

# Statistische Mechanik ungeordneter Systeme, SoSe 2005

Timo Aspelmeier, Alexander K. Hartmann, Universität Göttingen

28. Mai 2005

## 6 Monte Carlo Simulationen

Methode um mit Zufallszahlen ein statistisches Ensemble von Konfigurationen zu erzeugen, die gemäß einer vorgegebenen Verteilung verteilt ist. Einführende Bücher von Newman/Barkema [1] und Landau/Binder [2].

### 6.1 Markov Ketten

Gegeben: System mit (endlich) vielen Zuständen  $\mathbf{y} = \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K$  und Wahrscheinlichkeitsdichte  $P(\mathbf{y})$ .

Typisch:  $P(\mathbf{y}) \in O(1)$  nur für wenige Zustände, aber exponentiell klein in  $K$  für die meisten Zustände.

Ziel: Messung von Mittelwerten von Observablen  $A(\mathbf{y})$

$$\langle A \rangle := \sum_{\mathbf{y}} A(\mathbf{y}) P(\mathbf{y}) \quad (1)$$

Annahme:  $K$  sehr groß (z.B.  $K = 2^N$  für  $N$  Ising Spins), so dass nicht alle Zustände enumeriert werden können. Weiterhin  $P(A) \in O(1)$  üblicherweise auch nur in einem exponentiell kleinem Bereich

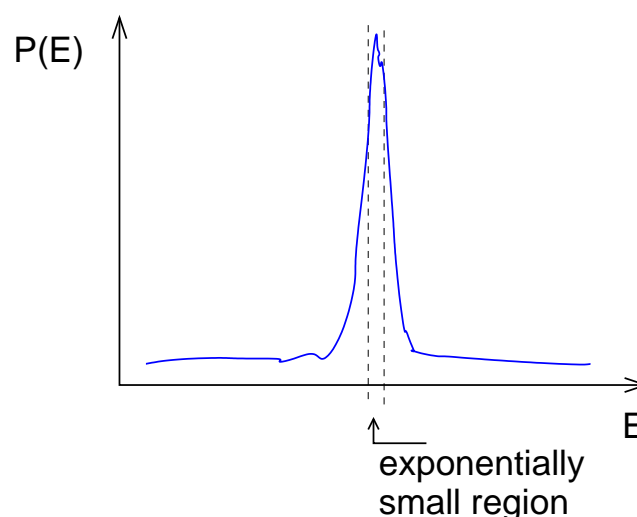


Abbildung 1: Typische Verteilung einer Zustandsdichte bei endlicher Temperatur.

*Simple Sampling:*

Generiere  $M$  Zustände  $\{\mathbf{y}^i\}$  ( $i = 1, \dots, M$ ) zufällig, so dass alle Zustände gleich wahrscheinlich sind, dann ist:

$$\langle A \rangle \approx \bar{A}^{(1)} := \sum_{\mathbf{y}^i} A(\mathbf{y}^i) P(\mathbf{y}^i) / \sum_{\mathbf{y}^i} P(\mathbf{y}^i)$$

Nachteil: für fast alle Zustände ist  $P(\mathbf{y}^i)$  exponentiell klein  $\rightarrow \bar{A}^{(1)}$  sehr ungenau.

*Importance Sampling:*

Viel besser: erzeuge die  $M$  Konfigurationen  $\mathbf{y}^i$  so, dass sie gemäß  $P(\mathbf{y}^i)$  verteilt sind (generiere also die *wichtigen Zustände* häufiger), dann:

$$\langle A \rangle \approx \bar{A}^{(2)} := \sum_{\mathbf{y}^i} A(\mathbf{y}^i) / M \quad (2)$$

Aber: meistens gibt es keinen Algorithmus um die  $\mathbf{y}^i$  direkt gemäß  $P(\mathbf{y}^i)$  zu erzeugen (Verteilungsfunktion nicht ausrechenbar bzw. nicht invertierbar).

$\rightarrow$

Grundidee: die Zustände  $\mathbf{y}^i$  werden nicht unabhängig erzeugt, sondern durch eine *probabilistische Dynamik* zwischen den Zuständen  $\mathbf{y}(t)$  an diskreten Zeiten  $t = 0, 1, 2, \dots$ :  $\mathbf{y}(0) \rightarrow \mathbf{y}(1) \rightarrow \mathbf{y}(2) \rightarrow \dots$

Annahme: Zustände  $\mathbf{y}(t+1)$  hängt nur von Zufallszahl und von  $\mathbf{y}(t)$  ab.  $\{\mathbf{y}(t) | t = 0, 1, 2, \dots\}$  heißt dann *Markov Kette*.

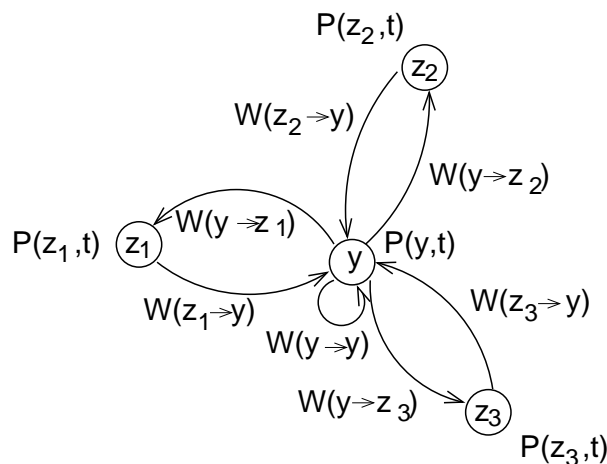
Beschreibung der Übergänge  $\mathbf{y}(t) \rightarrow \mathbf{y}(t+1)$  durch  $W_{\mathbf{yz}} = W(\mathbf{y} \rightarrow \mathbf{z}) =$  Wahrscheinlichkeit vom Zustand  $\mathbf{y}$  (zur Zeit  $t$ ) in den Zustand  $\mathbf{z}$  (zur Zeit  $t+1$ ) zu gehen, wobei  $W_{\mathbf{yz}}$  nicht von  $t$  abhängen soll.

Eigenschaften:

$$\begin{aligned} W_{\mathbf{yz}} &\geq 0 \quad \forall \mathbf{y}, \mathbf{z} \quad (\text{Positivität}) \\ \sum_{\mathbf{z}} W_{\mathbf{yz}} &= 1 \quad \forall \mathbf{y} \quad (\text{Erhaltung}) \end{aligned}$$

Der Zustandsraum zusammen mit den Übergangsraten nennt einen *Markov Prozess*.

Sei  $P(\mathbf{y}, t)$  die Wahrscheinlichkeit, dass das System zur Zeit  $t$  im Zustand  $\mathbf{y}(t) = \mathbf{y}$  ist. Bilanz für den Zustand  $\mathbf{y}$ :



Also gilt:

$$\Delta P(\mathbf{y}, t) := P(\mathbf{y}, t+1) - P(\mathbf{y}, t) = \sum_{\mathbf{z}} W_{\mathbf{zy}} P(\mathbf{z}, t) - \sum_{\mathbf{z}} W_{\mathbf{yz}} P(\mathbf{y}, t) \quad \forall \mathbf{y} \quad (3)$$

Unter bestimmten Umständen (unter anderem, wenn es nur einen Eigenwert  $\lambda = 1$  der Matrix  $(W_{\mathbf{yz}})$  gibt, siehe [3]), konvergiert die Wahrscheinlichkeitsverteilung gegen die *stationäre* (zeitunabhängige) Verteilung

$$P_{ST}(\mathbf{y}) := \lim_{t \rightarrow \infty} P(\mathbf{y}, t)$$

Das ist unabhängig vom gegebenen Startzustand  $\mathbf{y}(0)$ , man nennt so ein System *ergodisch*.

Ziel: Wähle  $W_{\mathbf{yz}}$  so, dass  $P_{ST} = P$

Da  $P(\cdot)$  zeitunabhängig, folgt aus (3) mit

$$0 = \Delta P(\mathbf{y}) = \sum_{\mathbf{z}} W_{\mathbf{zy}} P(\mathbf{z}) - \sum_{\mathbf{z}} W_{\mathbf{yz}} P(\mathbf{y}) \quad \forall \mathbf{y}$$

eine weitere Bedingung für die Übergangsraten.

Das ist z.B. erfüllt, wenn man

$$W_{\mathbf{zy}} P(\mathbf{z}) - W_{\mathbf{yz}} P(\mathbf{y}) = 0 \quad \forall \mathbf{y}, \mathbf{z} \quad (4)$$

(genannt *detaillierte Balance*) wählt.

Damit: der Markov Prozess erzeugt Zustände, die gemäß  $P(\cdot)$  verteilt sind, man kann also Mittelwerte wie in (2) ausrechnen. Achtung: Zu Beginn hängen die Konfigurationen vom Startzustand  $\mathbf{y}(0)$  ab, man lässt also die ersten Zustände  $t < t_{equi}$  bei der Mittelwertbildung weg (“Equilibrierung”). Außerdem ist  $\mathbf{y}(t+1)$  üblicherweise “ähnlich” zu  $\mathbf{y}(t) \rightarrow$  nur entferntere Zustände  $\mathbf{y}(t), \mathbf{y}(t+\Delta t), \mathbf{y}(t+2\Delta t), \dots$  sind unabhängig. Hinweis:  $t_{equi}, \Delta t$  hängen STARK vom Modell und Parametern ab.

## 6.2 Ungeordneter (verdünnter) Ferromagnet

Modell:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} e_i s_i e_j s_j \quad J > 0. \quad (5)$$

$\langle i, j \rangle$  : Summe über wechselwirkende Nachbarn (z.B. nächste N.)

$e_i = 0, 1$  : Gitterplatz frei (mit Wahrscheinlichkeit  $p$ )/besetzt ( $1-p$ )

$s_i = \pm 1$  : Spin-Orientierung “hoch” / “runter”.

Konfiguration:  $\mathbf{x} = (s_1, s_2, \dots, s_N)$  (Spins mit  $e_i = 0$  werden effektiv ignoriert)

Ziel: Simulation im *kanonischen Ensemble*, also

$$P(\mathbf{x}) = \frac{1}{Z} \exp(-\mathcal{H}(\mathbf{x})/T), \quad (6)$$

mit  $Z =$  Zustandssumme  $Z = \sum_{\{\mathbf{x}\}} \exp(-\mathcal{H}(\mathbf{x})/T)$ .

Methode: benutze Markovkette. Hier:

Metropolis Algorithmus definiert durch Übergangsraten  $W_{\mathbf{y}\mathbf{z}} = W(\mathbf{y} \rightarrow \mathbf{z})$ .

Grundidee: (gegeben  $\mathbf{y} = \mathbf{y}(t)$ )

1. Suche Konfiguration  $\mathbf{z}$  *zufällig* aus  $A(\mathbf{y} \rightarrow \mathbf{z})$
2. Mit Wahrscheinlichkeit  $\tilde{W}(\mathbf{y} \rightarrow \mathbf{z})$  wird  $\mathbf{z}$  *akzeptiert*, also  $\mathbf{y}(t+1) = \mathbf{z}$ .  
 $\tilde{W}(\mathbf{y} \rightarrow \mathbf{z})$  heißt *Akzeptanz-Wahrscheinlichkeit*  
Mit Wahrscheinlichkeit  $1 - \tilde{W}(\mathbf{y} \rightarrow \mathbf{z})$  wird  $\mathbf{z}$  *verworfen*, also  $\mathbf{y}(t+1) = \mathbf{y}$

→ Gesamt-Wahrscheinlichkeit:

$$W(\mathbf{y} \rightarrow \mathbf{z}) = A(\mathbf{y} \rightarrow \mathbf{z}) \tilde{W}(\mathbf{y} \rightarrow \mathbf{z}) \quad (\mathbf{y} \neq \mathbf{z}). \quad (7)$$

(Gesamt-Wahrscheinlichkeit, dass  $\mathbf{y}$  bleibt, ist  $W(\mathbf{y} \rightarrow \mathbf{y}) = 1 - \sum_{\mathbf{z} \neq \mathbf{y}} W(\mathbf{y} \rightarrow \mathbf{z})$ ).

Einsetzen Gl.. (7) in detaillierte Balance Bedingung Eq. (4):

$$\frac{\tilde{W}(\mathbf{y} \rightarrow \mathbf{z})}{\tilde{W}(\mathbf{z} \rightarrow \mathbf{y})} = \frac{P(\mathbf{z}) A(\mathbf{z} \rightarrow \mathbf{y})}{P(\mathbf{y}) A(\mathbf{y} \rightarrow \mathbf{z})}. \quad (8)$$

Für Metropolis Algorithmus [4], Wahl:

$$\tilde{W}(\mathbf{y} \rightarrow \mathbf{z}) = \min \left( 1, \frac{P(\mathbf{z}) A(\mathbf{z} \rightarrow \mathbf{y})}{P(\mathbf{y}) A(\mathbf{y} \rightarrow \mathbf{z})} \right), \quad (9)$$

Man sieht leicht, dass Gl. (8) gilt.

Am einfachsten: Single-Spin-Flip Dynamik:

Sei  $\mathbf{y} = (s_1, s_2, \dots, s_N)$ . Wahl eines Spins  $i$  zufällig, damit  $\mathbf{z} = (s'_1, s'_2, \dots, s'_N)$  mit

$$s'_i = \begin{cases} -s_i & \text{für } i = j \\ s_i & \text{sonst} \end{cases}$$

Alle Spins gleich wahrscheinlich:  $A(\mathbf{y} \rightarrow \mathbf{z}) = 1/N$

(besser: wähle nur unter Spins  $e_i \neq 0$ :  $A(\mathbf{y} \rightarrow \mathbf{z}) = 1/N_{\text{spins}}$ ,  $N_{\text{spins}} = \sum_i e_i$ )

in Gl. (9) →

$$\tilde{W}(\mathbf{y} \rightarrow \mathbf{z}) = \min \left( 1, \frac{P(\mathbf{z})}{P(\mathbf{y})} \right) = \min (1, \exp(-[\mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{y})]/T)) . \quad (10)$$

Akzeptanz-Wahrscheinlichkeit hängt nur von *Energieänderung*  $\Delta\mathcal{H} = \mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{y})$  ab → leicht zu berechnen, weil nur Nachbarn von  $j$  berücksichtigt werden müssen:

$$\Delta\mathcal{H} = \Delta\mathcal{H}(j) = 2J \sum_{\langle i,j \rangle} e_i s_i e_j s_j$$

→

$$\tilde{W}(\mathbf{y} \rightarrow \mathbf{z}) = \begin{cases} 1 & \text{für } \Delta\mathcal{H} < 0 \\ \exp(-\Delta\mathcal{H}/T) & \text{sonst} \end{cases} \quad (11)$$

Beobachtung: bei tiefen Temperaturen sind Änderungen, die die Energie erhöhen selten.  
Realisierung in C:

```

/***** diluted_mc_T() *****/
/** Does metropolis MC-simulation with diluted ferromagnet **/
/** at T>0. **/
/** PARAMETERS: (*)= return-paramter **/
/**      spin: spin-configuration [1..N] **/
/**      dim: dimension of system **/
/**      N: number of spins **/
/**      next: gives neighbours next[0..N][0..2*dim+1] (t2)**/
/**      e: sites are occupied (e[i]=1) or empty (=0) **/
/** mc_steps: number of MC steps **/
/**      T: temperature in units of j **/
/** RETURNS: **/
/**      0 **/
/*****/
int diluted_mc_T(short int *spin, int dim, int N, int *next,
                short int *e, int mc_steps, double T)
{
    int t,r, t2, num_mc;
    double delta_e;

    for(num_mc=0; num_mc<mc_steps*N; num_mc++)          /* do MC */
    {
        double p, factor;
        do
            t = 1+ (int) (drand48() * N);                /* choose spin randomly */
        while(e[t] == 0);
        delta_e = 0.0;
        for(r=0; r<2*dim; r++)                            /* calculate local energy -> delta_e */
        {
            t2 = NEXT(t,r);
            delta_e += 2.0*spin[t]*spin[t2]*e[t2];
        }
        p = drand48();
        if( (delta_e <= 0) || (p< exp(-delta_e/T) ))        /* flip spin ? */
        {
            spin[t]=-spin[t];                             /* flip spin */
        }
    }

    return(0);
}

```

Hauptprogramm: diluted\_sim. Aufruf:

USAGE: diluted\_sim {<options>} <L> <#sweeps>

OPTIONS:

-T <T>: set temperature (d:2.3)  
 -p <p>: set hole prob. (d:0.0)  
 -up: start with all spins up (d:random)  
 -seed <s>: seed for rnd number (d:10000)

-appendix <a>: for output filename

erzeugt Ausgabe-File `dilutedL<L>p<p>T<T>.out` mit 3 Spalten: Zeitschritt, Energie, Magnetisierung  $m = \frac{1}{N_{\text{spins}}} \sum_i s_i e_i$ .

Da immer nur ein Spin geflipt wird: Algorithmus ist langsam, Konfigurationen ändern sich nur langsam global.

Messung: Autokorrelationsfunktion (Mittelwert  $\langle m \rangle = \int m(t) dt$ ):

$$C(t) = \int_{t'=0}^{t_{\max}-t} (|m(t')||m(t'+t)| - \langle m^2 \rangle) \quad (12)$$

Am langsamsten: in der Nähe von Phasenübergängen. Bsp.: geordneter Ferromagnet ( $L = 40, p = 0$ ), Auswertung mit Programm `diluted_analysis`

USAGE: `diluted_analysis [OPTIONS] <t_min> <files>`

OPTIONS:

-correl <t>: correlation functions till <t>

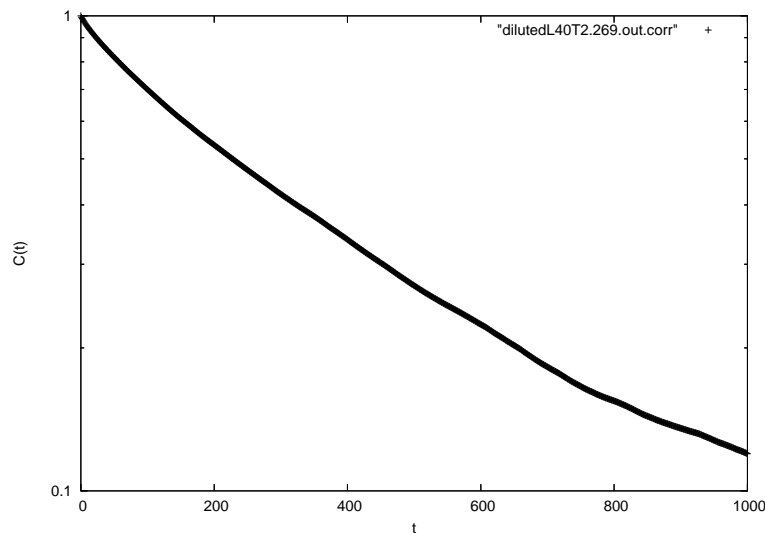


Abbildung 2: Autokorrelationsfunktion der Magnetisierung für 2-dimensionalen Ferromagnet  $L = 40$  bei  $T = T_c = 2.269$

Definiere: Korrelationszeit  $\tau$  = Zeit wo  $C(t)/C(0)$  den Wert  $e^{-1}$  erreicht (Annahme  $C(t) = C(0)e^{-t/\tau}$ ). Damit:

Man findet

$$\tau(L) \sim L^z \quad (13)$$

mit  $z \approx 2.1$ , siehe Abb. 3.

Ausweg: Cluster Algorithmen.

## Literatur

- [1] M.E.J. Newman and G.T. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, 1999)

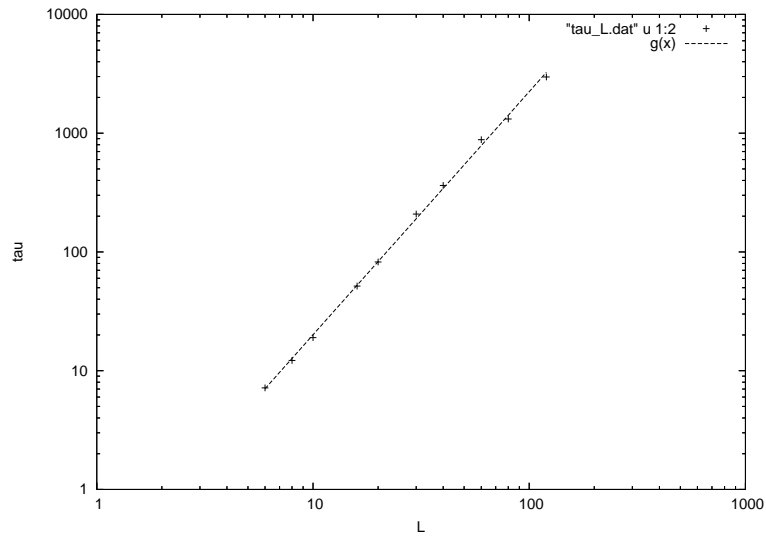


Abbildung 3: Korrelationszeit am Phasenübergang  $T = T_c = 2.269$  als Funktion der Systemgröße  $L$  für 2-dimensionalen Ferromagnet.

- [2] D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, (Cambridge University Press, Cambridge 2000)
- [3] L.E. Reichl, *A Modern Course in Statistical Physics*, (John Wiley & Sons, New York 1998)
- [4] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953)