        if(dummye<=0) {
          eend=j-1;
          break;
        }
      }
      fclose(fperg);
    }

    for(j=1; j<=send-DELTA+1; j++)
      if((corrs[j]=calloc(send-DELTA+2,sizeof(double)))==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    for(j=1; j<=eend-DELTA+1; j++)
      if((corre[j]=calloc(eend-DELTA+2,sizeof(double)))==NULL) {
        printf("Couldn’t allocate memory!");
        exit(1);
      }
    thetastild=calloc(lmax/2,sizeof(double));
    thetaetild=calloc(lmax/2,sizeof(double));
    if(thetastild==NULL||thetaetild==NULL) {
      printf("Couldn’t allocate memory!");
      exit(1);
    }

    strcat(strcpy(file,argv[1]),"korr.erg");
    fperg=fopen(file,"r");

    for(j=DELTA; j<lmax/2-DELTA; j++) {
      fscanf(fperg,"%d\t",&dummy2);
      fscanf(fperg,"%lf\t%lf\t%lf\t%lf\t",thetastild+j,&dummy,&v,&dummy);
      fscanf(fperg,"%lf\t%lf\t%lf\t%lf\n",thetaetild+j,&dummy,&v,&dummy);
    }

    fclose(fperg);

    error2(argv[1],lmax,bins,send,eend,sk,corrs,ek,corre,thetastild,thetaetild);

    for(j=1; j<=send-DELTA+1; j++)
      free(corrs[j]);
    for(j=1; j<=eend-DELTA+1; j++)
      free(corre[j]);
    free(sk);
    free(ek);
    free(thetastild);
    free(thetaetild);
  }
}

process(fperg,lmax,bins,vars,thetas,thetasi,thetasdot,thetastild,sk,sk_av,vare,thetae,thetae
i,thetaedot,thetaetild,ek,ek_av)
FILE *fperg;
int lmax, bins;
double  *vars, *thetas, *thetasi, *thetasdot, *thetastild, *sk, *sk_av, *vare, *thetae, *the
taei, *thetaedot, *thetaetild, *ek, *ek_av;
{
  int j, i, n;
  double vs, ve, s, d1, d2, e;
  double *sigd1s, *sigd2s, *sigxis, *sigd1e, *sigd2e, *sigxie;

  sigd1s=calloc(lmax/2,sizeof(double));
  sigd2s=calloc(lmax/2,sizeof(double));
  sigxis=calloc(lmax/2,sizeof(double));
  sigd1e=calloc(lmax/2,sizeof(double));
  sigd2e=calloc(lmax/2,sizeof(double));
  sigxie=calloc(lmax/2,sizeof(double));
  if(sigd1s==NULL || sigd2s==NULL || sigxis==NULL || sigd1e==NULL || sigd2e==NULL || sigxie=
=NULL) {
    printf("analyse: couldn’t allocate memory\n");
    exit(1);
  }

  for(j=0; j<lmax/2; j++) {
    *(vars+j)=0.0;
    *(vare+j)=0.0;
    *(sk_av+j)=0.0;
    *(ek_av+j)=0.0;
    *(thetasdot+j)=0.0;
```

```
    *(thetaedot+j)=0.0;
    sigd1s[j]=0.0;
    sigd2s[j]=0.0;
    sigd1e[j]=0.0;
    sigd2e[j]=0.0;
    sigxis[j]=0.0;
    sigxie[j]=0.0;
    for(n=0; n<bins; n++) {
      sk_av[j]+=*(sk+lmax/2*n+j);
      ek_av[j]+=*(ek+lmax/2*n+j);
    }
  }

  for(j=0; j<lmax/2; j++) {
    sk_av[j]/=bins;
    ek_av[j]/=bins;
  }

  for(j=DELTA; j<lmax/2-DELTA; j++) {
    for(i=0; i<bins; i++) {
      s=(*(sk+lmax/2*i+j)-*(sk+lmax/2*i+j-DELTA))-(*(sk_av+j)-*(sk_av+j-DELTA));
      sigd1s[j]+=s*s;
      s=(*(sk+lmax/2*i+j+DELTA)-*(sk+lmax/2*i+j))-(*(sk_av+j+DELTA)-*(sk_av+j));
      sigd2s[j]+=s*s;
      e=(*(ek+lmax/2*i+j)-*(ek+lmax/2*i+j-DELTA))-(*(ek_av+j)-*(ek_av+j-DELTA));
      sigd1e[j]+=e*e;
      e=(*(ek+lmax/2*i+j+DELTA)-*(ek+lmax/2*i+j))-(*(ek_av+j+DELTA)-*(ek_av+j));
      sigd2e[j]+=e*e;
    }
    sigd1s[j]=sqrt(sigd1s[j]/(bins*(bins-1)));
    sigd2s[j]=sqrt(sigd2s[j]/(bins*(bins-1)));
    d1=*(sk_av+j)-*(sk_av+j-DELTA);
    d2=*(sk_av+j+DELTA)-*(sk_av+j);
    if(!(d1==0 || d2==0 || d1/d2 <= 0))
      sigxis[j]=DELTA/(log(d1/d2)*log(d1/d2))*d2/d1*sqrt((sigd1s[j]/d2)*(sigd1s[j]/d2)+(d1/d
2*sigd2s[j]/d2)*(d1/d2*sigd2s[j]/d2));
    sigd1e[j]=sqrt(sigd1e[j]/(bins*(bins-1)));
    sigd2e[j]=sqrt(sigd2e[j]/(bins*(bins-1)));
    d1=*(ek_av+j)-*(ek_av+j-DELTA);
    d2=*(ek_av+j+DELTA)-*(ek_av+j);
    if(!(d1==0 || d2==0 || d1/d2 <= 0))
      sigxie[j]=DELTA/(log(d1/d2)*log(d1/d2))*d2/d1*sqrt((sigd1e[j]/d2)*(sigd1e[j]/d2)+(d1/d
2*sigd2e[j]/d2)*(d1/d2*sigd2e[j]/d2));
  }

  for(j=0; j<lmax/2; j++) {
    sk_av[j]*=bins;
    ek_av[j]*=bins;
  }

  for(i=0; i<bins; i++) {
    for(j=0; j<lmax/2; j++) {
      *(sk_av+j)-=*(sk+lmax/2*i+j);
      *(sk_av+j)/=bins-1;
      *(ek_av+j)-=*(ek+lmax/2*i+j);
      *(ek_av+j)/=bins-1;
     }
    for(j=DELTA; j<lmax/2-DELTA; j++) {
      *(thetasi+i*lmax/2+j)=0.0;
      *(thetaei+i*lmax/2+j)=0.0;
      if(((sk_av[j+DELTA]-sk_av[j])!=0) && ((sk_av[j]-sk_av[j-DELTA])*(sk_av[j+DELTA]-sk_av[
j])>0))
        if((sk_av[j]-sk_av[j-DELTA])!=(sk_av[j+DELTA]-sk_av[j]))
          *(thetasi+i*lmax/2+j)=DELTA/log((*(sk_av+j)-*(sk_av+j-DELTA))/(*(sk_av+j+DELTA)-*(
sk_av+j)));
      *(thetasdot+j)+=*(thetasi+i*lmax/2+j)/bins;
      if(((ek_av[j+DELTA]-ek_av[j])!=0) && ((ek_av[j]-ek_av[j-DELTA])*(ek_av[j+DELTA]-ek_av[
j])>0))
        if((ek_av[j]-ek_av[j-DELTA])!=(ek_av[j+DELTA]-ek_av[j]))
          *(thetaei+i*lmax/2+j)=DELTA/log((*(ek_av+j)-*(ek_av+j-DELTA))/(*(ek_av+j+DELTA)-*(
ek_av+j)));
      *(thetaedot+j)+=*(thetaei+i*lmax/2+j)/bins;
    }
    for(j=0; j<lmax/2; j++) {
      *(sk_av+j)*=bins-1;
      *(sk_av+j)+=*(sk+lmax/2*i+j);
      *(ek_av+j)*=bins-1;
      *(ek_av+j)+=*(ek+lmax/2*i+j);
    }
  }

  for(j=0; j<lmax/2; j++) {
    sk_av[j]/=bins;
    ek_av[j]/=bins;
  }

  for(j=DELTA; j<lmax/2-DELTA; j++) {
    for(i=0; i<bins; i++) {
      vs=*(thetasi+i*lmax/2+j)-*(thetasdot+j);
      *(vars+j)+=vs*vs;
      ve=*(thetaei+i*lmax/2+j)-*(thetaedot+j);
      *(vare+j)+=ve*ve;
    }
    *(vars+j)*=(bins-1)/((double)bins);
    *(vare+j)*=(bins-1)/((double)bins);
  }

  for(j=DELTA; j<lmax/2-DELTA; j++) {
    *(thetas+j)=0;
    if(((sk_av[j+DELTA]-sk_av[j])!=0) && ((sk_av[j]-sk_av[j-DELTA])*(sk_av[j+DELTA]-sk_av[j]
)>0))
      if((sk_av[j]-sk_av[j-DELTA])!=(sk_av[j+DELTA]-sk_av[j]))
        *(thetas+j)=DELTA/log((*(sk_av+j)-*(sk_av+j-DELTA))/(*(sk_av+j+DELTA)-*(sk_av+j)));
    *(thetastild+j)=bins**(thetas+j)-(bins-1)**(thetasdot+j);
    *(thetae+j)=0;
    if(((ek_av[j+DELTA]-ek_av[j])!=0) && ((ek_av[j]-ek_av[j-DELTA])*(ek_av[j+DELTA]-ek_av[j]
)>0))
      if((ek_av[j]-ek_av[j-DELTA])!=(ek_av[j+DELTA]-ek_av[j]))
        *(thetae+j)=DELTA/log((*(ek_av+j)-*(ek_av+j-DELTA))/(*(ek_av+j+DELTA)-*(ek_av+j)))
;
    *(thetaetild+j)=bins**(thetae+j)-(bins-1)**(thetaedot+j);
    fprintf(fperg,"%3d\t",j);
    fprintf(fperg,"%.15lf\t%.15lf\t%.15lf\t%.15lf\t",*(thetas+j),*(thetastild+j),sqrt(*(vars
+j)),sigxis[j]);
    fprintf(fperg,"%.15lf\t%.15lf\t%.15lf\t%.15lf\n",*(thetae+j),*(thetaetild+j),sqrt(*(vare
+j)),sigxie[j]);
  }

  free(sigd1s);
  free(sigd2s);
  free(sigxis);
  free(sigd1e);
  free(sigd2e);
  free(sigxie);
}

error(path,lmax,bins,sstart,send,estart,eend,sk,corrs,ek,corre,thetastild,thetaetild)
char *path;
int lmax, bins, sstart, send, estart, eend;
double *sk, **corrs, *ek, **corre, *thetastild, *thetaetild;
{
  int k, j, i, n;
  int *indexs, *indexe;
  double d, chi2s, chi2e;
```

```
  double *cs, *ce;
  double xis_av, xie_av, sums, sume, sumns, sumne;
  double xis_avn, xie_avn;
  double thetasm, thetasmdot, thetaem, thetaemdot;
  double vsm, vem, varsm, varem;
  double *thetasi, *thetasdot, *thetaei, *thetaedot;
  double *thetasmi, *thetaemi, *sk_av, *ek_av;
  double *invcorrs[LMAXMAX2], *invcorre[LMAXMAX2];
  FILE *fpend;
  char file[100];

  chi2s=chi2e=xis_av=xie_av=sums=sume=sumne=sumns=xis_avn=xie_avn=varsm=varem=0.0;
  thetaemdot=thetasmdot=0.0;

  thetasi=calloc(bins*lmax/2,sizeof(double));
  thetaei=calloc(bins*lmax/2,sizeof(double));
  thetasmi=calloc(bins,sizeof(double));
  thetaemi=calloc(bins,sizeof(double));
  thetasdot=calloc(lmax/2,sizeof(double));
  thetaedot=calloc(lmax/2,sizeof(double));
  sk_av=calloc(lmax/2,sizeof(double));
  ek_av=calloc(lmax/2,sizeof(double));
  indexs=calloc(send-sstart+2,sizeof(int));
  indexe=calloc(eend-estart+2,sizeof(int));
  cs=calloc(send-sstart+2,sizeof(double));
  ce=calloc(eend-estart+2,sizeof(double));

  for(j=1; j<=send-sstart+1; j++) {
    if((invcorrs[j]=calloc(send-sstart+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    for(k=sstart; k<=send; k++)
      invcorrs[j][k-sstart+1]=0.0;
    invcorrs[j][j]=1;
  }

  for(j=1; j<=eend-estart+1; j++) {
    if((invcorre[j]=calloc(eend-estart+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    for(k=estart; k<=eend; k++)
      invcorre[j][k-estart+1]=0.0;
    invcorre[j][j]=1;
  }

  if(thetasi==NULL||thetaei==NULL||thetasdot==NULL||thetaedot==NULL||thetasmi==NULL||thetaem
i==NULL||sk_av==NULL||ek_av==NULL||indexs==NULL||indexe==NULL||cs==NULL||ce==NULL) {
    printf("Couldn't allocate memory!");
    exit(1);
  }

  for(j=sstart-DELTA; j<=send+DELTA; j++) {
    *(sk_av+j)=0.0;
    *(thetasdot+j)=0.0;
    for(n=0; n<bins; n++)
      *(sk_av+j)+=*(sk+lmax/2*n+j);
  }
  for(j=estart-DELTA; j<=eend+DELTA; j++) {
    *(ek_av+j)=0.0;
    *(thetaedot+j)=0.0;
    for(n=0; n<bins; n++)
      *(ek_av+j)+=*(ek+lmax/2*n+j);
  }

  for(i=0; i<bins; i++) {
    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      *(sk_av+j)-=*(sk+lmax/2*i+j);
      *(sk_av+j)/=bins-1;
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      *(ek_av+j)-=*(ek+lmax/2*i+j);
      *(ek_av+j)/=bins-1;
    }

    for(j=sstart; j<=send; j++) {
      *(thetasi+i*lmax/2+j)=DELTA/log((*(sk_av+j)-*(sk_av+j-DELTA))/(*(sk_av+j+DELTA)-*(sk_a
v+j)));
      *(thetasdot+j)+=*(thetasi+i*lmax/2+j)/bins;
    }
    for(j=estart; j<=eend; j++) {
      *(thetaei+i*lmax/2+j)=DELTA/log((*(ek_av+j)-*(ek_av+j-DELTA))/(*(ek_av+j+DELTA)-*(ek_a
v+j)));
      *(thetaedot+j)+=*(thetaei+i*lmax/2+j)/bins;
    }

    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      *(sk_av+j)*=bins-1;
      *(sk_av+j)+=*(sk+lmax/2*i+j);
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      *(ek_av+j)*=bins-1;
      *(ek_av+j)+=*(ek+lmax/2*i+j);
    }
  }

  for(j=sstart; j<=send; j++)
    for(k=j; k<=send; k++) {
      corrs[j-sstart+1][k-sstart+1]=0.0;
      for(i=0; i<bins; i++)
        corrs[j-sstart+1][k-sstart+1]+=(*(thetasi+i*lmax/2+j)-*(thetasdot+j))*(*(thetasi+i*l
max/2+k)-*(thetasdot+k));
      corrs[j-sstart+1][k-sstart+1]*=(bins-1)/((double)bins);
      corrs[k-sstart+1][j-sstart+1]=corrs[j-sstart+1][k-sstart+1];
    }

  for(j=estart; j<=eend; j++)
    for(k=j; k<=eend; k++) {
      corre[j-estart+1][k-estart+1]=0.0;
      for(i=0; i<bins; i++)
        corre[j-estart+1][k-estart+1]+=(*(thetaei+i*lmax/2+j)-*(thetaedot+j))*(*(thetaei+i*l
max/2+k)-*(thetaedot+k));
      corre[j-estart+1][k-estart+1]*=(bins-1)/((double)bins);
      corre[k-estart+1][j-estart+1]=corre[j-estart+1][k-estart+1];
    }

  for(j=sstart; j<=send; j++) {
    xis_avn+=*(thetastild+j)/corrs[j-sstart+1][j-sstart+1];
    sumns+=1/corrs[j-sstart+1][j-sstart+1];
  }
  xis_avn/=sumns;

  for(j=estart; j<=eend; j++) {
    xie_avn+=*(thetaetild+j)/corre[j-estart+1][j-estart+1];
    sumne+=1/corre[j-estart+1][j-estart+1];
  }
  xie_avn/=sumne;

  ludcmp(corrs,send-sstart+1,indexs,&d);
  for(j=1; j<=send-sstart+1; j++)
    lubksb(corrs,send-sstart+1,indexs,invcorrs[j]);

  ludcmp(corre,eend-estart+1,indexe,&d);
```

```
  for(j=1; j<=eend-estart+1; j++)
    lubksb(corre,eend-estart+1,indexe,invcorre[j]);

  for(j=sstart; j<=send; j++) {
    cs[j-sstart+1]=0.0;
    for(i=sstart; i<=send; i++)
      cs[j-sstart+1]+=invcorrs[j-sstart+1][i-sstart+1];
    sums+=cs[j-sstart+1];
    xis_av+=*(thetastild+j)*cs[j-sstart+1];
  }
  xis_av/=sums;

/*   for(j=sstart; j<=send; j++) */
/*     printf("%.15lf\t%.15lf\n",thetastild[j],cs[j-sstart+1]/sums); */

  for(j=estart; j<=eend; j++) {
    ce[j-estart+1]=0.0;
    for(i=estart; i<=eend; i++)
      ce[j-estart+1]+=invcorre[j-estart+1][i-estart+1];
    sume+=ce[j-estart+1];
    xie_av+=*(thetaetild+j)*ce[j-estart+1];
  }
  xie_av/=sume;

  for(i=sstart; i<=send; i++)
    for(j=sstart; j<=send; j++)
      chi2s+=(thetastild[i]-xis_av)*(thetastild[j]-xis_av)*invcorrs[i-sstart+1][j-sstart+1];

  for(i=estart; i<=eend; i++)
    for(j=estart; j<=eend; j++)
      chi2e+=(thetaetild[i]-xie_av)*(thetaetild[j]-xie_av)*invcorre[i-estart+1][j-estart+1];

  for(i=0; i<bins; i++) {
    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      *(sk_av+j)-=*(sk+lmax/2*i+j);
      *(sk_av+j)/=bins-1;
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      *(ek_av+j)-=*(ek+lmax/2*i+j);
      *(ek_av+j)/=bins-1;
    }

    *(thetasmi+i)=0.0;
    for(j=sstart; j<=send; j++)
      *(thetasmi+i)+=cs[j-sstart+1]*DELTA/log((*(sk_av+j)-*(sk_av+j-DELTA))/(*(sk_av+j+DELTA
)-*(sk_av+j)));
    *(thetasmi+i)/=sums;
    thetasmdot+=*(thetasmi+i)/bins;

    *(thetaemi+i)=0.0;
    for(j=estart; j<=eend; j++)
      *(thetaemi+i)+=ce[j-estart+1]*DELTA/log((*(ek_av+j)-*(ek_av+j-DELTA))/(*(ek_av+j+DELTA
)-*(ek_av+j)));
    *(thetaemi+i)/=sume;
    thetaemdot+=*(thetaemi+i)/bins;

    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      *(sk_av+j)*=bins-1;
      *(sk_av+j)+=*(sk+lmax/2*i+j);
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      *(ek_av+j)*=bins-1;
      *(ek_av+j)+=*(ek+lmax/2*i+j);
    }
  }

  for(i=0; i<bins; i++) {
    vsm=*(thetasmi+i)-thetasmdot;
    varsm+=vsm*vsm;
    vem=*(thetaemi+i)-thetaemdot;
    varem+=vem*vem;
  }
  varsm=sqrt(varsm*(bins-1)/((double)bins));
  varem=sqrt(varem*(bins-1)/((double)bins));


  printf("xis_corrweighted = %10.10lf  (Werte %3d -%3d)\n",xis_av,sstart,send);
  printf("sd_jack(xis)     = %10.10lf\n",sqrt(1/sums));
  printf("sd_jck2(xis)     = %10.10lf\n\n",varsm);
  printf("xis_varweighted  = %10.10lf\n",xis_avn);
  printf("sd_jack(xis)     = %10.10lf\n\n",sqrt(1/sumns));
  printf("chi_s_squared    = %10.10lf\n",chi2s);
  printf("chi_s2/DegOFree  = %10.10lf\n\n\n",chi2s/(send-sstart));
  printf("xie_corrweighted = %10.10lf  (Werte %3d -%3d)\n",xie_av,estart,eend);
  printf("sd_jack(xie)     = %10.10lf\n",sqrt(1/sume));
  printf("sd_jck2(xie)     = %10.10lf\n\n",varem);
  printf("xie_varweighted  = %10.10lf\n",xie_avn);
  printf("sd_jack(xie)     = %10.10lf\n\n",sqrt(1/sumne));
  printf("chi_e_squared    = %10.10lf\n",chi2e);
  printf("chi_e2/DegOFree  = %10.10lf\n\n\n",chi2e/(eend-estart));
  printf("xis/xie          = %10.10lf\n",xis_av/xie_av);
  printf("sd_jack(xis/xie) = %10.10lf\n",sqrt((sqrt(1/sums)/xie_av)*(sqrt(1/sums)/xie_av)+(x
is_av*sqrt(1/sume)/(xie_av*xie_av))*(xis_av*sqrt(1/sume)/(xie_av*xie_av))));
  printf("sd_jck2(xis/xie) = %10.10lf\n",sqrt((varsm/xie_av)*(varsm/xie_av)+(xis_av*varem/(x
ie_av*xie_av))*(xis_av*varem/(xie_av*xie_av))));

  strcat(strcpy(file,path),"final");
  fpend=fopen(file,"w");
  fprintf(fpend,"%.15lf\t%.15lf\t%.15lf\t%.15lf\n",xis_av,sqrt(1/sums),xie_av,sqrt(1/sume));
  fclose(fpend);

  free(thetasmi);
  free(thetaemi);
  free(thetasi);
  free(thetaei);
  free(thetasdot);
  free(thetaedot);
  free(sk_av);
  free(ek_av);
  for(j=1; j<=send-sstart+1; j++)
    free(invcorrs[j]);
  for(j=1; j<=eend-estart+1; j++)
    free(invcorre[j]);
  free(cs);
  free(ce);
  free(indexs);
  free(indexe);
}

error2(path,lmax,bins,send,eend,sk,corrs,ek,corre,thetastild,thetaetild)
char *path;
int lmax, bins, send, eend;
double *sk, **corrs, *ek, **corre, *thetastild, *thetaetild;
{
  int i, j, k, n, start, length;
  int ssopt[10], esopt[10], seopt[10], eeopt[10];
  int *indexs, *indexe;
  double d, chi2s, chi2smin[10], chi2e, chi2emin[10];
  double xis_av, xie_av, sums, sume;
  double *act_corrs[LMAXMAX2], *act_corre[LMAXMAX2];
  double *act_invcorrs[LMAXMAX2], *act_invcorre[LMAXMAX2];
  double *sk_av, *thetasi, *thetasdot, *ek_av, *thetaei, *thetaedot;
  double *cs, *ce;
  char file[100];
```

```
  FILE *chifile;

  for(j=1; j<=send-DELTA+1; j++) {
    if((act_corrs[j]=calloc(send-DELTA+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    if((act_invcorrs[j]=calloc(send-DELTA+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
  }
  for(j=1; j<=eend-DELTA+1; j++) {
    if((act_corre[j]=calloc(eend-DELTA+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    if((act_invcorre[j]=calloc(eend-DELTA+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
  }

  thetasi=calloc(bins*lmax/2,sizeof(double));
  thetaei=calloc(bins*lmax/2,sizeof(double));
  thetasdot=calloc(lmax/2,sizeof(double));
  thetaedot=calloc(lmax/2,sizeof(double));
  sk_av=calloc(lmax/2,sizeof(double));
  ek_av=calloc(lmax/2,sizeof(double));

  indexs=calloc(send-DELTA+2,sizeof(int));
  indexe=calloc(eend-DELTA+2,sizeof(int));
  cs=calloc(send-DELTA+2,sizeof(double));
  ce=calloc(eend-DELTA+2,sizeof(double));

  if(thetasi==NULL||thetaei==NULL||thetasdot==NULL||thetaedot==NULL||sk_av==NULL||ek_av==NUL
L||indexs==NULL||indexe==NULL||cs==NULL||ce==NULL) {
    printf("Couldn't allocate memory!");
    exit(1);
  }

  for(j=0; j<=send+DELTA; j++) {
    *(sk_av+j)=0.0;
    *(thetasdot+j)=0.0;
    for(n=0; n<bins; n++)
      *(sk_av+j)+=*(sk+lmax/2*n+j);
  }
  for(j=0; j<=eend+DELTA; j++) {
    *(ek_av+j)=0.0;
    *(thetaedot+j)=0.0;
    for(n=0; n<bins; n++)
      *(ek_av+j)+=*(ek+lmax/2*n+j);
  }

  for(i=0; i<bins; i++) {
    for(j=0; j<=send+DELTA; j++) {
      *(sk_av+j)-=*(sk+lmax/2*i+j);
      *(sk_av+j)/=bins-1;
    }
    for(j=0; j<=eend+DELTA; j++) {
      *(ek_av+j)-=*(ek+lmax/2*i+j);
      *(ek_av+j)/=bins-1;
    }

    for(j=DELTA; j<=send; j++) {
      *(thetasi+i*lmax/2+j)=DELTA/log((*(sk_av+j)-*(sk_av+j-DELTA))/(*(sk_av+j+DELTA)-*(sk_a
v+j)));
      *(thetasdot+j)+=*(thetasi+i*lmax/2+j)/bins;
    }
    for(j=DELTA; j<=eend; j++) {
      *(thetaei+i*lmax/2+j)=DELTA/log((*(ek_av+j)-*(ek_av+j-DELTA))/(*(ek_av+j+DELTA)-*(ek_a
v+j)));
      *(thetaedot+j)+=*(thetaei+i*lmax/2+j)/bins;
    }

    for(j=0; j<=send+DELTA; j++) {
      *(sk_av+j)*=bins-1;
      *(sk_av+j)+=*(sk+lmax/2*i+j);
    }
    for(j=0; j<=eend+DELTA; j++) {
      *(ek_av+j)*=bins-1;
      *(ek_av+j)+=*(ek+lmax/2*i+j);
    }
  }

  for(j=DELTA; j<=send; j++)
    for(k=j; k<=send; k++) {
      corrs[j-DELTA+1][k-DELTA+1]=0.0;
      for(i=0; i<bins; i++)
        corrs[j-DELTA+1][k-DELTA+1]+=(*(thetasi+i*lmax/2+j)-*(thetasdot+j))*(*(thetasi+i*lma
x/2+k)-*(thetasdot+k));
      corrs[j-DELTA+1][k-DELTA+1]*=(bins-1)/((double)bins);
      corrs[k-DELTA+1][j-DELTA+1]=corrs[j-DELTA+1][k-DELTA+1];
    }

  for(j=DELTA; j<=eend; j++)
    for(k=j; k<=eend; k++) {
      corre[j-DELTA+1][k-DELTA+1]=0.0;
      for(i=0; i<bins; i++)
        corre[j-DELTA+1][k-DELTA+1]+=(*(thetaei+i*lmax/2+j)-*(thetaedot+j))*(*(thetaei+i*lma
x/2+k)-*(thetaedot+k));
      corre[j-DELTA+1][k-DELTA+1]*=(bins-1)/((double)bins);
      corre[k-DELTA+1][j-DELTA+1]=corre[j-DELTA+1][k-DELTA+1];
    }

  for(i=0; i<10; i++) {
    chi2smin[i]=chi2emin[i]=0;
    ssopt[i]=seopt[i]=esopt[i]=eeopt[i]=0;
  }

  strcat(strcpy(file,path),"chi2stable");
  chifile=fopen(file,"w");

  for(start=DELTA; start<=send-1; start++)
    for(length=1; length<=send-start; length++) {
      chi2s=xis_av=sums=0.0;
      d=0.0;
      for(i=start; i<=start+length; i++) {
        indexs[i-start+1]=0;
        for(j=start; j<=start+length; j++) {
          act_corrs[i-start+1][j-start+1]=corrs[i-DELTA+1][j-DELTA+1];
          act_invcorrs[i-start+1][j-start+1]=(i==j);
        }
      }
      ludcmp(act_corrs,length+1,indexs,&d);
      for(j=1; j<=length+1; j++)
        lubksb(act_corrs,length+1,indexs,act_invcorrs[j]);
      for(j=start; j<=start+length; j++) {
        cs[j-start+1]=0.0;
        for(i=start; i<=start+length; i++)
          cs[j-start+1]+=act_invcorrs[j-start+1][i-start+1];
        sums+=cs[j-start+1];
        xis_av+=*(thetastild+j)*cs[j-start+1];
      }
```

```
      xis_av/=sums;
      for(i=start; i<=start+length; i++)
        for(j=start; j<=start+length; j++)
          chi2s+=(thetastild[i]-xis_av)*(thetastild[j]-xis_av)*act_invcorrs[i-start+1][j-sta
rt+1];
      chi2s/=length;
      fprintf(chifile,"%d\t%d\t%.10lf\n",start,start+length,chi2s);
      for(i=0; i<10; i++)
        if((fabs(chi2s-1) <= fabs(chi2smin[i]-1)) || (chi2smin[i]==0)) {
         if(chi2smin[i]!=0)
           for(j=8; j>=i; j--) {
             chi2smin[j+1]=chi2smin[j];
             ssopt[j+1]=ssopt[j];
             seopt[j+1]=seopt[j];
           }
         chi2smin[i]=chi2s;
         ssopt[i]=start;
         seopt[i]=start+length;
         break;
        }
    }

  fclose(chifile);
  strcat(strcpy(file,path),"chi2etable");
  chifile=fopen(file,"w");

  for(start=DELTA; start<=eend-1; start++)
    for(length=1; length<=eend-start; length++) {
      chi2e=xie_av=sume=0.0;
      d=0.0;
      for(i=start; i<=start+length; i++) {
        indexe[i-start+1]=0;
        for(j=start; j<=start+length; j++) {
          act_corre[i-start+1][j-start+1]=corre[i-DELTA+1][j-DELTA+1];
          act_invcorre[i-start+1][j-start+1]=(i==j);
        }
      }
      ludcmp(act_corre,length+1,indexe,&d);
      for(j=1; j<=length+1; j++)
        lubksb(act_corre,length+1,indexe,act_invcorre[j]);
      for(j=start; j<=start+length; j++) {
        ce[j-start+1]=0.0;
        for(i=start; i<=start+length; i++)
          ce[j-start+1]+=act_invcorre[j-start+1][i-start+1];
      sume+=ce[j-start+1];
      xie_av+=*(thetaetild+j)*ce[j-start+1];
      }
      xie_av/=sume;
      for(i=start; i<=start+length; i++)
        for(j=start; j<=start+length; j++)
          chi2e+=(thetaetild[i]-xie_av)*(thetaetild[j]-xie_av)*act_invcorre[i-start+1][j-sta
rt+1];
      chi2e/=length;
      fprintf(chifile,"%d\t%d\t%.10lf\n",start,start+length,chi2e);
      for(i=0; i<10; i++)
        if((fabs(chi2e-1) <= fabs(chi2emin[i]-1)) || (chi2emin[i]==0)) {
          if(chi2emin[i]!=0)
            for(j=8; j>=i; j--) {
              chi2emin[j+1]=chi2emin[j];
              esopt[j+1]=esopt[j];
              eeopt[j+1]=eeopt[j];
            }
          chi2emin[i]=chi2e;
          esopt[i]=start;
          eeopt[i]=start+length;
          break;
        }
    }

  fclose(chifile);

  printf("\nOptimale Parameterwahl:\n\n");
  for(i=0; i<10; i++)
    printf("sstart: %3d, send: %3d, chi2s/DegOFree: %10.10lf, |chi2s-1|: %10.10lf\n",ssopt[i
],seopt[i],chi2smin[i],fabs(chi2smin[i]-1));
  printf("\n");
  for(i=0; i<10; i++)
    printf("estart: %3d, eend: %3d, chi2e/DegOFree: %10.10lf, |chi2e-1|: %10.10lf\n",esopt[i
],eeopt[i],chi2emin[i],fabs(chi2emin[i]-1));
  printf("\n");

  free(thetasi);
  free(thetaei);
  free(thetasdot);
  free(thetaedot);
  free(sk_av);
  free(ek_av);
  for(j=1; j<=send-DELTA+1; j++) {
    free(act_invcorrs[j]);
    free(act_corrs[j]);
  }
  for(j=1; j<=eend-DELTA+1; j++) {
    free(act_invcorre[j]);
    free(act_corre[j]);
  }
  free(cs);
  free(ce);
  free(indexs);
  free(indexe);
}

error_dj(path,lmax,bins,sstart,send,estart,eend,sk,corrs,ek,corre,thetastild,thetaetild)
char *path;
int lmax, bins, sstart, send, estart, eend;
double *sk, **corrs, *ek, **corre, *thetastild, *thetaetild;
{
  int k, j, i, n, i1, i2;
  int *indexs, *indexe;
  double d, chi2s, chi2e;
  double *cs, *ce;
  double xis_av, xie_av, sums, sume, sumns, sumne;
  double xis_avn, xie_avn;
  double thetasm, thetasmdot, thetaem, thetaemdot;
  double vsm, vem, varsm, varem;
  double *thetasmi, *thetaemi, *sk_av, *ek_av;
  double *thetastildi, *thetaetildi, *thetastilddot, *thetaetilddot;
  double *invcorrs[LMAXMAX2], *invcorre[LMAXMAX2];
  FILE *fpend;
  char file[100];

  chi2s=chi2e=xis_av=xie_av=sums=sume=sumne=sumns=xis_avn=xie_avn=varsm=varem=0.0;
  thetaemdot=thetasmdot=0.0;

  thetasmi=calloc(bins,sizeof(double));
  thetaemi=calloc(bins,sizeof(double));
  thetastildi=calloc(bins*lmax/2,sizeof(double));
  thetaetildi=calloc(bins*lmax/2,sizeof(double));
  thetastilddot=calloc(lmax/2,sizeof(double));
  thetaetilddot=calloc(lmax/2,sizeof(double));
  sk_av=calloc(lmax/2,sizeof(double));
  ek_av=calloc(lmax/2,sizeof(double));
  indexs=calloc(send-sstart+2,sizeof(int));
  indexe=calloc(eend-estart+2,sizeof(int));
  cs=calloc(send-sstart+2,sizeof(double));
```

```
  ce=calloc(eend-estart+2,sizeof(double));

  for(j=1; j<=send-sstart+1; j++) {
    if((invcorrs[j]=calloc(send-sstart+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    for(k=sstart; k<=send; k++)
      invcorrs[j][k-sstart+1]=0.0;
    invcorrs[j][j]=1;
  }

  for(j=1; j<=eend-estart+1; j++) {
    if((invcorre[j]=calloc(eend-estart+2,sizeof(double)))==NULL) {
      printf("Couldn't allocate memory!");
      exit(1);
    }
    for(k=estart; k<=eend; k++)
      invcorre[j][k-estart+1]=0.0;
    invcorre[j][j]=1;
  }

  if(thetastildi==NULL||thetaetildi==NULL||thetastilddot==NULL||thetaetilddot==NULL||thetasm
i==NULL||thetaemi==NULL||sk_av==NULL||ek_av==NULL||indexs==NULL||indexe==NULL||cs==NULL||ce=
=NULL) {
    printf("Couldn't allocate memory!");
    exit(1);
  }

  for(j=sstart-DELTA; j<=send+DELTA; j++) {
    *(sk_av+j)=0.0;
    *(thetastilddot+j)=0.0;
    for(n=0; n<bins; n++)
      *(sk_av+j)+=*(sk+lmax/2*n+j);
  }
  for(j=estart-DELTA; j<=eend+DELTA; j++) {
    *(ek_av+j)=0.0;
    *(thetaetilddot+j)=0.0;
    for(n=0; n<bins; n++)
      *(ek_av+j)+=*(ek+lmax/2*n+j);
  }

  for(i1=0; i1<bins; i1++) {
    for(j=sstart-DELTA; j<=send+DELTA; j++)
      *(sk_av+j)-=*(sk+lmax/2*i1+j);
    for(j=estart-DELTA; j<=eend+DELTA; j++)
      *(ek_av+j)-=*(ek+lmax/2*i1+j);

    for(j=sstart; j<=send; j++)
      *(thetastildi+i1*lmax/2+j)=DELTA*(bins-1)/log((sk_av[j]-sk_av[j-DELTA])/(sk_av[j+DELTA
]-sk_av[j]));
    for(j=estart; j<=eend; j++)
      *(thetaetildi+i1*lmax/2+j)=DELTA*(bins-1)/log((ek_av[j]-ek_av[j-DELTA])/(ek_av[j+DELTA
]-ek_av[j]));

    for(i2=0; i2<bins; i2++)
      if(i2!=i1) {
        for(j=sstart-DELTA; j<=send+DELTA; j++)
          *(sk_av+j)-=*(sk+lmax/2*i2+j);
        for(j=estart-DELTA; j<=eend+DELTA; j++)
          *(ek_av+j)-=*(ek+lmax/2*i2+j);

        for(j=sstart; j<=send; j++)
          *(thetastildi+i1*lmax/2+j)-=DELTA/log((sk_av[j]-sk_av[j-DELTA])/(sk_av[j+DELTA]-sk
_av[j]))*(bins-2)/((double)(bins-1));
        for(j=estart; j<=eend; j++)
          *(thetaetildi+i1*lmax/2+j)-=DELTA/log((ek_av[j]-ek_av[j-DELTA])/(ek_av[j+DELTA]-ek
_av[j]))*(bins-2)/((double)(bins-1));

        for(j=sstart-DELTA; j<=send+DELTA; j++)
          sk_av[j]+=*(sk+lmax/2*i2+j);
        for(j=estart-DELTA; j<=eend+DELTA; j++)
          ek_av[j]+=*(ek+lmax/2*i2+j);
      }

    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      sk_av[j]+=*(sk+lmax/2*i1+j);
      if((j>=sstart)&&(j<=send))
        *(thetastilddot+j)+=*(thetastildi+i1*lmax/2+j)/bins;
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      ek_av[j]+=*(ek+lmax/2*i1+j);
      if((j>=estart)&&(j<=eend))
        *(thetaetilddot+j)+=*(thetaetildi+i1*lmax/2+j)/bins;
    }
  }

  for(j=sstart; j<=send; j++)
    for(k=j; k<=send; k++) {
      corrs[j-sstart+1][k-sstart+1]=0.0;
      for(i=0; i<bins; i++)
        corrs[j-sstart+1][k-sstart+1]+=(*(thetastildi+i*lmax/2+j)-*(thetastilddot+j))*(*(the
tastildi+i*lmax/2+k)-*(thetastilddot+k));
      corrs[j-sstart+1][k-sstart+1]*=(bins-1)/((double)bins);
      corrs[k-sstart+1][j-sstart+1]=corrs[j-sstart+1][k-sstart+1];
    }

  for(j=estart; j<=eend; j++)
    for(k=j; k<=eend; k++) {
      corre[j-estart+1][k-estart+1]=0.0;
      for(i=0; i<bins; i++)
        corre[j-estart+1][k-estart+1]+=(*(thetaetildi+i*lmax/2+j)-*(thetaetilddot+j))*(*(the
taetildi+i*lmax/2+k)-*(thetaetilddot+k));
      corre[j-estart+1][k-estart+1]*=(bins-1)/((double)bins);
      corre[k-estart+1][j-estart+1]=corre[j-estart+1][k-estart+1];
    }

  for(j=sstart; j<=send; j++) {
    xis_avn+=*(thetastild+j)/corrs[j-sstart+1][j-sstart+1];
    sumns+=1/corrs[j-sstart+1][j-sstart+1];
  }
  xis_avn/=sumns;

  for(j=estart; j<=eend; j++) {
    xie_avn+=*(thetaetild+j)/corre[j-estart+1][j-estart+1];
    sumne+=1/corre[j-estart+1][j-estart+1];
  }
  xie_avn/=sumne;

  ludcmp(corrs,send-sstart+1,indexs,&d);
  for(j=1; j<=send-sstart+1; j++)
    lubksb(corrs,send-sstart+1,indexs,invcorrs[j]);

  ludcmp(corre,eend-estart+1,indexe,&d);
  for(j=1; j<=eend-estart+1; j++)
    lubksb(corre,eend-estart+1,indexe,invcorre[j]);

  for(j=sstart; j<=send; j++) {
    cs[j-sstart+1]=0.0;
    for(i=sstart; i<=send; i++)
      cs[j-sstart+1]+=invcorrs[j-sstart+1][i-sstart+1];
    sums+=cs[j-sstart+1];
    xis_av+=*(thetastild+j)*cs[j-sstart+1];
  }
```

```
  xis_av/=sums;

  for(j=estart; j<=eend; j++) {
    ce[j-estart+1]=0.0;
    for(i=estart; i<=eend; i++)
      ce[j-estart+1]+=invcorre[j-estart+1][i-estart+1];
    sume+=ce[j-estart+1];
    xie_av+=*(thetaetild+j)*ce[j-estart+1];
  }
  xie_av/=sume;

  for(i=sstart; i<=send; i++)
    for(j=sstart; j<=send; j++)
      chi2s+=(thetastild[i]-xis_av)*(thetastild[j]-xis_av)*invcorrs[i-sstart+1][j-sstart+1];

  for(i=estart; i<=eend; i++)
    for(j=estart; j<=eend; j++)
      chi2e+=(thetaetild[i]-xie_av)*(thetaetild[j]-xie_av)*invcorre[i-estart+1][j-estart+1];

  for(i=0; i<bins; i++) {
    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      *(sk_av+j)-=*(sk+lmax/2*i+j);
      *(sk_av+j)/=bins-1;
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      *(ek_av+j)-=*(ek+lmax/2*i+j);
      *(ek_av+j)/=bins-1;
    }

    *(thetasmi+i)=0.0;
    for(j=sstart; j<=send; j++)
      *(thetasmi+i)+=cs[j-sstart+1]*DELTA/log((*(sk_av+j)-*(sk_av+j-DELTA))/(*(sk_av+j+DELTA
)-*(sk_av+j)));
    *(thetasmi+i)/=sums;
    thetasmdot+=*(thetasmi+i)/bins;

    *(thetaemi+i)=0.0;
    for(j=estart; j<=eend; j++)
      *(thetaemi+i)+=ce[j-estart+1]*DELTA/log((*(ek_av+j)-*(ek_av+j-DELTA))/(*(ek_av+j+DELTA
)-*(ek_av+j)));
    *(thetaemi+i)/=sume;
    thetaemdot+=*(thetaemi+i)/bins;

    for(j=sstart-DELTA; j<=send+DELTA; j++) {
      *(sk_av+j)*=bins-1;
      *(sk_av+j)+=*(sk+lmax/2*i+j);
    }
    for(j=estart-DELTA; j<=eend+DELTA; j++) {
      *(ek_av+j)*=bins-1;
      *(ek_av+j)+=*(ek+lmax/2*i+j);
    }
  }

  for(i=0; i<bins; i++) {
    vsm=*(thetasmi+i)-thetasmdot;
    varsm+=vsm*vsm;
    vem=*(thetaemi+i)-thetaemdot;
    varem+=vem*vem;
  }
  varsm=sqrt(varsm*(bins-1)/((double)bins));
  varem=sqrt(varem*(bins-1)/((double)bins));


  printf("xis_corrweighted = %10.10lf  (Werte %3d -%3d)\n",xis_av,sstart,send);
  printf("sd_jack(xis)     = %10.10lf\n",sqrt(1/sums));
  printf("sd_jck2(xis)     = %10.10lf\n\n",varsm);
  printf("xis_varweighted  = %10.10lf\n",xis_avn);
  printf("sd_jack(xis)     = %10.10lf\n\n",sqrt(1/sumns));
  printf("chi_s_squared    = %10.10lf\n",chi2s);
  printf("chi_s2/DegOFree  = %10.10lf\n\n\n",chi2s/(send-sstart));
  printf("xie_corrweighted = %10.10lf  (Werte %3d -%3d)\n",xie_av,estart,eend);
  printf("sd_jack(xie)     = %10.10lf\n",sqrt(1/sume));
  printf("sd_jck2(xie)     = %10.10lf\n\n",varem);
  printf("xie_varweighted  = %10.10lf\n",xie_avn);
  printf("sd_jack(xie)     = %10.10lf\n\n",sqrt(1/sumne));
  printf("chi_e_squared    = %10.10lf\n",chi2e);
  printf("chi_e2/DegOFree  = %10.10lf\n\n\n",chi2e/(eend-estart));
  printf("xis/xie          = %10.10lf\n",xis_av/xie_av);
  printf("sd_jack(xis/xie) = %10.10lf\n",sqrt((sqrt(1/sums)/xie_av)*(sqrt(1/sums)/xie_av)+(x
is_av*sqrt(1/sume)/(xie_av*xie_av))*(xis_av*sqrt(1/sume)/(xie_av*xie_av))));
  printf("sd_jck2(xis/xie) = %10.10lf\n",sqrt((varsm/xie_av)*(varsm/xie_av)+(xis_av*varem/(x
ie_av*xie_av))*(xis_av*varem/(xie_av*xie_av))));

  strcat(strcpy(file,path),"final.bias");
  fpend=fopen(file,"w");
  fprintf(fpend,"%.15lf\t%.15lf\t%.15lf\t%.15lf\n",xis_av,sqrt(1/sums),xie_av,sqrt(1/sume));
  fclose(fpend);

  free(thetasmi);
  free(thetaemi);
  free(thetastildi);
  free(thetaetildi);
  free(thetastilddot);
  free(thetaetilddot);
  free(sk_av);
  free(ek_av);
  for(j=1; j<=send-sstart+1; j++)
    free(invcorrs[j]);
  for(j=1; j<=eend-estart+1; j++)
    free(invcorre[j]);
  free(cs);
  free(ce);
  free(indexs);
  free(indexe);
}
```

Danksagung

Mein herzlicher Dank geht an alle, die zum Gelingen dieser Arbeit beigetragen haben, insbesondere an:

Herrn Prof. Dr. Kurt Binder für die Möglichkeit, die Arbeit in seiner Gruppe unter hervorragenden Bedingungen durchführen zu können.

Herrn Prof. Dr. Wolfhard Janke für die faszinierende Themenstellung und die hervorragende Betreuung der Arbeit. Keine Arbeitsbelastung konnte ihn davon abhalten, meinen Fragen und Problemen ein offenes Ohr zu schenken.

Den Professoren Dr. M. Henkel und Dr. J. Cardy für klärende Hinweise und Anmerkungen.

Den Mitgliedern der Gruppe KOMA 331 für die gute Arbeitsatmosphäre, insbesondere der DS9–Crew Alexandra, Harald und Oli und den übrigen Mitgliedern des „a.out“–Teams, Dirk, Mathias und Frank, für die sportliche Gestaltung der Arbeitspausen.

Meinen Eltern für die mentale und materielle Unterstützung während meines Studiums.