PH4419/PAP723: Computational Physics

Assignment 2

INSTRUCTIONS

- *Submission:* Assignment is due: **22nd March 2019**.

- *Submission format:* All submission are to be done through NTULearn. There are 2 (two) questions in this assignment. Answer each question in a separate **python .py** file. Label your files using the following convention:

  Matric Number_Assignment 2_Question X.py

  For example: U1234567A_Assignment 2_Question 0.py

  Each file should begin with the lines

  ```
  from scipy import *
  import matplotlib.pyplot as plt
  ```

  Any other Python libraries used must be listed directly below these two lines.

- *Grading:* For full credit, code must follow good programming style. Key code blocks must be clearly commented. Function names must be properly named for ease of understanding your codes. The program structure should be modular; avoid unnecessary code duplication, and group numerical constant definitions neatly together. The output of your programs should be clear. Each generated plot should be well labeled, and if multiple curves are shown, they should be well-labeled.

- *PAP723:* Parts of questions marked with a $(*)$ are for students taking the course PAP723, they must be attempted to receive credit. Undergraduate students who can produce a working code with correct output for these parts will receive 1 (one) bonus mark per question.

- *Documentation:* Whenever the assignment mentions a Scipy function you might need, consult the Scipy/Numpy online documentation to learn what the function does and how to use it.

- *Techniques:* The core techniques you will need to apply in this problem set are: (i) Finite Difference Method for PDE and (ii) Eigenvalue Problem.

00. Incompressible Flow of Fluids [8 marks]

For an incompressible and irrotational flow, the velocity field of the fluid can be found by solving the Laplace equation of the stream function $\psi(x, y)$,

$$\nabla^2 \psi = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{1}$$

Using the Taylor approximation in a discretized uniform ($\Delta x = \Delta y$) Cartesian grid, the second derivatives can be approximated by:

$$\psi_{i,j} = \frac{1}{4}\Big(\psi_{i-1,j} + \psi_{i+1,j} + \psi_{i,j-1} + \psi_{i,j+1}\Big) \tag{2}$$

Using this finite difference approximation for $\psi_{i,j}$. update the value for $\psi_{i,j}$ starting from the initial guess (usually $\psi_{i,j} = 0$, except for the boundary values). If n represents the update loop, we will use the values of ψ^n to calculate new values of ψ^{n+1}, meaning

$$\psi^{n+1}_{i,j} = \frac{1}{4}\Big(\psi^n_{i-1,j} + \psi^n_{i+1,j} + \psi^n_{i,j-1} + \psi^n_{i,j+1}\Big) \tag{3}$$

The numerical method is said to be stable, if the value of ψ does not change considerable after each update,

$$|\psi^{n+1}_{i,j} - \psi^n_{i,j}| < \epsilon \tag{4}$$

where ϵ is a small enough threshold that is determined by the user. We will also set a maximum number of iterations (i.e and upper bound for n) so that the loop stops if the solution takes very long to converge. Consider the domain, where the bottom and top boundaries are impermeable walls, to the left is an inlet and the outlet is to the right as shown in Figure 0.

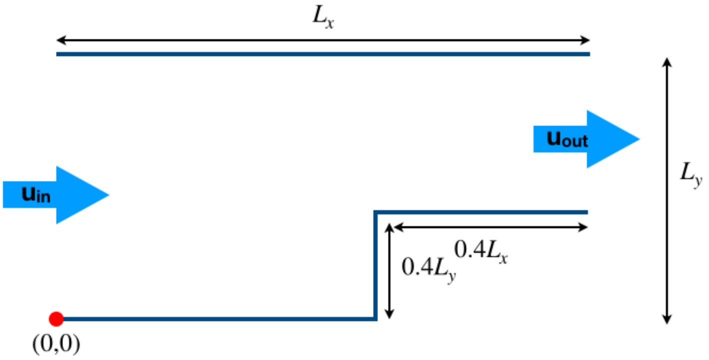

Figure 0: Dimensions of a channel with impermeable walls

Fluid flows in and out satisfying the following boundary conditions:

- Bottom Boundary: ψ is a constant.
- Top Boundary: ψ is a (different) constant.
- Left Boundary: ψ increases linearly from the bottom to the top boundary, satisfying $u_{in}y$

- Right Boundary: ψ increases linearly from the bottom to the top boundary satisfying the previous 3 boundary conditions.

where y is the vertical displacement from the origin. In our example we set $u_{in} \equiv u_{flow}$.

The velocity of the flow $\vec{\nu} = (u, v)$ can be found by performing the following calculation:

$$u = \frac{\partial \psi}{\partial y} \qquad\qquad v = -\frac{\partial \psi}{\partial x} \tag{5}$$

Write a function that solves the Laplace equation for the stream function and show the velocity of the flow.

```
def laplace_solver(epsilon, nmax, u_flow):
```

Input	
`epsilon`	A real number indicating the minimum allowed error before the system is deemed to have converged.
`nmax`	An integer indicating the maximum number of iterations before the algorithm stops as the system is deemed to not have converge to the desired threshold `epsilon`.
`u_flow`	A constant real number describing the flow as given by the boundary conditions.

Set `epsilon` $= 10^{-3}$, `nmax` $= 10^4$ and `u_flow` $= 5.0$. There are no output values for this function. Instead, the function should plot the flow of fluid inside the domain, indicating clearly the velocity of the fluid and print the number of iterations it took. Describe briefly how you could solve for the steady state solution with less iterations.

01. Band Gaps [12+5 marks]

(a) **Nearly Free Electron Model.** Consider a 1D lattice with lattice spacing $a = 5$ Å. In this problem, we will examine the effects of the lattice potential on the band gap of an electron in a periodic 1D lattice. Suppose we have the lattice potential

$$V(x) = 2V_1 \cos\left(\frac{2\pi x}{a}\right) \tag{6}$$

where we will set $V_1 = 0.2$ eV The analytic solution to any wave vector $\vec{k}$ in the first Brillouin Zone (1BZ) can be written as a superposition of plane waves as follows

$$\begin{aligned}|\psi_{\vec{k}}\rangle &= \sum_{n=-\infty}^{\infty} C_{\vec{k}}(G_n)|\phi_{\vec{k}+G_n}\rangle \\ &= \sum_{n=-\infty}^{\infty} C_{\vec{k}}(G_n)\mathcal{A}e^{i(\vec{k}+G_n)x}\end{aligned}$$

where $G_n = \frac{2\pi n}{a}$. A good approximation is to take a finite number of sums, for some sufficiently large N i.e

$$|\psi_{\vec{k}}\rangle = \sum_{n=-N}^{N} C_{\vec{k}}(G_n)|\phi_{\vec{k}+G_n}\rangle \tag{7}$$

Hence, we can now put this into the Schrödinger Equation to get an eigenvalue problem (left as an exercise) described by the following eigen-equation

$$\mathcal{V}\, C(G_n) = E(\vec{k})\, C(G_n) \tag{8}$$

satisfying

$$\mathcal{V} = V_1 \mathbb{I}_{-1} + \epsilon_n \mathbb{I}_0 + V_1 \mathbb{I}_1 \tag{9}$$

where we have defined $\epsilon_n\big(\vec{k'}\big) = \frac{\hbar^2|\vec{k'}|^2}{2m}$, where $\vec{k'} = \vec{k} + G_n$ for $n \in [-N, \ldots, N]$ and $\mathbb{I}_m$ is an $(2N+1) \times (2N+1)$ matrix where the m-th diagonal off the main diagonal is filled with 1 and the other entries 0 (i.e $\mathbb{I}_0$ is the identity matrix). For each of the following parts, take $N = 10$ (i.e $\mathcal{V}$ is a 21×21 matrix). Write a function `bandgap_plot(N,V_1,V_2)` that does the following:

(i) **[3 marks]** For each value of k in 1BZ, i.e $k \in [-\pi/a, \pi/a]$, solve for the smallest 3 eigenvalues of the eigen-equation above. This will give you $E_1(k)$, $E_2(k)$ and $E_3(k)$ of the 3 lowest bands. Plot a graph of these energies versus k for the 1BZ. What is the magnitude (in eV) of the band gap(s). Comment on whether the numerical result for the band gap(s) fits the theoretical result you are expecting.

(ii) **[2 marks]** Now, assume that the potential has an extra term

$$V(x) = 2V_1 \cos\left(\frac{2\pi x}{a}\right) + 2V_2 \cos\left(\frac{4\pi x}{a}\right) \tag{10}$$

where $V_2 = 0.4$ eV. For each value of k in 1BZ, i.e $k \in [-\pi/a, \pi/a]$, solve for the smallest 3 eigenvalues of the eigen-equation above. This will give you $E_1(k)$, $E_2(k)$ and $E_3(k)$ of the 3 lowest bands. Plot a graph of these energies versus k for the 1BZ. What is the magnitude (in eV) of the band gap(s). Comment on whether the numerical result for the band gap(s) fits the theoretical result you are expecting.

(iii) **[2 marks]** We change the lattice potential such that V_1 is no longer small, i.e

$$V(x) = 2V_1 \cos\left(\frac{2\pi x}{a}\right) \tag{11}$$

where $V_1 = 4.0$ eV. Assume $N = 10$. For each value of k in 1BZ, i.e $k \in [-\pi/a, \pi/a]$, solve for the smallest 3 eigenvalues of the eigen-equation above. This will give you $E_1(k)$, $E_2(k)$ and $E_3(k)$ of the 3 lowest bands. Plot a graph of these energies versus k for the 1BZ. What is the magnitude (in eV) of the band gap(s). Comment on the difference between the plot you obtain in this part and the plot you obtained in part (a), what does this mean physically?

For each of the parts above, **do not copy and paste** blocks of code. You should code in modules and define functions, with appropriate inputs and outputs, as required. Hint: You should define a separate function to generate the matrix $\mathcal{V}$.

(b) **Graphene.** A single layer of graphene consists carbon atoms in the form of a honeycomb lattice. There are four valence electrons (two 2s and two 2p electrons). Three of those participate in the chemical bonding in this part, we will examine the bands formed by the one remaining electron. We assume a tight-binding model in which the electron hops between neighbouring atoms, examining the band type π.

The honeycomb lattice (see Fig 1(a)) can be seen as a triangular lattice with a basis of 2 atoms in each unit cell. The lattice vectors are

$$\vec{a}_1 = \frac{a}{2}(3, \sqrt{3}) \qquad \vec{a}_2 = \frac{a}{2}(3, -\sqrt{3}) \tag{12}$$

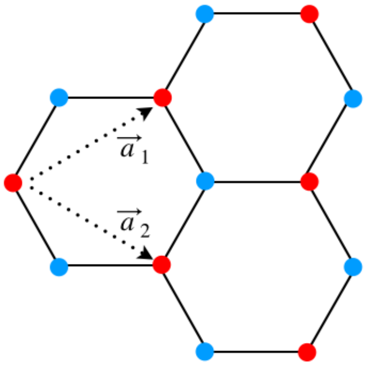

Figure 1(a): The honeycomb lattice of triangular lattice with a basis of 2 atoms.

where $a = 1.42$ Å is the nearest-neighbour C-C spacing. The reciprocal lattice vectors (see Fig 1(b)) are given by

$$\vec{b}_i = 2\pi \frac{\hat{z} \times \vec{a}_j}{|\vec{a}_i \times \vec{a}_j|} \tag{13}$$

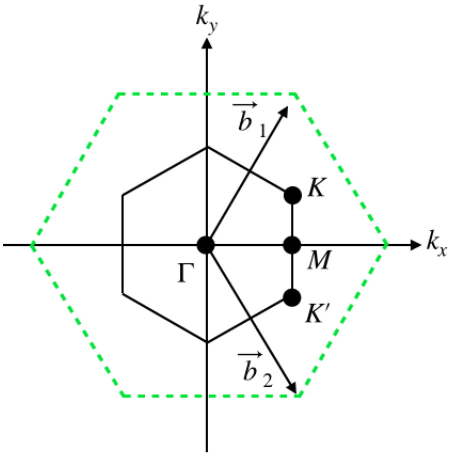

Figure 1(b): Reciprocal lattice and 1BZ of the honeycomb lattice.

where the dashed-lines represents the reciprocal lattice and the solid lines represent the first Bouillon zone. The points Γ, K, K' and M are points of high symmetry.

The reciprocal lattice vectors are given by

$$\vec{b}_1 = \frac{2\pi}{3a}(1, \sqrt{3}) \qquad\qquad \vec{b}_2 = \frac{2\pi}{3a}(1, -\sqrt{3}) \tag{14}$$

You check that the energies of the conduction and valence band as a function of $\vec{k}$ is given by

$$E(\vec{k}) = \pm\gamma\sqrt{1 + 4\cos\left(\frac{\sqrt{3}a}{2}k_y\right)\left[\cos\left(\frac{3a}{2}k_x\right) + \cos\left(\frac{\sqrt{3}a}{2}k_y\right)\right]} \tag{15}$$

(i) [**5 marks**] Write a function `graphene_plot(gamma=3.0,a)` that gives the following outputs in separate plots, taking $\gamma \approx 3.0$.

- $E(\vec{k})$ versus k_x, fixing $k_y = 0$.
- $E(\vec{k})$ versus k_y, fixing $k_x = 0$.
- $E(\vec{k})$ versus $\vec{k}$ along the trajectory $\Gamma \to M \to K \to \Gamma$.

(ii) [**5 marks**] ($*$) Write a function that plots $E(\vec{k})$ versus $\vec{k}$. Use your plots to briefly explain the properties of Graphene. [Hint: You will need to plot an energy surface]