

# Übungsblatt 7

10.06.2016 / B. Leder

Wissenschaftliches Rechnen III / CP III

## Aufgabe 7.1: *Spin-Modell: lokale Verteilung und Observablen*

Erweitern Sie ihr Programm zum Spin-Modell um:

1. Die lokale Verteilung  $p(\Phi_x)$  aus der Vorlesung. Da wir nur Verhältnisse  $p(\Phi'_x)/p(\Phi_x)$  betrachten werden, spielt die Normierung keine Rolle. Verifizieren Sie Ihre Implementierung, indem Sie überprüfen, dass für zwei Konfigurationen  $\Phi, \Phi'$ , die sich nur durch den Spin am Gitterpunkt  $x$  unterscheiden ( $\Phi_y = \Phi'_y, \forall y \neq x$ ), gilt

$$P[\Phi']/P[\Phi] = p(\Phi'_x)/p(\Phi_x)$$

mit  $P[\Phi] = \exp\{S[\Phi, h]\}$ .

2. Die Messung von  $M[\Phi]$  auf einer Konfiguration.

5 Punkte

## Aufgabe 7.2: *Monte-Carlo-Simulation: Pi*

Es soll  $\pi$  mit Hilfe von Monte-Carlo-Simulation bestimmt werden. Dazu nutzen wir die Beziehung zwischen  $\pi$  und dem Flächeninhalt eines Kreises

$$S_K = \pi r^2$$

Der Flächeninhalt des einschliessenden Quadrat mit Kantenlänge  $2r$  ist  $S_Q = 4r^2$  und wir erhalten

$$\pi = 4S_K/S_Q$$

Das Verhältnis kann bestimmt werden, indem man zufällige Punkte im Quadrat erzeugt (gleichverteilt, d.h. jeder Punkt ist gleich wahrscheinlich) und die Punkte im Kreis zählt. Das Verhältnis  $S_K/S_Q$  wird approximiert durch die Anzahl der Paare im Kreis geteilt durch die Gesamtzahl der Punkte.

In C können auf der CPU gleichverteilte Zahlen im Intervall  $[0, 1]$  erzeugt werden mit

```
float x = (float)rand() / (float)RAND_MAX;
```

Auf der GPU kann man die CURAND-Bibliothek benutzen:

```
#include <curand_kernel.h>

__global__ some_kernel(float *x, curandState *states)
{
    unsigned int tid = threadIdx.x + blockDim.x*blockIdx.x;

    curand_init(tid, 0, 0, &states[tid]);

    x[tid] = curand_uniform_double(&states[tid]);
}
```

In [zufallszahlen.cu](#) finden Sie ein Beispiel zur Benutzung der Bibliothek.

1. Bestimmen Sie  $\pi$  so genau wie möglich mit Zufallszahlen einfacher Genauigkeit (single precision).
2. Stellen Sie den Zusammenhang zwischen der oben beschriebenen Methode und Monte-Carlo-Simulation her. Welches Integral wird numerisch berechnet?

*10 Punkte*