



KERNFORSCHUNGSANLAGE JÜLICH GmbH

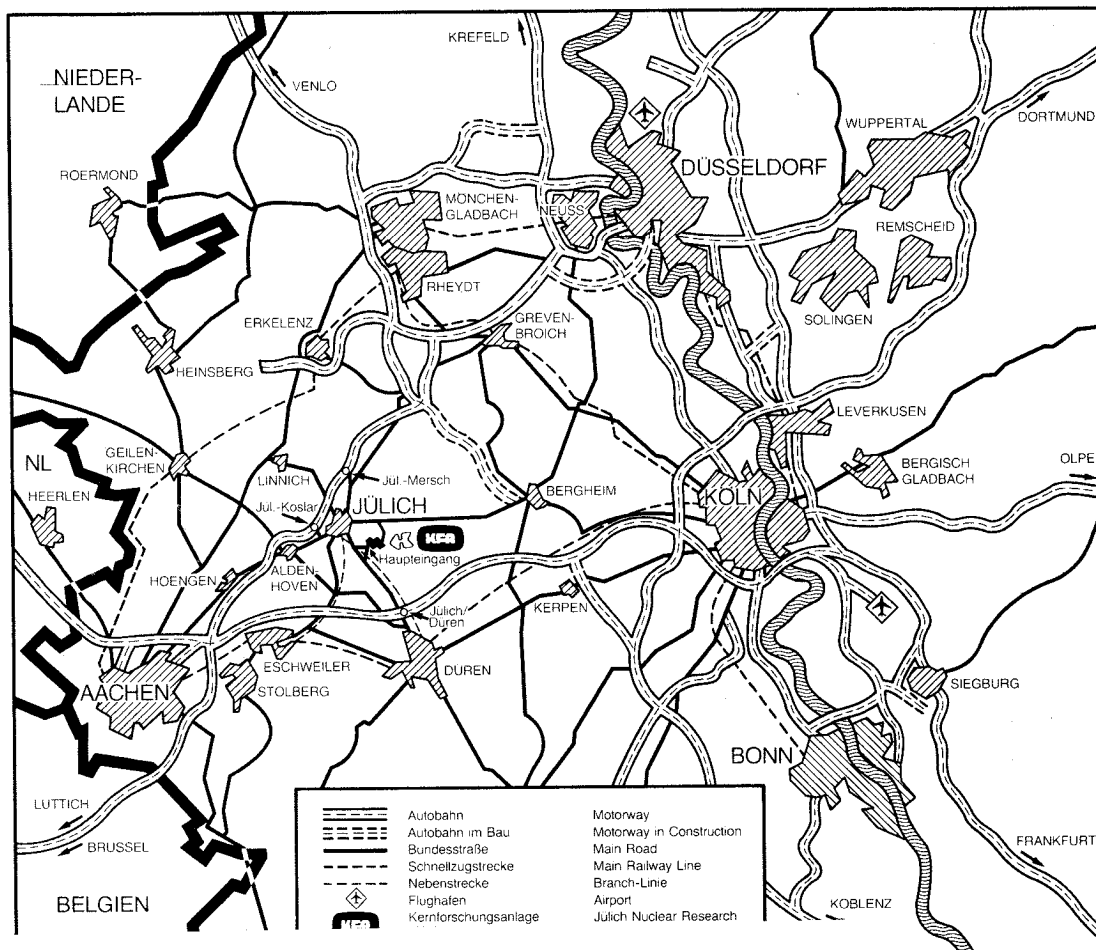
Institut für Festkörperforschung

**Zum Zeitverhalten des Einfangs
diffundierender Teilchen auf Gittern
mit statistisch verteilten Haftstellen**

von

Joachim Klaus Anlauf

Jül-2259
Januar 1989
ISSN 0366-0885



Als Manuskript gedruckt

Berichte der Kernforschungsanlage Jülich – Nr. 2259
 Institut für Festkörperforschung Jül-2259

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH
 Postfach 19 13 · D-5170 Jülich (Bundesrepublik Deutschland)
 Telefon: 02461/610 · Telex: 833556-0 kf d

Zum Zeitverhalten des Einfangs diffundierender Teilchen auf Gittern mit statistisch verteilten Haftstellen

von

Joachim Klaus Anlauf

Diss. Uni Köln

Summary

The trapping of diffusing particles on lattices with a random distribution of static traps is studied. A single particle performs a random walk in d dimensions and gets trapped immediately when it encounters a trap site for the first time. The traps are randomly distributed with concentration c . Some of the quantities studied in this thesis are the survival probability, the mean number of distinct sites visited and the mean square displacement after n steps. Asymptotic results for small and large step numbers are well known. In this thesis the crossover between the two regimes is studied in full detail for random walks on three dimensional lattices with a new Monte Carlo simulation method.

The contents of the thesis are as follows:

After an introduction in part I a detailed definition of the trapping problem is given in part II. For this purpose a discussion of the averaging process is needed, because two different types of randomness (random walks and statistically distributed traps) are involved. The known asymptotic results for the trapping problem are also collected in this part. Finally an exact correspondence between the trapping problem and the self-interacting-walk model (SIW) is described.

In part III an exact solution in one dimension is found for a correlated walk, where the probability of a step into a given direction depends on the direction of the last step performed. Asymptotically exact results are given for the distribution function for the number of distinct sites visited on a linear chain without any traps and for the survival probability on a chain with statistically distributed traps. These formulas include numerous correction terms to the leading behavior. Also an exact formula is given for the mean number of steps needed to find a site that was not visited before. In leading order some interesting scaling laws are found for this type of walk.

In higher dimensions no exact solution seems possible, but Monte Carlo simulations can be done to study the trapping problem. In part IV earlier simulation methods are analyzed. Since all are shown to have severe problems to reach the asymptotic regime of large step numbers, a new method is developed in this part of the thesis. It uses importance sampling to simulate effectively those walks that contribute most to the survival probability, and can realize much better statistics than any other known simulation method. To get small correlation times in the simulation a special algorithm is used that does not use a detailed balance condition. With the help of hash-coding techniques and a heap (although implemented in FORTRAN) many walks can be simulated simultaneously. This makes large parts of the algorithm vectorizable and guarantees an

effective use of vector computers (such as the CRAY X-MP in the Kernforschungsanlage Jülich).

The simulation results in three dimensions are given in part V, and scaling theories are used to analyze them. The crossover takes place at much lower step numbers and higher survival probabilities than expected previously, and can best be studied if one looks at the mean number of distinct sites visited in the presence of traps. There the crossover is much sharper than for the survival probability. A new and simple approximation for the survival probability in two and more dimensions emerges from the observed crossover behavior that is valid for all step numbers. The mean square displacement of surviving particles in the presence of traps shows the surprising result that, for small trap concentrations, there is a range of step numbers where the mean square displacement becomes smaller when the step number is increased. The final part VI contains a summary in German and some concluding remarks.

Although the survival probability at the crossover point is much higher than expected, it is still too small to be measured in present experiments. Moreover, even macroscopic lattices with $6 \cdot 10^{23}$ atoms are too small to show the necessary fluctuations in trap concentration that are needed to see the asymptotic stretched exponential decay of the survival probability, if the trap concentration is below 3%.

Zum Nachschlagen

Auf dieser Seite sind einige Symbole definiert und Beziehungen zusammengestellt, die in der Arbeit immer wieder benutzt werden. Damit ihre Bedeutung nicht jedesmal wiederholt werden muß, sind sie an zentraler Stelle aufgelistet, um jederzeit nachgeschlagen werden zu können.

d : Dimension des Gitters

n : Schrittzahl eines Random Walks

c : Haftstellenkonzentration, bzw. Wahrscheinlichkeit, daß ein gegebener Gitterplatz eine Haftstelle ist.

p, λ : Mit c verknüpfte Parameter. Es gelten die Beziehungen:

$$p = 1 - c = e^{-\lambda}$$

$$\lambda = -\ln(1 - c) = -\ln p$$

$$c = 1 - p = 1 - e^{-\lambda}$$

ω : Symbol für einen Walk, als Ganzes betrachtet

s : Anzahl der verschiedenen besuchten Plätze

Ψ_n : Überlebenswahrscheinlichkeit nach n Schritten

INHALTSVERZEICHNIS

I. Einleitung	1
II. Das Haftstellen-Problem	5
II.1. Haftstellen-Problem: Definition des Modells	5
II.2. Die Mittelung bei Anwesenheit von Haftstellen	6
II.3. Zusammenfassung der asymptotischen Theorie	9
II.4. Äquivalenz zwischen dem Haftstellenproblem und Self-Interacting Walks (SIW's)	12
III. Eindimensionaler korrelierter Random Walk mit Haftstellen	17
III.1. Korrelierte Walks auf endlichen Ketten mit absorbierenden Randbedingungen	19
III.2. Darstellung der Verteilungsfunktion $p_n(s)$	22
III.3. Einige exakte Ergebnisse	23
III.4. Asymptotische Verteilungsfunktion $p_n(s)$	26
III.5. Überlebenswahrscheinlichkeit	33
IV. Entwicklung eines neuen Monte-Carlo-Verfahrens für Haftstellen-Probleme in höheren Dimensionen	38
IV.1. Überblick über bekannte Simulationsverfahren	38
IV.2. Forderungen an neuen Algorithmus	45
IV.3. Realisierung der Forderungen in einem neuen MC-Verfahren	45
IV.4. Durchführung der Simulationen	70
V. Interpretation der Ergebnisse	75
V.1. Skalentheorie für $d \geq 3$	75
V.2. Mittlere Zahl der besuchten Plätze	77

V.3.	Die Überlebenswahrscheinlichkeit	79
V.4.	Der effektive Exponent α	84
V.5.	Die mittlere quadratische Auslenkung	85
V.6.	Die Rückkehrwahrscheinlichkeit	90
V.7.	Die Wärmekapazität	90
VI.	Zusammenfassung	93
 Anhänge:		
A.	Berechnung einiger spezieller Summen	96
B.	Beweis der Gl. (III.41)	98
C.	Asymptotische Reihen für einen speziellen Typ von Integralen	99
D.	Listing des Programms SIW3D	101
E.	Fehlerabschätzung von Monte Carlo Daten unter Berücksichtigung von Korrelationen	118
F.	Rekursionsformeln für $\langle s_n \rangle$ in $d = 2$ und $d = 3$	121
G.	Berechnung des effektiven Exponenten α als gewichteten Mittelwert	124
	 Literaturverzeichnis	 125

I. Einleitung

Stochastische Prozesse spielen in vielen Gebieten der Physik eine entscheidende Rolle. Typische Anwendungen sind der Transport von Teilchen in Materie, der häufig durch Random Walks beschrieben wird. Ein guter Übersichtsartikel über die Random Walk Theorie und ihre Anwendungen wurde von Weiss und Rubin [1] geschrieben.

Da in der Natur nur selten ideale Systeme realisiert sind, wächst in letzter Zeit das Interesse an ungeordneten Systemen. Der Review-Artikel von Haus und Kehr [2] ist eine gute Einführung in die Behandlung derartiger Probleme. In einer einheitlichen Notation findet man dort eine Fülle von Ergebnissen, sowohl für geordnete, als auch für ungeordnete Systeme. Ein weiterer, in gewisser Weise dazu komplementärer Übersichtsartikel von Havlin und Ben Avraham [3] befaßt sich mit einer etwas "exotischeren" Form der Unordnung, mit Random Walks auf Fraktalen.

Schon seit 1847 [4] ist immer wieder beobachtet worden, daß viele komplexe, langsam relaxierende Systeme ein Verhalten zeigen, das am besten mit einer "gestreckten Exponentialfunktion" (engl.: *stretched exponential*)

$$f(t) \sim e^{-(t/\tau)^\alpha}, \quad \alpha < 1$$

beschrieben werden kann [5]. Für diese anomale Relaxation wurde in letzter Zeit eine Klasse von Modellen mit "hierarchisch eingeschränkter Dynamik" vorgeschlagen [6], bei der gerade dieses Verhalten auf natürliche Weise entsteht.

Eine andere Art von Modellen erklärt das Verhalten mit einer parallelen Dynamik, bei der viele Relaxationsmoden so überlagert werden, daß die gestreckte Exponentialform entsteht. Es stellt sich heraus, daß das in dieser Arbeit zu untersuchende Haftstellen-Problem zu der letzten Klasse gehört.

Eine Theorie dieser Vorgänge hat unter anderem zur Aufgabe, den Wert des Exponenten α für die interessierenden Größen zu bestimmen. Dabei ist die Abhängigkeit von der Dimension des Modells von besonderem Interesse. Als einfacher Prototyp eines Relaxationsmodells bietet sich der Random Walk auf einem Gitter mit zufällig verteilten Haftstellen an.

Schon 1916 [7] wurde das Haftstellenproblem theoretisch untersucht. Damals wurde eine (zeitlich konstante) Einfangrate berechnet, was einer exponentiellen Relaxation entspricht. Rosenstock hat dann 1969 mit einer einfachen Näherung [8,9] den zeitlichen Verlauf der Lichtemission eines organischen Festkörpers mit Hilfe von Random Walks und Haftstellen für Exzitonen erklärt. Dabei wurde erstmalig für das Haftstellenproblem ein nichtexponentieller Zerfall, allerdings nur im ein- und zweidimensionalen

Fall, vorausgesetzt. Balagurov und Vaks [10] zeigten 1973 für Random Walks auf ein-, zwei- und dreidimensionalen Gittern, daß die Überlebenswahrscheinlichkeit der Teilchen jeweils einem gestreckten Exponentialgesetz folgt. Die Mathematiker Donsker und Varadhan bewiesen dann 1975 für Brownsche Bewegungen [11] und später auch für Random Walks auf Gittern [12], daß dieses Verhalten asymptotisch in d Dimensionen mit $\alpha = d/(d+2)$ gilt. Offenbar unabhängig davon wurde dieses Ergebnis 1982 von Grassberger und Procaccia einerseits [13], und Kayser und Hubbard andererseits [14], wiederentdeckt, indem im einen Fall eine untere, im anderen eine obere Schranke für die Überlebenswahrscheinlichkeit bestimmt wurden, die asymptotisch für große Schrittzahlen übereinstimmten.

In einer großen Zahl von Veröffentlichungen haben Blumen, Klafter und Zumofen [15–21] das Haftstellenproblem auf unterschiedlichsten Gittern simuliert und mit einer Kurzzeitentwicklung (Kumulantenentwicklung) verglichen, deren erster Term die Rosenstock-Näherung ist. Sie fanden zumeist eine gute Übereinstimmung für kurze Zeiten und kleine Haftstellenkonzentrationen. Das Langzeitverhalten von Balagurov und Vaks zu erhalten, ist jedoch sehr schwierig und gelingt mit ihrer Simulationsmethode praktisch nur auf niedrigdimensionalen Gittern. Ferner wurde von ihnen die Diffusion auf Fraktalen mit Haftstellen in diskreter Zeit [22] und in kontinuierlicher Zeit [19, 23–25] untersucht. Auch der Einfluß von Haftstellen auf die Diffusion in ultrametrischen Räumen [26] wurde diskutiert.

Ein weiterer Schritt in Richtung auf effektivere Simulationen gelang Havlin et. al. [27, 28], als sie 1984 eine neue Simulationsmethode einführten. Sie konnten zeigen, daß das Donsker und Varadhan Verhalten in $d = 2$ und $d = 3$ für Überlebenswahrscheinlichkeiten $\Psi(t)$ mit $\Psi(t) < 10^{-13}$ erfüllt ist (um die gleiche Zeit wurde von Fixman [29] gesagt, daß die Asymptotik erst bei 10^{-67} zu sehen ist). Da derzeit solche Überlebenswahrscheinlichkeiten nicht meßbar sind (zum Beispiel kommen Photostrom-Messungen nur bis etwa 10^{-4} hinab [30]), schien das asymptotische Verhalten für reale Systeme irrelevant zu sein.

Exakte Lösungen in einer Dimension [31, 32] und feldtheoretische Arbeiten [33, 34] in beliebigen Dimensionen zeigten, daß die Korrekturterme zum führenden Verhalten insbesondere in höheren Dimensionen für mittlere Zeiten noch ein sehr hohes Gewicht haben, weshalb es schwer ist, das reine asymptotische Verhalten zu sehen.

Bei Experimenten in höher-dimensionalen Systemen sind bisher keine Anhaltspunkte für ein nicht-exponentielles Verhalten gefunden worden. In eindimensionalen Systemen dagegen sind die Korrekturterme klein genug, um den asymptotischen Bereich

beobachten zu können. Photostrom-Messungen an linearen Polymeren konnten über mehrere Zehnerpotenzen an die asymptotische Theorie angepaßt werden [30, 35, 36].

In der Geschichte des Haftstellenproblems gab es auch einige Irrtümer. Stanley et. al. [37] sprachen 1983 von einem "neuen, superuniversellen Verhalten eines Random Walk Modells". Sie glaubten, durch exakte Abzählungen gezeigt zu haben, daß mehrere Exponenten beim Haftstellenproblem unabhängig von der Dimension seien, wenn die Haftstellenkonzentration an der Perkolationsschwelle liegt. Ferner nahmen sie an, daß der Exponent ν für die mittlere quadratische Auslenkung in der Formel $\langle R^2 \rangle \sim n^{2\nu}$ asymptotisch gegen 0 ginge. Sie waren jedoch mit ihren Abzählungen bei weitem nicht weit genug in den asymptotischen Bereich vorgestoßen. Beide Aussagen erwiesen sich später als falsch, denn Meirovitch [38] konnte 1985 mit einer ganz anderen Simulationsmethode wesentlich bessere Ergebnisse erzielen und so nahm er an, nun endgültig den asymptotischen Bereich für $\langle R^2 \rangle$ erreicht zu haben. Er verwarf sogar einen Datenpunkt, weil er ihn für einen Ausreißer hielt. So fand auch er einen viel zu kleinen Exponenten für die Abhängigkeit von $\langle R^2 \rangle$ von der Schrittzahl.

Einige Fragen und Probleme blieben seither offen. Die wichtigste ist die Frage nach dem Übergangspunkt vom Kurzzeit- (Rosenstock) zum Langzeitverhalten (Donsker und Varadhan) in höheren Dimensionen. Ist er in realistischen Systemen überhaupt zu beobachten? Wie kann man den intermediären Bereich beschreiben? Wie groß ist der Exponent ν für die mittlere quadratische Auslenkung?

Diese Fragen sollen in der vorliegenden Arbeit untersucht werden. Um mögliche Antworten in einer Simulation zu testen, ist aber ein wesentlich verbessertes Monte Carlo Verfahren notwendig, das als Teil der Arbeit entwickelt wird.

Die Arbeit gliedert sich deshalb in folgende Abschnitte:

Zunächst wird das Haftstellen-Problem in Teil II genau definiert, indem die verschiedenen Mittelwertbildungen beim Zusammentreffen von Random Walks und statistisch verteilten Haftstellen analysiert werden. Danach werden die bekannten, asymptotischen Ergebnisse in d Dimensionen zusammengestellt. Zuletzt wird das Haftstellen-Problem auf ein selbstwechselwirkendes Random Walk Modell abgebildet, das später eine effizientere Simulationsmöglichkeit eröffnen wird.

Da in einer Dimension das Haftstellenproblem exakt lösbar ist, wird in Teil III der korrelierte Random Walk auf einer linearen Kette näher betrachtet. Es werden unter anderem asymptotisch exakte Resultate für die Verteilungsfunktion der Zahl der besuchten Plätze und die Überlebenswahrscheinlichkeit auf einer Kette mit statistisch

verteilten Haftstellen berechnet.

In höheren Dimensionen gelingt eine exakte Behandlung des Problems nicht. Deshalb wird in Teil IV eine neue Methode zur Simulation des Haftstellenproblems entwickelt. Sie simuliert sehr effizient im wesentlichen nur die Walks, die am meisten zur Überlebenswahrscheinlichkeit beitragen (engl.: *importance sampling*) und ist dadurch den anderen, bekannten Verfahren deutlich überlegen.

Die mit dem Programm erzielten, zum Teil überraschenden Ergebnisse werden in Teil V dargestellt und mit Skalentheorien gedeutet. Dort werden dann auch die Antworten auf die oben gestellten Fragen gegeben.

Eine einordnende Zusammenfassung der Ergebnisse beschließt die Arbeit in Teil VI.

II. Das Haftstellen-Problem

Im folgenden soll zunächst das zu untersuchende Modell genau definiert werden, worauf sich einige allgemeine Bemerkungen über die Mittelung von Random Walk abhängigen Größen bei Anwesenheit von Haftstellen anschließen. Das führt zu zwei grundsätzlich verschiedenen Mittelungsmethoden, von denen dann im weiteren die eine untersucht werden wird. Es folgt eine knappe Darstellung der asymptotischen Theorie, die die Frage aufwirft, für welche Parameter die Theorie gültig wird. Die Abbildung des Haftstellen-Problems auf das Random Walk Modell des SIW's (Self-Interacting-Walk) erlaubt weitergehende Einsichten in das Modell und letztlich eine bessere Möglichkeit zur Computer-Simulation, auf die dann im übernächsten Kapitel eingegangen wird.

II.1. HAFTSTELLEN-PROBLEM: DEFINITION DES MODELLS

Es soll ein Teilchen betrachtet werden, das auf einem einfach kubischen, d dimensionalen Gitter einen Random Walk ausführt. Dabei soll es bei jedem Schritt mit derselben Wahrscheinlichkeit $W = 1/2d$ zu den $2d$ nächsten Nachbarn springen.

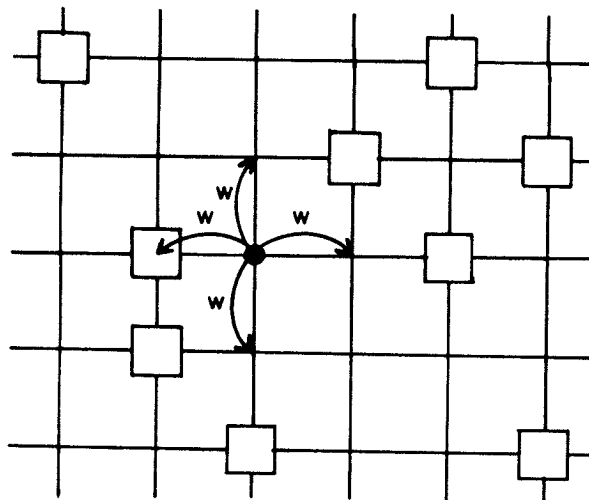


Bild II.1: Modell in $d = 2$. $\bullet$: diffundierendes Teilchen; $\square$: Haftstelle

Die Sprünge finden zu diskreten Zeitpunkten statt und die Schrittzahl wird mit n bezeichnet. Auf dem Gitter befinden sich zufällig verteilte, statische Haftstellen mit einer Konzentration c , d. h., die Wahrscheinlichkeit, daß ein gegebener Gitterplatz eine Haftstelle ist, ist c , und $p := 1 - c$ ist die Wahrscheinlichkeit, daß er ein normaler Gitterplatz ist. Das Teilchen startet an einem zufälligen Platz und diffundiert über das Gitter, bis es zum ersten Mal eine Haftstelle besucht. Dort wird es sofort eingefangen und steht für eine Fortführung des Random Walks nicht mehr zur Verfügung.

Eine interessierende Größe ist z.B. die Überlebenswahrscheinlichkeit Ψ_n , also die Wahrscheinlichkeit, daß das Teilchen nach n Schritten noch nicht auf eine Haftstelle gekommen ist. Ferner sind bei Anwesenheit von Haftstellen auch alle anderen Größen interessant, die schon bei normalen Random Walks untersucht worden sind. Das sind z.B. die mittlere quadratische Auslenkung, der Trägheitsradius, die Rückkehrwahrscheinlichkeit zum Startplatz, die Anzahl der verschiedenen besuchten Plätze und deren Schwankungen, wobei nur überlebende Walks berücksichtigt werden sollen. Im Experiment gemessen werden unter anderem die Mittelwerte dieser Größen. Da hier jedoch zwei statistische Effekte eine Rolle spielen (die Random Walk Mittelung und die Haftstellen-Konfigurationsmittelung), ist zunächst eine Betrachtung des Mittelungsprozesses notwendig.

II.2. DIE MITTELUNG BEI ANWESENHEIT VON HAFTSTELLEN

Im folgenden soll $f(\omega)$ eine beliebige Größe sein, die von einem Random Walk ω abhängt. Zunächst muß geklärt werden, was Mittelung von $f(\omega)$ bei Anwesenheit von Haftstellen bedeuten soll. Dazu gibt es grundsätzlich zwei verschiedene Möglichkeiten. In beiden Fällen sollen bei der Mittelung nur Walks berücksichtigt werden, die n Schritte überlebt haben.

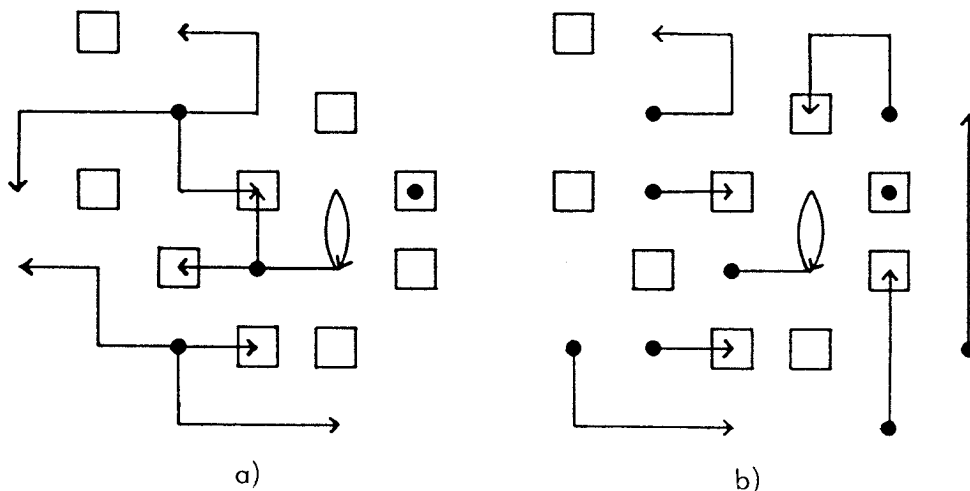


Bild II.2: Verschiedene Arten der Mittelung bei Anwesenheit von Haftstellen.

- (a) Viele Teilchen an einem Startplatz, anschließend Mittelung über die Startplätze
- (b) Jedes Teilchen an einem zufälligen Startplatz

*II.2.1. Viele Teilchen an einem Startplatz,
anschließend Mittelung über die Startplätze*

Das Teilchen findet beim Start eine zufällige Haftstellen-Konfiguration $\{T\}$ vor. Es überlebt nach n Schritten mit einer Wahrscheinlichkeit

$$\Psi_{\{T\}} = \frac{1}{(2d)^n} \sum'_{\omega} 1, \quad (II.1)$$

wobei der Strich an dem Summenzeichen bedeutet, daß nur über die Walks ω summiert wird, die in der gegebenen Haftstellen-Konfiguration $\{T\}$ überleben. Eine Größe $f(\omega)$ hat dann den Mittelwert in der gegebenen Haftstellen-Konfiguration $\{T\}$

$$\langle f \rangle_{\{T\}} = \frac{\sum'_{\omega} f(\omega)}{\sum'_{\omega} 1} = \frac{\frac{1}{(2d)^n} \sum'_{\omega} f(\omega)}{\Psi_{\{T\}}}. \quad (II.2)$$

Gemittelt über alle Haftstellen-Konfigurationen ergibt sich schließlich

$$\langle f \rangle = \sum_{\{T\}} p_{\{T\}} \langle f \rangle_{\{T\}} = \sum_{\{T\}} p_{\{T\}} \frac{\frac{1}{(2d)^n} \sum'_{\omega} f(\omega)}{\Psi_{\{T\}}}, \quad (II.3)$$

wobei $p_{\{T\}}$ die Wahrscheinlichkeit ist, die Haftstellen-Konfiguration $\{T\}$ vorzufinden. Diese Formel ist nicht weiter zu vereinfachen. Hier erhält jeder Random-Walk-Mittelwert — der als Zufallsvariable aufgefaßt wird, da er noch von der Haftstellenkonfiguration abhängt — dasselbe Gewicht im Endergebnis, obwohl an den verschiedenen Plätzen unterschiedlich viele Walks überleben. Bei einer solchen Betrachtungsweise können auch Fluktuationen bzgl. der Haftstellenkonfigurationsmittelung untersucht werden. Sie werden auch als Sinai-Fluktuationen [39, 40] bezeichnet. Eine solche Art der Mittelung ist in Experimenten jedoch meist nicht gegeben, da in der Regel alle überlebenden Teilchen ein bestimmtes Signal liefern und somit dasselbe Gewicht im Endergebnis haben. Dies führt uns zu der zweiten Möglichkeit zu mitteln.

II.2.2. Jedes Teilchen an einem zufälligen Startplatz

Das Teilchen findet am Start eine zufällige Haftstellen-Konfiguration vor und vollführt von dort aus nur *einen* Random Walk. Falls es die n Schritte überlebt hat, hat die zu mittelnde Größe den Wert $f(\omega)$. Jeder weitere Walk startet erneut an einem zufälligen Ort mit einer neuen, ungeordneten Umgebung. Es wird also simultan über alle Haftstellen-Konfigurationen $\{T\}$ und alle Walks ω gemittelt, d. h.,

$$\langle f \rangle = \frac{\sum_{\{T\}} \sum'_{\omega} p_{\{T\}} \frac{1}{(2d)^n} f(\omega)}{\sum_{\{T\}} \sum'_{\omega} p_{\{T\}} \frac{1}{(2d)^n}} = \frac{\sum_{\omega} f(\omega) \sum'_{\{T\}} p_{\{T\}}}{\sum_{\omega} \sum'_{\{T\}} p_{\{T\}}}. \quad (II.4)$$

Die Striche an den Summenzeichen im letzten Quotienten bedeuten Summation über die Haftstellen-Konfigurationen, in denen der gegebene Walk ω überlebt, d. h., all die Konfigurationen, in denen die s_ω Plätze, die der Walk besucht, keine Haftstellen sind. Damit ist $\sum'_{\{T\}} p_{\{T\}}$ einfach die Gesamtwahrscheinlichkeit, daß alle s_ω besuchten Plätze frei von Haftstellen sind, also

$$\sum'_{\{T\}} p_{\{T\}} = (1 - c)^{s_\omega} = e^{-\lambda s_\omega} \quad (II.5)$$

und deshalb

$$\langle f \rangle = \frac{\sum_\omega f(\omega) e^{-\lambda s_\omega}}{\sum_\omega e^{-\lambda s_\omega}}. \quad (II.6)$$

Die Vertauschung der Summationsreihenfolge in Gl. (II.4) ist problemlos, da beide Summen auf endlich viele Terme beschränkt werden können. Bei dieser Art der Mittelung geht jeder *Walk*, der überlebt, mit demselben Gewicht in Gl. (II.4) ein. Auf diese Weise muß man mitteln, wenn im Experiment die diffundierenden Teilchen an immer neuen Stellen starten, z. B. wenn die Teilchen in die Probe geschossen werden (z. B. Myonen) oder spontan an zufälligen Stellen entstehen (z. B. thermisch angeregte Gitterfehlstellen). Die Unordnung ist also eingefroren (engl.: *quenched*) für jeden einzelnen Walk. Jeder weitere Walk findet eine neue "*quenched*" Unordnung vor. Dadurch bekommt die Mittelung (II.6) einen ausgeheilten (engl.: *annealed*) Charakter, da jeder Walk nur noch mit einem statistischen Gewicht versehen wird, das ausdrückt, daß ein Walk um so unwahrscheinlicher überlebt, je mehr Plätze er besucht. Trotzdem werden die Fluktuationen in der Haftstellenkonzentration durch diese Vorgehensweise nicht ausgeschmiert. Es wird sich nämlich weiter unten zeigen, daß die Hauptbeiträge zu den Mittelwerten von Walks kommen, die in relativ großen, Haftstellen-freien Gebieten gestartet sind. In diesem Sinne ist die Mittelung nach wie vor "*quenched*".

Der wesentliche Punkt bei dieser Art der Mittelung ist, daß jedes überlebende, diffundierende Teilchen dasselbe Gewicht im Endergebnis hat. Auch in dem Fall, in dem Teilchen oft an denselben Stellen starten (z. B. thermisch angeregte Gitterfehlstellen an Versetzungslinien oder Exzitonen an Rekombinationszentren), ist diese zweite Art der Mittelung realisiert, sofern nur genügend viele solcher Startplätze in wechselnden Umgebungen vorhanden sind und die Anzahl der jeweils startenden Teilchen nicht mit der Umgebung des Startplatzes korreliert ist.

Diese Art der Mittelung bietet ferner den Vorteil, daß über die Wahrscheinlichkeitsverteilung der Haftstellen-Konfigurationen schon vorher exakt gemittelt werden

kann und nur noch Eigenschaften des Random Walk, nämlich die Anzahl der besuchten Plätze, berücksichtigt werden müssen.

Im folgenden ist aus diesen Gründen unter der Mittelung bei Anwesenheit von Haftstellen immer nur die zweite Art der Mittelung zu verstehen.

Mit der gleichen Überlegung, mit der man Gl. (II.6) hergeleitet hat, bekommt man auch die Überlebenswahrscheinlichkeit Ψ_n nach n Schritten

$$\Psi_n = \sum_{\{T\}} p_{\{T\}} \sum_{\omega}' \frac{1}{(2d)^n} = \frac{1}{(2d)^n} \sum_{\omega} e^{-\lambda s_{\omega}}, \quad (II.7)$$

was man mit Hilfe der Verteilungsfunktion $p_n(s)$, die die Wahrscheinlichkeit angibt, daß ein Random Walk in n Schritten s verschiedene Plätze besucht, auch als

$$\Psi_n = \sum_{s=1}^{n+1} p_n(s) e^{-\lambda s}, \quad (II.8)$$

schreiben kann. Diese Beziehung ist in der Literatur von verschiedenen Autoren benutzt worden [15, 37]. Bei dem hier betrachteten Modell ist nicht ausgeschlossen, daß auch der Startplatz selbst schon eine Haftstelle ist, d. h., es gilt

$$\Psi_0 = 1 - c = p, \quad (II.9)$$

da der Startplatz mit der Wahrscheinlichkeit c eine Haftstelle ist. Andere Autoren normieren die Überlebenswahrscheinlichkeit so, daß $\Psi_0 = 1$ ist. Es wird sich aber zeigen, daß die hier gewählte Normierung zu schönen Skaleneigenschaften führt und deshalb vorzuziehen ist.

II.3. ZUSAMMENFASSUNG DER ASYMPTOTISCHEN THEORIE

In diesem Abschnitt sollen kurz die vorhandenen Ergebnisse der asymptotischen Theorie für das Haftstellen-Problem zusammengefaßt werden. Asymptotisch bedeutet hierbei, daß die Schrittzahl n hinreichend groß ist, wobei die notwendige Schrittzahl wiederum von der Haftstellenkonzentration c abhängen kann.

Bei großen Schrittzahlen n sind die meisten Teilchen eingefangen. Lediglich die Teilchen, die in einem relativ großen Haftstellen-freien Gebiet gestartet sind, haben die Chance zu überleben. Die Hauptbeiträge zur Überlebenswahrscheinlichkeit von n -Schritt Random Walks kommen aus Symmetriegründen von d -dimensionalen, Haftstellen-freien Kugeln, deren Radius R von verschiedenen Autoren [10, 13] berechnet worden ist*). Es gilt

$$R = R(\lambda, n) = \left(\frac{\alpha_d^2}{d^2 \Omega_d} \frac{n}{\lambda} \right)^{\frac{1}{d+2}}, \quad (II.10)$$

*) Dies ist im wesentlichen das Lifshitz-Argument [41] für die Zustandsdichte von Elektronen in einem Zufallspotential. Siehe auch Ref. 42.

wobei Ω_d das Volumen der d -dimensionalen Kugel mit Radius 1

$$\Omega_d := \begin{cases} 2, & \text{für } d = 1 \\ \pi, & \text{für } d = 2 \\ \frac{4}{3}\pi, & \text{für } d = 3 \end{cases} \quad (II.11)$$

und α_d das Maximum aller ersten Nullstellen der Lösungen der Differentialgleichung

$$f''(x) + \frac{d-1}{x}f'(x) + f(x) = 0$$

ist. Letzteres ist eine Konsequenz aus dem Auftauchen des größten Eigenwerts des Laplaceoperators auf einer Kugel mit absorbierenden Randbedingungen. Für die ersten drei Dimensionen gilt

$$\alpha_d := \begin{cases} \frac{\pi}{2} = 1.57080, & \text{für } d = 1 \\ j_{0,1} = 2.40483, & \text{für } d = 2 \\ j_{\frac{1}{2},1} = \pi = 3.14159, & \text{für } d = 3. \end{cases} \quad (II.12)$$

Hier sind $j_{0,1}$ und $j_{\frac{1}{2},1}$ die ersten Nullstellen der Besselfunktionen $\mathcal{J}_0$ und $\mathcal{J}_{\frac{1}{2}}$. Der Wert dieses Radius stellt einen Kompromiß dar, denn größere Kugeln sind zu selten in einer zufälligen Haftstellen-Konfiguration zu finden, während die Überlebenswahrscheinlichkeit in kleineren Kugeln zu gering ist, um nennenswert zu Ψ_n beizutragen [13].

Da der Radius der Haftstellen-freien Kugeln bekannt ist, ergibt sich für die Anzahl der Plätze in diesen Kugeln

$$s_{\max} = s_{\max}(\lambda, n) = \Omega_d R^d = \left(\frac{\Omega_d \alpha_d^d}{d^d} \right)^{\frac{2}{d+2}} \left(\frac{n}{\lambda} \right)^{\frac{d}{d+2}}. \quad (II.13)$$

Die Wahrscheinlichkeit, daß ein gegebener Gitterplatz das Zentrum einer solchen Kugel mit Radius R ist, ist

$$p(R) = (1 - c)^{\Omega_d R^d} = \exp(-\lambda \Omega_d R^d). \quad (II.14)$$

Der Logarithmus der Überlebenswahrscheinlichkeit geht dann asymptotisch wie

$$-\ln \Psi_n \sim \frac{d+2}{2} (\Omega_d \lambda)^{\frac{2}{d+2}} \left(\frac{\alpha_d^2}{d^2} n \right)^{\frac{d}{d+2}}. \quad (II.15)$$

Dieses Ergebnis ist von den Mathematikern Donsker und Varadhan zunächst für Brownsche Bewegungen [11] und später auch für Random Walks auf Gittern [12] exakt bewiesen worden. Man beachte jedoch, daß die Autoren, wenn sie von einer "sphere with unit volume" schreiben, offenbar eine d -dimensionale Kugel mit Radius 1 meinen und nicht

mit *Volumen* 1. Andernfalls würde ihr Resultat nicht mit den von Physikern mit verschiedenen Methoden gefundenen Ergebnissen übereinstimmen [10, 13, 14, 33, 34]. Auch die weiter unten durchgeführten Simulationen bestätigen dieses Bild.

Eine weitere, wichtige Größe ist S_n , die mittlere Anzahl der verschiedenen, besuchten Plätze nach n Schritten. Es gilt für große Schrittzahlen n [43]

$$S_n \sim \begin{cases} \left(\frac{8n}{\pi}\right)^{1/2}, & \text{für } d = 1 \\ \frac{\pi n}{\ln(8n)}, & \text{für } d = 2 \\ C_d n, & \text{für } d \geq 3. \end{cases} \quad (II.16)$$

Das Ergebnis für $d = 2$ ist asymptotisch gleichbedeutend mit $\pi n / \ln(n)$ und ist auch so in Ref. 43 angegeben, doch wird in Ref. 44 gezeigt, daß die zusätzliche "8" in Gl. (II.16) eine bessere Näherung für endliche n ist, siehe auch Ref. 15.

Da auf diese Zusammenhänge im folgenden immer wieder zurückgegriffen werden wird, werden sie hier noch einmal zusammengefaßt, wobei die numerischen Konstanten in der nachfolgenden Tabelle zu finden sind. Es gilt

$$R(\lambda, n) = A_d \left(\frac{n}{\lambda}\right)^{\frac{1}{d+2}} \quad (II.17)$$

$$-\ln \Psi_n = B_d \lambda^{\frac{2}{d+2}} n^{\frac{d}{d+2}} \quad (II.18)$$

$$S_n = C_d \begin{cases} \sqrt{n}, & \text{für } d = 1 \\ \frac{n}{\ln(8n)}, & \text{für } d = 2 \\ n, & \text{für } d \geq 3. \end{cases} \quad (II.19)$$

$$s_{\max}(\lambda, n) = D_d \left(\frac{n}{\lambda}\right)^{\frac{d}{d+2}} \quad (II.20)$$

mit

d	A_d	B_d	C_d	D_d
1	1,072515	3,21754	1,595769	2.14503
2	0,823644	4,26244	3,141593	2.13122
3	0,764881	4,68608	0,659463	1.87443

Tab. II.1: Numerische Konstanten für $d = 1, 2$ und 3

Das eindimensionale Problem ist in vielen Veröffentlichungen behandelt worden. Im Teil III. der Arbeit wird ausführlich darauf eingegangen. An dieser Stelle soll lediglich erwähnt werden, daß in einer Dimension eine exakte Lösung gelingt, da die Haftstellen das eindimensionale Gitter in endliche Stücke aufteilen, falls das Teilchen nur zu den nächsten Nachbarn springt. Es genügt daher, das Problem auf einem Teilstück der

Länge s mit zwei Haftstellen an den Enden zu lösen und danach über die Verteilung der s zu mitteln. Das asymptotische Verhalten für $d = 1$ ergibt sich dabei zu [32]

$$\Psi_n = 8\sqrt{\frac{2}{3}} \frac{c^2}{(1-c)\lambda^2} \left(\frac{x}{\pi}\right)^{3/2} \exp\left[-\frac{3}{2}x + \left(\frac{17}{18} - \frac{(\pi\lambda)^2}{12}\right)\frac{1}{x} + O\left(\frac{1}{x^2}\right)\right] \quad (II.21)$$

mit

$$x := (\pi\lambda)^{2/3} n^{1/3} \gg 1. \quad (II.22)$$

Das führende Verhalten im Exponenten entspricht dem Wert, der in Tab. II.1 für $d = 1$ angegeben ist. Wie man erkennt, gilt asymptotisch ein Skalenverhalten, da Ψ_n in führender Ordnung nur von x und nicht separat von c und n abhängt. Erst die Korrekturterme zeigen Abweichungen von der Skalenform. Da diese Korrekturterme bekannt sind, können detaillierte Aussagen über den Gültigkeitsbereich der asymptotischen Theorie in $d = 1$ gemacht werden.

Für $d > 1$ ist eine solche, exakte Behandlung nicht möglich. Lubensky [33] hat mit feldtheoretischen Methoden den ersten Korrekturterm in d Dimensionen berechnet (siehe auch Ref. 34). Sein Ergebnis ist, daß der Logarithmus der Überlebenswahrscheinlichkeit wie

$$-\ln \Psi_n \sim \dots n^{\frac{d}{d+2}} + \dots n^{\frac{d-1}{d+2}} + O\left(n^{\frac{d-2}{d+2}}\right)$$

geht (die λ -abhängigen Vorfaktoren wurden unterdrückt). Dies zeigt, daß die Korrekturterme bei endlichen Schrittzahlen n für $d > 1$ ein erhebliches Gewicht haben. Zudem geht aus der Rechnung der genaue Zahlenwert des Vorfaktors des Korrekturterms nicht hervor. Es bleibt daher schwierig, etwas über den Gültigkeitsbereich der asymptotischen Theorie zu sagen.

Um eine bessere Aussage über den Gültigkeitsbereich machen zu können, müssen genaue Monte Carlo Simulationen durchgeführt werden, die bis in das asymptotische Regime hineinreichen. Mit den üblichen Methoden ist das kaum möglich. Die im nächsten Abschnitt durchgeführte Abbildung des Haftstellen-Problems auf ein Polymermodell führt letztlich zu einer neuen Simulationsmethode, mit der die gewünschten Berechnungen möglich werden.

II.4. ÄQUIVALENZ ZWISCHEN DEM HAFTSTELLENPROBLEM UND SELF-INTERACTING WALKS (SIW's)

Um weitere Einsichten in das Haftstellen-Problem zu gewinnen, wird nun die Analogie zu einem Polymermodell dargestellt, bei dem der Walk nicht mehr als Hintereinanderausführung von einzelnen Schritten gesehen wird, sondern der Walk als Ganzes wie bei einem Polymer als Kette betrachtet wird.

II.4.1. Der Self-Interacting Walk (SIW)

Die Abbildung des Haftstellen-Problems auf ein Polymermodell liefert zwar keine exakte Lösung des Problems, trotzdem ist die neue Betrachtungsweise sehr fruchtbar. Insbesondere erlaubt sie eine wesentlich effizientere numerische Simulation, wie später gezeigt werden wird.

Sei also ω ein n -Schritt Random Walk. Jeder Walk ω soll ein statistisches Gewicht $e^{-\lambda s_\omega}$ bekommen, wobei s_ω die Anzahl der verschiedenen Plätze ist, die der Walk ω besucht. Hierbei ist λ zunächst ein beliebiger Parameter. Identifiziert man dieses Gewicht mit einem Boltzmannfaktor $e^{-\beta E}$, so erkennt man die Analogie zwischen der inversen Temperatur β und λ einerseits, und der Energie E und s_ω andererseits. Für $\lambda = 0$ erhält man den normalen Random Walk zurück. Für $\lambda > 0$ bekommen Walks, die viele verschiedene Plätze besuchen, ein geringeres Gewicht als solche, die nur an wenigen Plätzen waren. Das heißt also, daß kompaktere Walks bevorzugt werden. Man kann solche Walks als selbst-anziehende Walks (engl.: *self-attracting walks*) bezeichnen. Für $\lambda < 0$ gilt das oben Gesagte entsprechend. Solche Walks sind selbst-abstoßend (engl.: *self-repulsive*) und vermeiden es, Plätze öfter als einmal zu besuchen.

Das so charakterisierte Modell des SIW (Self-Interacting Walk) haben Stanley et. al. [37] eingeführt und mit Hilfe von exakten Abzählungen untersucht. Dabei sind sie jedoch nicht, wie von ihnen zuerst angenommen, in den asymptotischen Bereich vorgestoßen und deshalb ist auch ihre Schlußfolgerung eines "superuniversellen" Verhaltens mit dimensionsunabhängigen Exponenten an der Perkolationsschwelle falsch. Dies wird bei den weiter unten durchgeführten Simulationen klar werden.

Neben dem SIW gibt es weitere Modelle für wechselwirkende Walks, wie den SAW (engl.: *self avoiding walk*), der die abstoßende Wechselwirkung bei Polymeren gut beschreibt [45], den TSAW (engl.: *true self-avoiding walk*) [46], bei dem die Wahrscheinlichkeit, auf einen Platz zu springen, davon abhängt, wie oft dieser Platz vorher schon besucht wurde und das Domb-Joyce Modell [47], bei dem jede Selbstüberschneidung ein statistisches Gewicht bekommt. In Ref. 48 werden einige Aspekte dieser Walks miteinander verglichen und in Ref. 49 wird ein generalisierter Random Walk eingeführt, der zwischen dem normalen Random Walk, dem SIW, dem TSAW und dem SAW interpoliert, und es wurden exakte Abzählungen in einer Dimension durchgeführt, die mit Hilfe kombinatorischer Formeln bis auf Schrittzahlen von $n = 29$ ausgedehnt werden konnten [50].

Es gibt einen fundamentalen Unterschied zwischen den meisten der oben angegebenen Modellen und dem SIW bzw. dem Domb-Joyce Modell. Die Wechselwirkung

in diesen beiden Modellen ist nämlich nicht lokal, wie in den meisten Polymermodellen, wo einzelne Monomere, die sich nahe kommen, miteinander wechselwirken. Beim SIW (und auch beim Domb-Joyce Modell, das aber im folgenden nicht weiter untersucht wird) muß zunächst der ganze Walk "wachsen", bevor sein statistisches Gewicht feststeht. Die "Energie" s_ω ist über den ganzen Walk verteilt, ohne daß man sie auf die einzelnen Schritte aufteilen könnte. Sie ist ein Funktional des gesamten Walks ω .

Alle statistischen Größen werden mit diesem Gewicht gemittelt, d. h., wir haben die Zustandssumme

$$Z = \sum_{\omega} e^{-\lambda s_{\omega}}, \quad (II.23)$$

und eine beliebige, von einem Random Walk abhängige Größe $f(\omega)$ hat den Mittelwert

$$\langle f \rangle_{\text{SIW}} = \frac{1}{Z} \sum_{\omega} f(\omega) e^{-\lambda s_{\omega}}. \quad (II.24)$$

Das entspricht genau der Gl. (II.6) für die Mittelung über Walks, die an zufälligen Plätzen auf einem Gitter mit Haftstellen starten, falls für den Wechselwirkungsparameter $\lambda = -\ln(1 - c)$ gewählt wird. Damit ist gezeigt, daß folgendes gilt:

Jeder Mittelwert einer Self-Interacting Walk abhängigen Größe ist auch gleichzeitig der entsprechende Mittelwert für Random Walks bei Anwesenheit von Haftstellen, falls nur die überlebenden Walks bei der Mittelung berücksichtigt werden.

Insbesondere ist $\langle R^2 \rangle_{\text{SIW}}$ die mittlere quadratische Auslenkung eines Teilchens, das bei Anwesenheit von Haftstellen n Schritte überlebt hat. Darüberhinaus gilt für die Überlebenswahrscheinlichkeit selbst wegen Gl. (II.7)

$$\Psi_n = \frac{1}{(2d)^n} Z. \quad (II.25)$$

Im Gegensatz dazu ist von E. H. Stanley et. al. in Ref. 37 gesagt worden, daß die Äquivalenz zwischen dem Haftstellen-Problem und Self-Interacting Walks nur auf die Überlebenswahrscheinlichkeit und die Anzahl der besuchten Plätze beschränkt ist. Andere Größen, wie z. B. die mittlere quadratische Auslenkung, seien nicht übertragbar. Oben angestellte Überlegungen zu Gl. (II.6) zeigen jedoch, daß alle Größen ihre Entsprechung in dem jeweils anderen Modell haben, wenn man unter Mittelung die in jenem Abschnitt beschriebene Art versteht.

An dieser Stelle sollen nun noch die Ausdrücke für einige relevante Größen der statistischen Physik im Polymerbild angegeben werden. Die Energie E und die Varianz

der Energie $\Delta^2 E$ des SIW (die Boltzmannkonstante k_B ist der Einfachheit halber auf 1 gesetzt) sind

$$E = \langle s \rangle_{\text{SIW}} = \frac{1}{Z} \sum_{\omega} s_{\omega} e^{-\lambda s_{\omega}} = -\frac{\partial \ln Z}{\partial \lambda} \quad (\text{II.26})$$

$$\Delta^2 E = \Delta^2 s = \langle s^2 \rangle_{\text{SIW}} - (\langle s \rangle_{\text{SIW}})^2 = -\frac{\partial \langle s \rangle_{\text{SIW}}}{\partial \lambda} = \frac{\partial^2 \ln Z}{\partial \lambda^2}. \quad (\text{II.27})$$

Ferner ergeben sich die Entropie S , die freie Energie F und die Wärmekapazität C zu

$$S = -\left\langle \ln \frac{e^{-\lambda s}}{Z} \right\rangle_{\text{SIW}} = \lambda \langle s \rangle_{\text{SIW}} + \ln Z = \ln Z - \lambda \frac{\partial \ln Z}{\partial \lambda} \quad (\text{II.28})$$

$$F = -\frac{1}{\lambda} \ln Z \quad (\text{II.29})$$

$$C = \frac{\partial E}{\partial T} = -\lambda^2 \frac{\partial \langle s \rangle_{\text{SIW}}}{\partial \lambda} = \lambda^2 \Delta^2 s. \quad (\text{II.30})$$

II.4.2. Schrittweiser Aufbau von Self-Interacting Walks

Wenn man SIW's mit der richtigen Wahrscheinlichkeitsverteilung schrittweise aufbauen möchte, wird man wieder zum Haftstellen-Problem zurückgeführt. Das kann man folgendermaßen einsehen.

Das Gesamtgewicht eines erzeugten Walks, also das Produkt der Gewichte für alle Einzelschritte, soll gerade p^s sein. Deshalb muß jeder Schritt, der einen neuen Platz besucht, das Gewicht p bekommen. Jeder andere Schritt erhält das Gewicht 1. Angenommen, das Teilchen ist zu einem Platz gekommen, von dem aus es k Möglichkeiten hat, im nächsten Schritt einen neuen Platz zu besuchen. In den restlichen $2d - k$ Richtungen wird es dann die Anzahl der besuchten Plätze nicht erhöhen. Die Summe der statistischen Gewichte für den nächsten Schritt ist also $kp + (2d - k)1 = 2d - kc$ und hängt somit von k ab (außer für $c = 0$, also Random Walks auf Gittern ohne Haftstellen). Wenn man die Gesamtwahrscheinlichkeit für den nächsten Schritt auf 1 normiert, ändern sich die statistischen Gewichte von Schritt zu Schritt, da sie jeweils von der Umgebung des zuletzt besuchten Platzes abhängen.

Zum Beispiel sind in Bild II.3 zwei Walks mit normierten Wahrscheinlichkeiten abgebildet, die sich nur in ihrer Aufbaurichtung unterscheiden. Sie haben verschiedene statistische Gewichte, obwohl sie in der Anzahl der besuchten Plätze übereinstimmen. Ferner ist dort ein weiterer Walk mit anderer Gestalt aber gleichen Parametern skizziert, der ebenfalls ein anderes Gewicht hat.

Daraus folgt, daß eine Normierung nicht zu dem richtigen Ensemble führt. Normiert man aber die Übergangswahrscheinlichkeiten nicht auf 1, sondern sorgt nur dafür, daß die Wahrscheinlichkeiten immer ≤ 1 sind, bekommen obige Gewichte noch einen

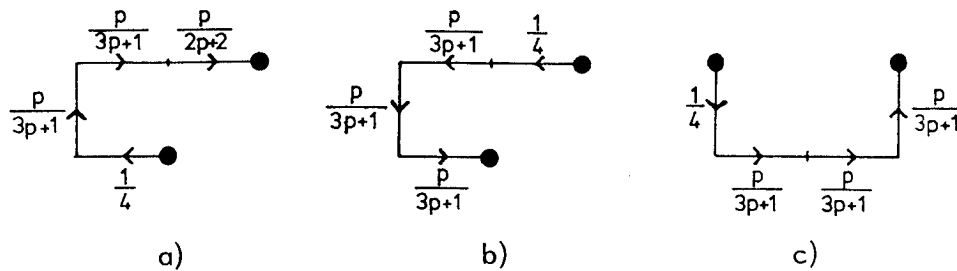


Bild II.3: Problem des schrittweisen Aufbaus von SIW's. Es sind drei verschiedene SIW's mit $n = 4$ und $s = 5$ dargestellt.

- a) statistisches Gewicht $\frac{1}{4} + \frac{2p}{3p+1} + \frac{p}{2p+2}$
- b) umgekehrt durchlaufener SIW: statistisches Gewicht $\frac{1}{4} + \frac{3p}{3p+1}$
- c) andere Gestalt: statistisches Gewicht $\frac{1}{4} + \frac{3p}{3p+1}$

zusätzlichen Faktor $1/2d$. Sind wieder k Richtungen vorhanden, einen neuen Platz zu besuchen, ist die Gesamtwahrscheinlichkeit für den nächsten Schritt $1 - \frac{k}{2d}c$ und damit für $c > 0$ kleiner als 1. Mit der Wahrscheinlichkeit $\frac{k}{2d}c$ darf also der nächste Schritt nicht ausgeführt werden, d. h., der Walk stirbt und es muß ein völlig neuer Walk begonnen werden.

Das ist aber genau die Interpretation des SIW's als Haftstellen-Problem, denn die k Nachbarplätze, die zum ersten Mal besucht werden würden, können mit der Wahrscheinlichkeit c Haftstellen sein. Die anderen Plätze sind schon besucht worden, und da der Walk noch nicht gestorben ist, müssen sie normale Gitterplätze sein.

Das Ergebnis dieser Überlegungen ist, daß man SIW's nicht schrittweise aufbauen kann, ohne dabei Verluste (engl.: *sample attrition*) zu erleiden. Dieser Punkt wird in der Diskussion der verschiedenen bekannten Computer-Simulationsverfahren weiter unten noch eine Rolle spielen.

III. Eindimensionaler korrelierter Random Walk mit Haftstellen

Bevor das Haftstellen-Problem in höheren Dimensionen mit Monte Carlo Methoden untersucht wird, soll der eindimensionale Fall noch etwas näher betrachtet werden, da hier eine asymptotisch exakte Lösung möglich ist.

Ein einführender Review-Artikel über Transporteigenschaften eindimensionaler, ungeordneter Systeme ist in Ref. 51 zu finden. Speziell Random Walks auf linearen Ketten mit Haftstellen wurden in vielen Veröffentlichungen behandelt [31, 37, 52–54]. Eine ausführliche Darstellung mit einer exakten Lösung findet sich in Ref. 32. Es wurde auch Drift [55] in das Modell mit einbezogen und exakt gelöst [56, 57]. Andere Autoren betrachteten das gleiche Problem in kontinuierlicher Zeit [58–62].

Eine weitere Verallgemeinerung besteht darin, auch Korrelationen in das Modell mit einzuschließen. Das ist in Ref. 63 getan worden, doch wurde dort nur die Rückkehrwahrscheinlichkeit an den Startplatz bei Anwesenheit von Haftstellen bestimmt, nicht die Überlebenswahrscheinlichkeit selbst. Während eine Drift die Teilchen in einer Dimension immer in die Haftstellen hinein treibt, und die Überlebenswahrscheinlichkeit deshalb asymptotisch immer rein exponentiell zerfällt, ist der Einfluß von Korrelationen nicht so offensichtlich. Betrachtet man Korrelationen mit einem Gedächtnis über einen Schritt, so daß z. B. Rückwärtssprünge mit höherer Wahrscheinlichkeit auftreten als Vorwärtssprünge, dann wird das Teilchen häufiger Plätze besuchen, auf denen es schon einmal gewesen ist und wo sich deshalb keine Haftstellen befinden. Die Überlebenswahrscheinlichkeit wird also ansteigen. Solche Korrelationen findet man bei der Teilchendiffusion in Kristallen, die durch Gitterfehlstellen vermittelt wird [64], denn der leere Platz, auf den das Teilchen gerade gesprungen ist, befindet sich kurz nach dem Sprung mit Sicherheit hinter dem Teilchen. Im umgekehrten Fall der verstärkten Vorwärtskorrelation, der die Trägheit der diffundierenden Teilchen ausdrückt, und der Anwendungen in der Konformation von Polymeren hat [65], würde man aus den gleichen Gründen eine verringerte Überlebenswahrscheinlichkeit erwarten.

Im folgenden soll nun die Überlebenswahrscheinlichkeit für korrelierte Random Walks auf linearen Ketten mit einer zufälligen Verteilung von Haftstellen bestimmt werden. Dazu wird zunächst die Verteilungsfunktion $p_n(s)$ für die Anzahl der verschiedenen besuchten Plätze berechnet. Dieser Teil der Arbeit ist recht technisch und liefert die Verteilungsfunktion in Form einer asymptotisch exakten Entwicklung. Einige Größen können aus ihr auch exakt abgeleitet werden, z. B. die mittlere Zahl der Schritte bis zum Einfang an einer Haftstelle. Zum Schluß wird eine asymptotische Entwicklung für die Überlebenswahrscheinlichkeit hergeleitet, die für beliebige Korrelationsfaktoren und

Haftstellen-Konzentrationen gültig ist.

Unter korrelierten Random Walks sollen im folgenden Walks verstanden werden, die ein ein-Schritt Gedächtnis haben. Die Wahrscheinlichkeit, daß der nächste Schritt in derselben Richtung erfolgt, wie der letzte, ist p_f , während $p_b = 1 - p_f$ die Wahrscheinlichkeit ist, daß der nächste Schritt ein Rücksprung ist. Definiert man den Korrelationsfaktor f als

$$f := p_f/p_b, \quad (III.1)$$

dann ergibt sich für die mittlere quadratische Auslenkung nach n Schritten zu [66]

$$\langle x^2 \rangle = fn + O(n^0); \quad n \gg 1 \quad (III.2)$$

in $d = 1$. Ein korrelierter Walk ist nun aber nicht nur ein einfacher Random Walk, bei dem die Schrittlängen und Schrittzahlen umskaliert sind. Das Verhalten bei mittleren Schrittzahlen kann auch nicht dadurch erhalten werden, daß man im Random Walk n durch fn ersetzt. Tatsächlich ist z. B. die erste Korrektur zu Gl. (III.2) $(1-f^2)/2$. Beim Random Walk mit $f = 1$ taucht diese gar nicht auf. Die vollständige Lösung für die mittlere quadratische Auslenkung wurde schon von Fuerth [66] angegeben. Eine andere Größe, die hier von besonderem Interesse ist, ist die mittlere Zahl von verschiedenen besuchten Plätzen. Sie ist asymptotisch gegeben durch [67]

$$\langle s_n \rangle = \sqrt{8fn/\pi} + 1 - f + O(n^{-1/2}). \quad (III.3)$$

Auch hier erhält man den führenden Beitrag durch Umskalierung aus dem entsprechenden Resultat für $f = 1$, die Korrekturen zum Skalenverhalten folgen diesem Schema jedoch nicht. Weitert man diese Argumente auf den Fall der Überlebenswahrscheinlichkeit aus, so erwartet man, daß das führende Verhalten durch Umskalierung $n \rightarrow fn$ aus den bereits bekannten Resultaten folgt. Die eng mit der Überlebenswahrscheinlichkeit verknüpfte Rückkehrwahrscheinlichkeit bei Anwesenheit von Haftstellen wurde von Prasad und Unnikrishnan [63] in $d = 1$ für den Fall der korrelierten Walks berechnet. In der Tat zeigt das Resultat auch hier das erwartete Skalenverhalten. Für die Korrekturen kann man das aber nicht erwarten. Da die Korrekturterme auch schon zur Beschreibung der Überlebenswahrscheinlichkeit von einfachen Random Walks wichtig sind [32], sollen sie im folgenden berechnet werden. Wegen

$$\Psi_n = \langle (1-c)^{s_n} \rangle = \sum_s p_n(s) e^{-\lambda s}, \quad (III.4)$$

siehe Gl. (II.8), ist der erste Schritt die Berechnung von $p_n(s)$, der Wahrscheinlichkeit, daß ein Random Walk nach n Schritten s verschiedene Plätze besucht.

Dazu wird zunächst die Mastergleichung für korrelierte Walks auf einer endlichen Kette mit absorbierenden Randbedingungen gelöst. Der Zusammenhang mit der Verteilungsfunktion $p_n(s)$ wird ausgearbeitet und anschließend werden daraus einige exakte Resultate abgeleitet, wie die mittlere Zahl der Schritte bis zum Einfang auf einer unendlichen Kette mit Haftstellen sowie höhere Momente hiervon. Ferner kann die mittlere Zahl der Schritte, während der die Anzahl der besuchten Plätze konstant bleibt, exakt berechnet werden. Im nächsten Abschnitt wird dann die Verteilungsfunktion $p_n(s)$ asymptotisch exakt bestimmt. Die Überlebenswahrscheinlichkeit auf einer Kette mit zufällig verteilten Haftstellen wird schließlich ebenfalls asymptotisch exakt hergeleitet.

III.1. KORRELIERTE WALKS AUF ENDLICHEN KETTEN MIT ABSORBIERENDEN RANDBEDINGUNGEN

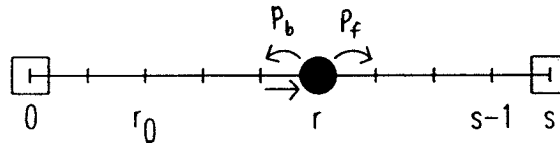


Bild III.1: Teilchen auf endlicher Kette mit Haftstellen.

Das folgende Modell wird betrachtet, siehe Bild III.1. Ein Teilchen wird zufällig auf eine endliche lineare Kette mit $s - 1$ Plätzen aufgebracht, an deren Enden sich zwei Haftstellen an den Positionen 0 und s befinden. Die Haftstellen sind perfekt, d. h., sie absorbieren das Teilchen sofort, wenn es auf die Haftstelle springt. Das Teilchen vollführt einen diskreten, korrelierten Random Walk, bei dem die Wahrscheinlichkeit für die Richtung des nächsten Schrittes von der Richtung des vorherigen Schrittes abhängt. Die Wahrscheinlichkeit, daß der nächste Schritt in dieselbe Richtung geht wie der vorherige, ist p_f , die Wahrscheinlichkeit für einen Schritt in die entgegengesetzte Richtung ist $p_b = 1 - p_f$. Der Korrelationsfaktor f ist definiert als $f := p_f/p_b$. Das Teilchen starte bei r_0 , $0 < r_0 < s$, mit gleicher Wahrscheinlichkeit für die beiden Richtungen des ersten Sprunges. Nach n Schritten ist das Teilchen am Platz r mit der Wahrscheinlichkeit $P_n^+(r)$, falls es von rechts auf diesen Platz kam, und mit der Wahrscheinlichkeit $P_n^-(r)$, falls es von links kam. Falls es im Verlauf des Walks auf eine der Haftstellen bei 0 und s kommt, wird es dort sofort absorbiert. Es gelten die folgenden Beziehungen, zunächst die Anfangsbedingung

$$P_0^\pm(r) = \frac{1}{2} \delta_{r,r_0}; \quad \text{für } 0 < r, r_0 < s. \quad (\text{III.5a})$$

Für den diskreten Walk gilt die Mastergleichung, siehe auch Ref. 68, 69,

$$\begin{aligned} P_{n+1}^-(r) &= p_f P_n^-(r-1) + p_b P_n^+(r-1) \\ P_{n+1}^+(r) &= p_b P_n^-(r+1) + p_f P_n^+(r+1) \end{aligned}; \quad \text{für } 0 < r < s \quad \text{und } n \geq 0. \quad (III.5b)$$

Schließlich sind die Randbedingungen

$$P_n^\pm(0) = P_n^\pm(s) = 0, \quad \text{für } n \geq 0 \quad (III.5c)$$

zu beachten. Für die Rechnung ist es nützlich zu bemerken, daß aus obigen Gleichungen unmittelbar

$$\begin{aligned} P_n^-(1) &= \frac{1}{2} \delta_{n,0} \delta_{r_0,1} \\ P_n^-(s-1) &= \frac{1}{2} \delta_{n,0} \delta_{r_0,s-1} \end{aligned} \quad (III.6)$$

folgt. Als Standardmethode, diese Gleichungen zu lösen, werden Erzeugende Funktionen [43, 70] eingeführt

$$P^\pm(r, z) := \sum_{n=0}^{\infty} P_n^\pm(r) z^n \quad (III.7)$$

wobei z zunächst als reell angenommen wird. Ferner wird von der endlichen, diskreten Fourier-Transformation [43, 70]

$$P^\pm(k_m, z) := \sum_{r=0}^{s-1} P^\pm(r, z) e^{-ik_m r} \quad (III.8)$$

mit $k_m := 2\pi m/s$ und ganzzahligem m mit $0 \leq m < s$ Gebrauch gemacht. Die Umkehrtransformation ist dann durch

$$P^\pm(r, z) = \frac{1}{s} \sum_{m=0}^{s-1} P^\pm(k_m, z) e^{ik_m r} \quad (III.9)$$

gegeben. Aus Gl. (III.5) und Gl. (III.6) folgt damit ein lineares Gleichungssystem für $P^\pm(k_m, z)$, dessen Lösung

$$\begin{aligned} P^-(k_m, z) &= a_{11}(k_m, z) P^-(r=s-1, z) + a_{12}(k_m, z) P^+(r=1, z) + c_-(k_m, z, r_0) \\ P^+(k_m, z) &= a_{21}(k_m, z) P^-(r=s-1, z) + a_{22}(k_m, z) P^+(r=1, z) + c_+(k_m, z, r_0) \end{aligned} \quad (III.10a)$$

mit

$$\begin{aligned} a_{11}(k_m, z) &:= -\frac{p_f z(1 - p_f z e^{ik_m})}{D(k_m, z)} \\ a_{12}(k_m, z) &:= -\frac{p_f p_b z^2 e^{-ik_m}}{D(k_m, z)} \\ a_{21}(k_m, z) &:= [a_{12}(k_m, z)]^* \\ a_{22}(k_m, z) &:= [a_{11}(k_m, z)]^* \end{aligned} \quad (III.10b)$$

$$c_{\pm}(k_m, z, r_0) := \frac{(1 - p_f z e^{\mp i k_m}) (e^{-i k_m r_0} - p_b z \delta_{r_0, \pm 1}) + p_b z e^{\pm i k_m} (e^{-i k_m r_0} - p_b z \delta_{r_0, \mp 1})}{2D(k_m, z)}$$

ist. In obigen Gleichungen ist $\delta_{r_0, -1}$ jeweils als $\delta_{r_0, s-1}$ zu verstehen. Ferner wurde die Determinante

$$D(k_m, z) := 1 - 2p_f z \cos k_m + (p_f - p_b)z^2 \quad (III.10c)$$

benutzt. In obiger Lösung hängt lediglich $c_{\pm}$ vom Startplatz r_0 ab. Rücktransformation und Auswertung der Gleichungen an den Plätzen $r = 1$ and $r = s - 1$ führen zu linearen Gleichungen für $P^+(r = 1, z)$ and $P^-(r = s - 1, z)$, die man als

$$\begin{aligned} C_- &= dP^-(r = s - 1, z) + eP^+(r = 1, z) \\ C_+ &= eP^-(r = s - 1, z) + dP^+(r = 1, z) \end{aligned} \quad (III.11a)$$

mit

$$\begin{aligned} d &= d^* = d(z) := 1 - \frac{1}{s} \sum_{m=0}^{s-1} a_{11}(k_m, z) e^{-i k_m} \\ e &= e^* = e(z) := -\frac{1}{s} \sum_{m=0}^{s-1} a_{21}(k_m, z) e^{i k_m} \\ C_{\pm} &= C_{\pm}(z, r_0) := \frac{1}{s} \sum_{m=0}^{s-1} c_{\pm}(k_m, z, r_0) e^{\pm i k_m} \end{aligned} \quad (III.11b)$$

schreiben kann. Diese Funktionen werden in Anhang A explizit berechnet. Setzt man Gl. (III.11) in Gl. (III.10) ein, erhält man das Resultat

$$\begin{aligned} P^-(k_m, z) &= \frac{a_{11}(C_- d - C_+ e) + a_{21}^*(C_+ d - C_- e)}{d^2 - e^2} + c_- \\ P^+(k_m, z) &= \frac{a_{21}(C_- d - C_+ e) + a_{11}^*(C_+ d - C_- e)}{d^2 - e^2} + c_+. \end{aligned} \quad (III.12)$$

Das Resultat Gl. (III.12) gibt —im Fourierraum und als Erzeugende Funktion— die Wahrscheinlichkeit an, das Teilchen am Platz r auf einer endlichen, linearen Kette mit absorbierenden Randbedingungen zu finden, wenn es einen korrelierten Random Walk ausführt. Summiert man diese Wahrscheinlichkeit über alle erreichbaren Plätze, erhält man die Überlebenswahrscheinlichkeit auf der Kette bei spezifizierter Startposition.

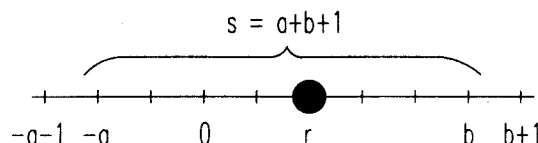


Bild III.2: Teilchen auf unendlicher Kette.

III.2. DARSTELLUNG DER VERTEILUNGSFUNKTION $p_n(s)$

In diesem Abschnitt wird eine Darstellung für die Verteilungsfunktion der Zahl der verschiedenen besuchten Plätze abgeleitet, die anschließend einige exakte Resultate ermöglicht.

Da in einer Dimension die Anzahl s der besuchten Plätze um 1 größer ist, als der Abstand zwischen dem äußerst rechten und dem äußerst linken Platz, die im Verlauf des Walks besucht wurden, kann man die Verteilungsfunktion wie folgt berechnen.

Dazu wird der Bequemlichkeit halber eine leicht veränderte Notation verwendet, siehe Bild III.2. Das Teilchen startet jetzt am Ursprung einer unendlichen, linearen Kette ohne Haftstellen. Für zwei beliebige nicht-negative, ganze Zahlen a und b wird definiert:

$f_n^{a,b}$ ist die Wahrscheinlichkeit, daß das Teilchen in den ersten n Schritten niemals einen der Plätze $-a$ oder b besucht hat.

$v_n^{a,b}$ ist die Wahrscheinlichkeit, daß das Teilchen in den ersten n Schritten niemals einen der Plätze $-(a+1)$ oder $b+1$ besucht hat, aber mit Sicherheit an den Plätzen $-a$ und b gewesen ist.

Offensichtlich ist $f_n^{a,b}$ die Überlebenswahrscheinlichkeit eines Teilchens, das auf einer linearen Kette am Ursprung zwischen zwei Haftstellen an den Plätzen $-a$ und b gestartet ist. Sie kann mit dem Resultat aus dem letzten Abschnitt berechnet werden. Auf der anderen Seite ist $v_n^{a,b}$ mit der Anzahl der besuchten Plätze verknüpft, denn ein Walk der bis zu den Plätzen $-a$ und b gelangt ist, hat genau $a+b+1$ verschiedene Plätze besucht. Es ist klar, daß folgende Identitäten gelten:

$$f_n^{a,b} = 0; \quad \text{für } a = 0 \quad \text{oder} \quad b = 0, \quad (III.13)$$

$$f_n^{a,b} = \sum_{a'=0}^{a-1} \sum_{b'=0}^{b-1} v_n^{a',b'}. \quad (III.14)$$

Damit ist durch Einsetzen einfach zu zeigen, daß

$$v_n^{a,b} = f_n^{a+1,b+1} - f_n^{a,b+1} - f_n^{a+1,b} + f_n^{a,b} \quad (III.15)$$

erfüllt ist. Die Größe $f_n(s)$ wird nun als die mittlere Zahl der Teilchen definiert, die zwischen zwei Haftstellen im Abstand s starten, wenn an jedem Platz ein Teilchen startet (zusammen also $s - 1$ Teilchen):

$$f_n(s) := \sum_{a=1}^{s-1} f_n^{a, s-a}. \quad (III.16)$$

Die Wahrscheinlichkeit $p_n(s)$, daß ein Teilchen s verschiedene Plätze besucht, ist auf der anderen Seite

$$p_n(s) = \sum_{a=0}^{s-1} v_n^{a, s-a-1}, \quad (III.17)$$

wobei die Summe über alle Möglichkeiten läuft, bei denen ein Teilchen am Ursprung startet und s verschiedene Plätze besucht. Einsetzen von Gl. (III.15) in Gl. (III.17) unter Benutzung von Gl. (III.14) und Gl. (III.16) liefert schließlich das Resultat

$$p_n(s) = f_n(s+1) - 2f_n(s) + f_n(s-1). \quad (III.18)$$

Diese Beziehung ist das zentrale Resultat dieses Abschnitts. Es verknüpft die Wahrscheinlichkeit, s verschiedene Plätze auf einer unendlichen, linearen Kette ohne Haftstellen zu besuchen, mit der Überlebenswahrscheinlichkeit auf endlichen Ketten zwischen Haftstellen. Letztere kann mit Hilfe der entsprechenden Mastergleichungen berechnet werden. Für den Fall der korrelierten Random Walks wird dies im übernächsten Abschnitt durchgeführt werden.

III.3. EINIGE EXAKTE ERGEBNISSE

Die Darstellung Gl. (III.18) der Verteilungsfunktion für die Anzahl der besuchten Plätze wird nun benutzt, um exakte Resultate für die mittlere Zahl der Schritte bis zum Einfang und höhere Momente, sowie für die mittlere Zahl der Schritte, während der die Anzahl der besuchten Plätze konstant bleibt, für den korrelierten Random Walk herzuleiten. Dazu wird zunächst die Größe $f_n(s)$, die in Gl. (III.16) definiert wurde, mit Hilfe der Resultate aus Abschnitt III.1 berechnet. Es gilt

$$f_n(s) = \sum_{r_0=1}^{s-1} \sum_{r=0}^{s-1} [P_n^+(r) + P_n^-(r)] \quad (III.19)$$

oder mit Hilfe von Erzeugenden Funktionen

$$f(s, z) = \sum_{r_0=1}^{s-1} \sum_{r=0}^{s-1} [P^+(r, z) + P^-(r, z)] = \sum_{r_0=1}^{s-1} [P^+(k_m = 0, z) + P^-(k_m = 0, z)]. \quad (III.20)$$

Berücksichtigt man ferner, daß a_{11} und a_{21} an der Stelle $k_m = 0$ reell sind, bekommt man sofort

$$f(s, z) = \frac{a_{11}(k_m = 0, z) + a_{21}(k_m = 0, z)}{d(z) + e(z)} \sum_{r_0=1}^{s-1} [C_+(z, r_0) + C_-(z, r_0)] \\ + \sum_{r_0=1}^{s-1} [c_+(k_m = 0, z, r_0) + c_-(k_m = 0, z, r_0)]. \quad (III.21)$$

Die Summen können exakt ausgewertet werden (siehe Anhang A) und nach etwas Algebra erhält man schließlich das Resultat

$$f(s, z) = \frac{1}{1-z} \left(s - \frac{fz(e_1 - e_2)}{(1-z)h(z)} \right) \quad (III.22a)$$

mit

$$e_{1,2} := a \pm \sqrt{a^2 - 1}; \quad a := \frac{f+1+(f-1)z^2}{2fz} \quad (III.22b)$$

und

$$h(z) := \frac{f+1-(f-1)ze_2}{1-e_2^s} - \frac{f+1-(f-1)ze_1}{1-e_1^s}. \quad (III.22c)$$

Da $\Psi_{n-1} - \Psi_n$ die Wahrscheinlichkeit ist, daß das Teilchen genau beim Schritt n eingefangen wird, ist das k -te Moment der Verteilungsfunktion für die Zahl der Schritte bis zum Einfang

$$\langle n^k \rangle = \sum_{n=1}^{\infty} n^k (\Psi_{n-1} - \Psi_n) = \sum_{l=0}^{k-1} \binom{k}{l} \sum_{n=0}^{\infty} n^l \Psi_n. \quad (III.23)$$

Mit Gl. (III.4) findet man

$$\sum_{n=0}^{\infty} n^l \Psi_n = \sum_{s=1}^{\infty} (1-c)^s \sum_{n=0}^{\infty} n^l p_n(s) = \sum_{s=1}^{\infty} (1-c)^s \left(z \frac{d}{dz} \right)^l p(s, z) \Big|_{z=1}. \quad (III.24)$$

Mit Hilfe von Gl. (III.18), geschrieben als Erzeugende Funktionen $p(s, z) = f(s+1, z) - 2f(s, z) + f(s-1, z)$, kann man die Momente nun leicht exakt berechnen. Die ersten zwei Momente ergeben sich zu

$$\langle n \rangle = \frac{1-c}{c} \left(1 + \frac{1}{f} \frac{1-c}{c} \right) \quad (III.25)$$

$$\langle n^2 \rangle = \frac{1-c}{c} \left\{ \frac{2-c}{c} + \frac{1}{f} \frac{1-c}{c} \left[\frac{8-3c}{c} + \frac{1}{f} \left(\frac{2-c}{c} \right)^2 \right] \right\}. \quad (III.26)$$

Diese Ergebnisse sind Verallgemeinerungen der klassischen Resultate von Montroll [71] für einfache Random Walks auf korrelierte (bis auf einen trivialen Faktor $1-c$, der daher

kommt, daß in dieser Arbeit auch der Startplatz eine Haftstelle sein darf.) Seine Ergebnisse werden wiedergefunden, wenn man in obigen Gleichungen $f = 1$ setzt. (Siehe dazu auch [72–74], wo sich weitere Ergebnisse für $\langle n \rangle$ auch anderer Walktypen und für höhere Dimensionen finden.) Ferner resultieren aus obigen Betrachtungen zwei interessante Summenregeln

$$\sum_{n=0}^{\infty} f_n(s) = f(s, z = 1) = \frac{s(s-1)(s+3f-2)}{6f} \quad (III.27)$$

und

$$\sum_{n=0}^{\infty} p_n(s) = p(s, z = 1) = \frac{s+f-1}{f}. \quad (III.28)$$

Sie reduzieren sich für den unkorrelierten Fall zu den bemerkenswert einfachen Ausdrücken

$$\sum_{n=0}^{\infty} f_n(s) = \frac{s(s-1)(s+1)}{6} \quad (III.29)$$

und

$$\sum_{n=0}^{\infty} p_n(s) = s. \quad (III.30)$$

Gl. (III.28) hat eine einfache physikalische Bedeutung. Dazu definiere man eine Zufallsgröße $\eta_n(s)$, die einem Random Walk zugeordnet ist, und die 1 ist, wenn das Teilchen nach n Schritten genau s verschiedene Plätze besucht hat und 0 sonst. Der Mittelwert dieser Größe, gemittelt über alle Walks mit $N \geq n$ Schritten, ist wiederum die Verteilungsfunktion $p_n(s)$. Betrachtet man nun ein bestimmtes s , ergibt die Größe $\nu_s^N := \sum_{n=0}^N \eta_n(s)$ die Gesamtzahl der Schritte des N -Schritt Random Walks, während der die Anzahl der besuchten Plätze genau s war. Mittelt man diese Zufallsgröße über alle N -Schritt Walks, erhält man

$$\langle \nu_s^N \rangle = \sum_{n=0}^N \langle \eta_n(s) \rangle = \sum_{n=0}^N p_n(s). \quad (III.31)$$

Im Grenzfall $N \rightarrow \infty$ erhält man die mittlere Zahl von Schritten $\langle \nu_s \rangle$, während der die Anzahl der besuchten Plätze konstant auf s geblieben ist,

$$\langle \nu_s \rangle = \sum_{n=0}^{\infty} p_n(s). \quad (III.32)$$

Dieses Ergebnis gilt allgemein für beliebige Walks in beliebiger Dimension auf beliebigen Gittern. Für den eindimensionalen korrelierten Random Walk erhält man nach (III.28)

$$\langle \nu_s \rangle = \frac{s+f-1}{f}, \quad (III.33)$$

bzw. für den unkorrelierten Walk

$$\langle \nu_s \rangle = s. \quad (III.34)$$

Je größer also die Zahl der besuchten Plätze ist, um so länger dauert es, bis der nächste neue Platz besucht wird. Bemerkenswert ist es, daß die Zahl der benötigten Schritte in einer Dimension im Mittel linear mit s anwächst.

III.4. ASYMPTOTISCHE VERTEILUNGSFUNKTION $p_n(s)$

In diesem Abschnitt wird eine asymptotische Formel für die Wahrscheinlichkeit, daß ein n -Schritt Random Walk gerade s verschiedene Plätze besucht, für $n \gg 1$ hergeleitet. Im wesentlichen wird die Inversionsformel für Erzeugende Funktionen benutzt und nach Potenzen von n entwickelt. Dabei muß sorgfältig auf die n -Abhängigkeit der einzelnen Terme geachtet werden, um eine systematische, asymptotische Entwicklung zu erhalten. Die Ableitung selbst ist recht technisch, weshalb der Leser, der nur an den Resultaten interessiert ist, gleich zur Diskussion am Ende dieses Kapitels übergehen kann.

Die Umkehrung der Erzeugenden Funktion $f(s, z)$ wird durch

$$f_n(s) = \frac{1}{2\pi i} \oint \frac{f(s, z)}{z^{n+1}} dz \quad (III.35)$$

erreicht [75], wobei nun z als komplexe Variable aufgefaßt wird und der Integrationsweg ein Kreis um den Ursprung mit Radius $|z| < 1$ ist. Das Integral wird mit Hilfe des Residuen-Satzes ausgewertet. Einfache, aber sehr mühsame Algebra zeigt, daß das Residuum bei $z = 0$ Null ist, und daß alle einfachen Pole z_k , die sich in der Menge der Nullstellen von $h(z)$ in Gl. (III.22c) befinden, die Gleichung

$$e_1^s = -f + (f - 1)ze_1 \Big|_{z=z_k} \quad (III.36)$$

erfüllen, wobei der Index k zwischen verschiedenen Polen unterscheidet. Diese Gleichung kann nicht in geschlossener Form für beliebiges f nach z_k aufgelöst werden. Das ist nur im unkorrelierten Fall möglich. Trotzdem kann man eine systematische, asymptotische Entwicklung für das Integral durchführen. Einfache Algebra zeigt, daß das Residuum von $f(s, z)$ bei $z = z_k$ als

$$\text{Res}[f(s, z), z_k] = \frac{1}{z_k^n} \frac{z_k + 1}{z_k - 1} \frac{(f + 1)^2 - (f - 1)^2 z_k^2}{s(f + 1) - (s - 2)(f - 1)z_k^2} \quad (III.37)$$

geschrieben werden kann. Damit gilt

$$f_n(s) = - \sum_k \text{Res}[f(s, z), z_k]. \quad (III.38)$$

Zur Orientierung wird nun zunächst der unkorrelierte Fall betrachtet. Für $f = 1$ gilt $e_1^s(z_k) = -1$, was

$$e_1(z_k) = e^{i\phi_k}, \quad (III.39a)$$

mit

$$\phi_k := \frac{(2k+1)\pi}{s}; \quad k = 0, 1, 2, \dots, s-1. \quad (III.39b)$$

bedeutet. Mit der Definition von e_1 , Gl. (III.22b), für $f = 1$ und Gl. (III.39) erhält man unmittelbar $z_k = 1/\cos \phi_k$. In Kombination mit Gl. (III.37), Gl. (III.38) und Gl. (III.39b) erhält man das exakte Resultat

$$f_n(s) = \frac{2}{s} \sum_{\substack{m=1 \\ m \text{ ungerade}}}^{s-1} \cot^2\left(\frac{m\pi}{2s}\right) \cos^n\left(\frac{m\pi}{s}\right). \quad (III.40)$$

Ein exakter Ausdruck für die Verteilungsfunktion $p_n(s)$ folgt unmittelbar mit Gl. (III.18). Dies ist ein bekanntes, auf anderen Wegen erzielt Resultat [76–81]. Asymptotisch, für $n \rightarrow \infty$, tragen im unkorrelierten Fall nur die Summanden bei, für die $|\cos(m\pi/s)|$ am größten ist. Das sind die Terme mit $m = 1$ und, falls s gerade ist, $m = s-1$. Dies ist ausführlich in Ref. 32 untersucht worden.

Analog tragen im allgemeinen Fall (f beliebig), der nun untersucht werden soll, nur die Pole asymptotisch bei, für die $|z_k|$, wegen des $1/z_k^n$ Terms in Gl. (III.37), am kleinsten sind. Im Anhang B ist gezeigt, daß asymptotisch für $n, s \rightarrow \infty$ nur Lösungen von Gl. (III.36) mit der Form

$$e_{1,2} = e^{\pm i\phi}, \quad \phi \text{ reell}, \quad (III.41)$$

beitragen können. Dann folgt aus Gleichung (B1), daß alle relevanten Pole die Form

$$z_k^\pm := \frac{1}{f-1} \left(f \cos \phi \pm \sqrt{1 - f^2 \sin^2 \phi} \right) \quad (III.42)$$

haben. Falls $|\sin \phi| > 1/f$ ist (woraus $f > 1$ folgt), sind die $z_k^\pm$ komplex mit $|z_k^\pm|^2 = (f+1)/(f-1) > 1$. Diese Pole tragen asymptotisch nicht bei, da es weitere Lösungen mit kleineren Absolutbeträgen gibt. Für $|\sin \phi| \leq 1/f$ ist z reell, und es ist einfach zu zeigen, daß nur $z_k^-(\phi \rightarrow 0) \rightarrow 1$ und $z_k^+(\phi \rightarrow \pi) \rightarrow -1$ relevant in dem Sinne sind, daß ihre Absolutbeträge am kleinsten sind. Deshalb ist es sinnvoll, $z_k^\pm$ aus Gl. (III.42) in Potenzen von ϕ , bzw. $\phi' := \phi - \pi$ zu entwickeln. Es gilt

$$\begin{aligned} z_k^-(\phi) &= 1 + \frac{f}{2}\phi^2 + O(\phi^4) \\ z_k^+(\pi + \phi') &= -z_k^-(\phi'). \end{aligned} \quad (III.43)$$

Da die $z_k^\pm$ gerade Funktionen von ϕ bzw. ϕ' sind, genügt es, positive ϕ und ϕ' zu betrachten. Wendet man auf beiden Seiten von Gl. (III.36) den Logarithmus an und entwickelt die rechte Seite der Gleichung wiederum in Potenzen von ϕ , erhält man

$$(2k+1)\pi = (s+f-1)\pi\phi + \frac{f(f^2-1)}{6}\phi^3 + O(\phi^5). \quad (III.44)$$

Hier ist k eine beliebige ganze Zahl, bei der man wegen $\phi \geq 0$ auch $k \geq 0$ annehmen kann. Da die asymptotische Form von $p_n(s)$ gesucht werden soll, sind die relevanten s zusammen mit n ebenfalls groß. Gl. (III.44) legt den Ansatz für ϕ

$$\phi = \frac{\kappa_-}{s'} \left(1 + \frac{\phi_1}{s'} + \frac{\phi_2}{s'^2} + \frac{\phi_3}{s'^3} + \frac{\phi_4}{s'^4} + O(s'^{-5}) \right) \quad (III.45)$$

nahe, wobei

$$s' := s + f - 1 \quad \text{und} \quad \kappa_- := (2k+1)\pi \quad (III.46)$$

eingeführt wurden. Einsetzen in Gl. (III.44) liefert

$$\begin{aligned} \phi_1 &= \phi_2 = \phi_4 = 0 \\ \phi_3 &= -\frac{f(f^2-1)\kappa_-^2}{6}. \end{aligned} \quad (III.47)$$

Mit der gleichen Argumentation findet man analoge Formeln für den Fall $\phi' \rightarrow 0$, man muß nur ϕ durch ϕ' und κ_- durch

$$\kappa_+ := \left(2k + \frac{1+(-1)^s}{2} \right) \pi \quad (III.48)$$

ersetzen. Hierbei kann wieder $k \geq 0$ angenommen werden. Nach Einsetzen von ϕ und ϕ' in Gl. (III.43) erhält man die asymptotische Entwicklung

$$\begin{aligned} z_k^\pm &= \mp \left[1 + \frac{f\kappa_\pm^2}{2} \frac{1}{s'^2} + \frac{f(3f^2+3f-1)\kappa_\pm^4}{24} \frac{1}{s'^4} - \frac{f^2(f^2-1)\kappa_\pm^4}{6} \frac{1}{s'^5} \right. \\ &\quad \left. + \frac{f(45f^4+45f^3-15f^2-15f+1)\kappa_\pm^6}{720} \frac{1}{s'^6} + O(s'^{-7}) \right]. \end{aligned} \quad (III.49)$$

Nun können auch die Residuen von $f_n(s)$ nach Gl. (III.37) und Gl. (III.38) durch Einsetzen berechnet werden. Lediglich der Term

$$\begin{aligned} \frac{1}{(z_k^\pm)^n} &= (\mp 1)^n \exp \left[-n \ln(\mp z_k^\pm) \right] \\ &= (\mp 1)^n \exp \left[-\frac{\kappa_\pm^2}{2} \frac{fn}{s'^2} - \frac{(3f^2-1)\kappa_\pm^4}{24} \frac{fn}{s'^4} + \dots + O(s'^{-7}) \right] \end{aligned} \quad (III.50)$$

benötigt etwas Aufmerksamkeit, denn dort gibt es einen Wettbewerb zwischen zwei Größen, n und s' , die beide groß gegen 1 sind. Im folgenden wird angenommen, daß

$s' = s + f - 1$ von der Ordnung $(fn)^{1/2}$ ist. Das bedeutet, daß nur s -Werte betrachtet werden, die in der Nähe ihres Mittelwertes liegen, siehe Gl. (III.3). Bei der Berechnung der Überlebenswahrscheinlichkeit im nächsten Kapitel wird diese Annahme durch eine andere ersetzt werden. Dort werden auch alle Terme der Entwicklung Gl. (III.49) benötigt. Für die Zwecke in diesem Abschnitt werden nun Terme höherer Ordnung vernachlässigt.

Es ist bequem,

$$\tau := \frac{\sqrt{fn}}{s + f - 1} = O(1) \quad (III.51)$$

und $\epsilon := \frac{1}{\sqrt{fn}} \ll 1.$

einzuführen. Dann können alle Größen systematisch als Potenzreihe in ϵ entwickelt werden, wobei τ ein Parameter der Ordnung $O(1)$ ist. Um nun eine asymptotische Entwicklung für $p_n(s)$ zu bekommen, wird Gl. (III.18) benutzt, wobei $f_n(s-1)$ und $f_n(s+1)$ als Taylorreihen um s entwickelt werden. Dabei sind gerade und ungerade s wegen der s -Abhängigkeit in Gl. (III.48) separat zu behandeln. Als endgültiges Resultat für die asymptotische Verteilungsfunktion der Zahl der verschiedenen besuchten Plätze erhält man schließlich

$$p_n(s) = \sum_{\substack{k=1 \\ \kappa := k\pi}}^{\infty} 8\tau^3 \epsilon \left[b_0 + b_2 \epsilon^2 + O(\epsilon^3) \right] \exp\left(-\frac{\kappa^2 \tau^2}{2}\right), \quad (III.52a)$$

wobei für gerade k

$$\begin{aligned} b_0 &:= 0 \\ b_2 &:= (-1)^{n+s} \frac{f^2 \kappa^2}{8} \end{aligned} \quad (III.52b)$$

und für ungerade k

$$\begin{aligned} b_0 &:= \kappa^2 \tau^2 - 1 \\ b_2 &:= \frac{\kappa^6}{12} \tau^8 - \frac{\kappa^4}{24} [(3f^2 - 1)\kappa^2 + 28]\tau^6 + \frac{\kappa^2}{24} [(27f^2 - 13)\kappa^2 + 78]\tau^4 \\ &\quad - \frac{1}{24} [(9f^2 - 8)\kappa^2 + 6]\tau^2 - \frac{1}{3} - (-1)^{n+s} \frac{f^2 \kappa^2}{8} \end{aligned} \quad (III.52c)$$

definiert ist. Prinzipiell kann man noch Korrekturterme von höherer Ordnung in ϵ bestimmen, doch sind die erhaltenen Ausdrücke sehr unhandlich.

Ähnliche Formeln findet man in Ref. 77, 79, 82 für die Verteilungsfunktion beim *unkorrelierten* Random Walk. Um einen ersten Eindruck von der Verteilungsfunktion für den korrelierten Walk zu gewinnen, ist $p_n(s)$ in Bild III.3 für $n = 200$ und verschiedene Korrelationsfaktoren f aufgetragen. Die Daten wurden mit einer Erweiterung

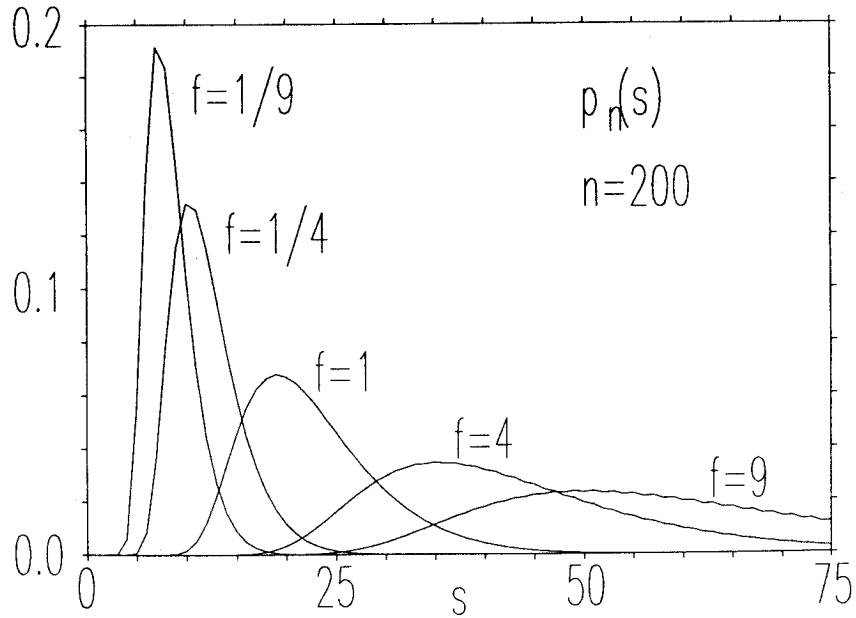


Bild III.3: Verteilungsfunktion $p_n(s)$ aus exakten Abzählungen. Die Funktion ist für $n = 200$ und verschiedene Korrelationsfaktoren f aufgetragen.

der exakten Abzählmethode von Havlin et. al. [27] auf korrelierte Walks berechnet. Mit steigendem f , d. h. mit immer stärker werdenden Vorwärtskorrelationen, verschiebt sich das Maximum der Verteilungsfunktion zu immer größeren s -Werten, und die Funktion als Ganzes wird breiter.

Ein paar Bemerkungen über die unendliche Summe in Gl. (III.52a) sind angebracht. Die Korrekturterme der Ordnung $O(\epsilon^2)$ enthalten alternierende Beiträge proportional zu $(-1)^{n+s}$. Falls man an einer glatten Verteilungsfunktion interessiert ist, genügt es, den führenden Beitrag zu berücksichtigen. In diesem Fall tragen nur ungerade k zur Summe bei. Für den unkorrelierten Fall, $f = 1$, wurde diese Formel von Rubin und Weiss [77, 79] angegeben. Für praktische Anwendungen benötigt man jedoch nicht die ganze, unendliche Summe.

Der $k = 1$ -Term allein genügt zur Beschreibung der Verteilungsfunktion nicht nur für kleine s -Werte, sondern auch für s -Werte bis über das Maximum hinaus. Erst wenn die Verteilungsfunktion wieder in die Nähe von Null gelangt, werden die Abweichungen beträchtlich, und man sollte den $k = 3$ -Term mit berücksichtigen. Das sollte für alle praktischen Probleme genügen. In Bild III.4 wird dies an Hand der rechten Flanke der Verteilungsfunktion demonstriert, indem die Summe in Gl. (III.52a) bis zu verschiedenen k -Werten ausgewertet wird. In der Abbildung ist der ϵ^2 -Korrekturterm mitgenommen worden.

Wichtiger als viele Terme der Summe mitzunehmen ist es, bei mittleren Schritt-

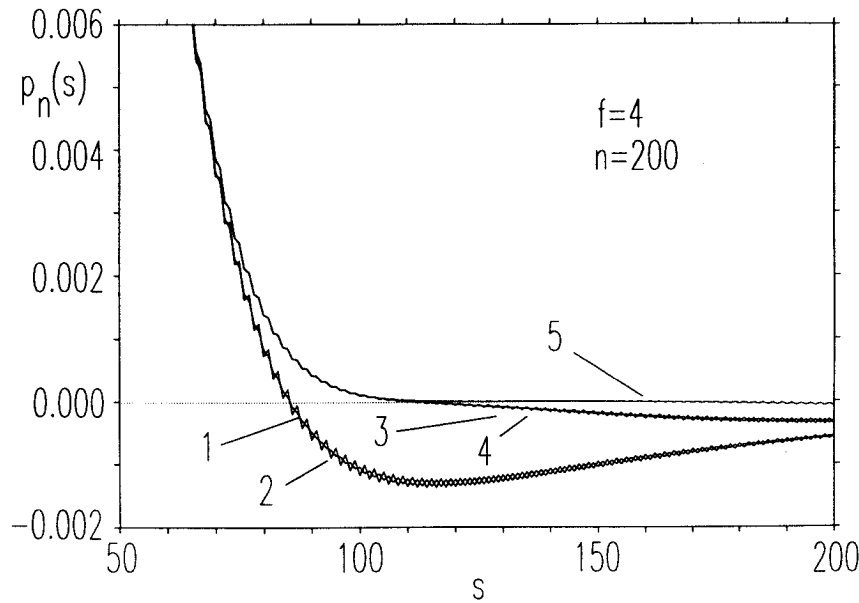


Bild III.4: Rechte Flanke der Verteilungsfunktion $p_n(s)$. Plot von Gl. (III.52) für $f = 4$, wobei die Summe von $k = 1$ bis zu der jeweils an der Kurve angegebenen Zahl läuft.

zahlen n die Korrekturterme von höherer Ordnung in ϵ zu berücksichtigen. In Bild III.5a ist die Verteilungsfunktion, Gl. (III.52) einschließlich des $k = 5$ -Terms, für $n = 200$ und $f = 9$ aufgetragen. Eine solch hohe Vorwärtskorrelation wurde gewählt, um die gerade/ungerade-Oszillationen zu betonen, die für nicht zu große Schrittzahlen auftreten. Der relative Fehler von Gl. (III.52) im Vergleich zu den exakten numerischen Daten ist in Bild III.5b dargestellt. Der Fehler ist beträchtlich, wenn man nur den führenden, glatten Beitrag b_0 berücksichtigt. Er wird deutlich reduziert, wenn man den Korrekturterm b_2 , der alternierende Beiträge enthält, mit einschließt.

Obwohl die asymptotische Theorie für große Schrittzahlen n entwickelt wurde, ist der relative Fehler unter 2% für $n = 200$ und dem großen Korrelationsfaktor $f = 9$, bei dem die Sprungrichtung im Durchschnitt nur 20 mal geändert wird.

Im führenden Beitrag hängt $p_n(s)$ von f , n und s nur über ϵ und τ ab. Das bedeutet, daß asymptotisch die Skalenform

$$p_n^{\text{kor.}}(s) = p_{fn}^{\text{unkorr.}}(s + f - 1) \quad (\text{III.53})$$

gültig ist. Eine einfache Konsequenz ist der Ausdruck Gl. (III.3) für die mittlere Zahl der besuchten Plätze, die in Ref. 67 mit direkten Methoden gefunden wurde. Doch zeigt Gl. (III.53), daß sich neben der Umskalierung $n \rightarrow fn$ auch die Verschiebung um $1 - f$ auf die ganze Verteilungsfunktion bezieht und nicht nur auf ihren Mittelwert. Ebenfalls folgt hieraus, daß sich die Breite der Verteilung aus derjenigen für den unkorrelierten Walk einfach durch Umskalierung $n \rightarrow fn$ finden läßt, da die Verschiebung hier keinen

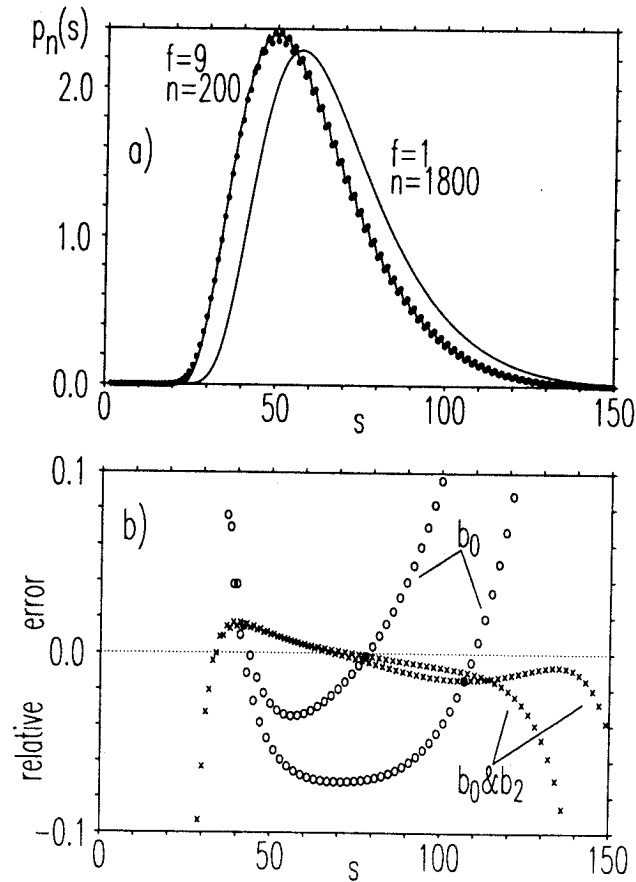


Bild III.5: Verteilungsfunktion $p_n(s)$ und relative Fehler.

- (a) Die Punkte sind exakte Daten und die durchgezogenen Linien sind Gl. (III.52) mit $k = 1, \dots, 5$ einschließlich des b_2 -Terms
- (b) Relative Fehler von Gl. (III.52) bezogen auf exakte numerische Daten. $\circ$: nur der b_0 -Term wurde berücksichtigt; $\times$: Auch der b_2 -Term wurde mit eingeschlossen.

Einfluß hat. In Bild III.5a sind die exakten Daten für $p_n(s)$ auch für $f = 1$ und $n = 1800$ mit eingezeichnet. Die Verschiebung dieser Funktion gegenüber derjenigen mit $n = 200$ und $f = 9$ ist tatsächlich konsistent mit einem Betrag von $f - 1 = 8$. Deshalb ist die Skalenform Gl. (III.53) tatsächlich auch für mittlere Schrittzahlen eine gute Näherung.

III.5. ÜBERLEBENSWAHRSCHEINLICHKEIT

In diesem Kapitel wird die Überlebenswahrscheinlichkeit Ψ_n für korrelierte Random Walks auf linearen Ketten mit zufällig verteilten Haftstellen der Konzentration c berechnet. Der Ausgangspunkt ist Gl. (III.4). Die Verteilungsfunktion $p_n(s)$ wurde im asymptotischen Bereich $n, s \rightarrow \infty$, mit $n^{1/2}/s$ endlich, im vorherigen Kapitel berechnet. Diese Funktion muß mit einer exponentiell abfallenden Funktion $(1-c)^s = \exp(-\lambda s)$ gewichtet werden, und es ist bekannt, daß die größten Beiträge von s -Werten kommen, bei denen $n^{1/3}/s$ endlich ist [10, 13]. Das gilt zumindest im unkorrelierten Fall, doch erwartet man keine prinzipiellen Änderungen für korrelierte Random Walks. Asymptotisch ist dann aber $n^{1/2}/s$ sehr groß, so daß die asymptotische Formel Gl. (III.52) keine systematische Entwicklung in diesem Bereich darstellt, da τ nicht mehr von der Größenordnung $O(1)$ ist. Im folgenden ist es bequem,

$$\begin{aligned}\Lambda &:= \pi k \lambda \\ x &:= (\pi k \lambda)^{2/3} (f_n)^{1/3} \gg 1 \\ t &:= \frac{s' \lambda}{x} = (\pi k)^{-2/3} \lambda^{1/3} n^{-1/3} s' = O(1).\end{aligned}\tag{III.54}$$

($s' := s + f - 1$, wie vorher) als Abkürzungen einzuführen. Wie im vorherigen Abschnitt kann man nun eine Potenzreihe für $f_n(s)$ herleiten, bei der aber nun t als Parameter der Größenordnung $O(1)$ auftritt und $1/x$ die kleine Größe ist, nach der systematisch entwickelt wird. Ausgehend von Gl. (III.49) und Gl. (III.50) kann dies leicht berechnet werden. Das Ergebnis ist

$$f_n(s) = \sum_{k=1}^{\infty} \frac{8\lambda t}{\Lambda^2} x \left[1 - \frac{(3f^2 - 1)\Lambda^2}{24t^4} \frac{1}{x} + \dots + O(x^{-5}) \right] \exp\left(-\frac{x}{2t^2}\right), \tag{III.55}$$

wobei wegen der Kürze nur ein Korrekturterm angegeben wurde. Um das endgültige Resultat weiter unten zu erhalten, ist es nötig, $f_n(s)$ bis zur Ordnung $O(x^{-4})$ zu entwickeln. Das ist kaum fehlerfrei per Hand zu machen und tatsächlich wurde die Programmiersprache REDUCE für Formelmanipulation benutzt, um die Berechnungen durchzuführen.

Unter der Annahme von $n^{1/3}/s$ endlich führen die Ableitungen von $f_n(s)$ nach s auf Terme derselben Größenordnung wie $f_n(s)$ selbst. Deshalb kann die Taylorreihe diesmal nicht benutzt werden, um $p_n(s) = f_n(s+1) - 2f_n(s) + f_n(s-1)$ systematisch zu entwickeln. Durch Verschieben des Summationsindex in Gl. (III.4) erhält man jedoch

$$\Psi_n = \frac{c^2}{1-c} \sum_{s=1}^{\infty} f_n(s) e^{-\lambda s}, \tag{III.56}$$

wobei $f_1(s) = f_2(s) = 0$ (gültig für $n > 0$) benutzt wurde. Die Summation kann durch Integration ersetzt werden, denn es kann gezeigt werden, siehe Ref. 32, Anhänge H und I, daß

$$\Psi_n = \frac{c^2}{1-c} \int_0^\infty f_n(s) e^{-\lambda s} ds + O\left(x^{3/2} e^{-3/2 ax}\right), \quad \text{mit } a > 1 \quad (III.57)$$

gilt. Der Korrekturterm kann vernachlässigt werden, wie weiter unten klar werden wird. Das Integral kann auf Integrale der Form

$$I_j := \int_0^\infty t^j \exp\left[-x\left(\frac{1}{2t^2} + t\right)\right] dt \quad (III.58)$$

zurückgeführt werden. Diese wurden ebenfalls in Ref. 32, Kapitel II.7 und Anhänge A und B, als asymptotische Reihen in $1/x$ für $x \gg 1$ berechnet. Der Fall korrelierter Walks erfordert die Auswertung von Gl. (III.58) auch für gerade j , im Gegensatz zu Ref. 32. Deshalb sind in Anhang C die Ergebnisse für I_j für die benötigten j -Werte in der benötigten Genauigkeit ohne Herleitung aufgelistet.

Es zeigt sich, daß nur der $k = 1$ -Term bei der Berechnung der Überlebenswahrscheinlichkeit benötigt wird. Man erhält schließlich das Resultat für die Überlebenswahrscheinlichkeit Ψ_n auf linearen Ketten mit zufällig verteilten Haftstellen

$$\Psi_n = \frac{8c^2}{(1-c)f\lambda^2} \left(\frac{2x^3}{3\pi^3}\right)^{\frac{1}{2}} \exp\left[-\frac{3}{2}x + \frac{b_1}{x} + \frac{b_2}{x^2} + \frac{b_3}{x^3} + \frac{b_4}{x^4} + O(x^{-5})\right] \quad (III.59a)$$

mit

$$\begin{aligned} b_1 &= \frac{17}{18} - \frac{3f^2 - 1}{24} \Lambda^2 \\ b_2 &= -\frac{7}{54} + \frac{3f^2 - 7}{36} \Lambda^2 + \frac{f(f^2 - 1)}{6\pi} \Lambda^3 \\ b_3 &= -\frac{26}{243} - \frac{21f^2 - 115}{648} \Lambda^2 - \frac{f(f^2 - 1)}{6\pi} \Lambda^3 - \frac{45f^4 - 30f^2 - 7}{2160} \Lambda^4 \\ b_4 &= \frac{167}{972} - \frac{f^2 + 11}{108} \Lambda^2 + \frac{17f(f^2 - 1)}{162\pi} \Lambda^3 \\ &\quad + \left[\frac{45f^4 + 45f^2 - 59}{2160} + \frac{f^2}{32} (-1)^n \right] \Lambda^4 + \frac{f(f^2 - 1)(21f^2 + 11)}{1080\pi} \Lambda^5 \end{aligned} \quad (III.59b)$$

und

$$\begin{aligned} \Lambda &= \pi\lambda \\ x &= (\pi\lambda)^{2/3} (fn)^{1/3}. \end{aligned} \quad (III.59c)$$

Nun ist auch klar, daß man die Korrektur in Gl. (III.57) vernachlässigen kann. Eigentlich entstehen bei obiger Rechnung die Korrekturterme als Terme vor der exp-Funktion.

Es ergibt sich jedoch eine wesentliche Vereinfachung, wenn man die Terme, wie oben geschehen, in den Exponenten schreibt.

Die Form der Überlebenswahrscheinlichkeit stimmt mit den Ergebnissen von Lubensky [33] und Schotland [34] überein, die mit feldtheoretischen Methoden die Art der Korrekturen in beliebigen Dimensionen gefunden haben. Die Koeffizienten b_2, b_3 etc. und den Exponenten des Vorfaktors $x^{3/2} \propto n^{1/2}$ haben sie jedoch nicht explizit angegeben.

Obige Rechnung wurde bis zu der Ordnung durchgeführt, bei der gerade/ungerade Oszillationen auftreten. Falls man nur an einer glatten Überlebenswahrscheinlichkeit interessiert ist, genügt es, die ersten drei Korrekturterme mitzunehmen. In führender Ordnung gilt wieder eine einfache Skalenbeziehung

$$\Psi_n^{\text{kor.}} = \frac{1}{(1-c)^{f-1}} \Psi_{fn}^{\text{unkorr.}}, \quad (\text{III.60})$$

die aber auch schon aus Gl. (III.4) und Gl. (III.53) folgt. Der Vorfaktor hat seinen Ursprung in der Verschiebung der Verteilungsfunktion $p_n(s)$ um $1-f$. Für kleine Haftstellenkonzentrationen c , so daß $\Lambda \ll 1$ und $\Lambda f \ll 1$, kann man die konzentrationsabhängigen Terme in den Korrekturtermen $b_1, b_2, \dots$ vernachlässigen. Ebenso kann man dann den zusätzlichen, konzentrationsabhängigen Vorfaktor

$$\frac{c^2}{(1-c)\lambda^2} = 1 + \frac{c^2}{12} + O(c^3) \quad (\text{III.61})$$

vernachlässigen (selbst für $c = 0.5$ ist seine Abweichung von 1 kleiner als 4%). In diesem Fall, dem physikalisch relevanten Fall, hängt die skalierte Überlebenswahrscheinlichkeit $(1-c)^{f-1} \Psi_n$ von der Haftstellenkonzentration c und der Schrittzahl n nur noch über die Skalenvariable x ab.

In Bild III.6 ist $(1-c)^{f-1} \Psi_n$ gegen x für verschiedene Korrelationsfaktoren f und verschiedene Konzentrationen c aufgetragen. Die Punkte sind exakte, numerische Werte, die Linien repräsentieren Gl. (III.59) in verschiedenen Ordnungen, wobei alle oben erwähnten Terme vernachlässigt wurden. Nur die führende Ordnung (0) und die erste Korrektur (1) wurden dargestellt. Schließt man noch weitere Korrekturen mit ein, sind die Kurven auf der gewählten Skala nicht von Kurve (1) zu unterscheiden. Nur für extrem hohe Haftstellenkonzentrationen in Kombination mit hohen Korrelationsfaktoren (z. B. $c = 0,5$ und $f = 4$) kann man noch eine Abweichung von der Skalenform finden. Für Überlebenswahrscheinlichkeiten die größer sind als etwa $10^{-3}/(1-c)^{f-1}$, ist der Einfluß des ersten Korrekturterms beträchtlich, und man kann nicht erwarten, das asymptotische Verhalten in einem Experiment zu sehen. Die Rosenstock-Näherung

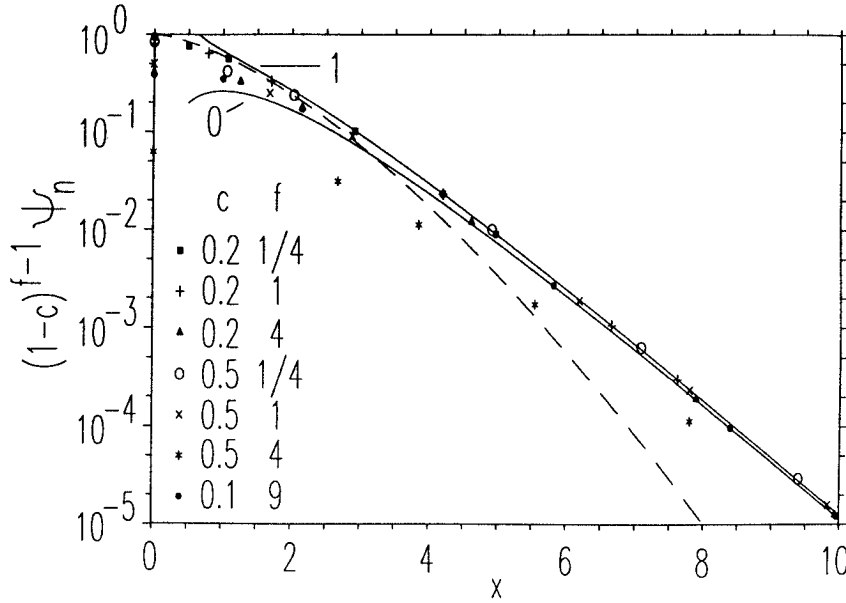


Bild III.6: Skalenplot der Überlebenswahrscheinlichkeit. •: skalierte Überlebenswahrscheinlichkeit für verschiedene Haftstellenkonzentrationen c und Korrelationsfaktoren f von exakten Abzählungen. Durchgezogene Linien: Gl. (III.59) mit $\Lambda = 0$ in den Korrekturtermen (siehe Text!). Unterbrochene Linie: Rosenstock-Näherung.

[8, 9], der erste Term der Kumulantenentwicklung von Gl. (III.4), liest sich in den hier benutzten Variablen als

$$\Psi_n^{\text{Ros.}} = e^{-\lambda \langle s_n \rangle} = \frac{1}{(1-c)^{f-1}} \exp \left(-\sqrt{8x^3/\pi^3} \right), \quad (\text{III.62})$$

wenn man Gl. (III.3) einsetzt. Sie hat dieselbe Skaleneigenschaft wie die asymptotische Entwicklung, doch ist sie nur für kleine Haftstellenkonzentrationen und relativ hohe Überlebenswahrscheinlichkeiten eine gute Näherung. Bild III.6 zeigt nun, daß man mit der asymptotischen Theorie Gl. (III.59) schon unter Mitnahme des ersten Korrekturterms eine Formel erhält, die selbst für so hohe Überlebenswahrscheinlichkeiten von der Ordnung $O(10^{-1})$ besser als die Rosenstock-Näherung ist.

Um den Einfluß der Korrekturterme auf die Überlebenswahrscheinlichkeit Ψ_n zu demonstrieren, wurde sie in Bild III.7 als Funktion von x für zwei verschiedene Parameter-Wahlen aufgetragen.

Es ist instruktiv, den effektiven Exponent α als

$$\alpha := \ln \left(\frac{\ln \Psi_{n+1}}{\ln \Psi_{n-1}} \right) / \ln \left(\frac{n+1}{n-1} \right) \quad (\text{III.63})$$

zu definieren.

In Bild III.8 ist α gegen $1/x$ für verschiedene c und f aufgetragen. Der asymptotische Wert $1/3$ wird erst für so große x -Werte erreicht, bei denen die Überlebenswahr-

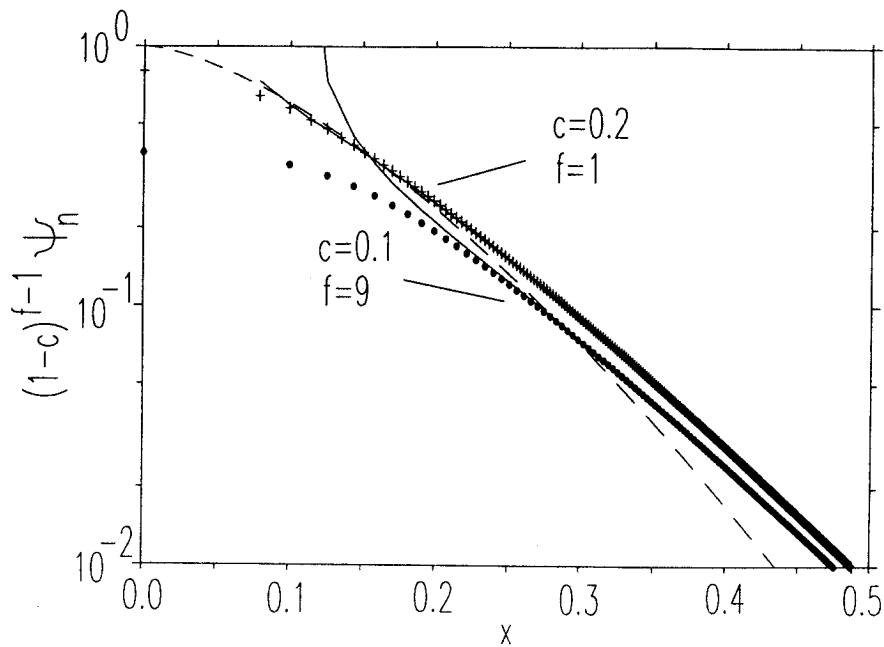


Bild III.7: **Einfluß der Korrekturterme auf die Überlebenswahrscheinlichkeit.** Skalierte Überlebenswahrscheinlichkeiten für verschiedene c und f aus exakten Abzählungen. Durchgezogene Linien: Gl. (III.59) einschließlich Korrekturterme bis b_4 . Unterbrochene Linie: Rosenstock-Näherung Gl. (III.62).

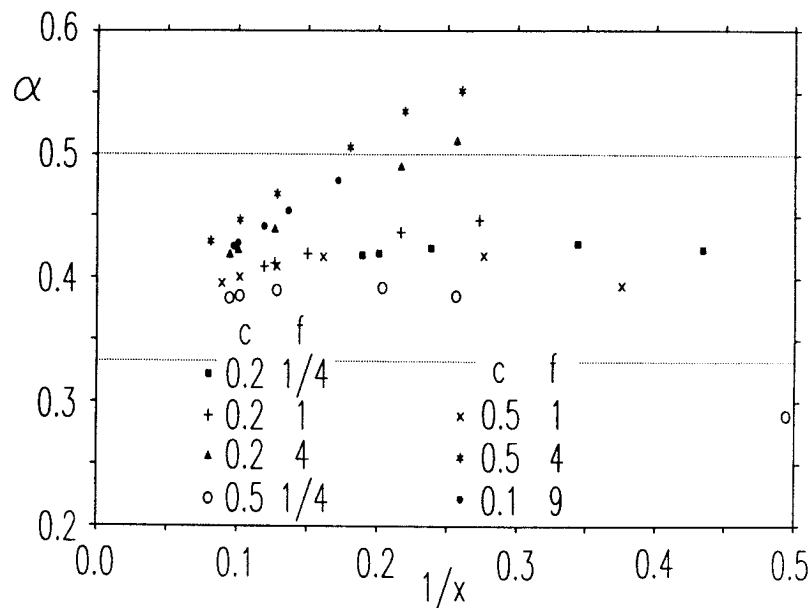


Bild III.8: **Effektiver Exponent α .** Punkte: effektiver Exponent α für verschiedene c und f von exakten Abzählungen. Gestrichelte Linien: Rosenstock-Wert $1/2$ bzw. asymptotischer Wert $1/3$.

scheinlichkeit unter jeder physikalisch relevanten Schwelle liegt. Aber selbst für mittlere x -Werte ist die Abweichung vom Rosenstock-Wert $1/2$ beträchtlich.

IV. Entwicklung eines neuen Monte-Carlo-Verfahrens für Haftstellen-Probleme in höheren Dimensionen

In diesem Kapitel wird die Entwicklung eines neuen Monte-Carlo-Verfahrens (im folgenden kurz MC-Verfahren) zur Simulation von Haftstellen-Problemen in d Dimensionen beschrieben. Dazu wird zunächst ein Überblick über die bekannten Verfahren gegeben, wobei das besondere Augenmerk auf die Vor- und Nachteile der verschiedenen Methoden gelegt wird. Es wird sich zeigen, daß alle Verfahren an ihre Grenzen stoßen, wenn es gilt, das asymptotische Verhalten für große Schrittzahlen n zu untersuchen. Daraus resultiert die Forderung nach einer Methode, die im wesentlichen nur die relevanten Random Walks simuliert (*"importance sampling"*). Das neue MC-Verfahren, das im folgenden dargestellt wird, benutzt eine Art Metropolis-Algorithmus, um *importance sampling* zu erreichen, und es wird gezeigt werden, daß das Verfahren gegen das richtige Statistische Ensemble konvergiert. Es schließt sich eine Betrachtung über Korrelationszeiten mit dem Ergebnis an, daß das neue Verfahren allen bekannten überlegen ist. Dazu trägt auch bei, daß die Methode zu einem großen Teil vektorisiert, was durch die gleichzeitige Simulation vieler Walks und Speicherplatzeinsparung durch Hash-Tabellen ermöglicht wird. Zum Schluß werden noch einige weitere technische Details des Programms erläutert, dessen Listing sich im Anhang D befindet.

Da hier nicht auf alle Einzelheiten von MC-Verfahren eingegangen werden kann, wird auf das Buch, das K. Binder [83] herausgegeben hat, verwiesen (siehe auch Ref. 84, 85).

IV.1. ÜBERBLICK ÜBER BEKANNTE SIMULATIONSVERFAHREN

Im folgenden werden die bisher bekannten und eingesetzten Verfahren zur Simulation von Haftstellen-Problemen kurz erläutert. Aus den Nachteilen dieser Methoden ergeben sich Forderungen an einen neuen Algorithmus, der anschließend entwickelt wird.

IV.1.1. Direkte Simulation: Teilchen auf Gitter mit Haftstellen

Die direkteste Art, das Haftstellen-Problem auf einem Computer zu simulieren, besteht darin, ein konkretes Gitter mit Haftstellen zu speichern und ein Teilchen auf diesem Gitter diffundieren zu lassen, bis es von einer Haftstelle eingefangen wird. Bei dieser Methode überlebt aber nur einer von $1/\Psi$ Walks. Alle anderen simulierten Walks fallen für die Statistik aus. Da die interessanten Effekte aber erst bei Schrittzahlen zu erwarten sind, bei denen die Überlebenswahrscheinlichkeit Ψ sehr klein ist, etwa 10^{-8} , siehe unten, müßte man ein Vielfaches von 10^8 Walks simulieren, um mit genügender Statistik in den interessanten Bereich vorstoßen zu können. Das ist mit derzeitigen

Rechnern nicht möglich. Selbst wenn das aus Rechenzeitgründen in Zukunft möglich werden sollte, würden doch die Argumente, die weiter unten die Havlin-Methode aus Speicherplatzgründen in ihrer Anwendbarkeit einschränken, ebenso für die naive Methode gelten. Lediglich für kleine Schrittzahlen und kleine Haftstellenkonzentrationen, also für hohe Überlebenswahrscheinlichkeiten, liefert es gute Resultate.

IV.1.2. Havlin Methode: Mastergleichung exakt auf Gitter mit Haftstellen

Eine andere Möglichkeit ist die von S. Havlin et. al. benutzte Methode [27, 28], bei der eine konkrete Realisierung eines Gitters mit Haftstellen abgespeichert wird, dann aber die Mastergleichung für die Aufenthaltswahrscheinlichkeiten des Teilchens numerisch exakt berechnet wird (*“exact enumeration on random configurations”*). Man kann dabei das Teilchen auf allen Gitterplätzen mit der gleichen Wahrscheinlichkeit starten lassen und die Überlebenswahrscheinlichkeit Schritt für Schritt exakt bestimmen. Der Algorithmus läuft auf Vektorrechnern sehr schnell, da er voll vektorisierbar ist. Bei dieser Methode wird die Random Walk Mittelung exakt behandelt, während die Haftstellen-Konfigurations-Mittelung durch die Simulation eines großen Gitters (Selbstmittelung) erreicht wird. Es muß jedoch darauf geachtet werden, daß das Gitter so groß ist, daß die relevanten Haftstellen-freien Gebiete auch tatsächlich genügend oft vorkommen, denn der asymptotische Bereich wird von seltenen, relativ großen Haftstellen-freien Gebieten dominiert, siehe Ref. 10, 13 und die Diskussion vor Gl. (II.10). Das ist der begrenzende Faktor dieser Methode. Da p^s etwa die Wahrscheinlichkeit ist, daß ein gegebener Gitterplatz im Zentrum einer Haftstellen-freien Kugel vom Radius R mit $s_{\max} = \Omega_d R^d$ Plätzen liegt, siehe Gl. (II.13), ist die mittlere Zahl von solchen Zentren etwa $L p^s$, falls L Gitterplätze im Computer Platz finden. Diese Zahl muß groß gegen 1 sein. Daraus folgt sofort als Kriterium für die Anwendbarkeit der Havlin-Methode

$$n\lambda^{2/d} \ll \left(\frac{1}{\Omega_d}\right)^{\frac{2}{d}} \left(\frac{d}{\alpha_d}\right)^2 (\ln L)^{\frac{d+2}{d}}, \quad (IV.1)$$

wobei Ω_d in Gl. (II.11) und α_d in Gl. (II.12) definiert sind. Für erreichbare Werte von L der Größenordnung von z. B. 10^6 ergibt dies in $d = 3$ Dimensionen

$$n\lambda^{2/3} \ll 27,9 ,$$

d. h., schon für Schrittzahlen von $n = 100$ ist man auf Haftstellenkonzentrationen $c \ll 0,14$ beschränkt. Nur in diesem Bereich kann man realistische Simulationsergebnisse mit der Havlin-Methode erwarten. Es wird sich aber weiter unten zeigen, daß der interessante Bereich bei höheren c bzw. größeren n -Werten liegt.

Bei dieser Methode werden im asymptotischen Bereich also fast nur zu kleine Haftstellen-freie Gebiete simuliert, die kaum zur Überlebenswahrscheinlichkeit beitragen. Eine gewisse Abhilfe schafft die zweite von S. Havlin et. al. benutzte Methode [28], bei der die Mastergleichung auf allen kleinen, Haftstellen-freien Gebieten numerisch gelöst wird (*“exhaustive enumeration of configurations”*). Die Gebiete sind Cluster von normalen Gitterplätzen, die von Haftstellen völlig umschlossen sind. Man kann nun die Überlebenswahrscheinlichkeit systematisch als Summe von Beiträgen aller Cluster schreiben. Die Abzählung der topologisch verschiedenen Cluster und die Feststellung ihrer Symmetrieeigenschaften ist aber so aufwendig, daß die Methode bisher auf Clustergrößen bis etwa 12 Gitterplätzen in $d = 2$ und $d = 3$ beschränkt blieb. Die Reihe konvergiert aber nur für extrem große Haftstellenkonzentrationen und kleine Schrittzahlen schnell genug, da sonst aus obigen Gründen größere Haftstellen-freie Gebiete den Hauptbeitrag zur Überlebenswahrscheinlichkeit liefern.

Mit s_c als maximal handhabbare Clustergröße folgt mit Gl. (II.13) aus $s_{\max}(c, n) \ll s_c$ für den Gültigkeitsbereich der Methode

$$\frac{n}{\lambda} \ll \frac{1}{\Omega_d^{2/d}} \left(\frac{d}{\alpha_d} \right)^2 s_c^{\frac{d+2}{d}}. \quad (IV.2)$$

Hier müßte eigentlich noch $s_c \gg 1$ vorausgesetzt werden, aber diese Bedingung ist mit $s_c = 12$ sowieso nur bedingt erfüllt. Für extrem hohe Haftstellenkonzentrationen c liefert diese Methode gute Ergebnisse, doch ist es aufgrund der Erfahrungen mit dem eindimensionalen Modell [32] fraglich, ob man damit das asymptotische Verhalten gut untersuchen kann, da dann Korrekturterme, die keine Skaleneigenschaft mehr haben, wichtiger werden können, als der *“führende”* Term.

IV.1.3. Teilchen auf Gitter ohne Haftstellen, Haftstellen-Mittelung exakt

Eine weitere Methode ist die Simulation der Verteilungsfunktion der Anzahl der besuchten Plätze auf einem Gitter ohne Haftstellen, aus der sich die Überlebenswahrscheinlichkeit nach Gl. (II.8) berechnen läßt. In dieser Gleichung ist die Haftstellen-Mittelung schon exakt ausgeführt. Diese Methode ist häufig für diskrete Walks angewandt worden [15, 16]. Auch Verallgemeinerungen für Walks in kontinuierlicher Zeit sind simuliert worden [18]. Diese Autoren haben die Verteilungsfunktion $p_n(s)$ für verschiedene Gitter und Fraktale [19, 23–25] untersucht. Die so erhaltenen Überlebenswahrscheinlichkeiten wurden mit der Kumulantenentwicklung, siehe z. B. Ref. 86, für Gl. (II.8) verglichen. Für *kleine* Schrittzahlen ergab sich eine gute Übereinstimmung, jedoch sollte beachtet werden, daß die Kumulanten mit Hilfe derselben Simulation bestimmt wurden. Würde man analytische Ausdrücke, die immer asymptotische Formeln

sind (also für *große* Schrittzahlen gelten), für die Kumulanten einsetzen, würde man wesentlich schlechtere Übereinstimmungen finden.

Zur Abschätzung des Gültigkeitsbereiches der Methode kann folgende Überlegung dienen. Die größten Beiträge zur Überlebenswahrscheinlichkeit kommen von relativ kleinen s -Werten, die in der Nähe von $s_{\max}$ nach Gl. (II.13) liegen. $s_{\max}$ selbst bestimmt sich aus dem Sattelpunkt bzgl. s des Produktes $p_n(s) \exp(-\lambda s)$ und asymptotisch gilt

$$\Psi_n \sim p_n(s_{\max}) e^{-\lambda s_{\max}}. \quad (IV.3)$$

Damit ist der exponentielle Anteil zu $p_n(s)$ bei $s = s_{\max}$

$$p_n(s_{\max}) \sim \exp \left[-\frac{d}{2} \left(\frac{\Omega_d \alpha_d^d}{d^d} \right)^{\frac{2}{d+2}} \lambda^{\frac{2}{d+2}} n^{\frac{d}{d+2}} \right]. \quad (IV.4)$$

Nicht exponentielle Korrekturen zu der Wahrscheinlichkeit sind für die folgende Argumentation irrelevant. Dies ist also die Wahrscheinlichkeit, daß ein normaler Random Walk einen s -Wert hat, der zufällig in der Nähe von $s_{\max}$ liegt. Simuliert man insgesamt N_{RW} Random Walks, so ist $N_{\text{RW}} p_n(s_{\max})$ die zu erwartende Zahl der Walks, die merklich zur Überlebenswahrscheinlichkeit beitragen. Diese Zahl sollte groß gegen 1 sein. Daraus ergibt sich das Kriterium

$$n \lambda^{2/d} \ll \frac{1}{\Omega_d^{2/d}} \left(\frac{d}{\alpha_d} \right)^2 \left(\frac{2}{d} \ln(N_{\text{RW}}) \right)^{\frac{d+2}{d}}. \quad (IV.5)$$

Bis auf numerische Konstanten sieht der Gültigkeitsbereich formal genauso aus wie derjenige bei der Havlin-Methode auf zufälligen Haftstellen-Konfigurationen, Gl. (IV.1). Hier ist nur die Speichergröße L ersetzt durch die Anzahl der zu simulierenden Walks N_{RW} , die ihrerseits proportional zur verfügbaren Rechenzeit ist. Im einen Fall begrenzt der vorhandene Speicherplatz, im anderen Fall die verfügbare Rechenzeit den Wert des Parameters $n \lambda^{2/d}$. Zu beachten ist ferner, daß der Speicherplatz bzw. die Rechenzeit logarithmisch eingehen, weshalb eine Vergrößerung der Parameter keine signifikante Verbesserung bringt.

IV.1.4. Biased sampling

Bei der zuletzt beschriebenen Methode ist das Problem, daß die simulierten Walks fast immer eine zu große Zahl von Plätzen besuchen. Nur die relativ seltenen, kompakten Walks mit einer Zahl von besuchten Plätzen im Bereich von $s_{\max}$ tragen zur Statistik bei. Es liegt nahe, die Walks durch Einführung eines sogenannten "*Bias*" dazu zu zwingen, weniger Plätze zu besuchen als normal. ("*Bias*" bedeutet im Englischen soviel wie *Vorurteil*.) Diese Methode wird auf Rosenbluth und Rosenbluth [87] zurückgeführt, die sie für SAW's (Self Avoiding Walks) bei $T = \infty$ anwandten. Sie wurde von K. Kremer für endliche T weiterentwickelt [88] und dient im allgemeinen dazu, Verteilungsfunktionen abseits von den Maxima mit besserer Statistik zu bestimmen.

Übertragen auf das Haftstellen-Problem oder auf die SIW-Kette würde man bei der Simulation der Random Walks die statistischen Gewichte für den nächsten Schritt davon abhängig machen, ob ein neuer Platz besucht würde oder nicht. Wie weiter oben diskutiert, siehe Abschnitt II.4.2, ist es nicht möglich, auf diese Weise SIW's mit der exakten Verteilungsfunktion zu erzeugen. Man würde Walks mit einer anderen Verteilungsfunktion bekommen. Da man jedoch die realisierten, statistischen Gewichte während des Aufbaus des Walks verfolgen kann, kennt man das Gesamtgewicht des so erzeugten Walks und kann den Fehler (den *Bias*) korrigieren. Im Limes unendlich vieler, simulierter Walks ist dieses Vorgehen exakt. Für endliche Rechenzeit wird jedoch das Ensemble der erzeugten Walks stark verzerrt. Das äußert sich zum Beispiel darin, daß Walks, die die gleiche Gestalt haben, aber in umgekehrter Richtung erzeugt wurden, mit unterschiedlichen Gewichten in der Simulation vorkommen. (Siehe auch die Diskussion zu Bild II.3 auf Seite 15.) Deshalb ist nicht klar, ob die Einführung eines starken *Bias* den Zustandsraum aller Walks nicht so stark verzerrt, daß er nur sehr ungleichmäßig abgetastet wird. Die Verbesserung der Statistik ist dann leider nur scheinbar beträchtlich.

Ferner ist die Wahl eines *Bias* immer recht willkürlich. Es müssen bei der Simulation Vorversuche gemacht werden, um den optimalen *Bias* zu finden.

Um das asymptotische Verhalten beim Haftstellenproblem zu untersuchen, muß der *Bias* gegenüber dem Random Walk recht groß sein, um merkliche Verbesserungen in der Statistik zu bekommen. Eine quantitative Abschätzung des Effektes auf den simulierbaren Parameterbereich scheint nicht möglich zu sein, da durch die Einführung des *Bias* ein völlig neues, kompliziertes Walk Modell entsteht.

Da also nicht klar ist, welche Effekte die Einführung eines starken *Bias* hat, wird im folgenden diese Methode nicht weiter diskutiert. Auch K. Kremer betont in seiner

Dissertation [88], daß die Methode mit Vorsicht anzuwenden ist. Die Ergebnisse müssen immer sorgfältig mit denen anderer Verfahren verglichen werden.

IV.1.5. Scanning Methode

Eine logische Weiterentwicklung des “*biased sampling*” ist die Scanning Methode von H. Meirovitch [89, 90]. Hier wird nicht nur der nächste Schritt zur Entscheidung herangezogen, welche Richtung eingeschlagen werden soll, sondern die nächsten b Schritte. Für $b = 1$ reduziert sich das Verfahren auf das *biased sampling* aus dem vorherigen Abschnitt.

Ursprünglich für SAW's entwickelt, läßt sich die Methode ohne Schwierigkeiten auch auf SIW's anwenden [38, 91]. Die Methode soll im folgenden nur ganz knapp beschrieben werden.

Seien zu einem Zeitpunkt schon k Schritte des Walks konstruiert worden. Dann berechnet man für die nächsten b Schritte eine exakte “zukünftige Zustandssumme”

$$Z_k(\nu, b) = \sum_{j=1}^{(2d)^{b-1}} e^{-\lambda s_j}, \quad (IV.6)$$

wobei in den b zukünftigen Schritten s_j neue Plätze besucht werden und die Summe über alle $(2d)^{b-1}$ denkbaren Fortsetzungswalks geht, die in eine gegebene Richtung ν starten. Diese Zustandssumme wird exakt berechnet, indem alle Walks durchprobiert werden, weshalb b nicht zu groß gewählt werden darf. Nun definiert man eine Übergangswahrscheinlichkeit

$$p_k(\nu, b) = \frac{Z_k(\nu, b)}{\sum_{\nu=1}^{2d} Z_k(\nu, b)}, \quad (IV.7)$$

mit der die Richtung ν anschließend ausgelost wird. Auf diese Weise wird ein Walk ω mit n Schritten erzeugt, dessen Konstruktionswahrscheinlichkeit durch

$$P_\omega(b) = \frac{1}{2d} \prod_{k=2}^n p_k(\nu_k, b), \quad (IV.8)$$

gegeben ist, wenn ν_k die tatsächlich gewählten Richtungen beim k -ten Schritt sind.

Würde man jeweils für b den Rest des noch zu erzeugenden Walks wählen, würde die Methode per definitionem die exakten SIW's liefern, da bei jedem Schritt genau mit der richtigen Wahrscheinlichkeit in die jeweilige Richtung gesprungen wird. Das geht natürlich für relevante Schrittzahlen n wegen der endlichen Rechenzeit nicht. Deshalb

wird b typischerweise zu 1–4 gewählt. Dadurch wird aber wieder ein *Bias* induziert, den man korrigieren muß. Es gilt für den Mittelwert einer beliebigen Größe V_ω

$$\langle V \rangle = \frac{\langle e^{-\lambda s_\omega} V_\omega P_\omega^{-1}(b) \rangle}{\langle e^{-\lambda s_\omega} P_\omega^{-1}(b) \rangle}. \quad (IV.9)$$

Hier wurde der *Bias* durch “*importance sampling*” [83] korrigiert. Meirovitch schlägt noch eine andere Methode vor, “Schmidt’s verallgemeinerte Monte Carlo Prozedur” [92], um von einem *biased* Ensemble eine Teilmenge von *unbiased* Walks zu extrahieren, auf die hier aber nicht weiter eingegangen werden soll.

Man kann sich überlegen, daß der *Bias* die Walks nicht weit genug komprimiert, deshalb führt Meirovitch einen weiteren, heuristischen Parameter ein, der die Walks in Richtung auf den Schwerpunkt der schon besuchten Plätze treibt. Dieser Parameter und die “Zukunftskettenlänge” b müssen durch Vorversuche optimiert werden.

Da die Konstruktionswahrscheinlichkeit jedes einzelnen Walks bekannt ist, kann auch die Entropie und die Freie Energie der SIW’s direkt bestimmt werden. Ferner können mit dieser Methode auch alle weiteren, interessanten Größen, wie die Rückkehrwahrscheinlichkeit, die mittlere quadratische Auslenkung, der Trägheitsradius, die mittlere Anzahl der besuchten Plätze und die Schwankungen hiervon untersucht werden.

Nachteile sind, daß die Methode sehr aufwendig ist, ohne zu einem *optimalen importance sampling* zu gelangen, bei dem die Walks mit einer Wahrscheinlichkeit, die dem statistischen Gewicht $e^{-\lambda s}$ exakt entspricht, in der Simulation auftauchen. Zum entstehenden *Bias* ist das gleiche zu sagen, wie im vorherigen Abschnitt. Insgesamt ist nicht klar zu überblicken, welchen Einfluß das Einführen eines weiteren heuristischen Parameters und der zusätzliche *Bias* auf das Ensemble der simulierten Walks hat. Es kann stark verzerrt sein und somit die berechneten, statistischen Fehler verfälschen.

Die von Meirovitch erzielten Ergebnisse zeigen, daß die Methode besser ist, als die vorher beschriebenen. Doch auch hier gilt, daß er bei der Simulation nicht weit genug in das asymptotische Gebiet vordringen konnte, so daß die Exponenten, die er gefunden hat, zum Teil erheblich von den richtigen Werten abweichen (siehe unten).

IV.2. FORDERUNGEN AN NEUEN ALGORITHMUS

Für die Suche nach einem besseren MC-Verfahren ergeben sich aus der vorhergegangenen Diskussion die folgenden Forderungen:

- **keine heuristischen Parameter**

Es sollen keine weiteren, künstlichen Parameter in der Simulation vorkommen, die das erzeugte Ensemble in unkontrollierbarer Weise verzerren, sondern die Simulation soll in natürlicher Weise genau die Walks generieren, die für das Haftstellenproblem relevant sind.

- **optimales Importance-Sampling**

Während der Simulation sollen die Walks mit einer Wahrscheinlichkeit auftauchen, die ihrem statistischen Gewicht $e^{-\lambda s}$ entspricht. Damit ist gewährleistet, daß nur die relevanten Walks simuliert werden und nicht Rechenzeit für Walks vergeudet wird, die ohnehin nicht merklich zu den zu berechnenden Mittelwerten beitragen.

- **Vektorisierbarkeit**

Der Algorithmus sollte möglichst vektorisierbar sein, da Vektorrechner, wie die in der Kernforschungsanlage Jülich verfügbare CRAY X-MP, in Zukunft eine immer größere Bedeutung erlangen werden. Diese Rechner entfalten ihre volle Leistungsfähigkeit erst bei vektorisierbaren Algorithmen. Alle oben genannten Simulationsverfahren, bis auf die Havlin-Methode, erfüllen diese Forderung nicht.

- **Sammeln vieler relevanter Daten**

Wenn es möglich ist, ein Ensemble von Random Walks mit den korrekten statistischen Gewichten zu erzeugen, sollten nicht nur die Überlebenswahrscheinlichkeit, sondern auch weitere interessante Größen, wie z. B. die Rückkehrwahrscheinlichkeit, die mittlere quadratische Auslenkung und der Trägheitsradius (alles bei Anwesenheit von Haftstellen) mitberechnet werden.

IV.3. REALISIERUNG DER FORDERUNGEN IN EINEM NEUEN MC-VERFAHREN

Das im folgenden beschriebene Monte-Carlo-Verfahren, das obige Forderungen erfüllt, wurde von mir im Rahmen der Dissertation entwickelt. Dabei wurden einige Methoden der Informatik verwendet, die offenbar bei Physikern, die sich mit Simulationen beschäftigen, nicht allgemein bekannt sind. Deshalb wird im folgenden der Algorithmus etwas ausführlicher beschrieben.

Als zugrunde liegende MC-Methode wird das Reptationsverfahren benutzt, das

von der Polymerphysik [93] her bekannt ist [94, 95]. Man kann mit diesem Verfahren dynamische Eigenschaften von Polymeren untersuchen, obwohl das Rousemodell [96], bei dem Modifikationen an inneren Teilstücken des Walks durchgeführt werden, wegen kürzerer Korrelationszeiten und der realistischeren Bewegungen besser dazu geeignet ist [88]. Die von mir benutzte Modifikation des Reptationsalgorithmus übertrifft jedoch für SIW's das Rousemodell bzgl. der Korrelationszeiten bei weitem, wie die weiter unten folgende Diskussion zeigen wird.

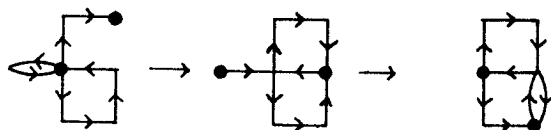


Bild IV.1: **Reptation eines Random Walks auf einem Quadratgitter.** Der Walk "kriecht" über das Gitter, indem jeweils der erste Schritt entfernt und ein neuer Schritt hinten angefügt wird.

Bei der Reptation wird nicht jeder Walk völlig neu erzeugt, sondern der Walk selbst bewegt sich als Ganzes schlangenartig über das Gitter. Das wird dadurch erreicht, daß ein weiterer Schritt (oder Monomer bei Polymeren) an den vorhandenen Walk hinten angehängt wird, während man gleichzeitig vorne einen Schritt entfernt, siehe Bild IV.1. Natürlich ist der neue Walk nicht statistisch unabhängig von dem alten, da nur einer von n Schritten verändert wurde. Würde man andererseits n solcher "Kriechschritte" hintereinander ausführen, dann hätte man *einen* statistisch völlig unabhängigen Walk erzeugt.

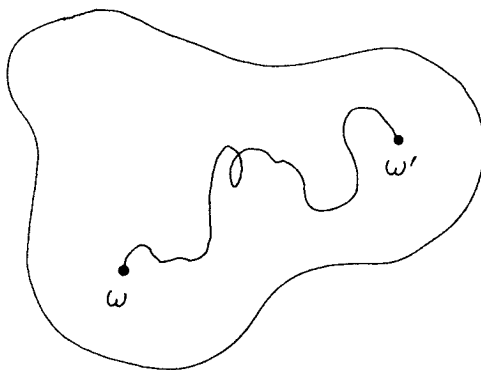


Bild IV.2: **Zustandsraum aller n -Schritt Random Walks.** Beim Kriechen des Walks wandert der Punkt, der den Zustand des Walks angibt, durch den Zustandsraum. Nach n Kriechschritten ist er an einem zufälligen, anderen (mit dem Ausgangszustand ω nicht mehr korrelierten) Punkt ω' im Zustandsraum angekommen.

Im Zustandsraum aller denkbaren n -Schritt Random Walks wäre man von einem Punkt zu einem völlig zufälligen, anderen Punkt übergegangen (siehe Bild IV.2). Da man jedoch auch alle Walks zwischen den beiden Punkten zur Mittelung benutzen kann, hat man bei gleichem Aufwand für das Erzeugen der Walks viel mehr Informationen über den Zustandsraum bekommen. Man würde ungenauere Mittelwerte erhalten, wenn man nur jeden n -ten Walk benutzen würde. Bei der Bestimmung des statistischen Fehlers muß man jedoch davon ausgehen, daß nur jeder n -te Walk unabhängig ist. Um Rechenzeit zu sparen, werden die Mittelwerte normalerweise nicht nach jedem Schritt berechnet. Da diese Berechnung jedoch vektorisierbar ist (siehe unten), im Gegensatz zum Programmteil, der das Kriechen durchführt, ist der Rechenzeitverlust in diesem Fall tragbar.

Würde man die neu angehängten Schritte jeweils mit derselben Wahrscheinlichkeit in alle Richtungen (es gibt $2d$ Richtungen in d Dimensionen) auswürfeln und würde man ferner den Walk nur vorwärts kriechen lassen, so würde man ungewichtete Random Walks erzeugen und somit den Zustandsraum im Limes unendlich langer Simulation gleichmäßig abtasten.

Hier soll jedoch jeder Walk mit einem statistischen Gewicht $e^{-\lambda s}$ während der Simulation auftauchen, wobei s die Anzahl der verschiedenen besuchten Plätze ist und $\lambda \geq 0$ ein Parameter ist. Für das folgende ist es nützlich, den Walk als Polymerkette oder als SIW zu betrachten, siehe Seite 13. Dann ist $e^{-\lambda s}$ ein Boltzmannfaktor, λ die inverse Temperatur und s die Energie der Kette.

Bei der Temperatur ∞ hat die Energie keinen wichtenden Einfluß auf die Kette. Der SIW ist ein gewöhnlicher Random Walk. Ist die Temperatur jedoch endlich, bevorzugt der SIW Zustände geringerer Energie s , d.h., er versucht möglichst wenige, verschiedene Plätze zu besuchen. Diese Tendenz ist um so größer, je kleiner die Temperatur ist. Ist die Temperatur schließlich 0 ($\lambda \rightarrow \infty$ bzw. $c \rightarrow 1$), geht der SIW in den Grundzustand, also in den Zustand mit der geringsten Energie, d.h., es ist $s = 2$, für $n \geq 1$. Die Frage, ob der Übergang von der ausgedehnten Kette für $\lambda = 0$ zu der kollabierten Kette für $\lambda = \infty$ durch einen Phasenübergang geschieht und wenn ja, bei welchem λ , wird weiter unten diskutiert werden.

Das MC-Verfahren soll durch Übergänge im Zustandsraum eine Markovkette [83, 86, 97] von Walks ω erzeugen, die mit einem statistischen Gewicht $P(\omega)$ auftreten. Dazu werden Übergangswahrscheinlichkeiten $w_{\omega \leftarrow \omega'}$ von einem Walk ω' zu einem Walk ω definiert. Der Algorithmus erzeugt nun anhand dieser Übergangswahrscheinlichkeiten ausgehend von einem Start-Walk eine Markovkette

aus Walks. Damit das entstehende Ensemble im Limes unendlich vieler Sprünge schließlich die gewünschte Wahrscheinlichkeitsverteilung $P(\omega)/Z$ (mit der Zustandssumme $Z := \sum_{\omega} P(\omega)$) hat, genügt es, wenn die Übergangswahrscheinlichkeiten vier Bedingungen erfüllen (siehe auch Referenzen [83–86, 98]):

MC1: Jeder Punkt des diskreten Zustandsraumes muß mit endlicher (> 0) Wahrscheinlichkeit erreichbar sein, d.h., der Algorithmus muß zu jedem Punkt des Zustandsraumes in endlich vielen Schritten gelangen können, wobei die Übergangswahrscheinlichkeiten für jeden Schritt größer als 0 sind.

MC2: Für alle ω und ω' muß gelten:

$$0 \leq w_{\omega \leftarrow \omega'} \leq 1. \quad (IV.10)$$

MC3: Für alle ω' muß gelten:

$$\sum_{\omega} w_{\omega \leftarrow \omega'} = 1. \quad (IV.11)$$

Es ist also sicher, daß ein Übergang irgendwohin ausgeführt wird. Das schließt natürlich auch einen Übergang auf den Zustand selbst mit ein ($w_{\omega \leftarrow \omega} > 0$), d.h., der Zustand bleibt erhalten. Zusammen mit Punkt MC2 bedeutet das, daß es sich bei den Übergangswahrscheinlichkeiten um wohldefinierte Wahrscheinlichkeiten handelt.

MC4: Für alle ω muß gelten:

$$\sum_{\omega'} w_{\omega \leftarrow \omega'} P(\omega') = P(\omega). \quad (IV.12)$$

Die Wahrscheinlichkeitsverteilung muß also stationär sein.

Die letzte Bedingung MC4 wird meist durch die hinreichende Bedingung des detaillierten Gleichgewichts MC4' ersetzt:

MC4': Für alle ω und ω' muß gelten:

$$w_{\omega \leftarrow \omega'} P(\omega') = w_{\omega' \leftarrow \omega} P(\omega). \quad (IV.13)$$

Man zeigt durch Einsetzen sofort, daß Gl. (IV.12) aus Gl. (IV.13) und Gl. (IV.11) folgt. Die meisten MC-Verfahren basieren auf der einfacheren Bedingung Gl. (IV.13). Für das aktuelle Problem des SIW's wurde zunächst auch diese Gleichung verwendet. Es wird sich jedoch später zeigen, daß man mit einem abgewandelten Verfahren, das auf der Stationaritätsbedingung Gl. (IV.12) beruht (kurz *stationäres MC-Verfahren* genannt), eine deutliche, qualitative Verbesserung der Methode bekommt. Der Einfachheit halber

wird jedoch zunächst das Verfahren, das im folgenden das Verfahren mit *detailliertem Gleichgewicht* genannt wird, beschrieben.

In unserem Fall ist das statistische Gewicht $P(\omega) := e^{-\lambda s_\omega} = p^{s_\omega}$, wobei s_ω die Anzahl der verschiedenen Plätze ist, die der Walk ω besucht. Gewichtete Mittelwerte über beliebige Random Walk abhängige Größen $f(\omega)$

$$\langle f(\omega) \rangle := \frac{\sum_{\omega} f(\omega) e^{-\lambda s_\omega}}{\sum_{\omega} e^{-\lambda s_\omega}} \quad (\text{IV.14})$$

können dann durch arithmetische Mittelwerte über das erzeugte Walk-Ensemble ersetzt werden. Es gilt

$$\langle f(\omega) \rangle \approx \frac{1}{T_{\text{MC}}} \sum_{t=1}^{T_{\text{MC}}} f(\omega_t), \quad (\text{IV.15})$$

wobei T_{MC} die Anzahl der Walks ω_t im Ensemble ist, die vom MC-Verfahren erzeugt wurden.

IV.3.1. MC-Verfahren mit detailliertem Gleichgewicht

IV.3.1.1. Beschreibung des Algorithmus

Nach Metropolis [98] (siehe auch Referenzen [83–85]) macht man die Übergänge im Zustandsraum von der Energieänderung des Übergangs abhängig. Man kann den Algorithmus ohne Schwierigkeiten auf das betrachtete Problem des SIW's übertragen. Der Algorithmus ist in Analogie zu der Simulation eines Isingmodells zu sehen. Dort werden Zustandsänderungen vorgenommen, indem einzelne Spins in Abhängigkeit von der dabei bewirkten Energieänderung gedreht werden. Nach einem solchen Schritt hat derselbe Spin die Chance, wieder zurückgedreht zu werden, wenn man die Spins zufällig auswählt. Die energieabhängigen Wahrscheinlichkeiten werden so gewählt, daß detailliertes Gleichgewicht gewährleistet ist. Der Algorithmus für die SIW's wird wie folgt festgelegt:

dG1: Man erzeugt einen geeigneten Start-Walk.

dG2: Man wählt eine beliebige Richtung, in die der Walk kriechen soll. Dabei kann der Walk sowohl vorwärts, wie rückwärts kriechen und zwar auf jeden nächsten Nachbarplatz, unabhängig davon, ob dieser Platz schon besucht ist oder nicht. Damit gibt es insgesamt $4d$ Kriechmöglichkeiten, die alle zunächst gleich wahrscheinlich sind.

dG3: Für die gewählte Richtung wird berechnet, wie groß die Energieänderung Δs ist. Hier kommen für Δs nur die Werte $+1$, -1 und 0 in Frage, je nachdem, ob

der neu entstandene Walk einen Platz mehr, einen Platz weniger oder genau so viele Plätze besucht wie der alte Walk.

dG4: Falls $\Delta s \leq 0$, wird der neue Walk angenommen, d. h., der Kriechschritt wird tatsächlich ausgeführt. Falls $\Delta s = +1$, falls also ein Platz mehr besucht werden würde als vorher, wird der Kriechschritt nur mit der Wahrscheinlichkeit $e^{-\lambda \Delta s} = p \leq 1$ ausgeführt. Andernfalls wird der neue Walk verworfen.

dG5: Der so ermittelte Walk (sei es der neue, angenommene oder der alte, falls der neue verworfen wurde) wird zum Ensemble zur Mittelung aller interessierenden Größen hinzugenommen. Falls also der Kriechschritt verweigert wurde, wird der alte Walk erneut in die Mittelung mit einbezogen.

dG6: Die Schritte dG2–dG5 werden so lange wiederholt, bis genügend Daten gesammelt sind, um statistisch genaue Aussagen machen zu können. Da sich zuerst ein “thermisches” Gleichgewicht einstellen muß, wird zu Beginn der Simulation eine genügend große Zahl von Walks nicht zur Mittelung herangezogen.

IV.3.1.2. Beweis der Konvergenz gegen das gewünschte Ensemble

Es ist offensichtlich, daß dieser Algorithmus die Bedingungen MC1–MC3 auf Seite 48 erfüllt, denn man kann mit n Schritten in dieselbe Richtung, die alle eine positive Wahrscheinlichkeit haben, jeden n -Schritt Random Walk erzeugen. Ferner sind die Übergangswahrscheinlichkeiten wohldefiniert. Die Bedingung des detaillierten Gleichgewichts MC4' kann man auch leicht nachprüfen, indem man die verschiedenen Fälle $\Delta s > 0$ und $\Delta s \leq 0$ einzeln diskutiert. Dies geht völlig analog zu entsprechenden Rechnungen zum Isingmodell [83, 85].

IV.3.1.3. Korrelationszeiten

Der Walk wird durch obigen Algorithmus dauernd hin- und herkriechen. Der mittlere Teil der Kette wird dabei recht lange unverändert bleiben, da ja nur an den beiden Enden Veränderungen vorgenommen werden. Im folgenden wird die Zeit, d. h. die Anzahl der MC-Schritte, nach der auch das letzte Teilstück der Kette einmal neu ausgewürfelt worden ist, als Korrelationszeit τ bezeichnet.

Markiert man sich ein Ende des Random Walks und beobachtet, wie es vor- und zurückkriecht, dann vollführt es einen eindimensionalen Random Walk, wenn man die effektive Anzahl der Kriechschritte in Vorwärtsrichtung gegen die MC-Schritte aufträgt, die zu einem Übergang geführt haben, Bild IV.3. Es ist genau dann eine völlig neue Kette erzeugt worden, wenn die Anzahl der verschiedenen Plätze, die dieser eindimensionale Random Walk besucht hat, gerade n , die Kettenlänge, ist. Die Wahrscheinlichkeit,

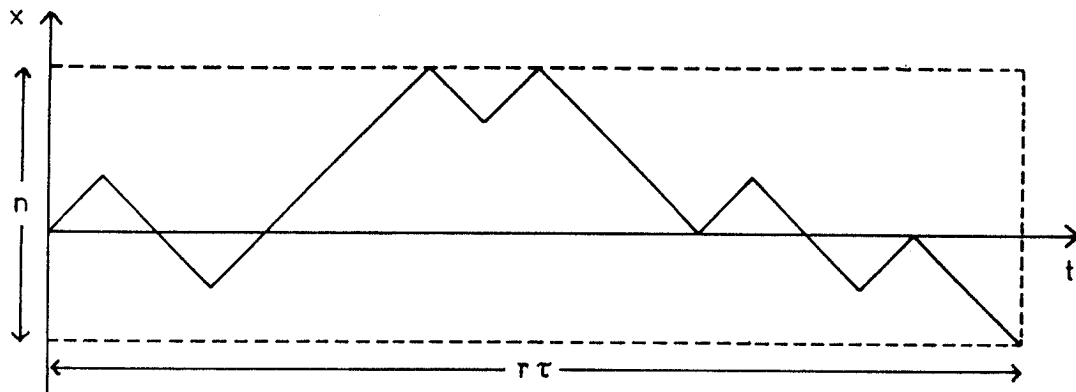


Bild IV.3: Effektive Zahl der Vorwärtsschritte beim detaillierten Gleichgewicht. Typische Realisierung als Funktion der akzeptierten MC-Schritte.

daß das nach τ erfolgreichen Schritten der Fall ist, kann exakt berechnet werden (siehe z. B. Ref. 97), doch ist klar, daß τ wegen der Diffusionseigenschaft von Random Walks (mittlere quadratische Auslenkung ist proportional zur Schrittzahl) wie n^2 geht. Nimmt man ferner an, daß ein vorgeschlagener Kriechschritt mit einer konstanten Wahrscheinlichkeit r akzeptiert wird, so ist τ auch gemessen an der echten MC-Zeit proportional zu n^2 (und natürlich umgekehrt proportional zu r). Diese Annahmewahrscheinlichkeit r (engl.: acceptance rate) ist um so kleiner, je größer der Parameter λ , d. h., je größer die Haftstellenkonzentration c ist. Die Korrelationszeit τ ist der begrenzende Faktor dieses MC-Simulationsverfahrens, denn erst nach der Zeit τ kann man davon ausgehen, daß man einen statistisch unabhängigen Walk vorliegen hat. Die gesamte Simulationszeit muß ein Vielfaches von τ betragen, damit man Mittelwerte mit kleinen statistischen Schwankungen berechnen kann. Da die Zeit aber proportional zu n^2 ist, geht die Rechenzeit quadratisch mit der Schrittzahl des SIW. Diese Eigenschaft gilt auch für das echte Reptationsmodell für Polymerketten [99]. Bei vernünftigen Rechenzeiten (von der Größenordnung einer Stunde etwa auf einer CRAY X-MP), kann man mit dieser Methode nicht wesentlich weiter als in die Bereiche vorstoßen, die auch noch mit der Scanning-Methode [38] behandelt werden können.

IV.3.2. MC-Verfahren mit stationärem Gleichgewicht

Der entscheidende Schritt liegt nun in der Möglichkeit, den Algorithmus so abzuwandeln, daß qualitativ kürzere Korrelationszeiten die Folge sind. Dazu wird auf die Bedingung des detaillierten Gleichgewichts verzichtet.

IV.3.2.1. Beschreibung des Algorithmus

Der Algorithmus wird so modifiziert, daß die Kette die Tendenz hat, immer in dieselbe Richtung zu kriechen. Das ist der Fall, solange die neu erzeugten Walks akzeptiert werden. Der SIW vollführt damit in gewisser Weise einen korrelierten Random Walk (siehe auch Teil III der Arbeit). Erst wenn der Übergang aus energetischen Gründen verweigert wird, kehrt der Walk seine Kriechrichtung um. Da der Walk nun nicht jeden Kriechschritt im nächsten Schritt wieder rückgängig machen kann, ist die Bedingung des detaillierten Gleichgewichts nicht mehr erfüllt. Im folgenden wird aber gezeigt, daß die Stationaritätsbedingung Gl. (IV.12) weiterhin erfüllt ist.

Jeder simulierte Walk besitzt nun eine Orientierung, die man dadurch kennzeichnen kann, daß der Walk einen "Kopf" und einen "Schwanz" hat. Vorwärtskriechen bedeutet nun, daß der Kopf sich einen neuen Platz sucht und der Schwanz hinterhergezogen wird.

Der Algorithmus für das stationäre Gleichgewicht (sG) entspricht im wesentlichen dem Algorithmus für das detaillierte Gleichgewicht (dG). Es werden lediglich die Schritte dG2 und dG4 durch die folgenden Schritte sG2 und sG4 ersetzt.

sG2: Man wählt eine beliebige Richtung, in der der Walk kriechen soll. Dabei kriecht der Walk immer vorwärts, womit er nur noch $2d$ Möglichkeiten hat. Die Kriechrichtung (vorwärts oder rückwärts) wird solange beibehalten, bis sie im Schritt sG4 geändert wird.

sG4: Falls $\Delta s \leq 0$, wird der neue Walk angenommen, d. h., der Kriechschritt wird tatsächlich ausgeführt und die Kriechrichtung für den nächsten Schritt wird beibehalten. Falls $\Delta s = +1$, falls also ein Platz mehr besucht werden würde als vorher, wird der Kriechschritt nur mit der Wahrscheinlichkeit $p \leq 1$ ausgeführt. Die Kriechrichtung bleibt auch dann erhalten. Andernfalls wird der neue Walk verworfen und der alte bleibt erhalten, jedoch wird die Kriechrichtung für den nächsten Schritt umgekehrt, d. h., Kopf und Schwanz tauschen ihre Identität.

Es muß nun gezeigt werden, daß sich das richtige Ensemble bei diesem Verfahren einstellt.

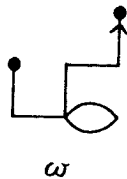
IV.3.2.2. Beweis der Konvergenz gegen das gewünschte Ensemble

Trivialerweise sind hier ebenfalls die Bedingungen MC1–MC3 auf Seite 48 erfüllt und es bleibt zu zeigen, daß auch die Stationaritätsbedingung MC4 gilt.

Sei also zu einem Zeitpunkt in der Simulation ein Walk ω erzeugt worden. Betrachte alle Walks ω' , aus denen ω durch einen MC-Schritt entstanden sein kann. Es

ist klar, daß sich die Anzahl der besuchten Plätze s höchstens um 1 verändert haben kann. Sei ferner $\tilde{\omega}$ der Umkehrwalk zu ω , d. h. der Walk, der aus ω entsteht, indem man "Kopf" und "Schwanz" vertauscht. Der Walk $\tilde{\omega}$ sieht also genau so aus wie ω , nur die Kriechrichtung ist umgekehrt. Es ist nun zweckmäßig, zwei Fälle zu unterscheiden, je nachdem, ob der betrachtete Walk ω mit seinem Kopf, also seinem n -ten Schritt oder Monomer, einen neuen Platz besucht oder nicht.

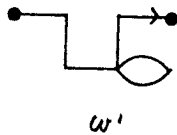
1. Fall: Der Platz, auf dem sich der Kopf befindet, wird nur vom Kopf allein besucht.



Daraus folgt, daß der Umkehrwalk $\tilde{\omega}$ bei einem Schritt die Anzahl der besuchten Plätze s nicht vergrößern kann. (Denn dann müßte der Schwanz von $\tilde{\omega}$ — der Kopf von ω — beim Kriechen seinen Platz verlassen. Da der Kopf von $\tilde{\omega}$ höchstens einen neuen Platz besuchen kann, kann s nicht größer werden.) Der Umkehrwalk würde also auf jeden Fall weiter vorwärts kriechen, da die "Energie" s nicht zunähme. Somit gibt es in diesem Fall keinen Beitrag zu ω durch Walkumkehr. Alle zu betrachtenden Walks ω' haben also dieselbe Kriechrichtung wie ω .

a) " $s \rightarrow s$ "

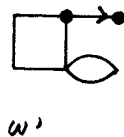
Sei $s := s_\omega = s_{\omega'}$. Dann gilt



$$P(\omega') w_{\omega \leftarrow \omega'} = p^s \frac{1}{2d}. \quad (IV.16)$$

b) " $s - 1 \rightarrow s$ "

Sei $s := s_\omega = s_{\omega'} + 1$. Dann gilt, da der Übergang von ω' nach ω nur mit der Wahrscheinlichkeit $p/2d$ ausgeführt wird,



$$P(\omega') w_{\omega \leftarrow \omega'} = p^{s-1} \frac{p}{2d} = p^s \frac{1}{2d}. \quad (IV.17)$$

Damit gilt für die Summe über alle $2d$ möglichen Walks ω'

$$\sum_{\omega'} P(\omega') w_{\omega \leftarrow \omega'} = p^s = P(\omega), \quad (IV.18)$$

womit die Stationaritätsbedingung Gl. (IV.12) für diesen Fall gezeigt ist.

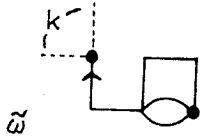
2. Fall: Der Platz, auf dem sich der Kopf befindet, wird auch von einem anderen Teil des Walks besucht.



Der Umkehrwalk $\tilde{\omega}$ kann nun durch Kriechen die Anzahl der besuchten Plätze um 1 erhöhen. Sei k die Anzahl der Richtungen, in die $\tilde{\omega}$ kriechen kann, um s zu vergrößern. In diesem Fall gibt es Beiträge zu $P(\omega)$ durch Walkumkehr.

Andererseits kann nun $\tilde{\omega}$ die Anzahl der besuchten Plätze nicht verkleinern, da der Schwanz von $\tilde{\omega}$ beim Kriechen einen Platz verläßt, der noch von anderen Teilen des Walks besucht wird.

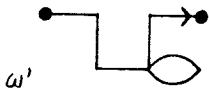
a) "Walkumkehr"



In k Richtungen würde der Umkehrwalk $\tilde{\omega}$ die Anzahl der besuchten Plätze s um 1 vergrößern. Mit der Wahrscheinlichkeit $1 - p$ wird ein solcher Walk verweigert und durch Walkumkehr in den Walk ω umgewandelt. In diesen Fällen gilt

$$P(\omega')_{\omega \leftarrow \omega'} = p^s \frac{1-p}{2d} \quad (k\text{-mal}). \quad (IV.19)$$

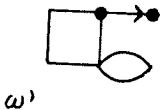
b) " $s + 1 \rightarrow s$ "



Sei $s := s_\omega = s_{\omega'} - 1$. Das ist ebenfalls nur in k Fällen möglich, nämlich genau dann, wenn der Umkehrwalk $\tilde{\omega}$ durch Kriechen einen Platz mehr besuchen könnte. Dann gilt

$$P(\omega')_{\omega \leftarrow \omega'} = p^{s+1} \frac{1}{2d} \quad (k\text{-mal}). \quad (IV.20)$$

c) $s \rightarrow s$



Sei $s := s_\omega = s_{\omega'}$. Das ist in den restlichen $2d - k$ Richtungen der Fall, denn dann hat der Walk ω' genauso viele Plätze besucht, wie ω . Es gilt

$$P(\omega')_{\omega \leftarrow \omega'} = p^s \frac{1}{2d} \quad (2d - k\text{-mal}). \quad (IV.21)$$

Die Unterfälle a), b) und c) zusammen ergeben

$$\sum_{\omega'} P(\omega') w_{\omega \leftarrow \omega'} = \frac{1}{2d} [p^s(k + 2d - k) + p^{s+1}(-k + k)] = p^s = P(\omega).$$

Auch hier ist also die Stationaritätsbedingung Gl. (IV.12) erfüllt.

In allen Fällen gilt somit

$$\sum_{\omega'} P(\omega') w_{\omega \leftarrow \omega'} = P(\omega),$$

womit gezeigt ist, daß dieses Verfahren, das *stationäre Verfahren*, ebenfalls gegen das richtige Ensemble $\{\omega\}$ mit $P(\omega) = p^s$ konvergiert.

IV.3.2.3. Korrelationszeiten

Der wesentliche Vorteil dieser neuen Methode liegt in den deutlich verkürzten Korrelationszeiten. Würde nämlich der Walk die ganze Zeit nur in eine Richtung kriechen, hätte man nach n Schritten einen neuen, statistisch unabhängigen Walk und die Korrelationszeit τ wäre n und ginge nicht wie n^2 wie beim detaillierten Gleichgewicht. Das tatsächliche Verhalten liegt natürlich irgendwo dazwischen. Wenn man die weiter oben (siehe Seite 50) gemachten Überlegungen zu den Korrelationszeiten für dieses Verfahren wiederholt, sieht man, daß die Kriechrichtung nun einen korrelierten Random Walk macht, siehe auch Teil I der Arbeit, d. h., sie hat die Tendenz, ihre einmal eingeschlagene Richtung beizubehalten. Nimmt man wieder an, daß die Wahrscheinlichkeit, die Kriechrichtung umzukehren, zeitlich konstant ist, zeigt eine einfache Skalenanalyse, daß die Korrelationszeit τ immer noch wie n^2 gehen sollte, da beim korrelierten Random Walk sowohl im Ausdruck für die Anzahl der besuchten Plätze, wie auch in dem für die mittlere quadratische Auslenkung nach Gl. (III.2) bzw. Gl. (III.3) im wesentlichen die Schrittzahl n durch fn zu ersetzen ist, wobei f der Korrelationsfaktor ist [2, 66, 67, 100]. Wegen der Tendenz, länger in die gleiche Richtung zu laufen, sind die Korrelationszeiten bei der neuen Methode deutlich kleiner, als bei der Methode des detaillierten Gleichgewichts.

Obige Betrachtungen legen nahe, daß die n -Abhängigkeit der Korrelationszeit gegenüber der anderen Methode gleich bleibt, daß also nur der Vorfaktor abnimmt. Wie sich bei der Durchführung der Simulationen aber gezeigt hat, ändert sich auch die n -Abhängigkeit, so daß τ nur noch wie n^α bei konstanter Haftstellenkonzentration c geht, mit $\alpha \approx 1,5$ für $d = 2$ und $\alpha \approx 1,6$ für $d = 3$. Die Ursache ist darin zu suchen, daß bei steigender Schrittzahl n (also bei wachsender Kettenlänge) der Walk mehr und mehr dazu gezwungen wird, immer wieder über sich selbst hinwegzukriechen und eine immer kompakter werdende Form anzunehmen. Das bedeutet, daß der Kopf des Walks

eine immer höher werdende Wahrscheinlichkeit hat, sich im Innern des Knäuels zu befinden. Dort kann er aber beim nächsten Schritt die Anzahl der besuchten Plätze nicht vergrößern und kann deshalb auch seine Richtung nicht umkehren. Damit steigt bei dem hier betrachteten stationären Verfahren der Korrelationsfaktor mit wachsendem n an und ist nicht etwa unabhängig von n . Nach Gl. (III.2) wird eine gegebene Strecke damit noch schneller durchlaufen und der Effekt führt insgesamt zu einer schwächeren n -Abhängigkeit der Korrelationszeit, als das einfache Verfahren mit detailliertem Gleichgewicht.

Damit kann man in den Simulationen zu wesentlich höheren n -Werten gehen, so daß es diese Methode erlaubt, viel weiter in den asymptotischen Bereich (große n und große c) vorzudringen, als alle anderen, oben erwähnten Verfahren.

IV.3.3. Überlebenswahrscheinlichkeit und Entropiebestimmung

Da im Metropolis-Algorithmus nur Verhältnisse von Übergangsraten eingehen, die den entsprechenden Verhältnissen von statistischen Gewichten entsprechen müssen, geht die Information über das gesamte statistische Gewicht verloren. Das heißt, daß es zunächst nicht möglich ist, die Zustandssumme direkt zu bestimmen. Zwar ist Z immer noch wegen Gl. (II.24) mit $f(\omega) = \exp(+\lambda s_\omega)$ als Ensemblemittelwert zu schreiben,

$$Z = \frac{(2d)^n}{\langle e^{+\lambda s} \rangle_{\text{SIW}}}, \quad (\text{IV.22})$$

— diese Methode ist auch in der Literatur [101] vorgeschlagen worden — doch stehen wir hier vor dem gleichen Problem, das auch schon die direkte Simulation des Haftstellen-Problems aufwirft. Man kann nämlich zeigen, daß die Hauptbeiträge zum Mittelwert in obiger Formel von s -Werten herrühren, die mit der Methode des “importance sampling” nur sehr selten erreicht werden. Die Argumentation geht analog zu den Überlegungen im Abschnitt IV.1.1, nur wird hier mit einer steigenden Exponentialfunktion $\exp(+\lambda s)$ gewichtet.

Ohne die Zustandssumme wäre aber auch die Überlebenswahrscheinlichkeit des Random Walks, die Entropie und die freie Energie des SIW nicht bestimmbar.

Einen Ausweg bieten die ersten Ableitungen der Zustandssumme, z. B. die Energie des SIW, Gl. (II.26),

$$E = \langle s \rangle_{\text{SIW}} = \frac{1}{Z} \sum_{\omega} s_{\omega} e^{-\lambda s_{\omega}} = -\frac{\partial \ln Z}{\partial \lambda}, \quad (\text{IV.23})$$

woraus sofort

$$-\ln Z = \int_0^{\lambda} \langle s \rangle_{\text{SIW}} d\lambda - n \ln(2d) \quad (\text{IV.24})$$

folgt, da $Z(\lambda = 0) = \sum_{\omega} 1 = (2d)^n$ gilt. Die Überlebenswahrscheinlichkeit bei Anwesenheit von Haftstellen ergibt sich wegen Gl. (II.25) zu

$$-\ln \Psi_n = \int_0^\lambda \langle s \rangle_{\text{SIW}} d\lambda. \quad (\text{IV.25})$$

Es scheint zunächst ein großer Nachteil gegenüber der Scanning-Methode [38] zu sein, daß hier die Zustandssumme nur über eine Integration zu bestimmen ist, jedoch ist die numerische Integration nicht schwierig, da $\langle s \rangle_{\text{SIW}}$ eine glatte Funktion von λ ist und ohnehin für verschiedene λ -Werte bestimmt werden soll. Ferner liefert die Gl. (IV.25) direkt einen Ausdruck für den *Logarithmus* der Überlebenswahrscheinlichkeit, also für die Größe, die am meisten bei der Diskussion des asymptotischen Verhaltens interessiert. Insbesondere ist das Verhalten der Korrekturen interessant. Theoretische Aussagen über die Korrekturterme können aber wegen Gl. (IV.23) auch anhand der mittleren Zahl der besuchten Plätze $\langle s \rangle_{\text{SIW}}$ überprüft werden, wenn die Theorie die λ -Abhängigkeit mit einschließt.

IV.3.4. Vektorisierung durch Parallelbearbeitung vieler Walks

Der oben beschriebene Metropolis-Algorithmus ist seiner Natur nach nicht vektorisierbar, da an keiner Stelle viele gleichartige Operationen mit einem ganzen Vektor von Werten durchgeführt werden, ohne daß IF-Abfragen den Programmablauf verzweigen.

Eine Vektorisierung ist nur dadurch möglich, daß man eine große Anzahl von Walks gleichzeitig simuliert. Dann sind viele (nicht alle) Einzelschritte vektorisierbar. Statt nur eine einzelne Variable z. B. auf Null zu initialisieren, kann nun immer gleich ein ganzer Vektor initialisiert werden. Anstatt nur eine Zufallszahl für den nächsten Schritt zu erzeugen, kann nun ein ganzer Vektor von Zufallszahlen erzeugt werden, der für den nächsten Schritt aller simulierten Walks benutzt wird. Da Vektoroperationen auf der benutzten CRAY X-MP erheblich schneller als Skalaroperationen ablaufen, kann auf diese Weise eine beachtliche Geschwindigkeitssteigerung erzielt werden.

Möglich ist eine simultane Bearbeitung von vielen Walks aber nur dann, wenn der Speicherbedarf für jeden einzelnen Walk gering ist. Mit Hilfe der im nächsten Abschnitt beschriebenen Hash-Tabellen und der Verwendung eines Haldenspeichers (engl.: *heap*) gelingt genau das.

IV.3.5. Hash-Tabellen und Haldenspeicher

Die naheliegende Implementation der Simulation eines Random Walks benötigt Speicherplatz für verschiedene Größen, die den Walk charakterisieren. Den größten Speicherbedarf haben die folgenden Daten:

- **Richtungen der einzelnen Schritte**

Da der Walk über das Gitter kriecht, müssen Informationen über jeden einzelnen Schritt abgespeichert sein, z. B. in Form von Schrittrichtungen (als Zahlen von 0 bis $2d - 1$). Sonst ist es dem Programm nicht möglich, beim Kriechen "den Schwanz hinterherzuziehen".

- **Besuchszahlen aller Gitterplätze**

Es ist notwendig, die Anzahl der Besuche auf jedem einzelnen Gitterplatz abzuspeichern, da nur so festgestellt werden kann, ob sich die Gesamtzahl der besuchten Plätze verändert oder nicht. Diese Information steckt zwar auch in der gespeicherten Liste der Richtungen der Einzelschritte, ist aber nur mit erheblichem Aufwand aus dieser zu bekommen, nämlich indem man die Richtungen der Reihe nach durchgeht, und den jeweils erreichten Gitterplatz als "besucht" markiert, und alle dabei neu besuchten Plätze mitzählt. Auch bei dieser Methode müßte man Speicher für jeden erreichbaren Gitterplatz reservieren, aber nur einmal für alle simulierten Walks. Jedoch verbietet sich diese Methode aus Gründen der Rechenzeit, da die Richtungsliste sehr lang werden kann (bei den durchgeführten Simulationen bis zu 3200 Schritte).

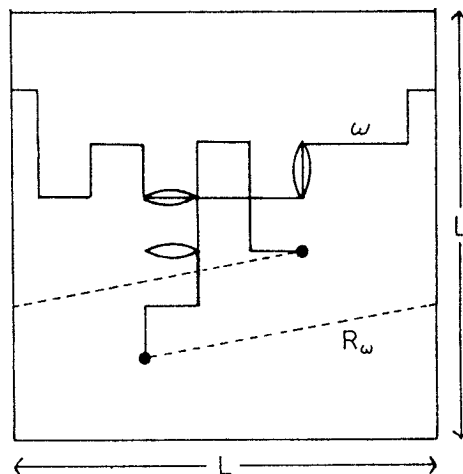


Bild IV.4: Zu kleine Gitter mit periodischen Randbedingungen. Die Anzahl der besuchten Plätze erscheint zu klein, wenn die Kantenlänge des Gitters nicht ausreichend groß gewählt wird.

Der Speicherbedarf für das Gitter stellt dabei das größte Problem dar. Selbst unter Zuhilfenahme von periodischen Randbedingungen müßte das gespeicherte Gitter bei einer Walklänge von n Schritten eine Kantenlänge von mindestens $n + 1$ haben, wenn man keine zusätzlichen Effekte durch Walks haben möchte, die über den Rand hinweg wieder in sich selbst hineinlaufen, siehe Bild IV.4. In höheren Dimensionen ($d > 1$) ist dadurch der Speicheraufwand erheblich und es wäre unmöglich, gleichzeitig viele Walks zu simulieren. Der in Bild IV.4 beschriebene Effekt kann immer noch vernachlässigt werden, wenn man die Kantenlänge verringert. Die effektive Anzahl der Schritte, die der Walk in eine gegebene Raumrichtung ausführt, ist in d Dimensionen im Mittel n/d . Die mittlere quadratische Auslenkung ist dann auch n/d , und damit das Teilchen das Gitter in eine gegebene Richtung nicht über die periodischen Randbedingungen verläßt, sollte die Kantenlänge mindestens $\sqrt{n/d}$ sein. Dadurch ist der Speicherbedarf bei der Simulation von N verschiedenen n -Schritt Random Walks von der Ordnung

$$\left(\frac{n}{d}\right)^{d/2} N, \quad (IV.26)$$

wächst also wie $n^{d/2}$. Im Gegensatz dazu ist aber die tatsächlich benötigte Anzahl von Speicherplätzen nur proportional zu

$$n^{\frac{d}{d+2}} N,$$

da jeder Walk im Mittel nur von der Ordnung $n^{d/(d+2)}$ verschiedene Plätze besucht, siehe Gl. (II.19). Die tatsächlich benötigte Speichermenge wächst also langsamer als n . Damit könnte sehr viel Speicherplatz gespart werden, wenn man statt eines ganzen Gitters, nur eine Liste aller tatsächlich besuchten Plätze abspeichern würde, die neben der Anzahl der Besuche auf diesem Platz auch eine Information über die Koordinaten des Gitterplatzes enthält. Um festzustellen wie oft ein gegebener Platz schon besucht wurde, müßte diese Liste nach dem Gitterplatz durchsucht werden. Da die Liste aber von der Ordnung $n^{d/(d+2)}$ viele Einträge hätte, würde auch dieses Verfahren zu viel Rechenzeit erfordern.

Einen Ausweg bietet die Anwendung von Hash-Tabellen [102–104]. Die Grundidee dabei ist, mit Hilfe einer sogenannten Hash-Funktion einem gegebenen Gitterpunkt eine Stelle in der Tabelle zuzuordnen, an die die Information potentiell gespeichert werden soll. Damit kann man die Stelle in der Tabelle berechnen, statt sie zu suchen. Da die Tabelle aber wesentlich kleiner ist als die Anzahl der möglichen Gitterplätze, ist die Zuordnung nicht umkehrbar, d. h., mehreren Gitterplätzen wird derselbe Platz in der Tabelle zugewiesen. Deshalb liefert die Hash-Funktion nur einen *potentiellen* Tabellenplatz, und es kann passieren, daß zwei oder mehr Gitterplätze abzuspeichern sind, die

äquivalent sind und an dieselbe Stelle der Hashtabelle gehören. Dann kommt es zu einer sogenannten Kollision. Es gibt in der Literatur [103, 104] eine Reihe von Möglichkeiten, diese Kollisionen aufzulösen, indem man z. B. im einfachsten Fall den nächsten freien Platz in der Tabelle zur Speicherung benutzt. Ein Wiederauffinden der gespeicherten Information geschieht dann dadurch, daß man mit der Hash-Funktion den Tabellenplatz berechnet und dort mit der Suche nach der gespeicherten Information beginnt. Da zusätzlich zu den gesuchten Daten auch noch der Gitterplatz selbst abgespeichert ist, ist es möglich, die gesuchte Information zu erkennen. Trifft man dann auf den gewünschten Tabelleneintrag, ist die Information gefunden. Trifft man hingegen auf einen leeren, ungenutzten Tabellenplatz, ist die Suche auch beendet, denn dann ist klar, daß die gewünschte Information nicht in der Hash-Tabelle gespeichert ist.

Es gibt weit kompliziertere und bessere Methoden der Kollisionsauflösung, die im Mittel weniger Tabellenzugriffe benötigen, doch sind all diese Methoden nur gut geeignet, wenn die Hash-Tabelle ausschließlich aufgefüllt wird. Keine der fortgeschritteneren Methoden hingegen erlaubt ein effizientes Entfernen von Tabelleneinträgen. Bei obiger, einfacher Methode z. B. würde das Entfernen eines Eintrages zu einem "Loch" in der Tabelle führen, das das Wiederfinden der Informationen unmöglich macht, da das Kriterium für das erfolglose Suchen nicht mehr gegeben wäre. Solche Löcher müßten beim Entfernen von Einträgen gleich wieder gestopft werden (durch Umspeichern anderer Einträge, die wegen eines Hash-Konfliktes nicht an ihrem bevorzugten Platz stehen). Doch alle fortgeschrittenen Hash-Techniken erlauben kein effizientes Umspeichern, wenn sehr viele Löcher zu stopfen sind. Bei der Simulation der SIW's, bei der die Ketten über das Gitter kriechen, werden aber im Mittel genau so viele Plätze neu besucht (→ Neueinträge in Hashtabelle), wie Plätze vom Walk verlassen werden (→ Entfernen eines Tabelleneintrages). Das Austragen aus der Hash-Tabelle sollte also genau so effektiv zu bewerkstelligen sein, wie das Neueintragen.

Es bleibt dafür nur das Verfahren der "Kollisionsauflösung durch verkettete Listen" übrig (siehe auch das Kapitel: *Collision resolution by "chaining"* in Ref. 103). Dabei wird die Kollisionsauflösung nicht in der Hash-Tabelle selbst vorgenommen. Statt dessen steht in der Hash-Tabelle jeweils nur ein Zeiger auf den Anfang einer verketteten Liste (engl.: *linked list*), in der alle besuchten Gitterplätze abgespeichert sind, die denselben Wert in der Hash-Funktion haben, also Kollisionen bewirken, siehe Bild IV.5. Die Hash-Funktion hat nun die Aufgabe, eine dieser Listen auszuwählen, in der die gesuchte Information stehen muß, falls sie gespeichert ist. Macht man die Hash-Tabelle groß genug, sind diese Listen sehr klein und die gewünschte Information wird sehr schnell gefunden. Wählt man z. B. die Hash-Tabelle etwa so groß wie die Anzahl der besuch-

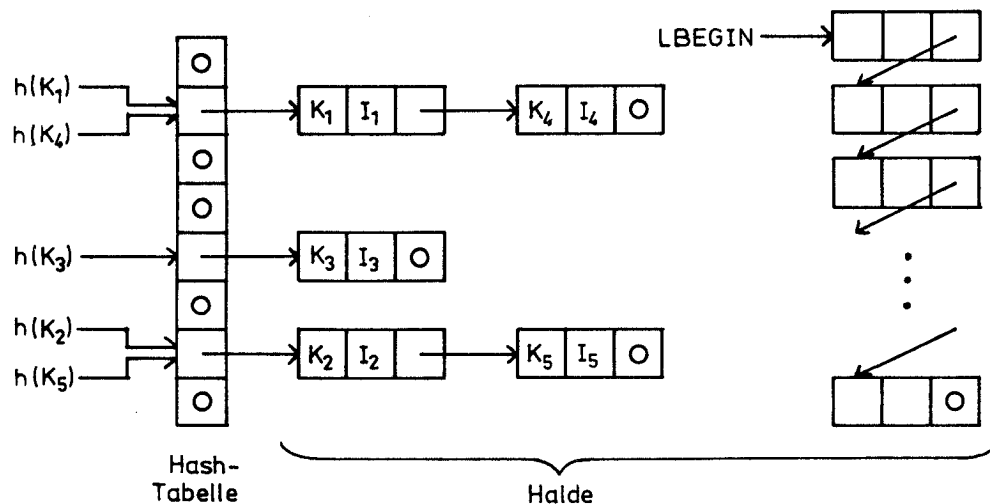


Bild IV.5: Hashtabelle und Halde. K_i : Schlüssel; I_i : unter dem Schlüssel K_i zu speichernde Information; $h(..)$: Hashfunktion; o : nil-Pointer

ten Plätze, haben die Listen im Mittel nur noch einen Eintrag und die Informationen werden praktisch sofort gefunden.

Die verketteten Listen selbst stehen nun in einem Teil des Speichers, der sogenannten Halde (engl.: *heap*), der Platz für die Listen aller zu simulierenden Walks bietet. Für die Halde muß also nur etwas mehr Speicher reserviert werden, als für die mittlere Zahl der besuchten Plätze aller Walks zusammen benötigt wird. Durch die große Anzahl der simulierten Walks gleichen sich Schwankungen im Speicherbedarf weitgehend aus, und es muß nicht für jeden einzelnen Walk der ungünstigste Fall vorgesehen werden.

Ein Eintrag in einer verketteten Liste — ein sogenannter Knoten — besteht jeweils aus den gespeicherten Informationen und zusätzlich aus einem Zeiger, der auf den nächsten Knoten in der Liste zeigt. Der letzte Knoten schließlich enthält einen besonderen Zeiger (engl.: *nil-pointer*), der angibt, daß dies das Ende der Liste ist. Derselbe besondere Zeiger wird auch benutzt, um in der Hash-Tabelle anzugeben, daß dort keine verkettete Liste beginnt. Neueinträge in die Liste werden durch Umsetzen von Zeigern auf noch nicht benutzte Knoten in der Halde vorgenommen. Diese werden am Anfang des Programms in einer großen Liste verkettet, auf die der Zeiger LBEGIN zeigt. Austragungen werden ebenfalls durch Umsetzen von Zeigern durchgeführt, indem der nicht mehr benötigte Knoten wieder der Liste der leeren Knoten hinzugefügt wird. Die Liste, aus der der Knoten entfernt wurde, wird dann durch Umsetzen von Zeigern wieder korrekt verkettet.

IV.3.5.1. Hash-Funktion für $d = 2$

Die Wahl einer geeigneten Hash-Funktion ist entscheidend für den Erfolg der Methode. Die Funktion sollte die Gitterpunkte so auf die Hash-Tabelle abbilden, daß eine optimale Kollisionsvermeidung erreicht wird. Der Programmteil, der Kollisionen auflöst, ist nicht vektorisierbar. Deshalb kann etwas mehr Aufwand bei der Hash-Funktion betrieben werden, um Kollisionen weitestgehend zu vermeiden, denn die Hash-Funktion selbst kann in vektorisierter Form berechnet werden.

Die Wahrscheinlichkeit, daß der Random Walk zwei verschiedene Plätze besucht, die auf dieselbe Zeile in der Hash-Tabelle abgebildet werden, sollte also möglichst klein sein. Das ist dann der Fall, wenn der Abstand zwischen zwei solchen äquivalenten Plätzen möglichst groß ist. Umgekehrt formuliert ist also das Problem dadurch gegeben, bei vorgegebenem minimalen Abstand eine Unterteilung des Gitters in "Elementarzellen" mit H Plätzen zu finden, bei der H , die Länge der Hash-Tabelle, möglichst klein ist. Hierbei ist mit Abstand die minimale Anzahl von Schritten gemeint, die ein Teilchen benötigt, um von dem einen Punkt zu dem anderen zu kommen.

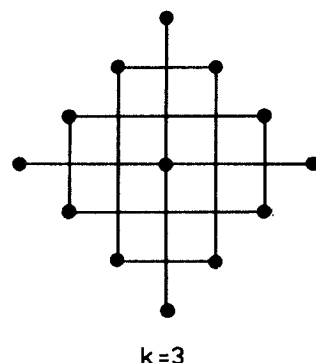


Bild IV.6: "Kugel" in $d = 2$ mit Radius k .

Ideal wären Kugeln bezüglich dieser Metrik, Bild IV.6, da dann gewährleistet ist, daß in jeder Richtung dieselbe Zahl von Schritten benötigt wird. Es muß aber das ganze Gitter durch die Elementarzellen abgedeckt werden können.

In $d = 2$ werden Kugeln durch auf der Spitze stehende Quadrate — im folgenden kurz Rhomben genannt — repräsentiert, siehe Bild IV.7, denn alle Punkte, die durch einen Random Walk z. B. in k Schritten erreicht werden können, liegen auf dem Rand eines Rhombus. Da man mit Hilfe von Rhomben die Ebene lückenlos abdecken kann, kann man solche Rhomben als Elementarzellen benutzen. Dann liegen äquivalente Punkte genau $2k$ Plätze voneinander entfernt. Der Rhombus enthält $2k^2$ Gitterplätze, ein Quadrat wie in Bild IV.7a hätte bei gleichem minimalen Abstand von $2k$ ein Volumen von

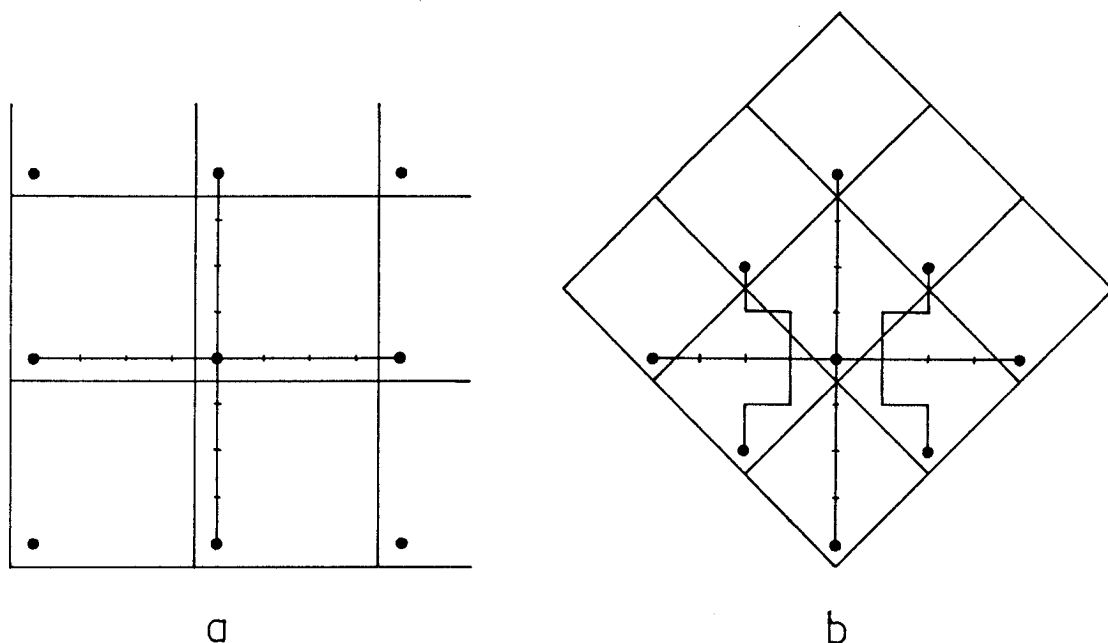


Bild IV.7: Aufteilung des Gitters in Elementarzellen für $d = 2$. Das zwei-dimensionale Gitter wird in "Elementarzellen" aufgeteilt, die das Gitter ganz abdecken. Die Abstände zu äquivalenten Punkten sind eingetragen.

- a) Abdeckung mit quadratischen Elementarzellen.
- b) Abdeckung mit "rhombischen" Elementarzellen.

$4k^2$, die Hash-Tabelle hätte also die doppelte Länge, bei gleicher Wahrscheinlichkeit von Kollisionen.

Die Gitterpunkte einer Elementarzelle werden jeweils auf die gesamte Hash-Tabelle verteilt, und zwar in allen Zellen auf dieselbe Art und Weise. Wie die Zuordnung innerhalb einer Zelle geschieht, ist völlig willkürlich. Man wählt sie so, daß die Berechnungen möglichst schnell vom Computer durchzuführen sind.

In Bild IV.8 ist eine Nummerierung der Gitterplätze zu sehen, die mit modulo-Operationen zu berechnen ist. Es gilt

$$\begin{aligned} u &= x + y \bmod 2k \\ v &= y - x \bmod 2k \\ h &= kv + \left\lfloor \frac{u}{2} \right\rfloor \end{aligned} \tag{IV.27}$$

Dabei bedeutet $\lfloor \dots \rfloor$ die Gaußklammer, also den ganzzahligen Anteil bei einer nicht negativen, reellen Zahl. Wenn man für k eine Zweierpotenz wählt, also $k = 2^\nu$, reduzieren sich die modulo-Operationen darauf, daß einfach die letzten $\nu + 1$ Bits der Zahl beibehalten werden und man den Rest auf Null setzt. Das kann aber durch eine logische AND-Operation mit der Binärzahl erreicht werden, deren letzte $\nu + 1$ Bits

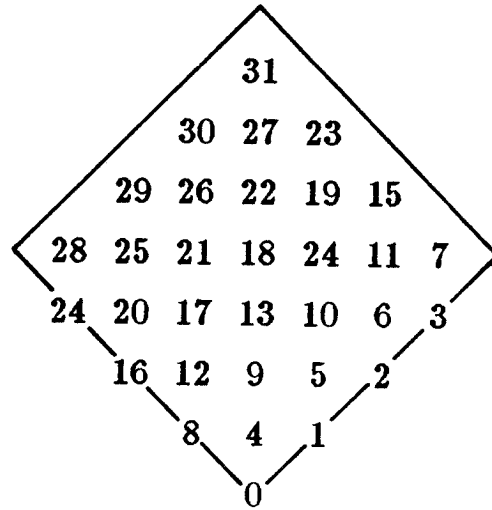


Bild IV.8: Hash-Funktion in $d = 2$ auf Rhombus.

aus Einsen bestehen. Da die logischen Operationen wesentlich weniger Zeit kosten, als aufwendige Divisionen mit Rest, wird die Berechnung der Hash-Funktion erheblich beschleunigt. Auch alle anderen Operationen einschließlich der Gaußklammer können mit Bit-Manipulationen dargestellt werden. Mit $NU := \nu$ und $MASK := 2k - 1 = 2^{\nu+1} - 1$ kann die Hash-Funktion schließlich als

$$h(x, y) = \text{OR}(\text{SHIFTL}(\text{AND}(y - x, \text{MASK}), NU), \text{SHIFTR}(\text{AND}(x + y, \text{MASK}), 1)) \quad (\text{IV.28})$$

geschrieben werden, wobei $\text{SHIFTL}(N, K)$ das Bitmuster der Zahl N um K Bits nach links verschiebt, was einer Multiplikation mit 2^K entspricht. $\text{SHIFTR}(N, 1)$ ist eine Verschiebung um ein Bit nach rechts, also eine Division durch 2. Da das letzte Bit bei dieser Operation verlorenggeht, ist damit auch die Gaußklammer richtig berücksichtigt. Die Auswertung dieser Formel kann für alle Random Walks gleichzeitig, d.h. vektorisiert, durchgeführt werden.

IV.3.5.2. Hash-Funktionen für $d > 2$

Obige Betrachtungen gelten nur in zwei Dimensionen. Die Konstruktion einer "Kugel" in $d = 3$ führt zu einem Oktaeder. Jedoch kann man mit Oktaedern den Raum nicht lückenlos füllen, wie in Bild IV.9 gezeigt ist. Es bleiben Tetraeder zwischen den Oktaedern frei. An jeden Oktaeder grenzen acht Tetraeder, während ein Tetraeder wiederum vier Oktaedern gemeinsam ist. Deshalb müßte man zu den Oktaedern jeweils noch zwei Tetraeder hinzunehmen, um eine Elementarzelle zu bekommen. Bei einer Kantenlänge von $\sqrt{2}k$ hätte die Elementarzelle ein Volumen von

$$V_{\min} = V_{\text{Okt.}} + 2V_{\text{Tet.}} = 2k^3. \quad (\text{IV.29})$$

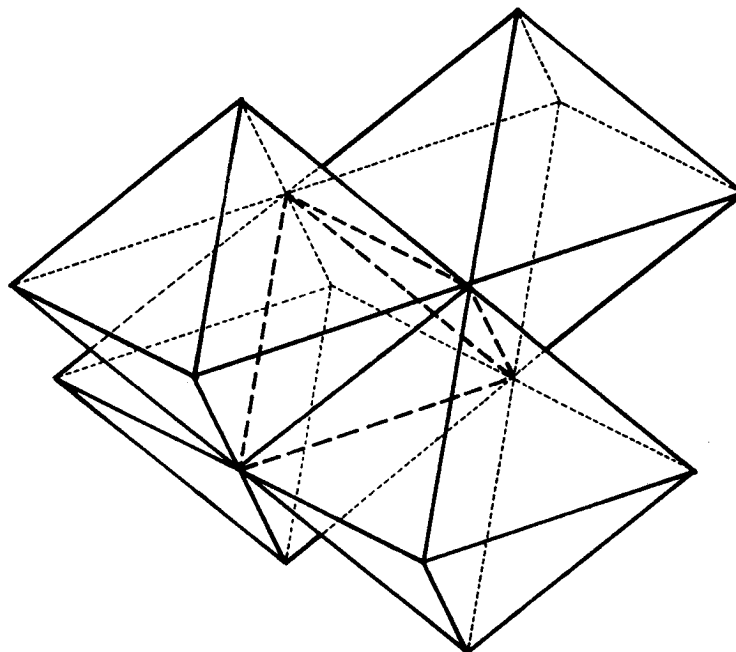


Bild IV.9: "Lückenhafte" Füllung des dreidimensionalen Raumes mit Oktaedern. Stärker hervorgehoben sind die Umrissse des Tetraeders, der zwischen den vier Oktaedern übrig bleibt.

Hier ist aber die Grenze erreicht, wo der zusätzliche Aufwand zur Berechnung der Position innerhalb einer Elementarzelle den Gewinn durch Vermeidung von Kollisionen übersteigt.

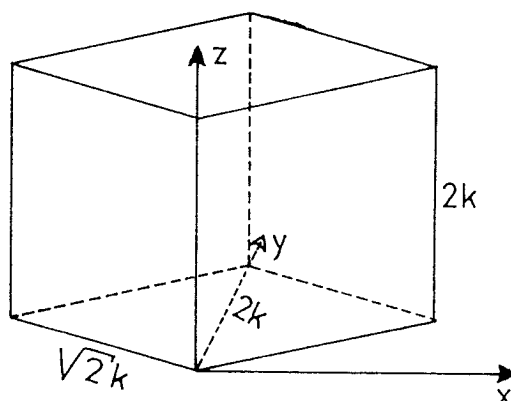


Bild IV.10: Elementarzelle der Hash-Funktion für $d = 3$. In z -Richtung wird der Rhombus der x,y -Ebene um $2k$ verschoben. Das überstrichene Volumen ist die neue Elementarzelle. Sie ist genau so hoch, wie die Diagonalen des Rhombus lang sind.

In mehr als zwei Dimensionen wird deshalb auf eine Hash-Funktion mit minimaler Länge der Tabelle bei gegebener Wahrscheinlichkeit für Kollisionen verzichtet. Statt

dessen wird jede weitere Dimension wie ein einfacher, achsenparalleler Kubus behandelt, siehe Bild IV.10. Damit ist immer noch der minimale Abstand zwischen äquivalenten Punkten $2k$, doch ist die Länge der Hash-Tabelle — das Volumen der Elementarzelle — größer, als es möglich wäre. Denn die so erhaltene Elementarzelle hat ein Volumen von

$$V_{EZ} = 4k^3. \quad (IV.30)$$

Die allereinfachste Zelle, die aus einem achsenparallelen Kubus mit einer Kantenlänge von $2k$ besteht, hat das Volumen $8k^3$. Trotz des Verzichts auf eine optimale Hash-Funktion spart man also immer noch einen Faktor

$$\frac{V_{EZ}}{8k^3} = \frac{1}{2}. \quad (IV.31)$$

Verglichen mit dem möglichen Verhältnis von

$$\frac{V_{\min}}{8k^3} = \frac{1}{4} \quad (IV.32)$$

könnte die Elementarzelle halb so groß sein, doch wäre der Rechenaufwand unverhältnismäßig größer.

Alle zusätzlichen Dimensionen werden also wie ein einfacher Kubus behandelt, d. h. die Hash-Funktion ist z. B. für $d = 3$

$$\begin{aligned} u &= x + y \bmod 2k \\ v &= y - x \bmod 2k \\ h &= 2k^2 z + kv + \left\lfloor \frac{u}{2} \right\rfloor. \end{aligned} \quad (IV.33)$$

Mit Bitmanipulationen schreibt man dann analog zu Gl. (IV.28)

$$\begin{aligned} h(x, y, z) &= \text{OR}(\text{OR}(\text{SHIFTL}(\text{AND}(y - x, \text{MASK}), \text{NU}), \\ &\quad \text{SHIFTR}(\text{AND}(y + x, \text{MASK}), 1), \text{SHIFTL}(\text{AND}(z, \text{MASK}), \text{NU} + 1)). \end{aligned} \quad (IV.34)$$

IV.3.6. Weitere technische Details

Im folgenden Abschnitt sollen noch einige weitere technische Details des Programms dargelegt werden, die die Speicherung von Gitterkoordinaten, die Wahl der Start-Walks, die Bestimmung der Korrelationszeiten und die Vektorisierung betreffen. Zum Schluß ist noch eine Bemerkung über die in das Programm eingebauten Testhilfen angefügt.

IV.3.6.1. Speicherung der Gitterkoordinaten

Jeder benutzte Knoten auf der Halde kennzeichnet einen besuchten Gitterplatz. Er enthält als Informationen eine Identifikation des Gitterplatzes selbst, die Anzahl der Besuche auf dem Platz und einen Zeiger auf den nächsten Knoten in der Liste (falls vorhanden). Zur Identifikation des Gitterplatzes sind die d Koordinaten des Punktes recht ungeeignet, weil dann immer d Größen miteinander verglichen, abgespeichert, etc. werden müssen. Statt dessen ist es günstiger, den Punkt durch eine einzige Zahl zu kennzeichnen. Diese Zahl wird im folgenden als *Schlüssel* bezeichnet.

Dazu denkt man sich ein großes Gitter mit einer Kantenlänge L , die größer ist als n . Man wählt als Schlüssel k , z. B. in $d = 3$

$$k = x + Ly + L^2z \quad \text{mit} \quad 0 < x, y, z \leq L. \quad (IV.35)$$

Damit ist sichergestellt, daß der Schlüssel eindeutig ist. Die Kantenlänge L wird als Zweierpotenz gewählt, damit die Berechnung des Schlüssels mit Hilfe von Bitmanipulationen sehr schnell erfolgen kann, denn der Schlüssel ist dann einfach die Aneinanderreihung der Bitmuster der Koordinaten, wodurch Multiplikationen vermieden werden. In $d = 3$ kann dann für den Schlüssel k geschrieben werden

$$k(x, y, z) = \text{OR}(\text{OR}(\text{SHIFTL}(\text{AND}(y, \text{MSKKEY}), \text{ISHKEY}), \text{AND}(x, \text{MSKKEY}), \text{SHIFTL}(\text{AND}(z, \text{MASK}), 2 * \text{ISHKEY})), \quad (IV.36)$$

wobei

$$L =: 2^{\text{ISHKEY}} \\ \text{MSKKEY} := 2^{\text{ISHKEY}} - 1 \quad (IV.37)$$

eingeführt wurden. (Die Notation ist angelehnt an das FORTRAN Programm in Anhang D.)

IV.3.6.2. Wahl der Start-Walks

Das Programm gibt immer nach einer vorgegebenen Zahl von MC-Schritten Mittelwerte über die Daten aller Walks aus, die seit der letzten Ausgabe simuliert wurden. Am Anfang wird unabhängig davon eine gewisse Zahl von MC-Schritten ausgeführt, um sicherzustellen, daß das thermische Gleichgewicht erreicht ist. Dazu ist ein Kriterium erforderlich, mit dem man feststellen kann, wann das Gleichgewicht erreicht ist.

Deshalb wird der nach Punkt dG1 auf Seite 49 erforderliche Start-Walk auf verschiedene Arten gewählt. Einerseits möchte man schnell ins thermische Gleichgewicht gelangen und sollte deshalb einen Walk möglichst nahe am Gleichgewicht erzeugen. Doch ist a priori nicht klar, wie ein solcher Walk auszusehen hat, da noch

nicht einmal die Wahrscheinlichkeitsverteilung $p_n(s)$ und schon gar nicht die Struktur eines Gleichgewichts-Random-Walks bekannt sind. Nach Kapitel II.4.2 ist es auch nicht möglich, einen Gleichgewichts-SIW schrittweise aufzubauen.

Aus diesem Grunde wurden drei verschiedene Anfangsbedingungen realisiert, auf die die zu simulierenden Walks beliebig verteilt werden können. (In der Praxis wurde dann jedoch immer jede Anfangsbedingung mit etwa einem Drittel der Walks besetzt.)

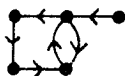
- **kompakteste Konfiguration**



Der Start-Walk ist der hin- und hergefaltete Walk, der nur zwei Plätze besucht. Bei hohen Haftstellenkonzentrationen liegt er gar nicht so weit weg vom Gleichgewicht, doch muß er durch stetiges Vergrößern des besuchten Bereiches beim Kriechen ins Gleichgewicht gelangen. Dabei wird sehr oft

der neue Walk zurückgewiesen und die Kriechrichtung umgekehrt.

- **Random-Konfiguration**



Als Start-Walk wird ein normaler Random-Walk verwendet, wie er bei Abwesenheit von Haftstellen entstehen würde. Er besucht bei endlicher Haftstellenkonzentration schon zu viele Gitterplätze, wird aber schnell durch Übereinanderkriechen auf Plätze, die er schon besucht hat, ins Gleichgewicht kommen.

men.

- **gestreckteste Konfiguration**



Der Start-Walk ist hier ein langgestreckter Walk, der entsteht, wenn ein Teilchen immer nur in eine Richtung springt. Er besucht $n + 1$ verschiedene Plätze und braucht entsprechend länger, um ins Gleichgewicht zu kommen. Dabei wird er aber anfangs kaum seine Kriechrichtung ändern, da es

sehr unwahrscheinlich ist, daß die Anzahl der besuchten Plätze wieder zunehmen würde. Diese Anfangsbedingung stellt das umgekehrte Extrem zu der kompaktesten Konfiguration dar.

Während der Rechnung werden alle simulierten Größen für die Walks mit verschiedenen Anfangsbedingungen getrennt gesammelt und ausgegeben. Dadurch kann man genau

erkennen, wann das Gleichgewicht erreicht wird, denn die mittlere Anzahl der besuchten Plätze ist z. B. für die drei Anfangsbedingungen sehr unterschiedlich. Erst wenn sich die Zahlen begegnen, kann man annehmen, daß das Gleichgewicht erreicht ist. Die Anzahl der MC-Schritte, die dazu nötig ist, stimmt in etwa mit den Korrelationszeiten, die in den Kapiteln IV.3.1.3 und IV.3.2.3 diskutiert wurden, überein.

IV.3.6.3. Bestimmung der Korrelationszeiten

Die Korrelationszeit τ selbst wird bestimmt, indem man die Kriechrichtung als Sprungrichtung eines ein-dimensionalen Random Walks betrachtet und die Größe des besuchten Bereiches mit Hilfe des am weitesten rechts und des am weitesten links besuchten Punktes berechnet. Hat dieser eindimensionale Walk $n + 1$ verschiedene Plätze besucht, sind einmal alle Einzelschritte des d -dimensionalen SIW's neu berechnet worden und der SIW wird als nächster, statistisch unabhängiger Walk gezählt. Die mittlere Anzahl der MC-Schritte, die dazu nötig ist, entspricht der Korrelationszeit τ . Zur zusätzlichen Kontrolle der statistischen Unabhängigkeit wird die Anzahl der Walks berechnet, die niemals zu einem statistisch unabhängigen SIW geführt haben. Bei sinnvoller Wahl der Simulationszeiten sollte diese Zahl immer Null sein.

IV.3.6.4. Vektorisierung

Viele Schritte des Algorithmus sind vektorisierbar, da sie für alle simulierten Walks in gleicher Weise ausgeführt werden können. Es bleibt ein nichtvektorisierbarer Teil übrig. Dort findet man den Code, der Kollisionen in den Hash-Tabellen auflöst, der die Entscheidung trifft, ob der neue Walk akzeptiert wird und in Abhängigkeit davon den eigentlichen Kriechschritt ausführt oder die Kriechrichtung umkehrt. Es sind viele logische Entscheidungen zu treffen, die einer Vektorisierung im Wege stehen. Trotzdem können einige Schritte, wie das Erzeugen von Zufallszahlen für die einzuschlagende Kriechrichtung und für die Entscheidung, ob der neue Walk akzeptiert wird, sowie das Berechnen der Hash-Funktionen und der Schlüssel-Codes aus diesem Teil herausgenommen werden und vorher in vektorisierter Form ausgeführt werden. Die Berechnung der zu mittelnden Größen kann dann nach diesem Programmteil ebenfalls wieder vektorisiert für alle Walks erfolgen. Die Anzahl der simulierten Walks sollte dabei ein Vielfaches von 64 betragen, da die Vektorregister der verwendeten CRAY gerade 64 Zahlen speichern können und Vektoroperationen am schnellsten laufen, wenn die Vektorregister immer voll ausgenutzt werden. Obwohl der Algorithmus an sich nicht vektorisierbar ist, wird in diesem Programm durch die gleichzeitige Simulation vieler Walks ein verhältnismäßig hoher Grad der Vektorisierung erreicht.

IV.3.6.5. Eingebaute Testhilfen

Da der Algorithmus vergleichsweise kompliziert ist, mußte besonders viel Wert auf einen einfachen Programmtest gelegt werden. Dazu wurde schon beim Schreiben des Programms darauf geachtet, daß alle Größen, die zu Fehlverhalten des Programms führen können, auf ihre Gültigkeit hin überprüft werden. Zum Beispiel werden alle Eingabedaten daraufhin getestet, ob sie den Programmkonventionen entsprechen. Während der Test- und auch während der Produktionsphase hat sich das immer wieder bewährt.

Sehr wesentlich war auch ein Test auf innere Konsistenz am Ende des Programms. Dort werden (auf Wunsch) alle überprüfbaren Werte noch einmal getestet. Zum Beispiel kann es durch ein fehlerhaftes Programm dazu kommen, daß Ketten während der Simulation "auseinanderreißen". Deshalb wird unter anderem aus dem gespeicherten Anfangspunkt des SIW und den gespeicherten Richtungen der Einzelschritte der Endpunkt der Kette berechnet und mit dem gespeicherten Endpunkt verglichen. Auch wird überprüft, ob die Hash-Tabellen korrekt geführt worden sind, indem der Eintrag jedes einzelnen Schrittes aller SIW's überprüft wird. Ferner wird die Anzahl aller benutzten Knoten bestimmt. Auf diese Weise kann festgestellt werden, ob durch fehlerhaftes Umspeichern ein Knoten aus der Liste aller verfügbaren Knoten verloren gegangen ist oder doppelt benutzt wurde.

In der Testphase konnten so einige Programmierfehler entdeckt werden. Da der Aufwand für diesen Test auf innere Konsistenz vergleichsweise gering ist, wurde er auch in der Produktionsphase nicht abgeschaltet. So konnte sogar ein selten auftretender Fehler gefunden werden, der nur unter ganz speziellen Bedingungen auftrat*). Ferner muß natürlich ein solcher Algorithmus auch anhand lösbarer Grenzfälle getestet werden. Erst dann kann man ein gewisses Vertrauen in die physikalischen Aussagen haben. Dieses wird im nächsten Kapitel beschrieben.

IV.4. DURCHFÜHRUNG DER SIMULATIONEN

Im folgenden wird kurz auf zwei Aspekte bei der Durchführung der Simulationen eingegangen.

*)Der Fehler trat nur dann auf, wenn der Kopf eines SIW's auf einen unbesuchten Platz kriecht, der äquivalent zu dem Platz ist, auf dem sich der Schwanz gerade befindet. Aber nur dann, wenn der Schwanz alleine diesen Platz besucht und zudem als erster in der Kollisionskette auf der Halde steht.

IV.4.1. Korrelationszeiten

Für die gewünschten Parameter (Schrittzahl n , Haftstellenkonzentration c) wurde zunächst ein kurzer Testlauf gemacht, aus dem die ungefähren Korrelationszeiten τ bestimmt wurden. Nach τ MC-Schritten sind im Mittel alle simulierten Walks in davon statistisch unabhängige übergegangen. Um mit Sicherheit statistisch unabhängige Walks in den einzelnen Gruppen zu haben, wurde das Doppelte dieser Korrelationszeit (im Programm: NMCOUT) als Zahl der MC-Schritte gewählt, nach denen jeweils die Zwischenergebnisse ausgegeben werden. Wiederum etwa das Dreifache von dieser Zahl wurde als Zahl der MC-Schritte gewählt, um am Anfang der Simulation ins Gleichgewicht zu gelangen (NEQUI). Für die drei verschiedenen Start-Walks ergaben sich nach dieser Zahl der Schritte keine systematischen Unterschiede mehr in den simulierten Größen, wodurch sich bestätigt hat, daß die Korrelationszeit tatsächlich die Zeitskala bestimmt, auf der die Relaxation ins Gleichgewicht stattfindet.

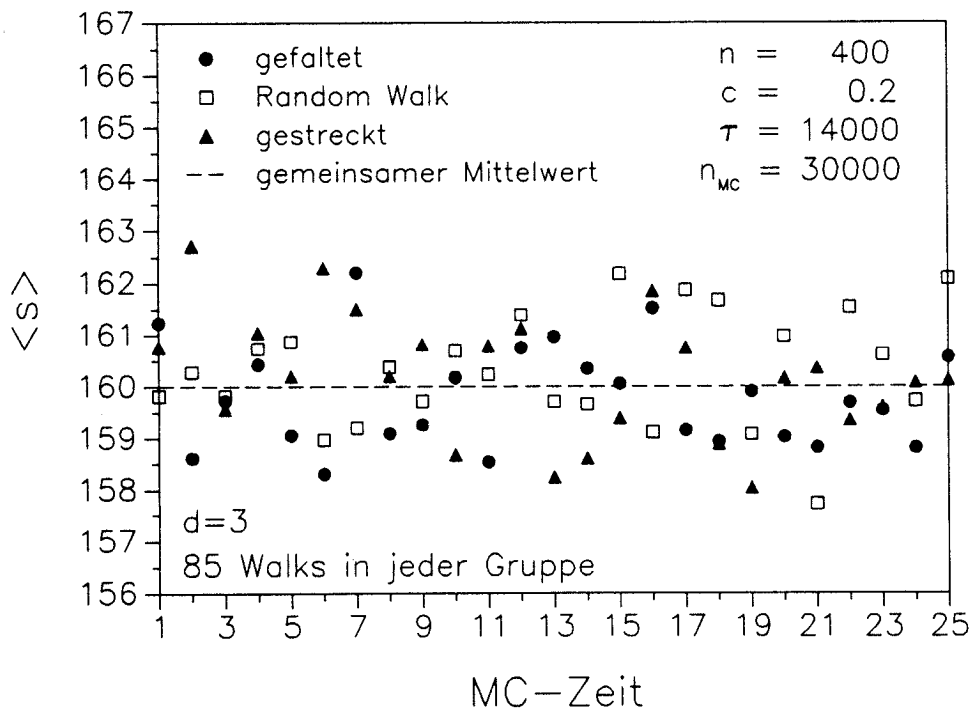


Bild IV.11: Statistische Abhängigkeit der Monte Carlo Mittelwerte. Unabhängig von den drei verschiedenen Anfangsbedingungen schwanken die einzelnen Mittelwerte um den gemeinsamen Mittelwert. Es sind nur Korrelationen über wenige aufeinanderfolgende Schritte zu erkennen (siehe Text).

In Bild IV.11 sind die mittleren Zahlen der besuchten Plätze nach jeweils NMCOUT MC-Schritten für die drei verschiedenen Start-Walks dargestellt. Man erkennt, daß es keinerlei Abhängigkeit der Werte von den verschiedenen Anfangsbedingungen gibt.

Ferner schwanken die Werte weitgehend zufällig um den gemeinsamen Mittelwert. Es gibt lediglich Korrelationen über wenige, aufeinanderfolgende Mittelwerte. Diese Korrelationen sind eine Konsequenz daraus, daß aufeinanderfolgende Mittelwerte Walks beinhalten, die näher als die Korrelationszeit τ voneinander entfernt sind. In Anhang E werden diese Korrelationen berechnet. Dort wird auch gezeigt, wie man die Schwankung der einzelnen Mittelwerte benutzen kann, um Fehlerabschätzungen für die simulierten Größen zu bekommen. Dabei ist es wichtig, die auftretenden Korrelationen zu berücksichtigen.

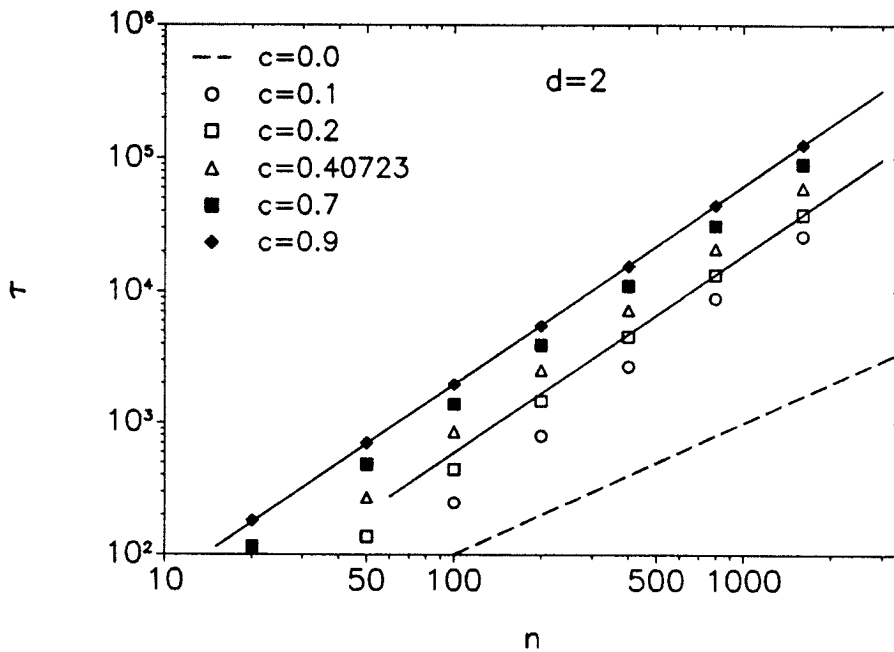


Bild IV.12: Korrelationszeiten in $d = 2$. Doppelt logarithmische Auftragung der Korrelationszeiten gegen die Schrittzahl für verschiedene Haftstellenkonzentrationen. Die beiden eingezeichneten Geraden stellen ein $\tau \sim n^{1,5}$ -Gesetz dar.

In Bild IV.12 sind die ermittelten Korrelationszeiten in $d = 2$ als Funktion der Schrittzahl n für verschiedene Haftstellenkonzentrationen c doppelt logarithmisch aufgetragen. Im asymptotischen Bereich erkennt man deutlich eine Potenzgesetzmäßigkeit. Die eingezeichneten Geraden entsprechen einer Abhängigkeit

$$\tau_{2d} \sim n^{1,5}. \quad (IV.38)$$

In Bild IV.13 ist dasselbe für $d = 3$ aufgetragen. Hier ergibt sich die Abhängigkeit zu

$$\tau_{3d} \sim n^{1,6}. \quad (IV.39)$$

(Siehe dazu auch Kapitel IV.3.2.3.)

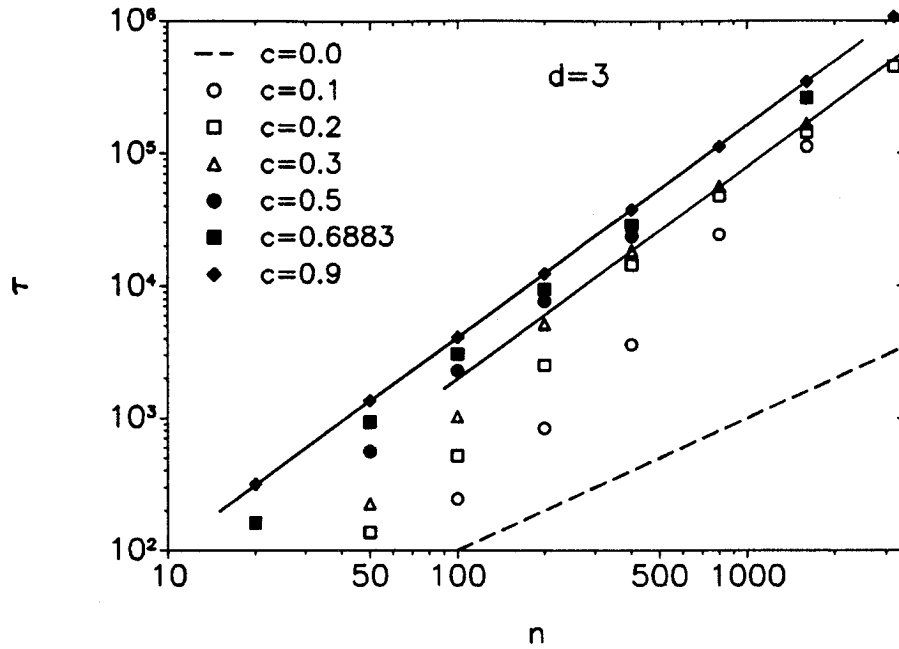


Bild IV.13: Korrelationszeiten in $d = 3$. Doppelt logarithmische Auftragung der Korrelationszeiten gegen die Schrittzahl für verschiedene Haftstellenkonzentrationen. Die beiden eingezeichneten Geraden stellen ein $\tau \sim n^{1,6}$ -Gesetz dar.

Aus diesen Diagrammen sind die Korrelationszeiten auch ohne einen vorherigen Probelauf des Programms zu entnehmen. Die Korrelationszeit bestimmt die Gesamtzahl NRW der benötigten Monte Carlo Schritte. Daraus und aus der Zahl NW der simulierten Walks läßt sich überschlägig die benötigte Rechenzeit mit Hilfe der Formel

$$\text{Rechenzeit [min]} = \text{NMC} \cdot \text{NW} \begin{cases} 0,12 \cdot 10^{-6} \text{ min,} & \text{für } d = 2 \\ 0,13 \cdot 10^{-6} \text{ min,} & \text{für } d = 3 \end{cases}$$

bestimmen. Damit ist der Zeitbedarf für einen Produktionslauf ohne einen Probelauf abschätzbar.

IV.4.2. Vergleich der Ergebnisse mit denen anderer Methoden

Für Random Walks auf Gittern ohne Haftstellen sind viele der betrachteten Größen exakt bekannt. So z. B. die mittlere quadratische Auslenkung

$$\langle R^2 \rangle = n, \quad (\text{IV.40})$$

und der Trägheitsradius G [99, 105]

$$G^2 := \frac{1}{n+1} \sum_{k=0}^n \vec{r}_k^2 - \left(\frac{1}{n+1} \sum_{k=0}^n \vec{r}_k \right)^2 \quad (\text{IV.41})$$

$$\langle G^2 \rangle = \frac{n}{6} \frac{n+2}{n+1}.$$

Die mittlere Zahl der besuchten Plätze ist asymptotisch für $n \rightarrow \infty$ in $d = 2$ und $d = 3$ nach Gl. (II.16) bekannt. Für einen genauen Test reicht das jedoch bei durchaus endlichen Schrittzahlen n nicht aus. Im Anhang F werden Rekursionsformeln für die exakte mittlere Zahl von besuchten Plätzen in $d = 2$ und $d = 3$ nach n Schritten hergeleitet. Diese Werte wurden mit den Simulationsergebnissen für $c = 0$ verglichen, und die simulierten Werte stimmten im Rahmen der statistischen Fehler mit den exakten Werten überein.

Für $c > 0$ liegen in $d > 1$ keine exakten Resultate vor. Es können lediglich exakte Abzählungen mit dem Computer durchgeführt werden. Mit vertretbaren Rechenzeiten geht das problemlos bis etwa $n = 15$ in $d = 2$ und bis $n = 10$ in $d = 3$. In den weiter unten dargestellten Resultaten sind für diese kleinen Schrittzahlen jeweils die Ergebnisse der exakten Abzählungen eingetragen.

Ferner wurde für einige Parameterwerte die Havlin Methode auf zufälligen Haftstellen-Konfigurationen — Abschnitt IV.1.2 — und mit der Simulation von Walks auf Gittern ohne Haftstellen — Abschnitt IV.1.3 — angewandt. Alle Resultate stimmten im Rahmen der Fehlergrenzen mit der neuen Simulationsmethode überein. Dies stimmt auch für die Scanning-Methode von Meirovitch. Details sind bei der Darstellung der Ergebnisse auf den folgenden Seiten zu finden.

Alles in allem kann gesagt werden, daß das Programm zuverlässig arbeitet und korrekte Ergebnisse in allen überprüfbaren Bereichen liefert.

V. Interpretation der Ergebnisse

Bei Random Walks ist die Dimension 2 eine Grenzdimension. Bis einschließlich der Dimension 2 ist es zum Beispiel sicher, daß ein Teilchen bei einem Random Walk irgendwann einmal wieder an den Ausgangspunkt zurückkehrt. Im Mittel dauert es allerdings in zwei Dimensionen unendlich lange, bis dieses Ereignis eintritt. Letztlich ist das der Grund, warum nur im zweidimensionalen Fall ein logarithmischer Term in der mittleren Zahl der besuchten Plätze auftaucht. Diese Abhängigkeit führt dazu, daß man keine Skalentheorien für eine Reihe von interessanten Größen, wie auch der Überlebenswahrscheinlichkeit auf Gittern mit Haftstellen, angeben kann. Daß das in drei und höheren Dimensionen ganz anders ist, wird die nun folgende Diskussion der Simulationsergebnisse zeigen.

V.1. SKALENTHEORIE FÜR $d \geq 3$

Die Größe, die bei der Simulation die primäre Rolle spielt, ist die mittlere Zahl der besuchten Plätze $\langle s \rangle_{\text{SIW}}$. Da man über sie bei endlichen Haftstellenkonzentrationen keine Informationen hat, muß man für eine Skalentheorie auf die Überlebenswahrscheinlichkeit Ψ_n zurückgreifen, für die man sowohl Kurzzeit- als auch Langzeitergebnisse hat. Deshalb wird zunächst eine Skalenform für Ψ_n hergeleitet, aus der anschließend auch eine solche für $\langle s \rangle_{\text{SIW}}$ folgt.

Im folgenden sei immer

$$n \gg 1 \quad (V.1)$$

vorausgesetzt. Wie weiter oben schon diskutiert wurde, ist die Kumulantenentwicklung von

$$-\ln \Psi_n = -\ln \langle e^{-\lambda s} \rangle = \lambda \langle s \rangle - \frac{1}{2} \lambda^2 \Delta^2 s(c=0) + O(\lambda^3) \quad (V.2)$$

für endliche Schrittzahlen n und genügend kleine Haftstellenkonzentrationen c eine gute Näherung für die Überlebenswahrscheinlichkeit. Für $\lambda \rightarrow 0$ stellt der erste Term im Exponenten den Hauptbeitrag dar (Rosenstock-Näherung). Für genügend große Schrittzahlen n gilt das Donsker und Varadhan-Verhalten, Gl. (II.18). Deshalb kann man in d Dimensionen für die Überlebenswahrscheinlichkeit Ψ_n schreiben

$$-\ln \Psi_n \sim \begin{cases} \lambda S_n, & \text{für } \lambda \rightarrow 0, n \text{ endlich} \\ B_d \lambda^{\frac{2}{d+2}} n^{\frac{d}{d+2}}, & \text{für } \lambda \text{ endlich, } n \rightarrow \infty. \end{cases} \quad (V.3)$$

Dabei kommt das Kurzzeitverhalten von der Kumulantenentwicklung Gl. (V.2) und das Langzeitverhalten von Gl. (II.18). Beschränkt man sich auf Dimensionen mit $d \geq 3$,

kann man für S_n nach Gl. (II.19)

$$S_n = C_d n \quad (V.4)$$

schreiben. Macht man nun den Skalenansatz

$$-\ln \Psi_n = n^\alpha f(\lambda n^\Phi), \quad (V.5)$$

so folgt für die Skalenfunktion $f(x)$

$$f(x) \sim \begin{cases} C_d x, & \text{für } x \rightarrow 0 \\ B_d x^{\frac{2}{d+2}}, & \text{für } x \rightarrow \infty, \end{cases}$$

damit man die richtigen λ -Abhängigkeiten bekommt. Einsetzen in Gl. (V.3) liefert sofort $\alpha = (d-2)/d$ und $\Phi = 2/d$, also

$$-\ln \Psi_n = n^{\frac{d-2}{d}} f(\lambda n^{\frac{2}{d}}). \quad (V.6)$$

Damit sollte $(-\ln \Psi_n)n^{(2-d)/d}$ nur von der Skalenvariable $x = \lambda n^{2/d}$ abhängen. Andere Autoren [28] haben

$$\rho = \lambda^{\frac{2}{d+2}} n^{\frac{d}{d+2}}$$

als Skalenvariable vorgeschlagen und bekamen unterhalb von $\rho = 10$ starke Streuungen in ihrem Skalenplot, den sie auf Korrekturen zum Skalenverhalten zurückführten. Diese Streuungen kommen aber zumindest zum Teil auch von der Vernachlässigung des Kurzzeitverhaltens ($x \rightarrow 0$).

Nach Gl. (II.26) gilt

$$\langle s \rangle_{\text{SIW}} = -\frac{\partial \ln Z}{\partial \lambda} = n f'(\lambda n^{\frac{2}{d}}). \quad (V.7a)$$

Darin geht $f'(x)$ wie

$$f'(x) \sim \begin{cases} C_d, & \text{für } x \rightarrow 0 \\ \frac{2}{d+2} B_d x^{-\frac{d}{d+2}}, & \text{für } x \rightarrow \infty. \end{cases} \quad (V.7b)$$

Für die Varianz $\Delta^2 s$ erhält man nach Gl. (II.27) entsprechend

$$\Delta^2 s = -\frac{\partial \langle s \rangle_{\text{SIW}}}{\partial \lambda} = n^{\frac{d+2}{d}} f''(\lambda n^{\frac{2}{d}}) \quad (V.8a)$$

mit

$$f''(x) \sim \begin{cases} 0, & \text{für } x \rightarrow 0 \\ \frac{2d}{(d+2)^2} B_d x^{-2\frac{d+1}{d+2}}, & \text{für } x \rightarrow \infty. \end{cases} \quad (V.8b)$$

Hier ist der führende Term in der Kurzzeitentwicklung exakt verschwunden. Wenn man das Verhalten für kleine x genauer haben wollte, müsste man eine eigene Skalentheorie für $\Delta^2 s$ entwickeln. Da aber nach Ref. 106 für $\Delta^2 s$ auf Gittern ohne Haftstellen

$$\Delta^2 s \sim n \ln n \quad \text{für } n \gg 1, \quad \lambda \rightarrow 0 \quad (\text{V.9})$$

gilt, ist eine Skalentheorie, die das Kurzzeitverhalten richtig mit einschließt, wegen des logarithmischen Terms für die Varianz von s in $d = 3$ nicht möglich.

V.2. MITTLERE ZAHL DER BESUCHTEN PLÄTZE

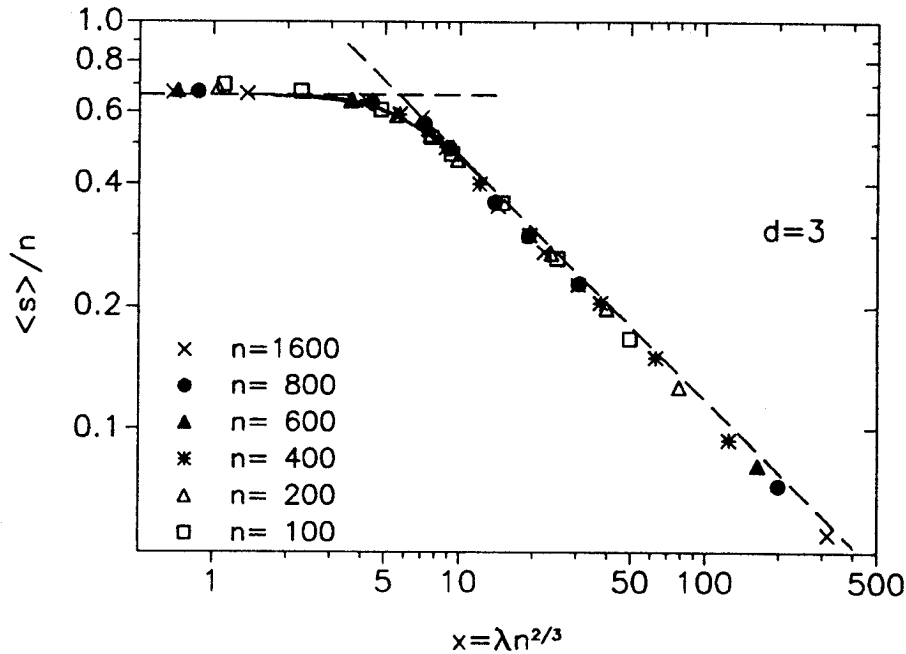


Bild V.1: Skalierte mittlere Zahl der besuchten Plätze als Funktion der Skalenvariablen x für $d = 3$. Doppelt logarithmische Auftragung für verschiedene Schrittzahlen. Die durchgezogene Linie interpoliert zwischen Kurz- und Langzeitverhalten, siehe unten, Abschnitt Überlebenswahrscheinlichkeit.

In Bild V.1 ist $\langle s \rangle_{\text{SIW}}/n$ als Funktion von x doppelt logarithmisch aufgetragen. Bemerkenswert ist, daß die simulierten Werte nach Gl. (V.7) nicht nur für sehr große und sehr kleine x -Werte auf einer universellen Kurve liegen, sondern sogar über den ganzen Bereich hinweg. Gl. (V.7) ist also auch im Crossoverbereich zwischen Kurzzeit- und Langzeitverhalten gut erfüllt. Die beiden Grenzfälle sind als gestrichelte Geraden mit eingezeichnet. Die Kurve schmiegt sich von unten an die beiden Asymptoten an. Der Crossover-Punkt x_c liegt dort, wo sich die beiden Asymptoten schneiden, d. h.

$$C_d = \frac{2}{d+2} B_d x_c^{-\frac{2}{d+2}}$$

oder

$$x_c = \left(\frac{2}{d+2} \frac{B_d}{C_d} \right)^{\frac{d+2}{d}}. \quad (V.10)$$

Für $d = 3$ ergibt sich

$$x_c = 5,70338. \quad (V.11)$$

Für kleine Schrittzahlen spürt das diffundierende Teilchen noch nichts davon, daß es sich auf einem Gitter mit Haftstellen befindet. Alle Größen verhalten sich ähnlich wie bei einem reinen Random Walk. Insbesondere entspricht die Anzahl der besuchten Plätze dem Mittelwert auf Gittern ohne Haftstellen nach Gl. (II.16).

Im Grenzfall extrem großer Schrittzahlen tragen nur noch die Teilchen zur Überlebenswahrscheinlichkeit bei, die sich in den Haftstellen-freien Kugeln mit Radius $R(c, n)$ nach Gl. (II.10) aufhalten. Diese Kugeln haben nach Gl. (II.19) $s_{\max}(c, n)$ Gitterplätze. Die Walks, die am meisten beitragen, werden diesen Raum auch ganz ausnutzen, da die mittlere Zahl von besuchten Plätzen nach Gl. (II.16) ohne Haftstellen wesentlich größer wäre.

Der Crossover findet also dort statt, wo die Anzahl der besuchten Plätze ohne Haftstellen S_n so groß wird wie die relevante Zahl von besuchten Plätzen $s_{\max}$.

Für $d \geq 3$ folgt aus $S_n = s_{\max}$ sofort wieder Gl. (V.10). Aus dieser Überlegung ergibt sich unmittelbar, daß $x = \lambda n^{2/d}$ die richtige Skalenvariable ist.

Aus Bild V.1 entnimmt man, daß der Crossover vom Kurzzeit- zum Langzeitverhalten relativ scharf geschieht. Man könnte nun folgende Näherung für $\langle s \rangle_{\text{SIW}}$ aufstellen

$$\frac{\langle s \rangle_{\text{SIW}}}{n} \approx \begin{cases} C_d, & \text{für } x \leq x_c \\ \frac{2}{d+2} B_d x^{-\frac{d}{d+2}}, & \text{für } x > x_c, \end{cases} \quad (V.12)$$

mit x und x_c wie oben. Dabei ist die Skalenfunktion $f(x)$ vor und nach x_c durch die jeweilige Asymptotik ersetzt worden. Bei x_c macht die Steigung der Näherungsfunktion also einen Sprung. Die Folgerungen dieser Näherung für die Überlebenswahrscheinlichkeit werden im nächsten Abschnitt dargestellt.

V.3. DIE ÜBERLEBENSWAHRSCHEINLICHKEIT

Die Überlebenswahrscheinlichkeit ist nicht auf direktem Wege aus der Simulation zu erhalten, da sie der Zustandssumme entspricht, also die Summe aller statistischen Gewichte ist, während in eine Monte Carlo Simulation nach Metropolis nur die Verhältnisse der Gewichte zwischen zwei Zuständen eingehen. Wie oben schon erwähnt, kann die Überlebenswahrscheinlichkeit jedoch als Integral der Anzahl der besuchten Plätze dargestellt werden. Es gilt nach Gl. (IV.25)

$$-\ln \Psi_n = \int_0^\lambda \langle s \rangle_{\text{SIW}} d\lambda. \quad (\text{V.13})$$

Die Werte von $\langle s \rangle_{\text{SIW}}$ für verschiedene λ aus der Simulation müssen numerisch integriert werden, wobei aber nur die Werte an wenigen Stützstellen bekannt sind. Um derartige Integrale zu berechnen, werden üblicherweise kubische Spline-Funktionen benutzt, mit denen man die Ergebnisse interpoliert. Die Spline-Funktionen können dann exakt integriert werden. Kubische Spline-Funktionen sind stückweise zwischen zwei Stützstellen definierte Polynome dritten Grades, die so bestimmt werden, daß sie an den Stützstellen nicht nur stetig sind, sondern auch in der Steigung und Krümmung übereinstimmen. Gibt man neben den Stützpunkten zusätzlich für den ersten und den letzten Stützpunkt noch die Steigung oder die Krümmung vor, sind sie eindeutig definiert [107].

Interpolationspolynome neigen jedoch dazu, in Bereichen mit großen Krümmungen überzuschwingen. Das Crossovergebiet ist ein solcher Bereich. Um das zu verhindern, kann man vor der Interpolation das globale Verhalten der Funktion, das durch die beiden asymptotischen Theorien gegeben ist, von $f(x) = \langle s \rangle_{\text{SIW}}/n$ abziehen. Damit im Crossover-Gebiet kein Knick (und damit eine große Krümmung) entsteht, kann man die Funktion

$$h(x) := \left[C_d^{-\frac{d+2}{d}\beta} + \left(\frac{2}{d+2} B_d \right)^{-\frac{d+2}{d}\beta} x^\beta \right]^{-\frac{d}{d+2}\frac{1}{\beta}} \quad (\text{V.14})$$

abziehen, die zwischen dem Rosenstock-Verhalten bei kleinen x und dem Donsker-Varadhan-Verhalten für große x interpoliert, siehe Gl. (V.7). Der Exponent β wurde für $d = 3$ willkürlich zu 4 gewählt. Das Crossover-Gebiet von $f(x)$ wird dadurch recht genau nachgebildet, siehe Bild V.1. Die Differenzwerte $f(x) - h(x)$ sind über den gesamten x -Bereich relativ klein und im wesentlichen konstant. Man kann sogar auf Spline-Funktionen verzichten und die Integration der Differenzfunktion mit der gewöhnlichen Trapezregel vornehmen, denn der Hauptbeitrag zum Integral über $f(x)$ kommt vom der Integration über die Funktion $h(x)$. Sie ist leider nicht in geschlossener Form durchführbar, doch ist eine numerische Integration problemlos möglich, da die

Funktion im gesamten Bereich stetig ist und sich über weniger als eine Dekade erstreckt. Dazu wurde das Programm QROMB aus Ref. 107 benutzt, das eine Romberg-Integration durchführt.

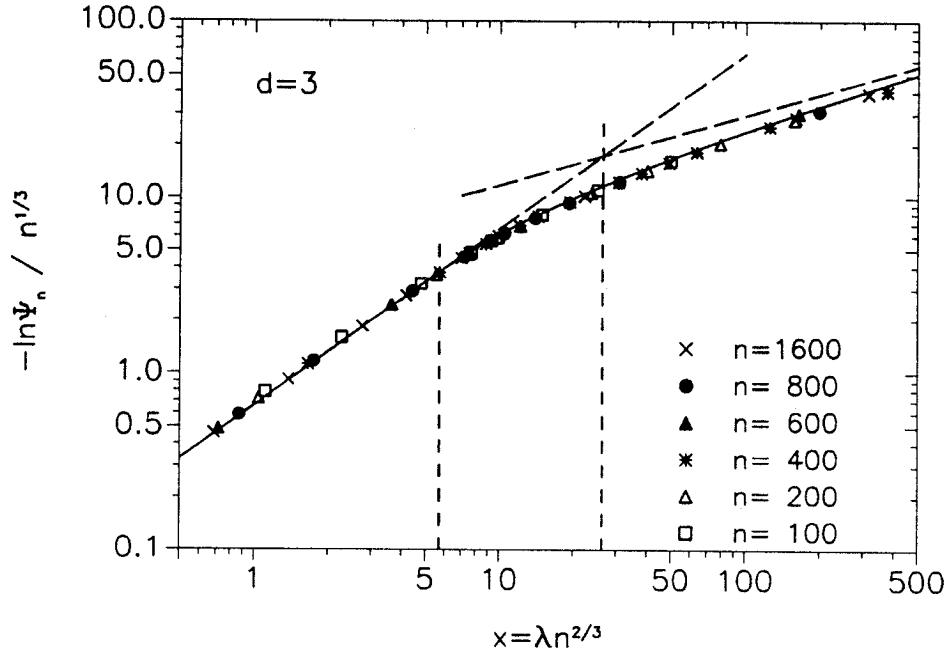


Bild V.2: Skalenplot der Überlebenswahrscheinlichkeit in $d = 3$. Punkte: numerisch integrierte Monte Carlo Daten. Durchgezogene Linie: analytische Näherung.

In Bild V.2 sind für verschiedene Haftstellenkonzentrationen die Überlebenswahrscheinlichkeiten als Funktion der Skalenvariablen x aufgetragen. Man erkennt, daß die Werte nach der Integration immer noch sehr gut auf einer universellen Kurve liegen. In drei Dimensionen ist also auch das Skalenverhalten Gl. (V.6) sehr gut erfüllt.

Man kann die Näherung Gl. (V.12) benutzen, um einen analytischen Ausdruck für die Überlebenswahrscheinlichkeit zu bekommen. Nach Gl. (IV.25) ergibt sich die Überlebenswahrscheinlichkeit zu

$$-\ln \Psi_n = n^{\frac{d-2}{d}} \int_0^x \frac{\langle s \rangle_{\text{SIW}}}{n} dx. \quad (\text{V.15})$$

Mit Gl. (V.12) folgt

$$\frac{-\ln \Psi_n}{n^{\frac{d-2}{d}}} \approx \begin{cases} C_d x, & \text{für } x \leq x_c \\ B_d x^{\frac{2}{d+2}} - \left(B_d x_c^{\frac{2}{d+2}} - C_d x_c \right), & \text{für } x > x_c, \end{cases} \quad (\text{V.16})$$

Der Unterschied zu Gl. (V.3) ist die zusätzliche Integrationskonstante für $x > x_c$, die garantiert, daß beide Bereiche stetig ineinander übergehen.

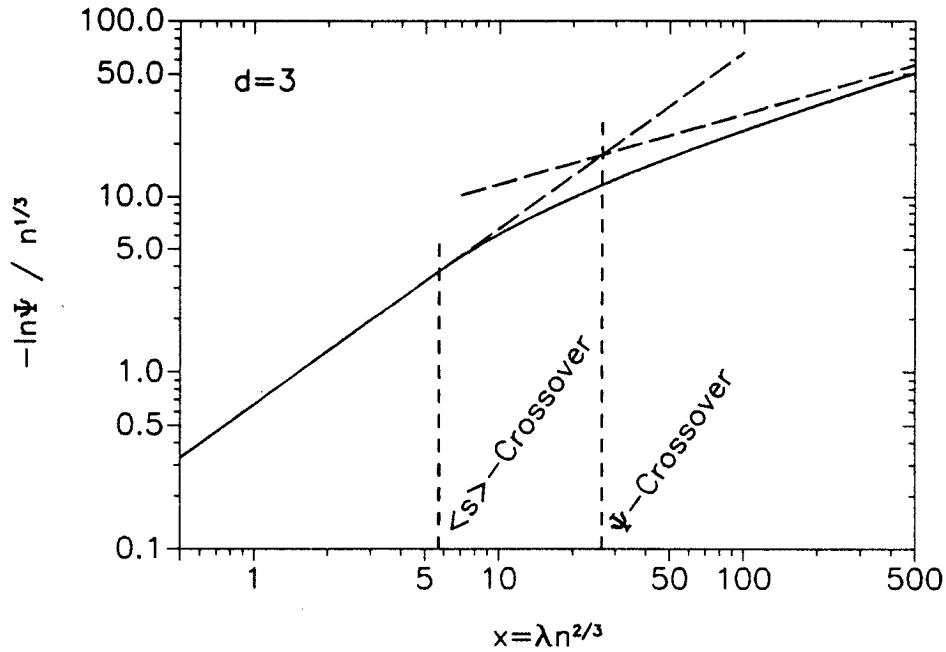


Bild V.3: Approximation der Überlebenswahrscheinlichkeit für $d = 3$. Eingezeichnet sind die Crossover-Punkte von $\langle s \rangle_{\text{SIW}}$ und Ψ_n .

In Bild V.3 ist die Näherung für Ψ_n für $d = 3$ aufgetragen. Man erkennt, daß der Crossover für Ψ_n , wenn man ihn als Schnittpunkt der beiden asymptotischen Geraden in der doppelt logarithmischen Auftragung definiert, bei einem höheren x -Wert liegt als x_c . Obwohl für $x > x_c$ das asymptotische Verhalten von $\langle s \rangle_{\text{SIW}}$ als exakt angenommen wurde, nähert sich Ψ_n nur sehr langsam dem Verhalten für $x \rightarrow \infty$. Das scheint mir der Grund dafür zu sein, daß es mit konventionellen Methoden so schwierig ist, die Asymptotik von Ψ_n zu sehen.

Es gilt für den Ψ -Crossoverpunkt x_c^Ψ in $d \geq 3$

$$x_c^\Psi = \left(\frac{B_d}{C_d} \right)^{\frac{d+2}{d}}. \quad (\text{V.17})$$

Für $d = 3$ erhält man

$$x_c^\Psi = 23,264. \quad (\text{V.18})$$

Noch drastischer präsentieren sich die Unterschiede, wenn man die Überlebenswahrscheinlichkeit selbst betrachtet. Beispielsweise ist für die Schrittzahl $n = 100$ in $d = 3$ die Überlebenswahrscheinlichkeit Ψ am Crossoverpunkt x_c in dieser Näherung

$$\Psi(x_c, n = 100) = 2,6 \cdot 10^{-8},$$

während für den Ψ -Crossoverpunkt x_c^Ψ

$$\Psi(x_c^\Psi, n = 100) = 2,9 \cdot 10^{-24},$$

gilt. Letzterer wurde bisher ausschließlich in der Literatur betrachtet. Wie sich durch obige Betrachtungen aber ergibt, findet der "wahre" Crossover — also der bei x_c — bei deutlich höheren Überlebenswahrscheinlichkeiten statt. Leider ist zu sagen, daß Überlebenswahrscheinlichkeiten von 10^{-8} experimentell wohl nicht oder nur sehr schwer zugänglich sind. Für noch kleinere Schrittzahlen n ist zwar die Überlebenswahrscheinlichkeit am Crossoverpunkt höher, doch ist dann $n \gg 1$ nicht mehr gegeben und der Crossover wird weiter aufgeweicht. Das Kurzzeit- ist nicht mehr klar vom Langzeitverhalten getrennt.

In Bild V.2 ist die analytische Näherung ebenfalls mit eingetragen worden. Man erkennt, daß die Abweichungen zu den Monte Carlo Daten auf der logarithmischen Skala vernachlässigbar klein sind. Das legt nahe, experimentelle Resultate über das Langzeitverhalten der Überlebenswahrscheinlichkeit $\Psi(t)$ als Funktion der Zeit t in $d = 3$ mit der zweiparametrigen Fitfunktion

$$-\ln \Psi(t) = at^{3/5} - bt^{1/3} \quad (V.19)$$

zu untersuchen. Man beachte, daß der hinzugefügte Term $bt^{1/3}$ nicht der ersten Korrektur nach Lubensky [33] oder Schotland [34] entspricht. Er ist in dieser Form noch nicht in der Literatur erschienen. Das Hinzufügen dieses Terms erlaubt eine wesentlich bessere Anpassung an die Daten, als wenn man nur nach dem reinen asymptotischen Verhalten $at^{3/5}$ sucht. Nimmt man an, daß die Sprünge nach einer mittleren Verweilzeit τ stattfinden, also

$$n = \frac{t}{\tau},$$

so können die beiden Fitparameter a und b mit den Werten für τ und λ nach Gl. (V.16) in Verbindung gebracht werden. Es gilt

$$\begin{aligned} \tau &= \left(\frac{B_3 x_c^{2/5} - C_3 x_c}{b} \right)^3 = \left(\frac{5,642}{b} \right)^3 \\ \lambda &= \left(\frac{a}{B_3} \right)^{5/2} \tau^{3/2} = \left(\frac{a}{4,686} \right)^{5/2} \tau^{3/2}. \end{aligned} \quad (V.20)$$

Da für diese Formeln nur das Langzeitverhalten herangezogen wurde, sollte sichergestellt werden, daß die verwendeten Meßergebnisse auch aus diesem Bereich stammen, daß also

$$x = \lambda n^{2/3} \gg 5,703 = x_c$$

für die verwendeten Werte gilt.

Abschließend soll noch einmal betont werden, daß der Punkt, an dem der Crossover stattfindet, bei Überlebenswahrscheinlichkeiten

$$\Psi_n \approx \exp(-3,9 \cdot n^{1/3})$$

liegt und damit für große n zur Zeit nicht experimentell zugänglich ist. Ob diese Effekte in drei Dimensionen meßbar sind, war bisher eine offene Frage.

Bei den Betrachtungen zur Havlinmethode in Abschnitt IV.1.2 wurde eine Bedingung für das Auftreten des Crossovers bei Simulationen im Computer angegeben, bei der der Speicherplatz, d.h. die Anzahl der simulierten Gitterplätze, logarithmisch einging. Man kann nun die Frage stellen, ob in realen Systemen mit einer Zahl von Gitterplätzen von der Ordnung 10^{23} überhaupt ein Crossover zu beobachten wäre. Mit

$$K_d := \left(\frac{1}{\Omega_d}\right)^{\frac{2}{d}} \left(\frac{d}{\alpha_d}\right)^2 (\ln L)^{\frac{d+2}{d}} \quad (V.21)$$

schreibt sich Gl. (IV.1) als

$$n\lambda^{2/d} \ll K_d. \quad (V.22)$$

Da der Crossover bei einem bestimmten $x = \lambda n^{2/d}$ auftritt, ergibt sich daraus am Crossoverpunkt $x = x_c$

$$\lambda \gg \left(\frac{x_c^{d/2}}{K_d}\right)^{\frac{2d}{d^2-4}} \quad \text{bzw.} \quad n \ll \left(\frac{K_d}{x_c^{2/d}}\right)^{\frac{d^2}{d^2-4}}. \quad (V.23)$$

In diesen Bedingungen kann L einerseits als die maximale Zahl der zu speichernden Gitterplätze interpretiert werden (etwa $L \approx 10^6$), andererseits als die Zahl der in realistischen Proben zu erwartenden Gitterplätze (etwa $L = 6 \cdot 10^{23}$). Dann sagen obige Gleichungen aus, bei welchen Parametern überhaupt in der Natur ein Crossover zu dem Langzeitverhalten auftritt und (zumindest prinzipiell) meßbar ist.

L	$c \gg$	$n \ll$
10^6	0,344	49,6
$6 \cdot 10^{23}$	0,027	3087

Tab. V.1: Bereich der Parameter, in dem Crossover beobachtbar ist.

Mit obigen Zahlen erhält man in $d = 3$ die in Tabelle Tab. V.1 angegebenen Bereiche, wobei λ auf die Haftstellenkonzentration c umgerechnet wurde. Man erkennt, daß

die Havlinmethode nur für extrem kleine Schrittzahlen und entsprechend hohe Haftstellenkonzentrationen den asymptotischen Bereich erreicht. Die oben eingeführte neue Simulationsmethode hingegen erreicht diesen Bereich auch für deutlich höhere Schrittzahlen (auch noch für $n = 3200$). Bemerkenswert ist hingegen, daß selbst in der Natur die relevanten Haftstellen-freien Gebiete nur dann genügend oft auftreten, wenn die Haftstellenkonzentration deutlich größer als 2,7% ist, um das Crossover-Verhalten beobachten zu können. Man könnte sogar sagen, daß die in dieser Arbeit eingeführte Simulationsmethode besser als die Natur selbst funktioniert, da sie weiter in den asymptotischen Bereich hineinstößt, als es die Natur mit etwa 10^{23} Atomen kann.

V.4. DER EFFEKTIVE EXPONENT α

Wenn man Ψ_n durch einen effektiven Exponenten α als $\Psi_n = \exp(-Kn^\alpha)$ schreiben möchte, findet man mit Gl. (V.6)

$$\alpha := \frac{\partial \ln(-\ln \Psi_n)}{\partial \ln n} = \frac{d-2}{d} + \frac{2}{d} x \frac{f'(x)}{f(x)}, \quad (\text{V.24})$$

der damit ebenfalls Skalenform in $d \geq 3$ Dimensionen hat. Für die Näherung Gl. (V.16) erhält man $\alpha = 1$ für $x < x_c$. Für $x > x_c$ fällt α langsam auf $d/(d+2)$ ab.

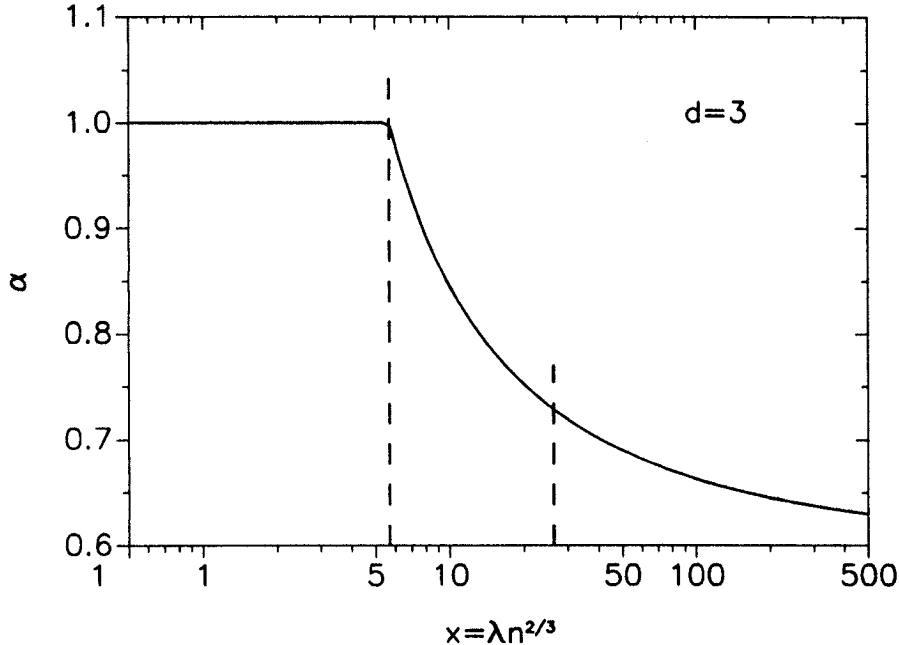


Bild V.4: Effektiver Exponent α als Funktion der Skalenvariablen x

Aus der Simulation kann man auch direkt Informationen über α gewinnen, obwohl die Überlebenswahrscheinlichkeit nur indirekt zu bestimmen ist. Im Anhang G wird

gezeigt, wie die Zustandssumme für einen $(n + 1)$ -Schritt SIW von der eines n -Schritt SIW's abhängt. Daraus ergibt sich für große n der Zusammenhang

$$-\ln(1 - cW) = \alpha K n^{\alpha-1}, \quad (V.25)$$

bzw.

$$\alpha = \frac{\partial \ln[-\ln(1 - cW)]}{\partial \ln n} + 1, \quad (V.26)$$

wobei W die über das SIW-Ensemble gemittelte Wahrscheinlichkeit ist, daß der $n + 1$ -te Schritt auf einen neuen Platz führt. Dies kann von dem Simulationsprogramm mitberechnet werden.

V.5. DIE MITTLERE QUADRATISCHE AUSLENKUNG

Es stellt sich heraus, daß die mittlere quadratische Auslenkung $\langle R^2 \rangle$ die interessanteste der untersuchten Größen ist.

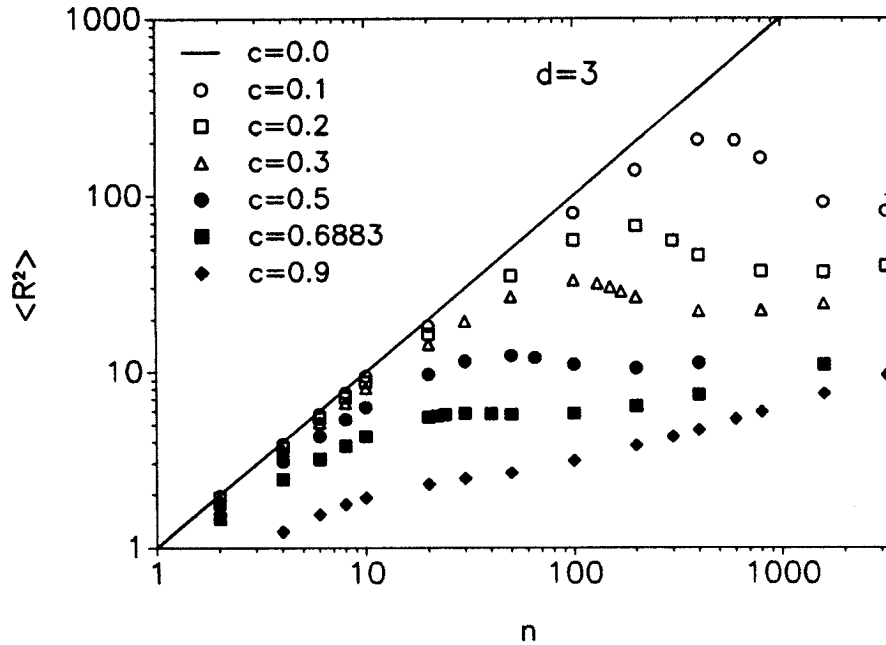


Bild V.5: Mittlere quadratische Auslenkung in $d = 3$

In einer (unskalierten) doppelt logarithmischen Auftragung, Bild V.5, kann man erkennen, daß es für kleine Haftstellen-Konzentrationen c einen Bereich gibt, in dem die mittlere quadratische Auslenkung als Funktion der Schrittzahl abnimmt. Da das zunächst ein merkwürdiges Verhalten ist, wurden für zwei ausgewählte Parameterwerte sehr genaue Simulationen mit einer anderen, völlig unabhängigen Methode durchgeführt. Sie entspricht der im Abschnitt IV.1.3 beschriebenen Methode, bei der die Verteilungsfunktion $p_n(s)$ auf einem Gitter ohne Haftstellen simuliert wird. Nach Gl. (II.6)

gilt

$$\langle R^2 \rangle = \frac{\sum_{\omega} R_{\omega}^2 e^{-\lambda s_{\omega}}}{\sum_{\omega} e^{-\lambda s_{\omega}}}. \quad (V.27)$$

Statt über alle denkbaren Walks zu summieren, wird über ein simuliertes Ensemble von *Random Walks* (keine SIW's) gemittelt.

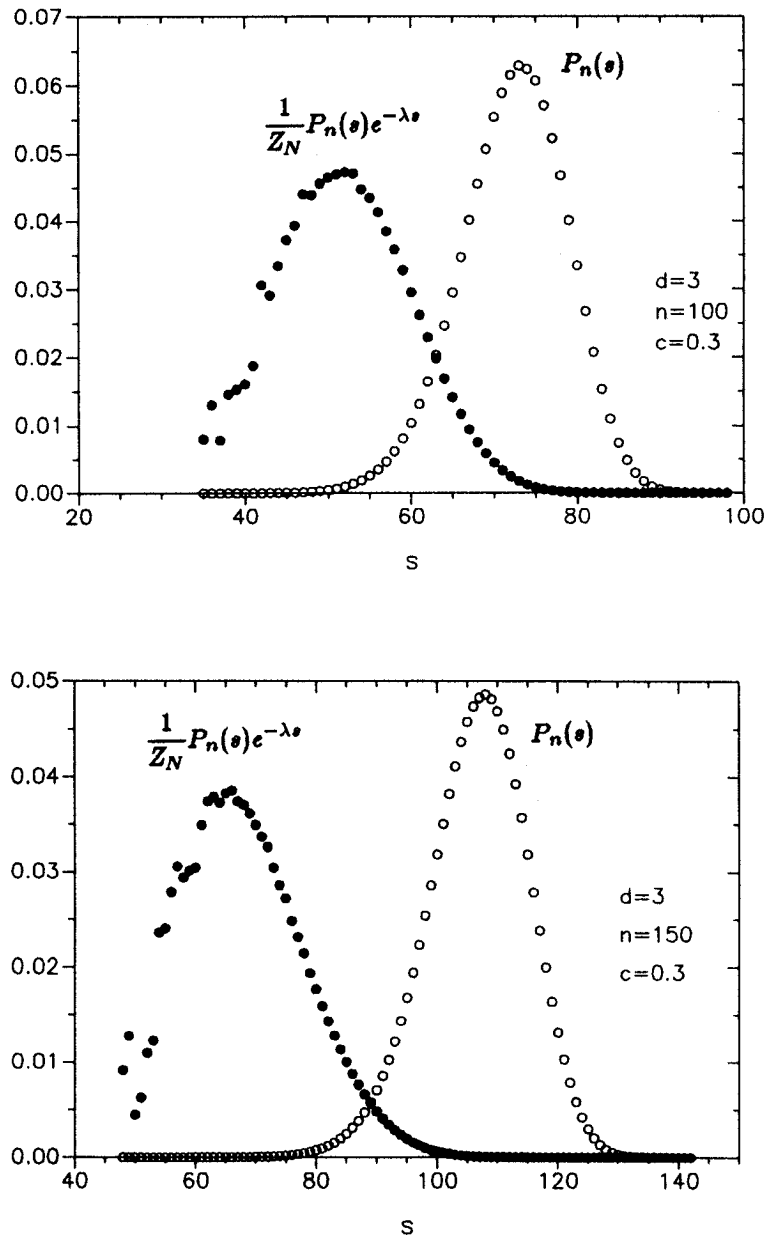


Bild V.6: Verteilungsfunktionen $p_n(s)$ für die Anzahl der besuchten Plätze bei zwei verschiedenen Schrittzahlen ($c = 0,3$; $d = 3$) o: Verteilungsfunktion; •: gewichtete, normierte Verteilungsfunktion
oben: $n = 100$; 8 Millionen Walks.
unten: $n = 150$; 240 Millionen Walks.

Der Effekt tritt an einer Stelle im Parameterraum auf, wo es mit dieser Methode gerade noch gelingt, eine qualitative Aussage zu machen. Für $c = 0,3$ und die Schrittzahlen $n = 100$ und $n = 150$ wurden zwei extensive Simulationen durchgeführt. Für die höhere Schrittzahl wurden insgesamt 240 Millionen Random Walks erzeugt. In Bild V.6 sind die Verteilungsfunktionen $p_n(s)$ für beide Parameter dargestellt. Obwohl die eigentliche Verteilungsfunktion $p_n(s)$ durch die hohe Zahl der simulierten Walks sehr glatt ist, ist die mit $e^{-\lambda s}$ gewichtete (und normierte) Funktion noch sehr unstetig, da nur Walks aus der linken Flanke der Verteilungsfunktion eine Rolle spielen. Bei größeren Schrittzahlen oder größeren Haftstellenkonzentrationen werden die Verhältnisse noch ungünstiger. Die oben angegebenen Parameter wurden nach einigen Vorversuchen so gewählt, daß sie in dem abfallenden Bereich der Kurve für $R^2(n)$ liegen, daß die gewichtete Verteilungsfunktion aber noch über ihr Maximum hinaus nach links gut dargestellt wird. (Bei ungeeigneter Wahl der Parameter zeigt die Verteilungsfunktion auf der linken Seite zu große Schwankungen und wird irgendwo abgeschnitten.)

Die Ergebnisse für die mittlere quadratische Auslenkung lauten wie folgt:

	$n = 100$	$n = 150$
modifizierte Havlin-Methode	$33,6 \pm 0,5$	$30,3 \pm 0,3$
neue Methode	$33,25 \pm 0,04$	$30,49 \pm 0,09$

Tab. V.2: Vergleich Havlin-Methode $\longleftrightarrow$ neue Methode für $c = 0,3$ in $d = 3$

Tatsächlich ergibt sich mit beiden Methoden ein Absinken der mittleren quadratischen Auslenkung mit wachsender Schrittzahl. Überdies sind die Ergebnisse innerhalb der Fehlergrenzen identisch, jedoch konnten sie mit Hilfe der neuen Methode in wesentlich kürzerer Zeit (und auch noch genauer) gewonnen werden.

Zunächst erscheint dieses Verhalten paradox. Ein Walk mit mehr Schritten entfernt sich nicht so weit vom Startpunkt, wie einer mit einer kleineren Schrittzahl. Dabei ist jedoch zu beachten, daß immer nur Walks bei der Mittelung berücksichtigt werden, die auch bis zur Schrittzahl n überlebt haben. Bei einer Vergrößerung der Schrittzahl hat man also zwei gegenläufige Tendenzen. Der zugrunde liegende *Random Walk* ist bestrebt, sich mit wachsender Schrittzahl weiter vom Ausgangsort zu entfernen (Diffusion). Da er aber mit zunehmender Schrittzahl auch mehr Plätze besucht, fällt die Wahrscheinlichkeit, die n Schritte zu überleben. Im fallenden Bereich der R_n^2 -Kurve sind die Teilchen, die die nächsten Schritte überleben, eben diejenigen, die bisher relativ wenige Plätze besucht haben und sich deshalb im Innern des besuchten Gebietes

aufhalten. Damit befinden sie sich noch in der Nähe des Startpunktes. Die Teilchen, die schon weit entfernt vom Ausgangspunkt sind, werden mit hoher Wahrscheinlichkeit in den nächsten Schritten aussterben, da sie schon viele Plätze besucht haben und sich deshalb am Rande des besuchten Bereiches aufhalten. Daher werden sie in den nächsten Schritten weitere, neue Plätze besuchen und mit relativ hoher Wahrscheinlichkeit auf eine Haftstelle stoßen. Da immer nur die überlebenden Teilchen gezählt werden, kann als Nettoeffekt eine Verringerung der mittleren quadratischen Auslenkung entstehen.

Festzuhalten ist, daß sich nicht etwa die einzelnen Teilchen mit wachsenden Schrittzahlen wieder in Richtung auf den Startplatz zu bewegen, sondern daß einfach die Teilchen, die weit entfernt sind, aussterben und nicht mehr zur Mittelung beitragen. Für eine Skalenanalyse sind wieder zwei Bereiche zu unterscheiden.

Läßt man bei festen Schrittzahlen n die Haftstellenkonzentration gegen Null gehen, merkt das Teilchen nicht, daß es sich in einem Medium mit Haftstellen befindet.

Für große Schrittzahlen n , andererseits, sind die relevanten Gebiete, in denen Teilchen überleben, Kugeln mit Radius $R(c, n)$ nach Gl. (II.17). Da diese Gebiete viel kleiner sind, als die Bereiche, die ein normaler Random Walk einnimmt, wird der Walk den zur Verfügung stehenden Platz meist voll ausnutzen und sich überdies in der Nähe des Randes aufhalten. Die mittlere quadratische Auslenkung sollte demnach in etwa dem Quadrat des Radius $R(c, n)$ entsprechen, zumindest aber mit ihm skalieren.

Deshalb sollte sich $\langle R^2 \rangle$ wie

$$\langle R^2 \rangle \sim \begin{cases} n, & \text{für } \lambda \rightarrow 0, n \text{ endlich} \\ A_d^2 \left(\frac{n}{\lambda}\right)^{\frac{2}{d+2}}, & \text{für } \lambda \text{ endlich, } n \rightarrow \infty. \end{cases} \quad (\text{V.28})$$

verhalten. Wie oben bei der Überlebenswahrscheinlichkeit ergibt sich dann aus dem Ansatz $\langle R^2 \rangle = n^\alpha f(\lambda n^\Phi)$ sofort

$$\langle R^2 \rangle = n f(\lambda n^{d/2}) \quad (\text{V.29})$$

mit

$$f(y) \sim \begin{cases} 1, & \text{für } y \rightarrow 0 \\ A_d^2 y^{\frac{2}{d+2}}, & \text{für } y \rightarrow \infty. \end{cases} \quad (\text{V.30})$$

Man beachte, daß die Skalenvariable hier nicht $x = \lambda n^{2/d}$ ist, sondern $y = \lambda n^{d/2}$. Dies hängt damit zusammen, daß mit der mittleren quadratischen Auslenkung eine neue Längenskala ins Spiel kommt, die zunächst einmal unabhängig von der Skala ist, die durch den Radius des vom Teilchen besuchten Gebietes gegeben ist.

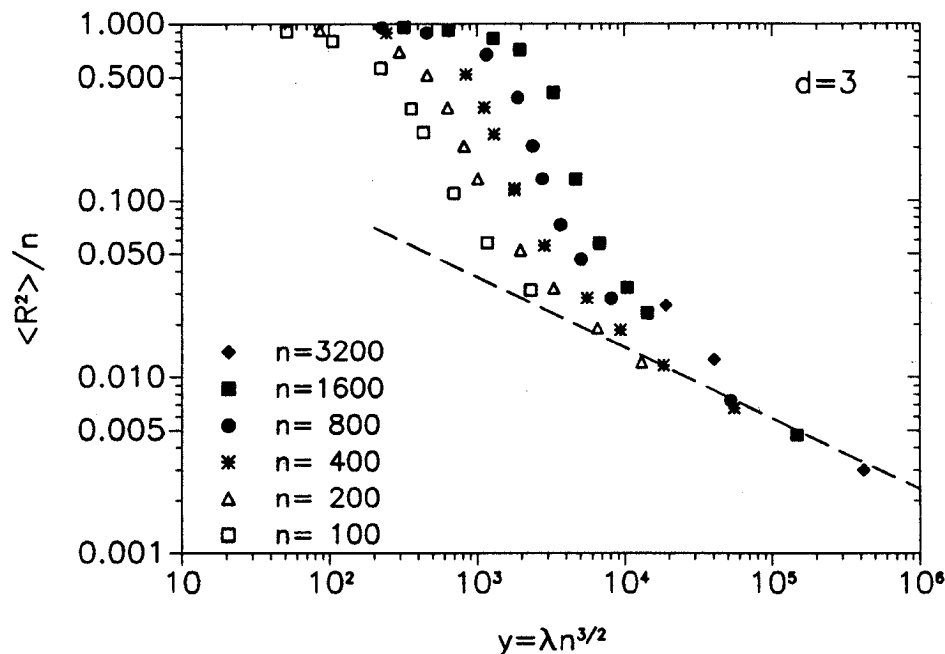


Bild V.7: Skalengplot der mittleren quadratischen Auslenkung in $d = 3$ mit SkalenvARIABLE $y = \lambda n^{d/2}$. Die gestrichelte Linie ist die asymptotische Theorie.

In Bild V.7 sind die Monte Carlo Daten entsprechend aufgetragen. Für kleine y skalen die Ergebnisse sehr gut. Dann kommt ein Übergangsbereich, in dem die Daten für verschiedene Schrittzahlen bei verschiedenen y -Werten absinken, um sich dann an der gestrichelten Linie, die das vorhergesagte, asymptotische Verhalten darstellt, wieder zu vereinigen. Die Existenz des Übergangsbereiches deutet darauf hin, daß der Crossover vom Kurzzeit- zum Langzeitverhalten nicht von der Variablen y dominiert wird. Es sieht jedoch so aus, als ob das vermutete Skalenverhalten für große y tatsächlich erfüllt ist.

Damit kann auch die in der Einleitung aufgeworfene Frage nach dem Exponenten ν in dem Gesetz

$$\langle R^2 \rangle \sim n^{2\nu} \quad (V.31)$$

beantwortet werden. Da Gl. (V.28) für große y offensichtlich von den Monte Carlo Daten erfüllt wird, gilt für große Schrittzahlen n

$$\langle R^2 \rangle \sim \lambda^{-\frac{2}{d+2}} n^{\frac{2}{d+2}}, \quad (V.32)$$

also

$$\nu = \frac{1}{d+2}. \quad (V.33)$$

V.6. DIE RÜCKKEHRWAHRSCHEINLICHKEIT

Für die Wahrscheinlichkeit, daß sich das Teilchen nach n Schritten wieder am Startplatz befindet, gelten ähnliche Aussagen, wie für die mittlere quadratische Auslenkung.

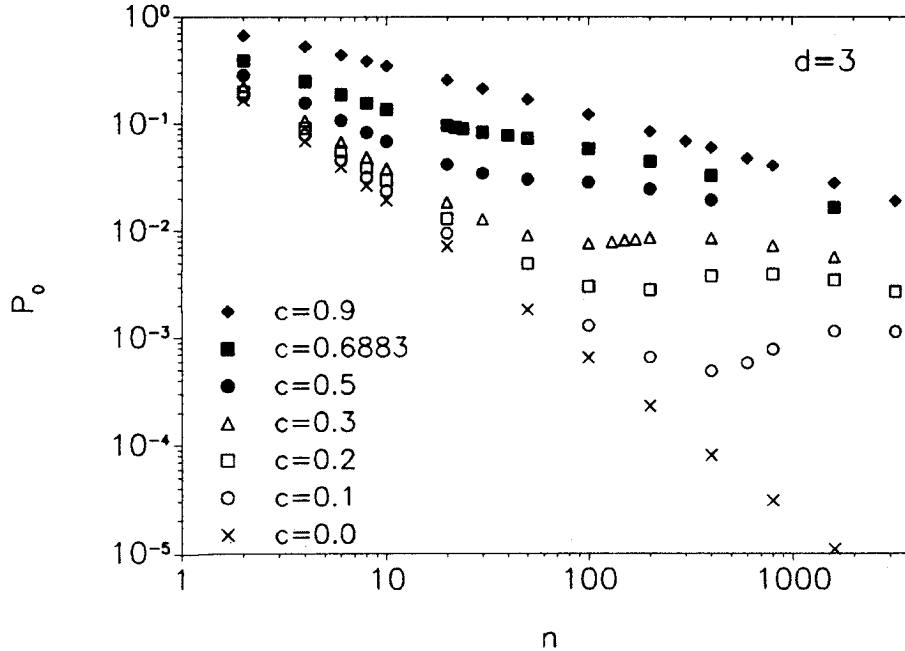


Bild V.8: Rückkehrwahrscheinlichkeit in $d = 3$. Die Werte bis einschließlich $n = 10$ entstammen exakten Abzählungen.

Auch hier gibt es einen Bereich, in dem die Wahrscheinlichkeit mit wachsender Schrittzahl überraschenderweise ansteigt. Eine Skalentheorie, die den ganzen Parameterbereich abdeckt, ist auch in diesem Fall nicht möglich.

V.7. DIE WÄRMEKAPAZITÄT

Nach Gl. (II.30) gilt für die Wärmekapazität

$$C = \lambda^2 \Delta^2 s. \quad (V.34)$$

Da in $d = 3$ für die Varianz $\Delta^2 s$ der Zahl der besuchten Plätze für $c = 0$

$$\Delta^2 s \sim n \ln n \quad (V.35)$$

gilt, ist eine Skalentheorie wie für die mittlere Zahl der besuchten Plätze hier nicht möglich.

Trägt man die Wärmekapazität $C = \lambda^2 \Delta^2 s$ gegen die inverse Temperatur λ auf, erkennt man, daß mit wachsender Kettenlänge n ein Maximum auftaucht, siehe Bild V.9.

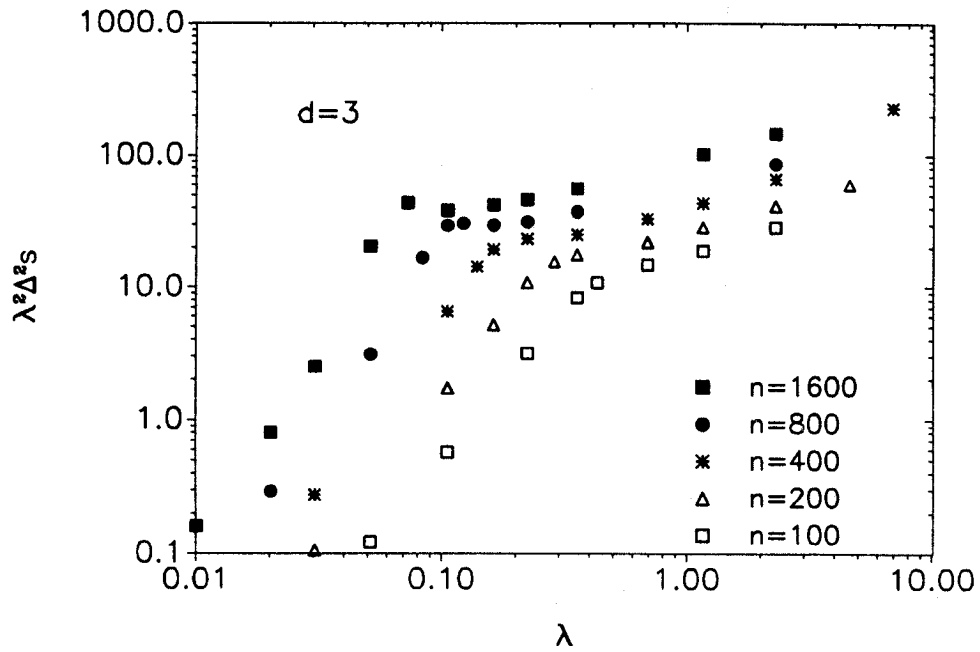


Bild V.9: Wärmekapazität C als Funktion der inversen Temperatur λ für verschiedene Kettenlängen n .

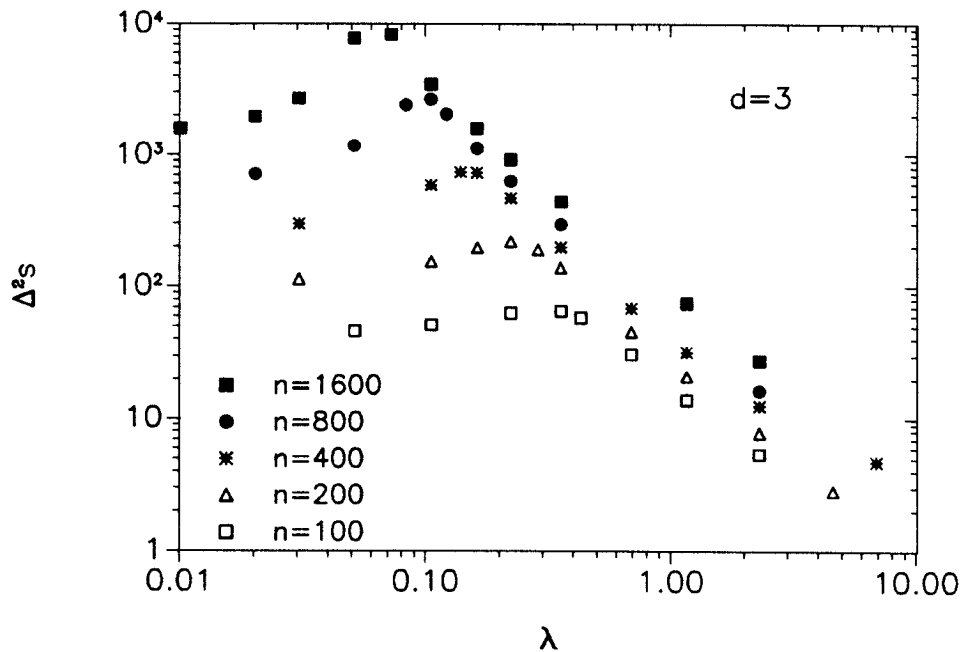


Bild V.10: Varianz der Zahl der besuchten Plätze als Funktion der inversen Temperatur λ für verschiedene Schrittzahlen n .

Noch deutlicher ist dieses Verhalten, wenn man nicht die Wärmekapazität, sondern nur die Varianz $\Delta^2 s$ aufträgt (Bild V.10). Man kann den Daten entnehmen, daß

in $d = 3$ der Wert des Maximums $\Delta^2 s_{\max}$ wie

$$\Delta^2 s_{\max} \sim n^{1,85 \pm 0,15} \quad (V.36)$$

divergiert. Für die inverse Temperatur $\lambda_{\max}$ des Maximums als Funktion der Schrittzahl n gilt

$$\lambda_{\max} \sim n^{-0,60 \pm 0,05}. \quad (V.37)$$

Ob man in $d \geq 3$ Dimensionen

$$\lambda_{\max} \sim n^{-\frac{d}{d+2}} \quad (V.38)$$

schreiben kann, müßte erst anhand von Simulationen in vier und mehr Dimensionen gezeigt werden.

Für $n \rightarrow \infty$ geht auf jeden Fall $\lambda_{\max}$ algebraisch gegen 0. Dieses Polymermodell hat damit keinen Phasenübergang bei endlichen Temperaturen, wie etwa das Modell des SAW mit attraktiver Wechselwirkung zwischen den Monomeren [45, 105, 108, 109]. Es steht damit ebenso im Gegensatz zu Clustern auf Gittern, (engl.: *lattice animals*), denen man eine attraktive Wechselwirkung aufgeprägt hat [110–113]. Auch dort findet man eine endliche Übergangstemperatur. Das gilt ebenso bei gerichteten Clustern [114].

VI. Zusammenfassung

Abschließend sollen die oben gefundenen Ergebnisse noch einmal kurz zusammengefaßt werden.

In Teil II der Arbeit wurde das Haftstellenproblem genau definiert, wobei besonderer Wert darauf gelegt wurde, die verschiedenen, auftretenden Mittelungen in der richtigen Art und Weise vorzunehmen. Es hat sich gezeigt, daß die analytisch einfacher zu behandelnde Mittelung auch die für die meisten Experimente wichtigere ist. Obwohl diese Mittelungsmethode einen "annealed"-Charakter hat, bedeutet das nicht, daß die Fluktuationen in der Haftstellenkonzentration vernachlässigt werden. Im Gegenteil, die asymptotischen Gesetzmäßigkeiten, die für große Schrittzahlen gültig sind, entstehen gerade durch das Vorhandensein relativ großer, Haftstellen-freier Gebiete.

Ferner wurde betont, daß das Haftstellenproblem äquivalent zu einem Polymermodell ist, bei dem jede Kette ein zusätzliches statistisches Gewicht bekommt, das zu einer effektiven, attraktiven Wechselwirkung führt. Diese Wechselwirkung ist nicht lokal. Alle Mittelwerte, die mit diesem zusätzlichen Gewicht berechnet werden, sind gleichzeitig die Mittelwerte der entsprechenden Größe bei einem Random Walk auf einem Gitter mit statistisch verteilten Haftstellen. Dabei dürfen aber nur die Teilchen bei dem Mittelungsprozess berücksichtigt werden, die überlebt haben.

Nur in einer Dimension sind exakte Lösungen dieser Modelle möglich. Im Teil III der Arbeit wurde das eindimensionale Haftstellenproblem asymptotisch exakt gelöst. Dabei konnten auch Korrelationen in der Sprungrichtung mitgenommen werden. Es wurde sowohl die Verteilungsfunktion für die Zahl der besuchten Plätze als auch die Überlebenswahrscheinlichkeit auf einer linearen Kette mit statistisch verteilten Haftstellen gefunden. In beiden Fällen konnte gezeigt werden, daß in führender Ordnung Skaleneigenschaften gelten. Die Korrekturterme jedoch, von denen eine ganze Reihe berechnet werden konnten, haben diese Skaleneigenschaften nicht. Ferner wurden exakte Ausdrücke sowohl für die mittlere Zahl der Schritte bis zum Einfang angegeben, als auch für die Zahl der Schritte, die ein Teilchen im Mittel benötigt, um die Zahl der besuchten Plätze zu vergrößern.

Der Fall höherer Dimensionen wurde im Teil IV mit Monte Carlo Verfahren untersucht. Dazu wurden zunächst die Schwächen und Beschränkungen der bekannten Verfahren aufgezeigt. Die daraus resultierenden Forderungen an einen neuen Algorithmus wurden dann in ein neues Verfahren umgesetzt, das sich an Simulationen aus der Polymerphysik orientiert. Es stellte sich allen anderen Verfahren zur Simulation des Haftstellenproblems als überlegen heraus. Das wurde letztlich dadurch erreicht, daß auf

die üblicherweise benutzte Bedingung des detaillierten Gleichgewichts verzichtet werden konnte. Es konnte gezeigt werden, daß das Verfahren trotzdem gegen die richtige, stationäre Verteilungsfunktion konvergiert. Diese Methode führt zu drastisch kürzeren Korrelationszeiten als ein entsprechendes Verfahren mit detailliertem Gleichgewicht. Entsprechend höhere statistische Genauigkeiten können damit erreicht werden. Speicherplatzprobleme wurden durch den Einsatz von Hash-Tabellen gelöst. Nur so konnten viele Teilchen gleichzeitig simuliert werden, wodurch das Verfahren zu einem großen Teil vektorisiert.

Im Teil V wurden dann die gefundenen Ergebnisse dargestellt und zum großen Teil mit Skalentheorien analysiert. Das funktioniert gut für drei Dimensionen, nicht dagegen im zweidimensionalen Fall, da dieser auch schon auf Gittern ohne Haftstellen marginal ist. Schon dort tauchen logarithmische Korrekturen in den asymptotischen Formeln auf. Deshalb und wegen der praktischen Bedeutung wurde hauptsächlich der dreidimensionale Fall behandelt. Es stellte sich heraus, daß der Crossover vom Kurzzeit- zum Langzeitverhalten relativ scharf verläuft, wenn man sich die mittlere Zahl der besuchten Plätze der überlebenden Teilchen ansieht. Dieser Crossover findet bei kleineren Schrittzahlen statt, als der entsprechende Crossover für die Überlebenswahrscheinlichkeit, der bisher ausschließlich in der Literatur betrachtet wurde. Der Crossoverpunkt selbst wurde als die Schrittzahl identifiziert, bei der die Zahl der besuchten Plätze auf einem Gitter ohne Haftstellen gerade so groß ist, wie die Zahl der besuchten Plätze, die asymptotisch am meisten bei gleicher Schrittzahl zur Überlebenswahrscheinlichkeit beiträgt. Bei kleineren Schrittzahlen spürt das Teilchen noch nichts von der Anwesenheit der Haftstellen, bei größeren gilt das asymptotische Verhalten. Aus dieser Betrachtung resultierte eine Skalenvariable, die sich von den Skalenvariablen anderer Autoren unterscheidet.

Ferner ergab sich eine einfache Näherung in drei und höheren Dimensionen. In ihr ersetzt man die mittlere Zahl der besuchten Plätze vor dem Crossoverpunkt durch die Rosenstock-Formel. Hinter dem Crossoverpunkt wird sie durch das asymptotische Donsker und Varadhan-Verhalten ersetzt. Berechnet man daraus die Überlebenswahrscheinlichkeit analytisch, findet man eine überraschend gute Übereinstimmung mit den Monte Carlo Daten. Die Näherung enthält einen Term mit einem Exponenten, der vorher noch nicht in der Literatur angegeben wurde. Sie ist eine einfache Fitfunktion, die die Monte Carlo Daten sehr gut beschreibt.

Die spezifische Wärme des Polymers, bzw. die Varianz in der Zahl der besuchten Plätze zeigte mit steigender Schrittzahl ein immer schärfer werdendes Maximum. Al-

lerdings führt das im Limes großer Schrittzahlen nicht auf einen Phasenübergang bei endlichen Temperaturen.

Das überraschendste Ergebnis war das der mittleren quadratischen Auslenkung. Für kleine Haftstellenkonzentrationen gibt es einen Bereich, in dem die mittlere quadratische Auslenkung mit wachsender Schrittzahl kleiner wird. Die Ursache dafür ist eine Korrelation zwischen der Überlebenswahrscheinlichkeit der nächsten Schritte und der Entfernung des Teilchens vom Startplatz. Sie führt dazu, daß Teilchen, die sich weit entfernt haben, schneller aussterben, als die, die sich noch in der Nähe des Startplatzes befinden. Da nur die überlebenden Teilchen in die Mittelung mit eingeschlossen werden, sinkt für gewisse Parameterwerte die mittlere quadratische Auslenkung.

Eine abschließende Betrachtung des dreidimensionalen Haftstellenproblems zeigte, daß auch in der Natur mit ihren typischerweise 10^{23} Gitterplätzen die notwendigen Fluktuationen in der Haftstellenkonzentration nicht mehr auftreten, die nötig sind, um das asymptotische Verhalten zu erzeugen, falls die Schrittzahl des Random Walks größer als etwa 3000 wird. Dasselbe gilt für Haftstellenkonzentrationen, die kleiner sind als 3%. In diesen Parameterbereichen sind die Proben zu klein, um die notwendigen, großen Regionen ohne Haftstellen zu enthalten. Der anomale, gestreckte Exponent, der das Donsker und Varadhan Verhalten kennzeichnet, tritt dann nicht auf. Die Überlebenswahrscheinlichkeit sollte dort einfach exponentiell zerfallen.

Anhang A. Berechnung einiger spezieller Summen

In diesem Anhang werden die Funktionen $d(z), e(z)$ und $C_{\pm}(z, r_0)$, die in Gl. (III.11) definiert wurden, in geschlossener Form berechnet. Dazu wird zunächst folgende Summe berechnet

$$A_r := \frac{1}{2s} \sum_{m=0}^{s-1} \frac{e^{ik_m r}}{a - \cos k_m}, \quad (A.1)$$

wobei r eine beliebige ganze Zahl ist, $|a| > 1$ und $k_m := 2\pi m/s$. Dann gilt offensichtlich

$$A_r = A_{s+r} = A_{-r} = A_r^* \quad (A.2)$$

und es kann im folgenden $0 \leq r < s$ angenommen werden. Führt man Exponentialfunktionen für den Cosinus ein, erhält man

$$A_r = -\frac{1}{e_1 - e_2} \left(e_1 \frac{1}{s} \sum_{m=0}^{s-1} \frac{e^{ik_m(r+1)}}{1 - e_1 e^{ik_m}} - e_2 \frac{1}{s} \sum_{m=0}^{s-1} \frac{e^{ik_m(r+1)}}{1 - e_2 e^{ik_m}} \right), \quad (A.3)$$

wobei

$$e_{1,2} := a \pm \sqrt{a^2 - 1} \quad (A.4)$$

die beiden Wurzeln der quadratischen Gleichung $x^2 - 2ax + 1 = 0$ sind. Es gilt $e_2 = 1/e_1$ und wegen $|a| > 1$ ist eine der beiden Wurzeln vom Betrage her größer, die andere kleiner als 1. Es wird daher $e_2 < 1 < e_1$ angenommen.

Dann gilt

$$\begin{aligned} \frac{1}{s} \sum_{m=0}^{s-1} \frac{e^{ik_m(r+1)}}{1 - e_2 e^{ik_m}} &= \sum_{n=0}^{\infty} e_2^n \frac{1}{s} \sum_{m=0}^{s-1} e^{ik_m(r+1+n)} \\ &= \frac{1}{s} \sum_{n=0}^{\infty} e_2^n \begin{cases} s; & \text{falls } n = ks - r - 1 \text{ mit } k = 1, 2, \dots \\ 0; & \text{sonst} \end{cases} \\ &= \frac{e_2^{s-1} - e_2^{-r}}{1 - e_2^s} = -\frac{e_1^r + 1}{1 - e_1^s}. \end{aligned} \quad (A.5)$$

Die Summation des ersten Terms erfolgt analog und man erhält schließlich

$$A_r = \frac{1}{e_1 - e_2} \left(\frac{e_2^r}{1 - e_2^s} - \frac{e_1^r}{1 - e_1^s} \right); \quad 0 \leq r < s. \quad (A.6)$$

Sei nun

$$a := \frac{1 + (p_f - p_b)z^2}{2p_f z} = \frac{f + 1 + (f - 1)z^2}{2fz} \quad (A.7)$$

und z nahe genug an 0, so daß $|a| > 1$, wie gefordert. Dann gilt schließlich wegen Gl. (III.10b)

$$d(z) = 1 - \frac{1}{s} \sum_{m=0}^{s-1} \frac{-p_f z (e^{-ik_m} - p_f z)}{1 + (p_f - p_b)z^2 - 2p_f z \cos k_m} = 1 + A_1 - p_f z A_0 \quad (A.8)$$

$$e(z) = -\frac{1}{s} \sum_{m=0}^{s-1} \frac{-p_f p_b z^2 e^{2ik_m}}{1 + (p_f - p_b)z^2 - 2p_f z \cos k_m} = p_b z A_2 \quad (A.9)$$

und ebenso

$$\begin{aligned} C_{\pm}(z, r_0) &= \\ &= \frac{A_{r_0 \mp 1} - p_f z A_{r_0} + p_b z A_{r_0 \mp 2} - p_b z [\delta_{r_0, \pm 1} (A_1 - p_f z A_0) + \delta_{r_0, \mp 1} p_b z A_2]}{2p_f z}. \end{aligned} \quad (A.10)$$

Hier ist wiederum $\delta_{r_0, -1}$ als $\delta_{r_0, s-1}$ zu lesen. Nun kann man leicht

$$\sum_{r_0=1}^{s-1} [c_+(k_m = 0, z, r_0) + c_-(k_m = 0, z, r_0)] = \frac{s-1-p_b z}{1-z} \quad (A.11)$$

und

$$\sum_{r_0=1}^{s-1} [C_+(z, r_0) + C_-(z, r_0)] = \frac{1}{1-z} + \frac{1+p_b z}{p_f z} (p_f z A_0 - A_1 - p_b z A_2). \quad (A.12)$$

berechnen. Einsetzen in Gl. (III.21) liefert

$$f(s, z) = \frac{1}{1-z} \left[s - \frac{1-p_b z^2}{(1-z)(1+A_1-p_f z A_0+p_b z A_2)} \right]. \quad (A.13)$$

Wegen

$$1 + A_1 - p_f z A_0 + p_b z A_2 = \frac{1-p_b z^2}{p_f z} [A_0 - (p_f - p_b)z A_1] \quad (A.14)$$

findet man schließlich nach der Substitution $p_f = f/(f+1)$ und $p_b = 1/(f+1)$ Gl. (III.22) im Hauptteil der Arbeit.

Anhang B. Beweis der Gl. (III.41)

Mit $e_{1,2}$ wie in Gl. (A.4) definiert und a wie in Gl. (A.7), findet man

$$\begin{aligned} a &= \frac{e_1 + e_2}{2} = \frac{\varrho e^{i\phi} + \frac{1}{\varrho} e^{-i\phi}}{2} \\ z &= \frac{fa \pm \sqrt{1 - f^2(1 - a^2)}}{f - 1}, \end{aligned} \quad (B.1)$$

wobei

$$e_1 = \varrho e^{i\phi} \quad (B.2)$$

mit $\varrho > 0$ eingeführt wurde (beachte: $e_2 = 1/e_1$). Gl. (III.36) ist auch mit e_2 an Stelle von e_1 erfüllt. Deshalb kann man die Absolutwerte als

$$|e_{1,2}^s| = \varrho^{\pm s} = \left| -f + \left(fa \pm \sqrt{1 - f^2(1 - a^2)} \right) \varrho^{\pm 1} e^{\pm i\phi} \right| \quad (B.3)$$

darstellen. Die rechte Seite der Gleichung hängt stetig von ϕ ab und hat keine Pole. Deshalb hat sie für gegebenes ϱ ein absolutes Maximum $M < \infty$ auf dem Intervall $[0, 2\pi]$.

Drei Fälle müssen unterschieden werden.

a) $\varrho > 1$ Dann gilt $\varrho^s \leq M < \infty$ bzw.

$$s \leq \frac{\ln M}{\ln \varrho} < \infty \quad (B.4)$$

muß erfüllt werden. Da das Maximum M unabhängig von s ist, verschwindet jeder einfache Pol, falls nur s groß genug ist.

b) $\varrho < 1$ Hier gilt $\frac{1}{\varrho^s} \leq M < \infty$ bzw.

$$s \leq \frac{\ln M}{-\ln \varrho} < \infty \quad (B.5)$$

muß erfüllt werden. Daher verschwindet für genügend große s auch jede Lösung von Gl. (III.36) für $\varrho < 1$. Daher bleibt nur

c) $\varrho = 1$ für die möglicherweise vorhandenen einfachen Pole, die auch für beliebig große s existieren.

Damit ist Gl. (III.41) bewiesen.

Anhang C. Asymptotische Reihen für einen speziellen Typ von Integralen

Integrale der Form

$$I_j := \int_0^\infty t^j \exp \left[-x \left(\frac{1}{2t^2} + t \right) \right] dt \quad (C.1)$$

können nach Ref. 32, Kapitel II.7 und Anhang A und B, für große x als asymptotische Reihen

$$I_j = \left(\frac{2\pi}{3x} \right)^{1/2} e^{-3x/2} \sum_{m=0}^{\infty} \frac{a_m^{(j)}}{x^m} \quad (C.2)$$

mit rationalen Koeffizienten $a_m^{(j)}$ geschrieben werden. Die Ergebnisse für $a_m^{(j)}$ finden sich in folgender Tabelle für alle j und m -Werte, die in der Arbeit benutzt wurden.

	m				
	0	1	2	3	4
j	1	17/18	205/648	-3115/34992	137305/2519424
	0	5/18	-35/648	665/34992	9625/2519424
	-1	-1/18	1/648	575/34992	-86975/2519424
	-2	-1/18	25/648	-1225/34992	89425/2519424
	-3	5/18	-35/648	665/34992	9625/2519424
	-4	17/18	-35/648	1925/34992	-175175/2519424
	-5	35/18	385/648	-5005/34992	185185/2519424
	-6	59/18	1801/648	-5005/34992	425425/2519424
	-7	89/18	5005/648	85085/34992	-1616615/2519424
	-8	125/18	11005/648	481265/34992	-1616615/2519424
	-9	167/18	21025/648	1616615/34992	37182145/2519424
	-10	215/18	36505/648	4244975/34992	240940945/2519424
	-11	269/18	59101/648	9564065/34992	929553625/2519424
	-12	329/18	90685/648	19358045/34992	2787925945/2519424
	-13	395/18	133345/648	36159515/34992	7127067745/2519424
	-14	467/18	189385/648	63430955/34992	16261125265/2519424
	-15	545/18	261325/648	105765605/34992	34029746905/2519424

Tab. I: Tabelle der $a_m^{(j)}$ für verschiedene j und m

Anhang D. Listing des Programms SIW3D

In diesem Anhang befindet sich das vollständige Listing des Programms SIW3D, in dem die im Hauptteil der Dissertation beschriebene Monte-Carlo-Simulation für Random Walks auf Gittern mit Haftstellen in drei Dimensionen realisiert ist. Das Programm ist im wesentlichen unabhängig von der zu simulierenden Dimension geschrieben. Für andere Dimensionen müssen lediglich der `PARAMETER ND`, der die Dimension angibt, und die beiden `INLINE`-Funktionen `HASH(...)` und `KEY(...)` in den Zeilen 153–156 und ihre Funktionsaufrufe in den Zeilen 479–482 geändert werden. Aus Rechenzeitgründen wurden diese Programmteile nicht in allgemeiner, dimensionsunabhängiger Form geschrieben.

Die Realisierung des Algorithmus mit Zeigern und Speicherung der Daten auf einer Halde würde eigentlich eine Implementierung in der Programmiersprache PASCAL nahelegen. Da jedoch auf der CRAY X-MP in der Kernforschungsanlage Jülich kein PASCAL-Compiler zur Verfügung stand, der vektorisierenden Code erzeugt, wurde das Programm in FORTRAN 77 geschrieben. Bis auf die Bitoperationen `AND`, `OR`, `SHIFTL` und `SHIFTR` wurde dabei ausschließlich Standard-FORTRAN benutzt. Diese Operationen sind spezielles CRAY-FORTRAN [115]. In anderen Dialekten gibt es aber entsprechende Funktionen, z. B. `IAND`, `IOR` und `ISHIFT` in VS-FORTRAN auf IBM-Mainframes [116].

Der Zufallszahlengenerator ist der R250 von Kirkpatrick und Stoll [117] nach dem Prinzip von Tausworthe [118]. Er wurde im Zentralinstitut für angewandte Mathematik (ZAM) der KFA in vektorisierendes CRAY-FORTRAN umgeschrieben [119]. Mit diesem besonders schnellen Zufallszahlengenerator sind sehr gute Erfahrungen gemacht worden, auch im Vergleich zu anderen Generatoren, die auf Supercomputern verwendet werden [120].

Je nach gewünschten Parametern der zu simulierenden SIW's sind die `PARAMETER` in den Zeilen 3–5 — siehe Programmlisting auf den folgenden Seiten — nach folgenden Regeln zu wählen:

- Zur Herabsetzung der Anzahl von Speicherbankkonflikten innerhalb der CRAY, die die Verarbeitungsgeschwindigkeit herabsetzen, ist es empfehlenswert, die folgenden `PARAMETER` nicht auf ein Vielfaches von 16 zu setzen, da die CRAY 16 Speicherbänke hat:

`NWMAX`: Maximale Zahl der zu simulierenden Walks

`NMAX`: Maximale Schrittzahl der Walks

`NCMAX`: Maximale Zahl der verschiedenen Gruppen von zu simulierenden Walks

- Wegen der besonderen Darstellung der Gitterkoordinaten in *einer* INTEGER-Zahl, darf ISHKEY nicht zu groß gewählt werden. Dieser Wert gibt die Anzahl der Bits an, um die jede Koordinate weiter nach links verschoben wird, als die vorherige, bevor alle Koordinaten addiert werden. Deshalb darf $ISHKEY \cdot (ND-1)$ nicht größer als 46 sein, da sinnvollerweise beim Übersetzen des Programms die Compiler-Option FASTMD [115] benutzt wird, die die schnellere 46-Bit Integer-Arithmetik zur Verfügung stellt. (Die Wortbreite der CRAY ist an sich 64-Bit.) Andererseits sollte $2^{**}ISHKEY$ auch nicht kleiner als $n + 1$, die maximal mögliche Ausdehnung eines SIW's, sein, da sonst die Gitterkoordinaten nicht mehr eindeutig dargestellt werden können. Beide Bedingungen werden vom Programm überprüft.
- Der Wert für NU entscheidet über die Menge des für die Hash-Tabellen benötigten Speicherplatzes. Es werden pro Walk in d Dimensionen $LHASH = 2^{d(NU+1)-1}$ Zeilen reserviert. LHASH sollte größer als die zu erwartende mittlere Anzahl der besuchten Plätze sein, damit wenig Kollisionen in der Hash-Tabelle auftreten.

ν	$k = 2^\nu$	$d = 2$ $LHASH = 2k^2$	$d = 3$ $LHASH = 4k^3$
0	1	2	4
1	2	8	32
2	4	32	256
3	8	128	2048
4	16	512	16384
5	32	2048	131072

Tab. D.1: Länge der Hash-Tabelle für $d = 2$ und $d = 3$: k gibt den halben minimalen Abstand zu äquivalenten Punkten an, LHASH ist die Länge der resultierenden Hash-Tabelle

In Tab. D.1 sind die aus ν folgenden Größen der Hash-Tabellen für $d = 2$ und $d = 3$ zusammengestellt.

- Der letzte zu wählende PARAMETER ist NNODES. Er legt die Gesamtzahl der Knoten auf der Halde fest und ist so zu wählen, daß die zu erwartende mittlere Zahl der besuchten Plätze multipliziert mit der Gesamtzahl aller Walks deutlich kleiner als NNODES ist. Es ist auch zu beachten, daß NNODES für die gewählten

Anfangsbedingungen ausreicht. Andernfalls bricht das Programm mit einer Fehlermeldung ab.

Die gewählten **PARAMETER** legen den benötigten Speicherplatz fest. Wird mehr angefordert als erlaubt oder vorhanden, muß der **PARAMETER** für die maximale Walkzahl **NWMAX** verkleinert werden. Damit kann dann auch die Anzahl der Knoten **NNODES** verringert werden. Der meiste Speicher wird für die Hash-Tabellen, für die Speicherung der Richtungen jedes einzelnen Schrittes und für die Halde benötigt. Das sind zusammen

$$\text{NWMAX} * \text{LHASH} + \text{NWMAX} * \text{NMAX} + 3 * \text{NNODES}$$

Worte. Diese Formel kann als Anhaltspunkt für den Speicherbedarf dienen. Auf den nächsten Seiten befindet sich das vollständige Programmlisting in der Version für dreidimensionale Gitter.

```

1      PROGRAM SIW3D
2      IMPLICIT INTEGER (H)
3      PARAMETER (VERS=3.8, NWMAX=193, NMAX=801)
4      PARAMETER (NCMAX=11, ND=3, NNODES=90001)
5      PARAMETER (NU=3, ISHKEY=10)
6  C *****
7  C *   SIMULATION OF RANDOM WALKS ON LATTICES IN 3 DIMENSIONS   *
8  C *           WITH STATISTICAL WEIGHT EXP(-LAMBDA*S)           *
9  C *           (SELF-INTERACTING WALKS: SIW)                     *
10 C *   USING REPTATION ALGORITHM AND HASH-CODING WITH LINKED LISTS *
11 C *           FOR CONFLICT RESOLUTION                           *
12 C *MANY WALKS ARE SIMULATED SIMULTANOUSLY TO MAKE PROGRAM VECTORIZABLE*
13 C *   -----
14 C *   NAME:                SIW3D                                *
15 C *   WRITTEN BY:         JOACHIM K. ANLAUF                    *
16 C *           IFF, THEORIE 2, ROOM 244, TEL.: 4519              *
17 C *****
18 C
19 C -----
20 C   ----- DESCRIPTION OF PARAMETERS -----
21 C   VERSION:              VERSION-NUMBER
22 C   NWMAX:                MAXIMAL NUMBER OF RANDOM WALKS
23 C   NCMAX:                MAXIMAL NUMBER OF DIFFERENT GROUPS OF WALKS
24 C   NMAX:                 MAXIMAL NUMBER OF STEPS
25 C   ND:                   DIMENSIONALITY
26 C   NNODES:               MAXIMAL NUMBER OF NODES IN NODE-LIST
27 C   LHASH:                LENGTH OF HASH-TABLES ( LH1 = LHASH-1 )
28 C   NU:                   DETERMINES LENGTH OF HASH-TABLE: LHASH=2** (ND*(NU+1)-1),
29 C                           USED FOR CALCULATION OF HASH-CODE
30 C   ISHKEY:               USED FOR CALCULATION OF KEY-CODE: 2**ISHKEY .GT. NMAX
31 C -----
32 C   ----- PHYSICAL PARAMETERS DESCRIBING SIMULATED WALKS -----
33 C   N:                     NUMBER OF STEPS OF RANDOM WALK ( .LE. NMAX )
34 C   NW:                    TOTAL NUMBER OF SIMULATED RANDOM WALKS (.LE. NWMAX)
35 C   NCW(1:NC):             NUMBER OF SIMULATED WALKS WITH LAMBDA=-LN(1-C(IC))
36 C   C(1:NC):               NCW(IC) WALKS HAVE TRAP CONCENTRATION C(IC)
37 C   ICONF(1:NC):           INITIAL CONFIGURATION OF WALKS
38 C                           1: MOST COMPACT
39 C                           2: RANDOM
40 C                           3: MOST EXTENDED
41 C   P(IW):                 INTERACTION PARAMETER FOR WALK IW (P = 1-C)
42 C -----
43 C   ----- GENERAL PURPOSE VARIABLES -----
44 C   IW:                    POINTER TO ACTUAL WALK (1..NW)
45 C   IDIR(ID,0:2*ND-1):     DIRECTION-VECTOR
46 C   NINT:                  NUMBER OF INTERVALS FOR PRINTING PARTIAL RESULTS
47 C   NEQUI:                 NUMBER OF MC-STEPS TO REACH EQUILIBRIUM
48 C   NMCOUT:                NUMBER OF MC-STEPS BETWEEN OUTPUTS OF PARTIAL RESULTS
49 C   NMC:                   ACTUAL MC-STEP NUMBER FOR CURRENT INTERVALL
50 C   NMCSUM:                TOTAL NUMBER OF SIMULATED MC-STEPS AFTER EQUILIBRATION
51 C   DEBUG:                 LOGICAL VARIABLE:
52 C                           IF TRUE, CONSISTENCY CHECK IS MADE AT END OF PROGRAM
53 C   ISEED:                 SEED FOR RANDOM NUMBER GENERATOR (ODD NUMBER!)
54 C   RND(IW):               RANDOM NUMBERS FOR EACH WALK
55 C   MZ(250):               WORK-VECTOR, USED TO GENERATE RANDOM NUMBERS
56 C   IWORK(IW):             WORK-VECTOR, USED TO VECTORIZE A LOOP
57 C -----
58 C   ----- VARIABLES FOR CORRELATION TIME MEASUREMENT -----
59 C   MOSTL(IW):             LEFTMOST POSITION OF STEP-1-POINTER LP(IW)
60 C   MOSTR(IW):             RIGHTMOST POSITION
61 C   NTAU(IW):              NUMBER OF STATISTICALLY INDEPENDENT WALKS
62 C
63 C   THE RESULTS <TAU> ARE NOT PARTIAL RESULTS. THEY ARE
64 C   ACCUMULATED DURING THE WHOLE SIMULATION

```

```

65 C -----
66 C ----- VARIABLES DESCRIBING CURRENT STATUS OF WALK IW -----
67 C IRO(IW,ID): SITE-VECTOR OF TAIL
68 C IRN(IW,ID): SITE-VECTOR OF HEAD
69 C ISUMR(IW,ID): SUM OF ALL SITE-VECTORS OF WALK IW
70 C LD(IW,1:N): LIST OF DIRECTIONS TO EACH STEP (CIRCULAR LIST)
71 C LP(IW): POINTER TO STEP 1 IN CIRCULAR LIST
72 C ICDIR(IW): CURRENT DIRECTION OF WALK IW:
73 C +1: POINTER TO CIRCULAR LIST MOVING TO THE RIGHT
74 C -1: POINTER TO LIST MOVING TO THE LEFT
75 C NS(IW): NUMBER OF DISTINCT SITES OF WALK IW
76 C NR2(IW): SQUARE END TO END DISTANCE OF WALK IW
77 C NSIG2(IW): SUM OF SQUARES OF COORDINATES OF WALK IW
78 C -----
79 C ----- VARIABLES USED FOR SUMMATION OF QUANTITIES TO AVERAGE -----
80 C NSUMS(IW): SUM OF NUMBER OF DISTINCT SITES DURING SIMULATION
81 C NSUMS2(IW): SUM OF SQUARE OF NUMBER OF DISTINCT SITES
82 C NSUMS3(IW): SUM OF THIRD POWER OF NUMBER OF DISTINCT SITES
83 C NSUMS4(IW): SUM OF FOURTH POWER OF NUMBER OF DISTINCT SITES
84 C NSUMR2(IW): SUM OF SQUARE END-TO-END-DISTANCE DURING SIMULATION
85 C NSUMR4(IW): SUM OF FORTH POWER OF END-TO-END-DISTANCE
86 C SUMG2(IW): (N+1) * SUM OF SQUARE RADIUS OF GYRATION
87 C SUMG4(IW): (N+1)**2 * SUM OF FORTH POWER OF RADIUS OF GYRATION
88 C SUMG2, SUMG4 ARE REAL TO AVOID OVERFLOWS
89 C NSUMP0(IW): SUM OF PROBABILITY FOR RETURN TO THE ORIGIN
90 C NSUMEX(IW): NUMBER OF TRYs TO VISIT ONE MORE SITE ("TO EXTEND")
91 C NSUMRE(IW): NUMBER OF REJECTED WALKS
92 C -----
93 C ----- VARIABLES USED FOR GENERATING A STEP -----
94 C LDNEW(IW): DIRECTION FOR NEW STEP
95 C IDIR1(IW,ID): DIRECTION-VECTOR FOR NEW STEP
96 C IRNEW(IW,ID): SITE-VECTOR TO GO TO
97 C IROLD(IW,ID): SITE-VECTOR TO LEAVE
98 C HNEW(IW): HASH CODE OF SITE TO GO
99 C HOLD(IW): HASH CODE OF SITE TO LEAVE
100 C KNEW(IW): KEY CODE OF SITE TO GO
101 C KOLD(IW): KEY CODE OF SITE TO LEAVE
102 C IRM2(IW): SQUARE END-TO-END DISTANCE (USED TO VECTORIZE)
103 C DECIS(IW): WILL BE .TRUE. WITH PROBABILITY P(IW)
104 C NEEDED FOR STEP DECISION
105 C -----
106 C ----- VARIABLES RELATED TO HASHING ALGORITHM -----
107 C LHEAD: LENGTH OF HEADER-TABLE
108 C HEADER(0:LH1,IW):
109 C HEADER OF LINKED LISTS
110 C ZERO DENOTES EMPTY LIST
111 C POSITIVE VALUES DENOTE POINTER TO HEAD OF LIST
112 C LBEGIN: POINTER TO FIRST EMPTY NODE ON NODE-LIST
113 C ISITE(1:NNODES):
114 C KEY-FIELD OF NODE-LIST, DENOTES LATTICE SITE
115 C IVISIT(1:NNODES):
116 C VALUE-FIELD OF NODE-LIST,
117 C DENOTES NUMBER OF VISITS TO SITE ISITE(...)
118 C -1: NODE HAS NOT BEEN USED
119 C -2: USED IN CONSISTENCY CHECK TO TEST FOR DOUBLE-USAGE
120 C LINK(1:NNODES):
121 C LINK-FIELD OF NODE-LIST.
122 C ZERO DENOTES END OF LIST
123 C POSITIVE VALUE DENOTES LINK TO NEXT NODE
124 C -----
125 C PARAMETER (LHASH=2**(ND*(NU+1)-1), MASK=2*2**NU-1, NU1=NU+1)
126 C PARAMETER (LH1=LHASH-1, NHMAX=NWMAX*LHASH)
127 C PARAMETER (MSKKEY=2**ISHKEY-1, ISHKE2=2*ISHKEY)
128 C PARAMETER (ND2=2*ND, ND4=4*ND, ND21=ND2-1)

```

```

129 C -----
130     LOGICAL DEBUG
131     LOGICAL DECIS(NWMAX)
132     DIMENSION HEADER(NHMAX)
133     DIMENSION HNEW(NWMAX),HOLD(NWMAX),KNEW(NWMAX),KOLD(NWMAX)
134     DIMENSION ICDIR(NWMAX),LDNEW(NWMAX)
135     DIMENSION IDIR(ND,0:ND21)
136     DIMENSION IR0(NWMAX,ND),IRN(NWMAX,ND),ISUMR(NWMAX,ND)
137     DIMENSION IRNEW(NWMAX,ND),IRM2(NWMAX)
138     DIMENSION ISITE(NNODES),IVISIT(NNODES),LINK(NNODES)
139     DIMENSION IWORK(NWMAX),IDIR1(NWMAX,ND)
140     DIMENSION JSUMR(ND)
141     DIMENSION LP(NWMAX),LD(NWMAX,NMAX)
142     DIMENSION MOSTL(NWMAX),MOSTR(NWMAX),NTAU(NWMAX)
143     DIMENSION MZ(250),RND(NWMAX),RND2(NMAX)
144     DIMENSION NCW(NCMAX),C(NCMAX),ICONF(NCMAX)
145     DIMENSION NSUMPO(NWMAX),NSUMEX(NWMAX),NSUMRE(NWMAX)
146     DIMENSION NSUMR2(NWMAX),NSUMR4(NWMAX)
147     DIMENSION NSUMS(NWMAX),NSUMS2(NWMAX),NSUMS3(NWMAX),NSUMS4(NWMAX)
148     DIMENSION NS(NWMAX),NR2(NWMAX),NSIG2(NWMAX)
149     DIMENSION P(NWMAX)
150     DIMENSION SUMG2(NWMAX),SUMG4(NWMAX)
151 C -----
152 C --- INLINE FUNCTIONS:
153     HASH(IX,IY,IZ)=OR(OR(SHIFTL(AND(IY-IX,MASK),NU),
154     * SHIFTR(AND(IX+IY,MASK),1)),SHIFTL(AND(IZ,MASK),NU1))*NW+IW
155     KEY(IX,IY,IZ)=OR(OR(SHIFTL(AND(IX,MSKKEY),ISHKEY),
156     * AND(IY,MSKKEY)),SHIFTL(AND(IZ,MSKKEY),ISHKE2))
157 C -----
158 C --- INPUT OF PARAMETERS
159     READ(5, '(I10)') DEBUG
160     READ(5, '(I10)') N
161     READ(5, '(I10)') ISEED
162     READ(5, '(I10)') NINT
163     READ(5, '(I10)') NEQUI
164     READ(5, '(I10)') NMCOUT
165     READ(5, '(I10)') NC
166     READ(5, '(I10,E10.1,I10)') (NCW(IC),C(IC),ICONF(IC),IC=1,NC)
167 C -----
168 C --- INIT. RANDOM NUMBER GENERATOR (ISEED HAS TWO BE ODD)
169     IF ((ISEED/2)*2 .EQ. ISEED) ISEED=ISEED+1
170     CALL INR250(MZ,ISEED)
171 C -----
172 C --- CALCULATE NW
173     NW=0
174     DO 6 IC=1,NC
175         NW=NW+NCW(IC)
176     6 CONTINUE
177 C -----
178 C --- CALCULATE SOME RELEVANT CONSTANTS
179     NH=NW*LHASH
180     FLN1I=1./FLOAT(N+1)
181 C -----
182 C --- OUTPUT OF PARAMETERS
183     WRITE(6, '(1X, ''SIW3D VERSION '',F4.1)') VERS
184     WRITE(6, '(1X, ''ND     ='',I10)') ND
185     WRITE(6, '(1X, ''N     ='',I10)') N
186     WRITE(6, '(1X, ''ISEED ='',I10)') ISEED
187     WRITE(6, '(1X, ''NW    ='',I10)') NW
188     WRITE(6, '(1X, ''LHASH ='',I10)') LHASH
189     WRITE(6, '(1X, ''NNODES='',I10)') NNODES
190     WRITE(6, '(1X, ''NC    ='',I10)') NC
191     WRITE(6, '(1X, ''NINT  ='',I10)') NINT
192     WRITE(6, '(1X, ''NEQUI ='',I10)') NEQUI

```

```

193     WRITE(6, '(1X, ''NMCOUT ='', I10)') NMCOUT
194     WRITE(6, '(1X, ''DEBUG ='', L10)') DEBUG
195     WRITE(6, '(1X, '' IC      NUMBER      C      ICONF'',
196     *      ''      1: COMPACT, 2: RANDOM, 3: EXTENDED''')
197     WRITE(6, 600) (IC, NCW(IC), C(IC), ICONF(IC), IC=1, NC)
198     600 FORMAT(1X, I3, I10, F10.6, I6)
199     WRITE(6, '(1X, 50(1H-))')
200 C -----
201 C --- CHECK MAXIMAL VALUES
202     IF (N .GT. NMAX) THEN
203         WRITE(6, '(1X, ''N LARGER THAN NMAX''')
204         STOP
205     ENDIF
206     IF (NC .GT. NCMAX) THEN
207         WRITE(6, '(1X, ''NC LARGER THAN NCMAX''')
208         STOP
209     ENDIF
210 C -----
211 C --- CALCULATION OF KEY-CODES CORRECT?
212 C --- SINGLE VALUED REPRESENTATION?
213     IF (2**ISHKEY .LE. N) THEN
214         WRITE(6, '(1X, ''2**ISHKEY IS NOT GREATER THAN N''')
215         STOP
216     ENDIF
217 C --- SHORT INTEGERS? (BECAUSE OF FASTMD OPTION)
218     IF (ISHKEY*(ND-1) .GT. 48) THEN
219         WRITE(6, '(1X, ''ISHKEY*(ND-1) IS GREATER THAN 48''')
220         STOP
221     ENDIF
222 C -----
223     WRITE(6, '(1X, ''STRUCTURE OF MC-DATA:''')
224     WRITE(6, '(1X, 50(1H-))')
225     WRITE(6, '(1X, ''      IINT      NMCSUM      NMC''')
226     WRITE(6, '(1X, '' IC      <S>      <S**2>      <S**3>''')
227     *      ''      <S**4>      <R**2>      <R**4>      <G**2>''')
228     *      ''      <G**4>      P0      PEX      PRE      <TAU>''')
229     WRITE(6, '(1X, 50(1H-))')
230 C -----
231 C --- PREPARE INTERACTION PARAMETERS FOR DIFFERENT WALKS
232 C      AND CALCULATE NUMBER OF SIMULATED WALKS
233     KW1=1
234     DO 10 IC=1, NC
235         KW2=KW1+NCW(IC)-1
236         DO 12 IW=KW1, KW2
237             P(IW)=1.0-C(IC)
238         12 CONTINUE
239         KW1=KW2+1
240     10 CONTINUE
241     IF (NW .GT. NWMAX) THEN
242         WRITE(6, 605) NW, NWMAX
243     605 FORMAT(1X, 'NUMBER OF WALKS (' , I5, ') EXCEEDS NWMAX =', I5)
244     STOP
245     ENDIF
246 C -----
247 C --- INIT. DIRECTION VECTOR (NEAREST NEIGHBOR JUMPS)
248     DO 20 I=1, ND
249         DO 22 ID=1, ND
250             J=0
251             IF (I .EQ. ID) J=1
252             IDIR(ID, 2*I-2)=J
253             IDIR(ID, 2*I-1)=-J
254         22 CONTINUE
255     20 CONTINUE
256 C -----

```

```

257 C --- INITIALIZE RANDOM WALKS
258 C --- POSITION OF STEP 0
259     DO 30 ID=1,ND
260         DO 32 IW=1,NW
261             IR0(IW,ID)=0
262             IRN(IW,ID)=0
263     32 CONTINUE
264     30 CONTINUE
265 C -----
266 C --- LIST OF DIRECTIONS
267     KW1=1
268     DO 40 IC=1,NC
269         KW2=KW1+NCW(IC)-1
270         IF (ICONF(IC) .EQ. 1) THEN
271 C --- MOST COMPACT CONFIGURATION, TOTALLY FOLDED: (LD: 0,1,0,1,...)
272             J=1
273             DO 42 K=1,N
274                 J=1-J
275                 DO 43 IW=KW1,KW2
276                     LD(IW,K)=J
277             43 CONTINUE
278             42 CONTINUE
279 C -----
280             ELSEIF (ICONF(IC) .EQ. 2) THEN
281 C --- RANDOM CONFIGURATION, NORMAL RANDOM WALK
282                 DO 52 IW=KW1,KW2
283                     CALL R250(MZ,RND2,N)
284                     DO 53 K=1,N
285                         LD(IW,K)=RND2(K)*ND2
286                 53 CONTINUE
287                 52 CONTINUE
288 C -----
289             ELSEIF (ICONF(IC) .EQ. 3) THEN
290 C --- MOST EXTENDED CONFIGURATION, TOTALLY STRETCHED (LD: 0,0,0,...)
291                 DO 57 K=1,N
292                     DO 58 IW=KW1,KW2
293                         LD(IW,K)=0
294                 58 CONTINUE
295                 57 CONTINUE
296             ELSE
297                 WRITE(6,612) IC,ICONF(IC)
298             612     FORMAT(1X,'UNDEFINED INITIAL CONFIGURATION: ICONF(',
299             *         I4,') =',I2)
300             STOP
301 C -----
302             ENDIF
303             KW1=KW2+1
304             40 CONTINUE
305 C -----
306 C --- CLEAR HASH-TABLES
307     DO 60 IH=1,NH
308         HEADER(IH)=0
309     60 CONTINUE
310 C -----
311 C --- CLEAR NODE-LIST AND LINK ALL NODES
312     DO 64 NO=1,NNODES
313         ISITE(NO)=-1
314         IVISIT(NO)=-1
315         LINK(NO)=NO+1
316     64 CONTINUE
317 C --- LAST NODE POINTS TO 0, (END OF LIST)-POINTER
318     LINK(NNODES)=0
319     LBEGIN=1
320 C -----

```

```

321 C --- POINTER TO STEP NUMBER 1 AND CURRENT DIRECTION
322     DO 70 IW=1,NW
323         LP(IW)=1
324         ICDIR(IW)=+1
325     70 CONTINUE
326 C -----
327 C --- INIT. VECTORS POINTING TO STEP N AND SUM OF ALL SITE-VECTORS
328 C     GET DATA FROM VECTOR TO STEP 0 AND STORED DIRECTIONS
329 C --- ENTER STEP 0 INTO SUM-VARIABLES
330     DO 72 ID=1,ND
331         DO 74 IW=1,NW
332             ISUMR(IW,ID)=IR0(IW,ID)
333             NSIG2(IW)=IR0(IW,ID)**2
334         74 CONTINUE
335     72 CONTINUE
336 C -----
337 C --- HANDLE HASH TABLE AND NUMBER OF DISTINCT SITES FOR STEP 0
338     DO 76 IW=1,NW
339         HOLD(IW) = HASH(IR0(IW,1),IR0(IW,2),IR0(IW,3))
340         KOLD(IW) = KEY(IR0(IW,1),IR0(IW,2),IR0(IW,3))
341     76 CONTINUE
342     DO 78 IW=1,NW
343 C --- SITE NOT YET OCCUPIED, ADD NEW NODE TO NODE-LIST
344         LB=LINK(LBEGIN)
345         IF (LB .EQ. 0) THEN
346             WRITE(6, '(1X, ''NODE-LIST FULL'')')
347             STOP
348         ENDIF
349         LINK(LBEGIN)=HEADER(HOLD(IW))
350         ISITE(LBEGIN)=KOLD(IW)
351         IVISIT(LBEGIN)=1
352         HEADER(HOLD(IW))=LBEGIN
353         LBEGIN=LB
354         NS(IW)=1
355     78 CONTINUE
356 C -----
357 C --- HANDLE EACH STEP NUMBER
358     DO 80 K=1,N
359         DO 82 ID=1,ND
360             DO 84 IW=1,NW
361                 IRN(IW,ID)=IRN(IW,ID)+IDIR(ID,LD(IW,K))
362             84 CONTINUE
363         82 CONTINUE
364 C -----
365 C --- IRN(1:IW,1:ID) ARE POSITIONS OF WALKS AFTER STEP K
366 C --- HANDLE HASH-TABLE AND NUMBER OF DISTINCT SITES VISITED
367     DO 86 IW=1,NW
368         HNEW(IW) = HASH(IRN(IW,1),IRN(IW,2),IRN(IW,3))
369         KNEW(IW) = KEY(IRN(IW,1),IRN(IW,2),IRN(IW,3))
370     86 CONTINUE
371     DO 88 IW=1,NW
372         LNEW=HEADER(HNEW(IW))
373     90 CONTINUE
374         IF (LNEW .EQ. 0) GOTO 92
375         IF (ISITE(LNEW) .EQ. KNEW(IW)) GOTO 92
376         LNEW=LINK(LNEW)
377     GOTO 90
378     92 CONTINUE
379     IF (LNEW .EQ. 0) THEN
380 C --- SITE NOT YET OCCUPIED, ADD NEW NODE TO NODE-LIST
381         LB=LINK(LBEGIN)
382         IF (LB .EQ. 0) THEN
383             WRITE(6, '(1X, ''NODE-LIST FULL'')')
384             STOP

```

```

385             ENDIF
386             LINK(LBEGIN)=HEADER(HNEW(IW))
387             ISITE(LBEGIN)=KNEW(IW)
388             IVISIT(LBEGIN)=1
389             HEADER(HNEW(IW))=LBEGIN
390             LBEGIN=LB
391             NS(IW)=NS(IW)+1
392         ELSE
393 C ---             SITE ALREADY OCCUPIED, INCREMENT NUMBER OF VISITS
394             IVISIT(LNEW)=IVISIT(LNEW)+1
395         ENDIF
396     88     CONTINUE
397 C -----
398         DO 100 ID=1,ND
399             DO 102 IW=1,NW
400                 ISUMR(IW,ID)=ISUMR(IW,ID)+IRN(IW,ID)
401                 NSIG2(IW)=NSIG2(IW)+IRN(IW,ID)**2
402             102     CONTINUE
403         100     CONTINUE
404     80     CONTINUE
405 C -----
406 C --- SQUARE END-TO-END DISTANCE
407         DO 110 IW=1,NW
408             NR2(IW)=(IRN(IW,1)-IRO(IW,1))**2
409         110     CONTINUE
410         DO 112 ID=2,ND
411             DO 114 IW=1,NW
412                 NR2(IW)=NR2(IW)+(IRN(IW,ID)-IRO(IW,ID))**2
413             114     CONTINUE
414         112     CONTINUE
415 C =====
416 C             INITIALIZATION DONE, START SIMULATION
417 C -----
418 C --- LOOP FOR PRINTING PARTIAL RESULTS
419 C --- FIRST EQUILIBRATE, AT THE END OF LOOP 200 SWITCH TO NMC=NMCOUT
420         IF (NEQUI.EQ. 0) THEN
421             NMC=NMCOUT
422             MINT1=1
423         ELSE
424             NMC=NEQUI
425             MINT1=0
426         ENDIF
427         DO 200 IINT=MINT1,NINT
428 C -----
429 C --- CLEAR SUMS
430         DO 205 IW=1,NW
431             NSUMS(IW)=0
432             NSUMS2(IW)=0
433             NSUMS3(IW)=0
434             NSUMS4(IW)=0
435             NSUMR2(IW)=0
436             NSUMR4(IW)=0
437             SUMG2(IW)=0.
438             SUMG4(IW)=0.
439             NSUMP0(IW)=0
440             NSUMEX(IW)=0
441             NSUMRE(IW)=0
442         205     CONTINUE
443         IF (IINT.LE. 1) THEN
444             NMCSUM=0
445             DO 207 IW= 1,NW
446                 MOSTL(IW)=LP(IW)
447                 MOSTR(IW)=LP(IW)
448                 NTAU(IW)=0

```



```

449      207      CONTINUE
450      ENDIF
451 C -----
452 C              MAIN MONTE CARLO LOOP
453 C -----
454      DO 210 IMC=1,NMC
455 C ---      CALCULATE DIRECTION OF MOVEMENT
456      CALL R250(MZ,RND,NW)
457      DO 220 IW=1,NW
458      LDNEW(IW)=ND2*RND(IW)
459      220      CONTINUE
460 C ---      CALCULATE RANDOM NUMBERS FOR STEP-DECISION
461 C      (IMPORTANCE SAMPLING)
462      CALL R250(MZ,RND,NW)
463      DO 222 IW=1,NW
464      DECIS(IW)=RND(IW) .LT. P(IW)
465      222      CONTINUE
466 C -----
467 C ---      CALCULATE SITE TO GO AND SITE TO LEAVE
468      DO 230 ID=1,ND
469      DO 231 IW=1,NW
470      IWORK(IW)=IDIR(ID,LDNEW(IW))
471      231      CONTINUE
472      DO 232 IW=1,NW
473      IRNEW(IW,ID)=IRN(IW,ID)+ICDIR(IW)*IWORK(IW)
474      232      CONTINUE
475      230      CONTINUE
476 C -----
477 C ---      CALCULATE HASH- AND KEY-CODES
478      DO 240 IW=1,NW
479      HNEW(IW) = HASH(IRNEW(IW,1),IRNEW(IW,2),IRNEW(IW,3))
480      HOLD(IW) = HASH(IR0(IW,1),IR0(IW,2),IR0(IW,3))
481      KNEW(IW) = KEY(IRNEW(IW,1),IRNEW(IW,2),IRNEW(IW,3))
482      KOLD(IW) = KEY(IR0(IW,1),IR0(IW,2),IR0(IW,3))
483      240      CONTINUE
484 C -----
485 C ---      THE FOLLOWING CODE IS NOT VECTORIZABLE
486 C      =====
487      DO 250 IW=1,NW
488 C -----
489 C ---      SEARCH FOR NODE 'LNEW' OF HEAD-DESTINATION SITE
490      LNEW=HEADER(HNEW(IW))
491      260      CONTINUE
492      IF (LNEW .EQ. 0) GOTO 270
493      IF (ISITE(LNEW) .EQ. KNEW(IW)) GOTO 270
494      LNEW=LINK(LNEW)
495      GOTO 260
496 C ---      SEARCH FOR NODE 'LOLD' OF TAIL-LEAVE-SITE
497 C      LLAST WILL BE A POINTER TO THE NODE WHICH IS POINTING TO
498 C      LOLD (WILL BE ZERO IF HEADER IS POINTING TO LOLD)
499      270      CONTINUE
500      LOLD=HEADER(HOLD(IW))
501      LLAST=0
502      280      CONTINUE
503      IF (ISITE(LOLD) .EQ. KOLD(IW)) GOTO 290
504      LLAST=LOLD
505      LOLD=LINK(LOLD)
506      GOTO 280
507 C -----
508 C ---      NOW SEE, WHETHER S IS INCREASING OR NOT
509      290      CONTINUE
510      IF ((LNEW .GT. 0) .OR. (IVISIT(LOLD) .EQ. 1)) GOTO 300
511 C ---      S WOULD INCREASE, DECIDE ON JUMP
512      NSUMEX(IW)=NSUMEX(IW)+1

```

```

513          IF (DECIS(IW)) GOTO 300
514 C ---- JUMP REJECTED, TURN DIRECTION OF MOVEMENT AND TAKE
515 C      OLD WALK AGAIN
516      NSUMRE(IW)=NSUMRE(IW)+1
517      ICDIR(IW)=-ICDIR(IW)
518 C -----
519      DO 292 ID=1,ND
520          IRNEW(IW,ID)=IRN(IW,ID)
521          IRN(IW,ID)=IRO(IW,ID)
522          IRO(IW,ID)=IRNEW(IW,ID)
523          IDIR1(IW,ID)=0
524      292 CONTINUE
525 C -----
526          GOTO 380
527 C -----
528 C --- DO THE JUMP
529      300 CONTINUE
530      IF (LNEW .EQ. 0) THEN
531 C --- ADD NEW NODE TO NODE-LIST
532      LB=LINK(LBEGIN)
533      IF (LB .EQ. 0) THEN
534          WRITE (6, '(1X, 'ERROR: NODE-LIST FULL' )')
535          STOP
536      ENDIF
537      LINK(LBEGIN)=HEADER(HNEW(IW))
538      ISITE(LBEGIN)=KNEW(IW)
539      IVISIT(LBEGIN)=1
540      HEADER(HNEW(IW))=LBEGIN
541 C --- IF SAME CONFLICT-CHAIN AND LNEW IS FIRST IN CHAIN
542 C      THEN ADJUST LLAST
543      IF (HOLD(IW) .EQ. HNEW(IW) .AND. LLAST .EQ. 0) LLAST=LBEGIN
544      LBEGIN=LB
545      NS(IW)=NS(IW)+1
546      ELSE
547 C --- INCREMENT NUMBER OF VISITS AT HEAD-DESTINATION-SITE
548      IVISIT(LNEW)=IVISIT(LNEW)+1
549      ENDIF
550      IF (IVISIT(LOLD) .EQ. 1) THEN
551 C --- REMOVE OLD NODE
552      IF (LLAST .EQ. 0) THEN
553          HEADER(HOLD(IW))=LINK(LOLD)
554      ELSE
555          LINK(LLAST)=LINK(LOLD)
556      ENDIF
557      LINK(LOLD)=LBEGIN
558      LBEGIN=LOLD
559      NS(IW)=NS(IW)-1
560      ELSE
561 C --- DECREMENT NUMBER OF VISITS AT TAIL-LEAVING-SITE
562      IVISIT(LOLD)=IVISIT(LOLD)-1
563      ENDIF
564 C --- UPDATE STATUS OF WALK
565      IF (ICDIR(IW) .GT. 0) THEN
566          DO 310 ID=1,ND
567              IDIR1(IW,ID)=IDIR(ID,LD(IW,LP(IW)))
568      310 CONTINUE
569          LD(IW,LP(IW))=LDNEW(IW)
570          IF (LP(IW) .EQ. MOSTR(IW)) THEN
571              MOSTR(IW)=MOSTR(IW)+1
572              IF (MOSTR(IW) .GT. N) MOSTR(IW)=1
573              IF (MOSTR(IW) .EQ. MOSTL(IW)) THEN
574                  MOSTL(IW)=MOSTR(IW)
575                  NTAU(IW)=NTAU(IW)+1
576              ENDIF

```

```

577             ENDIF
578             LP(IW)=LP(IW)+1
579             IF (LP(IW) .GT. N) LP(IW)=1
580         ELSE
581             IF (LP(IW) .EQ. MOSTL(IW)) THEN
582                 MOSTL(IW)=MOSTL(IW)-1
583                 IF (MOSTL(IW) .EQ. 0) MOSTL(IW)=N
584                 IF (MOSTL(IW) .EQ. MOSTR(IW)) THEN
585                     MOSTR(IW)=MOSTL(IW)
586                     NTAU(IW)=NTAU(IW)+1
587                 ENDIF
588             ENDIF
589             LP(IW)=LP(IW)-1
590             IF (LP(IW) .EQ. 0) LP(IW)=N
591             DO 320 ID=1,ND
592                 IDIR1(IW,ID)=IDIR(ID,LD(IW,LP(IW)))
593             320 CONTINUE
594             LD(IW,LP(IW))=LDNEW(IW)
595         ENDIF
596         DO 322 ID=1,ND
597             IRN(IW,ID)=IRNEW(IW,ID)
598         322 CONTINUE
599 C -----
600 C --- ARRIVE HERE IF NO JUMP HAS BEEN MADE
601     380 CONTINUE
602     250 CONTINUE
603 C --- END OF NON VECTORIZABLE CODE
604 C -----
605         DO 384 ID=1,ND
606             DO 386 IW=1,NW
607                 JROLD=IRO(IW,ID)
608                 JRNEW=IRNEW(IW,ID)
609                 JRDIF=JRNEW-JROLD
610                 ISUMR(IW,ID)=ISUMR(IW,ID)+JRDIF
611                 NSIG2(IW)=NSIG2(IW)+JRDIF*(JROLD+JRNEW)
612                 IRO(IW,ID)=IRO(IW,ID)+ICDIR(IW)*IDIR1(IW,ID)
613             386 CONTINUE
614         384 CONTINUE
615 C -----
616         DO 390 IW=1,NW
617             NR2(IW)=(IRN(IW,1)-IRO(IW,1))*2
618         390 CONTINUE
619         DO 392 ID=2,ND
620             DO 391 IW=1,NW
621                 NR2(IW)=NR2(IW)+(IRN(IW,ID)-IRO(IW,ID))*2
622             391 CONTINUE
623         392 CONTINUE
624 C -----
625 C --- TAKE DATA FOR AVERAGING
626 C --- FIRST, CALCULATE SQUARE OF SUM OF ALL SITES
627         DO 400 IW=1,NW
628             IRM2(IW)=ISUMR(IW,1)**2
629         400 CONTINUE
630         DO 410 ID=2,ND
631             DO 420 IW=1,NW
632                 IRM2(IW)=IRM2(IW)+ISUMR(IW,ID)**2
633             420 CONTINUE
634         410 CONTINUE
635 C --- NOW SUM ALL RELEVANT DATA
636         DO 430 IW=1,NW
637             IS=NS(IW)
638             IS2=IS*IS
639             NSUMS(IW)=NSUMS(IW)+IS
640             NSUMS2(IW)=NSUMS2(IW)+IS2

```

```

641      NSUMS3(IW)=NSUMS3(IW)+IS2*IS
642      NSUMS4(IW)=NSUMS4(IW)+IS2*IS2
643      NR2H=NR2(IW)
644      NSUMR2(IW)=NSUMR2(IW)+NR2H
645      NSUMR4(IW)=NSUMR4(IW)+NR2H*NR2H
646      G2=NSIG2(IW)-FLOAT(IRM2(IW))*FLN1I
647      SUMG2(IW)=SUMG2(IW)+G2
648      SUMG4(IW)=SUMG4(IW)+G2*G2
649      IF (NR2H .EQ. 0) NSUMP0(IW)=NSUMP0(IW)+1
650      430      CONTINUE
651  C -----
652      210      CONTINUE
653  C ---      UPDATE TOTAL NUMBER OF MONTE CARLO STEPS
654      NMCSUM=NMCSUM+NMC
655  C ----      PRINT PARTIAL RESULTS
656      WRITE(6, '(1X,3I12)') IINT,NMCSUM,NMC
657      KW1=1
658      DO 700 IC=1,NC
659          KW2=KW1+NCW(IC)-1
660          CS=0.
661          CS2=0.
662          CS3=0.
663          CS4=0.
664          CR2=0.
665          CR4=0.
666          CG2=0.
667          CG4=0.
668          CP0=0.
669          CEX=0.
670          CRE=0.
671          ICTAU=0
672          DO 710 IW=KW1,KW2
673              CS=CS+FLOAT(NSUMS(IW))
674              CS2=CS2+FLOAT(NSUMS2(IW))
675              CS3=CS3+FLOAT(NSUMS3(IW))
676              CS4=CS4+FLOAT(NSUMS4(IW))
677              CR2=CR2+FLOAT(NSUMR2(IW))
678              CR4=CR4+FLOAT(NSUMR4(IW))
679              CG2=CG2+SUMG2(IW)
680              CG4=CG4+SUMG4(IW)
681              CP0=CP0+FLOAT(NSUMP0(IW))
682              CEX=CEX+FLOAT(NSUMEX(IW))
683              CRE=CRE+FLOAT(NSUMRE(IW))
684              ICTAU=ICTAU+NTAU(IW)
685      710      CONTINUE
686      FNORMI=1./FLOAT(NCW(IC)*NMC)
687      FNORM2=FNORMI*FLN1I
688      CS=CS*FNORMI
689      CS2=CS2*FNORMI
690      CS3=CS3*FNORMI
691      CS4=CS4*FNORMI
692      CR2=CR2*FNORMI
693      CR4=CR4*FNORMI
694      CG2=CG2*FNORM2
695      CG4=CG4*FNORM2*FLN1I
696      CP0=CP0*FNORMI
697      CEX=CEX*FNORMI
698      CRE=CRE*FNORMI
699      IF (ICTAU .NE. 0) THEN
700          ICTAU=INT(FLOAT(NMCSUM*NCW(IC))/FLOAT(ICTAU))
701      ELSE
702          ICTAU=-1
703      ENDIF
704      WRITE(6,670) IC,CS,CS2,CS3,CS4,CR2,CR4,CG2,CG4,CP0,CEX,CRE,

```

```

705      *              ICTAU
706      670      FORMAT(1X,I2,1P,8E12.5,E11.4,0P,2F8.5,I7)
707      KW1=KW2+1
708      700      CONTINUE
709  C -----
710  C ---      IF NMC NOT YET SET TO NMCOUT, SET IT NOW
711      NMC=NMCOUT
712      200 CONTINUE
713  C -----
714  C --- CHECK NUMBER OF NODES NEVER USED
715      DO 800 NO=NNODES,1,-1
716      IF (IVISIT(NO) .NE. -1) GOTO 802
717      800 CONTINUE
718      WRITE(6, '(1X, ''NODE-LIST HAS NOT BEEN USED!?'')')
719      STOP
720  C -----
721      802 CONTINUE
722      NFREE=NNODES-NO
723      WRITE(6, '(1X,I12, '' NODES OF'',I12, '' NOT USED'')') NFREE,NNODES
724  C -----
725  C --- PROGRAM FINISHED. THE FOLLOWING IS A TEST FOR CONSISTENCY
726  C      IF WISHED
727      IF (.NOT. DEBUG) STOP
728  C -----
729  C --- THE FOLLOWING CODE IS NOT VECTORIZED. USED ONLY DURING DEBUG.
730  C --- CONSISTENCY CHECK OF HASH-TABLES
731      WRITE(6, '(1X,39(1H*))')
732      WRITE(6, '(1X, ''***** CONSISTENCY-CHECK *****'')')
733      DO 900 IW=1,NW
734      IR2=0
735      DO 9000 ID=1,ND
736      IR2=IR2+(IRN(IW,ID)-IR0(IW,ID))*2
737      JSUMR(ID)=0
738      9000 CONTINUE
739      JSIG2=0
740      IF (IR2 .EQ. NR2(IW)) GOTO 9010
741      WRITE(6, '(1X, ''ERROR IN NR2'')')
742      STOP
743  C -----
744      9010 CONTINUE
745  C --- CHECK EVERY STEP
746      IF (ICDIR(IW) .LT. 0) THEN
747  C ---      TURN WALK INTO STANDARD DIRECTION
748      ICDIR(IW)=1
749      DO 9012 ID=1,ND
750      IRHELP=IR0(IW,ID)
751      IR0(IW,ID)=IRN(IW,ID)
752      IRN(IW,ID)=IRHELP
753      9012 CONTINUE
754      ENDIF
755      DO 916 K=0,N
756      HO = HASH(IR0(IW,1),IR0(IW,2),IR0(IW,3))
757  C ---      CORRECT HASH-CODE?
758      HT=(HO-IW)/NW
759      IF (HT .GE. 0 .AND. HT .LT. LHASH) GOTO 902
760      WRITE(6, '(1X, ''WRONG HASH-CODE'')')
761      STOP
762  C -----
763      902 CONTINUE
764      HOLD(IW)=HO
765      KOLD(IW) = KEY(IR0(IW,1),IR0(IW,2),IR0(IW,3))
766  C -----
767  C --- UPDATE SUM-VARIABLES
768      DO 9030 ID=1,ND

```

```

769             JSUMR(ID)=JSUMR(ID)+IR0(IW,ID)
770             JSIG2=JSIG2+IR0(IW,ID)**2
771 9030         CONTINUE
772 C -----
773             LOLD=HEADER(HOLD(IW))
774             LLAST=0
775 904         CONTINUE
776 C ---
777             SITE HAS TO BE OCCUPIED
778             IF (LOLD .GT. 0) GOTO 906
779             WRITE(6, '(1X, ''ERROR IN LOLD. IR0 IS NOT OCCUPIED'' )')
780             STOP
781 C -----
782 906         CONTINUE
783             IF (ISITE(LOLD) .EQ. KOLD(IW)) GOTO 908
784             LLAST=LOLD
785             LOLD=LINK(LOLD)
786             GOTO 904
787 C ---
788 908         CONTINUE
789             IF (IVISIT(LOLD) .GT. 0) GOTO 910
790             WRITE(6, '(1X, ''ERROR IN IVISIT'' )')
791             STOP
792 C -----
793 910         CONTINUE
794             IVISIT(LOLD)=IVISIT(LOLD)-1
795             IF (IVISIT(LOLD) .GT. 0) GOTO 912
796 C ---
797             REMOVE NODE FROM NODE-LIST
798             IF (LLAST .EQ. 0) THEN
799                 HEADER(HOLD(IW))=LINK(LOLD)
800             ELSE
801                 LINK(LLAST)=LINK(LOLD)
802             ENDIF
803             LINK(LOLD)=LBEGIN
804             LBEGIN=LOLD
805             NS(IW)=NS(IW)-1
806 C -----
807 912         CONTINUE
808             IF (K .LT. N) THEN
809                 IF (LP(IW) .GT. 0 .AND. LP(IW) .LE. N) GOTO 913
810                 WRITE(6, '(1X, ''ERROR IN LP'' )')
811                 STOP
812 C -----
813 913         CONTINUE
814             KK=LP(IW)+K
815             DO 914 ID=1,ND
816                 IF (KK .GT. N) KK=KK-N
817                 IR0(IW,ID)=IR0(IW,ID)+IDIR(ID,LD(IW,KK))
818             CONTINUE
819             ENDDO
820 914         CONTINUE
821 C -----
822 916         CONTINUE
823 C ---
824 ALL STEPS REMOVED
825             IF (NS(IW) .EQ. 0) GOTO 918
826             WRITE(6, '(1X, ''ERROR IN NS'' )')
827             STOP
828 C -----
829 C --- DO IR0 AND IRN MATCH?
830 918         CONTINUE
831             DO 920 ID=1,ND
832                 IF (IRN(IW,ID) .EQ. IR0(IW,ID)) GOTO 922
833                 WRITE(6, '(1X, ''ERROR IN IRN'' )')
834                 STOP
835 C -----
836 922         CONTINUE

```

```

833     920    CONTINUE
834 C -----
835 C ---    SUM-VARIABLES OK?
836         IF (JSIG2 .EQ. NSIG2(IW)) GOTO 9040
837         WRITE(6, '(1X, ''ERROR IN NSIG2'')')
838         STOP
839 C -----
840     9040    CONTINUE
841         DO 9050 ID=1,ND
842             IF (JSUMR(ID) .EQ. ISUMR(IW,ID)) GOTO 9060
843             WRITE(6, '(1X, ''ERROR IN NSUMR'')')
844             STOP
845 C -----
846     9060    CONTINUE
847     9050    CONTINUE
848 C ---    ALL WALK VARIABLES OK
849 C -----
850 C ---    HEADER-TABLE EMPTY FOR WALK IW?
851         DO 924 IH=0,LH1
852             IF (HEADER(NW*IH+IW) .EQ. 0) GOTO 926
853             WRITE(6, '(1X, ''ERROR IN HEADER-TABLE (NOT EMPTY)'')')
854             STOP
855 C -----
856     926    CONTINUE
857     924    CONTINUE
858 C -----
859     900    CONTINUE
860 C ---    ALL WALKS CHECKED
861 C -----
862 C ---    CHECK NODE-LIST. IS IT EMPTY AND CORRECTLY LINKED?
863         INODES=0
864         NO=LBEGIN
865     928    CONTINUE
866         IF (NO .GT. 0 .AND. NO .LE. NNODES) GOTO 930
867         WRITE(6, '(1X, ''ERROR IN LINKAGE OF NODE-LIST'')')
868         STOP
869 C -----
870     930    CONTINUE
871         IF (IVISIT(NO) .GT. -2) GOTO 932
872         WRITE(6, '(1X, ''NODE TWICE IN NODE-LIST'')')
873         STOP
874 C -----
875     932    CONTINUE
876         IVISIT(NO)=-2
877         INODES=INODES+1
878         IF (INODES .EQ. NNODES) GOTO 934
879         NO=LINK(NO)
880         GOTO 928
881 C -----
882     934    CONTINUE
883 C ---    NODE-LIST IS OK
884 C -----
885         WRITE(6, '(1X, ''**** CONSISTENCY-CHECK SUCCESSFUL ****'')')
886         WRITE(6, '(1X, 39(1H*))')
887 C -----
888         STOP
889         END

```

Anhang E. Fehlerabschätzung von Monte Carlo Daten unter Berücksichtigung von Korrelationen

Im allgemeinen werden Monte Carlo Simulationen so eingerichtet, daß man im Laufe der Rechnung einzelne Ergebnisse y_k erhält, die statistisch unabhängig voneinander sind. Den geschätzten Fehler δz des gesamten Mittelwertes z aus K solchen "Messungen",

$$z := \frac{1}{K} \sum_{k=0}^{K-1} y_k, \quad (E.1)$$

erhält man dann aus der gemessenen Varianz der Daten $\Delta^2 y$ zu

$$\delta z = \sqrt{\frac{\Delta^2 y}{K}}. \quad (E.2)$$

Bei dieser Abschätzung geht wesentlich die statistische Unabhängigkeit der Daten ein. Da in der oben benutzten Simulation aber Mittelwerte über jeweils N einzelnen, korrelierten Werten ausgegeben werden, sind auch die Mittelwerte noch miteinander korreliert. Die Fehlerabschätzung wird deshalb im folgenden für den Fall wiederholt, daß man korrelierte Daten aus der Simulation erhält.

Die elementaren Daten x_i aus dem Monte Carlo Prozess, die stark korreliert sind, werden in Gruppen von je N einzelnen Monte Carlo Schritten zusammengefaßt. Nur die Mittelwerte aus diesen N Schritten, die weit weniger korreliert sind, werden ausgegeben. K solcher Mittelwerte werden insgesamt erzeugt. Das Ziel der folgenden Rechnung ist es, aus den Schwankungen der K Einzelmittelwerte eine Fehlerabschätzung für das Gesamtergebn zu gewinnen. Dazu wird im folgenden o.B.d.A. angenommen, daß die nach jedem einzelnen Monte Carlo Schritt erhaltenen Daten x_i den Mittelwert 0 haben. Die Korrelationszeit, die ebenfalls aus der Simulation gewonnen wird, sei τ , d. h. es wird angenommen, daß

$$\langle x_i x_{i+t} \rangle = \langle x^2 \rangle e^{-t/\tau} \quad (E.3)$$

gilt. In der Simulation werden nun jeweils N Werte gemittelt und das arithmetische Mittel y_k mit

$$y_k := \frac{1}{N} \sum_{i=0}^{N-1} x_{Nk+i} \quad (E.4)$$

wird ausgegeben. Im Mittel erhält man hier natürlich ebenfalls 0. Die Schwankung $\langle y^2 \rangle$ kann man dann mit Gl. (E.3) berechnen. Man erhält mit

$$q := e^{-1/\tau} \quad (E.5)$$

sofort

$$\langle y^2 \rangle = \frac{1}{N^2} \sum_{i,j=0}^{N-1} \langle x_{Nk+i} x_{Nk+j} \rangle = \frac{\langle x^2 \rangle}{(1-q)N} \left[1 + q - \frac{2q(1-q^N)}{N(1-q)} \right]. \quad (E.6)$$

Die einzelnen, vom Programm ausgegebenen y_k -Werte sind nicht statistisch unabhängig voneinander. Die Korrelationsfunktion ergibt sich nach kurzer Rechnung zu

$$\begin{aligned} \langle y_k y_{k+l} \rangle &= \frac{1}{N^2} \sum_{i,j=0}^{N-1} \langle x_{Nk+i} x_{N(k+l)+j} \rangle \\ &= \frac{\langle x^2 \rangle}{[N(1-q)]^2} q^{N(l-1)+1} (1-q^N)^2 \quad \text{für } l > 0. \end{aligned} \quad (E.7)$$

Insgesamt werden K solcher Ergebnisse ausgegeben, die anschließend gemittelt werden, um das endgültige Monte Carlo Ergebnis z zu erhalten,

$$z := \frac{1}{K} \sum_{k=0}^{K-1} y_k. \quad (E.8)$$

Auch hier gilt, daß im Mittel z gleich 0 ist. Für die Schwankung $\langle z^2 \rangle$ erhält man nach elementarer Rechnung mit Hilfe von Gl. (E.6) und Gl. (E.7)

$$\begin{aligned} \langle z^2 \rangle &= \frac{1}{K^2} \sum_{k,l=0}^{K-1} \langle y_k y_l \rangle \\ &= \frac{\langle x^2 \rangle}{N(1-q)K} \left(1 + q - \frac{2q(1-q^{NK})}{N(1-q)K} \right). \end{aligned} \quad (E.9)$$

Die Gl. (E.6) und Gl. (E.9) verknüpfen nun die Varianz der ursprünglich simulierten Daten x_i mit den Varianzen der ausgegebenen Mittelwerte y_k und der Varianz, also dem Quadrat des geschätzten Fehlers, von z . Aus der Simulation zugänglich sind aber nur die Werte y_k , aus denen man die Varianz $\langle y^2 \rangle$ näherungsweise bestimmen kann. Aus Gl. (E.6) erhält man $\langle x^2 \rangle$ und mit Hilfe von Gl. (E.9) schließlich die Varianz von z

$$\langle z^2 \rangle = F \frac{\langle y^2 \rangle}{K} \quad (E.10)$$

mit

$$F := \frac{1 + q - \frac{2q(1-q^{NK})}{N(1-q)K}}{1 + q - \frac{2q(1-q^N)}{N(1-q)}}. \quad (E.11)$$

(Zur Erinnerung: falls die y_k unkorreliert wären, würde $F = 1$ gelten; hier gilt jedoch $F > 1$.) Normalerweise wählt man für die Anzahl N der zu mittelnden Einzelwerte x_i

einen Wert in der Größenordnung der Korrelationszeit τ , die im allgemeinen wiederum groß gegen 1 ist,

$$N = \alpha\tau \gg 1. \quad (E.12)$$

Damit erhält man für F sofort

$$F := \frac{1 - \frac{1-e^{-\alpha K}}{\alpha K}}{1 - \frac{1-e^{-\alpha}}{\alpha}}. \quad (E.13)$$

Ferner sollte man für K , also die Zahl der ausgegebenen Mittelwerte, eine Zahl wählen, die groß gegen 1 ist. Sonst kann man die Varianz $\Delta^2 y$ nicht gut genug bestimmen. Dann ist

$$F := \frac{1}{1 - \frac{1-e^{-\alpha}}{\alpha}}. \quad (E.14)$$

Bei den oben durchgeführten Simulationen wurde immer $\alpha \approx 2$ gewählt (entspricht $F \approx 1,76$). Nach Gl. (E.10) ist der geschätzte Fehler δz

$$\delta z = \sqrt{F \frac{\langle y^2 \rangle}{K}},$$

also um das $\sqrt{F}$ -fache größer, als bei statistisch unabhängigen, simulierten Werten y_k .

Anhang F. Rekursionsformeln für $\langle s_n \rangle$ in $d = 2$ und $d = 3$

Bei Abwesenheit von Haftstellen können Rekursionsformeln für die mittlere Zahl der besuchten Plätze angegeben werden, wobei die Rekursion über die Schrittzahl n läuft. Für verschiedene Gittertypen in $d = 2$ ist dies schon von G. Zumofen und A. Blumen [15] durchgeführt worden. In leicht abgewandelter Form werden ihre Überlegungen hier dargestellt und auf $d = 3$ erweitert.

Nach Ref. 43 gilt für die erzeugenden Funktionen der Anzahl der besuchten Plätze S_n und für die Rückkehrwahrscheinlichkeit zum Startpunkt P_n nach n Schritten für $|z| < 1$

$$\begin{aligned} P(z) &:= \sum_{n=0}^{\infty} P_n z^n, \\ S(z) &:= \sum_{n=0}^{\infty} S_n z^n = \frac{1}{(1-z)^2 P(z)}. \end{aligned} \quad (F.1)$$

Also gilt

$$\sum_{n=0}^{\infty} \sum_{k=0}^n S_k P_{n-k} z^n = S(z) P(z) = \frac{1}{(1-z)^2} = \sum_{n=0}^{\infty} (n+1) z^n. \quad (F.2)$$

Aus dem Koeffizientenvergleich beider Seiten erhält man damit sofort die Rekursionsformel

$$\begin{aligned} S_0 &= 1 \\ S_n &= n+1 - \sum_{k=0}^{n-1} S_k P_{n-k}, \quad \text{für } n \geq 1. \end{aligned} \quad (F.3)$$

Hierbei sind $P_0 = 1$ und $S_0 = 1$ benutzt worden. Die Formel (F.3) gilt ganz allgemein für d Dimensionen und beliebige Gittertypen.

Auf einfach kubischen Gittern in d Dimensionen gilt $P_{2k+1} = 0$. Mit $a_l := P_{2l}$ kann Gl. (F.3) auch geschrieben werden als

$$\begin{aligned} S_0 &= 1 \\ S_n &= n+1 - \sum_{l=1}^{[n/2]} a_l S_{n-2l}, \quad \text{für } n \geq 1, \end{aligned} \quad (F.4)$$

wobei [...] die Gaußklammer bedeutet, also den ganzzahligen Anteil.

Nach Ref. 15 gilt für das zwei-dimensionale Quadratgitter

$$\begin{aligned} a_0 &= 1 \\ a_n &= P_{2n} = \left[\frac{(2n-1)!!}{(2n)!!} \right]^2 = \left(\frac{2n-1}{2n} \right) \quad \text{für } n \geq 1. \end{aligned} \quad (F.5)$$

Für das drei-dimensionale einfach-kubische Gitter hat G. S. Joyce eine Rekursionsformel für die Rückkehrwahrscheinlichkeit gefunden [121]. Es gilt

$$\begin{aligned}
 a_{-1} &= 0 \\
 a_0 &= 1 \\
 a_{n+1} &= \frac{2(2n+1)(10n^2 + 10n + 3)a_n - n(4n^2 - 1)a_{n-1}}{36(n+1)^3}, \quad \text{für } n \geq 0.
 \end{aligned}
 \tag{F.6}$$

Die Gleichungen (F.5) und (F.6) erlauben nun in Verbindung mit Gl. (F.4) die rekursive Berechnung der mittleren Zahl der besuchten Plätze nach n Schritten in zwei und drei Dimensionen. Die folgenden Tabellen enthalten die Ergebnisse für einige Schrittzahlen n . Zum Vergleich sind die Ergebnisse der asymptotischen Theorie, Gl. (II.16), als S_n^{asy} mit in die Tabellen eingetragen.

n	P_n	S_n	S_n^{asy}
0	1,0000	1,000	0,000
1	0,0000	2,000	1,511
2	$2,5000 \cdot 10^{-1}$	2,750	2,266
5	$0,0000 \cdot 10^{+0}$	4,844	4,258
10	$6,0562 \cdot 10^{-2}$	7,881	7,169
20	$3,1045 \cdot 10^{-2}$	13,340	12,380
50	$1,2606 \cdot 10^{-2}$	27,814	26,217
100	$6,3344 \cdot 10^{-3}$	49,504	46,997
200	$3,1752 \cdot 10^{-3}$	89,258	85,164
500	$1,2720 \cdot 10^{-3}$	197,541	189,388
1000	$6,3630 \cdot 10^{-4}$	363,584	349,563
2000	$3,1823 \cdot 10^{-4}$	673,492	649,066
5000	$1,2731 \cdot 10^{-4}$	1534,013	1482,354
10000	$6,3659 \cdot 10^{-5}$	2874,569	2782,687

Tab. F.1: Exakte Werte für P_n und S_n in $d = 2$: Rekursiv berechnete exakte Werte für die Rückkehrwahrscheinlichkeit P_n und die mittlere Zahl der besuchten Plätze S_n auf dem Quadratgitter ohne Haftstellen. S_n^{asy} sind die Ergebnisse der asymptotischen Theorie.

n	P_n	S_n	S_n^{asy}
0	1,0000	1,000	0,000
1	0,0000	2,000	0,659
2	$1,6667 \cdot 10^{-1}$	2,833	1,319
5	0,0000	5,250	3,297
10	$1,9392 \cdot 10^{-2}$	9,051	6,595
20	$7,1084 \cdot 10^{-3}$	16,381	13,189
50	$1,8386 \cdot 10^{-3}$	37,641	32,973
100	$6,5492 \cdot 10^{-4}$	72,287	65,946
200	$2,3242 \cdot 10^{-4}$	140,605	131,893
500	$5,8930 \cdot 10^{-5}$	343,156	329,731
1000	$2,0850 \cdot 10^{-5}$	678,200	659,463
2000	$7,3745 \cdot 10^{-6}$	1345,179	1318,925
5000	$1,8660 \cdot 10^{-6}$	3338,481	3297,313
10000	$6,5980 \cdot 10^{-7}$	6652,604	6594,627

$$d = 3$$

Tab. F.2: Exakte Werte für P_n und S_n in $d = 3$

Anhang G. Berechnung des effektiven Exponenten α als gewichteten Mittelwert

Zunächst betrachte man die Zustandssumme Z_{n+1} für SIW's mit $n+1$ Schritten

$$Z_{n+1} := \sum_{\omega_{n+1}} e^{-\lambda s_{\omega_{n+1}}} = \sum_{\omega_n} \sum_{\rho=1}^{2d} e^{-\lambda(s_{\omega_n} + \chi_{\omega_n}^\rho)}. \quad (G.1)$$

Hier wurde der letzte Schritt ρ vom $n+1$ -Schritt Walk ω_{n+1} abgespalten. Die Größe $\chi_{\omega_n}^\rho$ wird dabei als 1 definiert, falls der Schritt ρ auf einen neuen Platz führt, und 0 sonst. Sei nun mit k_{ω_n} die Anzahl der Richtungen bezeichnet, in denen der $(n+1)$ -te Schritt ρ einen neuen Platz besucht. Dann gilt

$$\sum_{\rho=1}^{2d} \exp(-\lambda \chi_{\omega_n}^\rho) = k_{\omega_n} e^{-\lambda} + (2d - k_{\omega_n}) e^0 = 2d - k_{\omega_n} c \quad (G.2)$$

und deshalb

$$Z_{n+1} = 2d \sum_{\omega_n} e^{-\lambda s_{\omega_n}} - c \sum_{\omega_n} k_{\omega_n} e^{-\lambda s_{\omega_n}} = 2d Z_n - c Z_n \langle k_{\omega_n} \rangle_{\text{SIW}}. \quad (G.3)$$

Definiert man

$$W := \left\langle \frac{k_{\omega_n}}{2d} \right\rangle_{\text{SIW}} \quad (G.4)$$

als die Wahrscheinlichkeit, beim $(n+1)$ -ten Schritt einen neuen Platz zu besuchen, erhält man

$$\frac{Z_{n+1}}{2d Z_n} = 1 - cW. \quad (G.5)$$

Für die Überlebenswahrscheinlichkeit Ψ_n gilt wegen Gl. (II.25)

$$\frac{\Psi_{n+1}}{\Psi_n} = 1 - cW. \quad (G.6)$$

Schreibt man Ψ_n mit einem effektiven Exponenten α als

$$\Psi_n \sim e^{-Kn^\alpha}, \quad (G.7)$$

bekommt man

$$-\ln \left(\frac{\Psi_{n+1}}{\Psi_n} \right) = \alpha K n^{\alpha-1} (1 + O\left(\frac{1}{n}\right))$$

und deshalb für große n

$$-\ln(1 - cW) = \alpha K n^{\alpha-1}. \quad (G.8)$$

Die Wahrscheinlichkeit W , beim $(n+1)$ -ten Schritt einen neuen Platz zu besuchen, kann ganz einfach bei der Simulation mitberechnet werden. Jedesmal wenn die Kette versucht, auf einen neuen Platz zu kriechen, überprüft der Algorithmus ohnehin, ob der neue Platz schon besucht ist, oder nicht. Damit kann W bestimmt werden.

LITERATURVERZEICHNIS

- [1] G. H. Weiss und R. J. Rubin, *Adv. Chem. Phys.* **52**, 363 (1983)
- [2] J. W. Haus und K. W. Kehr, *Diffusion in Regular and Disordered Lattices*, *Physics Reports* **150**, 263 (1987)
- [3] S. Havlin und D. Ben Avraham, *Adv. Phys.* **36**, 695 (1987)
- [4] R. Kohlrausch, *Ann. Phys. (Leipzig)* **12**, 393 (1847)
- [5] G. Williams und D. C. Watts, *Trans. Faraday Soc.* **66**, 80 (1970)
- [6] R. G. Palmer, D. L. Stein, E. Abrahams und P. W. Anderson, *Phys. Rev. Lett.* **53**, 958 (1984), *Phys. Rev. Lett.* **54**, 1965 (1985)(Errata), R. Zwanzig, *Phys. Rev. Lett.* **54**, 364 (1985)(Comment)
- [7] M. V. Smoluchowski, *Phys. Z.* **17**, 557 (1916)
- [8] H. B. Rosenstock, *Phys. Rev.* **187**, 1166 (1969)
- [9] H. B. Rosenstock, *J. Math. Phys.* **11**, 487 (1970)
- [10] B. Ya. Balagurov und V. G. Vaks, *Zh. Eksp. Teor. Fiz.* **65**, 1939 (1973) [Englische Übersetzung: *Sov. Phys.-JETP* **38**, 968 (1974)]
- [11] M. D. Donsker und S. R. S. Varadhan, *Commun. Pure Appl. Math.* **28**, 525 (1975)
- [12] M. D. Donsker und S. R. S. Varadhan, *Commun. Pure Appl. Math.* **32**, 721 (1979)
- [13] P. Grassberger und I. Procaccia, *Phys. Rev. A* **26**, 3686 (1982)
- [14] R. F. Kayser und J. B. Hubbard, *Phys. Rev. Lett.* **51**, 79 (1983)
- [15] G. Zumofen und A. Blumen, *J. Chem. Phys.* **76**, 3713 (1982)
- [16] G. Zumofen und A. Blumen, *Chem. Phys. Lett.* **88**, 63 (1982)
- [17] A. Blumen und G. Zumofen, *J. Stat. Phys.* **30**, 487 (1983)
- [18] G. Zumofen, J. Klafter und A. Blumen, *J. Chem. Phys.* **79**, 5131 (1983)
- [19] A. Blumen, G. Zumofen und J. Klafter, *J. Luminesc.* **31**, 639 (1984)

- [20] G. Zumofen, A. Blumen und J. Klafter, *J. Phys. A* **17**, L479 (1984)
- [21] A. Blumen, *Models for Dynamics of Relaxation in Glasses*, to appear in *Molecular Dynamics and Relaxation. Phenomena in Glasses*, ed. by Th. Dorfmüller and G. Williams, Lecture Notes in Physics (Springer, Heidelberg, 1987)
- [22] A. Blumen, J. Klafter und G. Zumofen, *Phys. Rev. B* **28**, 6112 (1983)
- [23] A. Blumen, J. Klafter, b. S. White und G. Zumofen, *Phys. Rev. Lett.* **53**, 1301 (1984)
- [24] J. Klafter, G. Zumofen und A. Blumen, *J. Phys. Lett.* **45**, L-49 (1984)
- [25] J. Klafter und A. Blumen, *J. Chem. Phys.* **80**, 875 (1984)
- [26] A. Blumen, J. Klafter und G. Zumofen, *J. Phys. A* **19**, L77 (1986)
- [27] S. Havlin, G. H. Weiss, J. E. Kiefer und M. Dishon, *J. Phys. A* **17**, L347 (1984)
- [28] S. Havlin, M. Dishon, J. E. Kiefer und G. H. Weiss, *Phys. Rev. Lett.* **53**, 407 (1984)
- [29] M. Fixman, *Phys. Rev. Lett.* **52**, 791 (1984)
- [30] I. G. Hunt, D. Bloor und B. Movaghar, *J. Phys. C* **16**, L623 (1983)
- [31] J. K. Anlauf, *Phys. Rev. Lett.* **52**, 1845 (1984)
- [32] J. K. Anlauf, Diplomarbeit, Universität zu Köln (1984), unveröffentlicht
- [33] T. C. Lubensky, *Phys. Rev. A* **30**, 2657 (1984)
- [34] J. Schotland, *Physica* **146 A**, 295 (1987)
- [35] U. Seiferheld, H. Bässler und B. Movaghar, *Phys. Rev. Lett.* **51**, 813 (1983)
- [36] S. D. D. V. Rughooputh, D. Bloor, D. Phillips und B. Movaghar, *Phys. Rev. B* **35**, 8103 (1987)
- [37] H. E. Stanley, K. Kang, S. Redner und R. L. Blumberg, *Phys. Rev. Lett.* **51**, 1223 (1983)
- [38] H. Meirovitch, *Phys. Rev. A* **32**, 3699 (1985)

- [39] H. van Beijeren, *Rev. Mod. Phys.* **54**, 195 (1982)
- [40] Ya. G. Sinai, in *Proceedings of the Berlin Conference on Mathematical Problems in Theoretical Physics*, ed. R. Schrader, R. Sailer und D. A. Uhlenbrock (Springer, Berlin, 1982)
- [41] I. M. Lifshitz *Adv. Phys.* **13**, 483 (1964)
- [42] Th. M. Nieuwenhuizen und J. M. Luck, *Physica* **145 A**, 161 (1987)
- [43] E. W. Montroll und G. H. Weiss, *J. Math. Phys.* **6**, 167 (1965)
- [44] F. S. Henyey und V. Seshadri, *J. Chem. Phys.* **76**, 5530 (1982)
- [45] P. G. de Gennes, *Scaling Concepts in Polymer Physics*, (Cornell University Press, Ithaca, N. Y., 1979)
- [46] D. J. Amit, G. Parisi und L. Peliti, *Phys. Rev. B* **27**, 1635 (1983)
- [47] C. Domb und G. S. Joyce, *J. Phys. C* **5**, 956 (1972)
- [48] P. M. Duxbury, S. L. A. de Queiroz und R. B. Stinchcombe, *J. Phys. A* **17**, 2113 (1984)
- [49] P. M. Duxbury und S. L. A. de Queiroz, *J. Phys. A* **18**, 661 (1985)
- [50] P. L. Leath und P. M. Duxbury, *J. Phys. A* **18**, 1435 (1985)
- [51] S. Alexander, J. Bernasconi, W. R. Schneider und R. Orbach, *Rev. Mod. Phys.* **53**, 175 (1981)
- [52] S. Redner und K. Kang, *Phys. Rev. Lett.* **51**, 1729 (1983)
- [53] S. Redner und K. Kang, Erratum in *Phys. Rev. Lett.* **52**, 401 (1984)
- [54] K. Unnikrishnan und M. A. Prasad, *Phys. Lett.* **100 A**, 19 (1984)
- [55] K. Kang und S. Redner, *J. Chem. Phys.* **80**, 2752 (1984)
- [56] M. Kantha und V. Balakrishnan, *J. Phys. C* **18**, 4819 (1985)
- [57] N. Agmon, *J. Stat. Phys.* **43**, 537 (1986)
- [58] B. Movaghar, D. Würtz und B. Pohlmann, *Z. Phys. B* **66**, 523 (1987)
- [59] D. Y. C. Chan und B. D. Hughes, *J. Phys. A* **17**, L121 (1984)

- [60] B. Movaghar, G. W. Sauer und D. Würtz, *J. Stat. Phys.* **27**, 473 (1982)
- [61] D. Würtz, B. Pohlmann und B. Movaghar, *Phys. Rev. A* **31**, 3526 (1985)
- [62] M. A. Prasad, *Phys. Rev. A* **37**, 1383 (1988)
- [63] M. A. Prasad und K. Unnikrishnan, *Phys. Rev. A* **31**, 523 (1985)
- [64] J. Bardeen und C. Herring, in *Imperfections in Nearly Perfect Crystals*, ed. W. Shockley, (Wiley, New York, 1952), p. 261
- [65] W. Kuhn, *Kolloid-Z.* **76**, 258 (1936)
- [66] R. Fürth, *Z. Phys.* **2**, 244 (1920)
- [67] K. W. Kehr und P. Argyrakis, *J. Chem. Phys.* **84**, 5816 (1986)
- [68] J. W. Haus und K. W. Kehr, *Solid State Commun.* **26**, 753 (1978)
- [69] J. W. Haus und K. W. Kehr, *J. Phys. Chem. Solids* **40**, 1019 (1979)
- [70] E. W. Montroll, in *Proceedings of the Sixteenth Symposium on Applied Mathematics*, (American Mathematical Society, Providence, R.I., 1964) Vol. 16, p.193
- [71] E. W. Montroll, *J. Phys. Soc. Japan Suppl.* **26**, 6 (1969)
- [72] K. E. Shuler, H. Silver und K. Lindenberg, *J. Stat. Phys.* **13**, 393 (1976)
- [73] W. Th. F. den Hollander, *Random Walks on Random Lattices*, Dissertation, Rijksuniversiteit te Leiden (1985)
- [74] W. Th. F. den Hollander, *J. Stat. Phys.* **37**, 331 (1984)
- [75] L. V. Ahlfors, *Complex Analysis*, (McGraw-Hill, New York, 1979)
- [76] H. E. Daniels, *Proc. Camb. Phil. Soc.* **37**, 244 (1941)
- [77] R. J. Rubin, *J. Chem. Phys.* **56**, 5747 (1972)
- [78] G. H. Weiss und S. Havlin, *J. Stat. Phys.* **37**, 17 (1984)
- [79] G. H. Weiss und R. J. Rubin, *J. Stat. Phys.* **14**, 333 (1975)
- [80] R. M. Mazo und P. Grassberger (Vorabdruck)

- [81] B. Preziosi (Vorabdruck)
- [82] D. A. Darling und A. J. F. Siegert, *Ann. Math. Stat.* **24**, 624 (1953)
- [83] K. Binder in: *Monte Carlo Methods in Statistical Physics*, Topics in Current Physics, Vol. 7, ed. by K. Binder, (Springer, Berlin, Heidelberg, 1979)
- [84] K. Binder in: *Applications of the Monte Carlo Method in Statistical Physics*, Topics in Current Physics, Vol. 36, ed. by K. Binder, (Springer, Berlin, Heidelberg, 1984)
- [85] D. W. Heermann, *Computer Simulation Methods in Theoretical Physics*, (Springer, Berlin, Heidelberg, 1986)
- [86] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, (North-Holland, Amsterdam, 1981)
- [87] M. N. Rosenbluth, A. W. Rosenbluth, *J. Chem. Phys.* **23**, 356 (1955)
- [88] K. Kremer, (Dissertation Universität Köln), Berichte der Kernforschungsanlage Jülich-Jül-1832
- [89] H. Meirovitch, *J. Phys. A* **15**, L735 (1982)
- [90] H. Meirovitch, *Macromolecules* **18**, 563 (1985)
- [91] H. Meirovitch, *Phys. Rev. A* **32**, 3709 (1985)
- [92] K. E. Schmidt, *Phys. Rev. Lett.* **51**, 2175 (1983)
- [93] P. G. de Gennes, *J. Chem. Phys.* **55**, 572 (1971)
- [94] I. Webman, M. H. Kalos und J. L. Lebowitz, *J. Phys. (Paris)* **41**, 579 (1981)
- [95] K. Kremer, A. Baumgärtner, K. Binder, *Z. Phys. B* **40**, 331 (1981)
- [96] P. E. Rouse, *J. Chem. Phys.* **21**, 1272 (1953)
- [97] F. Spitzer, *Principles of Random Walk*, (Springer-Verlag, New York, 2nd ed., 1976)
- [98] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *J. Chem. Phys.* **21**, 1087 (1953)

- [99] U. Buchenau in *Ferienkurs '81 Physik der Polymere*, herausgegeben von der Kernforschungsanlage Jülich GmbH, Institut für Festkörperforschung (KFA Jülich, 1981)
- [100] J. J. Anlauf, K. W. Kehr und S. M. Reulein, Vorabdruck eingesandt zum *J. Stat. Phys.*
- [101] Z. W. Salsburg, J. D. Jacobsen, W. Fickett und W. W. Wood, *J. Chem. Phys.* **30**, 65 (1959)
- [102] P. C. Jurs und J. E. Reissner, *J. Chem. Phys.* **55**, 4948 (1971)
- [103] D. E. Knuth, *The Art of Computer Programming*, Vol. 3, *Sorting and Searching* (Addison-Wesley, Reading, 1968)
- [104] R. Bayer, *Datenorganisation und Suchverfahren*, Einheit 5 des Kurses *Datenstrukturen*, herausgegeben von der Fernuniversität/Gesamthochschule Hagen (Hagen, 1982)
- [105] P. J. Flory, *Statistical Mechanics of Chain Molecules*, (Interscience, New York, 1969)
- [106] N. C. Jain und W. E. Pruitt, *J. d'Analyse Math.* **24**, 369 (1971)
- [107] W. H. Press, B. P. Flannery, S. A. Teukolsky und W. T. Vetterling, *Numerical Recipes. The Art of Scientific Computing*, (Cambridge University Press, Cambridge, 1986)
- [108] P. J. Flory, *Principles of Polymer Chemistry*, (Cornell University Press, Ithaca, N. Y., 1979)
- [109] K. Kremer, A. Baumgärtner, K. Binder, *J. Phys. A* **15**, 2879 (1982)
- [110] D. Stauffer, *Introduction to Percolation Theory*, (Taylor and Francis, London, 1985)
- [111] B. Derrida und H. J. Herrmann, *J. Physique* **44**, 1365 (1983)
- [112] R. Dickman und W. C. Schieve, *J. Physique* **45**, 1727 (1984)
- [113] R. Dickman und W. C. Schieve, *J. Stat. Phys.* **44**, 465 (1986)
- [114] P. M. Lam und J. A. M. S. Duarte, (Vorabdruck)

- [115] *CRAY FORTRAN (CFT), Reference Manual*, SR-0009
- [116] *VS-FORTRAN, Language and Library Reference*, Release 4.0
- [117] S. Kirkpatrick, P. E. Stoll, *J. Chem. Phys.* **40**, 517 (1981)
- [118] R. C. Tausworthe, *Math. Comp.* **19**, 201 (1965)
- [119] *Unterprogramme INR250 CFT und R250 CFT*, Zentralinstitut für angewandte Mathematik der Kernforschungsanlage Jülich, D-5170 Jülich
- [120] C. Kalle und S. Wansleben, *Comput. Phys.* **33**, 343 (1984)
- [121] G. S. Joyce, *Phil. Trans. Roy. Soc.* **73A 2**, 583 (1973)

Herrn Prof. Dr. K. W. Kehr danke ich sehr herzlich für die Überlassung des Themas und die geduldige Betreuung während ihrer Anfertigung.

Den Mitarbeitern des Instituts für Festkörperforschung der Kernforschungsanlage Jülich danke ich für ihre ständige Diskussionsbereitschaft und ein angenehmes Arbeitsklima.

Dem Zentralinstitut für angewandte Mathematik der Kernforschungsanlage bin ich zu besonderem Dank verpflichtet. Die Möglichkeit auf einer CRAY-XMP arbeiten zu können, hat entschieden zum Gelingen dieser Arbeit beigetragen.

Herrn Dr. R. Czech gilt mein besonderer Dank. Seine immerwährende Diskussionsfreude bot ständig Gelegenheit zu konstruktiven Gesprächen.

Ihm und Frau Gabriele Jankowski möchte ich für die tatkräftige und zum Teil nervenaufreibende Unterstützung während der endgültigen Fertigstellung der Arbeit danken.

