

PHYS10008 - extended assignment

The Boltzmann probability distribution.

The Boltzmann distribution is a very important concept in physics as it relates the probability of the occupation of a particle state, with energy E_i to the separation of the energy levels in the system, their degeneracy (the number of distinguishable particle states with the same energy) and the temperature of the system. This is a purely statistical process that you will demonstrate in this assignment with the simple system below.

In this assignment you will consider a system that contains a total of N particles, each of which may be in an energy state E_i where $i = 0, 1, 2, \dots$ and E_0 is the ground state (lowest energy state). We will also assume that there is only one state for each energy E_i (that is, the system is non-degenerate). The total energy of the system, that is the sum of the energies of all the particles in the system, will remain constant and at equilibrium, will be related to the temperature of the system.

At the equivalent of a temperature $T = 0$ in our system, all of the particles must be in the ground state E_0 as there is no energy that can be transferred to or from other particles in the system.

Now consider a system where we have somehow managed to put all of the particles into the next energy level E_1 so that we now have N particles with energy E_1 and no particles with energy E_0 . The total energy of the system is now $N \times E_1$ and is $N \times (E_1 - E_0)$ higher than when it was in the ground state.

As you know from your thermal physics course the chances that our system would be in this particular microstate (with all particles with energy E_1) is highly unlikely. So, if we allow the particles to continuously and randomly exchange energy with each other (thermalise) we expect that after sufficient time, they will show a wide variation in their energies. The distribution of these energies should be in accordance with the Boltzmann distribution where we have the mean occupation number for a state of energy E_n , for a system of N particles is given by

$$n(E_n) = N \exp\left(-\frac{E_n}{kT}\right) \dots (1)$$

where kT is related to the mean energy of the particles in the system. Hence we can consider the mean energy of the particles $\langle E \rangle$ as related to the temperature of our system. This distribution should hold for any system of N particles with energy states E_0, E_1, E_2, \dots

A simple system to model

In this problem you are going to consider a simple system where you have N particles and the energy levels of the system are given by

$$E_i = E_0 + i \times \Delta E$$

that is, all the energy levels are equally spaced and non-degenerate. For computational simplicity we will also assume that the lowest energy state is $E_0 = 1$ in arbitrary units and that $\Delta E = E_0$ (i.e. the energy levels are spaced E_0 apart).

The aim of your programme is to show, that by purely random (statistical) exchanges of energy (of ΔE) between the particles in our system, we do indeed get the Boltzmann distribution (1) for the energies of the particles.

The method.

In your programme you should 'create' the N particles for your system as an array that contains the energy of each individual particle. You will need to supply an initial energy (e.g. E_0, E_1, \dots) to each to each particle. For simplicity (the starting distribution should not affect the result) you should give the particles the same energy ($E = E_1$ or E_2 or. . .). In order for the system to reach equilibrium we need to introduce a method by which the particles can exchange energy with each other. The method you will use is very straight forward.

You will pick two particles, a and b at random and you will exchange energy between them as follows

$$E_a = E_a - \Delta E$$

$$E_b = E_b + \Delta E$$

The only other criterion you need to apply is that no energy exchange will take place if $E_a = E_0$, i.e. if a is already in the ground state it cannot 'give' up any energy.

You will note that in this process the total energy of the system does not change and the mean energy per particle $\langle E \rangle$ remains constant - it is only the way the energy in the system is distributed that changes. If we run this process enough times we expect the system to reach 'thermal' equilibrium where the energy distribution of the particles should follow the Boltzmann distribution (1).

Once you have completed enough steps you can find the number of particles in your system at each energy as a histogram (i.e. $n(E_0), n(E_1), n(E_2), \dots$) and compare this to the Boltzmann distribution you expect (1).

Steps in writing your programme

The aim of this exercise is to demonstrate that you can write a complete programme by yourself. Here are some tips about what you need to do

- i) Make sure that you import the numpy and matplotlib libraries. You may use the random number generator in numpy or if you prefer you can use the random number generator library (random) directly.
- ii) at the beginning of your programme assign a variable to the number of particles you will include in your simulation and then create a numpy array in which you will store the energy of each particle. For the assignment we will set N as 500.
- iii) set the ground state energy E_0 to 1 (in arbitrary units) in a variable, for example $E0=1$.
- iv) assign a variable to the value of ΔE you are going to use. In the assignment you should set $\Delta E = E_0$ (i.e. the energy levels are separated by E_0).
- v) setup the array of energies you have created so that all the particles have the same starting energy (i.e. $E_0 + n\Delta E$ where n is an integer). This step is equivalent to setting the 'temperature' of the system. If you set $n = 0$, all particles are in the ground state and nothing will happen. In the assignment you should set $n = 1$, i.e. all the particles are at energy $E_1 = E_0 + \Delta E$
- vi) assign a variable to the number of 'steps' you are going to use to and start with say 10 steps while testing. In the assignment you should set the number of steps to be 10000.
- vii) This is the crucial part of your programme. In this part you will need to do the random 'exchange of energy' between the particles. For each step in your programme do the following,
 - use the random number generator to pick a particle
 - check to see if the particle chosen is in the ground state (i.e. has energy E_0). If it is at E_0 skip the step.
 - if the particle has an energy $E > E_0$ then randomly choose another particle and then subtract ΔE from the first particle and add ΔE to the second particle.
 - repeat this process for the number of steps you have chosen.

viii) Note, at the end of this sequence the total energy of the system (i.e. the sum of all the energies should remain unchanged. You may like to check this.

ix) At the end, you should plot the energy of each particle (i.e. the values in your energy array), as a function of index number, you should observe that the particles now show a distribution in energy.

x) Finally create a histogram showing Energy (E_n) on the x-axis (i.e. $E_0, E_0 + \Delta E, E_0 + 2\Delta E, \dots$ vs. the number of particles with that energy. If we are correct and we have used enough steps, this should show our expected Boltzmann distribution (1). Tip. To decide the maximum energy value you need to display on the x-axis you can search for the maximum energy in your energy array.

Completing the assignment.

You will need to complete the assignment as an nbgrader assignment in notable. The assignment will be scaffolded into different code cells but it needs to run as a single programme. You should make sure it runs with the values of the variables specified. The assignment will be manually graded but you should make sure your completed assignment runs without error. As you are using a random number generator you will expect slightly different results each time, but your conclusions should be the same. You will also be asked to write a short section describing the results you obtained and your conclusions.

I recommend to start with, that you write your programme in a separate jupyter notebook, (or in this notebook). Once you have a working version you can then complete the nbgrader assignment.

Whatever you do DO NOT DELETE or try to upload a new copy of the assignment. If you do, nbgrader will not recognise the assignment and you will get a mark of zero. It is not difficult to copy and paste your code from your own notebook to the assignment.