Exercise_TI-01

January 11, 2023

[ ]:
```python
import sisl
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import functools
```

1 Quantum Hall Effect

In this exercise, we will build on **TB_07** and **A_03** to simulate the quantum Hall effect. In **A_03**, we learned how to created a Hall bar device and in **TB_07**, how to deal with magnetic fields.

Here, we will extract the Hall resistance from the transmissions calculated with `TBtrans` using the Landauer-Büttiker formalism.

1.1 Exercise Overview:

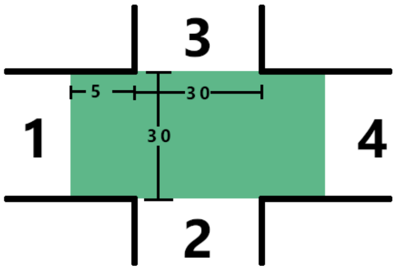
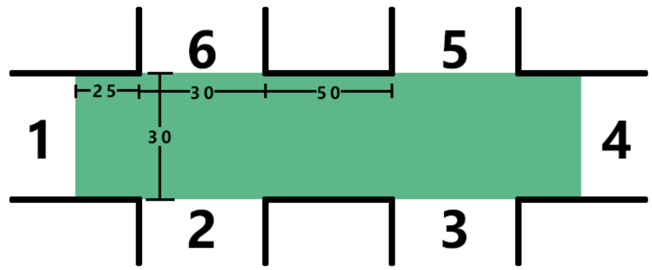
1. Create a Hall bar (see **A_03**)
2. Construct Hamiltonians and add magnetic fields (see **TB_07**)
3. Calculate the transmission with `TBtrans`
4. Extract the Hall resistance (R_H).
5. Extract the logitudinal resistance (R_L).

1.2 Exercise

1.2.1 1. Create a Hall bar

In order to be able to observe the quantum Hall effect, the size of the Hall bar needs to be big enough. For a 4(6) lead device reasonable dimensions are: 1. 4-lead device (square lattice): - Width of electrodes (perpendicular to the semi-infinite axis): 30 atoms - Offset of the electrodes 2(3) from the corner of the central part: >= 5 atoms

2. 6-lead device (square lattice):
   - Width of electrodes (perpendicular to the semi-infinite axis): 30 atoms
   - Spacing between electrodes on the same side: 50 atoms
   - Offset of the electrodes 2(3,5,6) from the corner of the central part: > 25 atoms
3. 6-lead Graphene Hall bar:
   - Width of electrodes (perpendicular to the semi-infinite axis): ~30 Å
   - Spacing between electrodes on the same side: ~50 Å
   - Offset of the electrodes 2(3,5,6) from the corner of the central part: ~ 15 Å

4-lead device	6-lead device
	

```
[ ]: # Create a Hall bar
```

1.2.2 2. Construct Hamiltonian and add magnetic fields

The required field strengths may vary depending on the size of the Hall bar. We should start with a corse grid, and create a finer grid once we have identified the correct range. A good starting point might by `B = 1 / np.arange(10,31)`.

```
[ ]: def peierls(self, ia, atoms, atoms_xyz=None, B=None):
         idx = self.geometry.close(ia, R=[0.1, 1.01], atoms=atoms, atoms_xyz=atoms_xyz)
         # Onsite
         self[ia, idx[0]] = 4

         # Hopping
         if B == 0:
             self[ia, idx[1]] = -1
         else:
             xyz = self.geometry.xyz[ia]
             dxyz = self.geometry[idx[1]]
             self[ia, idx[1]] = - np.exp(-0.5j * B * (dxyz[:, 0] - xyz[0])*(dxyz[:,1] + xyz[1]))
```

```
[ ]: H0 = sisl.Hamiltonian(geom, dtype=np.float64)
     H0.construct(functools.partial(peierls, B=0.))

     HB = sisl.Hamiltonian(geom, dtype=np.complex128)

     rec_phis = np.arange(10,31)
     for rec_phi in rec_phis:
         HB.construct(functools.partial(peierls, B=1/rec_phi))
         dH = ...
```

1.2.3 3. Calculate the transmission with TBtrans

The folder of this exercise contains the skeleton of an input file for a 4-lead (RUN-4.fdf) and 6-lead device (RUN-6.fdf), as well as a script to run TBtrans for all values of the magnetic field (run.sh).

Depending on the size of the Hall bar, this step might require a considerable amount of time.

1.2.4 4. Extract the Hall resistance (R_H)

In Hall effect we measure the build up of a potential difference between the measurement electrodes as response to an electric current. The Hall resistance (R_H) in a 4 lead Hall bar like the one shown above is given by

$$R_H = \left.\frac{V_2 - V_3}{I_1}\right|_{I_2=I_3=0}.$$

In `TBtrans` the chemical potentials of all electrodes can be specified and the currents are calculated as a response to the biases. Rather than trying many combinations of chemical potentials to find the Hall resistance, we use transmission curves to calculate R_H.

To start, we express the lead currents I_i in terms of applied biases V_i and the transmissions T_{ij} between leads i and j

$$I_i = \sum_j G_{ij}(V_i - V_j) \quad \text{where} \quad G_{ij} = \frac{2e^2}{h}T_{ij}.$$

We rewrite this relation as

$$\mathbf{I} = \mathcal{G}\mathbf{V} \quad , \text{where} \quad \mathcal{G}_{ii} = \sum_{i\neq j} G_{ij} \quad \text{and} \quad \mathcal{G}_{ij} = -G_{ij}.$$

Since the currents only depend on bias differences, we can set one of them to zero without loss of generality (here $V_4 = 0$). Further, Kirchhoff's current law allow us to eliminate one of current (here $I_4 = -I_1 - I_2 - I_3$). This leaves us with an invertible 3×3 matrix equation.

Using the inverse $\mathbf{R}$ of $\mathcal{G}$, we can express V_2 and V_3 in terms of the lead currents I_i and calculate the Hall conductance:

$$\mathbf{V} = \mathcal{G}^{-1}\mathbf{I} = \mathbf{R}\mathbf{I} \quad \Rightarrow \quad V_i = R_{i1}I_1 + R_{i2}I_2 + R_{i3}I_3,$$

and finally, we find the Hall resistance:

$$R_H = \left.\frac{R_{21}I_1 + R_{22}I_2 + R_{23}I_3 - (R_{31}I_1 + R_{32}I_2 + R_{33}I_3)}{I_1}\right|_{I_2=I_3=0} \tag{1}$$

$$= R_{21} - R_{31} \tag{2}$$

The derivation for the 6-lead device is analogous and yields:

$$R_H = R_{21} - R_{61}$$

If everything is set up correctly, the quantization of the Hall resistance should be visible.

$$R_H = \frac{h}{2ne^2} \qquad \text{for square lattice Hall bar} \qquad n \in \mathbb{N} \tag{3}$$

$$R_H = \frac{h}{2(2n-1)e^2} \qquad \text{for graphene Hall bar} \qquad n \in \mathbb{N} \tag{4}$$

1.2.5 5. Extract longitudinal resistance R_L

The longitudinal resistance can be extracted using the same approach used for the Hall resistance. With a 6-lead Hall bar we can replace the V_3 with V_6 and get it immediately.

$$R_L = R_{21} - R_{31}$$

With a 4-lead Hall bar we need to create a new device with electrodes 2 and 3 on the same side of the Hall bar.

If the energy mesh in TBTrans and the mesh for magnetic field strength are fine engough, spikes in R_L should be observable at each step of R_H.

[ ]:
```
# Create short-hand function to open files
gs = sisl.get_sile
# No magnetic field
tbt0 = gs('M_0/siesta.TBT.nc')
# All magnetic fields in increasing order
tbts = [gs('M_{}/siesta.TBT.nc'.format(rec_phi)) for rec_phi in rec_phis]
```

[ ]:
```
def calc_G(tbtsile, n):
    # Construct G
    G = np.zeros((tbtsile.nE, n, n))
    for i in range(n):
        for j in range(n):
            if i == j: continue
            Gij = tbtsile.transmission(i,j)
            G[:,i,j] = -Gij
            G[:,i,i] += Gij
    return G

# Number of electrodes used in the Hall bar
nEl = 6

# Calculate Conductance matrix G
G = np.asarray([calc_G(tbt, nEl) for tbt in tbts])

# Remove one row and column from the matrix
referenceElectrode = 3
G = np.delete(np.delete(G, referenceElectrode, axis=2), referenceElectrode,␣
 ↪axis=3)
```

```python
# Invert G to get resistance matrix R
# Hint: If the matrix inversion might fail for some values of rec_phi, because
  ↪the magentic field is too strong.
R = np.linalg.inv(G)

if nEl == 6:
    # For 6-lead device
    RH = R[:,:,1,0] - R[:,:,4,0]   # R_21 - R_61
    RL = R[:,:,1,0] - R[:,:,2,0]   # R_21 - R_31
elif nEl == 4:
    # For 4-lead device
    RH = R[:,:,1,0] - R[:,:,2,0]   # R_21 - R_31
```

[ ]:
```python
# Plot the Hall resistance
#  - as a function of energy for a fixed magnetic field strength
#  - as a function of magnetic field strength for a fixed energy

# E_idx = ...
# phi_idx = ...

fig, axs = plt.subplots(1,2, sharey=True, figsize=(12,4))

axs[0].set_title('$\phi = {:.5f}$'.format(1/rec_phis[phi_idx]))
axs[0].set_ylabel(r'$R \left[h/2e^2\right]$')
axs[0].set_xlabel('$E/t$')
axs[0].axvline(x=E[E_idx], ls='--', c='k')
# axs[0].plot(...)

axs[1].set_title(f'E = {E[E_idx]:.5f}')
axs[1].axvline(x=1/rec_phis[phi_idx], ls='--', c='k')
axs[1].set_xlabel('$\phi/\phi_0$')
# axs[1].plot(...)

axs[1].legend(loc='center right', bbox_to_anchor=(1.5,0.5), ncol=1)

plt.show()
```