1 Including optical phonons in the Debye model *(5 points)*

Consider a NaCl crystal that has a diatomic basis and therefore not only acoustic but also optical phonons. In the spirit of the Debye model, the density of states in the phonon spectrum can be approximated as

$$D(\omega) = D_{\mathrm{D}}(\omega) + \delta(\omega - 2\omega_{\mathrm{D}}) \qquad \text{with} \qquad D_{\mathrm{D}}(\omega) = \begin{cases} 3\frac{\omega^2}{\omega_{\mathrm{D}}^3} & \text{for } \omega \leq \omega_{\mathrm{D}} \\ 0 & \text{for } \omega > \omega_{\mathrm{D}} \end{cases} \tag{1}$$

The first contribution describes the acoustic phonons as discussed in the lecture. The second contribution approximately describes the optical phonons.

a) Calculate the specific heat of the crystal for high and low temperatures. ***(5 points)***

According to the Debye model, the modes are counted in the following way:

$$\sum_{\text{modes}} (...) = 3N \int_0^\infty \mathrm{d}\omega \, D(\omega) \, (...) \tag{2}$$

Thus, the energy is given by

$$E = \sum_{\text{modes}} \hbar\omega \left(\frac{1}{e^{\beta\hbar\omega} - 1} - \frac{1}{2} \right) \tag{3}$$

$$= E_0 + 3N \int_0^\infty \mathrm{d}\omega \, D(\omega) \, \frac{\hbar\omega}{e^{\beta\hbar\omega} - 1} \, . \tag{4}$$

The specific heat c_v is given by

$$\begin{aligned} c_v(T) = \frac{\partial E}{\partial T} &= \frac{3\hbar^2 N}{k_B T^2} \int_0^\infty \mathrm{d}\omega \, D(\omega) \, \frac{e^{\beta\hbar\omega} \, \omega^2}{\left(e^{\beta\hbar\omega} - 1\right)^2} \\ &= \frac{3\hbar^2 N}{k_B T^2} \int_0^\infty \mathrm{d}\omega \, \left[D_{\mathrm{D}}(\omega) + \delta(\omega - 2\omega_{\mathrm{D}}) \right] \, \frac{e^{\beta\hbar\omega} \, \omega^2}{\left(e^{\beta\hbar\omega} - 1\right)^2} \\ &= \frac{9Nk_B}{u_m^3} \int_0^{u_m} \mathrm{d}u \, \frac{e^u \, u^4}{\left(e^u - 1\right)^2} + \frac{12\hbar^2 N}{k_B T^2} \, \frac{e^{2\beta\hbar\omega_D} \, \omega_D^2}{\left(e^{2\beta\hbar\omega_D} - 1\right)^2} \, , \end{aligned} \tag{5}$$

where the first term is well known from the lecture and $u_m = \beta k_B T_D$ with the Debye temperature T_D and $\omega_D = k_B T_D / \hbar$. The second term approximately describes the optical phonons.

High temperature limit:

We know that the first term becomes $c_{v,\mathrm{D}}(T) \approx 3Nk_B$. The second term can be approximated by $c_{v,\delta}(T) \approx 3Nk_B$. Thus

$$c_v(T) \approx 6Nk_B \, . \tag{6}$$

Low temperature limit:

We know that the first term becomes $c_{v,\mathrm{D}}(T) \approx \frac{12}{5}\pi^4 Nk_B (T/T_D)^3$. The second term can be approximated by $c_{v,\delta}(T) \approx 0$. Thus

$$c_v(T) \approx \frac{12\pi^4}{5} Nk_B \left(\frac{T}{T_D} \right)^3 \, . \tag{7}$$

2 Rigid rotator *(8 points)*

Consider a molecule, such as carbon monoxide, which consists of two different atoms, one carbon and one oxygen, separated by a distance d. Such a molecule can exist in quantum states of different orbital angular momentum. Each state has the energy

$$\varepsilon_l = \frac{\hbar^2}{2I} \cdot l(l+1)\,, \tag{8}$$

where $I = \mu d^2$ is the moment of inertia of the molecule about an axis through its centre of mass and μ is the reduced mass defined by

$$\frac{1}{\mu} = \frac{1}{m_1} + \frac{1}{m_2}\,. \tag{9}$$

$l = 0, 1, 2, ...$ is the quantum number associated with the orbital angular momentum. Each energy level of the rotating molecule has the degeneracy

$$g_l = 2l + 1\,. \tag{10}$$

a) Find the general expression for the canonical partition function Z. *(1 point)*

Definition of partition sum:

$$Z = \sum_i \exp(-\beta E_i) \tag{11}$$

Let us define $k := \frac{\beta\hbar^2}{2I}$.

In the case of a rigid rotator with degenerate energy levels, the partition sum becomes

$$Z = \sum_{l=0}^{\infty} g_l \cdot \exp\Big(-k \cdot l(l+1)\Big) \tag{12}$$

$$= \sum_{l=0}^{\infty} (2l+1) \cdot \exp\Big(-k \cdot l(l+1)\Big) \tag{13}$$

b) Show that for high temperatures, Z can be approximated by an integral. Calculate the integral. HINT: For high T, find an approximate integral representation for the summands at given l and demonstrate that the integral can be extended over the complete summation range. *(2 points)*

For high temperatures, the energy spacing between the individual energy levels becomes small (when compared to $k_B T$). Thus, we can replace the summation by the integral

$$Z = \int_0^{\infty} (2l+1) \cdot \exp\Big(-k \cdot l(l+1)\Big) \cdot dl \tag{14}$$

$$= \int_0^{\infty} \exp\Big(-k \cdot l(l+1)\Big) \cdot d(l(l+1)) \tag{15}$$

$$= \frac{1}{k} = \frac{2I}{\hbar^2\beta} \tag{16}$$

c) Evaluate the high temperature mean energy E and the heat capacity C_V. *(2 points)*

The mean energy E (i.e. short for $\langle E \rangle$) can be calculated from the partition sum

$$E = -\frac{\partial \ln Z}{\partial \beta} \tag{17}$$

$$= -\frac{\partial}{\partial \beta} \ln\left(\frac{2I}{\hbar^2 \beta}\right) \tag{18}$$

$$= \frac{1}{\beta} = k_B T \tag{19}$$

Heat capacity:

$$C_V = \frac{\partial E}{\partial T} = k_B \tag{20}$$

d) Find the low-temperature approximations to the canonical partition function, the mean energy E and the heat capacity C_V. *(3 points)*

At low temperatures, the majority of the particles will be occupying the ground state. We can therefore approximate the partition function simply by ignoring all terms after the first two:

$$Z = \sum_{l=0}^{\infty} (2l+1) \cdot \exp\left(-l(l+1) \cdot \frac{\hbar^2 \beta}{2I}\right) \tag{21}$$

$$= 1 + 3 \cdot \exp\left(-\frac{\beta \hbar^2}{I}\right) \tag{22}$$

The average energy is thus

$$E = -\frac{\partial}{\partial \beta} \ln Z \tag{23}$$

$$= -\frac{\partial}{\partial \beta} \ln\left[1 + 3 \cdot \exp\left(-\frac{\beta \hbar^2}{I}\right)\right] \tag{24}$$

$$= \frac{3\hbar^2/I}{e^{\beta \hbar^2/I} + 3} \tag{25}$$

And the heat capacity:

$$C_V = \frac{\partial \beta}{\partial T} \frac{\partial}{\partial \beta} E \tag{26}$$

$$= -\frac{1}{k_B T^2} \cdot \frac{\partial}{\partial \beta}\left(\frac{3\hbar^2/I}{e^{\beta \hbar^2/I} + 3}\right) \tag{27}$$

$$= \frac{3\hbar^4}{k_B T^2 I^2} \cdot \frac{e^{\hbar^2 \beta/I}}{e^{\beta \hbar/I} + 3} \tag{28}$$

$$\approx 3k_B \left(\frac{\hbar^2}{I k_B T}\right)^2 \cdot \exp\left(-\frac{\hbar^2}{I k_B T}\right) \tag{29}$$

3 Adsorption of molecules to a surface *(7 points)*

Consider a gas in contact with a solid surface. The molecules of the gas can adsorb to specific sites on the surface. These sites are sparsely enough distributed over the surface that they do not directly interact. In total, there are N adsorption sites, and each can adsorb $n = 0$, $n = 1$ or $n = 2$ molecules. When an adsorption site is unoccupied, the energy of the site is zero. When an adsorption site is occupied by a single molecule, the energy of the site is ε_1. When an adsorption site is doubly occupied, the adsorption energy is ε_2. In addition, the two adsorbed molecules can interact in a vibrational mode with frequency ω, so that the energy of the doubly occupied adsorption site is

$$\varepsilon_2 + \nu\hbar\omega \qquad \text{with} \qquad \nu = 0, 1, 2, ...$$

The gas above the surface can be considered as a heat and particle reservoir with temperature T and chemical potential μ.

a) Calculate the grand canonical partition sum Z_G. *(2 points)*

The grand canonical partition sum is defined as

$$Z_G = \sum_i e^{-\beta(E_i - \mu N_i)} \,. \tag{30}$$

In our case, this is a sum over all states i with unoccupied sites, sites that are occupied only once, sites that are occupied only twice and a mixture of once and twice occupied sites. Since the N sites do not interact, the grand canonical partition sum can be calculated by

$$Z_G = z_G^N \,, \tag{31}$$

where z_G is the single particle partition sum,

$$z_G = 1 + e^{-\beta(\varepsilon_1 - \mu)} + \sum_{\nu=0}^{\infty} e^{-\beta(\varepsilon_2 + \nu\hbar\omega - 2\mu)} \,, \tag{32}$$

because one site can be either unoccupied, occupied once or occupied twice. When the site is occupied twice, there are many vibrational modes with $\nu = 0, 1, 2, ...$ Thus, we have

$$Z_G = \left(1 + e^{-\beta(\varepsilon_1 - \mu)} + \frac{e^{-\beta(\varepsilon_2 - \hbar\omega - 2\mu)}}{e^{\beta\hbar\omega} - 1} \right)^N \,. \tag{33}$$

b) Calculate the grand canonical potential Ψ. *(1 point)*

$$\Psi = -k_B T \ln Z_G = -N k_B T \ln \left(1 + e^{-\beta(\varepsilon_1 - \mu)} + \frac{e^{-\beta(\varepsilon_2 - \hbar\omega - 2\mu)}}{e^{\beta\hbar\omega} - 1} \right) \,. \tag{34}$$

c) Calculate the mean number of adsorbed molecules on the surface from Z_G. *(1 point)*

We can use a small trick here: First, calculate the mean number of adsorbed molecules on one site and then multiply by the total number of sites N (since all sites are equal and independent):

$$\begin{aligned}\langle N\rangle &= N\left(\frac{0\cdot 1+1\cdot e^{-\beta(\varepsilon_1-\mu)}+2\cdot\frac{e^{-\beta(\varepsilon_2-\hbar\omega-2\mu)}}{e^{\beta\hbar\omega}-1}}{z_G}\right)\\ &= N\left(\frac{e^{-\beta(\varepsilon_1-\mu)}+\frac{2e^{-\beta(\varepsilon_2-\hbar\omega-2\mu)}}{e^{\beta\hbar\omega}-1}}{z_G}\right).\end{aligned} \tag{35}$$

d) Calculate the mean number of adsorbed molecules on the surface directly from Ψ and convince yourself that it gives the same as calculated in c). *(2 points)*

$$\begin{aligned}\langle N\rangle &= \frac{1}{\beta}\partial_\mu \ln Z_G = -\partial_\mu\Psi\\ &= Nk_BT\,\frac{\beta e^{-\beta(\varepsilon_1-\mu)}+\frac{2\beta e^{-\beta(\varepsilon_2-\hbar\omega-2\mu)}}{e^{\beta\hbar\omega}-1}}{\left(1+e^{-\beta(\varepsilon_1-\mu)}+\frac{e^{-\beta(\varepsilon_2-\hbar\omega-2\mu)}}{e^{\beta\hbar\omega}-1}\right)}\\ &= N\left(\frac{e^{-\beta(\varepsilon_1-\mu)}+\frac{2e^{-\beta(\varepsilon_2-\hbar\omega-2\mu)}}{e^{\beta\hbar\omega}-1}}{z_G}\right).\end{aligned} \tag{36}$$

e) Give the probability that an adsorption site is in the state with $n=2$ and $\nu=3$. *(1 point)*

$$p(n=2,\nu=3) = \frac{e^{-\beta(\varepsilon_2+3\hbar\omega-2\mu)}}{z_G}. \tag{37}$$

4 Random walk & self-avoiding random walk *(10 bonus points)*

In this exercise, we will write two small computer programs for random walks on a $2D$ square lattice:

- one for the usual random walk, where the walker is allowed to come back to points already visited
- one for the so-called self-avoiding random walk (SAW), where the walker is not allowed to do so and hence does not cross its own path.

HINT: One can find a lot of algorithms for the SAW in the web. It is ok to use just the simplest one: When the walker revisits a position, you discard the walk. Otherwise you use the trajectory. Note that this algorithm for the SAW is not very efficient (you have to reject more and more trajectories for larger n), so you should restrict yourself to $n < N$ with, say, $N = 30$. For the usual random walk there is no such problem and you can explore larger N.

a) Write the program for a typical RW

The code for a non-self-avoiding random walk is pretty straight-forward: At each step, we randomly choose an axis along which we will move (x or y), and then choose a random number from the set $\{-1, +1\}$ to get the direction along the chosen axis: y-direction, respectively.

calc_random_walk_trajectory.py

```
import numpy as np


def main(nr_of_steps, x=0, y=0):

    def random_step(x, y):
        direction = np.random.choice(['x', 'y'])
        if direction == 'x':
            dx = np.random.choice([-1, 1])
            dy = 0
        elif direction == 'y':
            dx = 0
            dy = np.random.choice([-1, 1])

    trajectory = []
    for i in range(nr_of_steps):
        x, y = random_step(x, y)
        trajectory.append((x, y))

    return trajectory
```

It was not entirely clear from the formulation in the problem set whether diagonal steps are allowed as well. Including them would lead to the step size not being constant (sometimes being 1, sometimes $\sqrt{2}$). If one wishes to allow diagonal steps as well, one could simply adjust the random_step method like this:

calc_random_walk_trajectory.py

```
def random_step(x, y):
    dx = np.random.choice([-1, 0, 1])
    dy = np.random.choice([-1, 0, 1])
    return x+dx, y+dy
```

b) Write the program for a SAW

To make the random walk self-avoiding, we simply have to check at each step whether or not the position of the next step has already been reached at an earlier time. If it has, we just return None instead of the trajectory. Outside of calc_random_walk_trajectory in the main function, we run a while loop, which repeats the function execution until a non-None value has been returned.

calc_random_walk_trajectory.py

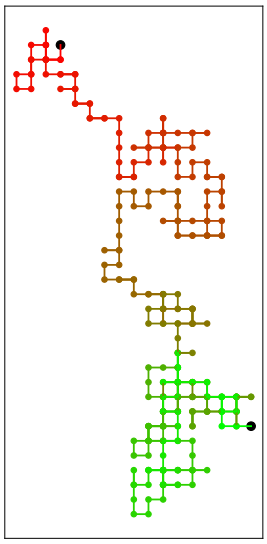
```python
import numpy as np


def main(nr_of_steps, self_avoiding=True, x=0, y=0):

    def random_step(x, y):
        direction = np.random.choice(['x', 'y'])
        if direction == 'x':
            dx = np.random.choice([-1, 1])
            dy = 0
        elif direction == 'y':
            dx = 0
            dy = np.random.choice([-1, 1])
        return x+dx, y+dy

    trajectory = []
    for i in range(nr_of_steps):
        x, y = random_step(x, y)

        if self_avoiding and (x, y) in trajectory:
            return None

        trajectory.append((x, y))

    return trajectory
```

main.py

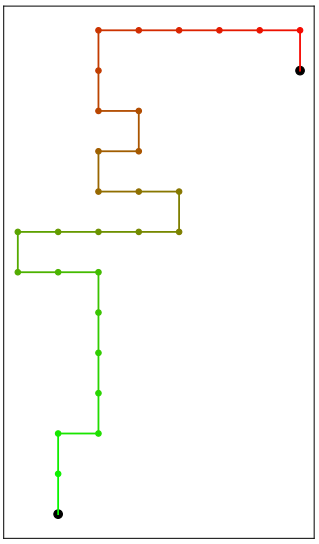
```python
from calc_random_walk_trajectory import main as calc_random_walk_trajectory


def main(nr_of_steps=30, self_avoiding=True):

    trajectory = calc_random_walk_trajectory(nr_of_steps, self_avoiding)
    if self_avoiding:
        while trajectory is None:
            trajectory = calc_random_walk_trajectory(
                nr_of_steps, self_avoiding
            )

    filename = 'SAW' if self_avoiding else 'RW'
    plot_trajectory(trajectory, nr_of_steps, filename)
```

c) Show some representative trajectories.

Without diagonal steps:

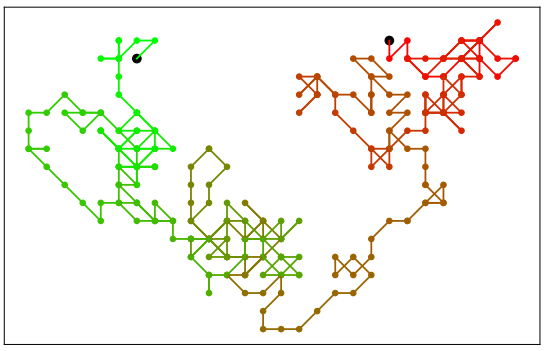


(a) normal random walk ($n = 400$)

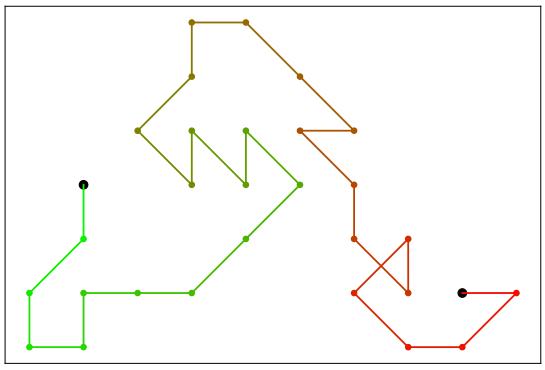


(b) self-avoiding random walk ($n = 30$)

With diagonal steps:



(a) normal random walk ($n = 400$)

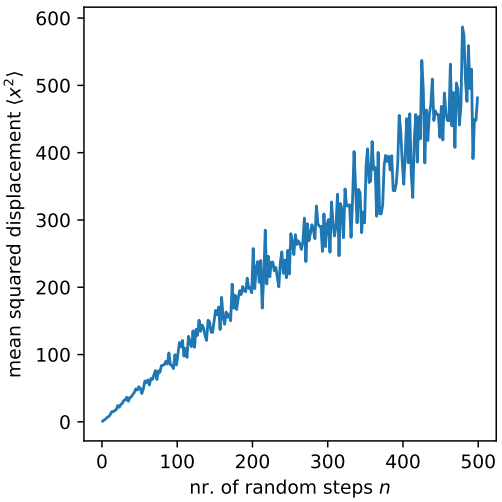


(b) self-avoiding random walk ($n = 30$)

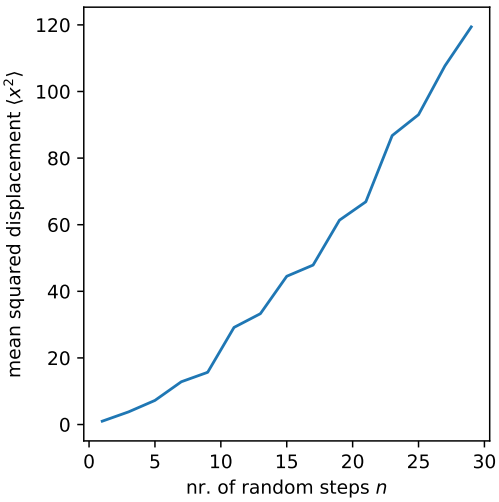
When the particle crosses its own path, the trajectory can look a bit messy. Therefore, we added a color coding from green (first step) to red (last step) to improve the visualization.

d) Measure the mean squared displacements (MSD) $\langle x^2(n)\rangle$ as a function of the number of steps n for both cases (the average is over different trajectories). If you plot the results in log-log-scale you can extract the exponents α, where $\langle x^2(n)\rangle \sim n^\alpha$, for the two cases. Compare and discuss.

To get the average over different trajectories, for each number of steps we calculate 100 trajectories (in this case, without allowing diagonal steps), then we calculate the squared displacements for each of the trajectories and take the mean. Then, we plot this in a linear fashion:

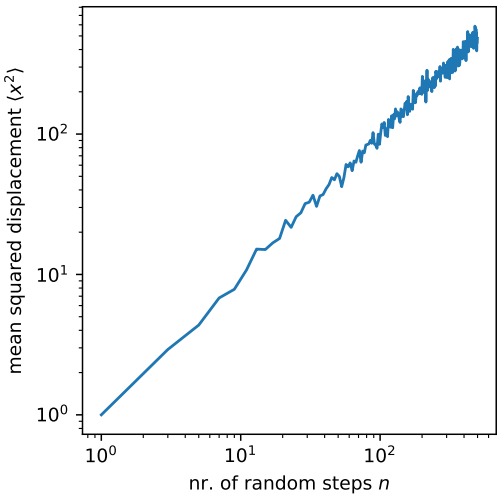


(a) linear plot for the normal RW

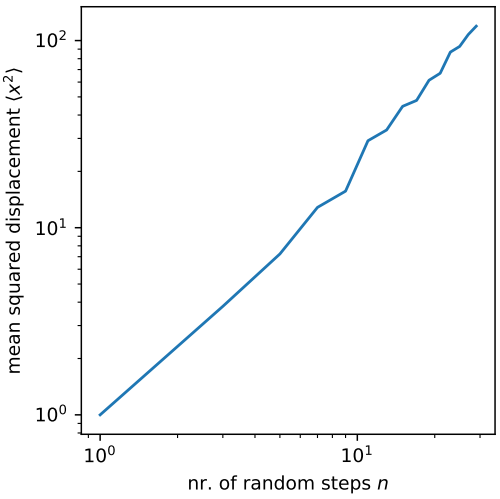


(b) linear plot for the SAW

And now as a log-log plot:



(a) log-log plot for the normal RW



(b) log-log plot for the SAW

As expected, we find $\alpha \approx 1$ for the normal random-walk. The self-avoiding random walk leads to an exponent of $\alpha > 1$, which also makes sense, since the movement of the particle is restricted in a way that makes it less likely to travel towards its point of origin.