

Monte Carlo Simulation des Ising Modell

Niek Andresen, Robert Hartmann, Jan Fabian Schimd

WS 14/15

Inhaltsverzeichnis

1	Problemstellung und Theorie	2
1.1	Das Ising-Modell	2
1.2	Monte-Carlo-Simulation und Metropolis-Algorithmus	2
1.3	Ziele	2
2	Das Programm	3
2.1	Allgemeiner Aufbau	3
2.2	Im Detail: Konvergenz des Metropolis Algorithmus	3
2.3	Im Deatail: Funktion weiterer Parameter	3
3	Ergebnisse und Auswertung	4
3.1	Phasenübergang zwischen Hoch- und Niedrigtemperaturphase	4
3.2	Schaltverhalten im Magnetfeld	4
4	Alternative: Cluster Monte Carlo Algorithmen	5
4.1	Wolff-Algorithmus	5
4.2	Flipp-Wahrscheinlichkeit des Clusters = 1	5
4.3	Konvergenzverhalten im Vergleich	6
5	Zusammenfassung	7
6	Quellen	8

1 Problemstellung und Theorie

1.1 Das Ising-Modell

Das Ising-Modell beschreibt näherungsweise den Ferromagnetismus in Festkörpern. Hierbei wird ein Kristall als äquidistantes Gitter gesehen, an dessen Gitterpunkten sich einzelne Spins befinden, welche ein magnetisches Moment haben. In der Näherung verändern sich die Positionen der Spins nicht. Nimmt man zusätzlich an, dass nur nächste Nachbarn im Gitter sich gegenseitig beeinflussen und beschränkt man nun noch die Spins in ihre z-Richtung und lässt sie nur zwei diskrete Werte annehmen (± 1), ergibt sich der Hamiltonoperator:

$$\hat{H} = -J \sum_{\langle i,j \rangle} S_i S_j - B \sum_{i=1}^N S_i$$

wobei $S_i \in \{-1, +1\}$. J ist die Kopplungskonstante zwischen benachbarten Spins, während die Summe nur über nächste Nachbarn $\langle i, j \rangle$ geht. B ist ein äußeres Magnetfeld. Somit liefern entgegengesetzt gerichtete Nachbarn einen positiven Energiebeitrag und gleichgerichtete einen negativen (für $J > 0$).

1.2 Monte-Carlo-Simulation und Metropolis-Algorithmus

Zur Simulation dieses Systems im Ising-Modell wird die Monte-Carlo-Methode verwendet. Hierbei werden mit Hilfe von Zufallszahlen mögliche Konfigurationen des Systems erstellt. Um einen einfachen Mittelwert einer gesuchten Größe bilden zu können, werden Konfigurationen gemäß der Metropolis-Methode gewählt.

Die Wahrscheinlichkeitsverteilung ist die Boltzmann-Verteilung, sodass sich folgende Wahrscheinlichkeit zur Durchführung eines Spin-Flips, der eine Energieänderung von ΔE zur Folge hätte, ergibt:

$$P(S_k \rightarrow -S_k) = \min\{1, e^{(-\frac{\Delta E}{k_B T})}\}$$

1.3 Ziele

Es soll das Verhalten des Gitters bei verschiedenen Temperaturen und äußeren Magnetfeldstärken sichtbar werden. Außerdem wird die Konvergenz des Systems und damit der Größe der Gesamtmagnetisierung $m = \sum_i S_i$ sowie der Energie betrachtet.

2 Das Programm

2.1 Allgemeiner Aufbau

Hier die allgemeine Struktur...

2.2 Im Detail: Konvergenz des Metropolis Algorithmus

Hier der Metropolis Block eines Monte Carlo Schritts mit Konvergenzverhalten

2.3 Im Detail: Funktion weiterer Parameter

Unterschied B T Mode etc. ..

3 Ergebnisse und Auswertung

3.1 Phasenübergang zwischen Hoch- und Niedrigtemperaturphase

Bilder, Bilder, Bilder...

3.2 Schaltverhalten im Magnetfeld

Bilder, Bilder, Bilder...

4 Alternative: Cluster Monte Carlo Algorithmen

Clusteralgorithmen wirken dem Problem hoher Autokorrelationszeiten im Bereich des Phasenübergangs entgegen.

Um dies zu erreichen arbeiten sie nicht mehr nur Lokal, wie der Metropolis Algorithmus, sondern bilden größere Bereiche (Cluster), die auf einmal manipuliert werden.

4.1 Wolff-Algorithmus

Der Wolff-Algorithmus ist ein Cluster-Algorithmus, der pro Monte-Carlo-Schritt von einem zufälligen Startpunkt aus ein Cluster bildet und dann mit einer bestimmten Wahrscheinlichkeit alle Spins innerhalb des Clusters flippt.

Allgemeiner Programmablauf:

1. Startkonfiguration (Abbildung 4.1)
2. Bestimme einen zufälligen Startpunkt (Abbildung 4.2)
3. Ausgehend vom Startpunkt werden benachbarte Atome mit gleichem Spin mit Wahrscheinlichkeit p in den Cluster aufgenommen (Abbildung 4.3a,b,c)
4. Flippe alle Spins des Clusters mit einer Wahrscheinlichkeit (Abbildung 4.4)
5. Gehe zu 1.

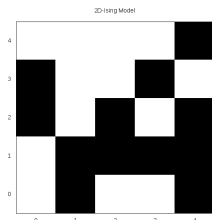


Abbildung 1

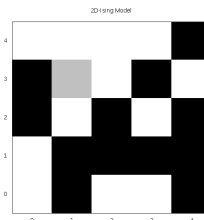


Abbildung 2

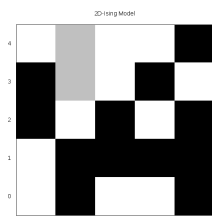


Abbildung 3

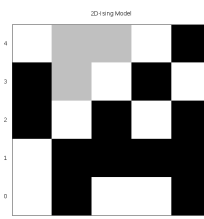


Abbildung 4

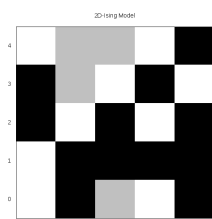


Abbildung 5

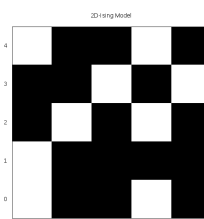


Abbildung 6

4.2 Flipp-Wahrscheinlichkeit des Clusters = 1

Der Algorithmus wird besonders effizient, da bei einer bestimmten Annahme-Wahrscheinlichkeit p , dass ein Zustand in den Cluster aufgenommen wird, der Cluster in jedem Monte-Carlo-Schritt

geflippt wird.

Warum dies so ist wird klar, wenn man sich die Wahrscheinlichkeit W_{ij} für den Übergang von der Konfiguration i zu j berechnet.

$$W_{ij} = \min\{1, \frac{P_j}{P_i}\} = \min\{1, \frac{A(j \rightarrow i) * P_j}{A(i \rightarrow j) * P_i}\} = \min\{1, \frac{A(j \rightarrow i) * e^{-\beta E_j}}{A(i \rightarrow j) * e^{-\beta E_i}}\} \quad (4.1)$$

Für die Wahrscheinlichkeit $A(i \rightarrow j)$ den Übergang von i nach j zu betrachten und für die Energie E_i im Zustand i gelten:

(Hierbei bedeutet *innen* jeweils innerhalb und *außen* außerhalb des Clusters

und n_{gleich} bzw. n_{diff} sind die Anzahl Spins, die am Rand gleich bzw. ungleich zu dem Spin innerhalb des Clusters sind.)

$$A(i \rightarrow j) = A_{\text{innen}} * (1 - p)^{n_{\text{gleich}}} \quad (4.2)$$

$$A_{\text{innen}} = p^{n_{\text{innen}} - 1} + Z \quad (4.3)$$

$$E_i = E_{\text{innen}} + E_{\text{außen}} - n_{\text{gleich}} * J + n_{\text{diff}} * J \quad (4.4)$$

$$(4.5)$$

Mit Z der Wahrscheinlichkeit einen Spin im Cluster als Startpunkt gewählt zu haben. Ebenso gilt:

$$A(j \rightarrow i) = A_{\text{innen}} * (1 - p)^{n_{\text{diff}}} \quad (4.6)$$

$$E_j = E_{\text{innen}} + E_{\text{außen}} + n_{\text{gleich}} * J - n_{\text{diff}} * J \quad (4.7)$$

$$(4.8)$$

Damit gilt:

$$W_{ij} = \min\{1, \frac{A_{\text{innen}} * (1 - p)^{n_{\text{diff}}}}{A_{\text{innen}} * (1 - p)^{n_{\text{gleich}}}} * \frac{e^{-\beta E_{\text{innen}} + E_{\text{außen}} + n_{\text{gleich}} * J - n_{\text{diff}} * J}}{e^{-\beta E_{\text{innen}} + E_{\text{außen}} - n_{\text{gleich}} * J + n_{\text{diff}} * J}}\} \quad (4.9)$$

$$= \min\{1, \frac{(1 - p)^{n_{\text{diff}}}}{(1 - p)^{n_{\text{gleich}}}} * \frac{e^{-\beta n_{\text{gleich}} * J} * e^{+\beta n_{\text{diff}} * J}}{e^{+\beta n_{\text{gleich}} * J} * e^{-\beta n_{\text{diff}} * J}}\} \quad (4.10)$$

$$= \min\{1, \frac{(1 - p)^{n_{\text{diff}}}}{(1 - p)^{n_{\text{gleich}}}} * \frac{e^{-2\beta n_{\text{gleich}} * J}}{e^{-2\beta n_{\text{diff}} * J}}\} \quad (4.11)$$

$$= \min\{1, \left(\frac{(1 - p)}{e^{-2\beta J}}\right)^{n_{\text{diff}}} * \left(\frac{e^{-2\beta J}}{(1 - p)}\right)^{n_{\text{gleich}}}\} \quad (4.12)$$

Hieraus lässt sich nun erkennen, dass die Wahrscheinlichkeit vom Zustand i zu j überzugehen gleich 1 wird, wenn wir $p = 1 - e^{-2\beta J}$ wählen.

4.3 Konvergenzverhalten im Vergleich

Der Vorteil des Cluster-Algorithmus ist die schnelle Konvergenzgeschwindigkeit in der Nähe des Phasenübergangs bzw. der kritischen Temperatur.

(Graphen)

Wie man in Abbildung 5.1 erkennt...

5 Zusammenfassung

Hier könnte ein Fazit stehen... :)

Abbildungsverzeichnis

1	5
2	5
3	5
4	5
5	5
6	5

6 Quellen

- blabla