

# Computer simulation: Computational Methods, assignment 2

Name: Rohan Kadam

Student number: 20334092

## Introduction:

The premise of this assignment is to solve further problems in condensed matter physics. The main components of this assignment are solved using the [numpy] and [scipy] libraries on python, however external libraries may also be installed for assistance. Using the [integrate] and [root] numerical functions from our imported packages we were able to compute the average energy density of a given 3-D electron gas system under a specific set of parameters. Another task was carried out in which the eigenvalues for a 1-D tight-binding model under selected 'alpha' values were found, as well as their corresponding Density of States. Finally for a different specified tight-binding model we found the inverse participation ratio (IPR) and plotted it against the corresponding energy eigenvalue. The lowest and highest IPR was found and the corresponding spatial probability distribution was plotted for each state. Sample pseudocode of how each task was carried out will be provided throughout the report in the form [].

## Exercise 1)

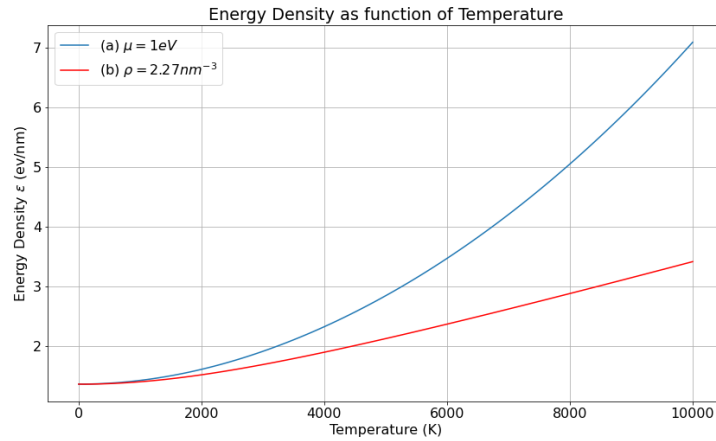
We wish to analyze the average energy density ' $\epsilon$ ' of a large system of electrons which is in contact with an external reservoir at temperature  $T$  and chemical potential  $\mu$ . We are given the integrals for the average electron density ' $p$ ' and the average energy density ' $\epsilon$ ' to assist us in analyzing the system. These integrals are shown below. The expression for the 3D electron gas which is described by the density of states, ' $d(w)$ ' given in per unit energy per unit volume is described below as well as the Fermi-Dirac distribution  $f(w)$ .

$$p = \int_0^{\infty} dw d(w) f(w) \qquad \epsilon = \int_0^{\infty} dw w d(w) f(w)$$
$$d(w) = \frac{\sqrt{w} m_e^{3/2}}{h^3 \pi^2 \sqrt{2}} \qquad f(w) = \frac{1}{1 + e^{\frac{(w-\mu)}{k_B T}}}$$

An appropriate unit system was chosen for this task. This included measuring lengths in nanometers, masses in electron mass, and energies in electron volts. For example the Boltzmann constant was taken as ' $8.62 \times 10^{-5} eV$ ' and  $\hbar$  as  $0.276 \text{ nm} \sqrt{eV m_e}$ . For part (a) of this exercise we plot the average energy density of the 3-D electron gas as a function of temperature for a set range, with a set chemical potential of  $\mu = 1 \text{ eV}$ . The set of temperatures for which we had to compute over was given by the range  $T \in [0 \text{ K}, 10^4 \text{ K}]$ . This range of temperatures allows for a wide and thorough analysis of the results. We begin by defining all the constants and functions mentioned above. An example of this can be seen with defining the Fermi-Dirac distribution '[def FermiDirac(w,T,mu):]'. '[fw = (exp((w-mu)/(KB\*T))+1)\*\*(-1)]'. The integrate.quad function from the scipy package was used to integrate both the average electron density ' $p$ ' and the average energy density ' $\epsilon$ '. The graph was plotted for the average energy density as a function of a given range of temperatures.

However for part (b) of exercise 1 we need to plot a similar graph, but this time at a fixed electron density  $p = 2.27 \text{ nm}^{-3}$ . From our equations we can deduce that this will mean that the chemical potential  $\mu$  is a variable and will need to be solved for the same temperature range  $T$  as part (a). To

do this we first needed to solve for the roots of the equation  $p - 2.27 = 0$ , for each temperature  $T$ , with the equation for  $p$  given above. We then solve for the roots of the equation by using another function from the scipy package which is `[optimize.fsolve]`. This is done like so `[root = optimize.fsolve(f, 0)]`. We then plot the another average energy density against the range of given temperature values. Both lines are given on the same graph as shown below.



Above is figure 1 which is the graphical output from part (a) and (b) of exercise 1.

We observe straight away that the energy density increases with the temperature range provided, with both increasing at a similar rate till the 1000K mark. The energy density increases at a quicker rate when  $\mu = 1\text{eV}$ , compared to when  $\rho = 2.27 \text{ nm}^{-3}$ . This is quickly understandable and an acceptable result by a quick observation of the formulae. It was noted that a runtime warning occurred because of an overflow in this exercise. From our calculating the underflow and overflow of my laptop, it was deduced that this error most likely occurs when evaluating integrals at infinity with the exponential in the denominator.

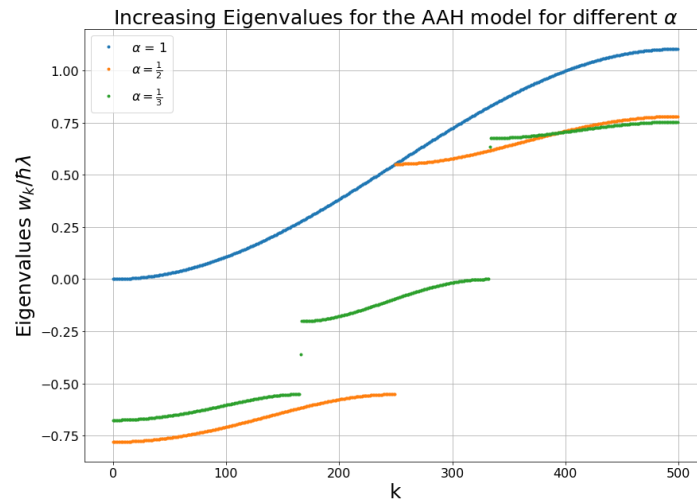
## Exercise 2)

In this exercise we are given a tight-binding model named the Aubrey-Andre-Harper (AAH) model defined by a Hamiltonian as shown below.

$$\hat{H} = \begin{pmatrix} \epsilon_0 & -\hbar\lambda & 0 & 0 & \dots \\ -\hbar\lambda & \epsilon_1 & -\hbar\lambda & 0 & \dots \\ 0 & -\hbar\lambda & \epsilon_2 & -\hbar\lambda & \dots \\ 0 & 0 & -\hbar\lambda & \epsilon_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Before proceeding further we must remind ourselves of what a tight-binding model is. Tight-binding models are effectively used to describe the motion of electrons in a solid; in our case a lattice. The onsite energy for our lattice is given as,  $\epsilon_l = u \cos(2\pi\alpha l)$ . Our tight-binding model has a tunneling rate  $\lambda$  between the nearest neighbor lattice sites. Tunneling is a quantum mechanical phenomenon when a particle is able to penetrate through a potential energy barrier that is higher in energy than the particle's kinetic energy. We need to find the eigenvalues for given values of  $\alpha$ , which are 1, 1/2, and 1/3. In our computation the number of sites  $L$  was given as 500.  $U$  was given as  $2\hbar\lambda$ . To tackle this problem we define all constants, and the onsite energy equation which is done by `[def el(alpha,l):]` and `[return u*cos(2*pi*alpha*l)]`. We now need to define the Hamiltonian for the AAH model. This is done like so, `[def mAAH(alpha,L,tun):]`, `[values = -tun*h*np.ones(L-1)]`, and `[H =`

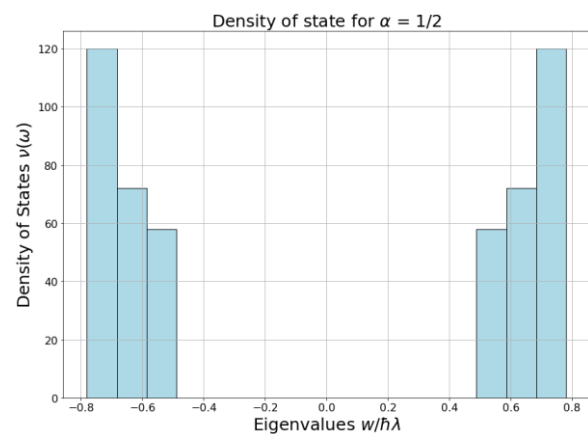
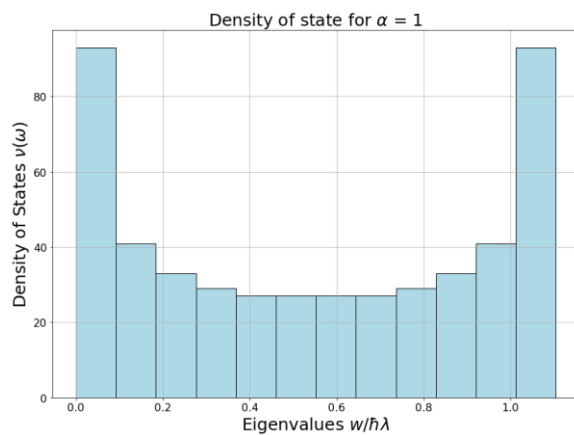
`np.diag(values,1) + np.diag(values,-1)]`. The eigenvalues for the Hamiltonian were found using the `[linalg.eig(H)]` function from numpy. The eigenvalues found were then plotted in increasing order.

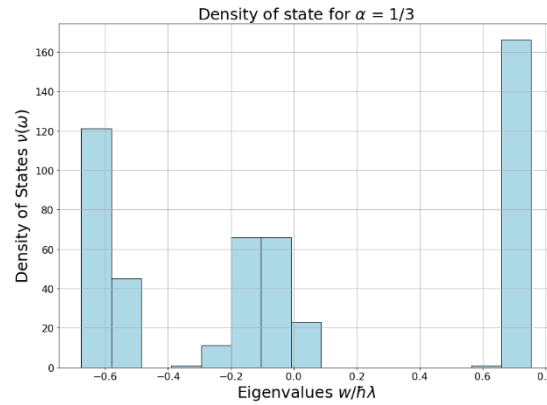


Above is figure 2 which shows the Eigenvalues for the Hamiltonian of the AAH model in increasing order for  $\alpha = 1$ ,  $\alpha = 1/2$ , and  $\alpha = 1/3$ .

We notice splits for the eigenvalues for  $\alpha = 1/2$  and  $\alpha = 1/3$  but not  $\alpha = 1$ . We must wonder what could lead to this discrepancy. After analyzing the equation for the onsite energy equation it is concluded these 'splits' are due to the cosine and the smaller the value for  $\alpha$  is the more 'splits' will occur. The eigenvalues are split into batches or groups and it is observed the number of groups is dependent on  $\alpha$ .

For the next part of this exercise we need to plot the unnormalised density of states (DOS)  $\nu(w)$ . This is defined as the number of eigenstates within an energy interval,  $w \in [w - \frac{\delta w}{2}, w + \frac{\delta w}{2}]$ . The tunneling is once again defined a `[tun = 1.]`. The number of sites  $L$  was once again 500. We define  $\delta w = \hbar\lambda/10$ , with  $u = 2\hbar\lambda$  once again. The density of states was plotted as a histogram for each  $\alpha$  specified and the following code was crucial. `'[H = matrixH(a,L,tun)], '[w,psi] = np.linalg.eig(H)'`, `[plt.hist(w, bins=n_bins)]`.





Above are figures 3,4 and 5 show the Density of States  $\nu(w)$ , for a specified energy interval for  $\alpha = 1$ ,  $\alpha = 1/2$ , and  $\alpha = 1/3$  respectively.

We find the same ‘splitting’ effect occurring in the histograms as we did when plotting the eigenvalues for the Hamiltonian. Showing the value for  $\alpha$  having an effect on the Density of states, with the smaller  $\alpha$ 's yielding more ‘groups’.

### Exercise 3)

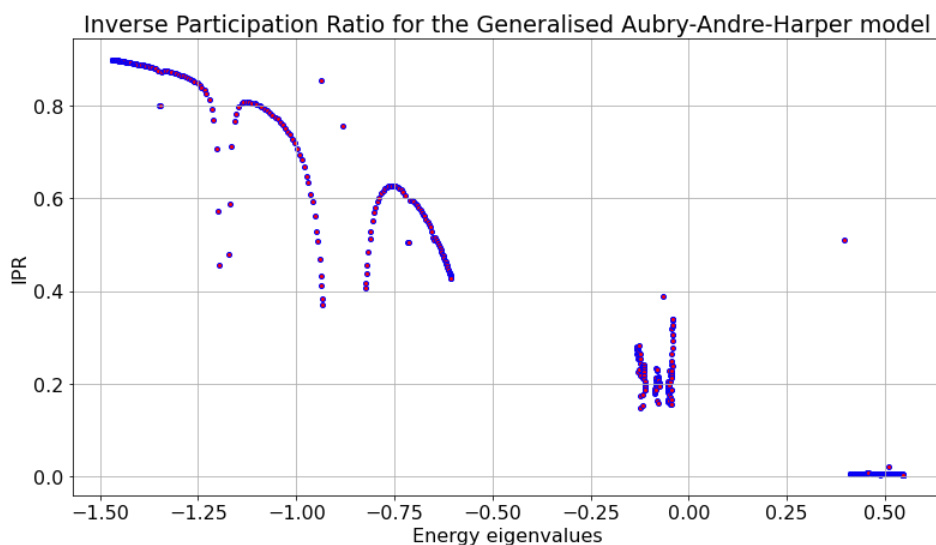
In this exercise we will be dealing with generalized version of the Aubrey-Andre-Harper (GAAH) model which is formulated the same as the AAH model apart from a change in the onsite energy.

The new onsite energy is defined as  $e_l = \frac{u \cos(2\pi\alpha l)}{1 + \beta \cos(2\pi\alpha l)}$

For part (a) of this exercise we want to find the inverse participation ratio (IPR) for each eigenstate of the GAAH model provided. For a normalized eigenstate  $|\psi\rangle = \sum_l c_l |l\rangle$

We define the IPR as  $I = \sum_l |c_l|^4$

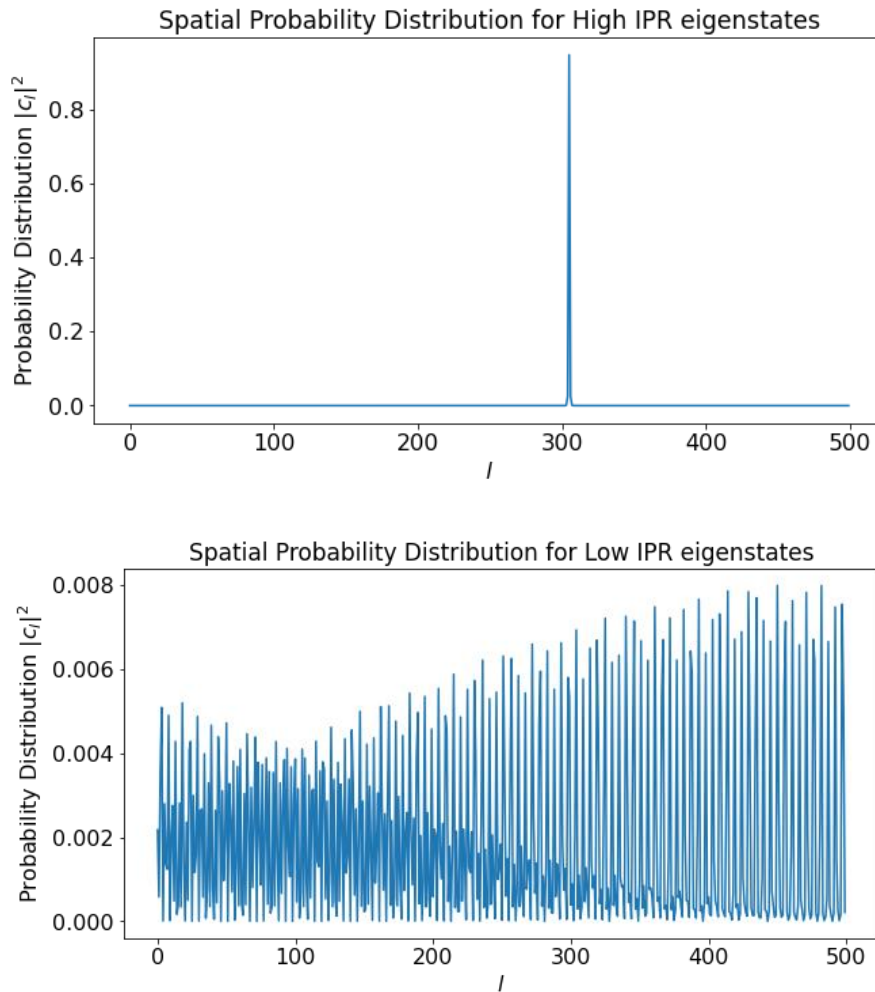
First we need to define our variables in the system once more, with  $\alpha = \frac{1}{2}(1 + \sqrt{5})$  and  $\beta = 0.6$ ,  $u = 2\hbar\lambda$  and the number of sites  $L$  as 500. We also define our new onsite energy as well as our Hamiltonian like in exercise 2. We define the IPR by `[def IPR(x):]` and `[return sum(np.abs(x)**4)]`. We then need to slice the array by `[IPR = [IPR(psi[:,i]) for i in range(len(psi))]]`.



*Above is figure 6 which is the Inverse Participation Ratio for the Generalized Aubry-Andre-Harper model for given  $\alpha$  and  $\beta$ .*

From figure 6 we can infer that high eigenvalues have low IPR's and low eigenvalues have high IPR's.

For part (b) we need to plot the spatial probability distribution  $|c_l|^2$  for an eigenstate with a low IPR and an eigenstate with a high IPR. To do this we will first find the maximum and minimum IPR values and take the column values from the  $\Psi$  (psi) matrix. We do this by  $[x = \max(\text{IPR})]$ ,  $[y = \min(\text{IPR})]$ ,  $[\text{High\_index} = \text{IPR.index}(x)]$ ,  $[\text{Low\_index} = \text{IPR.index}(y)]$ ,  $[\text{High\_IPR} = \text{psi[:,high\_index]}]$ ,  $[\text{Low\_IPR} = \text{psi[:,low\_index]}]$ ,  $[\text{order} = \text{np.arange}(0,L,1)]$ . We then plot on separate graphs the spatial probability distributions.



*Above are figures 7 and 8 which are the spatial probability distributions for the high and low IPR eigenstates respectively.*

From the high IPR plot we are immediately able to tell that the states are localized to a particular site as the distinctive peak observed is close to one. It is extremely likely to find an electron along this site, and conversely extremely unlikely to find an electron anywhere else. From the low IPR graph we can tell that the states are extended, which means there are no distinctive peaks like in the high IPR graph. The low IPR eigenstate is highly unlocalised and by observing the graph it is approximately equally likely that the electron could be anywhere in the system.

## Conclusion:

A deeper understanding of condensed matter physics was gained as well as more fluency in the numpy and scipy libraries. New techniques were solidified in python such as integration and solving for roots of functions. The average energy density of a 3D electron gas system, and different tight-binding models were successfully investigated and analyzed using python.