

# Ising Modell

Lösung durch Monte-Carlo Simulation

**Christian Gommeringer**

betreut durch Dr. Quandt und Dr. Burgio

Tübingen, den 29. März 2023

## Aufgabe 1

Zunächst soll eine Wahrscheinlichkeitsverteilung für eine Wahrscheinlichkeitsdichte numerisch unter der Benutzung eines Zufallszahlengenerators erstellt werden, der Zahlen gleichverteilt zwischen 0 und 1 erzeugt. Hierfür benutzen wir die Beziehung für eine Wahrscheinlichkeitsdichte im Intervall  $[a,b]$ .

$$y = F(z) = \int_a^z f(x)dx$$

Wenn wir uns nun die Funktion  $F^{-1}(y)$  anschauen, nimmt diese alle Werte im Intervall  $[0,1]$  an. Dass die Werte auch mit der entsprechenden Wahrscheinlichkeit  $f(x)$  gewählt werden, wenn  $y$  durch einen gleichverteilten Zufallsgenerator erzeugt wird, sieht man auf folgende Weise. Damit ein Wert  $x$  im Intervall  $[x, x+dx]$  angenommen wird, muss  $y$  in einem Intervall  $[y(x), y(x)+dy(x)]$  gezogen worden sein. Die Wahrscheinlichkeit dafür ist gerade  $dy(x)$ . Es gilt allerdings

$$dy(x) = f(x)dx$$

und demnach werden Werte aus  $[x, x+dx]$  gerade mit der richtigen Wahrscheinlichkeit  $f(x)$  gezogen. Wenn man nun eine Wahrscheinlichkeitsverteilung verwenden möchte, bei der eine analytische Integration und Invertierung nicht möglich ist, kann man nach folgender Weise vorgehen. Man wählt eine Funktion für die sich das erfüllen lässt und verschiebt sie auf soweit nach oben, dass sie für das ganze betrachtete Intervall über der gegebenen Wahrscheinlichkeitsdichte liegt. Wir nennen diese nach oben verschobene Funktion im folgenden  $g(x)$ . Nun erzeugt man Tupel von zwei Zahlen, von denen die erste ( $x$ ) nach vorherigem Verfahren für die einhüllende Funktion gezogen werden, und die zweite ( $y$ ) gleich verteilt in einem Intervall  $[0, g(x)]$  erstellt wird. Gilt  $f(x) \leq y$ , wird die Zahl  $x$  behalten, andernfalls wird sie verworfen. Auf diese Weise wird das Ensemble  $\{x\}$  nach der vorgegebenen Wahrscheinlichkeit  $f(x)$  erstellt. In unserem Fall möchten wir diese Verteilung für die Funktion

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

erstellen. Als Einhüllende wähle ich

$$g(x) = \exp(-|x - \mu| + \alpha)$$

ein ausreichendes  $\alpha$  zu bestimmen, berechne ich den Schnittpunkt der beiden Funktionen. Ich nehme zudem die vereinfachende Substitution  $x - \mu =: z$  vor

und beschränke mich auf den Fall  $z > 0$ . Dann muss für eine Intersektion gelten.

$$z - \alpha = z^2/2\sigma^2 + \ln \sqrt{\pi\sigma^2}$$

das führt auf eine Bedingung

$$z_{1/2} = \sigma^2 \left( 1 \pm \sqrt{1 - \frac{2}{\sigma^2} (\ln \sqrt{2\pi\sigma^2} + \alpha)} \right)$$

Damit es keine Schnittpunkte gibt muss gelten

$$\alpha \geq \frac{\sigma^2}{2} - \ln \sqrt{2\pi\sigma^2}$$

Das beschriebene Verwerfungsverfahren basiert darauf, dass man eine entsprechende Verteilung generiert, wenn man im zweidimensionalen Raum des Funktionsgraphen in der Fläche unterhalb der Wahrscheinlichkeitsfunktion Punkte mit einer gleichmäßigen Flächendichte erzeugt. Dass dies äquivalent zu obiger Wahrscheinlichkeitserzeugung durch Invertierung ist, sieht man, wenn man sich die Wahrscheinlichkeit betrachtet mit obigem Verfahren in eine Fläche  $dA = [x, x+dx][y, y+dy]$  ( $x$  und  $y$  unabhängig aber innerhalb der von der Funktion eingeschlossenen Fläche) zu treffen. Für das  $x$ -Intervall gilt nach der Wahrscheinlichkeitsverteilung die Wahrscheinlichkeit  $\rho_x = f(x)dx$  und für das  $y$ -Intervall, da dieses gleichverteilt gezogen wird,  $\rho_y = 1/f(x) dy$ . Und damit ist die Wahrscheinlichkeit in die Fläche zu treffen

$$\rho_A = \rho_x \cdot \rho_y = dx dy$$

überall konstant. Das dieses Verfahren auch für eine nichtnormierte Funktion  $f(x)$  eine gleichmäßige Flächendichteverteilung erzeugt, sieht man hier auch gut, da in diesem Fall  $\rho_x$  lediglich mit einer konstanten Normierungskonstanten multipliziert werden müsste.

Für unsere konkrete Aufgabe befassen wir uns mit dem Fall  $\mu = 0$ ,  $\sigma = 1$ . Dafür verwende ich, wie oben schon erwähnt, für die Erzeugung der einheitlichen Flächendichte die Funktion

$$g(x) = \frac{1}{2} e^{-|x|}$$

Es gilt für  $y < \frac{1}{2}, x < 0$

$$\begin{aligned} y &= \int_{-\infty}^x g(x) dx \\ &= \frac{1}{2} e^x \\ \Rightarrow x(y) &= \ln(2y) \end{aligned}$$

und für  $y \geq \frac{1}{2}, x \geq 0$

$$\begin{aligned} y &= \int_{-\infty}^x g(x) dx \\ &= \frac{1}{2} + \frac{1}{2}(1 - e^{-x}) \\ \Rightarrow x(y) &= -\ln(2(1 - y)) \end{aligned}$$

Mit dieser Wahrscheinlichkeit lässt sich nun mit Hilfe der Verwerfungsmethode auch eine Gaußverteilung erzeugen.

Wenn man nun das Integral einer Funktion berechnen will, gibt es die Möglichkeit, dies über den Vergleich zu einem bekannten Integral zu machen, wobei man wiederum einfach das Verwerfungsverfahren anwenden kann. Das Verhältnis zwischen den akzeptierten Punkten zur Gesamtzahl der erzeugten Punkte entspricht nämlich genau dem Flächenverhältnis der gesuchten zur einschließenden Fläche. Uns interessiert das Integral

$$\int_{-\infty}^{\infty} e^{-x^2/2}$$

Als einhüllende verwende ich  $\exp(-|x| + 0.6)$  und kann wie oben beschrieben die abgeleitete Wahrscheinlichkeitsverteilung von  $g(x) = \exp(-|x|)/2$  verwenden. Ich erhalte über diese Methode bei  $10^{1000000}$  Punkten und einer Rechenzeit von  $1.8\text{ s}$  einen Fehler von  $-0.000471$ , und bei  $10^{10000000}$  Punkten (10 mal mehr) und einer Rechenzeit von  $18.2\text{ s}$  einen Fehler von  $0.000136$ . Der Algorithmus skaliert also ziemlich schlecht und liefert auch kein besonders gutes Ergebnis.

Als zweite Möglichkeit könnte man die Funktion auch als Treppenfunktion approximieren. Die Stützstellen wähle ich zufällig nach einer Wahrscheinlichkeitsverteilung aus, die aus der Ableitung bestimmt wurde. Über dieses Verfahren erreiche ich mit  $10^5$  Stützstellen und einer Rechenzeit von  $0.2\text{ s}$  einen Fehler von  $3.06691\text{e-}8$ .

## Aufgabe 2

Nachdem wir uns mit der Grundmethodik der Montecarlo Simulation vertraut gemacht haben, möchten wir uns nun auch schon dem Ising Modell zuwenden. Das Ising Modell behandelt ein System von Spin-Zuständen, auf einem Gitter im Raum. Die Spins können die Werte -1 und 1 annehmen und Der Hamiltonian ihrer Wechselwirkung betrachtet nur Wechselwirkungen nächster Nachbarn.

$$H = -J \sum_{\langle i,j \rangle} s_i s_j - \sum_i s_i h$$

Wobei  $J$  der Kopplungsparameter der Spins und  $h$  das externe Feld ist, was im Folgenden 0 sein wird. In 2 Dimensionen wurde diese Problem von Onsager gelöst. Für  $h = 0$  und im thermodynamischen Limes, in dem die Anzahl der Spins unendlich groß wird, lautet die Lösung für die Energiedichte

$$\varepsilon(\beta) = -J \coth(2\beta J) \cdot \left[ 1 + (2 \tanh^2(2\beta J) - 1) \frac{2}{\pi} K(\xi) \right],$$

wobei

$$K(\xi) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1 - \xi^2 \sin^2 \theta}}$$

das vollständige elliptische Integral 1. Ordnung ist.

$$\xi = 2 \frac{\tanh(2\beta J)}{\cosh(2\beta J)}$$

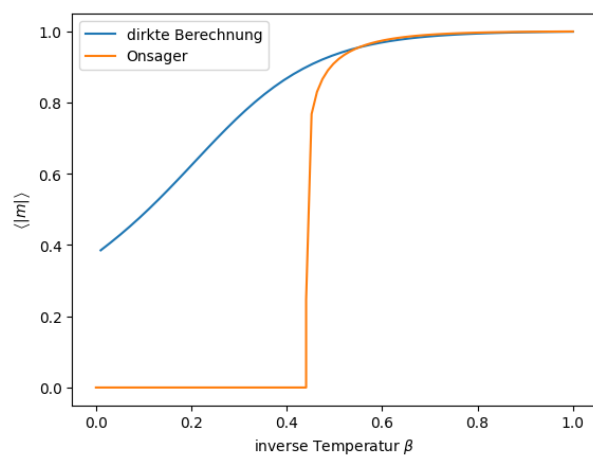
Des weiteren ist der mittlere Betrag der Magnetisierungsdichte gegeben durch

$$|m|(\beta) = [1 - \sinh^{-4}(2\beta J)]^{\frac{1}{8}}$$

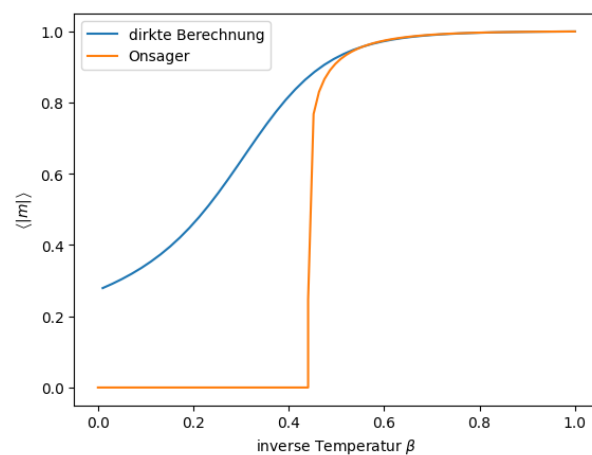
Wir werden nun Isingsystem mit  $L \times L$  quadratisch angeordneten Spins für  $L = 2, 3, 4$  betrachten. Hier soll die Energiedichte und der mittlere Betrag der Magnetisierung durch direktes Summieren aller Möglichkeiten mit deren entsprechender Wahrscheinlichkeit gemäß der kanonischen Zustandsdichte

$$Z = \sum_{\{s_i\}} \exp(-\beta H(\{s_i\}))$$

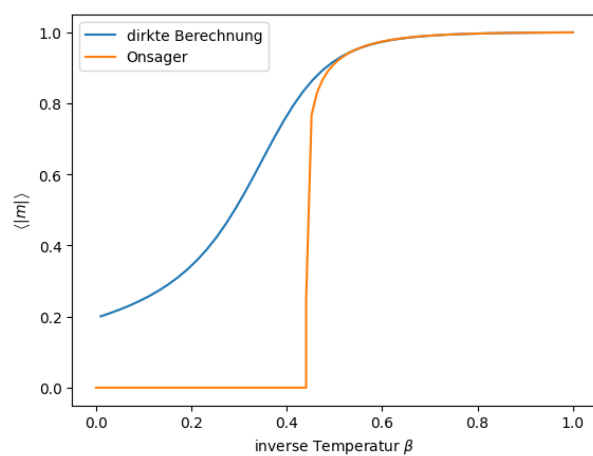
Danach sollen diese Größen mit der Lösung Onsagers verglichen werden. Ich stelle die Ergebnisse graphisch vor.



(a) das 2x2 Gitter



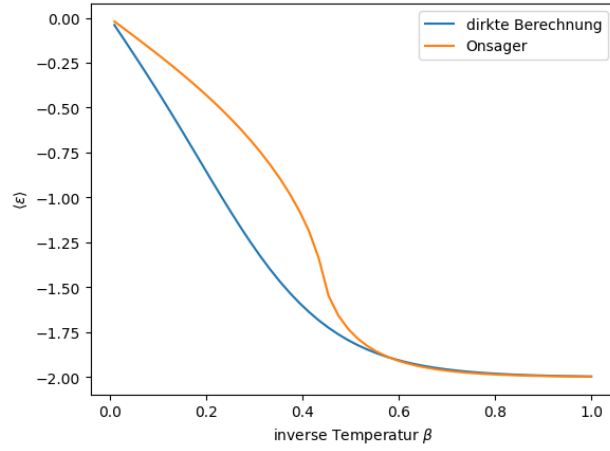
(b) das 3x3 Gitter



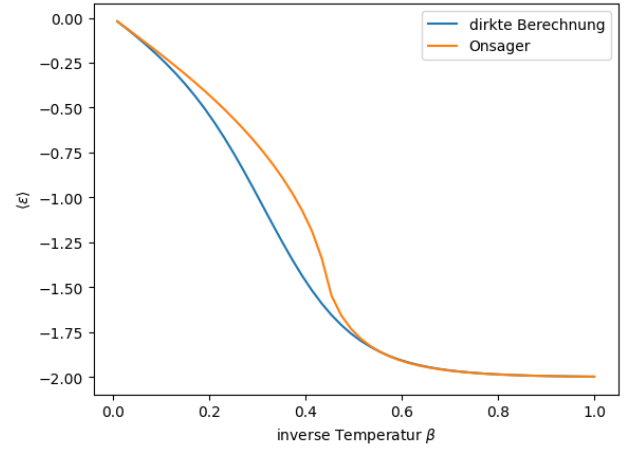
(c) das 4x4 Gitter

Abbildung 1: Verlauf von  $\langle |m| \rangle$  in Abhängigkeit der Temperatur

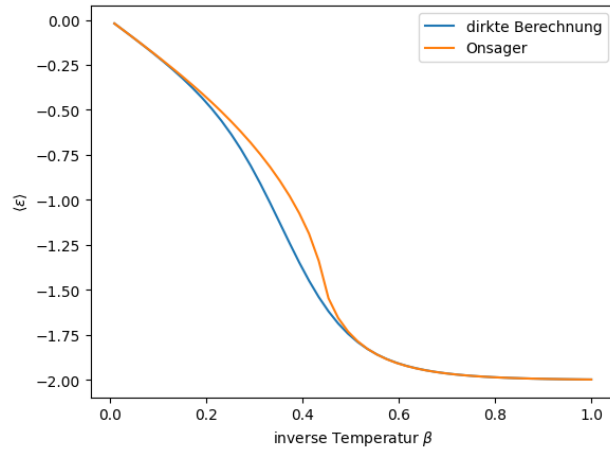
Es lässt sich erkennen, dass die Magnetisierungskurven für sehr kleine Anzahlen an Spins natürlich relative weit von Onsagers exakter Lösung abweichen, die ja für den thermodynamischen Limes gilt. Insbesondere zeigen die kleinen Spinsysteme keinen Phasenübergang 2. Ordnung.



(a) das 2x2 Gitter



(b) das 3x3 Gitter



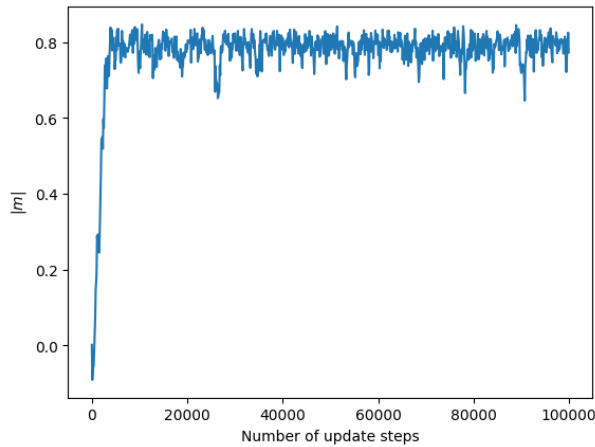
(c) das 4x4 Gitter

Abbildung 2: Verlauf von  $\langle \epsilon \rangle$  in Abhängigkeit der Temperatur

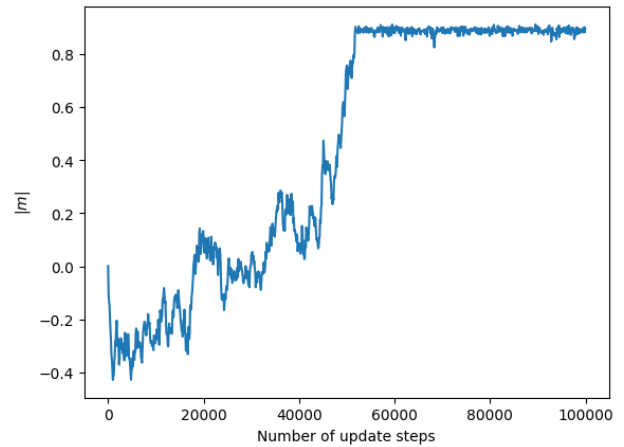
Auch bei den Energien kann man eine Abweichung erkennen. Eine Berechnung der mittleren Magnetisierung  $\langle m \rangle$  kann entfallen, da diese aus Symmetriegründen immer 0 ist. Eine Inversion aller Spins ist nämlich immer gleich wahrscheinlich wie der ursprüngliche Zustand und diese beiden Zustände heben sich in der Mittelung folglich weg.

## Aufgabe 3

Bei den kleinen Gittergrößen des letzten Abschnitts war es noch möglich die Mittelwerte explizit und exakt zu berechnen. Da die Anzahl an Möglichkeiten jedoch mit  $2^N$  mit der Anzahl der Spins steigt, ist diese Methode für größere Gitter nicht mehr praktikierbar. Hier kommt der Monte-Carlo Algorithmus ins Spiel. Wir möchten für ein Gitter der Größe  $128 \times 128$ , den mittleren Magnetisierungsbetrag  $\langle |m| \rangle$  und die mittlere Energiedichte ohne Einwirkung eines externen Feldes ( $h=0$ ) bestimmen. In diesem Abschnitt werden wir den multihit Metropolis Update Algorithmus verwenden. Zunächst müssen passende Thermalisierungszeiten bestimmt werden. Dazu messe ich nach jedem Update Schritt die Magnetisierung und Energie und trage dies gegen die Nummer des Updates auf. Das führe ich für verschiedene Temperaturen und multihit Werte durch.



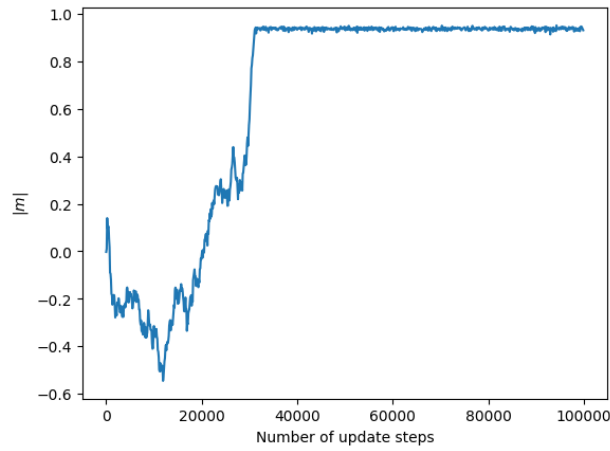
(a) Thermalisierungskurve für  $\beta = 0.46$



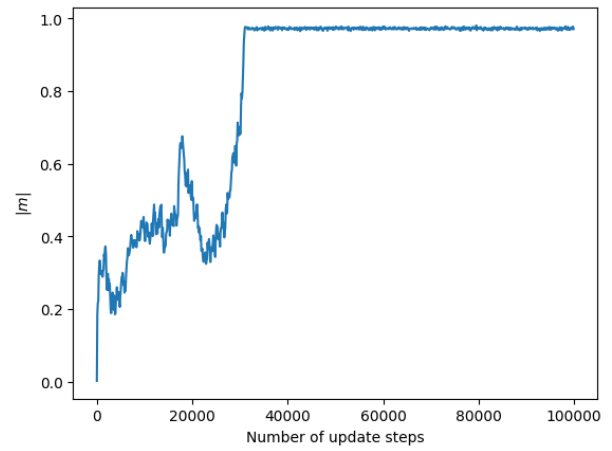
(b) Thermalisierungskurve für  $\beta = 0.49$

Abbildung 3: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für 3 multihits

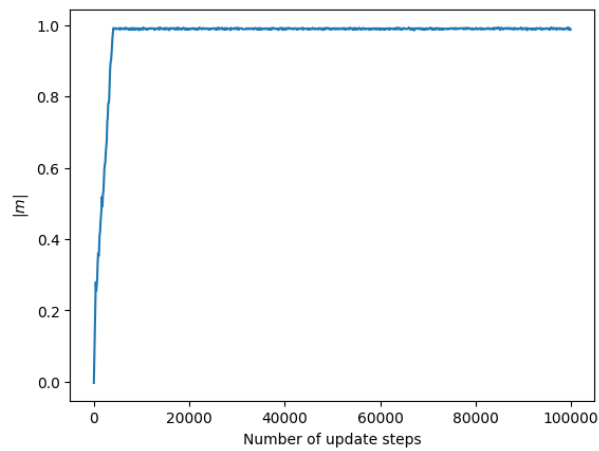




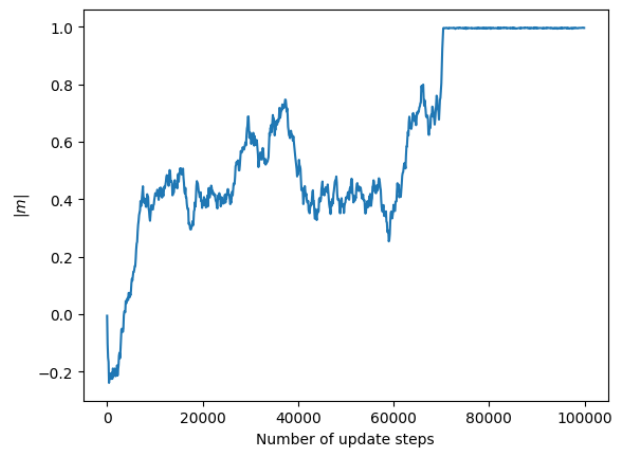
(a) Thermalisierungskurve für  $\beta = 0.53$



(b) Thermalisierungskurve für  $\beta = 0.6$

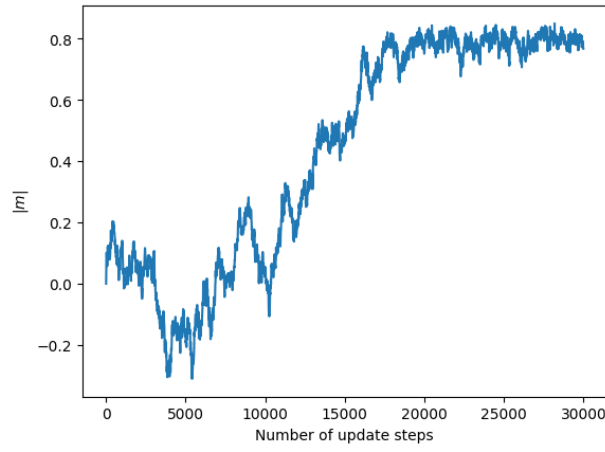


(c) Thermalisierungskurve für  $\beta = 0.7$

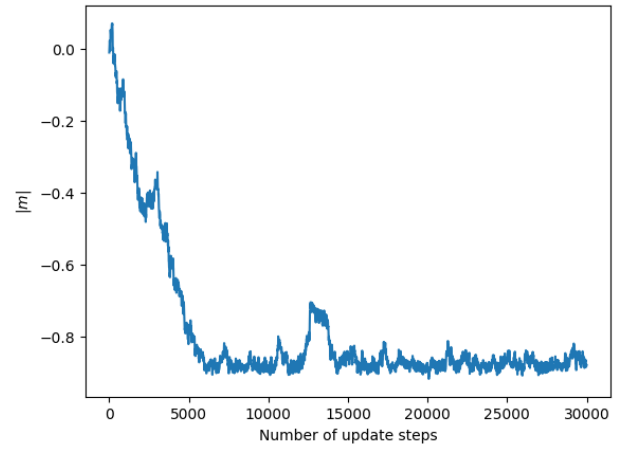


(d) Thermalisierungskurve für  $\beta = 0.8$

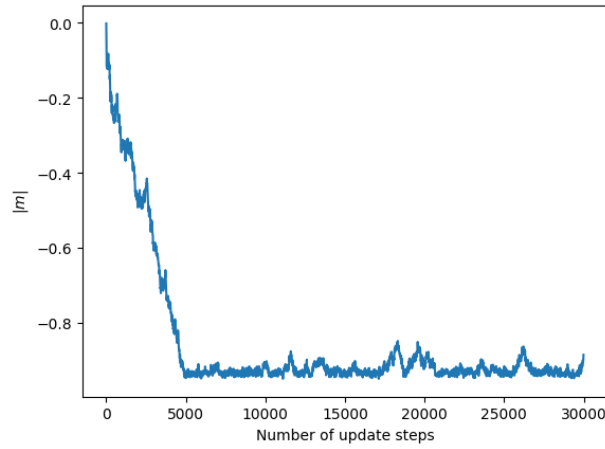
Abbildung 4: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für 3 multihits



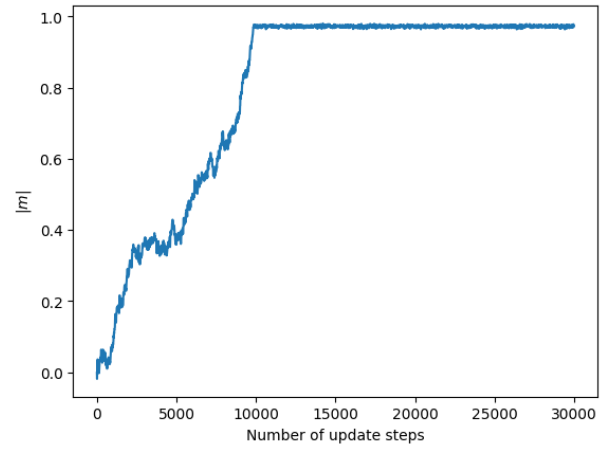
(a) Thermalisierungskurve für  $\beta = 0.46$



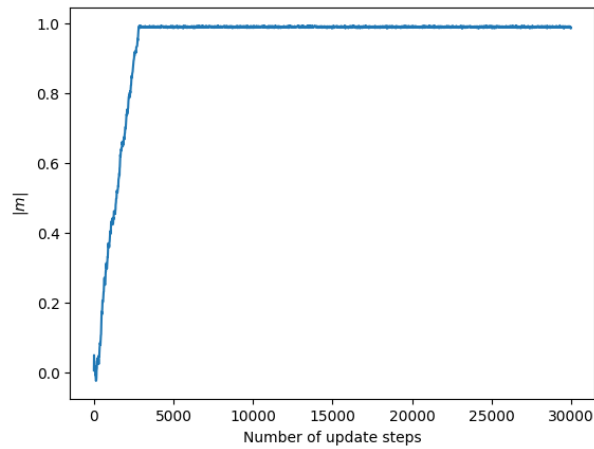
(b) Thermalisierungskurve für  $\beta = 0.49$



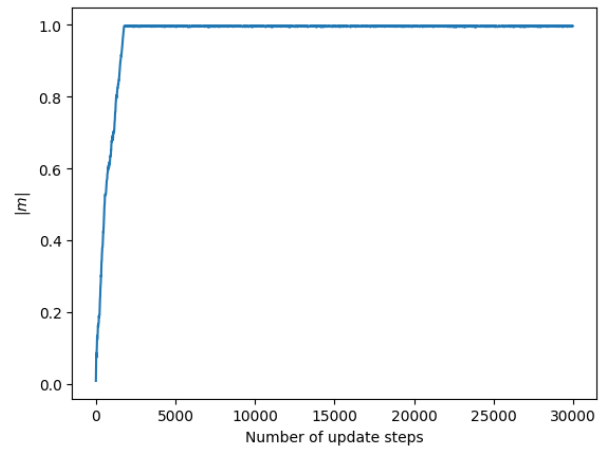
(c) Thermalisierungskurve für  $\beta = 0.53$



(d) Thermalisierungskurve für  $\beta = 0.6$

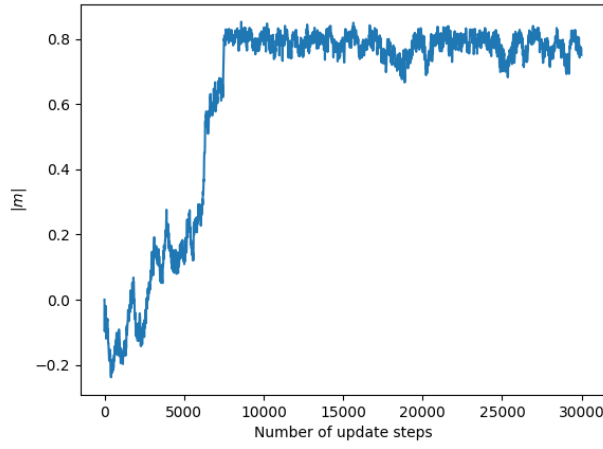


(e) Thermalisierungskurve für  $\beta = 0.7$

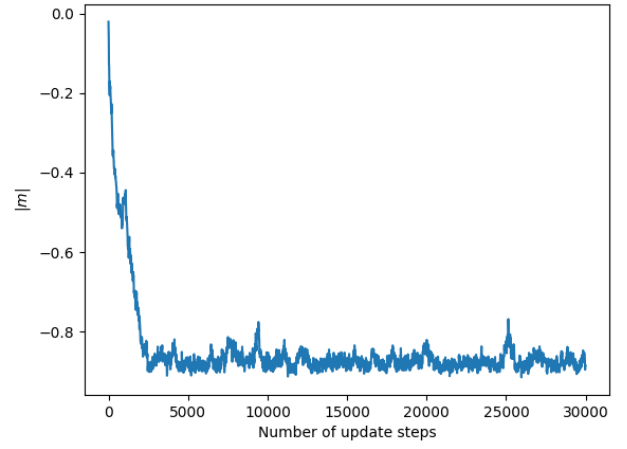


(f) Thermalisierungskurve für  $\beta = 0.8$

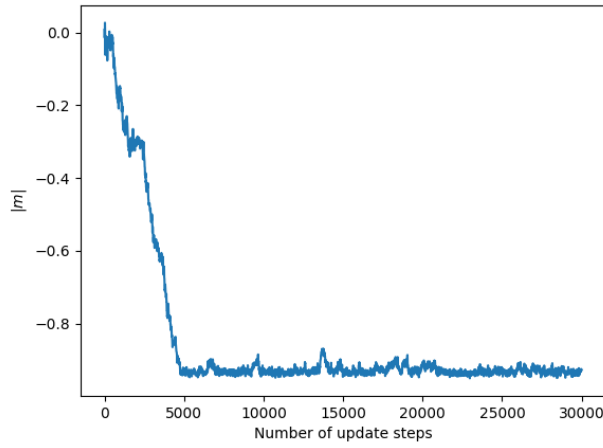
Abbildung 5: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für 4 multihits



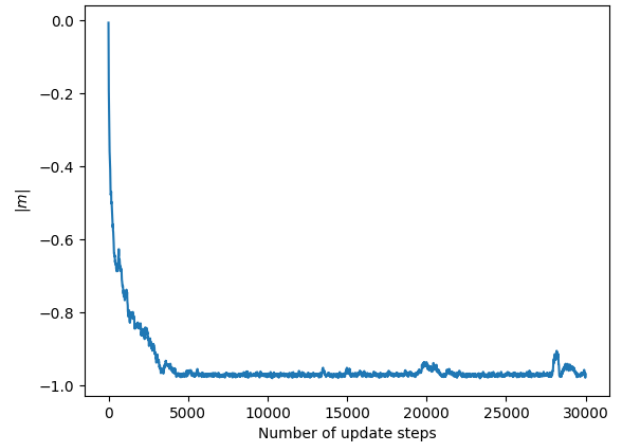
(a) Thermalisierungskurve für  $\beta = 0.46$



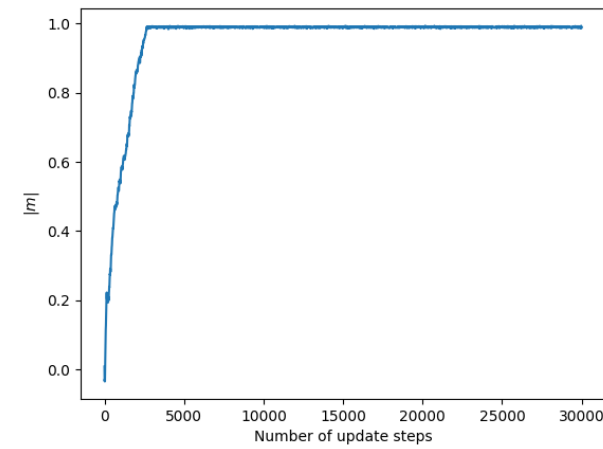
(b) Thermalisierungskurve für  $\beta = 0.49$



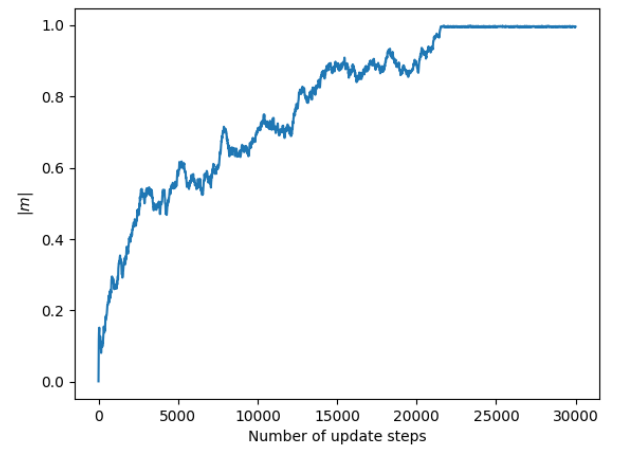
(c) Thermalisierungskurve für  $\beta = 0.53$



(d) Thermalisierungskurve für  $\beta = 0.6$

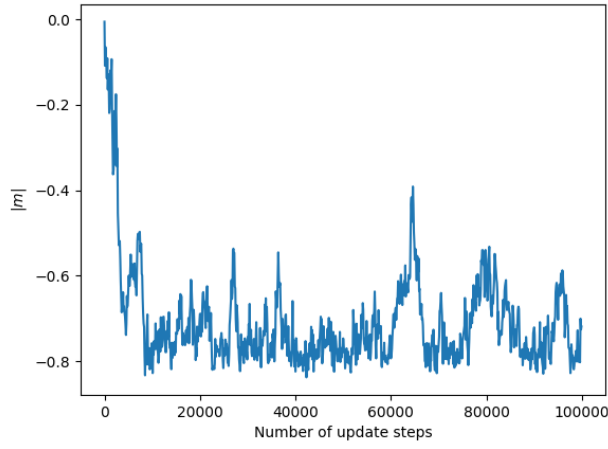


(e) Thermalisierungskurve für  $\beta = 0.7$

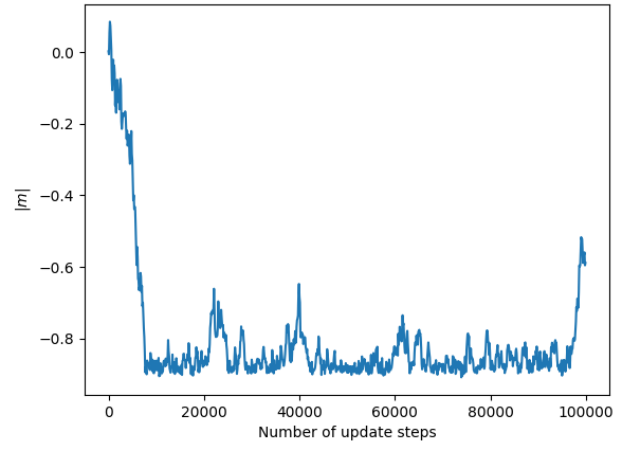


(f) Thermalisierungskurve für  $\beta = 0.8$

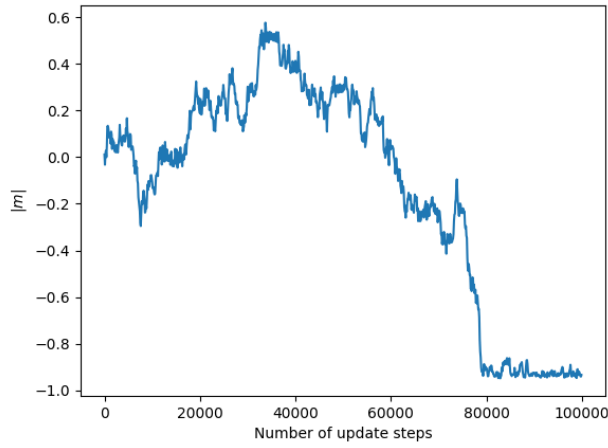
Abbildung 6: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für 5 multihits



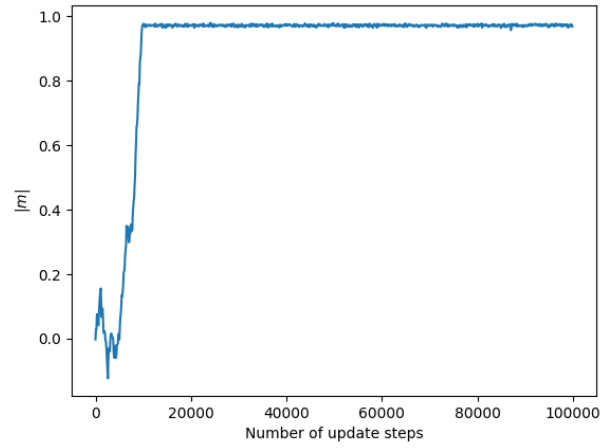
(a) Thermalisierungskurve für  $\beta = 0.46$



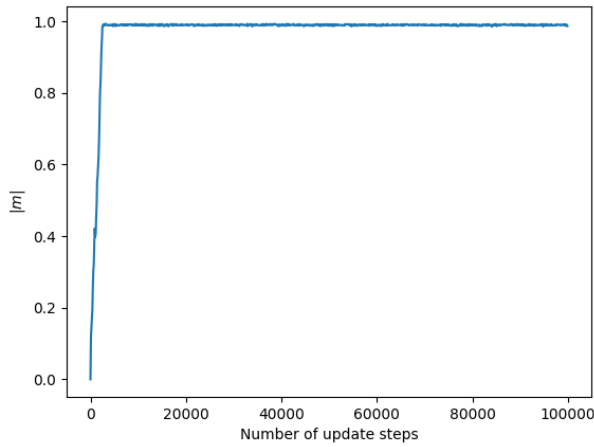
(b) Thermalisierungskurve für  $\beta = 0.49$



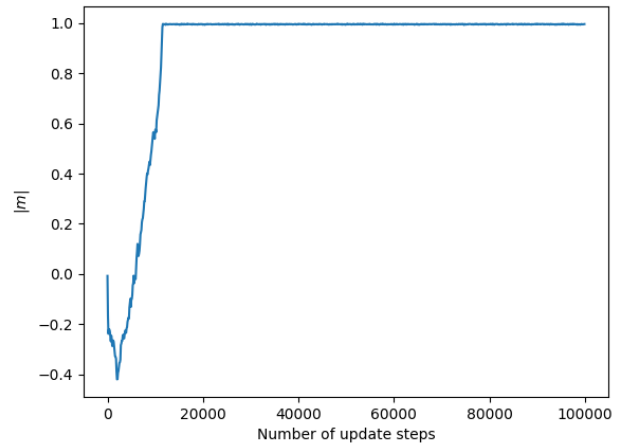
(c) Thermalisierungskurve für  $\beta = 0.53$



(d) Thermalisierungskurve für  $\beta = 0.6$



(e) Thermalisierungskurve für  $\beta = 0.7$



(f) Thermalisierungskurve für  $\beta = 0.8$

Abbildung 7: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für 6 multihits

Es ist zu erkennen, dass bei 5 multihits sich die Thermalisierung am schnellsten einstellt. Außerdem sind für verschiedene Temperaturen unterschiedliche Thermalisierungszeiten zu erwarten. Ich wähle für 5 multihits folgende Anzahlen an Decorrelierungsschritten.

- für  $\beta \leq 0.4$  6 000 Schritte
- für  $0.4 < \beta \leq 0.47$  10 000 Schritte
- für  $0.47 \geq \beta \leq 0.7$  7 000 Schritte
- für  $\beta > 0.7$  10 000 Schritte

Mit diesen Einstellungen ließ ich das Programm jeweils 100 Messpunkte für 35 Temperaturwerte aufnehmen, was ca. 1h 56min dauerte. Die Ergebnisse sind in folgendem Diagramm dargestellt.

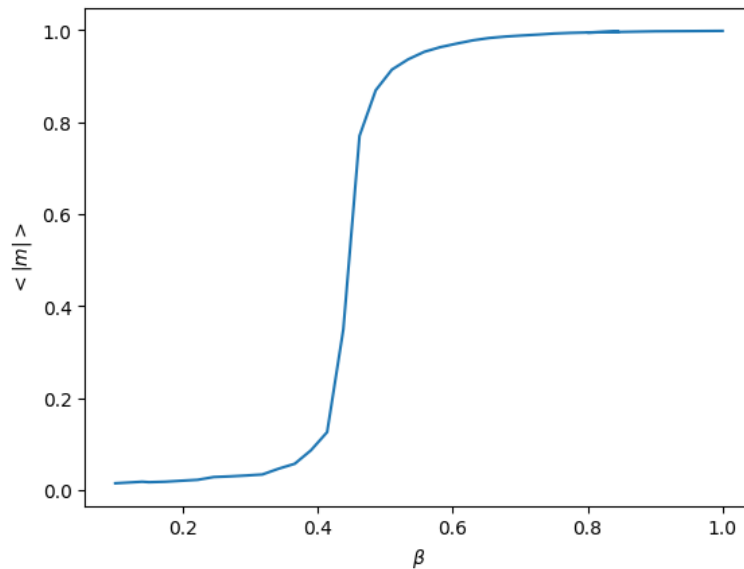


Abbildung 8: Magnetisierungskurve ermittelt durch den Metropolis Algorithmus

Als nächstes berechne ich noch  $\langle |m| \rangle$  und  $\langle m^2 \rangle$  am kritischen Punkt der Onsager Lösung für Gitter der Größe 32x32, 8x8, sowie 4x4 berechnen. Letzteres Ergebnis kann ich verwenden, um die Güte des Algorithmus für

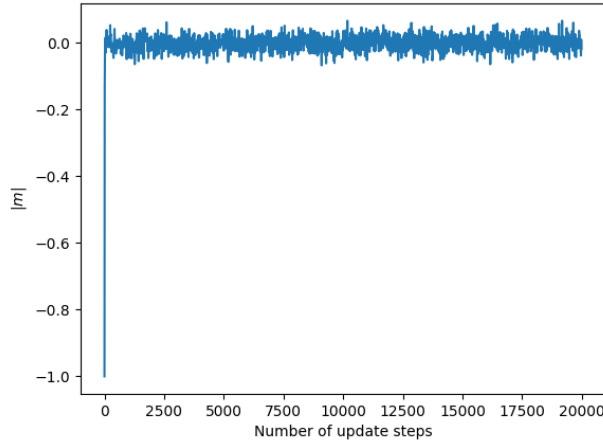
kleine Gittergrößen abzuschätzen, in dem ich es mit dem nach Aufgabe 2 exakt berechneten Wert vergleiche.

therm. sweeps	32x32	8x8	4x4	4x4	4x4(exakt)
	150 000	150 000	160 000	500 000	
$\langle  m  \rangle$	0.682	0.783	0.822	0.84	0.844
$\langle m^2 \rangle$	0.49	0.647	0.733	0.7595	0.761

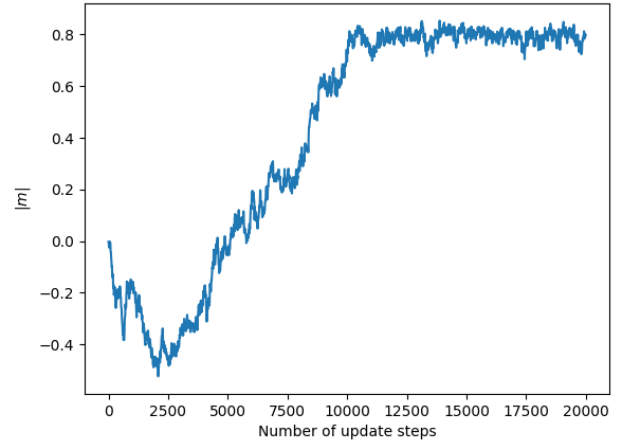
Tabelle 1: Diese Werte wurden für eine Temperatur von  $\beta = 0.4406868$  berechnet. In der ersten Zeile ist angegeben, wie groß ich die Dekorrelationszeit gewählt habe.

## Aufgabe 4

In dieser Aufgabe werde ich mich nun mit dem Wärmebad Update Schema auseinander setzen. Ich vergleiche dessen Performance mit dem Metropolis Algorithmus, indem ich mit dem Wärmebad Schema noch einmal die Aufgabe aus dem vorherigen Abschnitt bearbeite. Zunächst muss wieder die Decorrelierungslänge bestimmt werden.

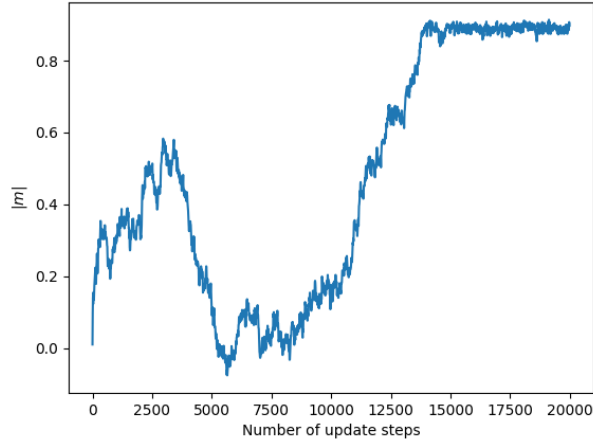


(a)  $\beta = 0.30$

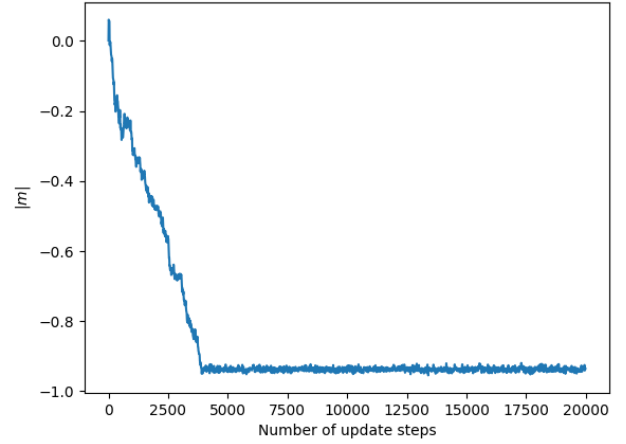


(b)  $\beta = 0.46$

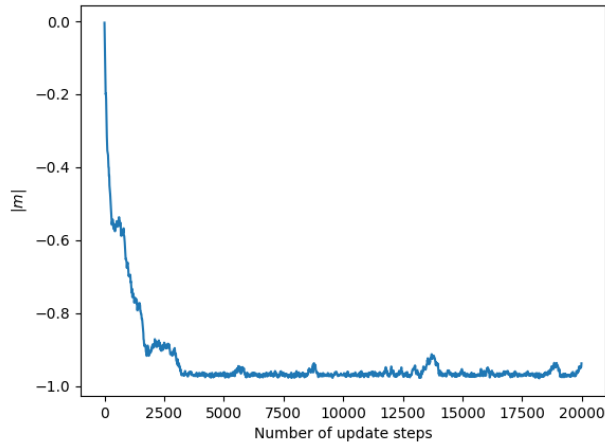
Abbildung 9: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für den Heat Bath Algorithmus



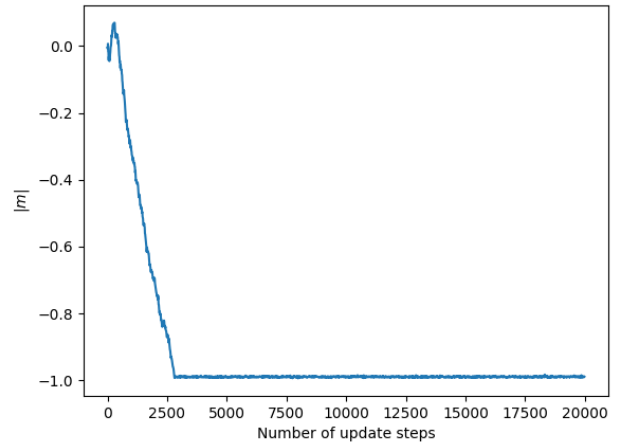
(a) Thermalisierungskurve für  $\beta = 0.49$



(b) Thermalisierungskurve für  $\beta = 0.53$



(c) Thermalisierungskurve für  $\beta = 0.6$



(d) Thermalisierungskurve für  $\beta = 0.7$

Abbildung 10: Thermalisierungsverhalten für verschiedene  $\beta$ -Werte für den Heat Bath Algorithmus

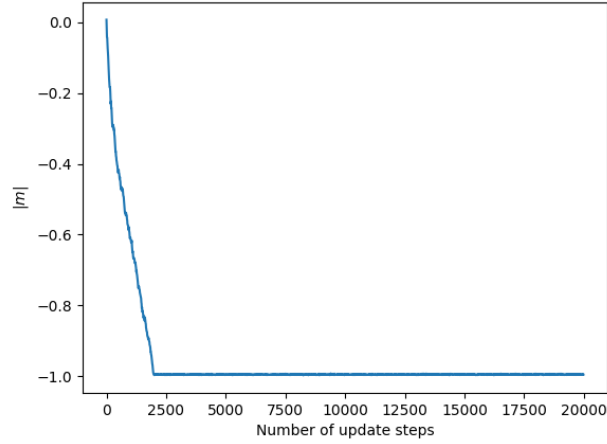


Abbildung 11: Thermalisierungskurve für  $\beta = 0.8$  für den Heat Bath Algorithmus

Aus diesen Thermalisierungskurven wähle ich für  $\beta \in [0.44, 0.53]$  ein Decorrelierungszeit von 15 000 Iterationen und ansonsten 5 000 Iterationen. Mit diesen Parametern erhalte ich folgende Magnetisierungskurve.

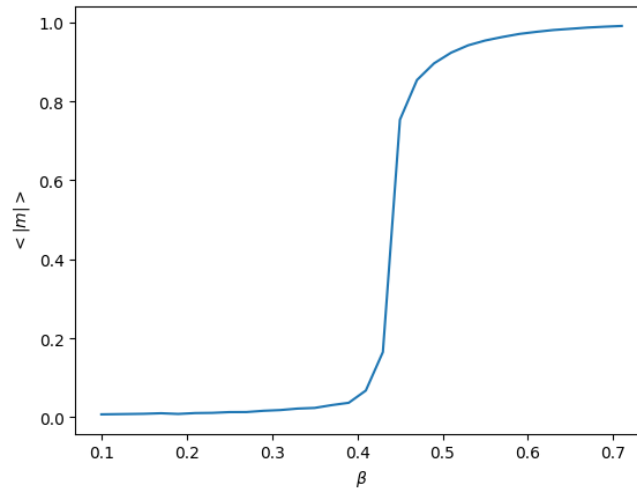


Abbildung 12: Magnetisierungskurve ermittelt durch den Heat Bath Algorithmus



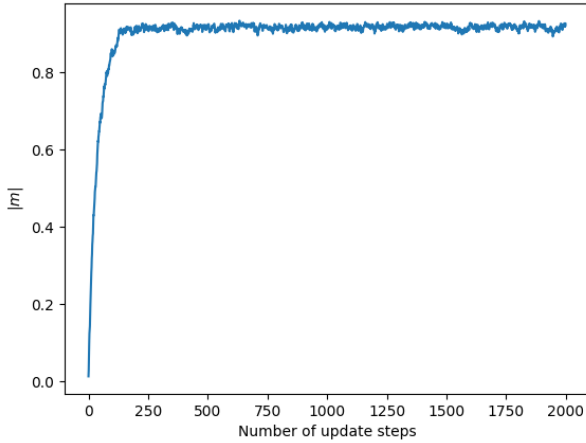
Auch hier kann ich die absolute Magnetisierung für kleinere Gitter am kritischen Punkt ausrechnen.

	32x32	8x8	4x4	4x4(exakt)
$\langle  m  \rangle$	0.524	0.752	0.879	0.844
$\langle m^2 \rangle$	0.319	0.605	0.799	0.761

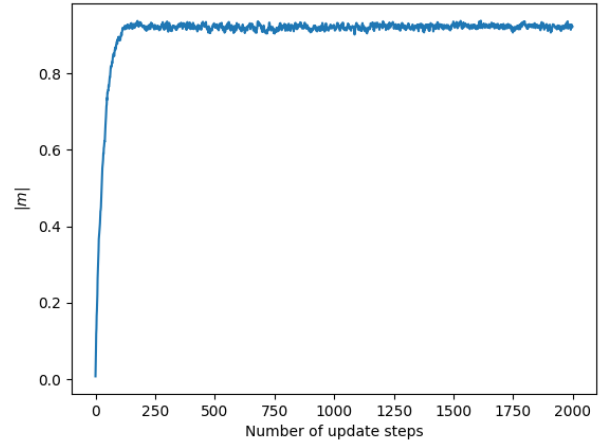
Tabelle 2: Diese Werte wurden für eine Temperatur von  $\beta = 0.4406868$  berechnet.

Die Ergebniss für das 4x4 Gitter stimmen dabei gut mit der exakten Lösung überein, obwohl eine Monte-Carlo Simulation bei diesen Dimensionen nicht von Vorteil ist.

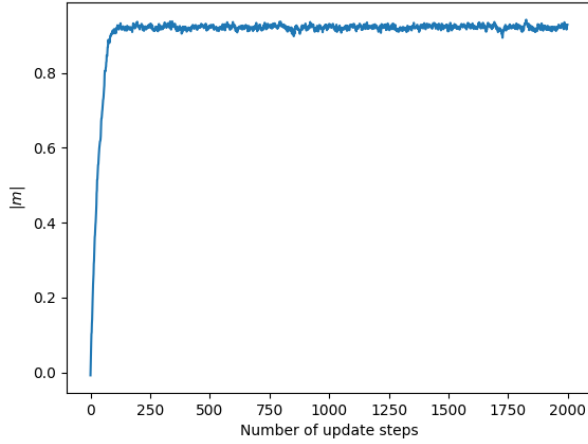
Als Abschluss lässt sich noch die Wirkung eines externen Magnetfeldes untersuchen. Dafür genügt eine leichte Modifizierung der Hamilton Funktion.



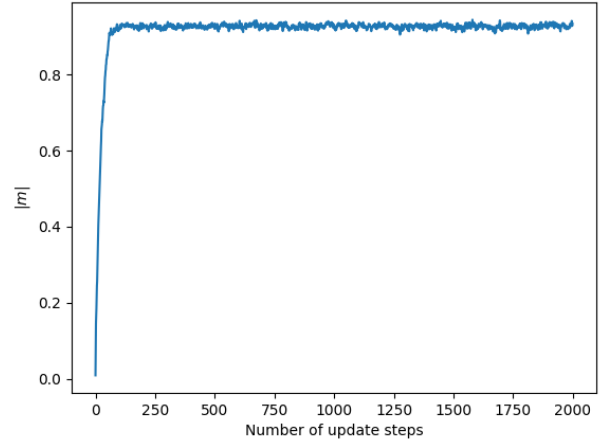
(a) Thermalisierungskurve für  $h = 0.04$



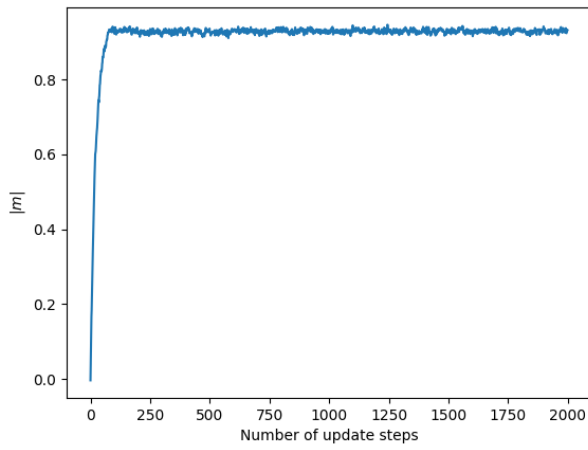
(b) Thermalisierungskurve für  $h = 0.05$



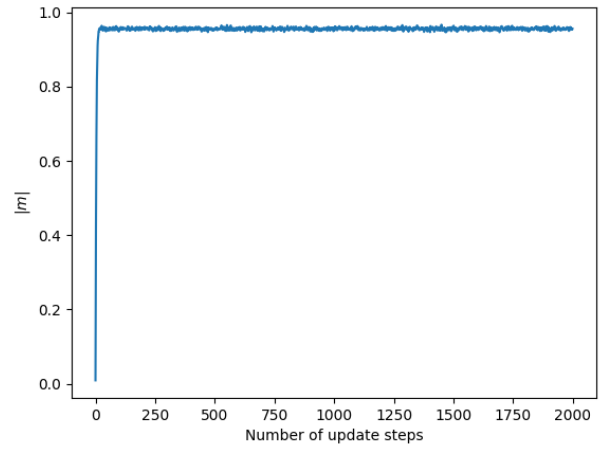
(a) Thermalisierungskurve für  $h = 0.06$



(b) Thermalisierungskurve für  $h = 0.07$



(c) Thermalisierungskurve für  $h = 0.08$



(d) Thermalisierungskurve für  $\beta = 0.3$

Abbildung 14: Thermalisierungsverhalten für verschiedene  $h$ -Werte für  $\beta = 0.6$

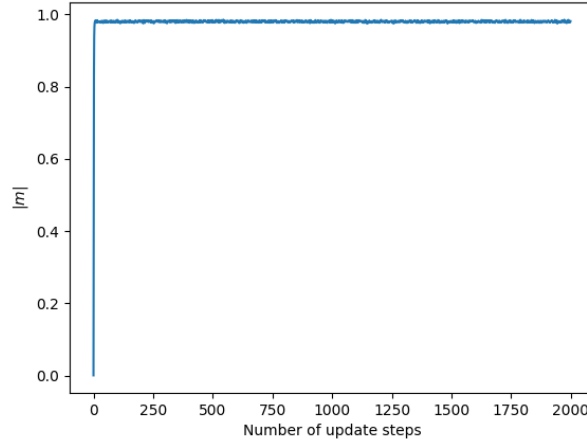


Abbildung 15: Thermalisierungsverhalten für  $h = 0.85$  für  $\beta = 0.6$

Wie zu erkennen ist braucht das System sehr wenige Updates bis es thermalisiert wird. Ich verwende daher als Dekorrelationszeit 500 Iterationen. Um das "Hysterese"-Verhalten zu beobachten, führe ich wieder nacheinander die Messungen für die mittlere absolute Magnetisierung durch. Allerdings ist es hier wichtig, das System nicht für jeden neuen Wert des Magnetfeldes erneut zu initialisieren, sondern dessen Konfiguration in die neue Messung mitzunehmen. Wenn ich bei meinem größten Wert des externen Magnetfeldes angekommen bin, führe ich von hier die Messung noch einmal komplett in die andere Richtung durch. Wenn das tue erhalte ich folgende Magnetisierungskurve.

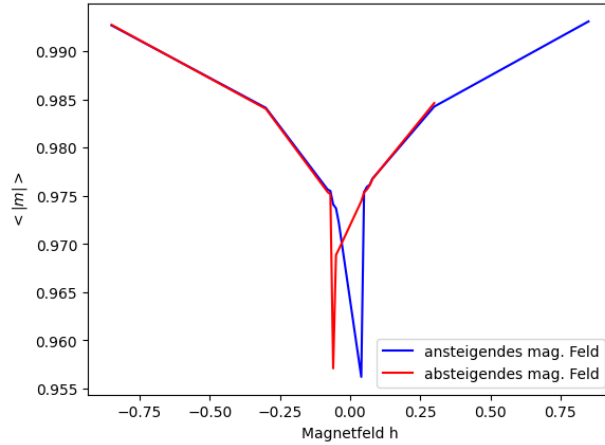


Abbildung 16: "Hysteresis"-Kurve

Trotz der geringen Messpunkte, die ich aufgrund des hohen Zeitaufwands für die Messungen durchgeführt habe, ist zu erkennen, dass sich der Magnetisierungswert für kleine äußere Magnetfelder für verschiedene Durchlaufrichtungen unterscheidet und davon abhängt, ob aus welcher Ausrichtung des Systems man kommt. Dies ist das gleiche Verhalten, dass auch bei Hysteresis beobachtet wird, allerdings handelt es sich hier nicht um eine akurate Simulation dieses physikalischen Phänomens, sondern ist viel mehr ein Zeichen der Schwäche der Implementation. In einem idealen Algorithmus sollte die Decorrelationslänge so gewählt sein, dass es eben keine Korrelation zwischen dem momentanen und den vorherigen Messungen gibt. Dies ist offenbar bei meiner Implementierung nicht der Fall.

## 1 Fazit

Dieser Versuch half mir zum einen zu verstehen, wie man beliebige Wahrscheinlichkeitsverteilungen generiert, und wie man dafür Wahrscheinlichkeitsfunktionen invertiert. Zum anderen lernte ich einen am Fall des Ising Modells den Typ der Monte-Carlo Simulationen, als wirkungsvolles, wenn auch rechenaufwändiges, verfahren zum Lösen von hochdimensionalen Integralen oder Summen kennen. Trotz dem größeren Maß an Zeit ist es jedoch wichtig, die Dekorrelationszeit groß genug zu wählen, was insbesondere auch bei der Hystersekurve deutlich wurde.