Solutions - Exercise sheet 0

Statistical mechanics and Monte Carlo algorithm basics

PUE Advanced Computational Physics
University of Vienna - Faculty of Physics

A particle in a homogeneous gravitational field

Consider a particle in a homogeneous, one-dimensional gravitational field. Its potential energy is given by

$$U(h) = mgh \tag{1}$$

where m is the mass, g is the gravitational acceleration and $h > 0$ is the height of the particle. The particle is coupled to a heat bath at temperature T.

1 Pen and paper analysis

a. What is the Hamiltionian, $\mathscr{H}(h,v)$ of the system? (v is the velocity of the particle)

$$\mathscr{H}(h,v) = \frac{p^2}{2m} + mgh \tag{2}$$

b. Calculate the classical approximation to the partition function Q.

$$Q = \frac{1}{h}\int_{-\infty}^{\infty} \mathrm{d}p\, e^{-\beta\frac{p^2}{2m}} \int_0^{+\infty} \mathrm{d}h e^{-\beta U(h)} \tag{3}$$

$$= \frac{1}{h}\sqrt{\frac{2\pi m}{\beta}} \int_0^{+\infty} \mathrm{d}h e^{-\beta mgh} \tag{4}$$

$$= \frac{1}{h}\sqrt{\frac{2\pi m}{\beta}} \left[-\frac{1}{\beta mg} e^{-\beta mgh}\right]_0^{+\infty} \tag{5}$$

$$= \sqrt{\frac{m}{2\pi\hbar^2\beta}}\ \frac{1}{\beta mg} \tag{6}$$

$$= \frac{1}{\sqrt{2\pi\hbar^2\beta mg^2}}\ \frac{1}{\beta} \tag{7}$$

$$= \frac{1}{\sqrt{2\pi\hbar^2 mg^2}}\ \frac{1}{\beta^{3/2}} \tag{8}$$

$$\tag{9}$$

c. Calculate the internal energy E.

$$E = \frac{\partial(\beta F)}{\partial\beta} \tag{10}$$

$$F = -\frac{1}{\beta}\ln Q \tag{11}$$

$$E = -\frac{\partial}{\partial\beta}\left[\ln\frac{1}{\sqrt{2\pi\hbar^2 mg^2}} - \ln\beta^{3/2}\right] \tag{12}$$

$$= -\frac{\partial}{\partial\beta}\left[-\frac{3}{2}\ln(\beta)\right] \tag{13}$$

$$= \frac{3}{2\beta} \tag{14}$$

d. Given an observable $A = A(h)$, write down the expectation value $\langle A \rangle$.

$$\langle A(h) \rangle = \frac{1}{Z} \int \mathrm{d}h\, A(h) e^{-\beta mgh} \tag{15}$$

$$Z = \int_0^{+\infty} \mathrm{d}h\, e^{-\beta mgh} = \frac{1}{\beta mg} \tag{16}$$

2 Monte Carlo simulation

Take the result from 1d. The key idea of Monte Carlo algorithms is to split the integrand into two factors $f(h)$ and $g(h)$. It can then be approximated by

$$\int \mathrm{d}h f(h) g(h) = \langle g \rangle_f \approx \frac{1}{N} \sum_{i=1}^{N} g(\xi_i), \tag{17}$$

where $\langle g \rangle_f$ is the expectation value of g under the probability distribution

$$\rho(h) = \frac{f(h)}{\int \mathrm{d}h f(h)} \tag{18}$$

and the ξ_i are a samples from ρ.

a. Set $A(h) = h$, how would you choose $f(h)$ and $g(h)$ and why?

The factor $f(h)$ has to be normalized/normalizable. Therefore, we can choose $f(h)$ as:

$$f(h) = \frac{1}{Z}\, e^{-\beta mgh} = \beta mg\, e^{-\beta mgh} \tag{19}$$

Consequently, the factor $g(h)$ corresponds to our observable $A(h) = h$. The expectation value of $A(h)$ can be calculated analytically, which is useful for later comparison with the numerical result:

$$\langle A(h) \rangle = \beta mg \int \mathrm{d}h\, h\, e^{-\beta mgh} \tag{20}$$

$$= \beta mg \left[\frac{1}{(-\beta mg)^2} \, (-mgh - 1)\, e^{-\beta mgh} \right]_0^{+\infty} \tag{21}$$

$$= \frac{1}{\beta mg} \tag{22}$$

b. Implement a simulation that uses the approximation (17) to calculate the average height and the average potential energy of the particle using the programming language of your choice. Use a random number generator to draw the sample ξ. Use the parameters $T = 300\,\mathrm{K}$, $m = 4.66 \times 10^{-26}\,\mathrm{kg}$, $k_\mathrm{B} = 1.38 \times 10^{-23}\,\mathrm{J\,K^{-1}}$, and $g = 9.81\,\mathrm{m\,s^{-2}}$. Discuss your results.

As described in equation 14, we need samples ξ drawn from $\rho(h)$ to estimate the expectation value of h and $U(h)$. An easy way to draw samples from a probability distribution is to make use of the inverse cumulative distribution function (CDF). A CDF can be obtained by integrating any probability density function up to a point x:

$$F_X(x) = \int_{-\infty}^{x} f_X(t) \mathrm{d}t \tag{23}$$

Inserting $\rho(h)$ into the above expression and adjusting the integration bounds leads to:

$$F_H(h) = \beta mg \int_0^h \mathrm{d}t \; e^{-\beta mgt} \tag{24}$$

$$= \beta mg \left[-\frac{1}{\beta mg} \; e^{-\beta mgt} \right]_0^h \tag{25}$$

$$= -e^{-\beta mgh} + 1 \tag{26}$$

When feeding samples from $\rho(h)$ into the CDF, the result is a uniform distribution in the range [0,1]. We want to find the inverse of the CDF so that we can feed this function with a uniform distribution and obtain samples from $\rho(h)$. The inverse is:

$$h(r) = -\frac{1}{\beta mg} \ln(1 - r) \tag{27}$$

where r is a random variable between 0 and 1 drawn from a uniform distribution. An example implementation in python is attached below.

[47]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[48]:
```
N_samples = 1000000

T = 300 # K
m = 4.66e-26 # kg
k_B = 1.38e-23 # J/K
g = 9.81 # m/s^2

beta = 1/(k_B*T)
```

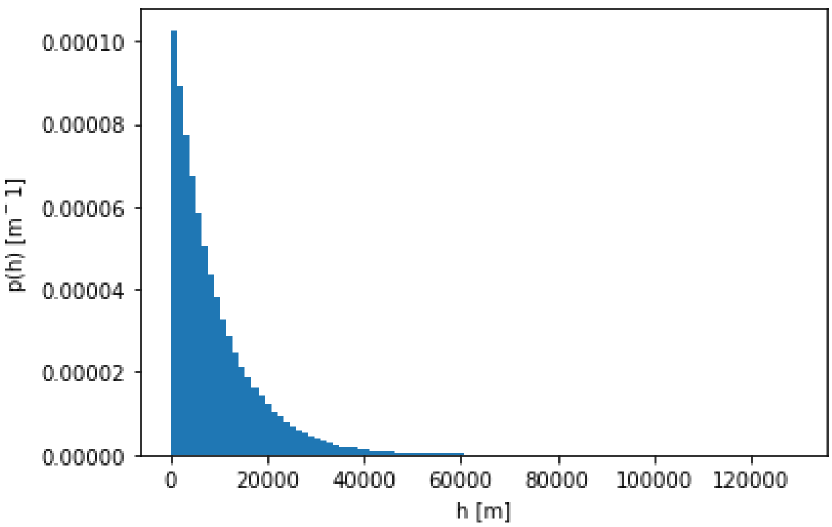
[49]:
```
def inv_cumulative(x, beta, m, g):
    """Returns the value of the inverse cumuative distribution function for␣
 ↪f(h)"""
    return -(1/(beta * m * g)) * np.log(1 - x)
```

[50]:
```
# Sample values from uniform distribution between 0 and 1
x = np.random.rand(N_samples)

# Draw from f(h) using the inverse cumulative distribution function
h = inv_cumulative(x, beta, m, g)
```

[51]:
```
# Let's have a look at the distribution of h
plt.hist(h, density=True, bins = 100)
plt.xlabel("h [m]")
plt.ylabel("p(h) [m$^-1$]")
```

[51]:
```
Text(0, 0.5, 'p(h) [m$^-1$]')
```



[52]:
```
avg_h = 1/N_samples * np.sum(h)
avg_h_analytical = 1/(beta*m*g)

avg_U = avg_h * m * g
avg_U_analytical = 1/beta
```

[53]:
```
print("Average height: {}".format(avg_h))
print("Analytical average height: {}\n".format(avg_h_analytical))

print("Average potential energy: {}".format(avg_U))
print("Analytical average potential energy: {}".format(avg_U_analytical))
```

```
Average height: 9069.380922886385
Analytical average height: 9056.187738709295

Average potential energy: 4.1460312113738195e-21
Analytical average potential energy: 4.14e-21
```

The estimate of the average height from simulations will asymptotically approach the true value with more trials.

We can repeat our simulation multiple times to get a better estimate of the average height:

[54]:
```
N = 100
avg_h_list, avg_U_list = [], []

for i in range(N):

    x = np.random.rand(N_samples)
    h = inv_cumulative(x, beta, m, g)

    avg_h_list.append(1/N_samples * np.sum(h))
    avg_U_list.append(avg_h_list[-1] * m * g)

print("Average Height from {} simulations: {}".format(N, np.average(avg_h_list)))
print("Analytical average height: {}\n".format(avg_h_analytical))

print("Average potential energy from {} simulations: {}".format(N, np.
 ↪average(avg_U_list)))
print("Analytical average potential energy: {}".format(avg_U_analytical))
```

```
Average Height from 100 simulations: 9057.178031174872
Analytical average height: 9056.187738709295

Average potential energy from 100 simulations: 4.140452708239468e-21
Analytical average potential energy: 4.14e-21
```