# Numerical simulation of quantum transport phenomena using the kwant package

Nagy Dániel

Final Presentation
16 May 2019

# 1 Schedule for the semester

Table 1: Status

| Week | Scheduled Task |
|---|---|
| feb. 18. - feb. 24. | Installing Kwant & Running an example ✓ |
| feb. 25. - mar. 3. | Reading the documentation & Running more examples ✓ |
| mar. 4 - mar. 10 | Struggling with graphene minimal conductivity - no result |
| mar. 11. - mar. 17. | 2DEG basics & Eigenstates and LDOS calculation ✓ |
| mar. 18. - mar. 24. | 2DEG in magnetic field ✓ |
| mar. 25. - mar. 31. | Mid term report ✓ |
| apr. 1. - apr. 7. | Topological Anderson Insulator 1. ✓ |
| apr. 8. - apr. 14. | Topological Anderson Insulator 2. ✓ |
| easter holiday | - |
| apr. 22. - apr. 28. | Topological Anderson Insulator 3. ✓ |
| apr. 29. - may 5. | Topological Anderson Insulator 4. ✓ |
| Eötvös/Pázmány days | - |
| may 13. - may 19. | Final report ✓ |

# Introduction

- What is Kwant?

  Kwant is a free (open source), powerful, and easy to use Python package for numerical calculations on tight-binding models with a strong focus on quantum transport [**?**, **?**]

- What You do with Kwant?

  You can define a general tight binding system: square lattice, bipartite lattice, graphene, etc.

  You can attach leads to the scattering region, then calculate transport properties

  You can calculate densities (spin-density, charge-density, density of states, etc.)

  You can calculate eigenvalues, eigenfunctions

  You can calculate currents (spin-current, charge current, etc.)

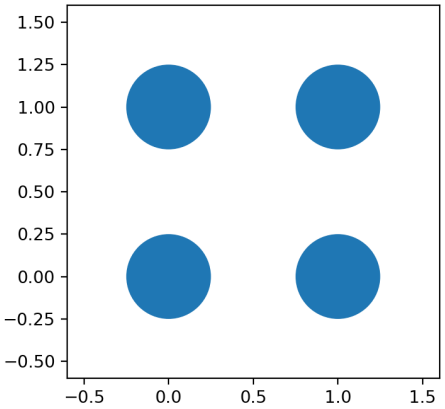
  Many more: [**?**].

# Defining a simple square lattice

```python
a = 1 # Set lattice constant
lat = kwant.lattice.square(a) # Create a lattice
syst = kwant.Builder() # Create a tight-binding system

# Specify onsite energies
syst[lat(0, 0)] = 2
syst[lat(1, 0)] = 2
syst[lat(0, 1)] = 2
syst[lat(1, 1)] = 2

# Plot the system
fig = kwant.plot(syst);
```



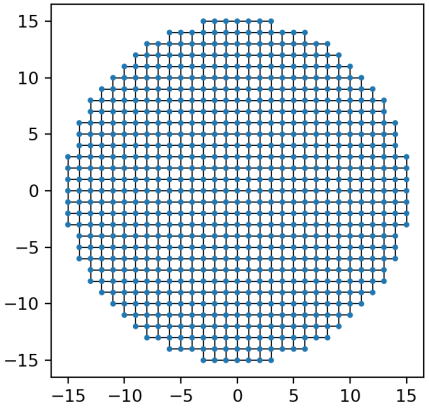
# Defining a circular disc of square lattice

```python
a = 1 # Set lattice constant
lat = kwant.lattice.square(a) # Create a lattice
syst = kwant.Builder() # Create a tight-binding system

# A shape can be defined using a function, that returns True, if a point is
# inside the shape, False otherwise
def circle(pos):
  x,y = pos
  return x**2 + y**2 < 240

# Fill the shape
syst[lat.shape(function=circle, start=(0,0))] = 4

# Adding hoppings to the lattice:
syst[lat.neighbors()] = t
```

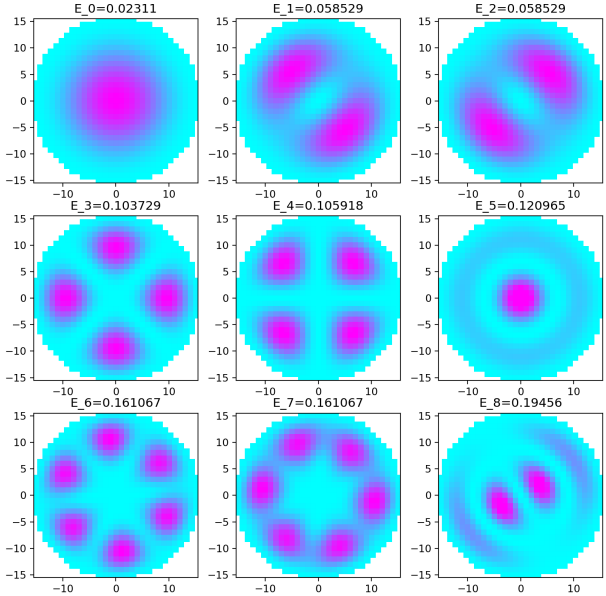


# Calculating the eigenfunctions

```
# Get the Hamiltonian
ham_mat = syst.finalized().hamiltonian_submatrix(sparse=True)

# Calculate eigenfunctions using scipy
evals, evecs = sla.eigsh(ham_mat.tocsc(), k=9, sigma=0)

# Plot the eigenfunctions using kwant.plotter.map
fig, axes = plt.subplots(3, 3, figsize=(10, 10), dpi=196)
for i in range(9):
  kwant.plotter.map(syst.finalized(),
                    np.abs(evecs[:, i])**2,
                    cmap='cool', background='w',
                    ax=axes[int(i/3)][i%3])
  axes[int(i/3)][i%3].set_title("E_{0}={1}".format(i, np.round(evals[i],6)))
```



# Calculating the transmission for a quantum point contact

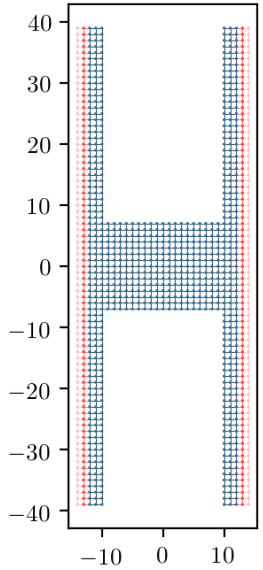



Figure 1: 2DEG system: a point contact

# Calculating the transmission for a quantum point contact

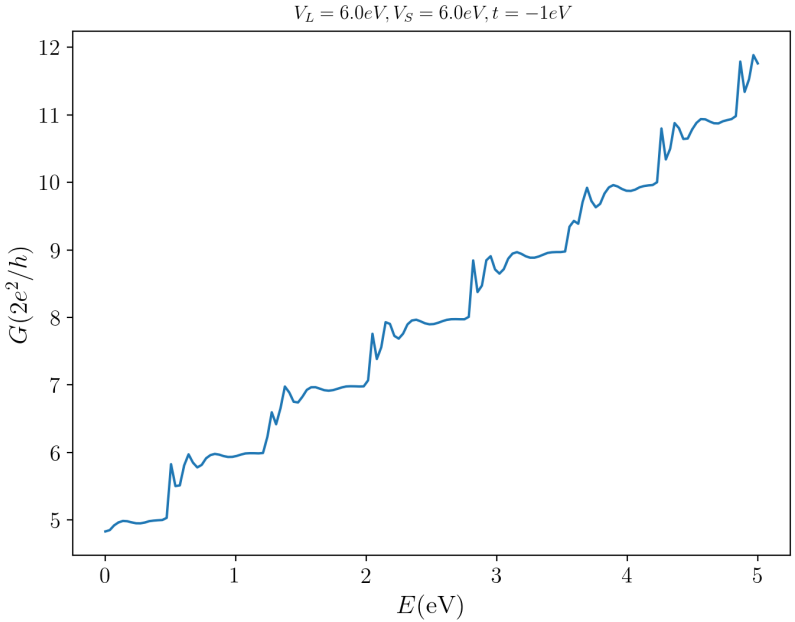



Figure 2: Transmission coefficient for a quantum point contact plotted against the energy.

# Calculating the shot-noise for a quantum point contact

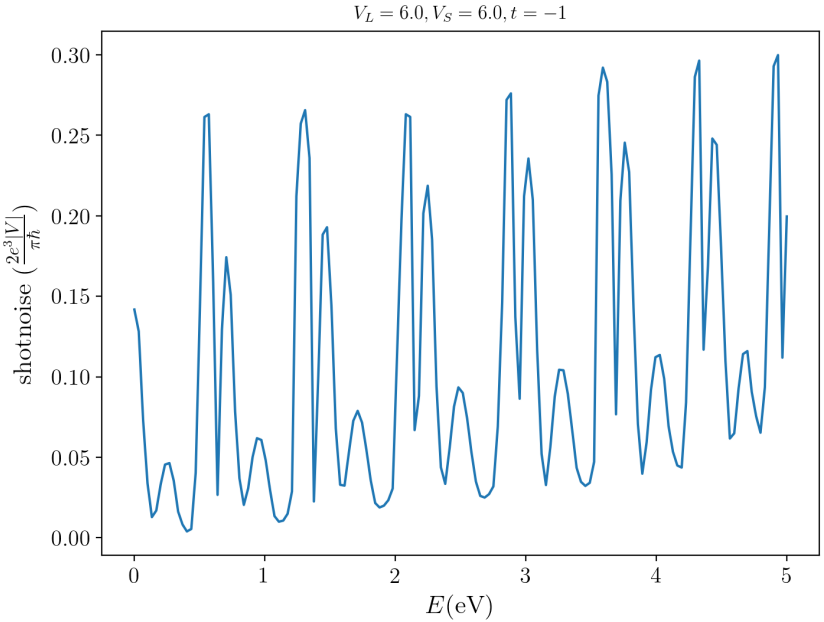



Figure 3: Shot-noise for a quantum point contact calculated with kwant.

# Adding magnetic field

If the system is placed in magnetic field, according to Peierls, we have to replace the hopping matrix elements:

$$t_{ij} \to t_{ij} \times \exp\left(i\frac{e}{\hbar}\int_{\mathbf{x}_j}^{\mathbf{x}_i}\mathbf{A}(\mathbf{x})d\mathbf{s}\right)$$

For 2D square lattices this can be written as

$$\exp\left(i\,2\pi\frac{\phi}{\phi_0}\frac{(y_i+y_j)(x_i-x_j)}{2a^2}\right)$$

where $\phi = Ba^2$ is the flux through a unit cell in the square lattice, and $\phi_0 = h/e$ the flux quantum.

## Adding magnetic field

```
# Define a hopping function:
def hopping(site_i, site_j, phi):
  xi, yi = site_i.pos
  xj, yj = site_j.pos
  return -exp( -0.5j * phi * (xi - xj) * (yi + yj) )

# Adding hoppings to the lattice:
syst[lat.neighbors()] = hopping
```

Figure 4: Adding Peierls-hoppings to the system

# Adding magnetic field

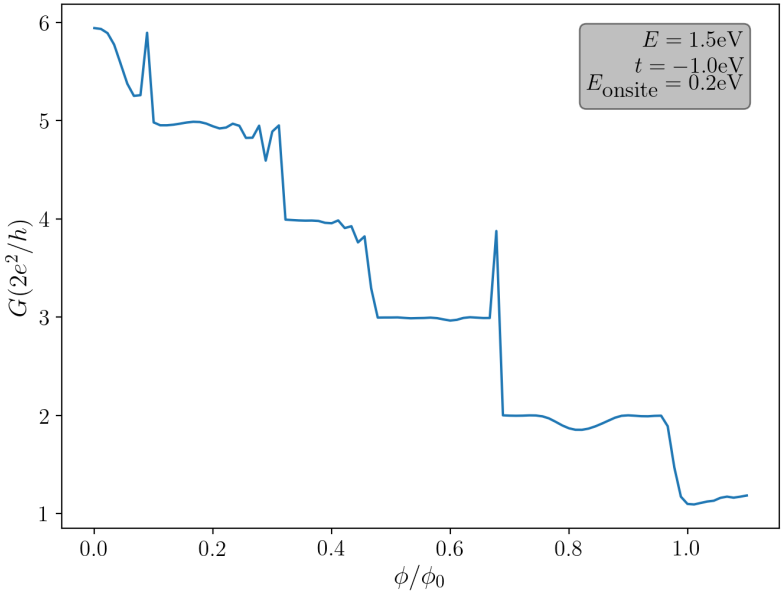



Figure 5: Transmission coefficient of a quantum point contact in function of the applied magnetic field

# Graphene minimal conductivity

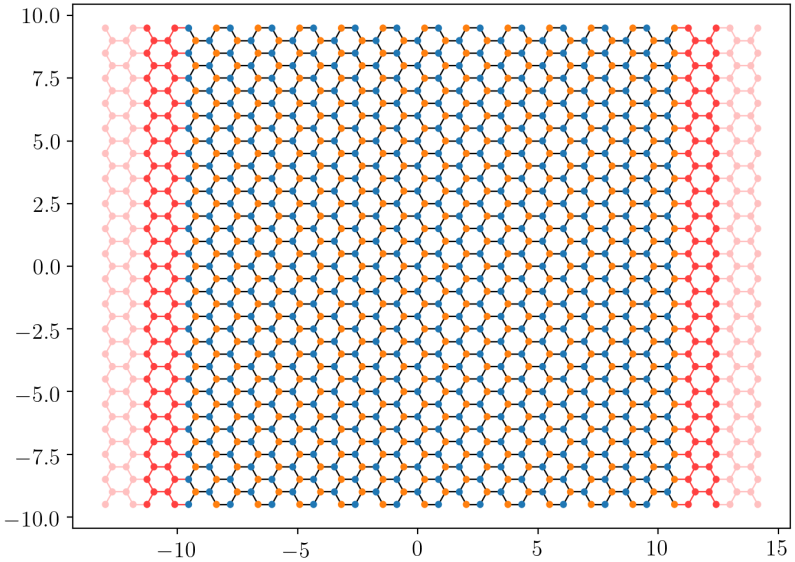



Figure 6: Graphene lattice defined in kwant

# Graphene minimal conductivity

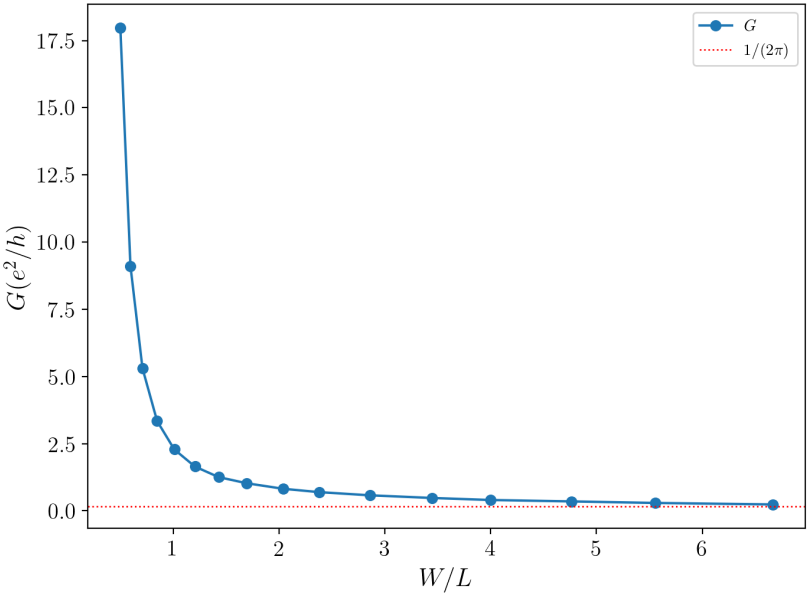



Figure 7: Graphene conductivity in function of $W/L$ ratio.

# Graphene minimal conductivity

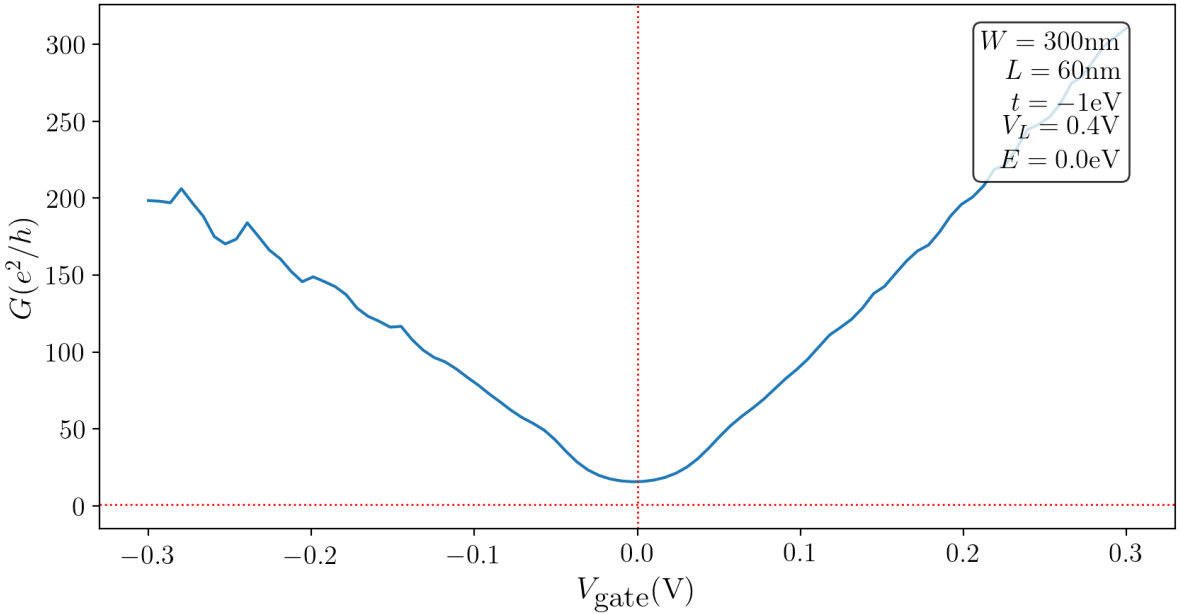
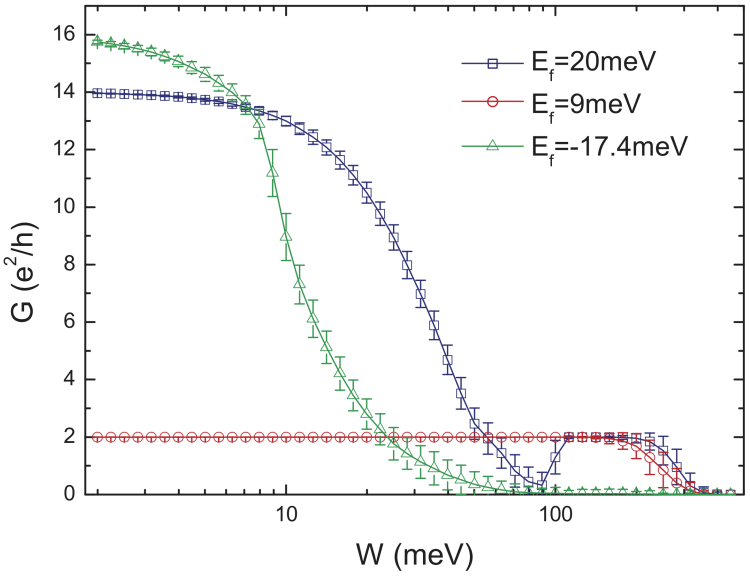



Figure 8: Graphene conductivity in function of energy.

# Topological Anderson Insulator

What are topological Andreson insulators? Main article [**?**]: https://arxiv.org/abs/0811.3045,

- The physics of a topological insulator is unaffected by weak disorder, but is destroyed by large disorder
- BUT: Disorder can create a topological insulator for parameters where the system was metallic in the absence of disorder
- These states are called topological Anderson insulators
- Disorder can be modeled as random on-site energy with a uniform distribution within $[-W/2, W/2]$
- the article discusses disordered strips of HgTe/CdTe quantum wells.
- Right: Results for a HgTe TAI, from the article [**?**]. The plateau starting at $W = 100, E_f = 20$meV is a signature of the Topological Anderson Insulator.



# Topological Anderson Insulator

- The paper about Topological Anders Insulator (https://arxiv.org/abs/0811.3045) defines the Hamiltonian for HgTe/CdTe heterostructure:

$$\mathcal{H}(\mathbf{k}) = \begin{pmatrix} h(\mathbf{k}) & 0 \\ 0 & h^{\dagger}(-\mathbf{k}) \end{pmatrix}$$

$$h(\mathbf{k}) = \epsilon(k) + \mathbf{d}(\mathbf{k}) \cdot \sigma, \ \mathbf{k} = (k_x, k_y)$$
$$\mathbf{d}(\mathbf{k}) = (Ak_x, Ak_y, M - Bk^2); \ \epsilon(k) = C - Dk^2$$

- This can be rewritten as a sum of a few terms:

$$= (C - Dk^2)\begin{pmatrix} \sigma_0 & 0 \\ 0 & \sigma_0 \end{pmatrix} + (M - Bk^2)\begin{pmatrix} \sigma_z & 0 \\ 0 & \sigma_z \end{pmatrix} + Ak_x\begin{pmatrix} \sigma_x & 0 \\ 0 & -\sigma_x \end{pmatrix} + Ak_y\begin{pmatrix} \sigma_y & 0 \\ 0 & -\sigma_y \end{pmatrix} + V_{\text{disorder}}\begin{pmatrix} \sigma_0 & 0 \\ 0 & \sigma_0 \end{pmatrix}$$

- A, B, C, D, M are empirical parameters. I used the values presented in an article [**?**].

# Topological Anderson Insulator

```
hamiltonian = """
(C-D*(k_x**2+k_y**2))*identity(4)
+ (M-B*(k_x**2+k_y**2))*kron(sigma_0, sigma_z)
+ A*k_x*kron(sigma_z, sigma_x)
+ A*k_y*kron(sigma_z, sigma_y)
+ V(x,y)*identity(4)
"""

template = kwant.continuum.discretize(hamiltonian, grid=a)

def shape(site):
  (x, y) = site.pos
  return (0 <= y < W and 0 <= x < L)

syst = kwant.Builder()
syst.fill(template, shape, (0, 0))
```

# Topological Anderson Insulator

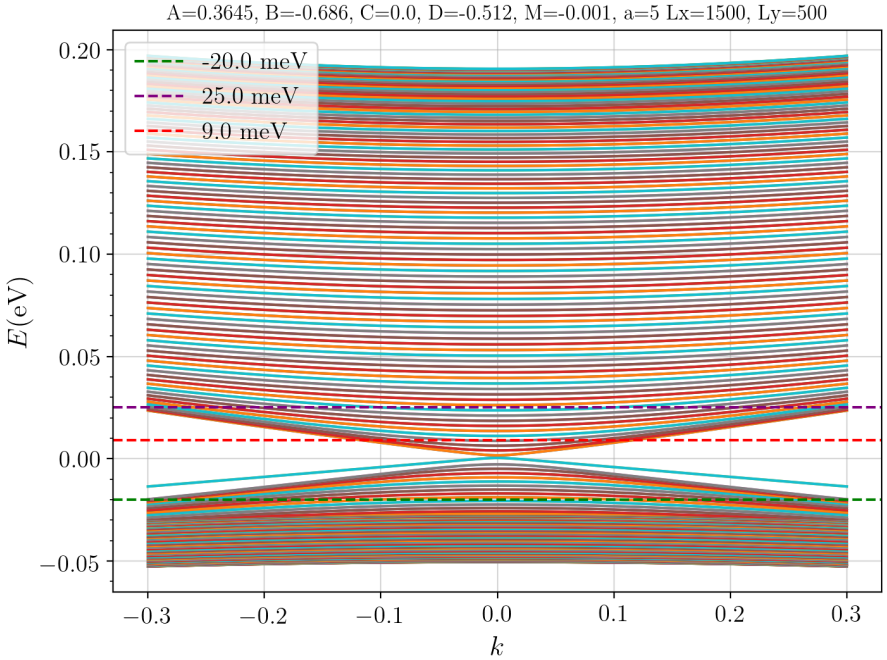



Figure 9: Band structure for the "inverted" quantum well, with $M = -1$meV

# Topological Anderson Insulator

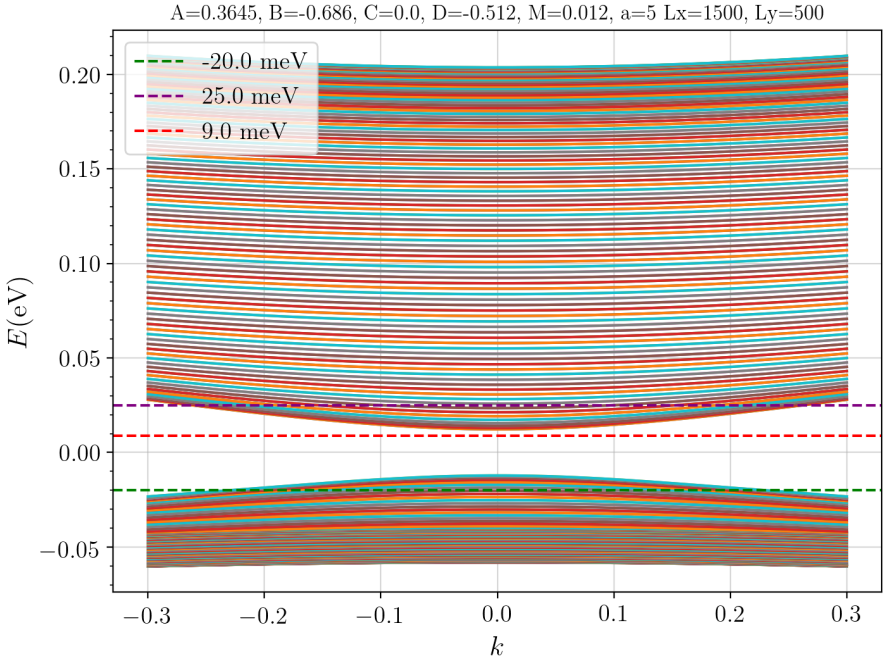



Figure 10: Band structure for the "normal" quantum well, with $M = 12$meV

# Topological Anderson Insulator

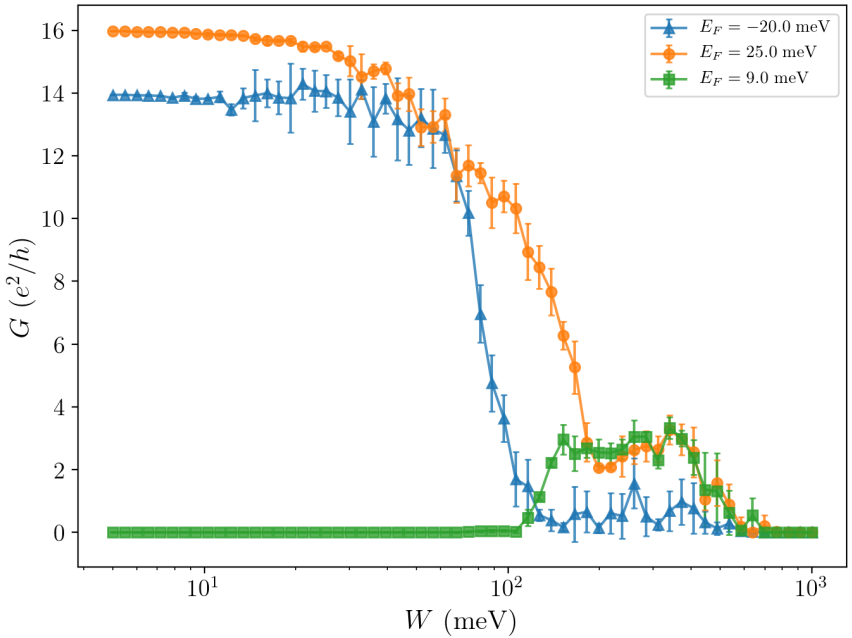
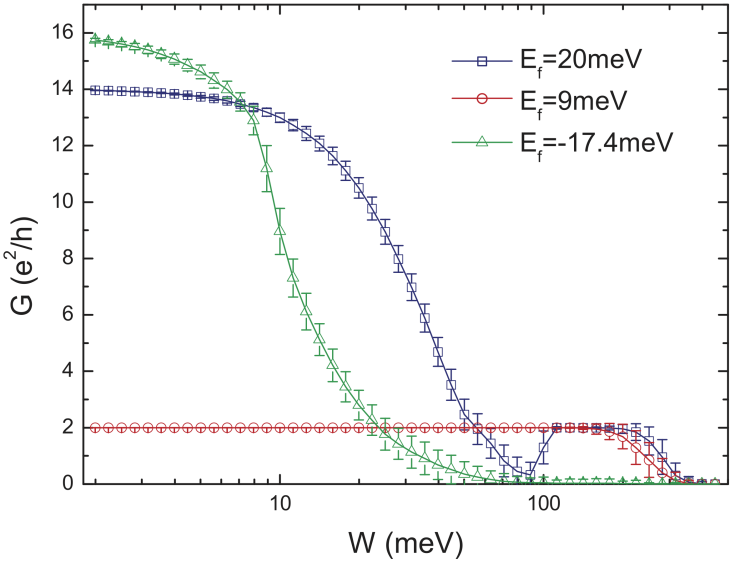
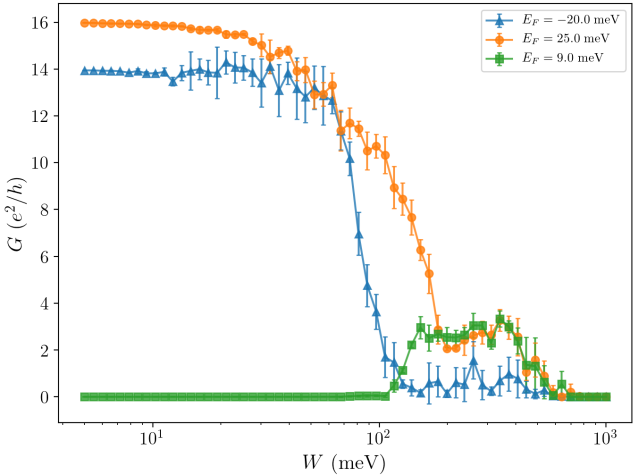



Figure 11: Conductivity plotted against disorder parameter for $M = 12$meV

# Topological Anderson Insulator



# References

[1] The official homepage of the kwant project: https://kwant-project.org.

[2] The official documentation for kwant 1.3.2 https://kwant-project.org/doc/1/.

[3] Supriyo Datta. Electronic transport in mesoscopic systems. Cambridge Studies in Semiconductor Physics and Microelectronic Engineering. Cambridge University Press, 1995.

[4] Christoph W. Groth, Michael Wimmer, Anton R. Akhmerov, and Xavier Waintal. Kwant: a software package for quantum transport. New Journal of Physics, 16:063065, Jun 2014.

[5] Hua Jiang, Lei Wang, Qing-Feng Sun, and X. C. Xie. Numerical study of the topological Anderson insulator in HgTe/CdTe quantum wells. Physical Review B, 80(16):165316, Oct 2009.

[6] Jian Li, Rui-Lin Chu, J. K. Jain, and Shun-Qing Shen. Topological Anderson Insulator. , 102(13):136806, Apr 2009.

[7] Thibaud Louvet, Pierre Delplace, Andrei A. Fedorenko, and David Carpentier. On the origin of minimal conductivity at a band crossing. Physical Review B, 92(15):155116, Oct 2015.

[8] J. Tworzydło, B. Trauzettel, M. Titov, A. Rycerz, and C. W. J. Beenakker. Sub-Poissonian Shot Noise in Graphene. , 96(24):246802, Jun 2006.

[9] H. van Houten and C. W. J. Beenakker. Quantum Point Contacts. arXiv e-prints, pages cond–mat/0512609, Dec 2005.

Thank You for listening!