



Computational Physics

Exercise Sheet 10

RENE-MARCEL LEHNER

Summer Term
2023

Vorwort	2	E1: Monte Carlo Simulation of a Single Spin	4
E0: Comprehension questions.....	3	E2: Two-Dimensional Ising Model.....	7

Vorwort

Wie auch schon in den Aufgaben selbst erwähnt, sind ist die Programmierung hier einmal dual mit `pybind11` gemacht. Da ich unsicher bin, ob das auf euren lokalen Systemen installiert ist, liefere ich die kompilierten `Solver` binaries mit; dann fehlt glaube ich nur noch `pybind11` über `pip`. `E1` habe ich so geschrieben, dass man 2 Zeilen löschen kann und dann alles über Python läuft. Bei `E2` habe ich das nicht mehr gemacht



E0: Comprehension questions

Task:

- (1) What is the difference between molecular dynamics (MD) simulation and Monte Carlo (MC) simulation for molecular systems? When is each method suitable?
- (2) What are the fundamental properties of a Markov process?

E1: Monte Carlo Simulation of a Single Spin

Task:

Simulate a single spin $\sigma = \pm 1$ using the Metropolis algorithm with the energy

$$\mathcal{H} = -\sigma H$$

in an external magnetic field H . In the Metropolis algorithm, offer spin flips $\sigma \rightarrow -\sigma$.

Numerically calculate the magnetization m at $k_B T = 1$. In other words, perform Monte Carlo simulations for at least 10^4 values of $H \in [-5, 5]$, each with 10^5 steps, and compare your numerical results with the analytical result

$$m = \tanh(\beta H).$$

In dieser Aufgabe wenden wir den Metropolis-Algorithmus auf ein einfaches Modell eines magnetischen Spins an. Das Metropolis-Verfahren ist eine Monte-Carlo-Methode, die in der statistischen Physik verwendet wird, um Systeme zu simulieren, die durch bestimmte Wahrscheinlichkeitsverteilungen beschrieben werden.

Das in dieser Aufgabe vorliegende System besteht aus einem einzigen magnetischen Spin, der sich in einem externen Magnetfeld H befindet. Der Spin kann zwei Zustände annehmen: "up" (also $\sigma = 1$) oder "down" (also $\sigma = -1$). Die Energie dieses Spinsystems ist durch die Hamilton-Funktion gegeben als $\mathcal{H} = -\sigma H$.

Der Metropolis-Algorithmus arbeitet durch zufällige "Vorschläge" für Zustandsänderungen - in unserem Fall schlagen wir vor, den Spin umzudrehen ($\sigma \rightarrow -\sigma$) - und akzeptiert oder lehnt diese Vorschläge dann basierend auf der Änderung in der Energie des Systems ab. Ein Vorschlag, der zu einer niedrigeren Energie führt, wird **immer** akzeptiert, während ein Vorschlag, der zu einer höheren Energie führt, mit einer **Wahrscheinlichkeit** akzeptiert wird, die durch die Boltzmann-Statistik ($\exp(-\beta \Delta E)$) gegeben ist. Hier ist ΔE die Änderung der Energie des Systems, die durch den Vorschlag verursacht wird.

Durch die Durchführung dieses Prozesses für viele Schritte und die Aufzeichnung des Zustands des Spins bei jedem Schritt, können wir eine Schätzung für die durchschnittliche Magnetisierung des Systems $m = \langle \sigma \rangle$ berechnen, wobei $\langle \dots \rangle$ den Durchschnitt über alle Schritte meint.

Diese numerische Berechnung wird dann mit dem analytischen Ergebnis $m = \tanh(\beta H)$ verglichen, wobei $\beta = 1/(k_B T)$ der Boltzmann-Faktor ist. In dieser Aufgabe setzen wir $k_B T = 1$.

Ziel dieser Aufgabe ist es, die Simulation für eine Vielzahl von Werten des externen Magnetfelds H durchzuführen und die numerischen Ergebnisse mit der analytischen Vorhersage zu vergleichen.

Ergebnisse

In Abbildung 1.1 sind die numerischen Ergebnisse für die Magnetisierung m als Funktion des externen Magnetfelds H dargestellt. Die numerischen Ergebnisse (blaue Linie) stimmen gut mit der analytischen Vorhersage (orange Linie) überein. Die numerischen Ergebnisse wurden mit 10^5 Schritten für 10^4 Werte von H berechnet.

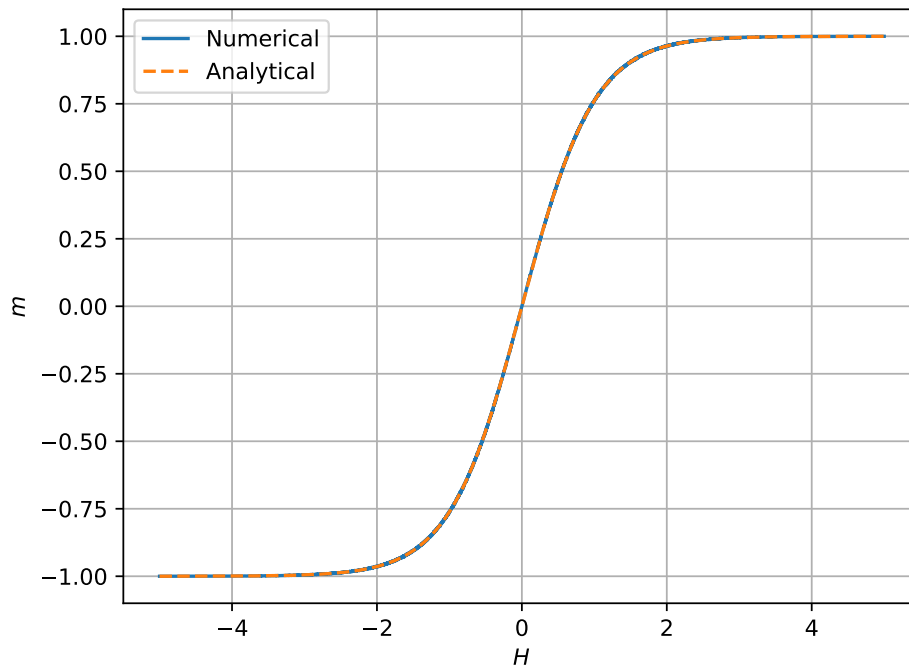


Figure 1.1: Magnetisierung m als Funktion des externen Magnetfelds H für $k_B T = 1$. Die numerischen Ergebnisse (blaue Linie) stimmen gut mit der analytischen Vorhersage (orange Linie) überein.

Programmcode

Der Metropolis-Algorithmus ist sowohl in C++ als auch in Python implementiert. Über die Datei `main.py` können drei verschiedene Modi gewählt werden. Dafür werden verschiedene Argumente bei Programmausführung übergeben.

Die Einbindung des C++-Codes erfolgt über die `pybind11`-Bibliothek. Für die Kompilierung des C++-Codes muss in dem Verzeichnis `make` ausgeführt werden. Danach kann der Python-Code ausgeführt werden.

Die drei Modi sind:

- `python main.py -python`: Ausführung des Python-Codes.
- `python main.py -cpp`: Ausführung des C++-Codes.
- `python main.py -compare`: Ausführung des Python-Codes und des C++-Codes und Vergleich der Ergebnisse. Hier wird zudem die Laufzeit der beiden Programme verglichen.

Aus Vorführungszwecken ist die Anzahl der Schritte von H auf 10^2 reduziert. Für die eigentliche Berechnung wurden 10^4 Schritte verwendet.

Für die Ausführung des Codes ist `pybind11` erforderlich. Falls `pybind11` nicht installiert ist, läuft der Code dennoch, sofern zwei Stellen in `main.py` entfernt werden. Die Stellen sind eindeutig markiert.

Wird kein `pybind11` verwendet, muss `make` auch nicht ausgeführt werden.

E2: Two-Dimensional Ising Model

Task:

Simulate the two-dimensional Ising model without a magnetic field using the Metropolis algorithm, with the energy given by

$$\mathcal{H} = - \sum_{i,j \text{ n.n.}} \sigma_i \sigma_j$$

where the sum is taken over nearest neighbors (n.n.). Use a square lattice of size 100×100 with periodic boundary conditions. (If there are issues with computation time, the lattice can be slightly reduced.)

Offer spin flips of randomly selected spins in the Metropolis algorithm. Choose randomly oriented spins or completely ordered spins as initial conditions. After a sufficiently long equilibration phase, perform 10^4 to 10^5 sweeps, where on average each spin is offered a flip once.

1. Generate and discuss graphical snapshots of the system for $k_B T = 1$ and $k_B T = 3$.

Consider multiple temperatures $k_B T \in [1.5, 3]$ for the following parts of the exercise. The range around the critical temperature (in the thermodynamic limit)

$$k_B T_c = \frac{2}{\ln(1 + \sqrt{2})} \approx 2.27$$

is particularly interesting, so it is recommended to resolve the temperature more precisely in this range. For parts of the exercise where plots as a function of time are required, it is sufficient to create plots for the two boundary values of the temperature range and for an intermediate value.

2. First, examine the equilibration phase by measuring the average energy per spin

$$e(t) = \frac{E(t)}{N} = \frac{\langle \mathcal{H}(t) \rangle}{N}$$

as a function of simulation time t , where N is the total number of spins. How long do you have to wait until the result becomes independent of the initial conditions? Examine both initial conditions.

3. Next, consider the time after equilibration. Calculate the averages of the energy, the magnetization

$$\langle m \rangle = \left\langle \frac{1}{N} \sum_i \sigma_i \right\rangle$$

and the absolute value of the magnetization

$$\langle |m| \rangle = \left\langle \frac{1}{N} \left| \sum_i \sigma_i \right| \right\rangle$$

per spin. How does the behavior of the system differ above and below T_c ? How does the magnetization $\langle m \rangle(t)$ behave as a function of simulation time at different temperatures?

4. In your simulation, calculate the specific heat per spin from the energy fluctuations

$$c(T) = \frac{\langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2}{k_B T^2 N}.$$

Allgemeines

Die Achsenbeschriftungen sind nicht perfekt und es "fehlen" Einheiten.

Das Programm ist wieder mit `pybind11` kompiliert. Ich liefere die kompilierte `Solver` Datei mit, damit man dies nicht auch noch tun müsste.

Anfangsbedingungen

Für die Simulation werden per Aufgabenstellung zwei Anfangsbedingungen vorgeschlagen; eine vollständig parallele und eine zufällige Ausrichtung der Spins.

Dies ist insofern relevant, als dass sich das System bei verschiedenen Anfangsbedingungen unterschiedlich entwickelt und sich auf dem Weg in das globale Minimum in **lokalen** Minima verfangen kann.

Für alle Simulationen in `E2` wurde dir **zufällige** Ausrichtung der Spins gewählt.

Für die verschiedenen Aufgabenteile wurde die folgende Anzahl von Sweeps verwendet:

- Aufgabenteil 1: **Simulation**: 10^4 Sweeps, **Equilibrierung**: 10^3 Sweeps
- Aufgabenteil 2: **Equilibrierung**: 10^4 Sweeps
- Aufgabenteil 3: **Simulation**: $2 * 10^5$ Sweeps, **Equilibrierung**: Dynamisch zwischen 10^3 und $2.5 * 10^3$ Sweeps
- Aufgabenteil 4: Wie 3.

1. Snapshots

Die Snapshots zu den Energien $k_B T = 1$ und $k_B T = 3$ sind in den Abbildungen `1.2`, `1.3` und `1.4` zu sehen.

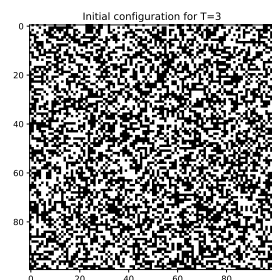
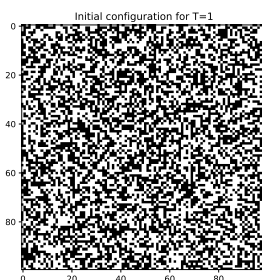


Figure 1.2: Initial configuration of the spins for $k_B T = 1$ (left) and $k_B T = 3$ (right).

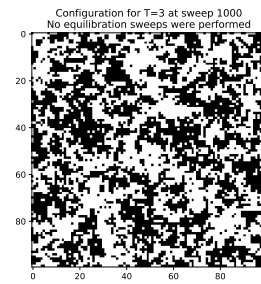
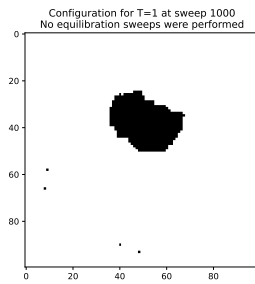


Figure 1.3: Configuration after 1000 sweeps for $k_B T = 1$ (left) and $k_B T = 3$ (right).

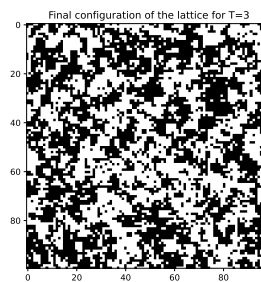
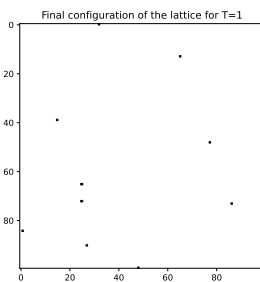


Figure 1.4: Final configuration of the spins for $k_B T = 1$ (left) and $k_B T = 3$ (right).

Interpretation

Für $k_B T = 1$ richten sich alle Spins nach einer relativ kurzen Zeit parallel aus. Ab und zu wird beobachtet, dass das System einen großen Streifen ausbildet und sich kaum ändert. Dies wäre dann ein lokales Minimum, in dem sich das System verfängt. Hier ist aber das globale Minimum gezeigt.

Für $k_B T = 3$ ist zu sehen, dass sich die Spins nicht parallel ausrichten, sondern eher in einem Muster, das an eine Zufallsverteilung erinnert. Dies ist auch zu erwarten, da die Temperatur höher ist und die Spins somit mehr Energie haben, um sich zu bewegen.

2. Equilibrierung

Es ist schwierig genau zu bestimmen, wann ein System vollständig equilibriert ist, da sich das System für Temperaturen unter der kritischen Temperatur in lokalen Minima verfangen kann. Dies ist in Abbildung 1.5 zu sehen. Hier wurde die Energie pro Spin für Temperaturen zwischen $k_B T = 1.5$ und $k_B T = 3$ aufgetragen. Es ist zu sehen, dass die Energie für $k_B T = 3$ nach ca. 10^3 Sweeps equilibriert ist, während die Energie für $k_B T = 1.5$ nach 10^3 Sweeps noch nicht equilibriert ist. Dies ist auch zu erwarten, da die Spins bei höheren Temperaturen mehr Energie haben und sich somit mehr bewegen können.

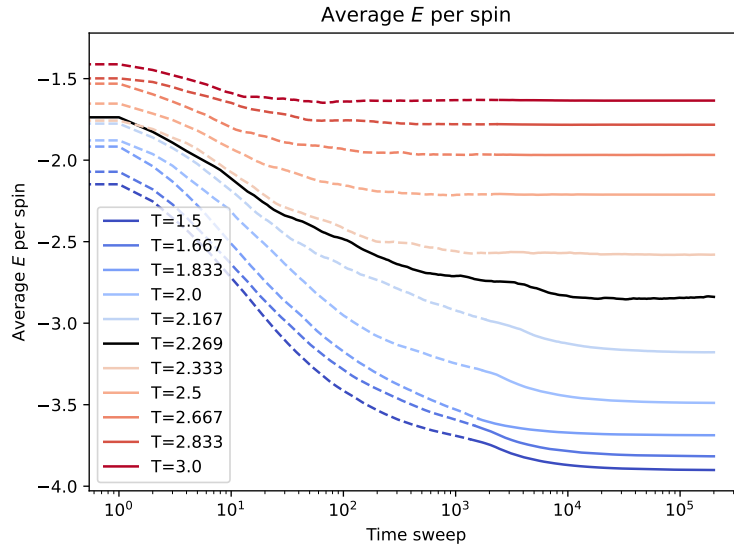


Figure 1.5: Energy per spin for temperatures between $k_B T = 1.5$ and $k_B T = 3$. The critical temperature is $k_B T_c = 2.269$ and is marked by the black line. The dashed lines represent the equilibration time for the different temperatures.

Um eine ungefähre Zeit für die Equilibrierung zu ermitteln, wird über die letzten 500 Sweeps gemittelt und eine Abweichung von etwa 0.1% betrachtet. Die Ergebnisse sind visuell in [Abbildung 1.6](#) dargestellt.

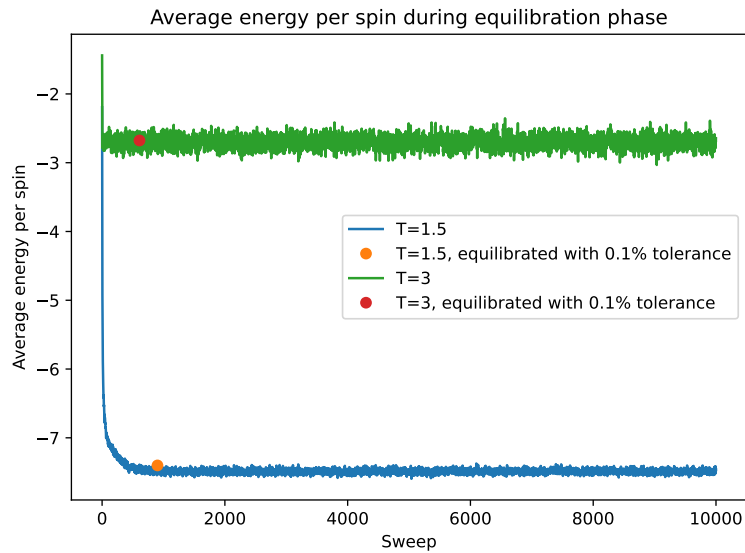


Figure 1.6: Average energy per spin. The dots mark the equilibration time for the different temperatures with a tolerance of 0.1% in the last 500 sweeps.

Für die weitere Simulation werden diese beiden Punkte als Equilibrierungszeit verwendet, indem linear

zwischen ihnen in Abhängigkeit der Temperatur interpoliert wird. Die Equilibrierungszeit ist wie erwähnt als gestrichelte Linie in den Plots dargestellt.

3. Magnetisierung

Die Equilibrierungsphase zeigt ein interessantes Verhalten für unterschiedliche Temperaturen, insbesondere für Temperaturen nahe der kritischen Temperatur. Aus diesem Grund sind die Equilibrierungsphasen mit in die Diagramme aufgetragen; als Kompromiss jedoch die Sweeps auf einer log-scale aufgetragen.

Da sich das System für Temperaturen unter der kritischen Temperatur parallel ausrichten möchte, landet das Modell zufällig in eine vollständige Up- oder Down-Ausrichtung.

Daher ist in Abbildung 1.7 zu sehen, dass die blauen Kurven gegen -1 und 1 konvergieren, die roten Kurven jedoch bei 0 verbleiben, da sich im Mittel alle Spins wegheben und es gleiche Anteile von Spin-Up und Spin-Down gibt.

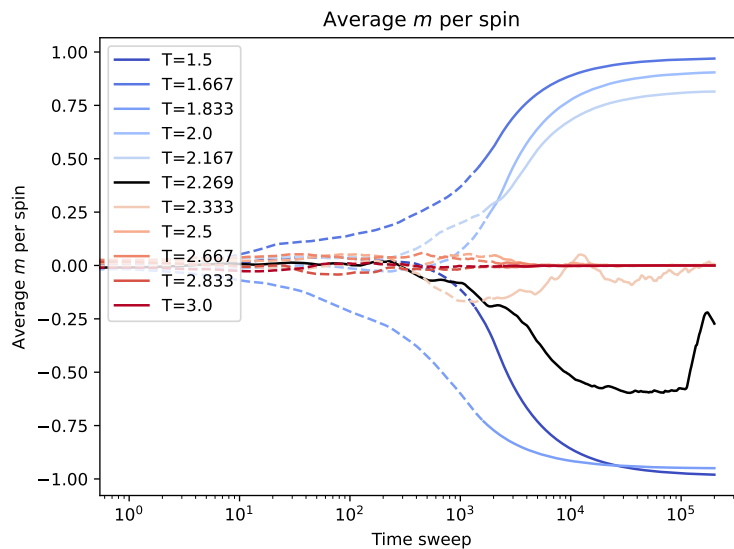


Figure 1.7: Average magnetization for temperatures between $k_B T = 1.5$ and $k_B T = 3$. The critical temperature is $k_B T_c = 2.269$ and is marked by the black line. The dashed lines represent the equilibration time for the different temperatures.

Wird der Betrag der durchschnittlichen Magnetisierung betrachtet, wird etwas deutlicher, dass sich alle Systeme unterhalb der kritischen Temperatur parallel ausrichten und alle Systeme oberhalb der kritischen Temperatur gleichverteilte Spins haben. Die kritische Temperatur ist genau der Punkt, an dem die Magnetisierung gegen 0.5 geht und sich das System parallel ausrichten möchte, aber zu viel Energie besitzt, um diese Konfiguration zu erreichen.

Die absoluten Magnetisierungen sind in Abbildung 1.8 dargestellt.

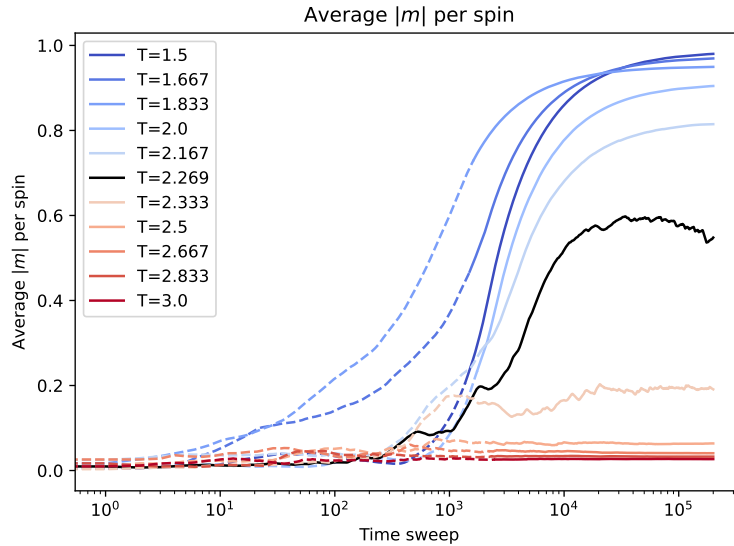


Figure 1.8: Absolute value of the average magnetization for temperatures between $k_B T = 1.5$ and $k_B T = 3$. The critical temperature is $k_B T_c = 2.269$ and is marked by the black line. The dashed lines represent the equilibration time for the different temperatures.

4. Wärmekapazität

Die Wärmekapazität ist in Abbildung 1.9 dargestellt. Es ist deutlich zu sehen, dass die Wärmekapazität maximal für die kritische Temperatur ist. Das bedeutet also, dass das System gegenüber allen anderen Systemen am meisten thermische Energie aufnehmen kann. Für Temperaturen unter- und oberhalb der kritischen Temperatur ist die Wärmekapazität geringer; für kleinere Temperaturen scheint das System noch nicht vollständig equilibriert zu sein, weswegen eine qualitative Aussage darüber, ob die Wärmekapazität für Temperaturen unterhalb der kritischen Temperatur geringer oder höher als für Temperaturen oberhalb der kritischen Temperatur ist, nicht getroffen werden kann.

Die Einheit der Wärmekapazität ist J/K und ist hier auf die Anzahl der Spins normiert. Es ist jedoch zu beachten, dass die Größenordnung wesentlich kleiner als die Zahlenwerte des Plots ist, da diese letztlich von der Größenordnung von \mathcal{H} abhängt, welche in der Aufgabenstellung nicht angegeben ist.

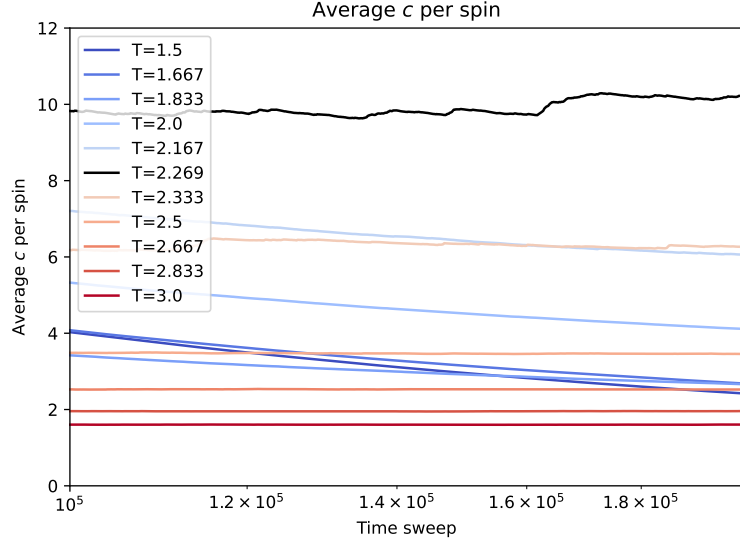


Figure 1.9: Heat capacity for temperatures between $k_B T = 1.5$ and $k_B T = 3$. The critical temperature is $k_B T_c = 2.269$ and is marked by the black line. The dashed lines represent the equilibration time for the different temperatures.