

HELSINGIN YLIOPISTO

LASKENNALLISEN FYSIIKAN OHJELMATYÖ

---

**Diffuusion Monte Carlo  
-simulointi yksiulotteisessa  
systemissä**

---

*Tehnyt:*

Joonas KYLLIÄINEN

17. marraskuuta 2016

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>3</b>
<b>2</b>	<b>Teoria</b>	<b>3</b>
<b>3</b>	<b>Teorian implementaatio</b>	<b>4</b>
3.0.1	Implementaatio . . . . .	4
3.0.2	Ohjeet . . . . .	4
<b>4</b>	<b>Tulokset</b>	<b>6</b>
<b>5</b>	<b>Yhteenveto</b>	<b>6</b>

# 1 Johdanto

Termillä Monte Carlo-simulaatio tarkoitetaan laajaa alaa erilaisia laskennallisia menetelmiä, joissa käytetään satunnaisotantaa numeerisen ratkaisun löytämiseksi. Termi on saanut nimensä Monacon Monte Carlon alueelta, joka on tunnettu kasinoistaan. Monte Carlo-menetelmät ovat erityisen hyödyllisiä fysiikassa varsinkin ongelmissa, joilla on useita vapausasteita kuten kineettinen kaasuteoria. Monte Carlo-simulaatiota käytetään usein juuri hiukkasten liikkeen mallintamiseen

Diffuusio yksiulotteisessa epäjärjestyneessä hilassa on ongelma, jota on hyvä lähteä ratkaisemaan Monte Carlo-simuloimalla. Työn tarkoituksena on demonstroida Monte Carlo-simulaation toimintaa yksinkertaisessa hilassa, joka on hyvä ponnahduslauta samankaltaisille usempien vapausasteiden ongelmille.

## 2 Teoria

Hiukkasen käyttäytymistä mallinnetaan usein Metropolis-algoritmillä, jossa hiukkasen todennäköisyys hypätä toiseen hilapaikkaan riippuu hilapaikkojen energioiden erotuksesta [1]

$$J_{i,i\pm 1} = \frac{1}{2}e^{-(E_{i\pm 1}-E_i)/k_B T}, \quad (1)$$

jossa  $J$  on hiukkasen todennäköisyys hypätä viereiseen hilapaikkaan,  $E_i$  hilapaikan  $i$  potentiaali,  $k_B$  Boltzmanin vakio ja  $T$  lämpötila.

Diffuusion mallintamisessa olemme kuitenkin kiinnostuneita nimenomaan potentiaalivallin korkeudesta emmekä hilapaikkojen energioiden erotuksesta. Jälkimmäinen sopii paremmin tasapaino-ominaisuuksien mallintamiseen. Siksi todennäköisyys  $J$  noudattaa yhtälöä [1]

$$J_{i,i-1} = J_{i,i+1} = \frac{1}{2}e^{-E_{b,i}/k_B T}, \quad (2)$$

jossa  $E_{b,i}$  on potentiaalivallin korkeus, jonka hiukkasen tarvitsee ylittää. Tämä luonnollisesti riippuu vain hiukkasen senhetkisen hilapaikan potentiaalikupan syvyydestä ja on sama molempiin suuntiin.

Hiukkasen neliöpoikkeama kasvaa epäjärjestyneessä hilassa lineaarisesti ajan funktiona [1]

$$\langle x(t)^2 \rangle \propto t \quad (3)$$

ja sen diffuusiovakio on [1]

$$D = \lim_{t \rightarrow \infty} \frac{1}{2} \frac{d \langle x(t)^2 \rangle}{dt} \quad (4)$$

kun  $0 \lll t$ . Neliöpoikkeama on helppo selvittää Monte Carlo - simulaatiolla ja sen kulmakerrointa tutkimalla saadaan siis diffuusiovakio selville. Tuloksia voidaan verrata teoreettisiin arvoihin [1]

$$D = \left\langle \frac{1}{J} \right\rangle^{-1}, \quad (5)$$

missä

$$\left\langle \frac{1}{J} \right\rangle = \frac{1}{2N} \sum_{i=1}^N \left( \frac{1}{J_{i,i-1}} + \frac{1}{J_{1,1+1}} \right) = \frac{2}{N} \sum_{i=1}^N e^{E_{b,i}/k_B T} \quad (6)$$

## 3 Teorian implementaatio

### 3.0.1 Implementaatio

Ohjelmassa käytettiin kuvan 1 mukaista hilaa niin, että  $N = 100$ . Suurilla  $N$ :n arvoilla kulmakerroin loivenee niin paljon, että hyviä tuloksia on vaikea saada. Hila luotiin valitsemalla potentiaalikuoppien erotus

$$| E_i - E_{i \pm 1} | = \epsilon \quad (7)$$

ja laskemalla

$$E_i = -\epsilon \left[ N_{1/2} - | i - N_{1/2} | \right] \quad (8)$$

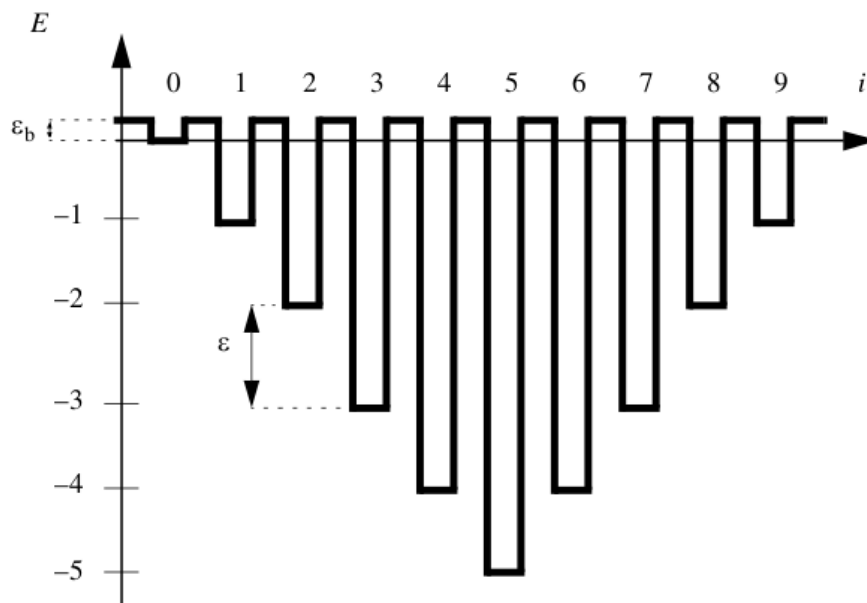
Jotta ensimmäisen hilapaikan hyppytodennäköisyys ei olisi aina 1, lisättiin hilaan pieni valli  $\epsilon_b$ . Tällöin hyppytodennäköisyyteen lisättiin termi:

$$J_{i,i-1} = J_{i,i+1} = \frac{1}{2} e^{-(\epsilon_b - E_i)/k_B T}. \quad (9)$$

### 3.0.2 Ohjeet

Ohjelma sekä myös lyhyet ohjeet löytyvät osoitteesta <https://github.com/joonasky/MCD>. Lähdekoodit löytyvät kansioista **src/** ja ajon tulokset kansioista **run/**. Ohjelmat käännetään komennolla

**make**



Kuva 1: Ohjelmassa käytetty hila [1]

Käännöksen tuloksena syntyvät ohjelmat tvals, MCD ja MCDM. Näistä jälkimmäisin käyttää Metropolis-algoritmia. Ohjelma tvals tulostaa teoreettiset arvot.

Työssä ohjelmien ajamiseen käytettiin bash-scrpitemää, joille annetaan ajo-oikeudet ja ajetaan komennoilla

```
chmod +x runner*
./runner.sh
./runner2.sh
./runner3.sh
```

Ohjelmia voi ajaa omillakin arvoilla, kun niille antaa komentoriviargumenteiksi lämpötilan sekä tiedostonnimen neliöpoikkeamalle ja paikan todennäköisyydelle:

```
./MCD 100 ../run/t100xt2.txt ../run/t100px.txt
```

Tulokset plotattiin python-scripteillä, jotka voidaan ajaa komennoilla

```
python plotdist.py
python plotprob.py
python plotpx.py
```

**plotdist.py** piirtää neliöpoikkeamat, **plotprob.py** luonnollisen logaritmin todennäköisyydestä löytää hiukkanen paikasta  $x$   $E_i/T$ :n funktiona ja **plotpx.py** todennäköisyyden  $x$ :n funktiona.

T	Teoria	Tulos
2	0.0000000001	0.000260
5	0.0002135648	0.002590
10	0.0164629090	0.016590
20	0.1101572633	0.110559
55	0.3050836278	0.303396
100	0.3842595802	0.380840
300	0.4590479020	0.461750
300	0.4808021068	0.482007
300	0.4874634000	0.490300

Taulukko 1: Tulokset verrattuna teoreettisiin arvoihin

## 4 Tulokset

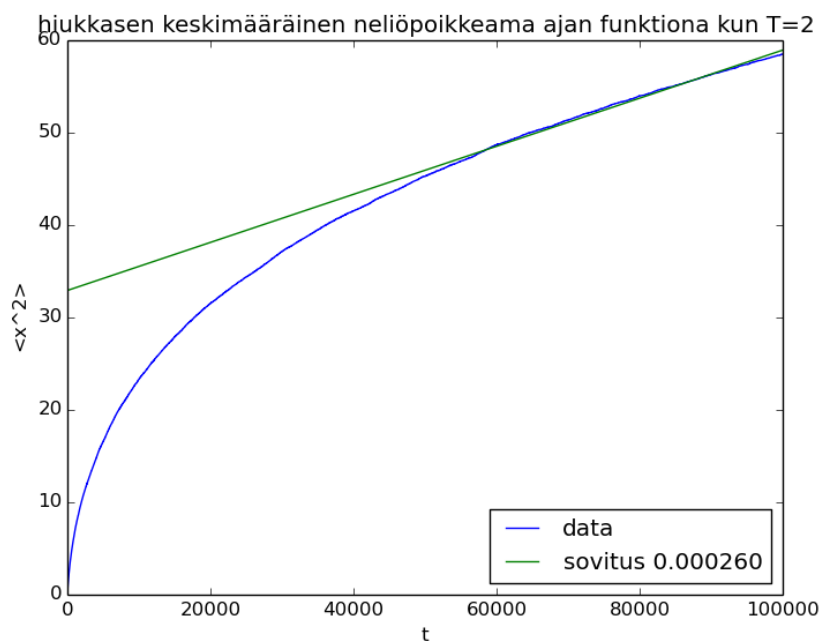
Tulokset noudattavat hyvin teoriaa. Kuvaajiin 2, 3, 4, 5, 6, 7, 8, 9 ja 10 on piirretty neliöpoikkema ajan funktiona. Taulukosta 1 löytyvät sekä teoreettiset arvot, että simulaatiolla saadut tulokset.

Käytetty MC-algoritmi täyttää myös detaljibalanssin, vaikka tulokset sitä tutkiessa paranevat hilapaikkojen lukumäärä kasvaessa ja lämpötilan ollessa tarpeeksi matala. Kuvassa 11 huomataan käyrän olevan kaukana suorasta. Kuvan simulaatiossa  $N=100$ . Kuvasta 12 kuitenkin näemme, että todennäköisyys paikan funktiona näyttää täysin oikealta, kun muistetaan hilan muoto. On epätodennäköisempää löytää hiukkanen hilapaikasta, jossa on korkeat vallit. Suuremmalla hilan koolla käyrä on jo paljon suurempi, tosin myös kohinainen kuten kuvasta 13 näkyy.

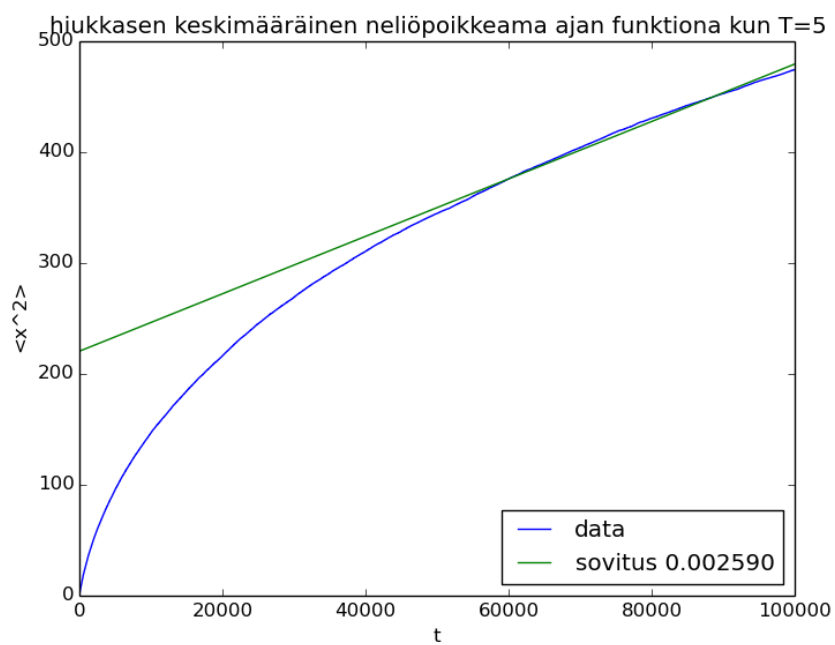
Simulaatio suoritettiin myös metropolis-algoritmilla. Kuvasta 14 näkyy, että tulos ei ole sitä mitä pitää.

## 5 Yhteenveto

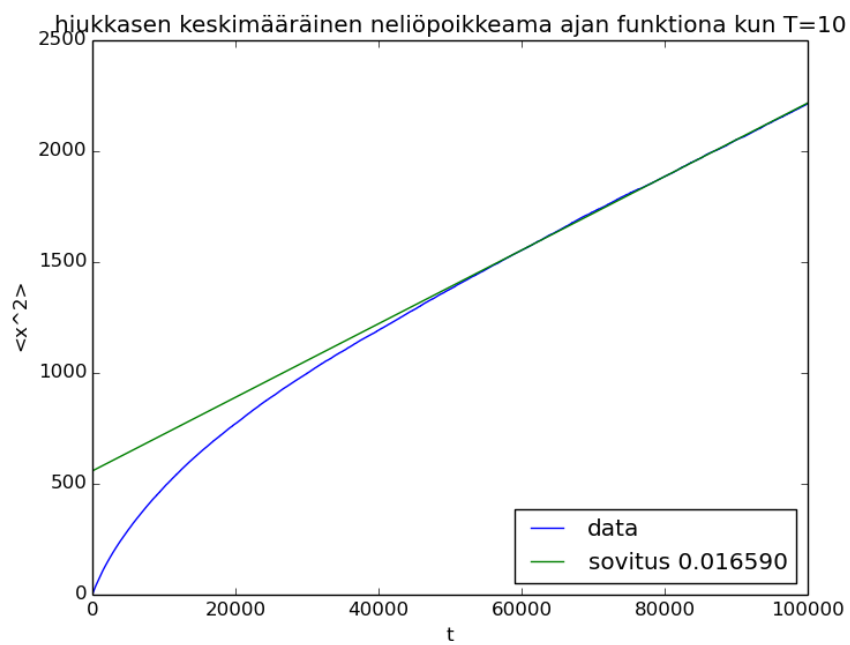
Hiukkasen diffuusiota voidaan mallintaa Monte Carlo-menetelmällä ja diffuusiokerroin voidaan tätä kautta määrittää ainakin yksiulotteisessa satunnaisloukkumallissa.



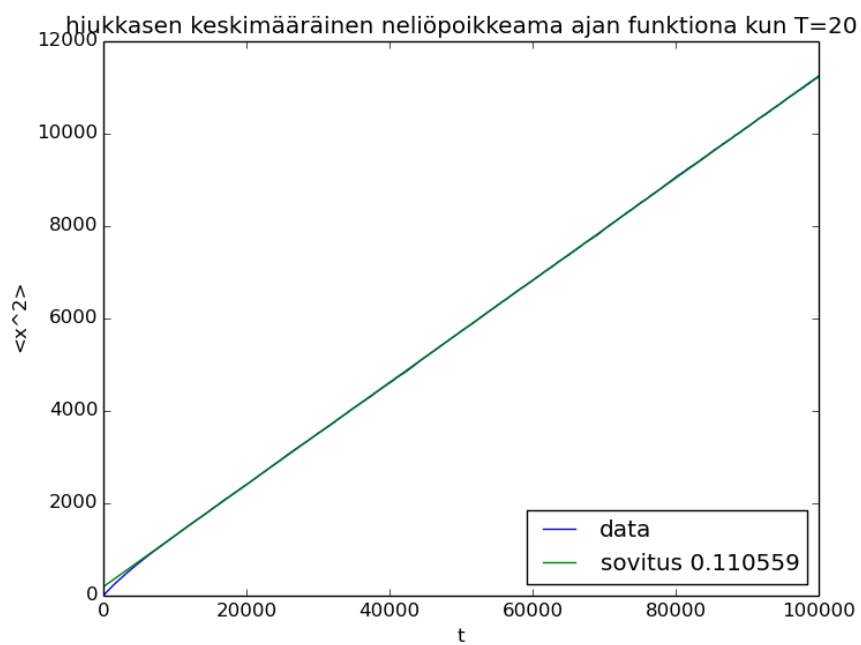
Kuva 2: Neliöpoikkeama kun  $T=2$



Kuva 3: Neliöpoikkeama kun  $T=5$

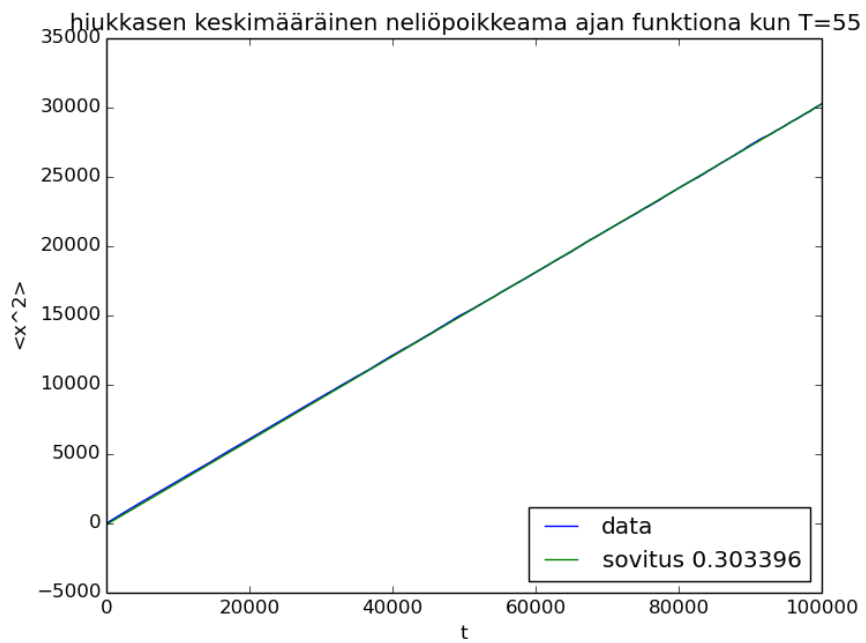


Kuva 4: Neliöpoikkeama kun  $T=10$

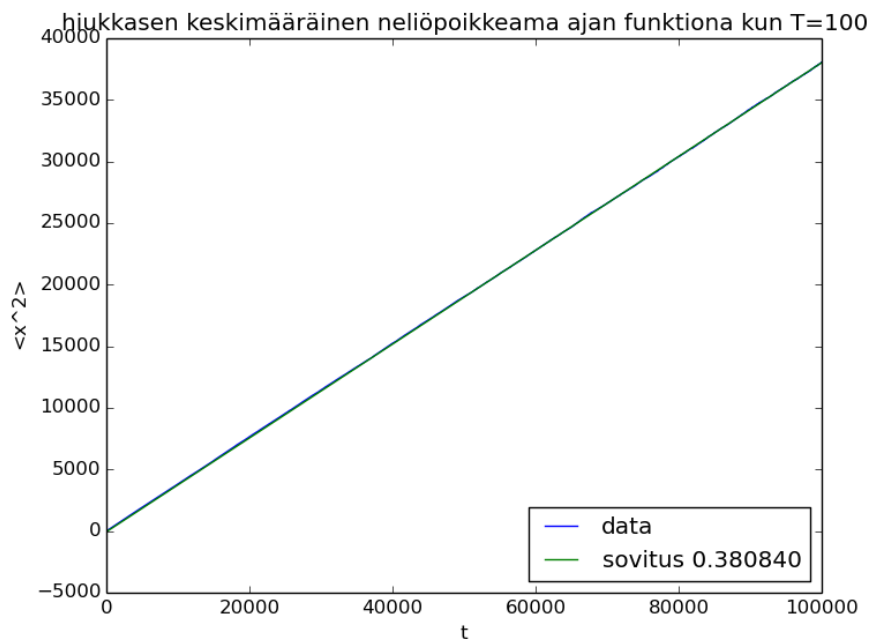


Kuva 5: Neliöpoikkeama kun  $T=20$

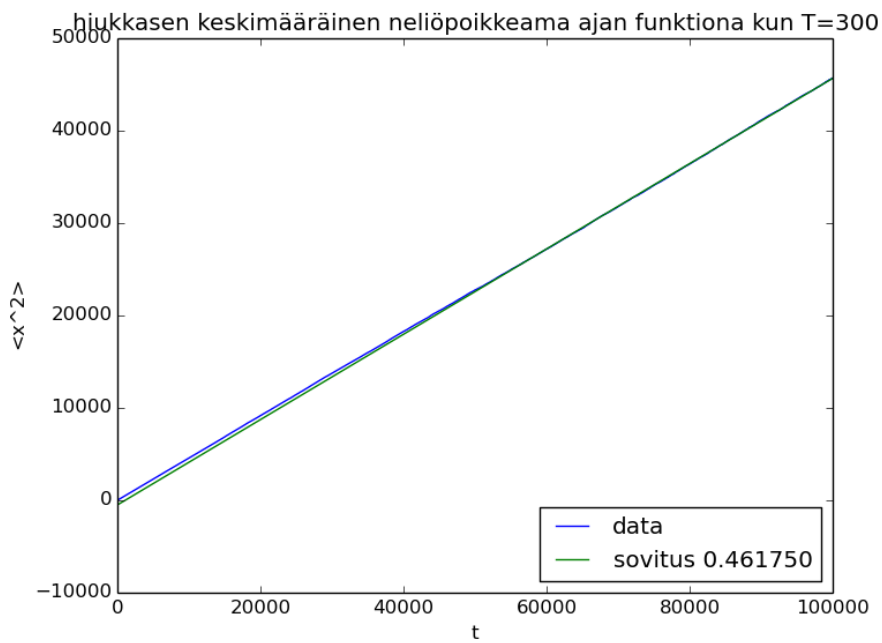




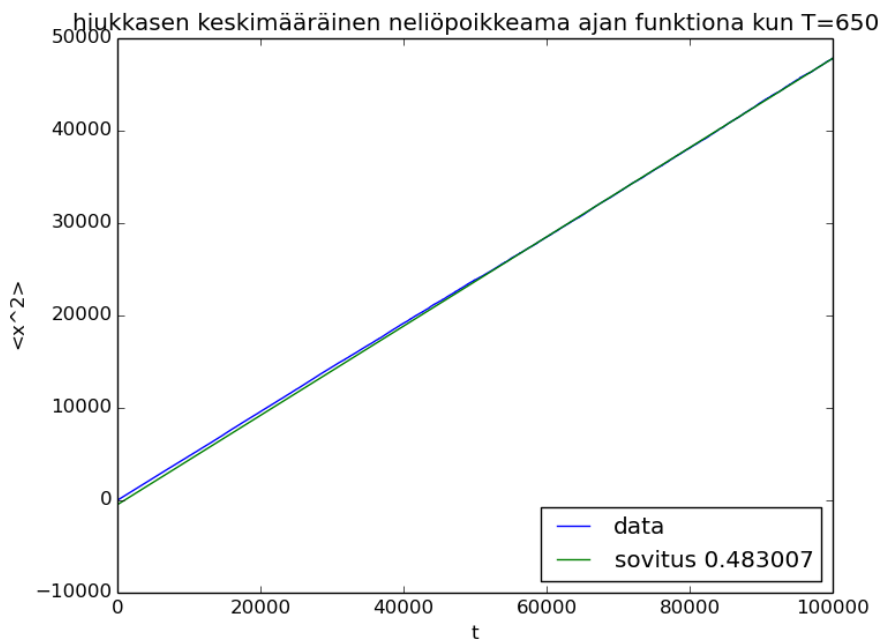
Kuva 6: Neliöpoikkeama kun  $T=55$



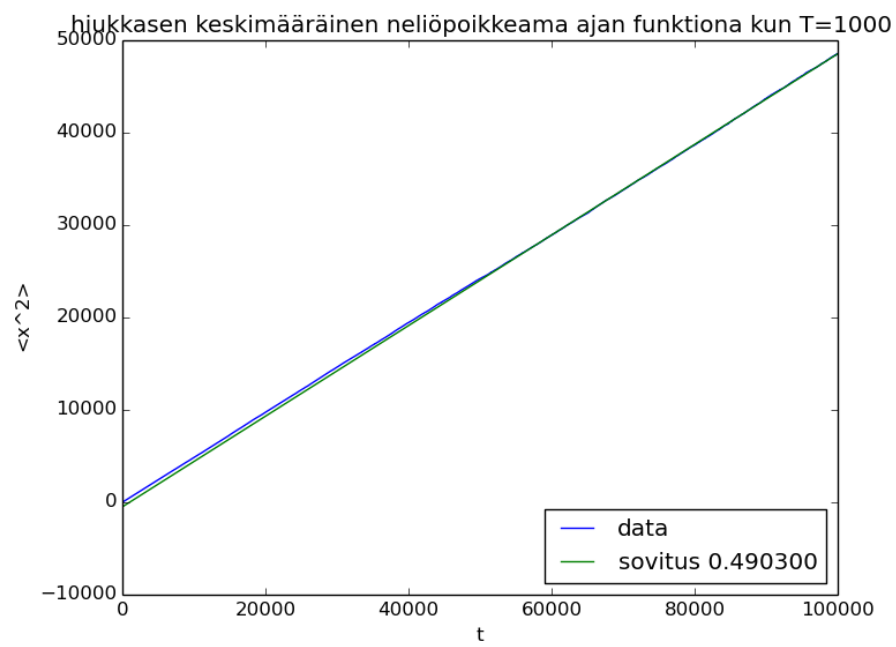
Kuva 7: Neliöpoikkeama kun  $T=100$



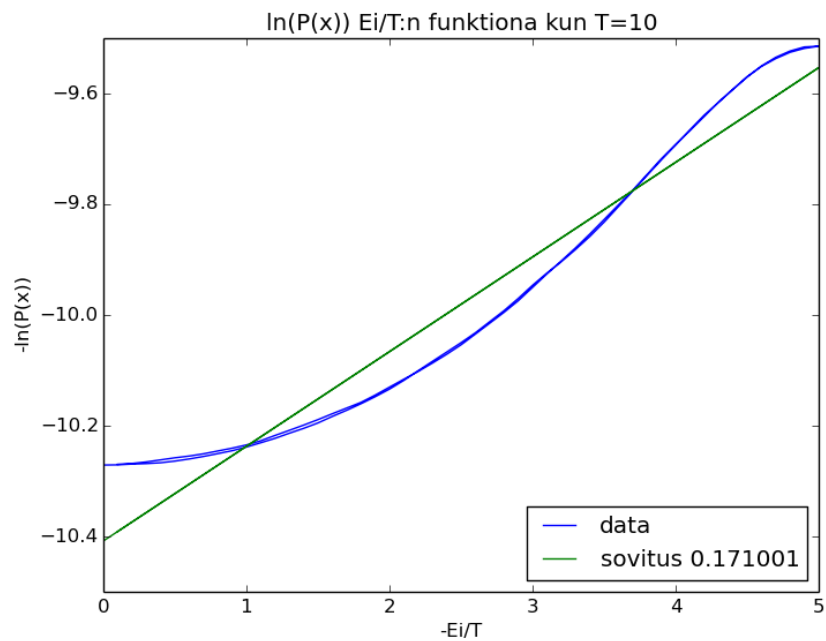
Kuva 8: Neliöpoikkeama kun  $T=300$



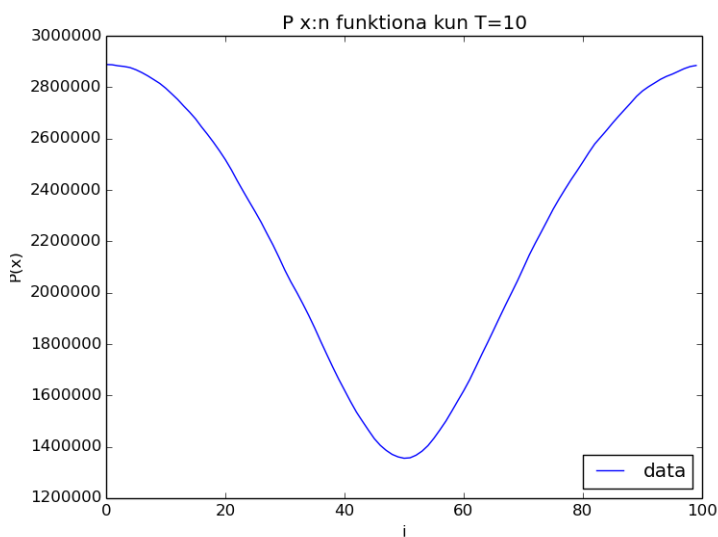
Kuva 9: Neliöpoikkeama kun  $T=650$



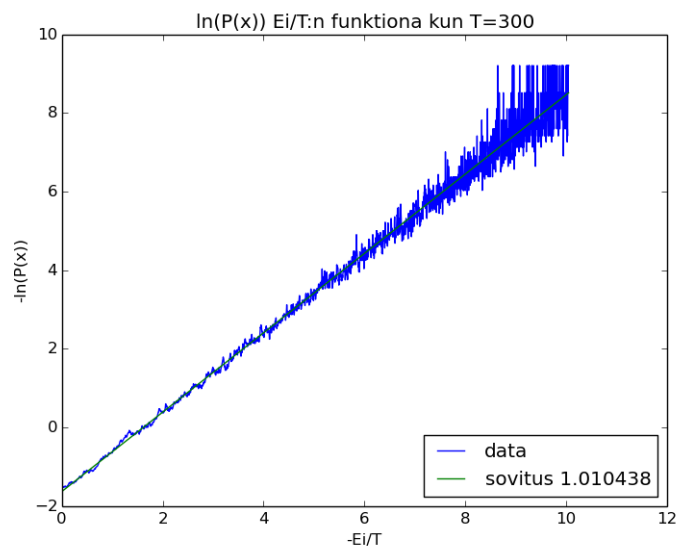
Kuva 10: Neliöpoikkeama kun  $T=1000$



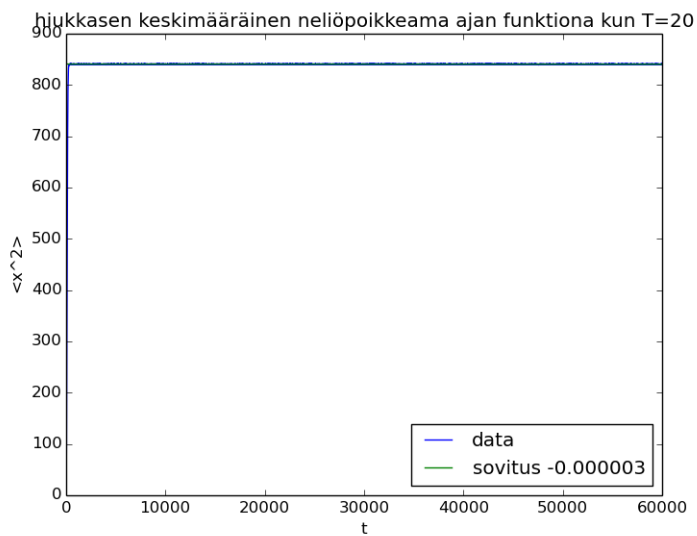
Kuva 11: paikan todennäköisyyden luonnollinen logaritmi energian ja lämpötilan osamäärän funktiona kun  $t=10$



Kuva 12: paikan todennäköisyys paikan funktiona kun  $t=10$



Kuva 13: paikan todennäköisyyden luonnollinen logaritmi energian ja lämpötilan osamäärän funktiona kun  $t=300$



Kuva 14: Metropolis-algoritmin epätoimivuus

## Viitteet

- [1] Antti Kuronen. *LASKENNALLISEN FYSIIKAN OHJELMA-  
TYÖ:Diffuusion Monte Carlo -simulointi yksiulotteisessa systeemissä.*  
Helsingin Yliopisto.