Practical: From Tight Binding to the SSH model

Updated by Siyu (November 7, 2022)

Preliminaries

In this practical session, you will write a program to explore the Su-Schrieffer-Heeger (SSH) model, a polyacetylene described by tight binding theory. You will learn how to solve the model numerically and see how topology plays a role in this system.

You are free to use any programming language that you like, but I recommend using a high-level programming language, such as `Python`, `Matlab`, or `Julia`, because they have many useful inbuilt functions for scientific computing and for plotting the results. Also, using object-oriented programming to organise data involved in this project would be a good idea.

Section 1: Implementation of tight binding theory

Recall from the first lecture that the key quantities to build a tight binding model of a crystal are the on-site and hopping energies, i.e. the Hamiltonian matrix elements $H_{ij}(\mathbf{R})$.

1. Consider a 1D crystal placed along the x axis with N localised atomic orbitals per unit cell. Write a program that reads on-site and hopping energies of the system, then store them in a proper way. Make sure that you are able to access all non-zero values of $H_{ij}(\mathbf{R})$.

   Helpful tips: To avoid hard-coding of the input parameters, see the demo input file: *hopping.info*.

2. Implement the formula

$$H_{ij}^{\mathbf{k}} = \sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} H_{ij}(\mathbf{R}) \tag{1}$$

   to obtain the Hamiltonian matrix in reciprocal space.

3. For simplicity, we assume that in the each cell there are only $N = 2$ atomic orbitals. The on-site and hopping energies are given by $H_{11}(\mathbf{0}) = H_{22}(\mathbf{0}) = 0.1$, $H_{12}(\mathbf{0}) = H_{21}^{\dagger}(\mathbf{0}) = -0.4$, $H_{12}(\pm\mathbf{1}) = H_{21}^{\dagger}(\pm\mathbf{1}) = 0.7$. Output $\mathcal{H}^{\mathbf{k}}$ at $\mathbf{k} = -\pi, 0, \pi$ to test your program and verify that the results are in agreement with what you expect from Eq. 1. Make sure the Hamiltonian is Hermitian.

4. Diagonalise $\mathcal{H}^{\mathbf{k}}$ for $\mathbf{k} \in [-\pi, \pi]$ to extract the eigenvalues at each $\mathbf{k}$ point. Plot the band structure.

5. (Optional) Extend your code to be able to handle the 2D case.

Section 2: Investigation of the SSH model

1. Recall the meaning of the intracell and intercell hopping energies v and w in the SSH model. (How do they relate to $H_{ij}(\mathbf{R})$?) Reproduce the band structure of the SSH model with $v = 0.6$ and $w = 1.0$ using your tight binding code obtained in the previous section.

2. Implement the discrete formula of the Berry phase

$$\phi = -\Im \ln \left[\langle u_0 | u_1 \rangle \langle u_1 | u_2 \rangle \cdots \langle u_{N-1} | u_0 \rangle \right] \tag{2}$$

   in your program. What is the Berry phase of the SSH model with $v = 0.6$ and $w = 1.0$? What about $v = 1.0$ and $w = 0.6$?

   Helpful tips: We are solving the Schrödinger equation by numerically diagonalising $\mathcal{H}^{\mathbf{k}}$,

$$\mathcal{H}^{\mathbf{k}} \, \mathcal{C}_{n\mathbf{k}} = E_{n\mathbf{k}} \, \mathcal{C}_{n\mathbf{k}}. \tag{3}$$

   The column vector $\mathcal{C}_{n\mathbf{k}}$ of expansion coefficients in the Bloch basis corresponds to the eigenstate $|u_{n\mathbf{k}}\rangle$, described using the Dirac notation in Eq. (2).

3. Recall that the topological invariant of the SSH model (i.e. the winding number) is given by

$$\nu = \frac{1}{\pi} \phi. \tag{4}$$

   Visualise the phase diagram of the SSH model by calculating the winding number with $v, w \in [0, 1]$.

4. (Optional) Build a finite SSH model with $v = 1.0$ and $w = 0.6$, and then show the charge distributions of the bulk state (i.e. the state with lowest energy) and the edge state (i.e. the state with the energy equal to the Fermi level).

   Helpful tips: A finite SSH model means applying the SSH model to a atmoic chain of finite length. You just need to build a supercell according to the unit cell of the SSH model and set the intercell hopping energy of the atoms at the boundary to zero. The eigenvector elements $\mathcal{C}_{n\mathbf{k}}$ of the finite model gives you the weights of the localised orbitals at each unit cell with band index n.