Exercise Ex2

Student: Firstname Lastname Sciper: 000000

**Please use this template to submit your answers.**  
If you had to modify code from the notebook, please include the modified code in your submission either as screenshot or in a

\begin{lstlisting}[language=Python]  
\end{lstlisting}

environment.

You only need to include the code cells that you modified.

Note, that references to other parts of the documents aren’t resolved in this template and will show as ??. Check the text of the exercises on website for the reference

**Exercise 1**  
A quantum harmonic oscillator has energy levels: .

Write down the corresponding canonical partition function . Note that in this case, the partition function forms an infinite geometric series, and can be rewritten in terms of the limit of the series. From the result you obtain, derive the expectation value of the energy. Use the limit of the geometric series for , rather than the sum-based form.

Your answer here

**Exercise 2- Bonus**  
Show that, based on ([[thermodynamic\_properties]](#thermodynamic_properties)), the canonical partition function ([[classical\_probability]](#classical_probability)) is obtained from ([[canonical\_density\_matrix]](#canonical_density_matrix)) if the Hamiltonian admits an orthonormal eigenbasis $\left\{\ket{\Psi\_i}\right\}$ and the commutator vanishes. (There is no need to use the commutator itself, applying conditions that follow from vanishing commutators is sufficient.)

Your answer here

**Exercise 3 - Bonus**  
Show from ([[density\_operator]](#density_operator)) and ([[canonical\_density\_matrix]](#canonical_density_matrix)) that the are indeed equal to for pure states, given that there exists a common eigenbasis $\left\{\ket{\Psi\_i}\right\}$ to the total Hamiltonian and explain the origin of this restriction. Explain why the density operator and the expression for the expectation value of an observable assume a general form (([[canonical\_density\_matrix]](#canonical_density_matrix)) and ([[thermodynamic\_properties]](#thermodynamic_properties))), rather than being defined directly in terms of ([[canonical\_density\_matrix\_recast]](#canonical_density_matrix_recast)) and ([[classical\_probability]](#classical_probability)). (You may link your answer to the assumptions made in the previous question).

Your answer here

**Exercise 4**  
Derive the Boltzmann distribution, ([[boltzmann]](#boltzmann)), from ([[recast\_expectation\_value]](#recast_expectation_value)), using the expectation value of the particle number in state , , as observable. *Hint*: It may be useful to remember that the ensamble average in ([[recast\_expectation\_value]](#recast_expectation_value)) can be always expressed also as the probability of being in that state, times their value: .

Your answer here

**Exercise 5**  
Modify the code below to calculate the occupancy of each state within the harmonic oscillator system. Present the entire code file within your report and comment upon the main features using #. Consider different (**reduced**) temperatures, 0.5, 1, 2 and 3, and energy levels set to 10 (recall ). You will implement different StateOccupancy() functions, correspondng to different degeneracies. Note that in the degeneracy, is the index of the energy level, and its degeneracy is , where assumes values in this Exercise.

Your answer here

**Exercise 6**  
What do you note as the temperature is increased in terms of state occupancy? Does it make sense physically?

Your answer here

**Exercise 7**  
Compare different degeneracies (no degeneracy, , and ) for different reduced temperatures. What can you infer?

Your answer here

**Exercise 8**  
Modify your program such that the state occupancy and partition function are calculated for a linear rotor with moment of inertia (copy your code in the cell below, and adapt it for the linear rotor case. Present also this code within your report). Compare your results at different temperatures with the approximate result ([[approximateresult]](#approximateresult)).

*Hint*: considering more than four reduced temperatures can be a good idea, you can easily obtain evenly spaced numbers over a specified interval using the numpy function np.linspace(start, stop, number\_of\_points). Note that if the higher temperatures considered are far from the highest temperature which was considered so far, more energy levels may become accessible to the system, and you should keep them into account.

Is this an approximated result for which temperature limit?

using

Note that the energy levels of a linear rotor are:

with and the degeneracy of level J is .

Your answer here