

A. Graphene Model

a) Direct and Reciprocal Lattice

As stated in every solid-state book the 2D honeycomb structure of graphene is not a proper Bravais lattice. To use all the powerful symmetry tools associated to translational invariance we have in fact to define complementary sublattices that are true Bravais lattices. This is equivalent to define a 2-atom cell in terms of which we can define all the machinery of Bloch theory. Our choice of basis, unit cell and vectors and the definition of some useful “bond vectors” is depicted in Figure 1. The coordinates of these vectors in a xy system, where x is parallel to a bond, are:

$$\begin{cases} \mathbf{v}_1 = \left(\frac{3}{2}, \frac{\sqrt{3}}{2}\right) \ell \\ \mathbf{v}_2 = \left(\frac{3}{2}, -\frac{\sqrt{3}}{2}\right) \ell \end{cases} \quad \begin{cases} \ell_1 = (1, 0) \ell \\ \ell_2 = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right) \ell \\ \ell_3 = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right) \ell \end{cases}$$

where ℓ is the lattice spacing length. Fixing the origin of the frame on a “red” atom we can specify also the basis vectors, then being: $\mathbf{a}_1 = \mathbf{0}$ and $\mathbf{a}_2 = \ell_1$.

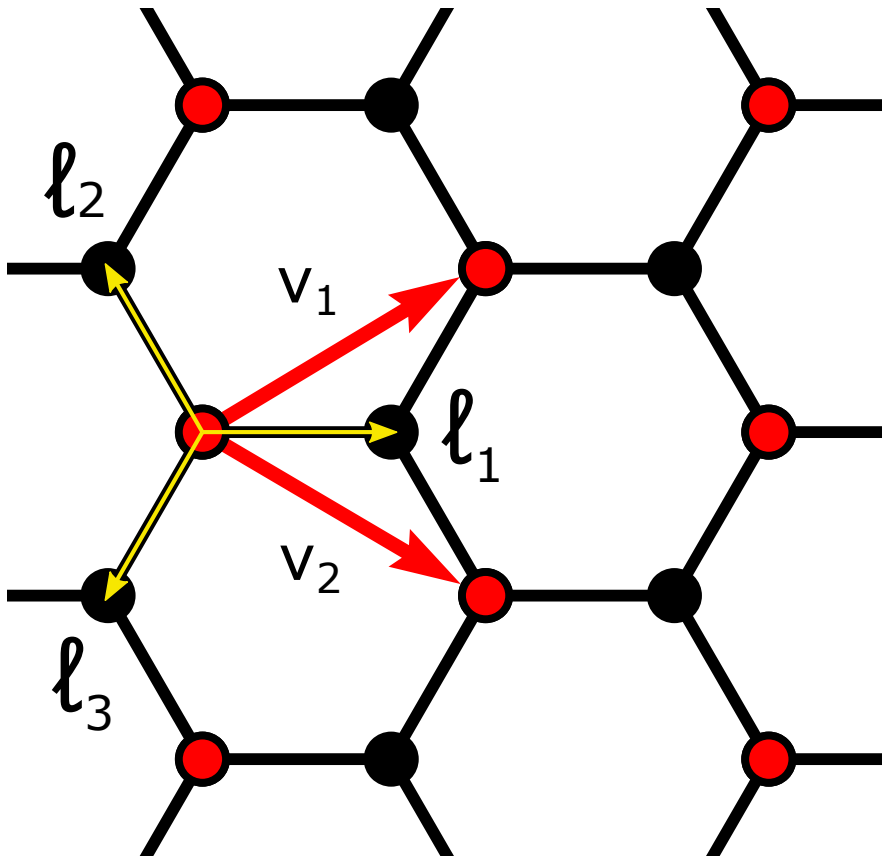


Figure 1: Primitive vectors and cell of the honeycomb lattice. The two sublattices are indicated with different colors. The three “bond vectors” ℓ_i are indicated for future convenience.

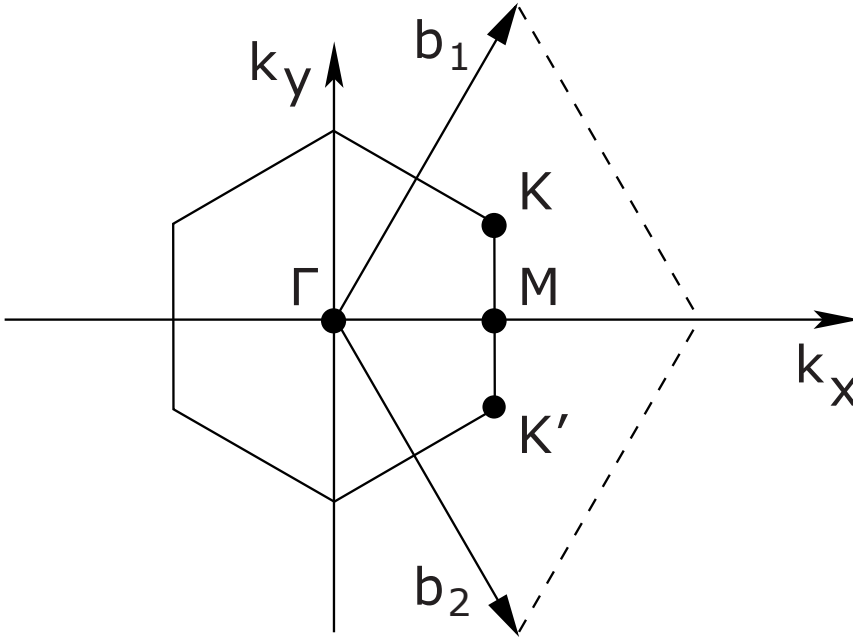


Figure 2: Reciprocal primitive vectors \mathbf{b}_i and Brillouin Zone of the honeycomb lattice are indicated along with high symmetry points Γ , K , K' and M .

As usual, the reciprocal lattice primitive vectors are defined by the relation:

$$\mathbf{b}_i \cdot \mathbf{v}_j = 2\pi\delta_{ij} \quad \forall i, j$$

A possible choice is:

$$\mathbf{b}_1 = \frac{2\pi}{3\ell}(1, \sqrt{3})$$

$$\mathbf{b}_2 = \frac{2\pi}{3\ell}(1, -\sqrt{3})$$

These vectors are displayed in Figure 2, together with the resulting 1st Brillouin Zone (Wigner-Seitz unit cell in \mathbf{k} space).

b) Π -bond Approximation

In Figure 3 is shown a sketch of the symmetry-induced linear combinations of graphene atomic orbitals, namely the $2sp^2$ in-plane orbitals (arising from the hybridization of $2s$, $2p_x$ and $2p_y$ eigenfunctions) and the remaining orthogonal-to-plane $2p_z$ orbital. The respective bonds and antibonds are of σ and π type. Because of the very different overlap, the latter energies will split much less when the interatomic perturbation is switched on. Thus, filling the resulting energy scheme with the 12 electrons present in the unit-cell, we may guess that the π and π^* molecular levels will give rise respectively to the valence and conduction bands in the solid. Actually this will be true only if there is no mixing between these levels, i.e. if the generic matrix element $\langle p_z | \mathcal{H} | sp^2 \rangle$ is zero. But indeed it is since $\langle r | p_z \rangle$ is odd with respect to \bar{z} and $\langle r | sp^2 \rangle$ together with \mathcal{H} of course is instead even: so the whole integrand is even and integrating on all physical space we get zero. Thus, we can state that the π and σ hamiltonians will be exactly separable due to orthogonality in physical space and that in order to study transport properties we can indeed focus on just the π problem¹.

¹In general we would say that in graphene there are essentially three electronic energy scales: the lower one corresponding to the core levels, accounting for the screening of nuclei positive charge; an intermediate scale related to the sp^2 - σ lattice, accounting for the cohesive energy of the monolayer; finally the higher scale of π bonds, accounting for all electronic transport properties of the material.

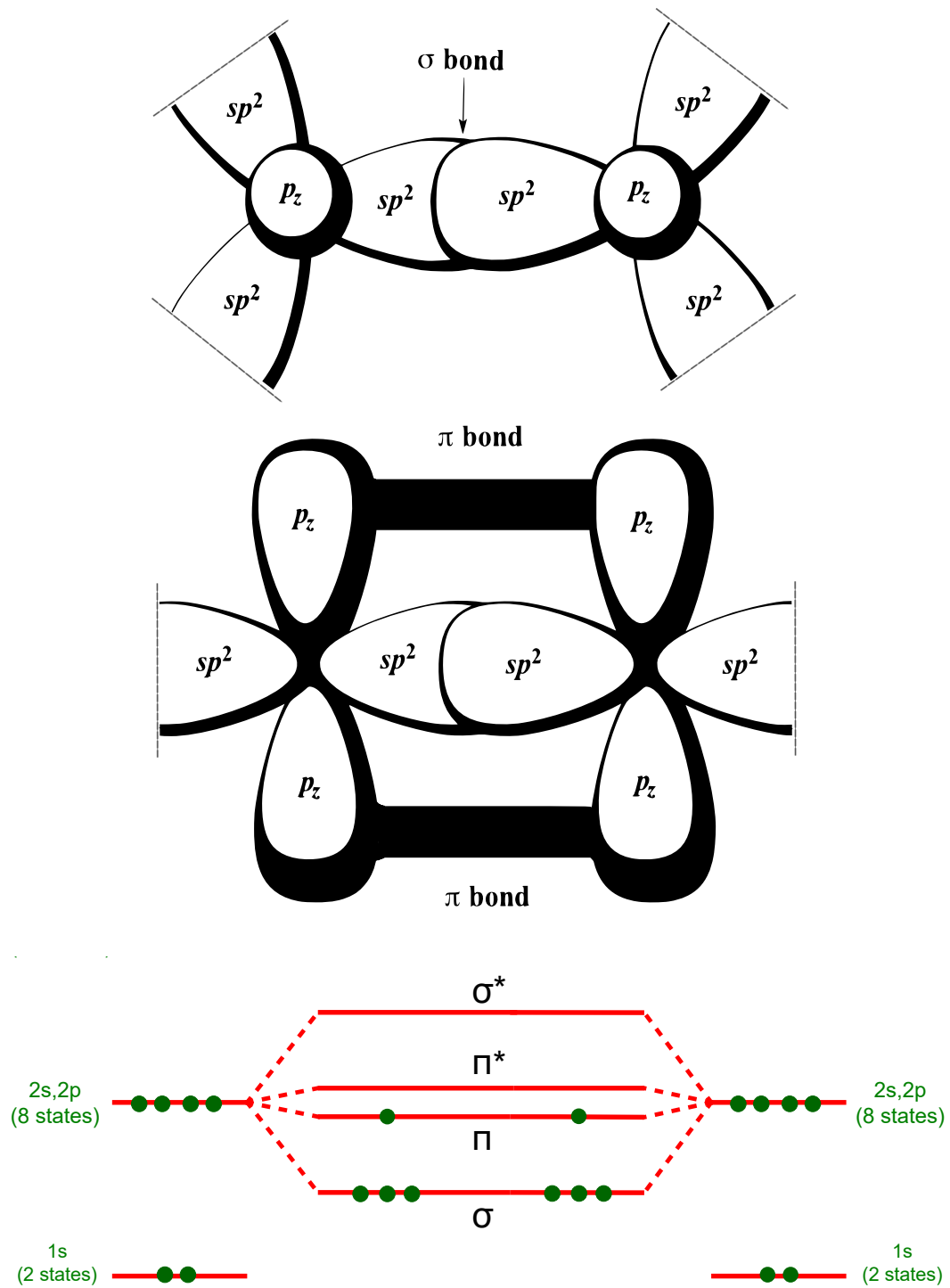


Figure 3: Depiction of the hybrid sp^2 and pure p_z orbitals (in a unit-cell) and the consequent energy scheme of σ and π bonds. Notice that being the π bond orbital orthogonal to the sp^2 their overlap can be neglected and so in perturbation theory the mixing of the correspondent levels will be irrelevant.

c) Tight-Binding Hamiltonian

The Bloch functions for the two basis atoms ($\mathbf{a}_1, \mathbf{a}_2$) are:

$$\begin{aligned} f_1(\mathbf{r}) &= \sum_{\mathbf{R}} e^{i\mathbf{k} \cdot (\mathbf{R} + \mathbf{a}_1)} \phi_1(\mathbf{r} - (\mathbf{R} + \mathbf{a}_1)) \\ f_2(\mathbf{r}) &= \sum_{\mathbf{R}} e^{i\mathbf{k} \cdot (\mathbf{R} + \mathbf{a}_2)} \phi_2(\mathbf{r} - (\mathbf{R} + \mathbf{a}_2)) \end{aligned} \quad (1)$$

Where ϕ is the p_z atomic orbital of Carbon. Hence Hamiltonian matrix elements are:

$$\begin{aligned} \langle f_i | H | f_j \rangle &= \sum_{\mathbf{R}, \mathbf{R}'} e^{i\mathbf{k} \cdot (\mathbf{R} - \mathbf{R}' + \mathbf{a}_j - \mathbf{a}_i)} \langle \phi_i(\mathbf{r} - (\mathbf{R}' + \mathbf{a}_i)) | \phi_j(\mathbf{r} - (\mathbf{R} + \mathbf{a}_j)) \rangle, \\ \langle f_i | H | f_i \rangle &= \sum_{\mathbf{R}, \mathbf{R}'} e^{i\mathbf{k} \cdot (\mathbf{a}_i - \mathbf{a}_i)} \delta_{\mathbf{R}, \mathbf{R}'} = \varepsilon, \end{aligned} \quad (2)$$

and, by considering nearest neighbor approximation and replacing $\mathbf{a}_1, \mathbf{a}_2$ with their values, off-diagonal matrix elements can be calculated as:

$$\begin{aligned} \langle f_1 | H | f_2 \rangle &= \sum_{\mathbf{R}, \mathbf{R}'} e^{i\mathbf{k} \cdot (\mathbf{R} - \mathbf{R}' + \boldsymbol{\ell}_1 - 0)} (-t) (\delta_{\mathbf{R}, \mathbf{R} + \boldsymbol{\ell}_1} + \delta_{\mathbf{R}, \mathbf{R} - (\boldsymbol{\ell}_1 - \boldsymbol{\ell}_2)} + \delta_{\mathbf{R}, \mathbf{R} - (\boldsymbol{\ell}_1 - \boldsymbol{\ell}_3)}) \\ &= -t (e^{i\mathbf{k} \cdot \boldsymbol{\ell}_1} + e^{i\mathbf{k} \cdot \boldsymbol{\ell}_2} + e^{i\mathbf{k} \cdot \boldsymbol{\ell}_3}) \end{aligned} \quad (3)$$

Where the three δ -functions select the three nearest \mathbf{a}_2 neighbors for atom \mathbf{a}_1 . (see Figure 1)

Thus within nearest neighbors tight-binding approximation the Hamiltonian of the system is represented by a 2×2 matrix in the basis of Bloch states:

$$\mathbb{H}_{\mathbf{k}} = \begin{pmatrix} \varepsilon & -\tau(\mathbf{k}) \\ -\tau^*(\mathbf{k}) & \varepsilon \end{pmatrix}, \quad \tau(\mathbf{k}) = t \sum_{n=1}^3 e^{i\mathbf{k} \cdot \boldsymbol{\ell}_n}, \quad t \simeq 2.5 \text{ eV}.$$

d) Electronic Structure

To immediately find the eigenvalues of our 2×2 matrices we used the following trick. Let us consider a generic 2×2 hermitian matrix in the form:

$$\mathbb{M} = \Pi_0 + \boldsymbol{\Pi} \cdot \boldsymbol{\sigma}$$

where $\boldsymbol{\sigma}$ indicates the vector of the three Pauli matrices. Then the eigenvalues of \mathbb{M} are proved to be:

$$\lambda = \Pi_0 \pm |\boldsymbol{\Pi}|$$

In our case ($\mathbb{M} = \mathbb{H}_{\mathbf{k}}$) we see that:

$$\Pi_0 = \varepsilon, \quad \Pi_x = -t \sum_{n=1}^3 \cos(\mathbf{k} \cdot \boldsymbol{\ell}_n), \quad \Pi_y = +t \sum_{n=1}^3 \sin(\mathbf{k} \cdot \boldsymbol{\ell}_n), \quad \Pi_z = 0.$$

Thus, the eigenvalues of $\mathbb{H}_{\mathbf{k}}$ are

$$\lambda_{\mathbf{k}}^{\pm} = \varepsilon \pm t \sqrt{\left(\sum_{n=1}^3 \cos(\mathbf{k} \cdot \boldsymbol{\ell}_n) \right)^2 + \left(\sum_{n=1}^3 \sin(\mathbf{k} \cdot \boldsymbol{\ell}_n) \right)^2} = \varepsilon \pm |\tau|.$$

The two bands $\lambda_{\mathbf{k}}^{\pm}$ touch each other in the points:

$$\mathbf{K} = \frac{2\pi}{3\ell} \left(1, \frac{1}{\sqrt{3}} \right) \quad \mathbf{K}' = \frac{2\pi}{3\ell} \left(1, -\frac{1}{\sqrt{3}} \right).$$

In fact in these points $|\tau| = 0$. For example, considering the point \mathbf{K} :

$$\tau(\mathbf{K}) = t \sum_{n=1}^3 e^{i\mathbf{K} \cdot \boldsymbol{\ell}_n} = t \left(e^{\frac{\mathbf{K} \cdot \boldsymbol{\ell}_1}{2\pi/3}} + 1 + e^{\frac{\mathbf{K} \cdot \boldsymbol{\ell}_3}{-2\pi/3}} \right) = 0.$$

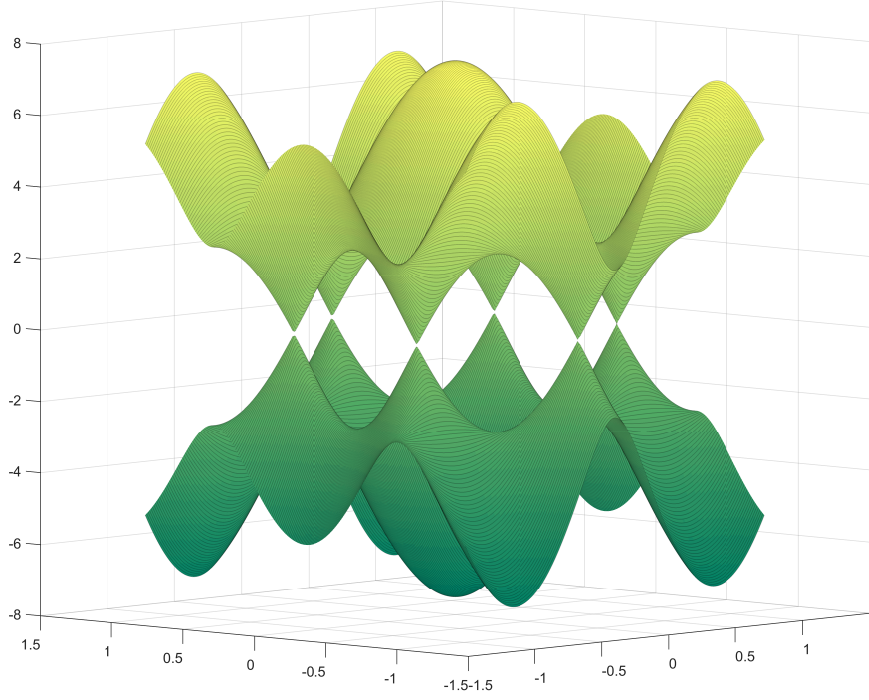


Figure 4: Band Structure of graphene as computed from nearest neighbors tight-binding. The surface-plot is not restricted to a Brillouin Zone to better display the shape of the bands (especially the Dirac cones). Energy is measured in eV. k_x, k_y are in atomic units.

e) Density of States

From the definition of density of states (DOS) we know that the quantity $\rho(E) dE$ corresponds to the number of states with energy in the interval $[E, E + dE]$. So we can compute numerically $\rho(E)$ by just evaluating the band structure $E(\mathbf{k})$ in a discrete meshgrid² $\{\mathbf{k}\} = (\{k_x\}, \{k_y\})$ and counting the number of points within appropriate³ energy bins. This procedure will correspond formally to calculate ρ as:

$$\rho(E) = \sum_i n_i \delta(E - E_i), \quad (4)$$

where E_i is the center of the i -th bin and n_i the number of normalized states comprised in it.

The resulting discrete plot is reported in Figure 5, where the count points have been linearly connected to better display the resulting DOS shape.

But the result is still unsatisfying in terms of prediction of an experimental measure (even if we assume arbitrary high resolution): we know that $\rho(E)$ should be normalized to the number of electrons in a unit cell but in our case we have

$$\sum_i n_i = 2$$

and so the area below the linearly connected shape will be much greater. The problem would be solved if we took infinitesimal bins but then would be impossible to control the count noise. Then we need to spread the weight around the discrete E_i points thus obtaining a continuous function with the correct integral. This can be done substituting the delta function in Eq. (4) with a smooth normalized kernel $g(E - E_i)$:

$$\sum_i n_i \delta(E - E_i) \mapsto \sum_i n_i g(E - E_i), \quad \int_{\mathbb{R}} g(E) dE = 1. \quad (5)$$

Our choice for this *smearing function* is a normalized gaussian centered at E_i , with standard deviation $\sigma = 0.1$ eV (so we obtain a *gaussian-smear*ed density of states):

$$g(E - E_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(E - E_i)^2}{2\sigma^2}\right).$$

The result is shown in Figure 6 and compared to the discrete count in Figure 7.

²To increase the number of counts we have defined a meshgrid spanning several Brillouin Zones.

³They should be much smaller than the total bandwidth in order to get the correct shape but not too much, to avoid excessive noise.

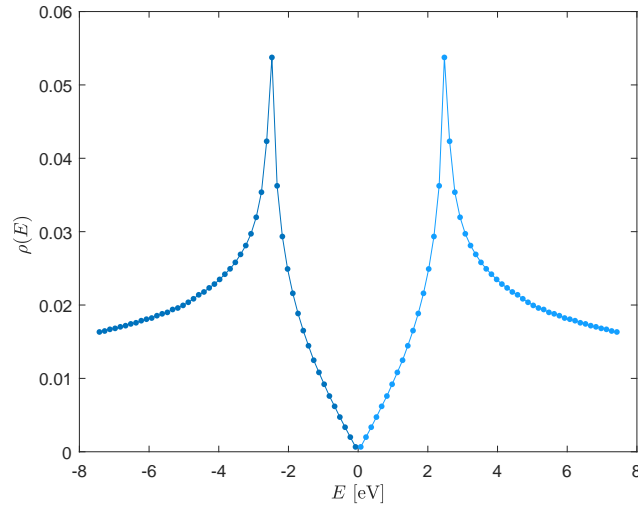


Figure 5: Discrete count of states in energy bins: accurate DOS shape

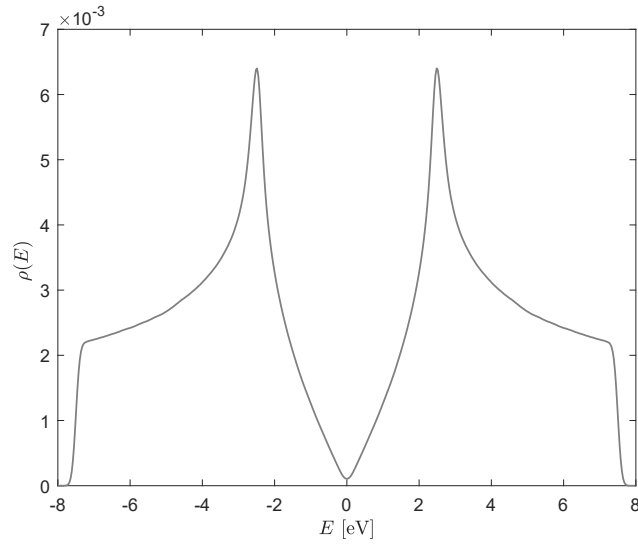


Figure 6: Gaussian smearing of the discrete count: experimental-like DOS

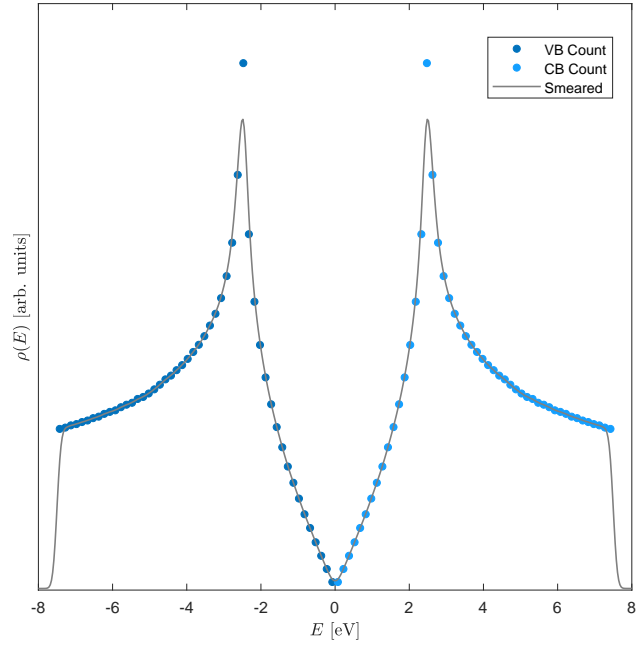


Figure 7: Comparison of discrete and smeared DOS: finite resolution effects

We see that the DOS of graphene has a very peculiar shape, especially at three relevant energies:

- at the Fermi energy we have zero states, so we can regard graphene as a *semimetal*. In particular we have verified that $\rho(E)$ goes to zero linearly by means of a least squares best fit. The result is reported in Figure 8.
- at $E = \pm t$ we have instead a divergence of states corresponding to the saddle points of the band structure. We have verified that that divergence is logarithmic again by least squares method. The result is reported in Figure 9.

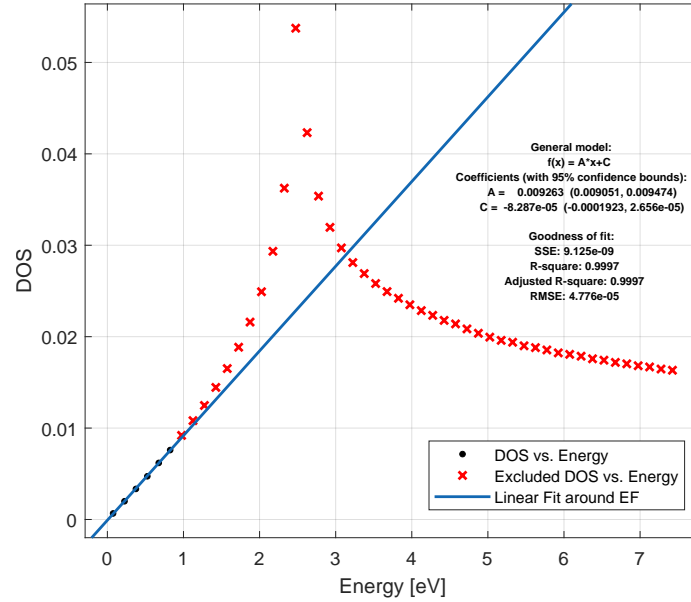


Figure 8: Linear Fit near the E_F : “zero-DOS” (semi)metal.

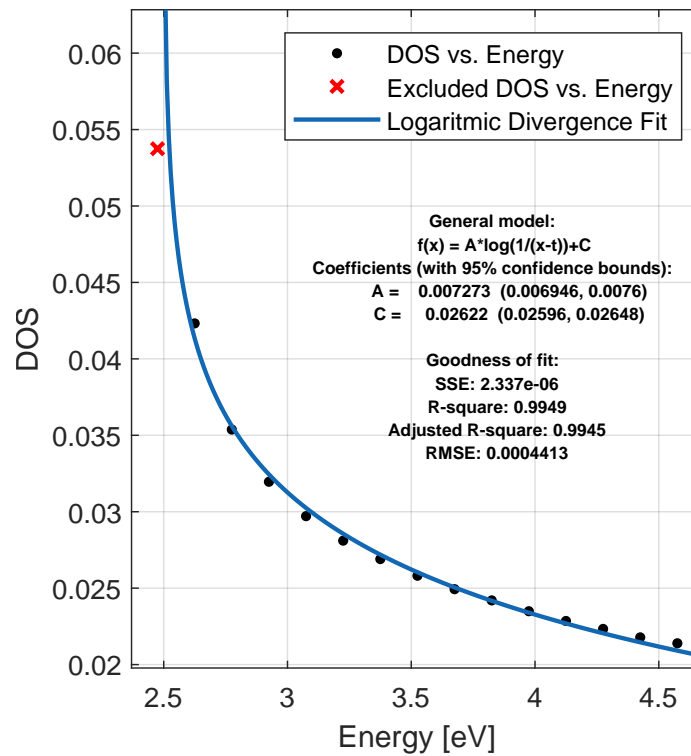


Figure 9: Logarithmic Fit of the divergence at $E = (\pm)t$.

f) Tunneling Probability of K-point Carriers

By expanding $\tau(\mathbf{k})$ around \mathbf{K} one obtains:

$$\begin{aligned}\tau(\mathbf{K} + \mathbf{q}) &\simeq it \sum_{n=1}^3 e^{i\mathbf{K} \cdot \boldsymbol{\ell}_n} (\boldsymbol{\ell}_n \cdot \mathbf{q}) \\ &= t \ell \left(e^{i\frac{2\pi}{3}} q_x + \left(-\frac{q_x}{2} + \frac{\sqrt{3}q_y}{2} \right) + e^{-i\frac{2\pi}{3}} \left(-\frac{q_x}{2} - \frac{\sqrt{3}q_y}{2} \right) \right) \\ &= \frac{3t\ell}{2} \left(q_x \left(i\frac{\sqrt{3}}{2} - \frac{1}{2} \right) + q_y \left(i\frac{\sqrt{3}}{2} + \frac{1}{2} \right) \right) = \frac{3t\ell}{2} e^{i2\pi/3} (q_x - iq_y) .\end{aligned}$$

Thus:

$$|\tau(\mathbf{K} + \mathbf{q})| \simeq \hbar v_F |\mathbf{q}|$$

in which we defined the Fermi velocity $v_F = 3t\ell/(2\hbar)$. In our case:

$$v_F \simeq \frac{3}{2} \frac{2.5 \text{ eV}}{6.58 \cdot 10^{-16} \text{ eV} \cdot \text{s}} 0.14 \cdot 10^{-9} \text{ m} \simeq 8 \cdot 10^5 \frac{\text{m}}{\text{s}} \approx \frac{c}{300}$$

It is straightforward to similarly obtain:

$$\tau(\mathbf{K}' + \mathbf{q}) \simeq \frac{3t}{2} e^{i2\pi/3} (q_x + iq_y)$$

and $|\tau(\mathbf{K}' + \mathbf{q})| \simeq \hbar v_F |\mathbf{q}|$. So in the proximity of the K -points, one has:

$$(\mathbb{H}_{\mathbf{K}} - \varepsilon \mathbb{1}) \simeq \hbar v_F \mathbf{q} \cdot \boldsymbol{\sigma} = \mathbb{H}_D, \quad \boldsymbol{\sigma} = (\sigma_x, \sigma_y),$$

except for an irrelevant phase-factor $e^{i2\pi/3}$.

The relevant hamiltonian is then:

$$\mathbb{H}_D = v_F \mathbf{p} \cdot \boldsymbol{\sigma}, \quad \mathbf{p} = \frac{\hbar}{i} \left(\frac{d}{dx}, \frac{d}{dy} \right),$$

which can be formally regarded as that of a relativistic (i.e. massless) particle of spin $1/2$, with an effective speed $c = v_F$.

We are now interested in the scattering process with the barrier potential:

$$V(x) = \begin{cases} 0 & x < 0 \\ V_0 & 0 \leq x \leq D \\ 0 & x > D \end{cases}$$

The eigenvalue problem is:

$$(\mathbb{H}_D + V(x))\Psi(x, y) = E\Psi(x, y)$$

in which $\Psi(x, y)$ is a two-components vector⁴. Since the problem is invariant under translations along y , we can look for solutions in the form:

$$\Psi(x, y) = \psi(x) e^{ik_y y}$$

For simplicity we consider the case $k_y = 0$ of a particle orthogonally inciding the barrier. In this case:

$$(\mathbb{H}_D + V(x))\Psi(x, y) = (v_F p_x \sigma_x + V(x))\psi(x) \quad p_y = \hbar k_y = 0$$

We can separately solve the scattering problem for the cases in which the spin is aligned/anti-aligned along x . In the first case we can write:

$$\psi(x) = \psi(x) |\uparrow\rangle$$

⁴The Schrödinger equation is completely equivalent to the Dirac equation $i\gamma^\mu \partial_\mu \psi = 0$, which is in a Lorentz-invariant form. The Dirac matrices γ^μ have to respect the algebra $\{\gamma^\mu, \gamma^\nu\} = 2\eta^{\mu\nu} \mathbb{1}$, where $\mathbb{1}$ is the d -dimensional identity matrix. In the case of a $2+1$ space-time a possible choice is $d = 2$ and: $\gamma^0 = \sigma_z$, $\gamma^1 = -i\sigma_y$, $\gamma^2 = i\sigma_x$, which gives rise to our Schrödinger equation.

(we use the convention $|\uparrow\rangle, |\downarrow\rangle$ for the two eigenvectors of σ_x). Defining $p = E/(\hbar v_F)$ and $u_0 = V_0/(\hbar v_F)$, we get:

$$\begin{cases} p_x \psi = p \psi & x < 0 \\ (p_x + u_0) \psi = p \psi & 0 \leq x \leq D \\ p_x \psi = p \psi & x > D \end{cases}$$

In the case $E < V_0$ we have $p = \hbar k < u_0$ and hence: $p - u_0 = -q < 0$. We must solve the above system of equations, also imposing the continuity of the wave-function in $x = 0$ and $x = +D$. A solution is simply:

$$\psi(x) = \begin{cases} e^{ikx} & x < 0 \\ e^{-iqx} & 0 \leq x \leq D \\ e^{ikx} e^{-i(q+k)D} & x > D \end{cases}$$

We see that the barrier is perfectly transparent in the case of normal incidence. In fact: $|T|^2 = 1$, $|R|^2 = 0$. This fact can also be seen by considering the velocity operator along the x -axis:

$$v_x = v_F \sigma_x$$

In the Heisenberg picture we get:

$$\frac{d}{dt} v_x = \frac{iv_F}{\hbar} [H, \sigma_x] = -\frac{2(v_F)^2}{\hbar} \sigma_z p_y = 0$$

Thus, the x -component of the velocity can not change and our massless Dirac particle can not be backscattered. This result is known as the *Klein paradox*.

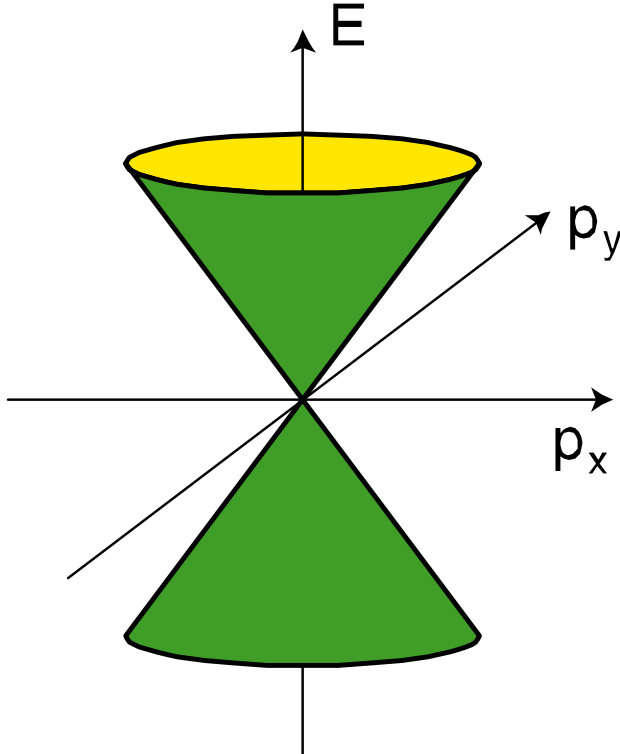


Figure 10: Schematic representation of the Dirac Cones placed in graphene band structure at the \mathbf{K}/\mathbf{K}' points in the Brilluoin Zone. The correspondent dispersion relation around E_F is then linear: formally the same of massless particles.

B2. Carbon Nanotubes

Now we address two types of Carbon Nanotubes (CNTs), known as “armchair” and “zig-zag” geometries. The way we cut and connect the graphene strips is depicted in Figure 11.

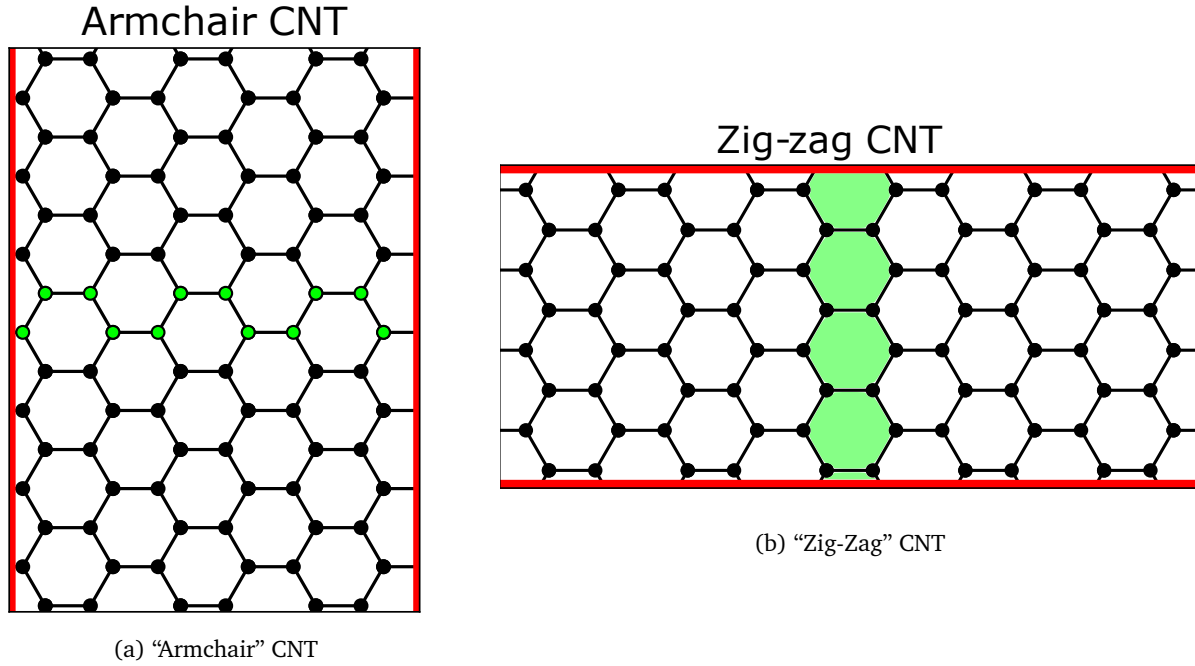


Figure 11: Geometries of the Nanotubes addressed in this section.

a) Unit Cells, Bases, Brillouin Zones

Unit cell of armchair and zig-zag CNTs is considered in a way to produce all CNT atoms by orthogonal repetition along \vec{x} and \vec{y} axes (see figure 12).

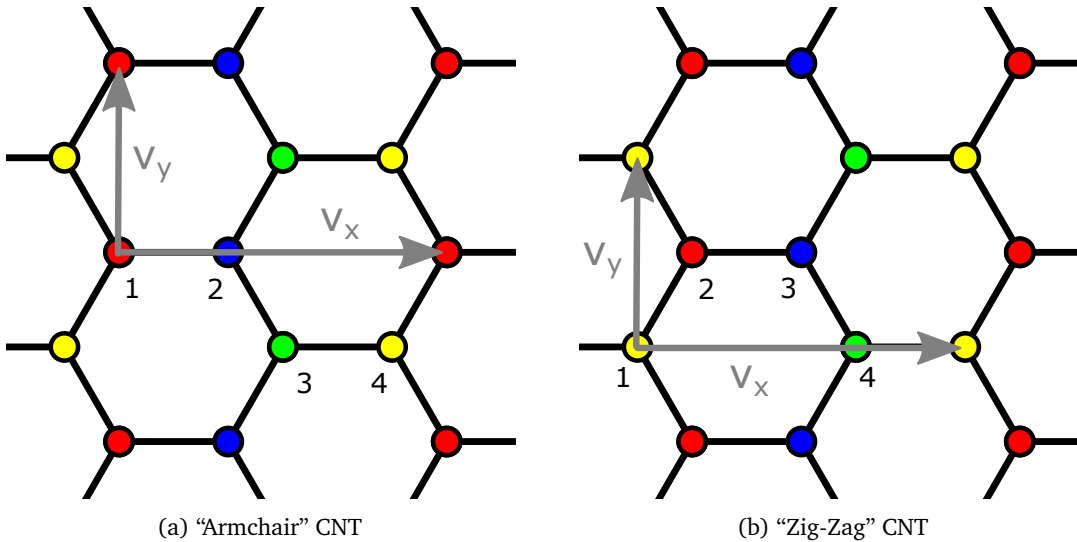


Figure 12: Unit cells of the two discussed CNTs. The four basis atoms are indicated with different colours and labeled according to consequent matrix notation. Primitive vectors along \vec{x} and \vec{y} are indicated in grey color.

In the armchair case, the atom positions in the cell are:

$$\mathbf{a}_1 = \mathbf{0}, \quad \mathbf{a}_2 = \ell_1, \quad \mathbf{a}_3 = \ell_1 - \ell_2, \quad \mathbf{a}_4 = 2\ell_1 - \ell_2; \quad (6)$$

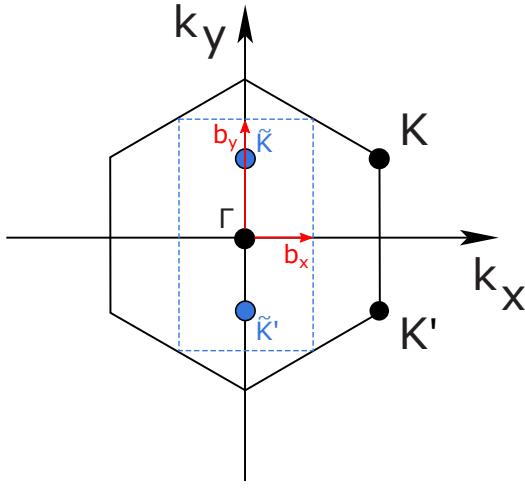


Figure 13: Our choice for the Nanotube Brillouin Zone and the mapping of Dirac cones inside it.

where the vectors ℓ_i are defined in section A. In the zigzag case we have instead:

$$\mathbf{a}_1 = \mathbf{0}, \quad \mathbf{a}_2 = -\ell_3, \quad \mathbf{a}_3 = \ell_1 - \ell_3, \quad \mathbf{a}_4 = \ell_1 - \ell_3 - \ell_2; \quad (7)$$

In both cases, the vectors generating the lattice in real space are:

$$\mathbf{v}_x = (3\ell, 0) \quad \mathbf{v}_y = (0, \sqrt{3}\ell) \quad (8)$$

The corresponding reciprocal vectors are

$$\mathbf{b}_x = \left(\frac{2\pi}{3\ell}, 0 \right) \quad \mathbf{b}_y = \left(0, \frac{2\pi}{\sqrt{3}\ell} \right), \quad (9)$$

which respect the relations:

$$\begin{aligned} \mathbf{b}_x \cdot \mathbf{v}_x &= 2\pi & \mathbf{b}_x \cdot \mathbf{v}_y &= 0 \\ \mathbf{b}_y \cdot \mathbf{v}_x &= 0 & \mathbf{b}_y \cdot \mathbf{v}_y &= 2\pi \end{aligned}$$

In the (k_x, k_y) space the first Brillouin zone (BZ₁) corresponds to the rectangle

$$\text{BZ}_1 = \left\{ \mathbf{k} \mid k_x \in \left[-\frac{\pi}{3\ell}, \frac{\pi}{3\ell} \right] ; k_y \in \left[-\frac{\pi}{\sqrt{3}\ell}, \frac{\pi}{\sqrt{3}\ell} \right] \right\}$$

in which the Dirac K-points are mapped by a reciprocal vector translation:

$$\begin{aligned} \tilde{\mathbf{K}} &= \mathbf{K} - \mathbf{b}_x = \left(\frac{2\pi}{3\ell} - \frac{2\pi}{3\ell}, \frac{2\pi}{3\sqrt{3}\ell} \right) = \left(0, \frac{2\pi}{3\sqrt{3}\ell} \right) \in \text{BZ}_1 \\ \tilde{\mathbf{K}}' &= \mathbf{K}' - \mathbf{b}_x = \left(\frac{2\pi}{3\ell} - \frac{2\pi}{3\ell}, -\frac{2\pi}{3\sqrt{3}\ell} \right) = \left(0, -\frac{2\pi}{3\sqrt{3}\ell} \right) \in \text{BZ}_1 \end{aligned}$$

It is easy to check that all the others (equivalent) Dirac points are mapped in $\tilde{\mathbf{K}}$ or $\tilde{\mathbf{K}}'$ (see Figure 13).

In the armchair (a) case the lattice is folded along the x direction, whereas in zigzag (b) case the folding is along y. Furthermore, along the direction of folding the \mathbf{k} vectors are “quantized”, i.e. they cannot assume all possible values.

- a) in the armchair case the possible values of k_x are:

$$k_x = \frac{2\pi}{3\ell N} m, \quad m = \begin{cases} 0, \pm 1, \dots, \pm \frac{N}{2} - 1, +\frac{N}{2} & N \text{ even} \\ 0, \pm 1, \dots, \pm \frac{N-1}{2} & N \text{ odd} \end{cases}$$

in which N is the number of cells along the x direction (they form a ring). Instead k_y can assume all values within the BZ₁.

- *b)* in the zigzag case the possible values of k_y are:

$$k_y = \frac{2\pi}{\sqrt{3}\ell N} m, \quad m = \begin{cases} 0, \pm 1, \dots, \pm \frac{N}{2} - 1, +\frac{N}{2} & N \text{ even} \\ 0, \pm 1, \dots, \pm \frac{N-1}{2} & N \text{ odd} \end{cases}$$

in which N is the number of cells along the x direction (they form a ring). Instead k_y can assume all values within the BZ_1 .

We immediately see an important feature of these foldings. In fact, in the *a)* armchair case the Dirac points $\tilde{\mathbf{K}}, \tilde{\mathbf{K}}'$ are always possible wave-vectors, with $m = 0$. On the other hand, in the *b)* zigzag case the $\tilde{\mathbf{K}}, \tilde{\mathbf{K}}'$ points are possible wave-vectors only if N is a multiple of 3 (corresponding to $m = \pm N/3$). Since the Dirac points are the points in which the gap vanishes, we expect that in the *a)* armchair case the system is gapless for all the values of N , whereas in the *b)* zigzag case the system is gapless only if N is a multiple of 3.

b) Hamiltonians and fixed- N Band Structures

To write armchair and zig-zag Hamiltonians we have (at least) two options:

- first, treating all the atoms in a ring as a single – of course nonprimitive – cell;
- second, considering a ring as N repeated unit cells and so applying Bloch Theorem along the ring.

We start from the first method, getting a $4N \times 4N$ matrix in both CNT cases. We will show that this matrix can be reduced to 4×4 matrices, which can be directly obtained from the second method.

Armchair (ACNT)

The full Hamiltonian of a ring is a $4N \times 4N$ matrix, which contains N unit-cell blocks on its diagonal, connected by inter-block hopping terms. On top-right and bottom-left corners we have put the periodic boundary hoppings.

$$\mathbb{H}_{\text{ring}}(\mathbf{k}) = \begin{pmatrix} \boxed{\mathbb{H}_{\text{cell}}(\mathbf{k})} & & & \cdots & & \mathbf{B}(\mathbf{k}) \\ & \mathbf{B}(\mathbf{k}) & & & & \cdots \\ & \mathbf{B}^*(\mathbf{k}) & \boxed{\mathbb{H}_{\text{cell}}(\mathbf{k})} & & & \\ & & & \mathbf{B}(\mathbf{k}) & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \mathbf{B}(\mathbf{k}) & \\ \mathbf{B}^*(\mathbf{k}) & & & & \mathbf{B}^*(\mathbf{k}) & \boxed{\mathbb{H}_{\text{cell}}(\mathbf{k})} \end{pmatrix},$$

where

$$\mathbb{H}_{\text{cell}}(\mathbf{k}) = \begin{pmatrix} \varepsilon & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} & 0 & 0 \\ -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} & \varepsilon & -t(e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3}) & 0 \\ 0 & -t(e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3}) & \varepsilon & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} \\ 0 & 0 & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} & \varepsilon \end{pmatrix},$$

$$\mathbf{B}(\mathbf{k}) = -t(e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3}).$$

If we apply instead Bloch Theorem on \vec{x} direction we reduce the full Hamiltonian to a 4×4 matrix: this is practically done by moving the $\mathbf{B}(\mathbf{k})$ element inside $\mathbb{H}_{\text{cell}}(\mathbf{k})$. We obtain:

$$\mathbb{H}_{\text{reduced}}(\mathbf{k}) = \begin{pmatrix} \varepsilon & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} & 0 & -t(e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3}) \\ -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} & \varepsilon & -t(e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3}) & 0 \\ 0 & -t(e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3}) & \varepsilon & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} \\ -t(e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3}) & 0 & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} & \varepsilon \end{pmatrix}$$

Zig-Zag (ZCNT)

The full Hamiltonian of a ring contains again N unit-cell blocks on its diagonal. But inter-block hopping terms are now placed between two different set of atoms (see Figure 12b for vertical hopping). We called them $B_{38}(\mathbf{k})$, $B_{25}(\mathbf{k})$ and placed on $(4N-1, 4(N+1))$ & $(4N-2, 4N+1)$ positions of the matrix, respectively.

So we have:

$$\mathbb{H}_{\text{ring}}(\mathbf{k}) = \begin{pmatrix} \boxed{\mathbb{H}_{\text{cell}}(\mathbf{k})} & 0 & 0 & 0 & 0 & \cdots \\ & B_{25}(\mathbf{k}) & 0 & 0 & 0 & \cdots \\ & 0 & 0 & 0 & B_{38}(\mathbf{k}) & \cdots \\ & 0 & 0 & 0 & 0 & \cdots \\ 0 & B_{25}^*(\mathbf{k}) & 0 & 0 & \boxed{\mathbb{H}_{\text{cell}}(\mathbf{k})} & \cdots \\ 0 & 0 & 0 & 0 & & \cdots \\ 0 & 0 & 0 & 0 & & \cdots \\ 0 & 0 & B_{38}^*(\mathbf{k}) & 0 & & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where

$$\mathbb{H}_{\text{cell}}(\mathbf{k}) = \begin{pmatrix} \varepsilon & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3} & 0 & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} \\ -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3} & \varepsilon & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} & 0 \\ 0 & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} & \varepsilon & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_2} \\ -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} & 0 & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2} & \varepsilon \end{pmatrix},$$

$$B_{25}(\mathbf{k}) = -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2},$$

$$B_{38}(\mathbf{k}) = -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3}.$$

Again, as in the armchair case, we can move in the inter-block elements to obtain the reduced matrix as follows:

$$\mathbb{H}_{\text{reduced}}(\mathbf{k}) = \begin{pmatrix} \varepsilon & -t(e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3} + e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_2}) & 0 & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} \\ -t(e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3} + e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2}) & \varepsilon & -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} & 0 \\ 0 & -te^{-i\mathbf{k}\cdot\boldsymbol{\ell}_1} & \varepsilon & -t(e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{-i\mathbf{k}\cdot\boldsymbol{\ell}_3}) \\ -te^{+i\mathbf{k}\cdot\boldsymbol{\ell}_1} & 0 & -t(e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_2} + e^{+i\mathbf{k}\cdot\boldsymbol{\ell}_3}) & \varepsilon \end{pmatrix},$$

The eigenvalues of both matrices, for a fixed N value, are reported in Figure 14, and discussed in the next paragraph.

Comparison between \mathbb{H}_{ring} and $\mathbb{H}_{\text{reduced}}$

As expected in case of diagonalizing $\mathbb{H}_{\text{reduced}}$ the result we get is the very same as for \mathbb{H}_{ring} , for any N value. But since there is only one allowed k point we cannot relate these one dimensional bands to the 2D graphene case, so we would expect physical insight to be much harder to obtain. (see Figure 14)

Thus we definitely prefer to diagonalize the 4×4 matrix for N times, instead of diagonalizing one time \mathbb{H}_{ring} .⁵

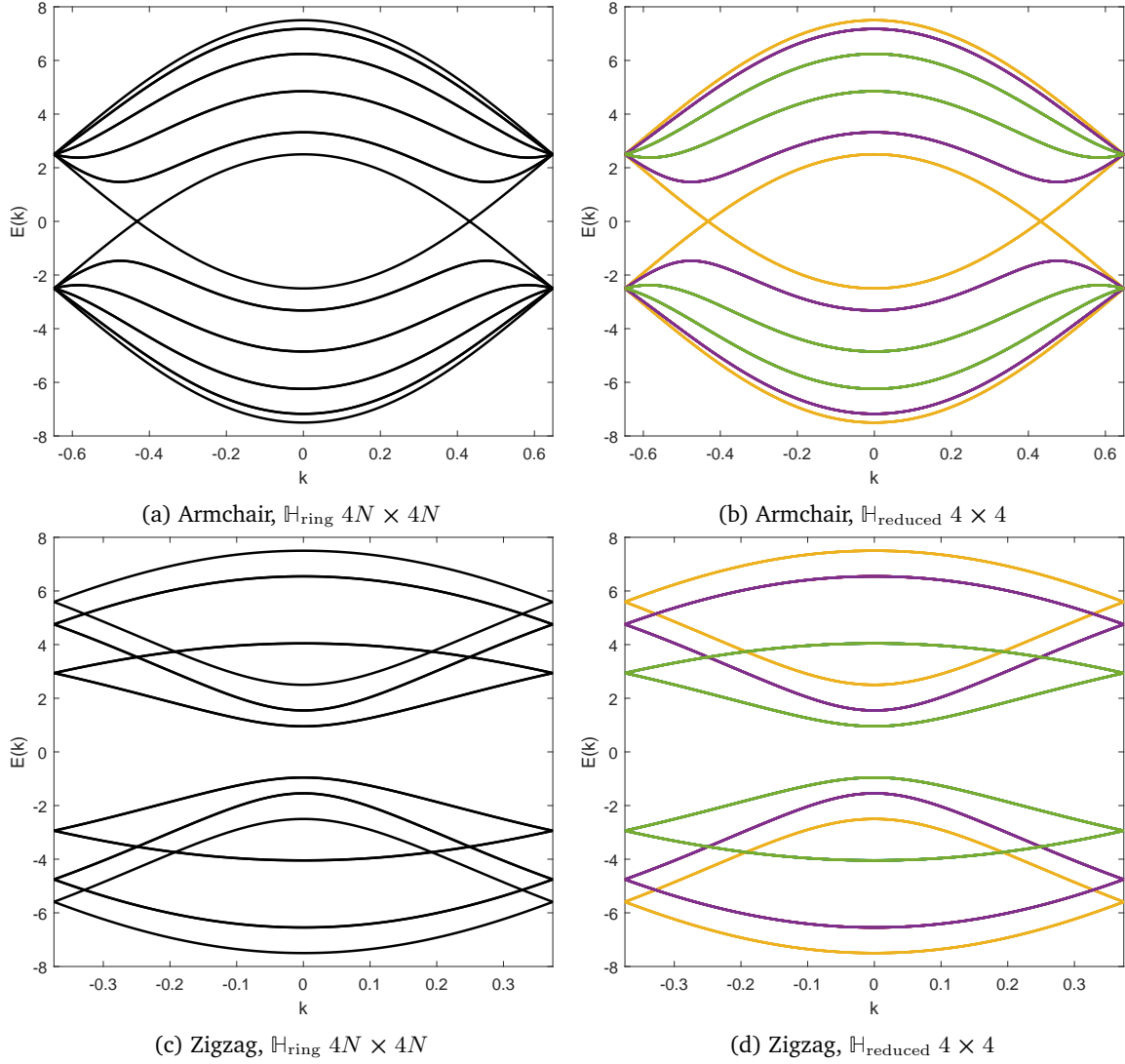


Figure 14: Two different Hamiltonians for $N = 5$. With \mathbb{H}_{ring} matrix (left), there is only one k_{ring} value while for $\mathbb{H}_{\text{reduced}}$ matrix (right) there are N k_{ring} values, each one represented with a different color. (Notice that some colors are hidden behind, due to band degeneracy)

⁵This will also improve computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$. In our “small N ” case this actually reduces algorithm performance, but we established that here the ease-of-interpretation gain would be indeed much more valuable.

c) Band Structures at increasing N

In Figures 15 and 16 we have reported our numerically calculated⁶ band structures (with the reduced matrix diagonalization method) at increasing N values, respectively for armchair and zig-zag CNTs. The following behaviour has been noticed:

- Armchair CNT has the same (semimetallic) band structure for all N values, it just get denser and denser. The closing of the gap happens at $\mathbf{k}_y = 2\pi/(3\sqrt{3}\ell)$.
- Zig-Zag changes its band structure for different N values, in particular it's a semimetal (at Γ) for N multiple of 3 and a semiconductor otherwise. Also we see flat bands for all even N values but not for odd values.

The origin of the first phenomenon is clear (if we refer to the reduced matrix method): changing N will change the allowed k_{ring} values and they may or may not cross the Dirac K-points. For armchair there is always some that fall on Dirac K-points, since these are placed at $k_x = 0$, that is always an allowed value; for zigzag we can or cannot fall on them depending on whether N is a multiple of 3 or not and that's exactly coherent with the prediction we presented in section a).

Conversely we have not figured out a reasonable explanation for the appearance of flat (dispersionless) bands in zig-zag CNTs, so we may discard it as an inherent limit of our simple nearest neighbors model.

⁶We have used Matlab® (<https://it.mathworks.com/products/matlab.html>) to carry out all the numerical calculations. The corresponding scripts are attached at the end of this document.

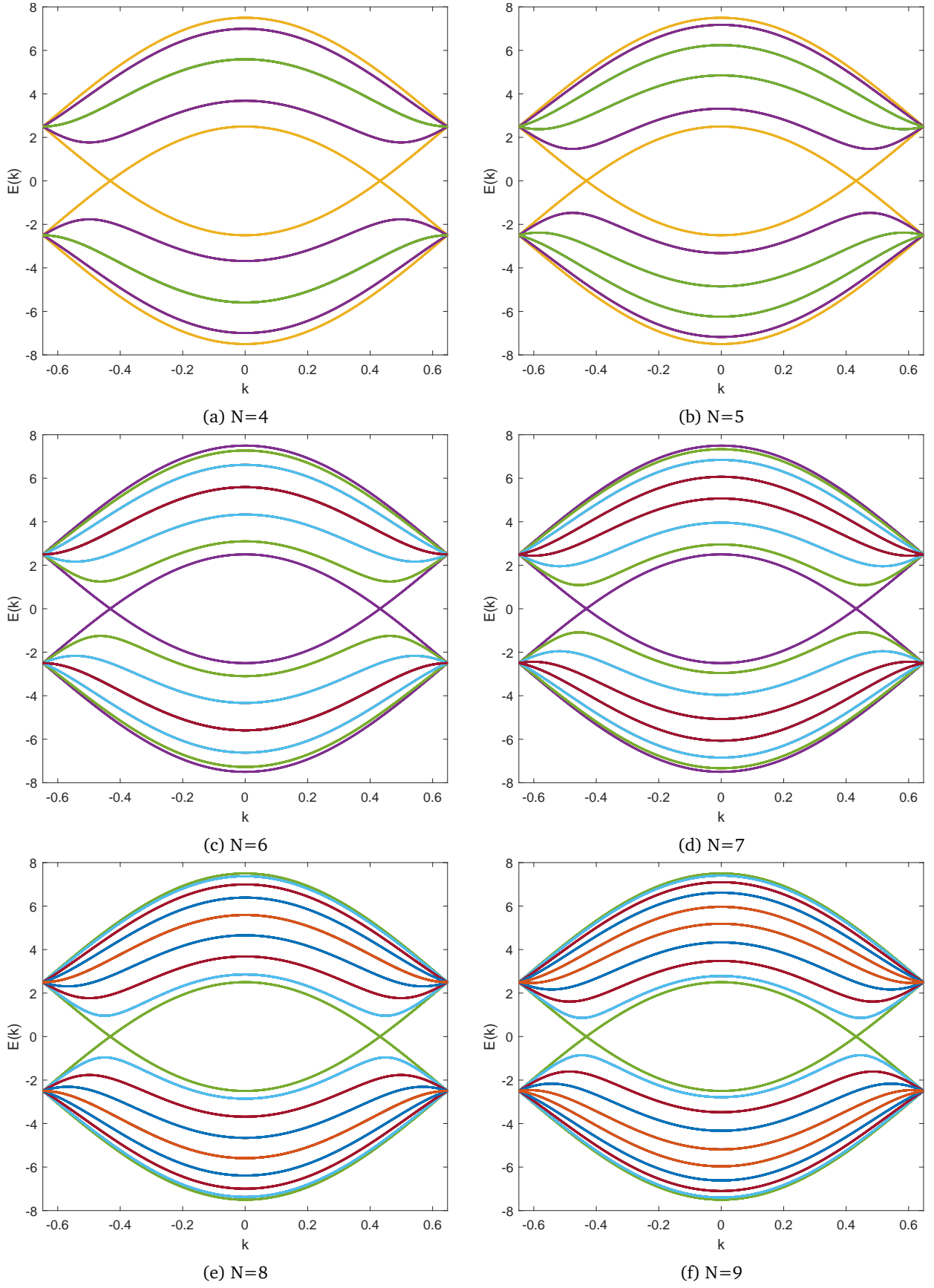


Figure 15: Electronic band structure of ACNT for different size of ring. N representing number of unit cells in one ring. Horizontal axis is K , which is simply k_y in case of armchair because it is folded along k_x . Vertical axis is energy in electronvolts (eV) unit and colors represent different k_x values.

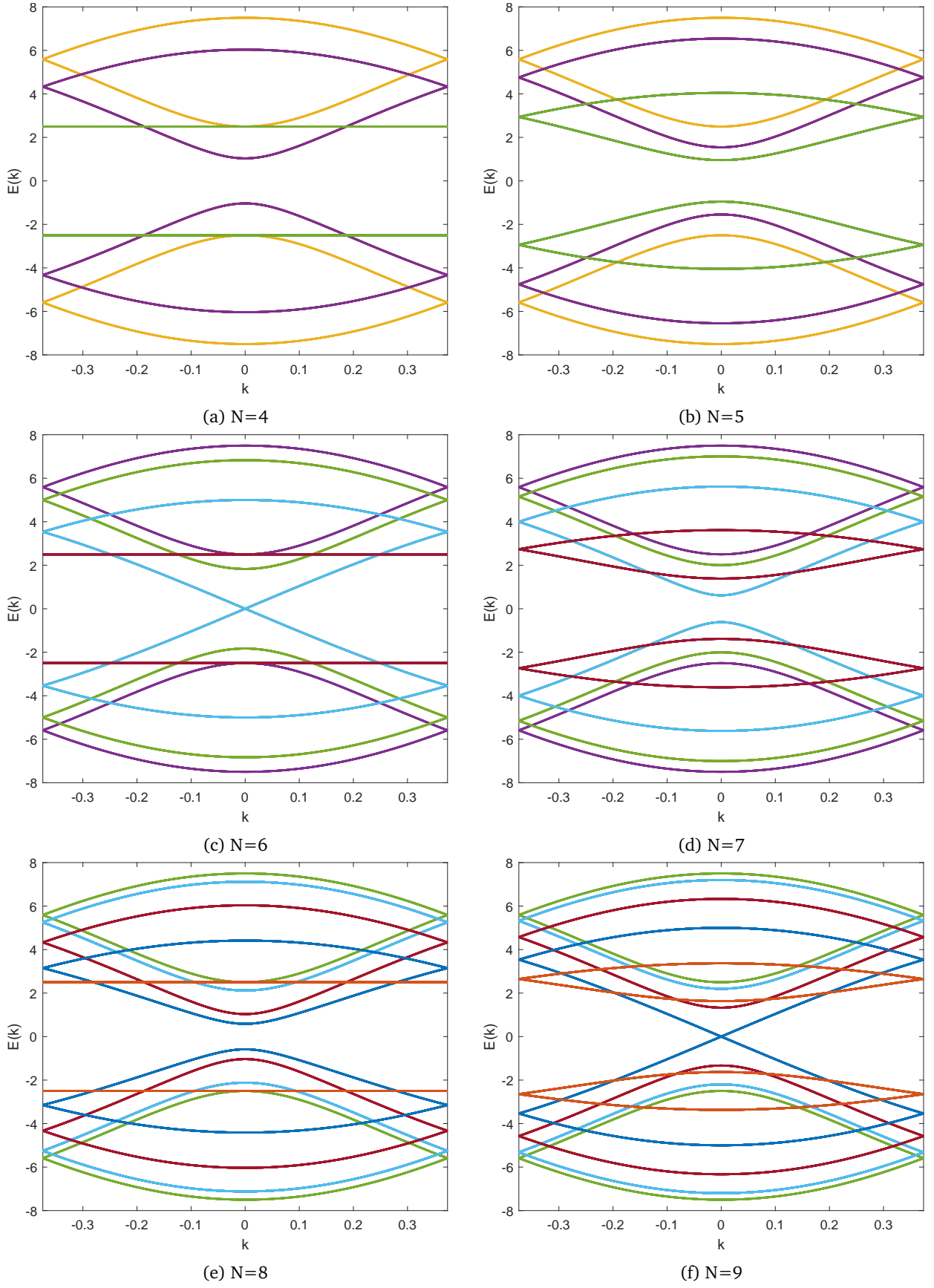


Figure 16: Electronic band structure of ZCNT for different size of ring. N representing number of unit cells in one ring. Horizontal axis is K , which is simply k_x in case of zigzag because it is folded along k_y . Vertical axis is energy in electronvolts (eV) unit and colors represent different k_y values.

MATLAB Scripts

Listing 1: Matlab script for Graphene Electronic Structure

```
1 a = 2.8; % Lattice parameter
2 b_1 = [1,0].*a; %%%%%%%%%%%%%%%
3 b_2 = [-0.5, sqrt(3)/2]*a; % Bond displacements %
4 b_3 = [-0.5, -sqrt(3)/2]*a; %%%%%%%%%%%%%%%
5
6 eps = 0; % -10.7 eV to vacuum but since eps = EF and we want to set EF = 0...
7 t = 2.5; % eV So we'll keep the energies in electronvolt
8
9 kmin = -pi/a; % Times 10 to calculate DOS
10 kmax = pi/a; % Times 10 to calculate DOS
11 step = 0.01;
12 number_of_ks = ceil((kmax - kmin)/step);
13 [k_1,k_2] = meshgrid(kmin:step:kmax,kmin:step:kmax);
14
15 Scos = cos(k_1*b_1(1)+k_2*b_1(2))+cos(k_1*b_2(1)+ k_2*b_2(2))+cos(k_1*b_3(1)+k_2*
    b_3(2));
16 Ssin = sin(k_1*b_1(1)+k_2*b_1(2))+sin(k_1*b_2(1)+ k_2*b_2(2))+sin(k_1*b_3(1)+k_2*
    b_3(2));
17
18 Ec = eps + t*sqrt(Scos.^2 + Ssin.^2);
19 Ev = eps - t*sqrt(Scos.^2 + Ssin.^2);
20
21 figure('name','Band Structure of Graphene')
22 surf(k_1,k_2,Ec);
23 hold on
24 surf(k_1,k_2,Ev)
25 colormap summer
26 hold off
27 print(gcf,'Band Structure','-bestfit','-dpdf','-r720')
28
29 %%%%%%%%%%%%%%%
30 % Density of States %
31 %%%%%%%%%%%%%%%
32 number_of_bins = 50;
33
34 % Counting the levels in a energy bin (E, E+dE)
35 [N_v,edgesv] = histcounts(Ev(:),number_of_bins);
36 [N_c,edgesc] = histcounts(Ec(:),number_of_bins);
37
38 % Normalization
39 D_v = N_v * (number_of_ks)^(-2);
40 D_c = N_c * (number_of_ks)^(-2);
41 N_tot = (sum(D_c) + sum(D_v)) % Check: has to be equal to 2 (#Bands)
42
43 % Bin centers
44 C = zeros(2,number_of_bins);
45 for i = 1:number_of_bins
46     C(1,i) = (edgesv(i)+edgesv(i+1))/2;
47     C(2,i) = (edgesc(i)+edgesc(i+1))/2;
48 end
49
50 figure('name','Density of States of Graphene')
51 plot(C(1,1:number_of_bins),D_v, '.')
52 hold on
53 plot(C(2,1:number_of_bins),D_c, '.')
54 print(gcf,'DOS','-bestfit','-dpdf','-painters')
55
56 %%%%%%%%%%%%%%%
57 % Gaussian Smearing of the DOS %
58 %%%%%%%%%%%%%%%
59
60 sp = 0.1; % Smearing Parameter
61
```

```

62 E = linspace(-8,8,500);
63 g_v = SMEARING(D_v,sp,C(1,:),E);
64 g_c = SMEARING(D_c,sp,C(2,:),E);
65
66 plot(E,g_v+g_c)
67
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69 % Normalized Gaussian
70 function y = Gauss(x,mean,dev)
71     y = 1/(sqrt(2*pi)*dev)*exp(-1/2*((x-mean)/dev).^2);
72 end
73 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74 % Gaussian Smearing
75 function y = SMEARING(v,dev,centers,domain)
76     y = zeros(1,length(domain));
77     for i = 1:length(domain)
78         for j = 1:length(v)
79             y(i)= y(i) + v(j)*Gauss(domain(i),centers(j),dev)/length(v);
80         end
81     end
82 end
83 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Listing 2: Matlab script for ACNT Full Matrix Method

```

1 a = 2.8; % Lattice parameter
2 ax = 3*a;
3 ay = sqrt(3)*a;
4 b_1 = [1,0]*a; %%%%%%%%%%%%%%%
5 b_2 = [-0.5, sqrt(3)/2]*a; % Bond displacements %
6 b_3 = [-0.5, -sqrt(3)/2]*a; %%%%%%%%%%%%%%%
7 eps = 0; %-10.7 eV to vacuum but since eps = EF and we want to set EF = 0...
8 t = 2.5; %eV So we'll keep the energies in electronvolt
9
10 M = 250;
11 ky_step = 2*pi/(ay*M);
12 ky_min= -pi/ay;
13 ky_max= +pi/ay;
14
15 Nmax=15;
16
17
18 for N = 2:1:Nmax
19
20     k_y = [ky_min:ky_step:ky_max];
21     eigs = zeros(length(k_y),4*N);
22
23     for j = 1:length(k_y)
24         ky = k_y(j);
25
26         H_cell = zeros(4,4);
27         H_cell(1,1) = eps;
28         H_cell(2,2) = eps;
29         H_cell(3,3) = eps;
30         H_cell(4,4) = eps;
31         H_cell(1,2) = -t*exp(1i*ky*b_1(2));
32         H_cell(2,3) = -t*(exp(-1i*ky*b_2(2))+exp(-1i*ky*b_3(2)));
33         H_cell(3,4) = -t*(exp(1i*ky*b_1(2)));
34         H_cell(3,2) = conj(H_cell(2,3));
35         H_cell(2,1) = conj(H_cell(1,2));
36         H_cell(4,3) = conj(H_cell(3,4));
37
38         H = zeros(N*4,N*4);
39
40         for ii= 1:N
41             H((ii-1)*4+1:(ii)*4, (ii-1)*4+1:(ii)*4) = H_cell;
42         end

```

```

43     PBC_Element = -t*(exp(1i*ky*b_2(2))+exp(1i*ky*b_3(2)));
44     H(1,N*4) = PBC_Element;
45     H(N*4,1) = conj(PBC_Element);
46
47
48     for ii = 2:N
49         H(4*(ii-1),4*(ii-1)+1)= PBC_Element;
50         H(4*(ii-1)+1,4*(ii-1)) = conj(PBC_Element);
51
52     end
53
54     eigs(j,:) = eig(H);
55
56 end
57
58 figure(N)
59 for n = 1:length(eigs(1,:))
60     plot(k_y,eigs(:,n),'black','LineWidth',1.5)
61     xlabel('k')
62     ylabel('E(k)')
63     xlim([ky_min,ky_max]);
64     hold on
65 end
66 set(gcf,'papersize',[6 5])
67 print(gcf,sprintf('Armchair1DBands%d',N),'-fillpage','-dpdf','-painters')
68 end

```

Listing 3: Matlab script for ACNT Reduced Matrix Method

```

1 a = 2.8; % Lattice parameter
2 ax = 3*a;
3 ay = sqrt(3)*a;
4 b_1 = [1,0]*a; %%%%%%%%%%%%%%
5 b_2 = [-0.5, sqrt(3)/2]*a; % Bond displacements %
6 b_3 = [-0.5, -sqrt(3)/2]*a; %%%%%%%%%%%%%%
7 eps = 0; %-10.7 eV to vacuum but since eps = EF and we want to set EF = 0...
8 t = 2.5; %eV So we'll keep the energies in electronvolt
9
10 M = 250;
11 ky_step = 2*pi/(ay*M);
12 ky_min= -pi/ay;
13 ky_max= +pi/ay;
14
15 Nmax=15;
16
17
18 for N = 15:1:Nmax
19
20     figure(N)
21
22     kx_step = 2*pi/(ax*N);
23     if mod(N,2) == 0
24         [k_x,k_y] = meshgrid(-N/2*kx_step:1*kx_step:N/2*kx_step, ky_min:ky_step:
ky_max);
25     else
26         [k_x,k_y] = meshgrid(-(N-1)/2*kx_step:1*kx_step:(N-1)/2*kx_step, ky_min:
ky_step:ky_max);
27     end
28     eigs = zeros(length(k_y(:,1)),4);
29
30     for i = 1:length(k_x(1,:))
31
32         for j = 1:length(k_y(:,1))
33             kx=k_x(1,i);
34             ky=k_y(j,1);
35
36             H_cell = zeros(4,4);

```

```

37     H_cell(1,1) = eps;
38     H_cell(2,2) = eps;
39     H_cell(3,3) = eps;
40     H_cell(4,4) = eps;
41     H_cell(1,2) = -t*exp(1i*(kx*b_1(1)+ky*b_1(2)));
42     H_cell(2,3) = -t*(exp(-1i*(kx*b_2(1)+ky*b_2(2)))+exp(-1i*(kx*b_3(1)+
    ky*b_3(2))));
43     H_cell(3,4) = -t*(exp(1i*(kx*b_1(1)+ky*b_1(2))));
44     H_cell(3,2) = conj(H_cell(2,3));
45     H_cell(2,1) = conj(H_cell(1,2));
46     H_cell(4,3) = conj(H_cell(3,4));
47
48     PBC_Element = -t*(exp(1i*(kx*b_2(1)+ky*b_2(2)))+exp(1i*(kx*b_3(1)+ky*
    b_3(2))));
49     H_cell(1,4) = PBC_Element;
50     H_cell(4,1) = conj(PBC_Element);
51
52     eigs(j,:) = eig(H_cell);
53
54     end
55
56     CM = lines(length(k_x(1,:)));
57     for jj = 1:length(eigs(j,:))
58         plot(k_y,eigs(:,jj),'-','color',CM(i,:), 'LineWidth',1.5)
59         xlabel('k')
60         ylabel('E(k)')
61         xlim([ky_min,ky_max]);
62         hold on
63     end
64
65     end
66
67     set(gcf,'papersize',[6 5])
68     print(gcf,sprintf('Armchair2DBands%d',N),'-fillpage','-dpdf','-painters')
69
70 end

```

Listing 4: Matlab script for ZCNT Full Matrix Method

```

1  a = 2.8; % Lattice parameter
2  ax = 3*a;
3  ay = sqrt(3)*a;
4  b_1 = [1,0]*a; %%%%%%%%%%%%%%
5  b_2 = [-0.5, sqrt(3)/2]*a; % Bond displacements %
6  b_3 = [-0.5, -sqrt(3)/2]*a; %%%%%%%%%%%%%%
7  eps = 0; %-10.7 eV to vacuum but since eps = EF and we want to set EF = 0...
8  t = 2.5; %eV So we'll keep the energies in electronvolt
9
10 M = 250;
11 kx_step = 2*pi/(ax*M);
12 kx_min= -pi/ax;
13 kx_max= +pi/ax;
14
15 Nmax=15;
16
17
18 for N = 2:1:Nmax
19
20     k_x = [kx_min:kx_step:kx_max];
21     eigs = zeros(length(k_x),4*N);
22
23     for j = 1:length(k_x)
24         kx=k_x(j);
25
26         H_cell = zeros(4,4);
27         H_cell(1,1) = eps;
28         H_cell(2,2) = eps;

```

```

29     H_cell(3,3) = eps;
30     H_cell(4,4) = eps;
31     H_cell(1,2) = -t*exp(-1i*(kx*b_3(1)));
32     H_cell(1,4) = -t*exp(-1i*(kx*b_1(1)));
33     H_cell(2,3) = -t*exp(+1i*(kx*b_1(1)));
34     H_cell(3,4) = -t*exp(-1i*(kx*b_2(1)));
35     H_cell(3,2) = conj(H_cell(2,3));
36     H_cell(2,1) = conj(H_cell(1,2));
37     H_cell(4,3) = conj(H_cell(3,4));
38     H_cell(4,1) = conj(H_cell(1,4));
39
40     H = zeros(N*4,N*4);
41
42     for ii= 1:N
43         H((ii-1)*4+1:(ii)*4, (ii-1)*4+1:(ii)*4) = H_cell;
44     end
45
46     PBC_Element25 = -t*exp(+1i*(kx*b_2(1)));
47     PBC_Element38 = -t*exp(-1i*(kx*b_3(1)));
48
49     H(1,N*4-2) = conj(PBC_Element25);
50     H(N*4-2,1) = PBC_Element25;
51
52     H(4,N*4-1) = conj(PBC_Element38);
53     H(N*4-1,4) = PBC_Element38;
54
55     for ii = 2:N
56         H(4*(ii-1)-2, 4*(ii-1)+1)= PBC_Element25;
57         H(4*(ii-1)+1, 4*(ii-1)-2) = conj(PBC_Element25);
58         H(4*(ii-1)-1, 4*(ii-1)+4)= PBC_Element38;
59         H(4*(ii-1)+4, 4*(ii-1)-1) = conj(PBC_Element38);
60
61     end
62
63     eigs(j,:) = eig(H);
64
65 end
66
67 figure(N)
68 for n = 1:length(eigs(1,:))
69     plot(k_x,eigs(:,n),'black','LineWidth',1.5)
70     xlabel('k')
71     ylabel('E(k)')
72     xlim([kx_min,kx_max]);
73     hold on
74 end
75
76 set(gcf,'papersize',[6 5])
77 print(gcf,sprintf('ZigZag1DBands%d',N),'-fillpage','-dpdf','-painters')
78 end

```

Listing 5: Matlab script for ZCNT Reduced Matrix Method

```

1  a = 2.8; % Lattice parameter
2  ax = 3*a;
3  ay = sqrt(3)*a;
4  b_1 = [1,0]*a; %%%%%%%%%%%%%%
5  b_2 = [-0.5, sqrt(3)/2]*a; % Bond displacements %
6  b_3 = [-0.5, -sqrt(3)/2]*a; %%%%%%%%%%%%%%
7  eps = 0; %-10.7 eV to vacuum but since eps = EF and we want to set EF = 0...
8  t = 2.5; %eV So we'll keep the energies in electronvolt
9
10 M = 250;
11 kx_step = 2*pi/(ax*M);
12 kx_min= -pi/ax;
13 kx_max= +pi/ax;
14

```

```

15 Nmax=15;
16
17 for N = 2:1:Nmax
18
19     figure(N)
20
21     ky_step = 2*pi/(ay*N);
22     if mod(N,2) == 0
23         [k_y,k_x] = meshgrid(-N/2*ky_step:1*ky_step:N/2*ky_step, kx_min:kx_step:
24 kx_max);
25     else
26         [k_y,k_x] = meshgrid(-(N-1)/2*ky_step:1*ky_step:(N-1)/2*ky_step, kx_min:
27 kx_step:kx_max);
28     end
29     eigs = zeros(length(k_x(:,1)),4);
30
31     for i = 1:length(k_y(1,:))
32
33         for j = 1:length(k_x(:,1))
34
35             ky=k_y(1,i);
36             kx=k_x(j,1);
37
38             H_cell = zeros(4,4);
39             H_cell(1,1) = eps;
40             H_cell(2,2) = eps;
41             H_cell(3,3) = eps;
42             H_cell(4,4) = eps;
43             H_cell(1,2) = -t*exp(-1i*(kx*b_3(1)+ky*b_3(2)));
44             H_cell(1,4) = -t*exp(-1i*(kx*b_1(1)+ky*b_1(2)));
45             H_cell(2,3) = -t*exp(+1i*(kx*b_1(1)+ky*b_1(2)));
46             H_cell(3,4) = -t*exp(-1i*(kx*b_2(1)+ky*b_2(2)));
47             H_cell(3,2) = conj(H_cell(2,3));
48             H_cell(2,1) = conj(H_cell(1,2));
49             H_cell(4,3) = conj(H_cell(3,4));
50             H_cell(4,1) = conj(H_cell(1,4));
51
52             PBC_Element25 = -t*exp(+1i*(kx*b_2(1)+ky*b_2(2)));
53             PBC_Element38 = -t*exp(-1i*(kx*b_3(1)+ky*b_3(2)));
54
55             H_cell(1,2) = H_cell(1,2) + conj(PBC_Element25);
56             H_cell(2,1) = conj(H_cell(1,2));
57
58             H_cell(3,4) = H_cell(3,4) + PBC_Element38;
59             H_cell(4,3) = conj(H_cell(3,4));
60
61             eigs(j,:) = eig(H_cell);
62
63         end
64
65         CM = lines(length(k_y(1,:)));
66         for jj = 1:length(eigs(j,:))
67             plot(k_x,eigs(:,jj),'-','color',CM(i,:), 'LineWidth',1.5)
68             xlabel('k')
69             ylabel('E(k)')
70             xlim([kx_min,kx_max]);
71             hold on
72         end
73     end
74
75     set(gcf,'papersize',[6 5])
76     print(gcf,sprintf('ZigZag2DBands%d',N),'-fillpage','-dpdf','-painters')
77 end

```