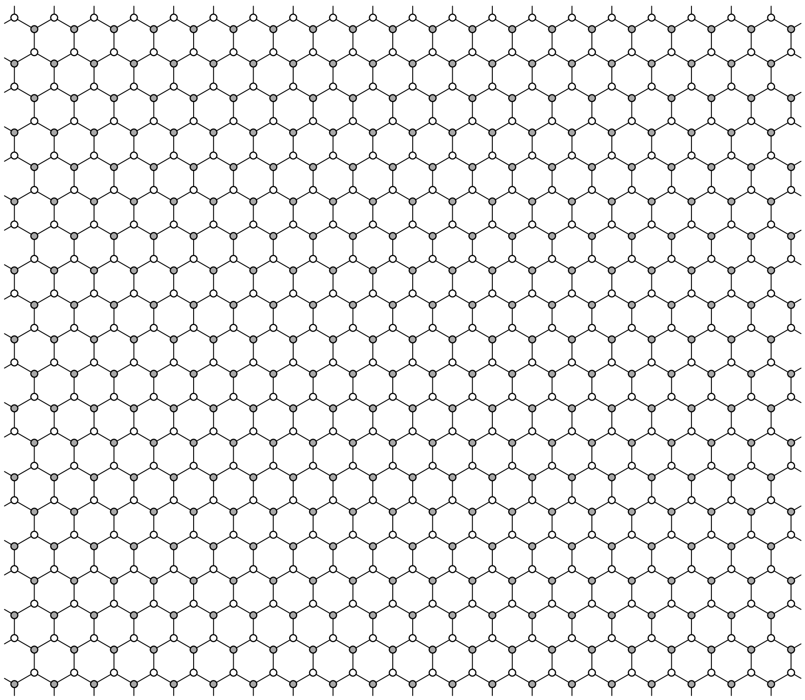
Physics 750: Assignment 3

due Tuesday, November 14, 2017

Your task in this assignment is to simulate a two-dimensional gas of noninteracting fermions living on a honeycomb lattice. This can be viewed in two ways. It's reasonable both as a model of electrons in graphene—the carbon membrane that serves as the building block for nanotubes and buckyballs—and as a model of massless, relativistic particles treated using a spatial discretization in the manner of lattice QCD.



For concreteness, we make the lattice finite and periodic with $N = 2L^2$ sites covering an area $\sqrt{3}L \times \frac{3}{2}L$. The Hamiltonian of this system can be expressed at the single-particle level as a real, symmetric $N \times N$ matrix whose rows and columns are indexed by the sites of the lattice. The matrix has nonzero entries only where two sites are connected:

$$H_{ij} = \begin{cases} -t & \text{if sites } i \text{ and } j \text{ and neighbours} \\ 0 & \text{otherwise} \end{cases}$$

For the infinite honeycomb sheet, this leads to an energy

$$E(\boldsymbol{k}) = \pm\frac{\sqrt{3}t}{2}|\boldsymbol{k}a| + O(k_x^2a^2, k_y^2a^2)$$

that—in the long-wavelength limit at least—disperses linearly with momentum $\hbar\boldsymbol{k}$ (as measured from the node of the light cone). In particular, so long as $\lambda = 2\pi/k = 2\pi/|\boldsymbol{k}|$ is much larger than the distance a between lattice sites, the system is largely unaware of the underlying spatial discretization. In the conventional vacuum, the antiparticle ($-\hbar kc$) branch is occupied and the particle ($+\hbar kc$) branch is empty (corresponding to a half-filled band in graphene).

To start, download and unpack the `assignment3.tar.gz` archive from the class website. In the `assignment3` directory there is a file named `graphene.cpp`. If you compile it and run the resulting program, you’ll get an Encapsulated PostScript version of the figure on the previous page.

```
$ make
g++ -o graphene graphene.cpp -O2 -ansi -pedantic -Wall
$ ./graphene
$ display config.eps // or "open config.eps" on MacOS
$ convert config.eps config.pdf
$ evince config.pdf // or "open config.pdf" on MacOS
```

Add to `graphene.cpp` whatever code is necessary to complete the following questions.

1. (5 points) The first action in `main` is a call to `build_honeycomb_lattice()`, which populates the container `hopping_list` with site index pairs that encode the nearest-neighbour connectivity of the lattice. Find the definition of `hopping_list` and trace back the class definitions until you understand exactly how it is structured.

   To construct the Hamiltonian, walk through `hopping_list` and for each element assign the relevant entry in `H`. You should work in units where $t = a = 1$. Note that `H` is a linear array into which the Hamiltonian matrix must be flattened. Do so in a way that is consistent with one of the packed storage schemes supported by LAPACK.

2. (15 points) Complete the `eigensolve` function so that it correctly solves the eigenvalue problem for `H` using LAPACK’s dspevd routine, putting the resulting eigenvalues and eigenvectors into `Eval` and `Evec`. You’ll have to assign various kinds of temporary storage. It’s fine if your function destroys `H` in the process. If you’ve done everything correctly, you should be able to run the code without receiving any warnings like the following.

   ```
   $ ./graphene 2
   There’s a problem with the eigensolver
   ```

3. (5 points) Make a histogram of the spectrum of eigenenergies for each of the lattice sizes $L = 10, 20, 30, 40$. Compare these to the density of states on the infinite lattice:

   $$D(E) = \sum_{\boldsymbol{k},\pm} \delta(E \pm |E(\boldsymbol{k})|) = \frac{\sqrt{3}}{3\pi t^2}|E|\biggl(1 + \frac{E^2}{3t^2} + \frac{5E^4}{27t^4} + \frac{31E^6}{243t^6} + \cdots\biggr).$$

   What you’re seeing here are energy slices of the light cone for particles ($E > 0$) and antiparticles ($E < 0$).

   ```
   $ ./graphene 3
   $ gnuplot
   > plot "dos.dat" using 1:2 w l
   ```

4. (5 points) The last section of code in `main` computes the overlap of a guassian packet with each of the empty particle levels. It then reconstructs the wavefunction and propagates it forward in time. In the default setup, the wavepacket has no initial momentum, and it dies away by spreading out into incoherence. Compile with `L = 30` and watch the movie that is generated.

   ```
   $ ./graphene 4
   $ gnuplot movie.gp
   ```

   Explore the evolution of the wavepacket for different values of the variables `kx` and `ky`. Write code that computes the average position of the wavepacket as a function of time. Create a plot showing distance travelled versus time for initial momentum $k_x = \eta K_1 \cos\theta$ and $k_y = \eta K_1 \sin\theta$ (for $K_1 = 4\pi/3\sqrt{3}$) and with θ taking on a few values between 0 and 2π and η a few between 0.1 and 0.7. How does the wavepacket speed (the scalar group velocity) depend on θ and η? Provide an explanation for what you observe.