

# **Simulationsmethoden in der Physik 2**

**Jörg Main**

Skript erstellt von Bernhard Grotz und Sebastian Macke

## **Teil 2: Perkolation, Fraktale und Zellularautomaten**

## Inhaltsverzeichnis

<b>1.</b>	<b>Einleitung</b>	<b>3</b>
<b>2.</b>	<b>Simulation auf dem Computer</b>	<b>4</b>
2.1	Der Burning-Algorithmus	5
2.2	Der Hoshen-Kopelman-Algorithmus	6
<b>3.</b>	<b>Perkolation</b>	<b>7</b>
3.1	Kritische Exponenten	7
3.2	Skalierungsgesetze und Renormierungsgruppentheorie	8
3.3	Beispiel: Perkolation beim Quadratgitter	8
3.4	Monte-Carlo Renormierungsgruppentheorie	11
3.5	Berechnung kritischer Exponenten	11
<b>4.</b>	<b>Exkurs: Fraktale</b>	<b>13</b>
4.1	Definitionen	13
4.2	Beispiele	13
<b>5.</b>	<b>Wachstumsmodelle und Fraktale</b>	<b>14</b>
5.1	Algorithmus von Hammersley, Leath und Alexandrowicz	14
5.2	Das Edenmodell (Eden, 1961)	17
5.3	Diffusions-limitierte Anlagerung (DLA) „diffusion limited aggregation“	17
<b>6.</b>	<b>Strukturbildung und Selbstorganisation: Zellularautomaten</b>	<b>18</b>
6.1	Eindimensionale Zellularautomaten	18
6.2	Zweidimensionale Zellularautomaten: Conway's „Game of life“	20
<b>7.</b>	<b>Simulation von Polymeren</b>	<b>21</b>
7.1	Der Self-Avoiding-Walk (SAW)	22
7.2	Verbesserungen des SAW	23
7.3	True Self-Avoiding-Walk (TSAW)	25
<b>8.</b>	<b>Referenzen</b>	<b>26</b>

## 1. Einleitung

Es gibt verschiedene Varianten und Modelle von Perkolationen. Allen gemeinsam ist die Zufallsplatzierung von Objekten in einem (d-dimensionalen) Raum.

Es lassen sich die folgenden Varianten unterscheiden

- *Kontinuumsperkolation*  
Die Kontinuumsperkolation kann verwendet werden, um
  - den Hohlraum in einem Fels zu bestimmen, um die Frage zu beantworten, ob der Fels wasserdurchlässig ist.
  - die Löcher in Schweizer Käse zu bestimmen, um festzustellen, ob der Käse verbunden ist, oder in mehrere Stücke zerfällt.
  - Gittermodelle zu berechnen.
  
- Gittermodelle
  - *Site-Perkolation*  
Jeder Gitterplatz ist mit Wahrscheinlichkeit  $p$  besetzt. Benachbarte Plätze sind verbunden.  
(z.B. Legierung aus zwei Atomsorten, Frage nach Leitfähigkeit; Waldbrände, Verbreitung von Krankheiten)
  
  - *Bond-Perkolation*  
Bindungen zwischen benachbarten Gitterpunkten werden mit Wahrscheinlichkeit  $p$  geknüpft (z. B. Leitwert eines Drahtgitters, Gel-Bildung in Flüssigkeiten).

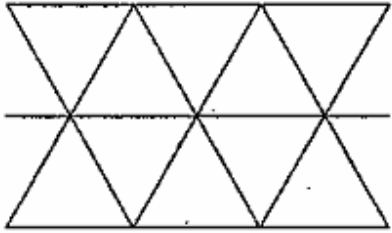
Typische Fragestellungen sind dabei

- Wie groß werden die Cluster in Abhängigkeit von  $p$ ?
- Sind die Ränder miteinander verbunden?
- Gibt es kritische Phänomene?  
(z.B. Clustergröße  $\rightarrow \infty$  bei  $p = p_c$ , kritische Punkte, Phasenübergänge)
- Skalierungsverhalten, fraktale Geometrie, Universalität?

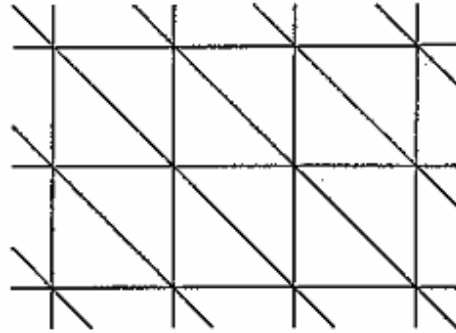
## 2. Simulation auf dem Computer

Zur Simulation auf dem Computer wird ein Gittermodell verwendet, bei dem jeder Gitterplatz den Zustand 0 hat, wenn er unbesetzt ist (Fels, Isolator), bzw. 1 wenn er besetzt ist (Hohlraum, Leitermaterial). Benachbarte Gitterplätze bilden einen Cluster.

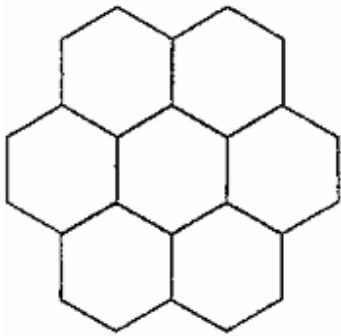
Ein System perkoliert, wenn mindestens ein durchgehender Cluster existiert.



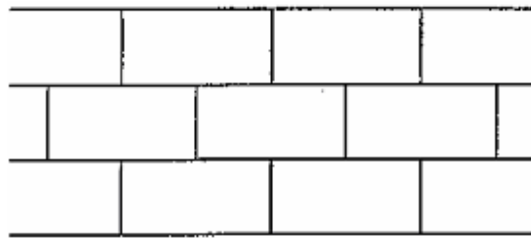
**Abbildung 1: Dreiecksgitter aus [1]**



**Abbildung 2: Implementierung des Dreiecksgitters im Computer aus [1]**



**Abbildung 3: Bienenwabengitter aus [1]**



**Abbildung 4: Implementierung am Computer aus [1]**

Gesucht ist nun ein effektiver Algorithmus zur Entscheidung, ob das System perkoliert oder nicht, wobei die Verfolgung einzelner Pfade keine sicheren Aussagen liefert.

## 2.1 Der Burning-Algorithmus

Eine mögliche Lösung dieses Problems stellt der Burning-Algorithmus dar. Dazu wird auf dem Gitter eine Art „Waldbrand“ simuliert.

Die Gitterzustände werden in drei Klassen unterteilt:

- 0: nicht entflammbar
- 1: brennbar
- 2: brennend, wobei das Feuer auf benachbarte brennbare Gitterplätze überspringen kann

### ***Ideenskizze des Burning-Algorithmus***

#### ***1. Schritt:***

Identifiziere besetzte Plätze (Zustand 1) an der unteren Feldkante und trage sie in eine Liste ein. Diese Punkte werden als brennend (Zustand 2) markiert. (Feuer wird an der unteren Feldkante gelegt.)

#### ***2. Schritt:***

Die Nachbarn der brennenden Plätze werden betrachtet. Ist unter den Nachbarn ein Platz brennbar (Zustand 1), so wird der Platz in eine neue Liste eingetragen und der Platz als brennend (Zustand 2) markiert. Die neue Liste entspricht der Front des Waldbrandes. Ist die Oberkante erreicht, perkoliert das System (und das Verfahren endet).

#### ***3. Schritt:***

Ist die Liste leer, perkoliert das System nicht (Ende). Ansonsten gehe zu Schritt 2.

Von Interesse ist nun, welcher Prozentsatz von Hohlraum, Leiter, brennbarem Material, ... erforderlich ist, damit das System perkoliert?

Dazu werden viele Systeme bei fester Wahrscheinlichkeit  $p$  untersucht und der Anteil  $P_{perk}$  der perkolierenden Systeme bestimmt. Anschließend wird die Wahrscheinlichkeit  $p$  variiert und  $P_{perk}$  als Funktion von  $p_i$  aufgetragen.

Für sehr große Systeme erfolgt ein scharfer Übergang zur Perkolation bei einem kritischen Wert (der Perkolationsschwelle)  $p = p_c$ .

$p_c$  hängt allein von der Art des Gitters und der Art der Perkolation (z.B. Site oder Bond) ab.

Gittertyp	Perkolationsschwelle $p_c$ (Site-Perkolation)
Quadratgitter	0,592745
Dreiecksgitter	0,5
Wabengitter	0,6962

Der Burning-Algorithmus liefert allerdings keine Information über die Anzahl der Cluster im Gitter und die Verteilung der Clustergrößen.

## 2.2 Der Hoshen-Kopelman-Algorithmus

Eine Verbesserung des Burning-Algorithmus stellt der Hoshen-Kopelman-Algorithmus dar. Er wird verwendet, um die Cluster zu nummerieren.

### Ideenskizze des Hoshen-Kopelman-Algorithmus

#### 1. Schritt:

Gehe zeilenweise durch das Gitter und markiere jeden besetzten Gitterpunkt durch ein Label (Nummer). Bei Berührung des Gitters mit bereits markierten Punkten vergib altes Label, sonst neues Label.

**Achtung:** Nummerierung ist fehlerhaft, wenn Cluster zusammenwachsen.

7	7		9	8	3	
7				8	3	3
	6				3	3
6	6	5	5	3	3	3
		5		4	3	3
1	1			4	3	3
1		2	2		3	

**Abbildung 5: fehlerhafte Nummerierung aus [2] (leicht verändert Übernommen)  
(Nummerierung der Cluster von links nach rechts und von unten nach oben)**

#### 2. Schritt:

Zweiter Durchgang durch das Gitter zur Überprüfung der Nummerierung (alle Nachbarn eines Gitterpunktes werden betrachtet). Es wird ein Indexregister  $np(i)$  erzeugt, das die korrekte

Nummer von Cluster  $i$  enthält.  $np(i)$  wird am Anfang auf 0 gesetzt, was für „keine

Veränderung“ steht. Um Rückkopplungen zu vermeiden definieren wir  $np(i) < i$ .

Kommt man an ein Feld mit zwei (unterhalb und links) unterschiedlichen Gitterzellen, wie z. B. in der Position (6, 2) so wird der Cluster mit dem größten Index in der Indexliste mit dem kleineren Wert verlinkt. Also  $np(4) = 3$ . Der Cluster 4 wird also dem Cluster 3 zugeordnet. Ebenso wird z.

B. an der Stelle (3, 4)  $np(6) = 5$  angegeben.

An der Position (5, 4) sollte aber nicht  $np(5) = 4$  gesetzt werden, da schon 4 mit 3 verlinkt ist.

Ein kurzer Nachschlag in der Indexliste führt auf den richtigen Wert  $np(5) = 3$ . Um Probleme im 3. Schritt zu verhindern, ist nach so einer Ersetzung immer ein Durchlauf der Indexliste notwendig, um eventuell doppelt verlinkte Cluster zu vermeiden. D. h.  $np(6) = 3$ .

3. Schritt:

Ersetze im Gitter alle (falschen) Nummern durch die korrekten Nummern  $np(i)$ , Vorschlag  $np(i) = i$ .

7	7		3	3	3	
7				3	3	3
	3				3	3
3	3	3	3	3	3	3
		3		3	3	3
1	1			3	3	3
1		2	2		3	

Abbildung 6: korrekte Nummerierung aus [2]

### 3. Perkolation

#### 3.1 Kritische Exponenten

Kritische Phänomene treten auf, zum Beispiel bei

- Phasenübergängen in der Thermodynamik (bei einer kritischen Temperatur  $T_c$ ).
- der Magnetisierung beim Ising-Modell (bei einer kritischen Temperatur  $T_c$ ).
- Clustergrößen in der Perkolation (bei einer Perkolationsschwelle  $p_c$ ).

Allgemein lässt sich sagen, dass kritische Phänomene bei langreichweitigen Korrelationen auftauchen.

Die Untersuchung der Clustergrößen ( $p < p_c$ ), bzw. Lochgrößen (bei  $p > p_c$ ) erfolgt durch den Hoshen-Kopelman-Algorithmus (siehe 2.2), mit dem sich die Cluster abzählen und identifizieren lassen.

Dabei lassen sich charakteristische Größen definieren:

- Der **Gyrationsradius**  $R_s$  eines Clusters aus  $s$ -Teilchen (besetzte Gitterpunkte) ist

$$\text{gegeben als } R_s^2 = \frac{1}{s} \sum_{i=1}^s (r_i - \bar{r})^2, \text{ mit } \bar{r} = \frac{1}{s} \sum_{i=1}^s r_i.$$

- Die **Korrelationslänge**  $\xi$  wird identifiziert mit dem Gyrationsradius des größten (nicht perkolierenden) Clusters (bei  $p > p_c$ ).  
(Hierzu gibt es auch alternative Definitionen.)

Im Limes unendlich großer Systeme (Gitter) gehorcht die Korrelationslänge  $\xi$  einem Potenzgesetz:

$$\xi(p) \sim |p - p_c|^{-\nu} \quad \text{mit } \nu = \begin{cases} \frac{4}{3} & \text{in 2 Dimensionen (d=2)} \\ 0,88 & \text{in 3 Dimensionen (d=3)} \end{cases}$$

- Die **mittlere Clustergröße** ist definiert als

$$S(p) \sim |p - p_c|^{-\gamma}, \quad \text{mit } \gamma = \begin{cases} 43/18 & ; d=2 \\ 1,8 & ; d=3 \end{cases}$$

- Die **Wahrscheinlichkeit**  $P_\infty(p)$ , die die Wahrscheinlichkeit angibt, mit der ein besetzter Gitterpunkt zum perkolierenden Cluster gehört, ist definiert als

$$P_\infty(p) = \frac{\text{Anzahl der Gitterpunkte im durchgehenden Cluster}}{\text{Gesamtzahl der besetzten Gitterpunkte}}$$

und berechnet sich nach

$$P_\infty(p) \sim |p - p_c|^\beta \quad \text{mit } \beta = \begin{cases} 5,36 & ; d=2 \\ 0,4\dots & ; d=3 \end{cases}$$

Die Exponenten  $\beta, \gamma, \nu$  werden als *kritische Exponenten* bezeichnet. Sie sind universelle Konstanten, unabhängig vom Gittertyp (Quadrat-, Dreiecks-, Wabengitter) und der Art der Perkolation (Site oder Bond).

### 3.2 Skalierungsgesetze und Renormierungstheorie

Zwischen den kritischen Exponenten gibt es Beziehungen, wie zum Beispiel  $2\beta + \gamma = \nu d$ . Diese Beziehungen werden *Skalierungsgesetze* genannt.

Ein tieferes physikalisches Verständnis der (numerischen) Resultate ergibt sich durch die *Renormierungstheorie* (K.T. Wilson, ~1971, Nobelpreis 1981). Ihre ursprüngliche Anwendung fand sie in der Thermodynamik, bei der Analyse der Phasenübergänge. Allerdings kann sie auch auf Perkolationen angewandt werden. Die Grundidee ist dabei (qualitativ), dass Gitterphänomene langreichweitigen Korrelationen entsprechen. Bei unendlich langen Korrelationen zeigt das System nun auf allen Skalen dasselbe physikalische Verhalten. Dies nennt man *Skaleninvarianz*.

### 3.3 Beispiel: Perkolation beim Quadratgitter

Eine Skalentransformation (Renormierung) ist möglich durch Zusammenfassung mehrerer Gitterpunkte ( $b \times b$ -Teilgitter, mit  $b = 1, 2, 3, \dots$ ) zu einem neuen Gitterpunkt. Der neue Gitterpunkt wird besetzt genau dann, wenn das  $b \times b$ -Teilgitter perkoliert.

Als einfachstes Beispiel wird ein  $2 \times 2$ -Teilgitter betrachtet. Dabei existieren 16 mögliche Konfigurationen.

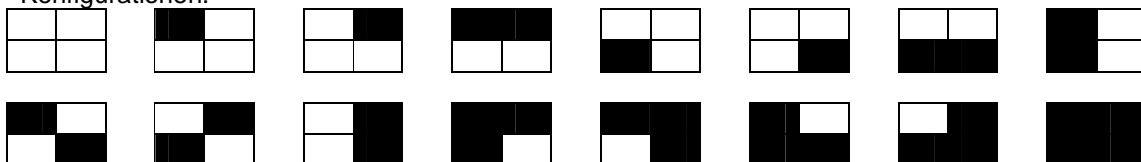


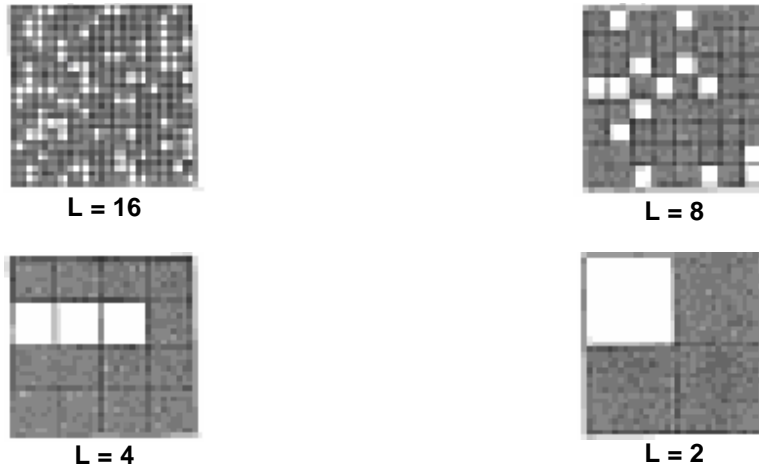
Abbildung 7: Konfigurationen beim  $2 \times 2$  - Quadratgitter

Von diesen Konfigurationen perkolieren 7.



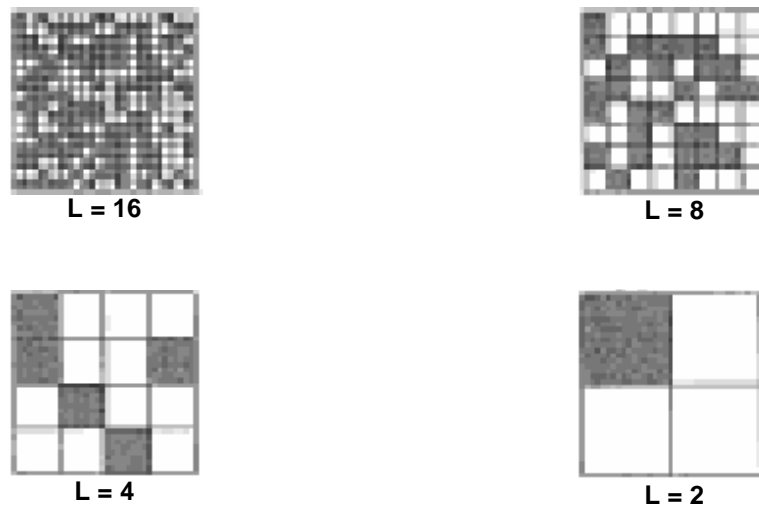
**Beispiele**

a)  $p > p_c$ : Auf größerer Skala wächst der Anteil besetzter Gitterpunkte ( $p \rightarrow 1$ )



**Abbildung 8: Perkolation, erzeugt mit  $p = 0,7$  aus [2]**

b)  $p < p_c$ : Auf größerer Skala sinkt der Anteil besetzter Gitterpunkte ( $p \rightarrow 0$ )



**Abbildung 9: Perkolation, erzeugt mit  $p = 0,5$  aus [2]**

**Frage:** Für welche Besetzungswahrscheinlichkeit  $p$  bleibt der Anteil besetzter Gitterpunkte invariant unter Skalentransformationen?

Jeder Gitterpunkt in  $b \times b$ -Gitter wird besetzt mit Wahrscheinlichkeit  $p$ . Daraus ergibt sich die Besetzungswahrscheinlichkeit  $p'$  für den neuen größeren Gitterpunkt zu

$$p' = R(p) = \sum_{n=0}^{n=N=b^2} \underbrace{S_n}_{\text{Anteil der perkolierenden Gitter mit } n \text{ besetzten Gitterpunkten}} \underbrace{\binom{N}{n} p^n (1-p)^{N-n}}_{\text{Binomialverteilung. Wahrscheinlichkeit, dass bei einer Besetzungswahrscheinlichkeit } p \text{ genau } n \text{ von } N=b^2 \text{ Gitterplätze besetzt sind}} .$$

Gesucht sind die Fixpunkte der Renormierungstransformation  $p = R(p)$ .

Zur Formel

$$p' = R(p) = \sum_{n=0}^{n=N=b^2} S_n \binom{N}{n} p^n (1-p)^{N-n}$$

- $R(p)$  gibt die Wahrscheinlichkeit an, mit der ein Gitterplatz auf dem größeren Gitterplatz besetzt ist. Um das zu bestimmen, müssen alle möglichen Konfigurationen beachtet werden. Dabei ist darauf zu achten, dass mit einer Besetzungswahrscheinlichkeit  $p$  nicht unbedingt genau  $p \cdot N$  Plätze belegt sind.
- Diese Verteilung ist die Binomialverteilung  $\binom{N}{n} p^n (1-p)^{N-n}$ .

Auf das Gitter bezogen gibt die Binomialverteilung an, wie hoch die Wahrscheinlichkeit ist, dass bei einer Besetzungswahrscheinlichkeit  $p$  und  $N$  Feldern genau  $n$  Felder besetzt werden.

Das entspricht der klassischen Urne, in der sich schwarze und weiße Kugeln befinden und schwarze mit Wahrscheinlichkeit  $p$  gezogen werden. Bei  $N$  Versuchen (mit Zurücklegen) gibt die Verteilung nun an, wie hoch die Wahrscheinlichkeit ist genau  $n$  schwarze Kugeln zu bekommen.

- Die  $S_n$  geben die Wahrscheinlichkeit an, dass bei  $n$  besetzten Gitterplätzen das Gitter perkoliert.
- Multipliziert mit der Binomialverteilung ist dies die Wahrscheinlichkeit, dass das Gitter mit Besetzungswahrscheinlichkeit  $p$  perkoliert, obwohl  $n$  Gitterplätze besetzt sind.
- Jetzt wird über alle Möglichkeiten, d.h. über alle  $n$ , summiert. Und das ist die Wahrscheinlichkeit, dass das größere Gitter bei Besetzungswahrscheinlichkeit  $p$  perkoliert.

Triviale Fixpunkte sind  $p = 0$  und  $p = 1$ , gesucht ist ein nicht trivialer Fixpunkt  $0 < p < 1$ .

Für  $b \rightarrow \infty$  liefert die Fixpunktgleichung die exakte Perkolationsschwelle  $p_c = 0,5927\dots$  für kleine  $b$  ergibt sich eine Näherung an  $p_c$ .

Die grösste Abschätzung für  $b = 2, N = 4$  ist  $S_0 = S_1 = 0, S_2 = \frac{1}{3}$  und  $S_3 = S_4 = 1$ .

Daraus ergibt sich die Renormierungstransformation  $R(p)$  zu

$$R(p) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2$$

Die Lösungen der Fixpunktgleichung  $R(p^*) = p^*$  sind

$$p^* = 0, p^* = 1 \text{ und } p^* = \frac{1}{2}(-1 \pm \sqrt{5}) = \begin{cases} 0,61803 \approx p_c \\ -1,61803 \end{cases}$$

Für ein Dreiecksgitter ( $b = 2, N = 4$ ) lassen sich die  $S_n$  abschätzen zu

$$S_0 = S_1 = 0, S_2 = \frac{1}{2} \text{ und } S_3 = S_4 = 1.$$

Die Renormierungstransformation ist

$$R(p) = p^4 + 4p^3(1-p) + 3p^2(1-p)^2$$

mit den Lösungen

$$p^* = 0, p^* = 1 \text{ und } p^* = \frac{1}{2} = p_c.$$

Besonders ist bei diesem Gitter, dass der nicht triviale Fixpunkt exakt der Perkolationschwelle entspricht.

### 3.4 Monte-Carlo Renormierungsgruppentheorie

Für große  $b$  wird die exakte Bestimmung der  $S_n$  zu aufwendig, da sie die Überprüfung von  $2^{b^2}$  Konfigurationen erfordert. Die Abhilfe von diesem Problem stellt die Bestimmung der  $S_n$  durch Monte-Carlo Methoden dar.

#### Ideenskizze der Monte-Carlo Renormierungsgruppentheorie

Die Plätze im  $b \times b$ -Gitter werden zufällig besetzt. Ein Teilchen wird addiert und es wird getestet, ob das System perkoliert.

Falls Perkolations bei  $S$  besetzten Plätzen stattfindet, wird  $S_n = S_n + 1$  gesetzt für alle  $n = S, S + 1, \dots, N = b^2$ .

Dieser Vorgang wird genügend oft wiederholt.

Zum Schluss werden die  $S_n$  normiert und in die Fixpunktgleichung

$$p = R(p) = \sum_{n=0}^{n=N=b^2} S_n \binom{N}{n} p^n (1-p)^{N-n}$$

eingesetzt mit der Näherung  $\binom{N}{n} p^n q^{N-n} \approx (2\pi Npq)^{-\frac{1}{2}} e^{-\frac{(n-pN)^2}{2Npq}}$ .

### 3.5 Berechnung kritischer Exponenten

Die Korrelationslänge ist definiert (siehe 3.1) als  $\xi(p) \sim |p - p_c|^{-\nu}$ .

Eine Skalentransformation (Renormierung) ergibt  $\xi' = \frac{\xi}{b}$ .

Für die Korrelationslänge gilt  $\xi(p) = \text{const.} \left| p - \underbrace{p^*}_{\text{Fixpunkt}} \right|^{-\nu}$ .

Mit  $\xi' = \frac{\xi}{b}$  folgt daraus  $\xi'(p) = \xi(p') = \text{const.} \cdot |p' - p^*|^{-\nu} = \frac{1}{b} \xi(p) = \frac{1}{b} \text{const.} \cdot |p - p^*|^{-\nu}$ ,

was auf  $|p' - p^*|^{-\nu} = \frac{1}{b} |p - p^*|^{-\nu}$

führt mit  $p' - p^* = R(p) - R(p^*)$ .

Für  $|p - p^*| \ll 1$  lässt sich dies abschätzen als  $p' - p^* \approx \lambda(p - p^*)$  mit  $\lambda = \left. \frac{dR}{dp} \right|_{p=p^*}$ .

Daraus folgt  $|p - p^*|^{-\nu} = \lambda^{-\nu} |p - p^*|^{-\nu} = \frac{1}{b} |p - p^*|^{-\nu}$ ,

was auf  $-\nu \ln \lambda = -\ln b$  führt, also  $\nu = \frac{\ln b}{\ln \lambda}$ .

Die einfachste und grösste Abschätzung für  $b = 2$  ist

$$R(p) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2 = -p^4 + 2p^2,$$

bzw.  $p^* \approx 0,618084$ .

Dies führt auf  $\lambda = \left. \frac{dR}{dp} \right|_{p=p^*} = -4(p^*)^3 + 4p^* \approx 1,5279$ , was  $\nu = 1,635$  bedeutet

(die exakte Lösung ist  $\nu = \frac{4}{3}$ ).

## 4. Exkurs: Fraktale

Die Masse  $M(R)$  eines Clusters, also das Volumen bzw. die Fläche eines besetzten Gebietes bei der Perkolation weist eine fraktale Dimension  $d_f$  auf.

Die Erwartung ist eigentlich, dass gilt  $M(R) \sim R^d$  mit  $R$ : Gyrationradius,  $d$ : Dimension. Dann ist das Objekt kompakt und die mittlere Dichte  $\bar{\rho}$  ist konstant.

Tatsächlich tritt bei Perkolation, insbesondere bei  $p \approx p_c$  ein fraktales Verhalten auf, so dass gilt

$$M(R) \sim R^{d_f},$$

mit  $d_f < d$  (nicht ganzzahlig), bzw.  $d_f$ : fraktale Dimension.

### 4.1 Definitionen

Betrachte eine Punktmenge in  $d$  Dimensionen und überdecke sie mit  $d$ -dimensionalen Kugeln mit Durchmesser  $\varepsilon$ . Sei  $N(\varepsilon)$  die Anzahl der zur Überdeckung notwendigen Kugeln.

Dann gilt  $N(\varepsilon) \sim \varepsilon^{-d}$  für  $\varepsilon \rightarrow 0$ , mit  $d$ : Dimension der Punktmenge.

Die Größe  $d = -\lim_{\varepsilon \rightarrow 0} \frac{\ln N(\varepsilon)}{\ln \varepsilon}$  heißt **Hausdorff-Dimension**.  $d$  muss nicht ganzzahlig sein.

Mengen mit nicht-ganzzahliger Hausdorff-Dimension heißen **Fraktale**.

### 4.2 Beispiele

#### Cantor-Menge

$$\begin{array}{l} \text{-----} \quad n=0, \quad l_0=1 \\ \text{-----} \quad \text{-----} \quad n=1, \quad l_1=\frac{2}{3} \\ \text{--- ---} \quad \text{--- ---} \quad n=2, \quad l_2=\left(\frac{2}{3}\right)^2 = \frac{4}{9} \\ \dots \end{array}$$

Die Überdeckung erfolgt in Stufen  $N(\varepsilon_n) = 2^n$ , mit Intervallen der Länge  $\varepsilon_n = \left(\frac{1}{3}\right)^n$ .

Die Hausdorff-Dimension ist  $D = -\lim_{n \rightarrow \infty} \frac{\ln N(\varepsilon)}{\ln \varepsilon_n} = \frac{\ln 2}{\ln 3} = 0,6309$

## Kochsche Kurve

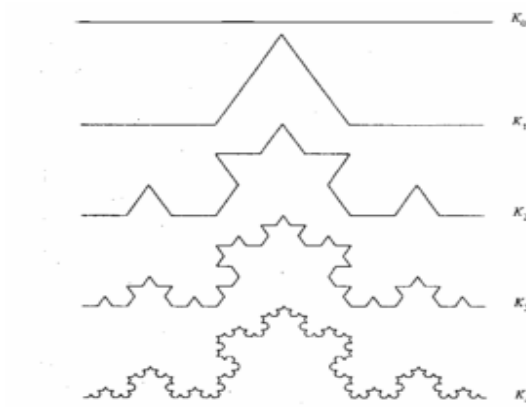


Abbildung 10: Bildung einer Kochkurve

$$l_0 = 3, l_1 = 3 \frac{4}{3}, \dots, l_n = 3 \left( \frac{4}{3} \right)^n$$

Die Überdeckung erfolgt in Stufen durch  $3 \cdot 4^n$  Intervalle der Länge  $\varepsilon_n = \left( \frac{1}{3} \right)^n$ . Es ist

$N(\varepsilon_n) = 3 \cdot 4^n$ , woraus sich die Hausdorff-Dimension ergibt zu

$$D = -\lim_{n \rightarrow \infty} \frac{\ln N(\varepsilon)}{\ln \varepsilon_n} = -\lim_{n \rightarrow \infty} \frac{\ln(3 \cdot 4^n)}{\ln \left( \frac{1}{3} \right)^n} = \frac{\ln 4}{\ln 3} = 1,26\dots$$

## 5. Wachstumsmodelle und Fraktale

Bei der Besetzung von Gitterpunkten mit der Wahrscheinlichkeit  $p$  entstehen i.A. viele Cluster unterschiedlicher Größe.

### 5.1 Algorithmus von Hammersley, Leath und Alexandrowicz

Mit diesem Algorithmus soll ein einzelnes Perkolationsmusters erzeugt werden.

1. Besetze beliebigen Startpunkt auf dem Gitter und markiere die 4 Nachbarn mit P (perimeter sites/Randpunkte).
2. Wähle zufälligen Randpunkt und besetze ihn mit Wahrscheinlichkeit  $p$ , sonst markiere ihn mit X: „tested unoccupied site“.
3. Markiere neue Randpunkte (P) und wiederhole 2. nach Belieben, oder bis keine Randpunkte mehr vorhanden sind.

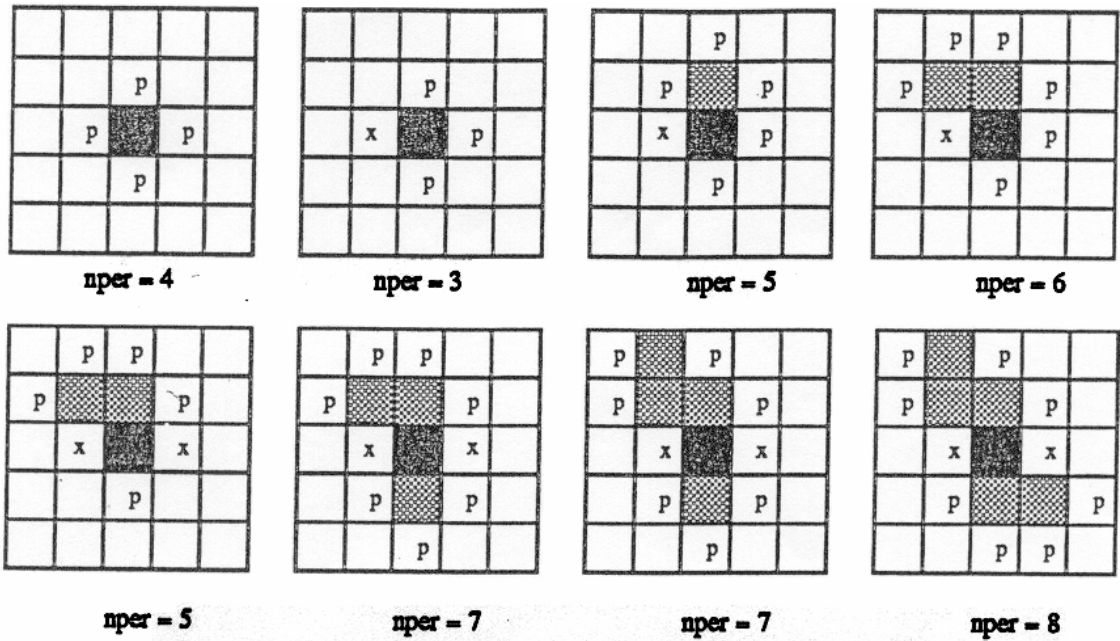


Abbildung 11: Algorithmus von Hammersley, Leath und Alexandrowicz aus [2]

**Anwendung des Algorithmus:** Wachstum eines Perkolationsclusters

Es gilt  $M(R) \sim R^{d_f}$ , was auf  $\ln M = \underbrace{d_f}_{\substack{\text{Steigung bei} \\ \text{logarithmischer} \\ \text{Auftragung}}} \ln R + \text{const.}$  führt.

Für  $p = p_c = 0,5927$  ist  $d_f = 1,85 \pm 0,10$ .

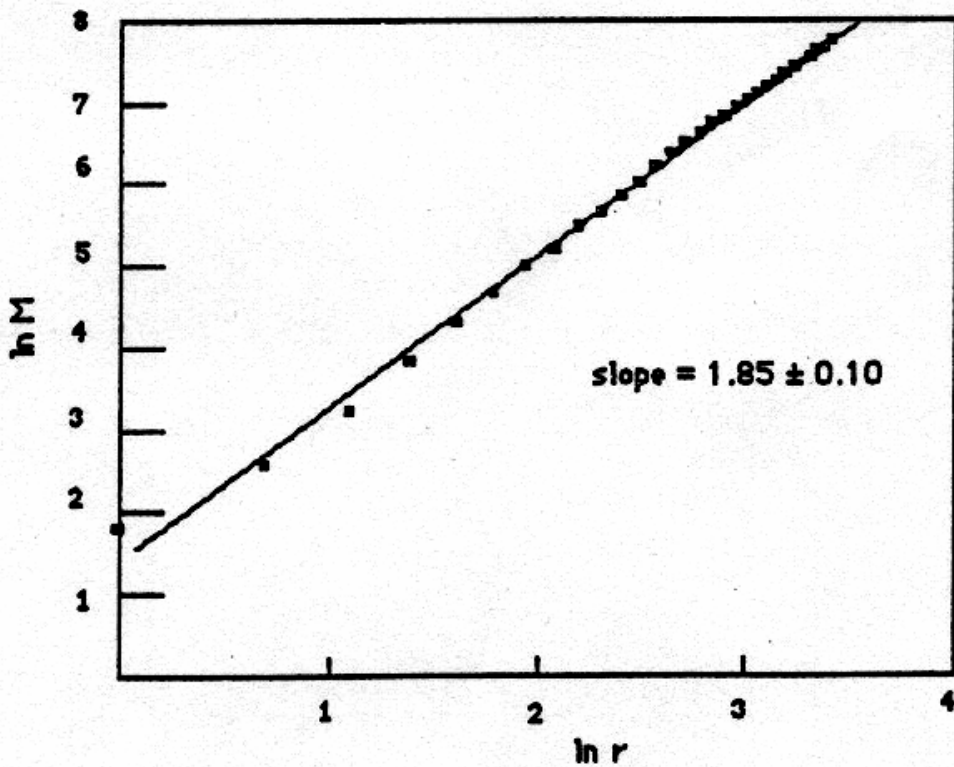


Abbildung 12:  $\ln(M)$  zu  $\ln(R)$  bei  $p = 0,5927$  auf einem  $61 \times 61$ -Gitter aus [2]

Der exakte Wert für  $d_f$  bei einem Perkolationscluster ( $p = p_c = 0,5927$ ) ist

$$d_f = \frac{91}{48} \approx 1,896.$$

Der **Zusammenhang** zwischen **fraktaler Dimension** (bei der Perkolationschwelle) und den **kritischen Exponenten** (Skalengesetz) ist

$$d_f = d - \frac{\beta}{\nu}.$$

Skizze der Begründung

$$\underbrace{P_\infty(L)}_{\substack{\text{wahrscheinlichkeit, dass} \\ \text{ein besetzter Gitterpunkt zum} \\ \text{perkolierenden Cluster gehört}}} = \frac{M(L)}{p \underbrace{L^d}_{\substack{\text{Anzahl der} \\ \text{Gitterplätze}}}} \Rightarrow M(L) \sim P_\infty L^d$$

$$\text{mit } L \rightarrow \infty, p \rightarrow p_c \Rightarrow P_\infty(p) \sim |p - p_c|^\beta, \xi(p) \sim |p - p_c|^{-\nu} \sim L$$

$$\Rightarrow |p - p_c| \sim L^{-\frac{1}{\nu}}$$

$$\Rightarrow P_\infty(p = p_c) \sim L^{-\frac{\beta}{\nu}}$$

$$\Rightarrow M(L) \sim L^{\frac{d-\beta}{\nu}}$$

$$\Rightarrow d_f = d - \frac{\beta}{\nu}$$



## 5.2 Das Edenmodell (Eden, 1961)

Das Modell wurde am Deutschen Krebsforschungszentrum zur Simulation von Tumorwachstum entwickelt.

1. Besetze beliebigen Startpunkt ( $U_r$ -Cluster) und markiere die vier Nachbarn (Randpunkte).
2. Besetze zufällig gewählten Randpunkt („Wachstum des Tumors“)
3. Markiere neu entstehende Randpunkte und wiederhole 2.

### **Bemerkungen**

- Das Edenmodell entspricht dem Wachstum eines einzelnen Perkolationsclusters (Algorithmus von Hammersley, Leath und Alexandrowicz) mit Besetzungswahrscheinlichkeit  $p = 1$ .
- Für die Masse gilt  $M(R) \sim R^2$  ( $d_f = 2$ ). Sie ist also kompakt und **kein** Fraktal
- Ein modifiziertes Eden-Wachstumsmodell, in dem Immunitätsraten  $\bar{p}$  eingebaut sind (benachbarte Gitterpunkte werden mit Wahrscheinlichkeit  $p = 1 - \bar{p}$  besetzt) entspricht dem Wachstum eines einzelnen Perkolationsclusters. Dabei sind fraktale Strukturen möglich, z.B. bei  $\bar{p} \approx 0,4 \hat{=} p = 1 - \bar{p} \approx 0,6 \approx p_c$ .

## 5.3 Diffusions-limitierte Anlagerung (DLA) „diffusion limited aggregation“

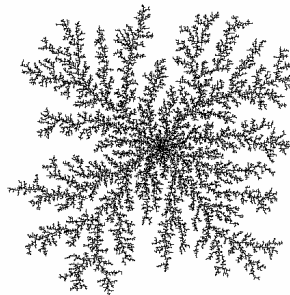
Im Modell wird auf einem Gitter ein Keim gesetzt, an dem sich Teilchen anlagern können. Die Teilchen in der Umgebung vollführen eine Brownsche Bewegung (Random Walk). Die Teilchen können sich an einem Keim anlagern.

Daraus ergeben sich die Fragen nach den Formen oder Strukturen, die sich ausbilden, oder wie die Ausdehnung des Objektes (der Gyrationradius  $R_g$ ) mit der Zahl der angelagerten Teilchen (Atome, Moleküle) wächst.

$$M(R_g) \sim R_g^{d_f}, \text{ mit } R_g = \sqrt{\frac{1}{N} \sum_{i=1}^N (r_i - \langle r \rangle_N)^2}, \quad \langle r \rangle_N = \frac{1}{N} \sum_{i=1}^N r_i$$

Auf dem Computer erfolgt die Simulation von DLA über Gittermodelle, auf denen ein Random Walk stattfindet.

Ein Teilchen (Random Walk) wird in genügendem Abstand vom Aggregat gestartet. Falls das Teilchen das System (Gitter) verlässt, wird ein neues Teilchen gestartet. Falls das Teilchen das Cluster berührt, lagert es sich an (der Keim wächst) und ein neues Teilchen wird gestartet.



**Abbildung 13: Diffusions-limitierte Anlagerung (DLA)**

## 6. Strukturbildung und Selbstorganisation: Zellularautomaten

Wachstumsmodelle, wie z.B. Perkolation (siehe 3.) oder DLA (siehe 5.3) zeigen eine Strukturbildung, bei der auch fraktale Strukturen und Selbstähnlichkeit möglich sind. Bei Perkolation und DLA erfolgt die Simulation mit Zufallszahlen (Besetzungswahrscheinlichkeiten, Randwerte).

Andere Modelle für Strukturbildung und Selbstorganisation sind Zellularautomaten (von Neumann, Ulam, 1948). Sie benutzen deterministische Regeln für die Entwicklung eines (zufälligen) Anfangszustandes (eines Gitters).

Die ursprüngliche Zielsetzung (von Neumann, Ulam) war die Erschaffung eines idealisierten Modelles für biologische Selbstproduktion, oder allgemeiner ein Modell für Strukturbildung aus zufälligen Anfangsbedingungen bei einer Diskretisierung von Raum (Gitter) und Zeit (Iterationen).

Die Charakteristika von Zellularautomaten sind

- Der Zustand jeder Zelle (Gitterpunkte) wird in einer Folge diskreter Zeitschritte aktualisiert.
- Die Variablen aller Zellen werden gleichzeitig (synchron) aktualisiert, abhängig von den Werten der Variablen zum vorherigen Zeitschritt.
- Die neuen Werte der Zellen hängen nur ab von den Werten der Zellen in der lokalen Umgebung.

### 6.1 Eindimensionale Zellularautomaten

Bei einem eindimensionalen Zellularautomat ist jede Zelle im Zustand 0 oder 1. Ihr Wert im nächsten Zeitschritt ist abhängig vom Wert der Zelle und den beiden Nachbarn. Daraus ergeben sich  $2^3 = 8$  mögliche Kombinationen von 3 Zellen, woraus sich  $2^8 = 256$  mögliche Regeln ergeben.

**Beispiel:** Regel 90

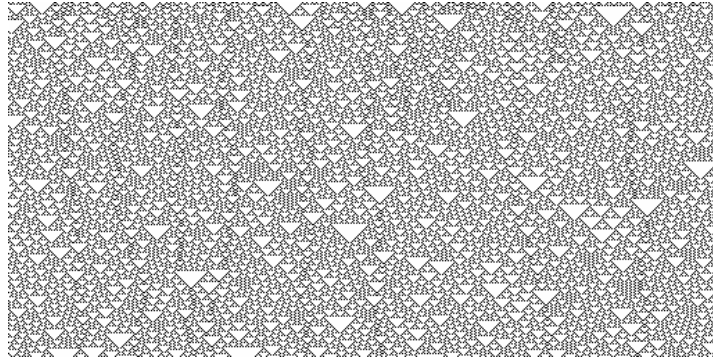
Zustand 1 wenn genau eine der beiden nächsten Nachbarzellen den Wert 1 hat (exklusiv oder).

t	111	110	101	100	011	010	001	000
t+1	0	1	0	1	1	0	1	0

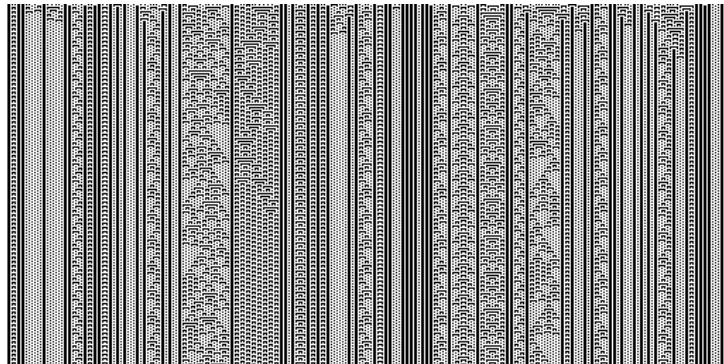
Der Name dieser Regel erklärt sich aus

110	⇒	6	$2^6$	64
100	⇒	4	$2^4$	16
011	⇒	3	$2^3$	8
001	⇒	1	$2^1$	2
				<b>90</b>

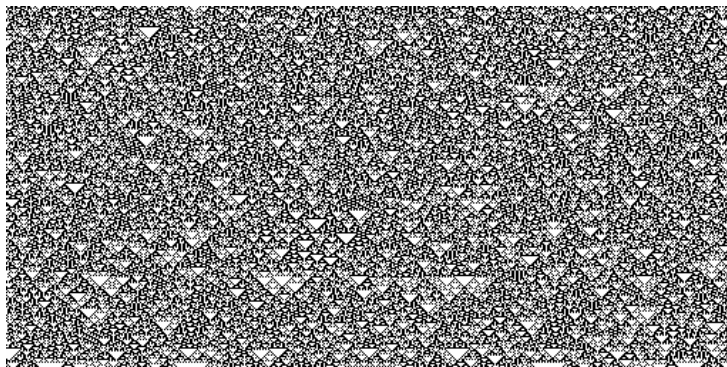
Werden diese Regeln auf ein eindimensionales Gitter  $f(i)$ ,  $i = 1, \dots, N$  (N Zellen) angewandt, so kann eine periodische Fortsetzung eingebaut werden, indem  $f(0) = f(N)$  und  $f(N+1) = f(1)$  gesetzt wird.



**Abbildung 14: Regel 18**



**Abbildung 15: Regel 73**



**Abbildung 16: Regel 90**

## 6.2 Zweidimensionale Zellularautomaten: Conway's „Game of life“

Ein Beispiel für zweidimensionale Zellularautomaten ist Conway's „Game of life“ (John Conway, 1970), das eine Populationsdynamik simuliert.

In einem zweidimensionalen Gitter haben Zellen den Wert 0 oder 1.

### Regel

Die Werte der acht nächsten Nachbarzellen (oben, unten, links, rechts, diagonal) werden addiert,

so dass sich als Summe ergibt  $k = \sum_{(i',j')=nn(i,j)} f(i',j')$ .

Die Dynamik ergibt sich durch die Funktion  $f(i,j)$

$$f'(i,j) = \begin{cases} 0 & , k < 2 \text{ oder } k > 3 \text{ (Vereinsamung, Überbevölkerung)} \\ f(i,j) & , k = 2 \\ 1 & , k = 3 \end{cases}$$

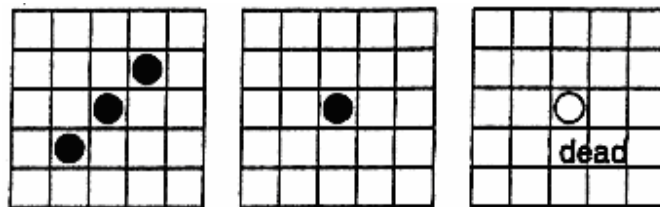


Abbildung 17: Aussterbende Figuren im Game of Life aus [2]

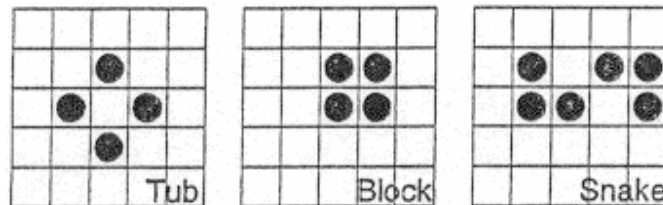


Abbildung 18: Stabile Figuren im Game of Life aus [2]

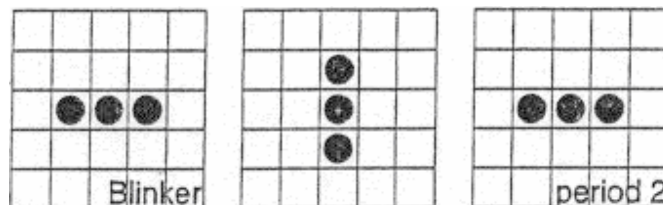


Abbildung 18: Periodische Figuren im Game of Life aus [2]

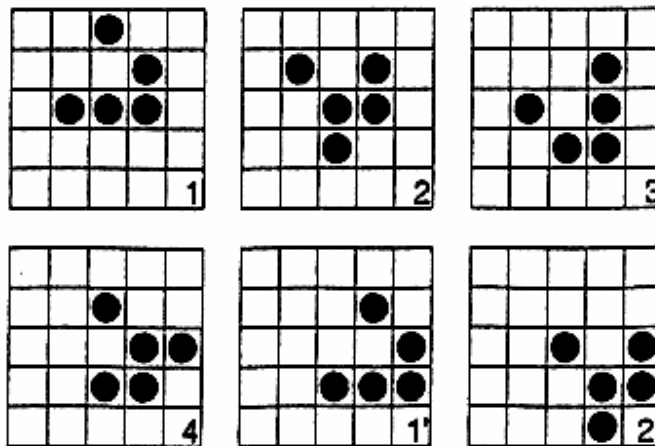
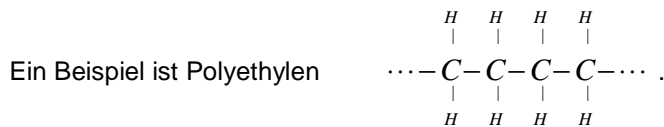


Abbildung 19: Vier Phasen des Gleiters, verbunden mit einer Bewegung nach rechts unten aus [2]

## 7. Simulation von Polymeren

Ein Polymer ist eine Molekülkette, die aus  $N$  Bausteinen (Monomere) zusammengesetzt ist, mit  $N \sim 10^3 - 10^5$ .



Sehr lange Ketten müssen nicht linear sein, sondern können Spaghettiknäuel bilden, z.B. Polymerketten in Lösungsmittel  $\hat{=}$  Spaghetti in warmem Wasser.

Viele Anwendungen erfordern detaillierte Kenntnisse über die Struktur der Polymere. Es gibt jedoch globale Eigenschaften von Polymeren, die unabhängig sind von der detaillierten Struktur der Kette, z.B. kollabieren Ketten nicht zu winzigen Bällen oder nehmen eine lineare Gestalt (maximale Länge) an, sondern sie bilden eine zufällige Struktur.

Zur Simulation von Polymeren auf dem Computer wird die Kette auf einem Gitter approximiert.

Es stellt sich wieder die Frage nach der Größe (oder besser der Ausdehnung) eines aus  $N$  Bausteinen bestehenden Polymers.

**Definition:** 
$$R_N = \sqrt{\langle r^2 \rangle_N - \langle r \rangle_N^2} = \sqrt{\frac{1}{N} \sum_{i=1}^N (r_i - \langle r \rangle_N)^2}$$

(vgl. Gyrationradius von Perkollationsclustern oder Aggregaten bei DLA)

Es gilt:  $R_N \sim aN^\nu$  mit  $\nu = 0,6$  ( $d = 3$ ), bzw.  $\nu = \frac{3}{4}$  exakt in zwei Dimensionen,

$a$  ist abhängig von der makroskopischen Struktur des Polymers und vom Lösungsmittel.

Die einfachste Idee zur Simulation eines Polymers ist ein Random Walk auf einem Gitter. Der Startpunkt ist der Ursprung und der gesamte Walk wird als Polymerkette angesehen.

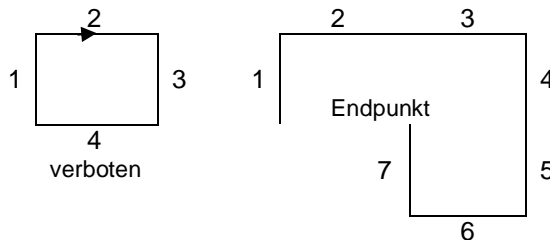
Der so entstehende Random Walk beschreibt eine Diffusionsbewegung  $\langle r_i^2 \rangle \sim i$ , was auf

$$R_N \sim \sqrt{N}, \text{ also } \nu = \frac{1}{2} \text{ f\u00fchrt.}$$

Dies steht im Widerspruch zur Realit\u00e4t (Beobachtung).

### 7.1 Der Self-Avoiding-Walk (SAW)

Der Fehler im obigen Modell (siehe 7.) liegt darin, dass sich Polymerketten nicht selbst schneiden k\u00f6nnen. F\u00fcr ein verbessertes Modell wird deshalb der Random Walk (RW) durch einen „self-avoiding walk“ (SAW) ersetzt. Dabei darf jeder Gitterpunkt h\u00f6chstens einmal besucht werden.

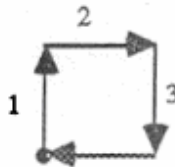


**Abbildung 20: Self-Avoiding Walk (SAW)**

Der SAW kann durch eine Monte-Carlo-Simulation geschehen.

#### Monte-Carlo-Simulation der zweidimensionalen SAW

1. W\u00e4hle Startpunkt und einen ersten Schritt (z.B. Richtung Norden).
2. W\u00e4hle eine von drei m\u00f6glichen Richtungen aus (siehe Zufallszahl  $\in \{1, 2, 3, 4\}$ ) und besetze den entsprechenden Gitterpunkt, falls er noch unbesetzt ist und wiederhole Schritt 2.  
Falls der Gitterpunkt besetzt ist, starte einen neuen SAW bei Schritt 1. Berechne die Daten der Polymere aus N Bausteinen (z.B.  $R_N$ ) zur Erstellung von Statistiken (z.B. Auftragung  $\log R_N$  \u00fcber  $\log N$  zur Bestimmung von  $\nu$ ).



**Abbildung 21: verbotener Walk aus [2]**

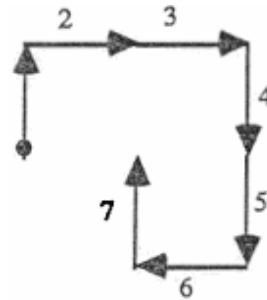


Abbildung 22: SAW, der sich im 8. Schritt schneidet aus [2]

### Probleme des Algorithmus

- Die meisten SAW werden bereits bei kurzen Kettenlängen abgebrochen, so dass es wenig Informationen über lange Ketten (N groß) gibt.
- Der Algorithmus ist nicht effizient. Der Anteil erfolgreicher Versuche für einen SAW der Länge N sinkt exponentiell mit N.

## 7.2 Verbesserungen des SAW

- Methode von M.N. Rosenbluth und A.W. Rosenbluth (1955)

Der Algorithmus wird genauso wie beim SAW ausgeführt, jedoch wird bei jedem Schritt der Faktor  $W_N$  ( $0 \leq W_N \leq 1$ ) bestimmt, der die Gewichtung des Schrittes angibt.

Bei  $W_1 = 1$  sind alle drei möglichen Richtungen erlaubt.

Allgemein ist  $W_N = \frac{m}{3} W_{N-1}$ , mit  $m \in \{0, 1, 2, 3\}$  Anzahl der erlaubten Richtungen.

Für  $m = 0$  endet der SAW.

Bei  $m = 1, 2, 3$  wird per Zufallszahl eine der erlaubten Richtungen ausgewählt.

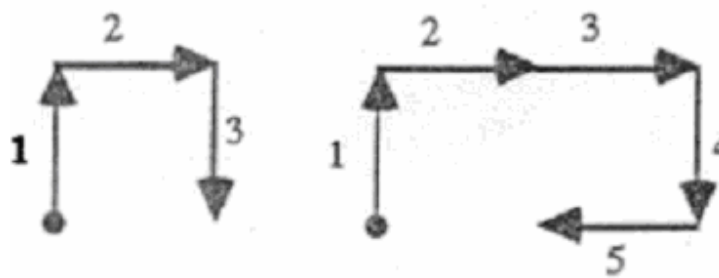


Abbildung 23: Beispiele der Gewichte nach der Rosenbluth-Methode aus [2]

Links:  $W_{1,2,3}=1, W_4=2/3$

Rechts:  $W_{1,2,3,4,5}=1, W_6=1/3$

Die Gewichte werden beispielsweise bei der Berechnung von  $R_N$  berücksichtigt

$$\langle R_N^2 \rangle = \frac{\sum_i W_{N,i} R_{N,i}^2}{\sum_i W_{N,i}}, \text{ mit } i: \text{ Index über alle Versuche.}$$

- **Reptationsmodell**

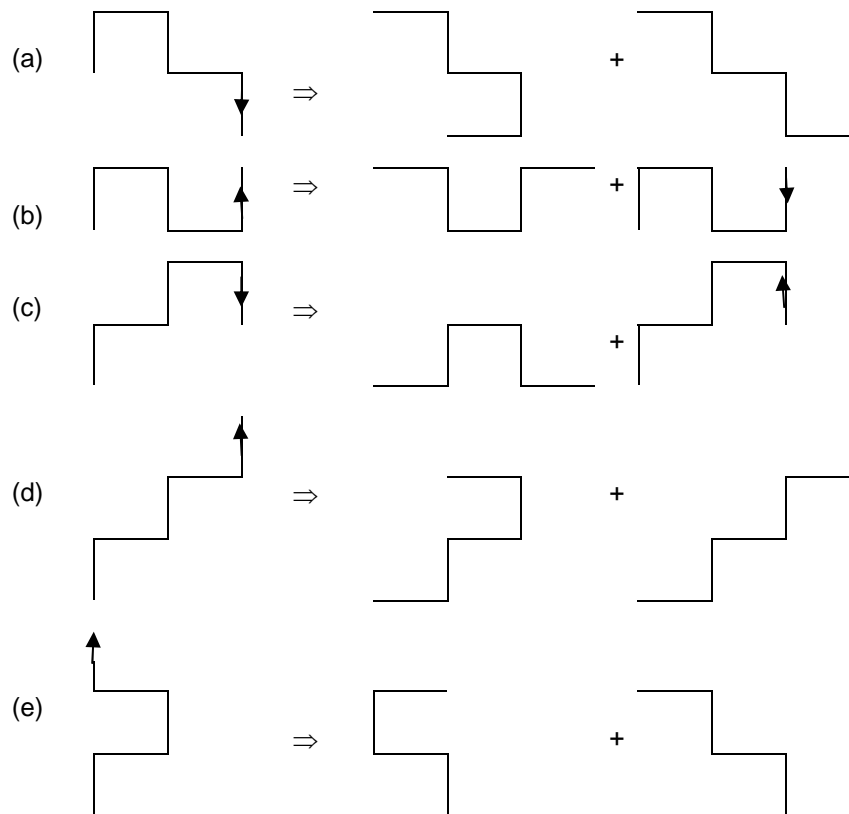
Eine Sequenz von Ketten mit Länge  $N$  wird durch Variation der Startkette erzeugt.

**Rezept für Reptation**

1. Wähle eine zufällige Kette und entferne das erste Kettenglied am Kettenanfang.
2. Versuche, ein Kettenglied am Kettenende anzuhängen (es gibt maximal zwei Möglichkeiten).
3. Falls sich kein Kettenglied ohne Selbstkreuzung anfügen lässt, nehme die Ausgangskette und vertausche die Nummerierung der Kettenglieder (Pfeilrichtung rückwärts). Diese Kette wird in den Statistiken berücksichtigt. Wiederhole die Schritte zur Erzeugung einer genügend großen Anzahl von Ketten für statistische Auswertungen (z.B.  $R_N$ ).

**Beispiel**

$N = 5$ , zur Vereinfachung sind alle Bindungswinkel  $90^\circ$ .  
Anwendung des Rezeptes auf 5 mögliche unabhängige Startketten.



**Abbildung 24: Reptation beim SAW**

Die verschiedenen Ketten lassen sich dabei auseinander konstruieren.

$$(a) \rightarrow \frac{1}{2}(c) + \frac{1}{2}(d)$$

$$(b) \rightarrow \frac{1}{2}(e) + \frac{1}{2}(b)$$



$$(c) (c) \rightarrow \frac{1}{2}(e) + \frac{1}{2}(a)$$

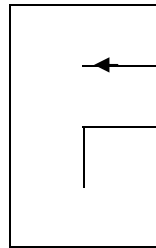
$$(d) (d) \rightarrow \frac{1}{2}(c) + \frac{1}{2}(d)$$

$$(e) (e) \rightarrow \frac{1}{2}(b) + \frac{1}{2}(a)$$

**Die Transformationen bei der Reptation erhalten die statischen Gewichte der verschiedenen Ketten.**

Es taucht allerdings das Problem auf, dass nicht alle Konfigurationen erreicht werden können.

Zum Beispiel kann die folgende Kette bei einem normalen SAW mit einem Winkel von  $90^\circ$  oder  $180^\circ$  nicht aus der Reptationstransformation einer anderen Kette erhalten werden.



**Abbildung 25: Beispiel einer double cul-de-sac Konfiguration, die nicht durch die Reptationsmethode gewonnen werden kann aus [2]**

Allerdings treten solche Konfigurationen nur sehr selten auf und der Fehler kann deshalb vernachlässigt werden.

### 7.3 True Self-Avoiding-Walk (TSAW)

Es wird ein Random Walk mit Nebenbedingung gestartet, so dass das Aufsuchen von Zellen mit einer bestimmten Wahrscheinlichkeit vermieden wird, die davon abhängt, wie oft die Zelle bereits in der Vergangenheit besucht wurde.

Der TSAW führt zu einer Reduktion von Selbstkreuzungen der Kette, aber nicht zur vollkommenen Vermeidung wie beim SAW.

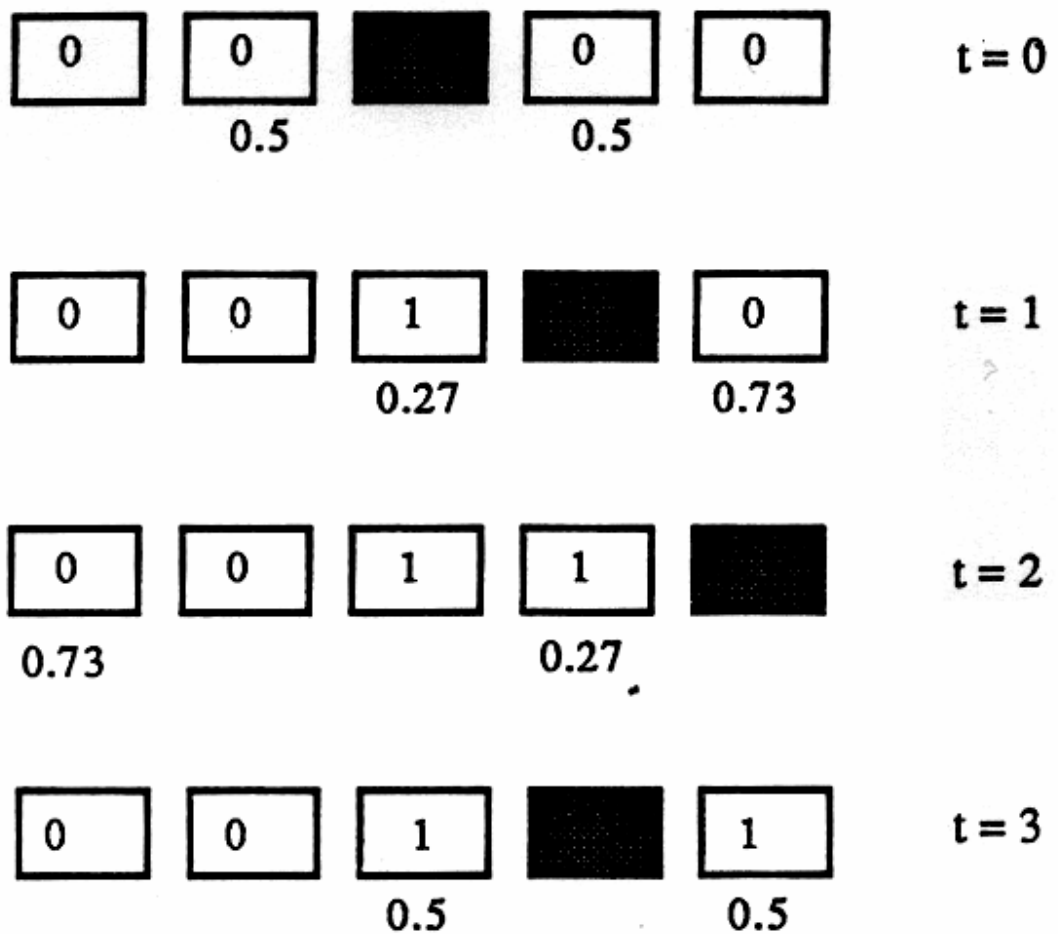
#### Eindimensionaler TSAW

Für den Walker in Zelle  $i$  gilt

$$P_{i\pm 1} = \frac{e^{-gn_{i\pm 1}}}{e^{-gn_{i+1}} + e^{-gn_{i-1}}},$$

mit  $P_{i+1} + P_{i-1} = 1$  und  $n_i$ : Anzahl der Besuche von Zelle  $i$ .

$g > 0$  ist ein Maß für die Stärke, mit der das wiederholte Aufsuchen derselben Zelle vermieden wird. Es gilt  $\lim_{g \rightarrow \infty} \hat{=} SAW$ , bzw.  $g = 0 \hat{=} RW$ .



**Abbildung 26: Zeitentwicklung eines TSAW mit  $g = 1$  aus [2]**  
 (der schwarze Kasten stellt die Position des Walkers zur Zeit  $t$  dar, die Anzahl der Besuche jedes Kastens steht in jedem Kasten und die Wahrscheinlichkeit für einen Walk zum nächsten Nachbarn steht darunter)

## 8. Referenzen

- [1] **Skript Herrmann: Simulationenmethoden in der Physik I+II**  
 (Wahlpflichtfach - Mathematische Methoden)
- [2] Harvey Gould und Jan Taboschnik, **An Introduction to Computer Simulation Methods**,  
 Addison-Wesley, Reading, MA 1988/1996