OBLIG 2 FYS2160

Tiril Sørum Gransjøen

1 The Einstein Crystal – unstructured answer

The Einstein crystal is a model used to represent an ideal crystal. We can therefore use this crystal to further model how temperature and energy transfers in solids. The model of our ideal crystal is built upon a few presumptions, as explained in *"Elementary Thermal and Statistical Physics Using Python"* page 117. The *N* atoms are evenly spaced, fixed in place in a tree-dimensional repeating pattern called a lattice structure. This structure, that is constant with no defects, is what makes the crystal perfect. We know that the atoms interact in a way that causes them to act as independent harmonic oscillators. This, we can use to our advantage when studying energy transfer within a solid, which is what we are going to do in this assignment.

We are interested in a system which consists of two of theses Einstein crystal in thermal contact, which we simply can model as a single Einstein crystal divided into two parts A and B with different conditions for the N_A, N_B oscillators and q_A, q_B energy units. We know that within the crystal itself, N and q are constant. We can therefore describe $N = N_A + N_B$ and $q = q_A + q_B = q_A + (q - q_A)$. If we now look at how N and q is expressed for an ideal crystal, we can set a couple condition for the scenario we wish to model, and then visualize energy partitions and the probability of finding the system in each of these states. We use the formula (6) from the assignment text to describe the number of microstates – meaning number of different energy partitionings within our system – meaning Ω_A and Ω_B. The multiplicity of the whole system is then the macrostate described as $\Omega = \Omega_A(N_A, q_A)\Omega_B(N_B, q_B)$. As a further result of the *fundamental assumption of statistical mechanics*, we express the probability of the energy partitioning of being in a certain condition is given as described on page 123 of *"Elementary Thermal and Statistical Physics Using Python"* – all accessible microstates are equally probable in our ideal crystal. Therefore, we express the probability of a certain energy partitioning as the number of macrostates for part A multiplied by the number of macrostates for part B of the crystal in the same condition (which will be the same for whichever state we choose), divided by the total amount of macrostates within our entire system:

$$P(q_A) = \frac{\Omega_A(N_A, q_A)\Omega_B(N_B, q - q_A)}{\Omega(N_A + N_B, q)} \tag{12}$$

We are further interested in the entropy and temperature in our system. Entropy is a measure of disorder or unavailability of thermal energy, defined as the logarithm of the multiplicity of our system. From theory on page 139 we get the expression for entropy as a function of the macrostate q_A:

$$S(q_A) = k_b * ln\Omega(q_A) = k_b * ln\frac{(q_A + N_A - 1)!\,(q_B + N_B - 1)!}{q_A!\,(N_A - 1)!\;q_B!\,(N_B - 1)!} \tag{13}$$

Where k_b is the Boltzmann constant. The temperature is defined from formula (6.66) as the inversely proportional to the slope of our entropy expression. Energy typically flows from a system with high temperature to one lower temperature, which means when we have a slope in our $S(q_A)$ curve larger than $S(q_B)$, the temperature will increase in B as a result of the energy transfer that takes place $\frac{1}{T} = \frac{\delta S}{\delta E_{N,V}}$ (14). These formulas can be used to plot entropy and temperature as a function of our macrostate if we need or want.

From this theory, we can easily write a python program that plots the probability of each total macrostate q_A:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import comb

def calculate_prob(N1, N2, E_total, omega):
    probabilities = np.zeros((E_total + 1, E_total + 1))
    for E1 in range(E_total + 1):
        for E2 in range(E_total - E1 + 1):
            microstates_1 = comb(N1 + E1 // omega - 1, N1) #formula 6
            microstates_2 = comb(N2 + E2 // omega - 1, N2) #formula 6
            total_microstates = comb(N1 + N2 + E_total // omega - 1, N1 + N2) #formula 6
            probabilities[E1, E2] = (microstates_1 * microstates_2) / total_microstates #formula 12
    return probabilities

N1 = 200  ; N2 = 200 ; E_total = 200
omega = 1.0  # Angular frequency of the oscillators

probabilities = calculate_prob(N1, N2, E_total, omega)
E1_values = np.arange(0, E_total + 1)

probability_values = probabilities[E1_values, :]
plt.title("Probability for energy partitionings of energy between two Einstein crystals")
plt.plot(E1_values, probability_values)
plt.xlabel('qA')
plt.ylabel('Probability P(qA, qB)')
plt.grid(True)
plt.show()
```

Figure 1: python code that calculates probability for energy partitionings between system A and B, and plots it as a function of q_A.

We run the code and get the plot:

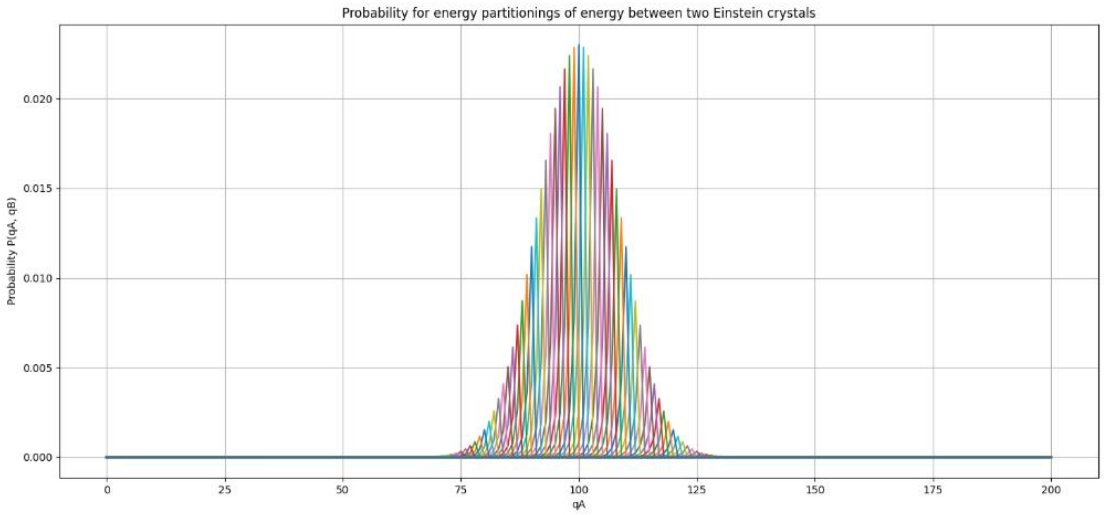

Figure 2: Plot as a result of code represented in Figure 1

We see a plot of the probability, with a Gaussian shape around $q_A = \frac{q}{2}$, which we can explain because of the maximum for our multiplicity at this point. The closer we get to this point, the higher the probability of this specific energy partitioning q_A and $q_B = q - q_A$ – this because the system naturally exchanges energy in a way that moves the system towards this equilibrium of energy.

1 The Einstein Crystal – structured answer

a. For a system with N=3 oscillators and q=3 we have the microstates;
(3,0,0), (0,3,0), (0,0,3), (2,1,0), (2,0,1), (0,2,1), (1,2,0), (0,1,2), (1,0,2), (1,1,1)

b. If inserted q=3 and N=3 into the formula (6) from the assignment text we get $\frac{(3+3-1)!}{3!(3-1)!} = \frac{5!}{3!2!} = \frac{5*4*3*2}{3*2*2} = 5 * 2 = 10$ which is consistent with our result above.

c. If we have N_A=2, q_A=5 and N_B=2, q_B=1, and we get the different combinations of microstates;
(6,0)A and (0,1)B, (6,0)A and (1,0)B
(5,1)A and (0,1)B, (5,1)A and (1,0)B
(4,2)A and (0,1)B, (4,2)A and (1,0)B
(3,3)A and (0,1)B, (3,3)A and (1,0)B
(2,4)A and (0,1)B, (2,4)A and (1,0)B
(1,5)A and (0,1)B, (1,5)A and (1,0)B
(0,6)A and (0,1)B, (0,6)A and (1,0)B

d. For $N_A = 2, N_B = 2$ and q=6, we know that $q_A + q_B = q$. The energy can be distributed in the following ways;
$q_A = 6\ and\ q_B = 0$, $q_A = 5\ and\ q_B = 1$, $q_A = 4\ and\ q_B = 2$, $q_A = 3\ and\ q_B = 3$,
$q_A = 2\ and\ q_B = 4$, $q_A = 1\ and\ q_B = 5$, $q_A = 0\ and\ q_B = 6$

e. We write a simple program that calculates amount of microstates as a function of NA, NB and qA;

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import comb
import random

def microstates(NA,NB,qa,q):
    qb=q-qa
    microA=comb(qa+NA-1,qa)
    microB=comb(qb+NA-1,qb)
    total_micro=microA*microB
    return microA, microB, total_micro

print(microstates(2,2,1,6))
```

Figure 3: short program that calculates number of microstates for a system of two Einstein crystals, defined by arguments NA, NB, q and qa. We use our comb function to describe the number of microstates given by formula (6)

Which prints (2,6,12) in our terminal.

f. We know that before the thermal contact each system will have its own number of microstates. After contact we will have the total number of microstates given by microA*microB. The amount of possible microstates will increase because our possibilities for combinations of distribution increases.

g. We use the formula for our probability (12) in our code, and plot the probability against qA varying from 0 to q=100:

```
NA = 50
NB = 50
q = 100
qA_values = list(range(0, q + 1))

max=0; maxq=0
prob = np.zeros((q + 1,q + 1))
for qa in range(q + 1):
    for qb in range(q - qa+ 1):
        microstates_1 = comb(NA + qa - 1, NA) #formula 6
        microstates_2 = comb(NB + qb - 1, NB) #formula 6
        total_microstates = comb(NA + NB + q - 1, NA + NB) #formula 6
        prob[qa, qb] = (microstates_1 * microstates_2) / total_microstates #formula 12
    if prob[qa, qb]>max:
        max=prob[qa, qb]
        maxq=qa

plt.plot(qA_values, prob)
plt.xlabel('qA')
plt.ylabel('Probability (P(qA))')
plt.title('Probability Distribution P(qA)')
plt.show()
```

Figure 4: Code for energy-distribution probability

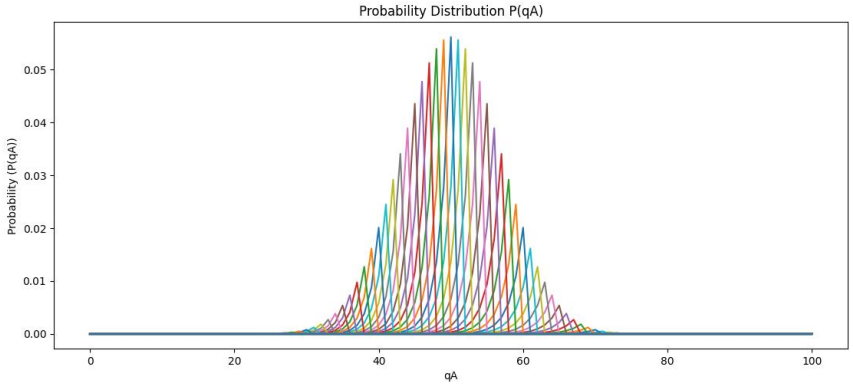

Figure 5: Plot as a result of code in Figure 6

We write additional code that prints the most probable macrostate, and how probable it is;

```
print(f"The most probable macrostate is qA = {maxq}, with probability P(qA) = {max}")
```

Which prints to our terminal; The most probable macrostate is qA=50, with probability P(qA)=0.0562078.

Ca. Full derivation of how we can approximate formula (8) ion the assignment text is on page 128/129 in *"Elementary Thermal and Statistical Physics Using Python"*.

$$ln\Omega(N,q) = ln\left(\frac{(q+N-1)!}{q!\,(N-1)!}\right) = \ln(q+N-1)! - \ln(q)! - \ln(N-1)!$$

We use the fact that $N \gg 1$, and $\ln(x)! \simeq xlnx - x$, and rewrite:

$$\ln(q+N)! - \ln(q)! - \ln(N)! = (q+N)\ln(q+N) - (q+N) - qln(q) + q - Nln(N) - N$$

$$= (q+N)\left(lnq + \ln\left(1+\frac{N}{q}\right)\right) - qlnq - NlnN$$

Since $\frac{N}{q} \ll 1$, and $lnx \simeq x$ we get the expression

$$ln\Omega(N,q) = (q+N)\left(lnq + \frac{N}{q}\right) - qlnq - NlnN = N + Nlnq + \frac{N^2}{q} - NlnN$$

$$= N\left(1 + lnq - lnN + \frac{N}{q}\right) = N\left(1 + \ln\left(\frac{q}{N}\right)\right)$$

Which is what we wanted to show. From this we can express the multiplicity if we wish:

$$\Omega(N,q) = e^{ln\Omega(N,q)} = e^{N\left(1+\ln\left(\frac{q}{N}\right)\right)} = e^N\left(\frac{q}{N}\right)^N$$

Cb. The entropy is given by a formula on page 139 of *Elementary Thermal and Statistical Physics Using Python:*

$$S = kln(\Omega) = kN(\ln\left(\frac{q}{N}\right) + 1)$$

We have shown that the multiplicity can be written as $\left(\frac{qe}{N}\right)^N$, so we get the expression for the entropy rewritten to

$$S = kln(\Omega) = kln\left(\left(\frac{qe}{N}\right)^N\right) = kNln\left(\frac{qe}{N}\right)$$

If we wish, we can also express the entropy in terms of the energy $E = q\Delta\epsilon$.

Cc. The temperature of the Einstein crystal can be found by using formula from page 142 in the textbook.

$$\frac{1}{T} = \left(\frac{\delta S}{\delta E}\right)_{N,V} \Longrightarrow \frac{1}{T} = \frac{Nk}{E} \Longrightarrow T = \frac{E}{Nk}$$

The Spin System – structured answer

a. We know each spin only can have two different states, which means we need to consider these two states per N spins. Furthermore, we know that to calculate the total amount of states possible, we need to express the situation mathematically as 2^N

b. The energy E has the expression $E = -S\mu B$ from the assignment text. To find the total energy as a function of spin, we need to sum this over all our N-spins within the system. The macrostate of spins has an expression given in the assignment text S=2s, and therefore we get the expression for total energy $E = -2s\mu B$

c. We now wish to illustrate a system that takes into consideration spin. The entropy and temperature is visualized by code by using the formulas (13) and (14). The program reuses a lot of code from the previous oblig answer.

```
1   import numpy as np
2   import matplotlib.pyplot as plt
3   import random
4
5   ns = 10000
6   ss = np.zeros(ns)
7   N = 50
8
9   for i in range(ns):
10      s_pos=np.random.randint(2,size=N)
11      s_pos=np.sum(s_pos==1)
12      s_neg=N-s_pos
13      s=(s_pos-s_neg)/2
14      ss[i]=s
15
16  n, s = np.histogram(ss,1000)
17  plt.plot(s[1:], n /ns)
18  plt.xlabel("s")
19  plt.ylabel("P(s)")
20  plt.show()
```

Figure 6: Code taking independent spin of the N atoms in our isolated Einstein crystal into consideration, and plots the probabilities as a function of net spin s.

We know that each atom spins in a way that is independent of other atoms in the system, and they can only be described as S_+ and S_-. Therefore we use the function *random.randint()* to decide a random number of our N atoms that are in the condition S_+, and the rest are S_-. We use a for-loop to calculate the net spin s, and then the *numpy* function histogram to plot a histogram, as asked. We run the code and get the plot:

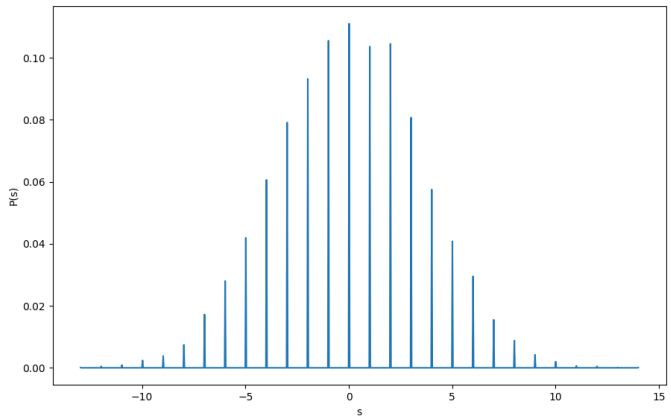

Figure 7: Plot as result of python code in figure 6. Shows distribution of net spin s within a system with N=50.

d. We want to find the multiplicity of a macrostate S_+. This means we want to find out how many different "combinations" we can find for S_+ within the N-spin system. We do this by

$$\Omega(N, S_+) = \binom{N}{S_+} = \frac{(N)!}{S_+!\,(N - S_+)!}$$

Since we know $(N - S_+)!$ Describes the states of the system when the spin is not S_+, we can replace this argument with S_-, and we see that we've shown what we wanted.

$$\Omega(N, S_+) = \binom{N}{S_+} = \frac{(N)!}{S_+!\,(S_-)!}$$

e. Writing the multiplicity as a function of s:

$$\Omega(N, S_+) = \binom{N}{s} = \frac{(N)!}{\left(\frac{N}{2} + s\right)!\left(\frac{N}{2} - s\right)!}$$

To show that this is possible, all we need to do is show that there is a connection between S_+ and $\frac{N}{2} + s$, as well as S_- and $\frac{N}{2} - s$. We do this easily:

$S_+ + S_- = N$ and $S_+ - S_- = 2s$, and since we want $\frac{N}{2} + s$, we add together our expressions for N and 2s: $N + 2s = S_+ + S_- + S_+ - S_- = 2S_+ \Rightarrow S_+ = N/2 + s$ Which is the expression we needed. By following the same logic, we do get the expression for $S_- = N/2 - s$. We see that the two expressions for multiplicity are equal.

f. Didn't manage to answer this question, unfortunately.

g. To compare the analytical result with our histogram we created previously, all we need to do is add a couple lines of code to the program we already have, that implements our formula (19), and plots the gaussian distribution that way:

```
22    gauss = np.max(n /ns) * np.exp(-2 * s[1:] ** 2 / N)
23    plt.plot(s[1:], gauss,label="GAUSS DISTRIBIUTION")
24    plt.legend()
25    plt.show()
```

Figure 8: code plotting the gaussian distribution of microstates of spin within our system.

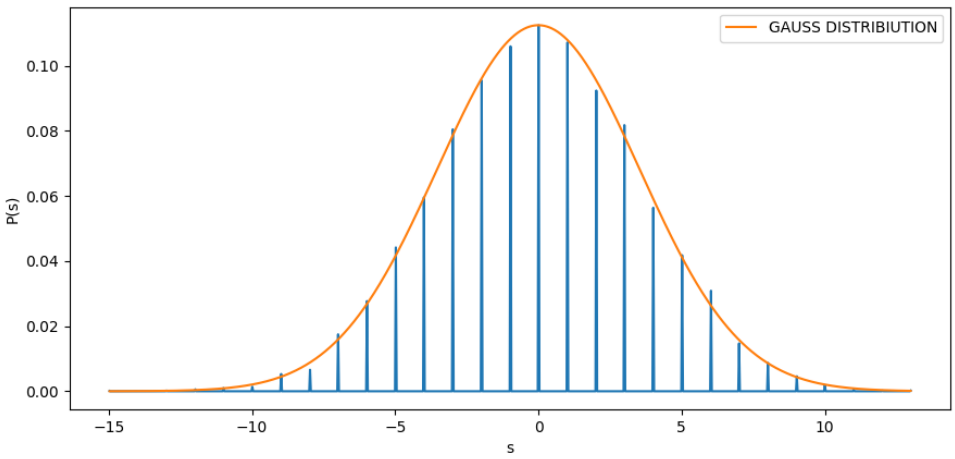

Figure 9: figure showing code represented in figure 8, comparing the analytical solution and the histogram we found previously.

h. The entropy as a function of N and S_+ is found easily by rewriting the expression we already know for entropy:

$$S = kln\big(\Omega(N, S_+)\big) = kln\left(\frac{(N)!}{S_+!\,(N-S_+)!}\right)$$
$$= NlnN - N - S_+lnS_+ + S_+ - (N-S_+)\ln(N-S_+) + (N-S_+)$$
$$= NlnN - S_+lnS_+ - (N-S_+)\ln(N-S_+)$$

i. We use the hint in the problem text to find an expression:

$$\frac{1}{T} = \left(\frac{\delta S}{\delta U}\right)_{N,B} = \left(\frac{\delta S_+}{\delta U}\right)\left(\frac{\delta S}{\delta S_+}\right)$$
$$\left(\frac{\delta S}{\delta S_+}\right) = \frac{\delta(NlnN - S_+lnS_+ - (N-S_+)\ln(N-S_+))}{\delta S_+} = -lnS_+ + \ln(N-S_+)$$
$$\left(\frac{\delta S_+}{\delta U}\right) = -\frac{1}{2\mu B}$$
$$\frac{1}{T} = -\frac{1}{2\mu B}\left(\frac{\delta S}{\delta S_+}\right) = -\frac{(-lnS_+ + \ln(N-S_+))}{2\mu B} = -\frac{\ln\left(\frac{N-S_+}{S_+}\right)}{2\mu B}$$

$$\Longrightarrow T = -\frac{2\mu B}{ln\left(\frac{N-S_+}{S_+}\right)}$$