

# Ćwiczenia VIII

## Model Isinga – ściśle i z Monte Carlo

Jakub Tworzydło

Katedra Teorii Materii Skondensowanej  
Instytut Fizyki Teoretycznej  
telefon: (022)5532-314, pokój 344

21/11/2012 Hoża, Warszawa

# Plan

1 Termiczne, równowagowe Monte Carlo

2 Ścisłe sumowanie

# Plan

- 1 Termiczne, równowagowe Monte Carlo
- 2 Ścisłe sumowanie

# Zadanie 1

Wykonać symulację Monte Carlo z próbnym ruchem i współczynnikiem akceptacji wg. metody Metropolis'a (z wykładu)

- układ spinów Isinga na sieci kwadratowej  $L \times L$ , z periodycznymi warunkami brzegowymi
- stała energii  $J = 1$ , bezwymiarowa temperatura  $T = 1/\beta = \{1, 2, 3, 4, 5\}$
- wykonać 1000 kroków MC termalizacji oraz 5000 kroków MC dla pomiarów, oddzielnie w każdej temperaturze
- porównać średni moduł magnetyzacji z wynikiem ścisłym dla siatki  $5 \times 5$

temp	1.	2.	3.	4.	5.
mag	0.998	0.912	0.512	0.334	0.275

# Zadanie 1

```
s = np.ones((Lbig,Lbig), dtype=np.int8) # siec spinow

for p in xrange(N):    # petla po probach przewrocenia
                        # pojedynczego spinu
    k    # wybierz przypadkowo wezel

    # oblicz sume po sasiadach
    sum += ...
    # oblicz wklad do zmiany energii
    delta = s[k]*sum

    # odwroc spin dla delta<=0
    s[i]=-s[i]
    # odwroc spin dla delta>0 z prawdop. prob[delta]
    s[i]=-s[i]

# zbieraj wyniki na magnetyzacje
# mcs -- Monte Carlo step -- licznik
magnet[mcs]= np.sum(s*1.0/Lbig/Lbig)
```

## Zadanie 2

Uzupełnić i uruchomić program obliczający średni moduł magnetyzacji przy pomocy ścisłego sumowania. Wygenerować kilka wyników dla siatki  $4 \times 4$  i porównać z rozwiązaniem symulacji Monte Carlo.

## Zadanie 2

```
for k in xrange(2**N):
    # przygotuj uklad s
    l = list(bin(k))[2:]
    s = np.array( ['0' for x in xrange(N-len(l))]\
                  + l, dtype=np.int8 )
    s = (s*2-1).reshape(Lbig,Lbig)
    # sumuj spiny sasiadow
    a = np.roll(s,1,axis=0)
    a += np.roll(s,-1,axis=0)
    a += np.roll(s,1,axis=1)
    a += np.roll(s,-1,axis=1)
    # oblicz czynnik Boltzmana
    boltz[k] = np.exp(beta*np.sum(s*a)/2.)
    # oblicz magnetyzacje
    mag[k] = np.sum(s*1.0/N)
```