

Számítógépes szimulációk gyakorlat – 2 dimenziós Ising-modell

Balogh László

2011. szeptember 27.

(eredeti dokumentum: 2011. január 21.)

1. A program leírása

2-dimenziós Ising-modell ($J = 1$ egységben):

$$H = - \sum_{\langle \alpha, \beta \rangle} \sigma_{\alpha} \sigma_{\beta} - B \sum_{\alpha} \sigma_{\alpha}, \quad (1)$$

ahol B a külső (homogén) mágneses tér J egységeiben, $\sigma_{\alpha} \in \{+1; -1\}$ és az α index egy $L \times L$ -es négyzetrács rácshelyeit indexeli; csak a szomszédos spinek között van kölcsönhatás. Periodikus határfeltételt alkalmaztam.

A σ_{α} spinváltozók helyett a programban $S[i][j]$ -vel jelölt, 0 vagy 1 értékű változókat használtam.¹

Az elemi dinamikai lépés: egy spint a Metropolis-algoritmus szerinti valószínűséggel:

$$P \{ \alpha\text{-dik spin átfordítása} \} = \begin{cases} 1, & \text{ha } \Delta E \leq 0 \\ e^{-\Delta E/T}, & \text{ha } \Delta E > 0 \end{cases} \quad (2)$$

fordítok át ellentétes irányba. Egy *lépésnek (step)* hívom a forráskódban, ha egymás után sorban az összes spinre végrehajtom a fenti elemi lépést.

Egy spin energiaváltozása csak 5-féle lehet (pl. külső tér nélkül, a szomszédainak az állásától függően: $+8J$; $+4J$; 0 ; $-4J$; vagy $-8J$). Ezért a (2) egyenlet szerinti valószínűségeket előre kiszámoltam, és egy `PROB` nevű 2×5 -ös tömbben tároltam. Válasszuk ki az α -dik spint! Ekkor ennek az átfordulási valószínűsését a

$$\text{PROB} \left[S_{\alpha} \right] \left[\sum_{\beta: \langle \alpha, \beta \rangle} S_{\beta} \right] \quad (3)$$

tömbem tartalmazza. Vagyis a tömbemlek:

```
71 // pre-calculate the acceptance ratios for the possible spin-configurations:
72 PROB [0] [0] = exp(-8.0/T) * exp(2.0*B/T);
73 PROB [0] [1] = exp(-4.0/T) * exp(2.0*B/T);
74 PROB [0] [2] = 1.0 * exp(2.0*B/T);
75 PROB [0] [3] = exp(4.0/T) * exp(2.0*B/T);
76 PROB [0] [4] = exp(8.0/T) * exp(2.0*B/T);
77 PROB [1] [0] = exp(8.0/T) * exp(-2.0*B/T);
78 PROB [1] [1] = exp(4.0/T) * exp(-2.0*B/T);
```

¹A dokumentációban α és β összetett indexet jelölnek, a programban egy spint két indexszel, a rácsbeli pozíciójával indexelek. Pl.: $\alpha \equiv (i, j)$.

```

79 PROB[1][2] = 1.0 * exp(-2.0*B/T);
80 PROB[1][3] = exp(-4.0/T) * exp(-2.0*B/T);
81 PROB[1][4] = exp(-8.0/T) * exp(-2.0*B/T);

```

Az adott valószínűséggel történő átfordítást egy $[0;1)$ tartományba eső véletlen szám segítségével valósíthatjuk meg:

```

94 // sum the neighbouring spins of (i,j)
95 // periodic boundary condition is implemented here:
96 sumj = 0;
97 sumj += S[(i+L-1)%L][j] + S[(i+1)%L][j] + S[i][(j+L-1)%L] + S[i][(j+1)%L];
98
99 // change the (i,j) spin with the Metropolis-probability (pre-calculated):
100 if (gsl_rng_uniform(rng) < PROB[S[i][j]][sumj]) {
101     S[i][j] = 1 - S[i][j];
102 }

```

A véletlenszám-generátort a GSL-ből² vettem; és a <http://www.random.org> oldalról letöltött számok alapján indítom el, és melegítem be.

Az eredményeket a standard outputra írom ki, 3 oszlopban: hőmérséklet, mágnesezettség, szuszceptibilitás. A futtatás parancssora, ha `ising2d.x` a futtatható fájl neve és az adatokat a `proba.dat` fájlba szeretnénk menteni:

```
./ising2d.x > proba.dat
```

Egy próba futtatás kimenetének első néhány sora:

```

1 # system size:          50 x 50
2 # external field:      0.00000E+00
3 # temperature range:   2.380000000000 --> 2.340000000000
4 # temperature resolution: 1.00000E-03
5 # simulation param.: therm_steps = 100000; every = 100; avrg_steps = 10000
6 # total number of MC scans: 4.510E+08
7 # estimated CPU time (@bell): 1.97 h
8     2.380000000000      0.2348198160      2.595243E+01
9     2.379000000000      0.2371544960      2.606002E+01
10    2.378000000000      0.2384437840      2.668208E+01
11    2.377000000000      0.2403933600      2.680609E+01
12    2.376000000000      0.2424826240      2.712841E+01

```

A szimulációk eredményeit Gnuplottal dolgoztam fel. (A Gnuplot figyelmen kívül hagyja a #-tel kezdődő sorokat.)

2. A program használata

A paramétereket a forráskódban kell beállítani:

```

20 // simulation control parameters:
21 const int    L = 50;           // system size: L x L
22 const double B = 0.0;         // external magnetic field
23 const double T0 = 2.34, T1 = 2.38;
24 // temperature range: [T0;T1] NOTE: T0 < T1 is required
25 const double dT = 0.001;
26 // temperature range resolution
27 const int therm_steps = 100000;
28 // number of steps, which are ignored (to thermalize the system)
29 const int every = 100;
30 // every ... steps are taken into account in the thermodynamic average

```

²GSL = GNU Scientific Library, honlap: <http://www.gnu.org/software/gsl/>

```
const int avrg_steps = 10000;
// number of samples to the thermodynamic average
```

A program minden hőmérséklet pontban eldobja az első `therm_steps` lépést, ezzel biztosítva, hogy a rendszer felveszi a kért hőmérséklet értéket. Ezután minden `every`-edik lépésben kiszámítja a teljes mágnesezettség abszolút értékét és négyzetét; és összegzi a `sigma` és a `sigma2` változóiban; összesen `avrg_steps`-szer. A mágnesezettség és a szuszceptibilitás, ha n jelöli a termodinamikai mintában lévő adatok számát:

$$m = \frac{1}{L^2} \cdot \frac{\text{sigma}}{n} \quad \chi = \frac{1}{L^2} \cdot \frac{1}{T} \cdot \left(\frac{\text{sigma2}}{n} - \frac{(\text{sigma})^2}{n^2} \right) \quad (4)$$

3. A szimulációs paraméterek meghatározása

A szimulációból akkor kaphatunk jó eredményt, ha minél nagyobb rendszerre futtatunk, minél több átlagolási lépést tudunk elvégezni, úgy hogy ezek függetlenek legyenek, illetve egy adott hőmérséklet pontban várjuk meg a termalizációt. Ezek a kritériumok versenyeznek a futásidővel.

50×50 ; 100×100 ; 200×200 ; 300×300 és 400×400 méretű rendszereket választottam.

Meghatároztam a τ korrelációs időt a következők szerint. $m(t)$ jelöli az egy rácshelyre normált mágnesezettséget a t -edik lépésben. Definiáljuk a $\Phi_{mm}(t)$ időbeli korrelációs függvényt az alábbiak szerint:

$$\Phi_{mm}(t) = \frac{\langle m(t')m(t'+t) \rangle - \mu^2}{\sigma^2}, \quad (5)$$

ahol μ a mágnesezettség várható értéke, σ^2 pedig a szórásnégyzete; és $\langle \dots \rangle$ t' szerinti időátlagot jelent. Ekkor:

$$\tau = \int_0^{\infty} \Phi_{mm}(t) dt. \quad (6)$$

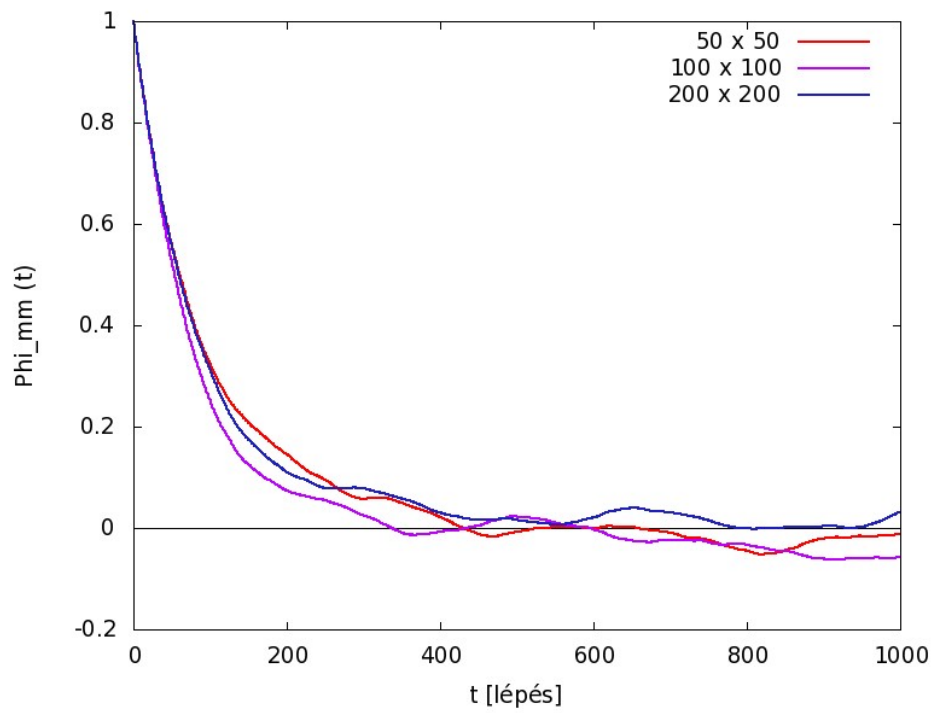
A $\Phi_{mm}(t)$ korrelációs függvényt 50×50 , 100×100 és 200×200 méretű rendszerekre határoztam meg, $T = 2,4$ hőmérsékletnél: 1. ábra. Ebből (6) alapján $\tau \approx 100$ adódott mindhárom rendszerre. Ezért a szimulációkban `every = 100` értéket állítottam be. Magasabb hőmérsékleten valószínűleg ennél kevesebb is elég lenne, hogy a mintavételek függetlennek tekinthetők legyenek, a kritikus hőmérsékletet megközelítve viszont sokkal több kellene...

A termalizációra is kb. megfelelne τ , ezt viszont kb. a teljes szimulációs lépések 10%-ának választottam.

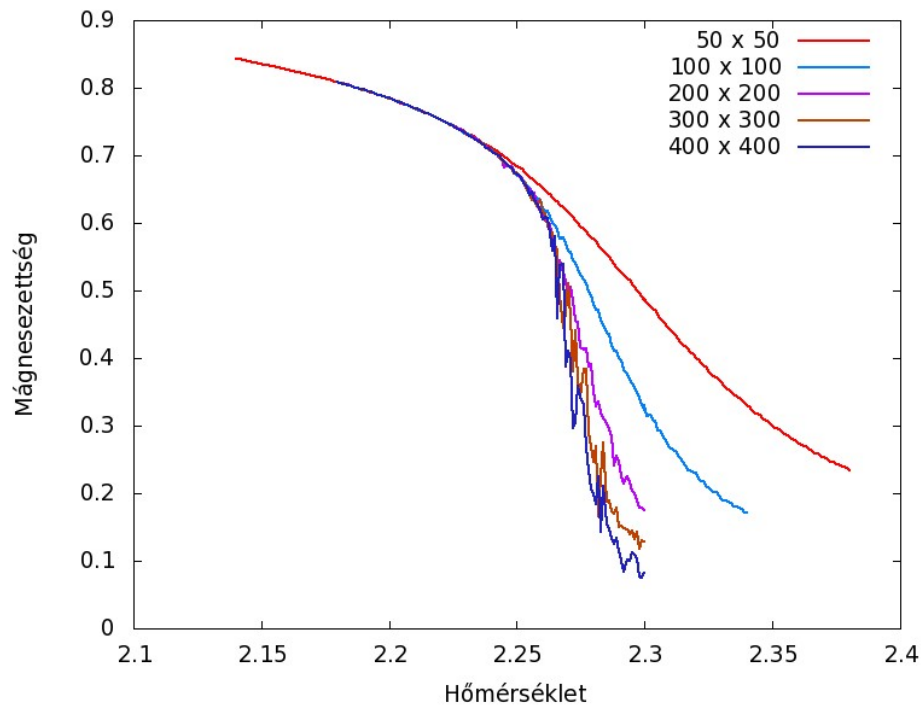
4. Eredmények

4.1. Mágnesezettség és szuszceptibilitás

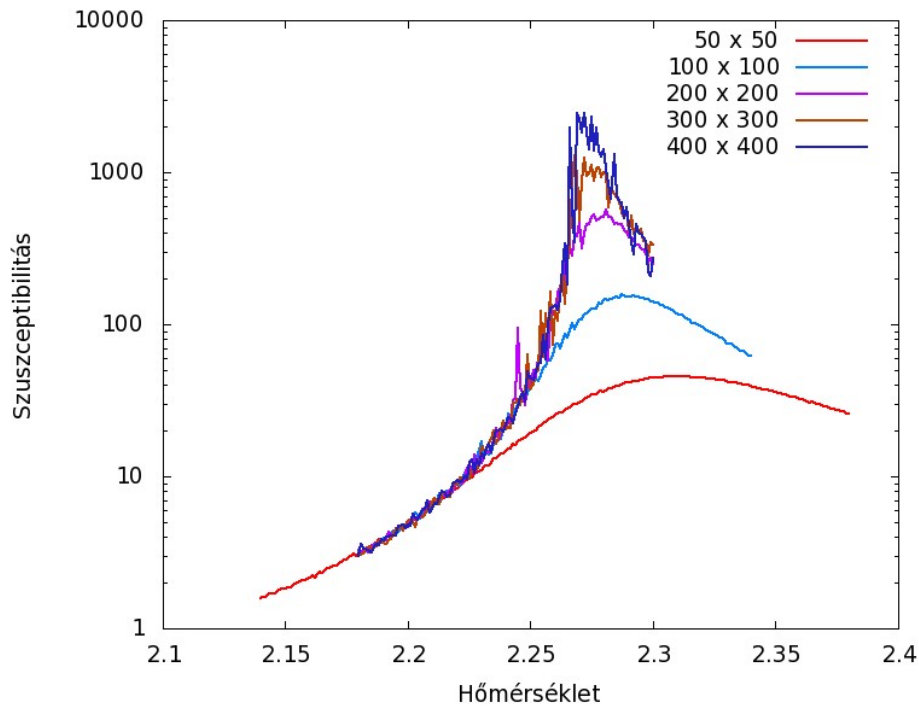
A szimulációs program kimenete a mágnesezettség és szuszceptibilitás, mint a hőmérséklet függvénye. Lásd: 2. és 3. ábra.



1. ábra. $\Phi_{mm}(t)$ időbeli korrelációs függvény $T = 2,4$ hőmérsékleten, három különböző rendszerméret esetén



2. ábra. Mágneszettség a hőmérséklet függvényében.



3. ábra. Szuszceptibilitás a hőmérséklet függvényében. (Log–lin skála!)

Függelék

A. Követelmények a fordításhoz, futtatáshoz

A.1. Operációs rendszer

Én Linux (Fedora 13) alatt dolgoztam, a konstanzi klaszteren³ SuSE 11.2 rendszer volt. Ha a további követelmények teljesülnek, Windows alatt is működni kellene – de nem próbáltam ki.

A.2. Szoftverek

Telepíteni kell a GSL-t. (Fedora esetén `gsl` a csomag neve, lehet, hogy a program lefordításához a `gsl-devel` csomag is szükséges. Nem kísérleteztem vele, mindkettő telepítve volt nálam. Debian esetén a szükséges csomag: `libgsl0-dev`.)

A véletlenszám-generátor indításához a kezdőértéket és a bemelegítés lépésszámát az internetről (a <http://www.random.org/> weboldalról) tölti le a program. Ehhez a `wget` nevű külső programot hívja meg, ennek is telepítve kell lenni. (Fedora 13-ban u.e. a csomag neve is.) Ha nincs telepítve, hibajelzést kapunk, és a rendszeridő lesz a generátor kezdőértéke; az első 100 000 lekérdezést dobja el bemelegítésként.

A.3. Forrásfájlok

- `downloadseed.h` `downloadseed.c`: a véletlenszám-generátor indítása és bemelegítése.
- `ising2d.c`: a feladat megoldása.

³A szimulációk kb. 12-12 óráig futottak (egy-egy processzoron) a Konstanzi Egyetem *Magnetic Materials: Theory and Computer Simulation* csoportjának a számítógépein. (<http://theorie.physik.uni-konstanz.de/uli/>)

A.4. Fordítás parancssora

```
gcc ising2d.c downloadseed.c -lm -lgsl -lgslcblas -o ising2d.x
```

A.5. A futtatott szimulációk paramétere

L	Hőmérséklet-tartomány	therm_steps	avrg_steps
50	2,14 – 2,38	10^6	10^5
100	2,18 – 2,34	250 000	25 000
200	2,18 – 2,30	60 000	6 000
300	2,18 – 2,30	20 000	2 000
400	2,18 – 2,30	15 000	1 500
<i>500</i>	<i>2,18 – 2,30</i>	<i>10 000</i>	<i>1 000</i>
<i>1000</i>	<i>2,18 – 2,30</i>	<i>2 500</i>	<i>250</i>

1. táblázat. A futtatások paramétere. Minden futtatásnál megegyeztek a következő paraméterek: külső tér: $B = 0.0$; a hőmérséklet-tartomány felbontása: $dT = 0.001$; az átlagolásba minden `every = 100`-adik MC lépés számított bele. A táblázatban *dőlttel* szedett sorok nem adtak megfelelően sima görbét, így ezeket nem használtam fel. A paramétereket úgy állítottam be, hogy kb. 12 óra legyen mindegyik szimuláció futásideje. Viszont a kevesebb átlagolásszámot tartalmazó futások zajosabb görbét eredményeztek, ott célszerűbb lett volna több átlagolást alkalmazni.

A.6. Megjegyzés a párhuzamosításhoz

A szimuláció tulajdonképpen jól párhuzamosítható; ha a hőmérséklet-tartományt részekre daraboljuk, a részek elindíthatók egymással párhuzamosan. Szélsőséges esetben akár hőmérsékletpontonként is számolhatunk külön-külön szimulációkkal, ekkor viszont `therm_steps` értékének növelése lehet szükséges – különösen a kritikus hőmérséklet közelében.