

Fortgeschrittenenpraktikum

Ising-Modell

Wintersemester 2018/19

Prof. Dr. W. Kilian

Simon Braß

Daniel Busch

Oktober 2018

Während dieses Praktikums sollen Sie sich im Rahmen einer *Monte-Carlo*-Simulation mit dem thermodynamischen Verhalten eines magnetischen Materials beschäftigen. Hierfür wird ein einfaches Modell, das *Ising*-Modell, angenommen, aus dem sich auf einfache Weise beispielsweise die Energie und die Magnetisierung des Systems ergeben. Das Kernproblem besteht dann darin, ein thermodynamisches Gleichgewicht zu simulieren. Dies soll unter Zuhilfenahme des *Metropolis*-Algorithmus erfolgen.

1 Ising-Modell und Metropolis-Algorithmus

Informieren Sie sich mittels geeigneten Quellen (Internet, Bücher¹, Skripte² etc.) über das Ising-Modell und den Metropolis-Algorithmus. Versuchen Sie, folgende Fragen zu beantworten:

- Wie ist die Vorgehensweise bei diesem Algorithmus, um das Herstellen eines thermodynamischen Gleichgewichts zu simulieren?
- Wie lassen sich in diesem Modell die Energie, die Magnetisierung und die spezifische Wärmekapazität für ein gegebenes System bestimmen?
- Was sollte bei Änderung der Temperatur zu beobachten sein?

2 Implementierung des Metropolis-Algorithmus

Implementieren Sie den Metropolis-Algorithmus. Die Implementierung soll in *modernem Fortran* objektorientiert und strukturiert geschrieben werden. Die Entwicklung sollte derart durchgeführt werden, dass Tests für die jeweiligen Unterprogramme geschrieben werden, bevor diese implementiert werden. Bedenken Sie insbesondere, dass der Algorithmus zunächst für eine, aber später auch für zwei und drei Raumdimensionen funktionieren soll.

¹Computational Physics: Problem Solving with Computers, Rubin H. Landau, Manuel J. Páez and Christian C. Bordeianu

²Computational Physics, Jan Kierfeld, TU Dortmund

Die einzelnen Schritte hin zum fertigen Programm sollten für die spätere Dokumentation festgehalten werden. Hierzu empfiehlt sich beispielsweise die Verwendung eines *version control systems (VCS)* wie *git*. Verwenden Sie auch Kommentare in Ihrem Code, sodass man diesen innerhalb kurzer Zeit verstehen kann.

3 Simulation in einer Dimension

Im Folgenden soll der Algorithmus für einige Simulationsaufgaben genutzt werden. Es kann sinnvoll sein, die Lösung der Aufgaben weitestgehend durch Unterprogramme zu automatisieren, die später auch für die Simulationen in zwei und ggfs. drei Dimensionen verwendet werden können.

- a) Simulieren Sie nun eine Kette von N Atomen, die in Kontakt mit einem Wärmebad steht. Nehmen Sie an, dass dabei kein externes magnetisches Feld existiert. Beobachten und beschreiben Sie die Bildung eines thermodynamischen Gleichgewichts, ausgehend von einem Zustand, in dem alle Spins in die gleiche Richtung ausgerichtet sind. Stellen Sie die Zeitentwicklung graphisch dar. Untersuchen Sie dies für unterschiedliche Temperaturen und variieren Sie auch die Größe des Systems, d.h. die Anzahl der Atome in der Kette.
- b) Bestimmen Sie die innere Energie U und die Magnetisierung M in Abhängigkeit der Temperatur T , d.h. ermitteln Sie U und M für eine Reihe von Temperaturwerten. Überlegen Sie sich, wie die statistische Unsicherheit der ermittelten Werte zustandekommt und wie Sie diese reduzieren können. Plotten Sie die Ergebnisse und beschreiben Sie Ihre Beobachtungen? Entsprechen die Verläufe Ihren Erwartungen? Vergleichen Sie die Ergebnisse mit den analytischen Lösungen.
- c) Für große Werte von N benötigt die Bildung eines thermodynamischen Gleichgewichts des Systems innerhalb der Simulation mehr Zeit als für kleine Werte von N . Sie sollten daher einen Kompromiss aus praktikabler Rechenzeit und Genauigkeit finden. Verwenden Sie die analytische Lösung für die innere Energie, um zu überprüfen, ob der gewählte Wert von N für die Simulation geeignet ist und passen Sie diesen gegebenenfalls an.
- d) Berechnen Sie zusätzlich die spezifische Wärmekapazität in Abhängigkeit der Temperatur und vergleichen Sie diese mit der analytischen Lösung.

4 Simulation in zwei (drei) Dimensionen

Führen Sie die Simulationen aus Aufgabe 3 nun in zwei (und ggfs. drei) Dimensionen durch. Erstellen Sie entsprechende Plots und vergleichen Sie die Ergebnisse mit denen der Simulationen in 1D. Sie sollten nun einen Phasenübergang beobachten können. Bestimmen Sie anhand Ihrer Simulationsergebnisse die *Curie*-Temperatur.

5 Zusatzaufgabe

Die Bearbeitung dieser Aufgaben soll nur erfolgen, falls genügend Zeit oder Interesse besteht.

- a) Erweitern Sie Ihren Algorithmus derart, dass er Wechselwirkungen mit übernächsten Nachbarn berücksichtigt.
- b) Erstellen Sie ein Profiling des Programms und finden Sie heraus, welche Teile des Programms zeitkritisch sind. Versuchen Sie, diese zu optimieren.
- c) Versuchen Sie, Ihre Implementierung an geeigneten Stellen unter Verwendung des `OpenMP`-Standards zu parallelisieren. Belegen Sie die Ergebnisse Ihrer Parallelisierung, indem Sie die Ausführungszeit von Simulationsaufgaben messen und dabei die Anzahl der Threads variieren.

6 Protokoll

- a) Erklären Sie kurz die für die Simulation relevanten physikalischen Grundlagen sowie den verwendeten Algorithmus. Dokumentieren Sie dann die einzelnen Schritte Ihrer Implementierung dieses Algorithmus. Beschreiben Sie dabei die verwendeten Datentypen und Prozeduren. Erklären Sie auch, an welchen Stellen des Codes welche Konzepte der objekt-orientierten Programmierung verwendet wurden und warum diese sinnvoll ist.
- b) Dokumentieren Sie Ihre Ergebnisse aus den Aufgaben 3 und 4 mittels Plots und Erklärungen. Gehen Sie auch kurz auf die benötigte Rechenzeit bei den Simulationsaufgaben ein und erklären Sie, wovon diese abhängt.
- c) Bewerten Sie anhand Ihrer Ergebnisse den verwendeten Algorithmus und Ihre eigene Implementierung.
- d) Erklären Sie kurz, wer welche Aufgaben übernommen hat.