

# Ising-modell és a Metropolis-Hastings algoritmus

Tekintsük az előadáson tanulmányozott 1D Ising láncot!

A programfejlesztés során kezdetben egy kis, mégpedig 64 spinből álló rendszerrel dolgozzunk! Feltételezve, hogy  $J > 0$ , a rendszer energiáját és a  $k_B T$  termikus energiát (hőmérsékletet)  $J$  egységekben mérjük, ami azt jelenti, hogy a  $J$  helyébe írhatunk 1-et.

**Az egyes feladatrészeket egymás után valósítsa meg a beadott program!**

**A következők legyenek a parancssori argumentumok:**

**lánc\_hossza lépések\_száma  $k_B T$ \_értéke**

## a) Metropolis-Hastings algoritmus

Programozzuk le a Metropolis-Hastings algoritmust, amelyet az előadáson tárgyaltunk! Tegyük ezt az algoritmust külön függvénybe. Ezután futtassuk 10000 lépésen keresztül! Használjunk egy 64 spinből álló Ising láncot nyitott határfeltétellel! Indítsuk a rendszert az előadáson definiált "hideg" kezdőállapotból! Készítsük el az algoritmus által szolgáltatott  $E_i$  energiaértékek gyakorisági hisztogramját  $k_B T = 1$  és  $k_B T = 3$  esetekre! Ehhez írassuk ki egy fájlba a rendszer energiáját minden Metropolis-Hasting lépés után új sorba! Ez persze úgy értendő, hogy ha elfogadtuk az új energiaértéket, akkor azt írjuk ki, ha nem fogadtuk el, akkor az előző lépésből megtartott értéket írjuk ki. Abrázoljuk egy results.ipynb notebookban az értékeket hisztogramon az alábbi módon: (A)

In [ ]:

```
%pylab inline
#-----
x1=loadtxt("file1.dat")
x2=loadtxt("file2.dat")
figsize(8,6)
xlim(-64,0)
hist(x1,32)
hist(x2,32);
```

## b) Kezdeti állapottól való függés

Ismételjük meg az a) pontbeli számolást a "forró" kezdőállapotból indulva! Az ábrázolás alá írjuk le, hogy ugyanazt az eloszlást kaptuk-e! (A)

## c) "Időfejlődés"

Ábrázoljuk a) részben definiált rendszer "időfejlődését" 256 lépésen keresztül! Ezt a következőképp tegyük:

Minden egyes iterációs lépésben irassuk ki a lánc minden egyes spinjének az értékét egy fájlba! Egy lépéshez tartozó spin értékek egy sorban szerepeljenek, space-vel elválasztva! Ezután ábrázoljuk az eredményeket a results.ipynb-ben! A vízszintes tengely az iterációs lépés számának, a függőleges tengely pedig a spin helyét jelző indexnek feleljen meg! Ezt pl. az alábbi utasításokkal lehet megvalósítani. Próbáljuk ezt ki két különböző hőmérsékletnél, pl.  $k_B T = 1$  és  $k_B T = 3$ . (A)

```
In [ ]: smx=loadtxt("chain-evolve.dat")
figsize(16,4)
matshow(smx.T);
```

## d) A teljes energia hőmérsékletfüggése

Határozzuk meg a rendszer teljes energiájának hőmérsékletfüggését! Ehhez először ábrázoljuk az a) részben kapott energiaértékeket a lépésszám függvényében! Láthatjuk, hogy néhány száz lépés után az energia értéke egy átlag körül fluktuál. Ezt nevezzük termalizációnak.

A program számolja ki az átlagenergiát a termalizáció beállta után (a sok egymás után energiaérték kiírása nélkül) pl. 2000 iterációra! Ezt ismételjük meg pl. 20 különböző hőmérséklet értékre a  $k_B T = 0.1, \dots 4.0$  intervallumban! Ábrázoljuk a teljes energiát a  $k_B T$  függvényében a results.ipynb-ben! Hasonlítsuk össze az analitikus számolásból kapható eredménnyel, ami  $\langle E \rangle = -(N - 1)J \tanh\left(\frac{J}{k_B T}\right)$  ! (T)

## e) Termalizáció időfüggés

A d) pontban láttuk, hogy a rendszer energiája egy idő után egy átlag körül fluktuál. Hogyan áll be ez az átlagos viselkedés, azt is láthatóvá tehetjük úgy, hogy sokszor indítjuk el a rendszert a kiválasztott kezdeti állapotból, és az azonos lépésszámhoz tartozó energiaértékeket összeátlagoljuk. Majd a kapott átlagértékeket a lépésszám függvényében vizsgáljuk. Ezt úgy is felfoghatjuk, mintha a rendszerből sok azonos példányt csináltunk volna, amiket egyszerre léptetnénk időben, és minden időpillanatban megnézzük az átlagos energiát. Technikai dolog csupán, hogy ezt egymás utáni futtatásokkal valósítjuk meg. Ábrázoljuk az átlagos energiát  $k_B T = 1$  hőmérsékleten a hideg kezdeti állapotból indulva a lépésszám függvényében! Majd a hosszú idejű átlagtól való eltérés logaritmusát is ábrázoljuk, és próbáljuk eldönteni, hogy exponenciális-e a lecsengés!

## f) Fázisátalakulás kétdimenziós Ising-modellben

Láttuk, hogy alacsony hőmérsékleten nagyobb domének alakulnak ki, mint magasabb hőmérsékleteken, és elég alacsony hőmérsékleten már rendeződik a rendszer, csak ritkán jelenik meg benne ellentétes spineket tartalmazó tartomány. Hasonló a helyzet kétdimenzióban is, csak az alacsonyabb és magasabb hőmérsékletek közötti átmenet ugrásszerűvé válik, azaz fázisátalakulás jelenik meg. Az ugrás helyét hívjuk kritikus hőmérsékletnek. Ez persze szigorúan véve végtelen rendszerben igaz, véges rendszer esetén folytonos az átmenet, de a méret növelésével egyre élesebbé válik.

Módosítsuk úgy a programot, hogy azzal vizsgálni tudjuk példaként egy négyzet alakú négyzetrács tartományban az Ising-modell viselkedését! Az energiában az első szomszéd spinek szorzatát kell összegezni. A fázisátalakulás legjobban a mágnesezettség hőmérsékletfüggésében látszik. Ez alatt azt értjük, hogy vesszük a spinek értékeinek az összegét, majd ennek abszolút értékét átlagoljuk. Bővítsük tehát a programot úgy, hogy amikor lépésenkénti kiírás van, akkor a pillanatnyi mágnesezettséget is kiírja, a futás végén pedig írja ki az átlagot! Ábrázoljuk az eredményt a hőmérséklet függvényében! Az egydimenziós esetben is elvégezve ezt, hasonlítsuk össze a két esetet!

Rajzoljuk ki a rendszer termalizált állapotát kétdimenzióban néhány kritikus pont alatti és feletti hőmérsékleten! Azt tapasztaljuk-e, amit vártunk?

Megint tanultunk valamit: ezért veszíti el a vasmágnésünk a mágnesezettségét, ha felmelegítjük. És persze, hogy hogyan kell ilyen rendszerre programot írni.