Übung 8
Computational Physics III

Matthias Plock (552335)   Paul Ledwon (561764)

5. Juli 2018

Inhaltsverzeichnis

1 Monte-Carlo-Simulation: Magnetisierung 1

2 Planung auf der GPU 1
2.1 Main Funktion auf der CPU 1
2.2 `randgpu(int N)` auf der GPU 1

1 Monte-Carlo-Simulation: Magnetisierung

Neben der Messung der Magnetisierung wurde auch die Wirkung S gemessen. Es sollten die Autokorrelationszeiten fuer die Wirkung und die Magnetisierung verglichen werden, diese lagen bei ungefaehr 2 und 5 Monte-Carlo-Zeitschritten.

Die Definition $\sigma^2 = \frac{2\tau_{int}}{N}\Gamma$ wurde ueberprueft, jedoch unterliegt diese Abweichungen von bis zu zwei Groessenordnungen.

Die benoetigte Monte-Carlo-Zeit bis zur Thermalisierung war nie geringer als ungefaehr 100, nach der Thermalisierung wurden nocheinmal 1000 sweeps durchgefuehrt. Damit ist die Bedingung, dass die Auto-Korrelationszeit viel kleiner sein muss als diese Groessen in guter Naeherung erfuellt.

Fuer die verschiedenen L wurde jeweils fuer die Punkte mit nichtverschwindender Magnetisierung ein Fit mit der Funktion $M(\kappa) = A\sqrt{\kappa - \kappa_c}$ durchgefuehrt, da aus der Vorlesung dieser Zusammenhang in der Naehe von κ_c bekannt war. Dies ist in Abb. 1 dargestellt. Mit steigendem L scheinen die Messpunkte besser dem theoretischen Zusammenhang zu folgen, da ein hoeheres L einer hoeheren Aufloesung des Systems entspricht, ist dies auch zu erwarten. Fuer $L = 16$, also die hoechste hier getestete Aufloesung erhaelt man aus dem Fit

$$\kappa_c = 0.2553 \pm 0.0005$$

2 Planung auf der GPU

2.1 Main Funktion auf der CPU

- `(ptr auf phi auf der GPU) phi = init_phi()` Erstelle ein anfaengliches ϕ
  - `(ptr auf Zufallszahlen auf GPU) rnd = randgpu(3*nvol*ntrial)`
  - `sweep()`

2.2 `randgpu(int N)` auf der GPU

Diese Funktion erstellt N Zufallszahlen auf der GPU und gibt einen Pointer auf dieses GPU Array zurueck.

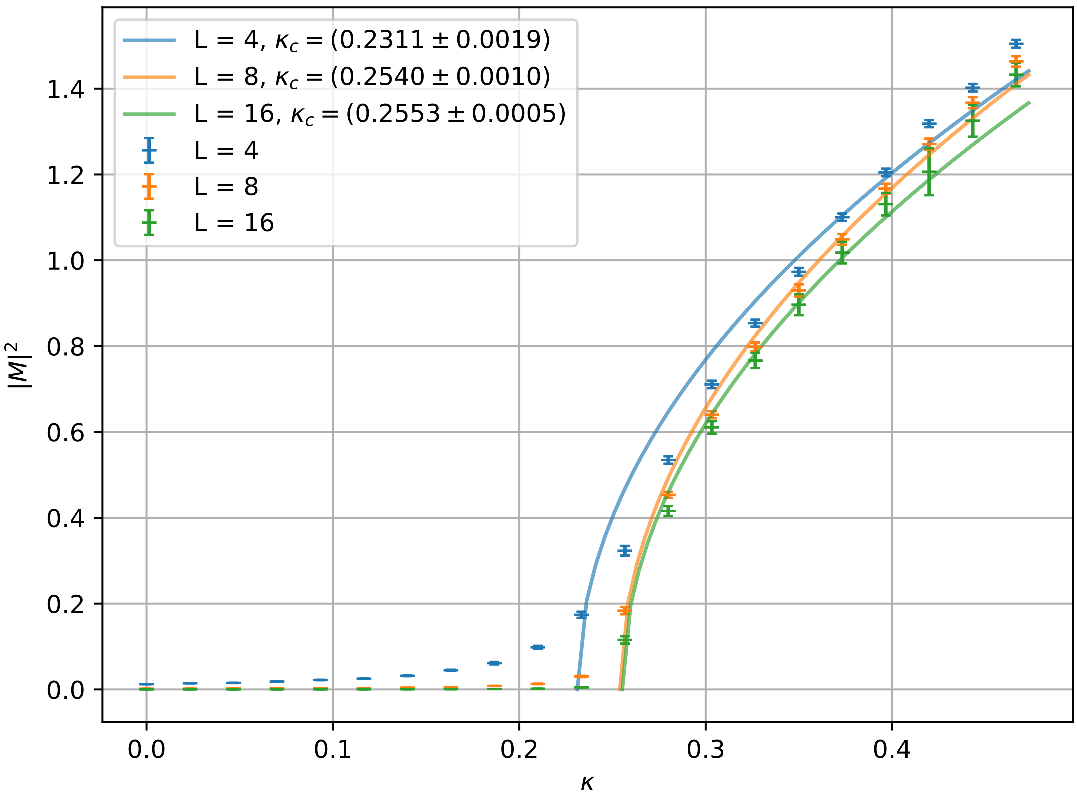

Abbildung 1: Magnetisierung fuer verschiedene L und Bestimmung von κ_c