Solid State Simulation Techniques.
Quantum Noise and High-Frequency Current Simulation.

Joaquín Gabriel Márquez Olguín

February 2026

Introduction

The aim of this homework is to extend our study of quantum transport beyond the mean current approximation used in our previous resonant tunneling diode (RTD) simulations. While the Landauer-Büttiker formalism provides an accurate description of the average DC current $\langle I \rangle$, real nanoscale devices exhibit intrinsic fluctuations around this mean, known as quantum noise. As detailed in the course notes, analyzing these fluctuations allows us to go beyond simple conductance and understand the correlation of charge carriers, like for instance how the Pauli exclusion principle suppresses noise (sub-Poissonian) compared to uncorrelated transport (Poissonian).

Furthermore, we address the limitations of the standard particle current model when operating at high frequencies. In dynamic regimes, the particle current alone is insufficient because it does not account for charge accumulation inside the device, leading to a violation of charge conservation. To resolve this, we introduce the concept of total current (the sum of particle and displacement currents), which restores continuity. Finally, we analyze the measurement process itself, treating the ammeter not as a projective detector that collapses the wavefunction, but as a "weak measurement" device that extracts information from the surroundings without disturbing the coherent quantum state inside the active region.

Exercise 1: Mean (particle) current

1.1. Computation of Mean Current.

Show analytically that the ensemble average of the current, defined as

$$\langle I \rangle = \sum_i I_i P(I_i), \tag{1}$$

is equivalent to the quantum mechanical expectation value:

$$\langle \Psi | \hat{I} | \Psi \rangle. \tag{2}$$

We begin with the statistical definition of the mean current. Let $\{\phi_i(x)\}$ be the set of orthonormal eigenfunctions of the current operator $\hat{I}$, such that $\hat{I}\phi_i(x) = I_i \phi_i(x)$. The state of the system $\Psi(x)$ (before measurement) can be expanded in this basis as follows:

$$|\Psi\rangle = \sum_i c_i |\phi_i\rangle, \quad \text{where } c_i = \langle \phi_i | \Psi \rangle. \tag{3}$$

The probability of measuring a specific current value I_i is given by the so-called Born rule, $P(I_i) = |c_i|^2 = c_i^* c_i$. Substituting this into Eq. (1):

$$\langle I \rangle = \sum_i I_i |c_i|^2 = \sum_i I_i c_i^* c_i.$$

Since I_i is a real scalar eigenvalue of the current operator, we can arrange terms to arrive to:

$$\langle I \rangle = \sum_i c_i^*(I_i c_i) = \sum_i \langle \Psi | \phi_i \rangle \, I_i \, \langle \phi_i | \Psi \rangle \, .$$

Recognizing that $\hat{I} | \phi_i \rangle = I_i | \phi_i \rangle$, and using the completeness relation of the basis states, $\sum_i | \phi_i \rangle \langle \phi_i | = \mathbb{I}$, we arrive at the final result:

$$\langle I \rangle = \sum_i \langle \Psi | \phi_i \rangle \langle \phi_i | \hat{I} | \Psi \rangle = \langle \Psi | \mathbb{I} \cdot \hat{I} | \Psi \rangle = \langle \Psi | \hat{I} | \Psi \rangle \, ,$$

which completes the proof.

1.2. The Particle Current Operator.

Using the operator definition

$$\hat{I} = \frac{S_D}{2m} (|x_s\rangle \langle x_s | \hat{p}_x + \hat{p}_x | x_s \rangle \langle x_s |), \tag{4}$$

show that

$$\langle \Psi | \hat{I} | \Psi \rangle = J_x(x, t), \tag{5}$$

where the current density is given by

$$J_x(x, t) = S_D \frac{\hbar}{m} Im \left(\Psi^*(x, t) \frac{\partial \Psi(x, t)}{\partial x} \right). \tag{6}$$

Let us evaluate the expectation value $\langle \Psi | \hat{I} | \Psi \rangle$ by using the operator definition (4) in the position basis:

$$\langle \hat{I} \rangle = \frac{S_D}{2m} \left(\langle \Psi | x_s \rangle \langle x_s | \hat{p}_x | \Psi \rangle + \langle \Psi | \hat{p}_x | x_s \rangle \langle x_s | \Psi \rangle \right).$$

We recall that in the position representation, the momentum operator is $\hat{p}_x = -i\hbar \frac{\partial}{\partial x}$ and the projection operator $|x_s\rangle \langle x_s|$ localizes the interaction to the surface S_D at position x. Using $\langle x | \Psi \rangle = \Psi(x)$ and $\langle \Psi | \hat{p}_x^\dagger = (-i\hbar \frac{\partial \Psi}{\partial x})^* = i\hbar \frac{\partial \Psi^*}{\partial x}$, we arrive to:

$$\langle \hat{I} \rangle = \frac{S_D}{2m} \left(\Psi^*(x) \left[-i\hbar \frac{\partial \Psi(x)}{\partial x} \right] + \left[i\hbar \frac{\partial \Psi^*(x)}{\partial x} \right] \Psi(x) \right)$$
$$= \frac{S_D \hbar}{2mi} \left(\Psi^*(x) \frac{\partial \Psi}{\partial x} - \Psi(x) \frac{\partial \Psi^*}{\partial x} \right).$$

The term in the parentheses is exactly $2i \cdot Im(\Psi^* \nabla \Psi)$, so:

$$\langle \hat{I} \rangle = S_D \frac{\hbar}{m} Im \left(\Psi^* \frac{\partial \Psi}{\partial x} \right),$$

which perfectly matches the standard definition of the probability current density $J_x(x, t)$ integrated over the cross-sectional area S_D.

1.3. Landauer Conductance Simulation.

In this section, we simulate the electronic transport in a one-mode quantum wire. In order to do so, we adapt the numerical framework we used for the Resonant Tunneling Diode (RTD) simulation.

This simulation also makes use of the Numerov method to solve the time-independent Schrödinger equation and find the transmission probability. But now, the current calculation is simplified to follow the 1D Landauer-Büttiker formalism:

$$\langle I \rangle = \frac{e}{\pi \hbar} \int_0^\infty dE \, T(E)[f_L(E) - f_R(E)], \tag{7}$$

where the prefactor $e/\pi\hbar$ replaces the 3D density-of-states constants from the RTD simulation. The chemical potentials are defined as $\mu_L = E_F$ and $\mu_R = E_F - eV$, where V is the applied bias.

Once the mean current $\langle I \rangle$ is determined, we focus on the study of the conductance G, defined as:

$$G(V) = \frac{\langle I \rangle}{V}. \tag{8}$$

We evaluate G across several physical configurations to observe how temperature and potential barriers affect the system from ideal ballistic transport. These results from the simulation are benchmarked against the theoretical quantum conductance value $G_0 = e^2/\pi\hbar \approx 7.75 \times 10^{-5}\,\Omega^{-1}$, and they are summarized in Figure 1.
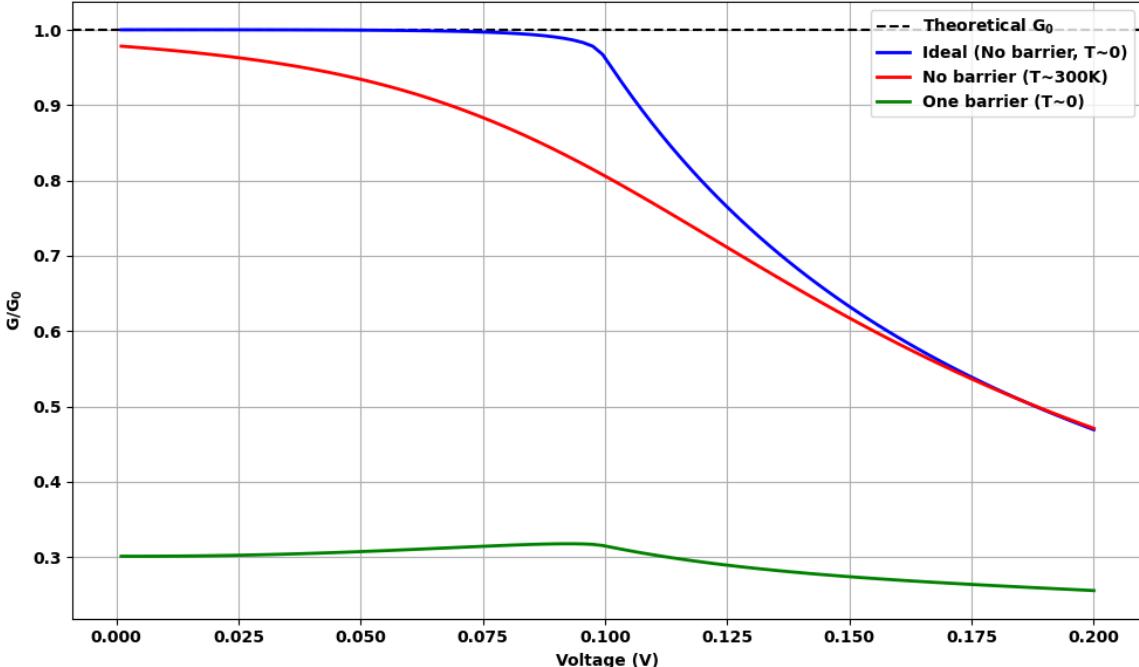

Figure 1: Numerical simulation of normalized conductance G/G_0 vs. applied voltage for different physical scenarios. Parameters used: Fermi level $E_F = 0.1$ eV, energy grid up to 0.5 eV with 1000 steps. The ideal case uses $T = 5$ K, while the room temperature case uses $T = 300$ K. The potential barrier configuration consists of a 0.3 eV barrier with a width of 4 nm.

- **The Ideal Case (No Barrier, $T \to 0$ K):**

  By setting the potential profile to zero ($V(x) = 0$) and taking a near-zero temperature ($T = 5$ K), we expect the quantum wire to behave as a waveguide in which we have perfect transmission.

  For voltages below the Fermi level ($V < 0.1$ V), we observe the conductance matches exactly the theoretical value G_0, implying a strictly linear relationship between the voltage and the current. This behavior can be explained by simply looking at the Landauer formalism for current (Eq. (7)): at near-zero temperatures, the Fermi-Dirac distributions $f_L(E)$ and $f_R(E)$ become sharp step functions, so that the difference $(f_L - f_R) = 1$ only within the specific energy window $[E_F - eV, E_F]$, and zero elsewhere. In this ballistic regime (where $T(E) = 1$), the integral then simplifies to $G_0 \cdot V$.

  However, as the applied bias V exceeds the Fermi level ($V > 0.1$ V), this linear relationship is lost and the system enters in a saturation regime. This is directly due to the Pauli Exclusion Principle, which dictates that each quantum state can only be occupied by a single electron.

In the ballistic model, the source reservoir can only inject electrons into available states up to its own Fermi level, E_F. At low voltages, increasing V lowers the chemical potential of the drain (μ_R), effectively "uncovering" more occupied states from the source that can propagate to the right. But once μ_R drops below the bottom of the conduction band ($E < 0$), the energy window available for transport reaches its physical limit. Now, even if the external voltage keeps widening the potential difference between the leads, the source simply has no more occupied states at lower energies to provide. Because of this, the number of electrons crossing the device cannot increase anymore, and the total current $\langle I \rangle$ reaches its maximum value. Since the current stays constant while the voltage keeps rising, the ratio $G = I/V$ starts to decrease like an inverse function, which is exactly what we see in the plot (blue line).

- **Effect of Room Temperature ($T = 300$ K):**

  When we increase the temperature to 300 K, we notice that the conductance (red line) behaves differently to the ideal case. Instead of staying perfectly flat at G_0 and then dropping suddenly at the Fermi level, the conductance starts to decrease gradually from the very beginning of the voltage sweep.

  This is because the thermal energy of the electrons ($k_B T \approx 26$ meV) is now significant. Looking at the Landauer formula again, the Fermi-Dirac distributions f_L and f_R are no longer sharp step functions. At $T = 5$ K, electrons are strictly packed below the Fermi level, but at room temperature, they are "smeared" out. This means some electrons gain enough thermal energy to jump into states above E_F, while leaving some states below E_F empty.

  Because of this gradual slope in the Fermi functions, the occupancy difference ($f_L - f_R$) is no longer a perfect rectangle of height 1. Even at low voltages, the "edges" of the energy window are blurred, which makes the transport slightly less efficient than the ideal ballistic case. As we increase the voltage, this thermal smearing causes the current to approach saturation much more smoothly rather than having a sharp corner at 0.1 V.

  Interestingly, for very large bias ($V \gg E_F$), the room temperature curve and the ideal case eventually match. This happens because the high voltage bias pushes the chemical potential of the drain μ_R far below the bottom of the conduction band, effectively "uncovering" every electron available in the source for transport. At this point, the energy window is so much larger than the thermal energy ($eV \gg k_B T$) that the fuzzy "smearing" at the edges of the Fermi levels no longer matters. Since the drain potential is so low that all these electrons can flow into empty states, the device carries the maximum possible current for a single mode, effectively making temperature effects negligible in the high-bias limit.

- **Effect of Potential Barriers:**

  In this case, we introduce a potential barrier of height $V_0 = 0.3$ eV (green line). As shown in the plots, the conductance drops significantly below the ideal G_0 value ($G < G_0$). This happens because the electrons are being partially reflected by the barrier, making the transmission probability no longer perfect ($T(E) < 1$).

  Since the barrier height of 0.3 eV is much higher than our Fermi level ($E_F = 0.1$ eV), the electrons do not have enough energy to pass over it classically. Therefore, the conductance we observe in the simulation is entirely due to quantum tunneling. This explains why the total current is so much smaller compared to the ballistic case, as only a small fraction of the incident wave function actually manages to tunnel through to the other side.

  Looking at the low voltage range, we see that the conductance remains almost constant except for a slight increase before it eventually drops near the Fermi level. The reason behind this is

because the applied voltage V creates a potential drop across the device that "tilts" the entire potential profile, making the barrier look more like a trapezoid than a rectangle. Physically, this reduces the "effective thickness" that the electrons have to tunnel through at certain energies, which slightly improves the transmission probability $T(E)$ as the bias increases.

However, just like in the ideal case, once the voltage exceeds the Fermi level ($V > 0.1$ V), the system enters the saturation regime. Here, even if the tilting continues to make tunneling easier, the source simply has no more occupied states at lower energies to provide. The current saturates to a maximum value, and thus the conductance drops as the V continues to increase.

Another interesting point in this 1D model is that the conductance G_0 does not depend on how wide the wire is. The cross-sectional area does not appear in any of the expressions for the current. Its only job is to provide the "confinement" needed to keep the electrons in a single mode, acting as a box that forces the electrons to behave like a 1D wave.

Mathematically, we can treat the wire as a 3D infinite potential well with dimensions $L \times W_y \times W_z$. The total energy of an electron is the sum of the kinetic energy along the wire (x-direction) and the quantized energy levels from the transverse confinement (y and z directions):

$$E(k_x, n_y, n_z) = \frac{\hbar^2 k_x^2}{2m^*} + E_{n_y, n_z} \tag{9}$$

where the transverse energy levels E_{n_y, n_z} are given by the well known particle-in-a-box solution:

$$E_{n_y, n_z} = \frac{\pi^2 \hbar^2}{2m^*} \left(\frac{n_y^2}{W_y^2} + \frac{n_z^2}{W_z^2} \right), \quad n_y, n_z = 1, 2, 3, \dots \tag{10}$$

As long as the lateral dimensions W_y and W_z are small enough, the energy required to reach the second transverse mode ($n = 2$) is much higher than the Fermi energy ($E_{2,1} > E_F$). In this case, only the first sub-band ($n_y = 1, n_z = 1$) is occupied, and the system behaves as a strictly 1D channel.

However, if we were to significantly increase the lateral area (making the wire wider), this confinement would eventually weaken. As W increases, the energy levels E_{n_y, n_z} drop due to the $1/W^2$ relationship, making higher-order sub-bands eventually fall below our Fermi level E_F, and opening new "channels" or modes for transport. Each of these new channels adds exactly G_0 to the total conductance, so that if we were to plot the conductance against the width of the wire, we would see a "staircase" behavior ($G \approx N G_0$).

Exercise 2: Quantum Noise

2.1. Analytical Partition Noise.

Show that for $f_R = 0$ and $f_L = 1$, the zero-frequency noise reduces to

$$S(0) = 2e\langle I \rangle (1 - T). \tag{11}$$

We start with the general expression for zero-frequency spectral density (noise) in a ballistic, one mode quantum wire:

$$S(0) = \frac{2e^2}{\pi \hbar} \int dE \left\{ T[f_L(1 - f_R) + f_R(1 - f_L)] - T^2 (f_L - f_R)^2 \right\}. \tag{12}$$

Under the condition of a large bias at zero temperature, we assume the left reservoir is full ($f_L = 1$) and the right reservoir is empty ($f_R = 0$). Substituting these values, the expression simplifies to:

$$S(0) = \frac{2e^2}{\pi \hbar} \int dE \left(T \cdot 1 - T^2 \cdot 1 \right) = \frac{2e^2}{\pi \hbar} \int dE \, T(1 - T).$$

On the other hand, we know that the mean current in this limit is

$$\langle I \rangle = \frac{e}{\pi \hbar} \int dE \, T(f_L - f_R) = \frac{e}{\pi \hbar} \int dE \, T.$$

If we assume transmission T is roughly constant over the integration window, we can rewrite $S(0)$ in terms of $\langle I \rangle$:

$$S(0) = 2e \left(\frac{e}{\pi \hbar} \int dE \, T \right) (1 - T) = 2e \langle I \rangle (1 - T). \tag{13}$$

This result shows that the Fano factor $F = S/2e\langle I \rangle$ equals $(1 - T)$. For low transmission ($T \to 0$), $F \to 1$ (Poissonian shot noise), while for high transmission ($T \to 1$), $F \to 0$ (noise suppression due to the Pauli exclusion principle).

2.2. Fano Factor Simulation.

In this section, we focus on simulating current fluctuations, known as quantum noise. We analyze a quantum wire containing a double-barrier potential ($V_0 = 0.4$ eV), which creates a resonant state within the quantum well.

To characterize the noise, we compute the zero-frequency noise spectral density $S(0)$ by implementing the general expression for a ballistic one-mode quantum wire (Eq. (12)). This expression accounts for both thermal noise and partition noise, named in the literature as shot noise. Once $S(0)$ and $\langle I \rangle$ are computed, we determine the Fano factor F, which measures the deviation of the noise from the classical Poissonian limit:

$$F(V) = \frac{S(0)}{2e\langle I \rangle}. \tag{14}$$

The simulation results for the double-barrier system ($V_0 = 0.4$ eV) are summarized in Figure 2. We compare the ideal low-temperature limit ($T = 5$ K, blue line) with the room-temperature case ($T = 300$ K, red line) to observe how thermal excitations affect the noise in the solid state device.

- **I-V Plot:**

  As shown in the I-V (top) plot and already discussed in detail in the previous assignment, the current follows the standard resonant tunneling behavior: it initially increases with voltage until it peaks due to the alignment of the discrete energy level of the quantum well with the source's energy window. For $T = 5$ K, this peak is slightly sharper than for 300 K, where the peak is a bit broader due to the thermal smearing of the Fermi distributions. Once the bias exceeds a certain threshold, the resonant state is pushed below the bottom of the conduction band of the source. This leads to the observed Negative Differential Resistance (NDR) region, where the current drops even as the voltage continues to increase (the electrons can no longer tunnel through the resonant state).

- **Noise Spectral Density $S(V)$ Plot:**

  The noise spectral density (middle plot) follows a similar tendency as the current but reveals the temperature dependence of the fluctuations. At $T = 5$ K, the noise is dominated by partition noise, which scales with the current and vanishes at zero bias. In contrast, at $T = 300$ K (room temperature), we observe a significant source of noise at low voltages corresponding to thermal noise. It arises from the random thermal motion of the carriers and, as we can notice, is proportional to the temperature. However, as the voltage increases and the device enters the resonant regime, the partition noise begins to dominate, and the two temperature curves converge near the resonant peak current.

- **Fano Factor Plot:**

  The Fano factor $F(V)$ acts as a normalized measure of noise, allowing us to quantify how correlated the electrons are in the electronic transport compared to a classical random process. By
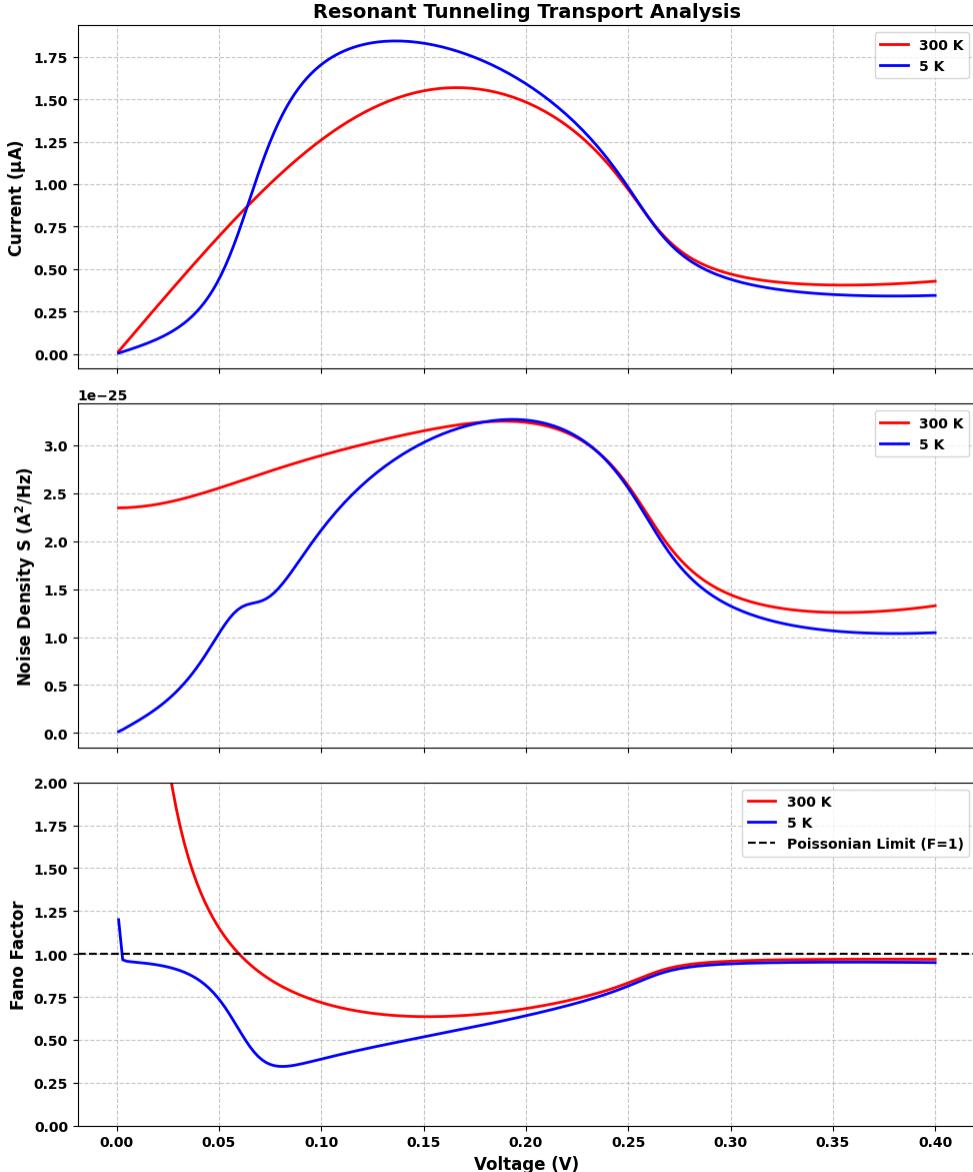
Figure 2: Simulated transport and noise characteristics for a double-barrier resonant tunneling structure. The device consists of two potential barriers ($V_0 = 0.4$ eV, width 2 nm) separated by a 5 nm quantum well. The subplots show: (Top) Current-Voltage (I-V) characteristic showing the resonant peak; (Middle) Zero-frequency noise spectral density $S(0)$; (Bottom) Fano factor $F = S/2e\langle I \rangle$. The dashed line at $F = 1$ represents the Poissonian limit. Simulations compare low temperature ($T = 5$ K, blue) and room temperature ($T = 300$ K, red) regimes.

defining $F = S/2e\langle I \rangle$, we can experimentally verify the analytical prediction derived in Eq. (12). This relationship suggests that for a single-mode conductor, the noise suppression is directly linked to the transmission probability by $F \approx (1 - T)$, provided the thermal energy is negligible.

Using this theoretical framework, we can categorize the three distinct transport regimes observed in the bottom plot of Figure 2:

– **Super-Poissonian Regime ($F \gg 1$):** Looking at the room temperature results (300 K) in the low-bias limit ($V \to 0$), we observe a sharp divergence where the Fano factor takes a value much larger than unity. In this regime, we saw the noise was primarily due to thermal fluctuations. These keep $S(0)$ finite while the mean current $\langle I \rangle$ vanishes, making thus the ratio diverge mathematically. Physically, this corresponds to positive correlations or the so-called "bunching" of the carriers due to thermal agitation.

– **Sub-Poissonian Regime ($F < 1$):** The most interesting quantum feature manifests as the voltage increases and the system reaches resonance. Here, the transmission probability approaches unity ($T \rightarrow 1$). As predicted by our relation $F \approx (1 - T)$, the noise is strongly suppressed, revealing a quantum anti-bunching effect. This behavior is a direct consequence of the Pauli Exclusion Principle: since two electrons cannot occupy the same quantum state simultaneously, a high transmission probability ($T \approx 1$) effectively forces the carriers to space themselves out in time. This prevents them from traversing the device in random clusters, resulting in a highly regulated, low-noise stream.

– **Poissonian Regime ($F \approx 1$):** In the off-resonance regions (either at very low bias for 5 K or past the NDR region), the transmission coefficient drops significantly ($T \ll 1$). In this limit, the factor $(1 - T)$ approaches unity, restoring the Fano factor to the Poissonian limit ($F \approx 1$). Here, the Pauli principle influence is negligible because the channel is mostly empty. Consequently, the electrons tunnel through the barriers as rare, independent, and uncorrelated random events. This stochastic behavior mirrors classical "shot noise", where the arrival of one carrier has no influence on the arrival of the next.

2.3. Equilibrium Noise Simulation.

The Fluctuation-Dissipation (FD) Theorem states that the fluctuations of a system at thermodynamic equilibrium (i.e., when $eV \ll k_BT$) are directly related to its linear response to an external perturbation. In the context of quantum transport, the "fluctuation" is the noise spectral density $S(\omega = 0, V = 0)$ and the "dissipation" (perturbation) is the electrical conductance $G(V \rightarrow 0)$. For a 1D conductor, the relationship is defined as:

$$S(\omega = 0, V = 0) = 4k_BTG(V \rightarrow 0). \tag{15}$$

In the last section, we could already see in Figure 2 that, even though the net current was zero at zero bias (near equilibrium), the noise remained finite due to purely the thermal motion of electrons in the reservoirs, creating spontaneous current fluctuations even in the absence of a driving field. This can easily be checked if we look at the expression for the noise spectral density (Eq. 12): at equilibrium ($V = 0$), the chemical potentials align ($\mu_L = \mu_R$), implying $f_L(E) = f_R(E) = f(E)$. Consequently, the second term in the integrand (which represents the partition noise) vanishes because there is no net flow of carriers to be partitioned. The remaining first term represents purely thermal noise (fluctuations) due to the motion of electrons in the leads.

We now want to prove graphically the result from the FD Theorem (Eq. (15)) by plotting the simulated noise at zero bias and zero frequency, $S(\omega = 0, V = 0)$, against the theoretical prediction $4k_BTG(V \rightarrow 0)$ for different case scenarios.

We first consider a quantum wire ($T(E) = 1$) at low temperature ($T = 5$ K). As shown in Figure 3, the agreement in the limit $V \rightarrow 0$ is exact, confirming the theoretical prediction by the FD Theorem. We can also notice that, as the bias voltage increases, the curves begin to diverge because the system is driven out of equilibrium, and the $(f_L - f_R)^2$ term in Eq. (12) becomes non-zero, adding partition noise to the background thermal noise.

Finally, to demonstrate the universality of the theorem, we extended the simulation to two distinct regimes: a high-temperature wire ($T = 300$ K) and a resonant tunneling structure (Double Barrier at 5 K). The results, compared in Figure 4, confirm that the simulation perfectly matches the theoretical prediction $S = 4k_BTG$ at low bias despite massive scaling differences. In the high-temperature case (top panel), despite the noise spectral density increasing by nearly two orders of magnitude due to the broadening of the thermal window k_BT, it still approaches the theoretical result near equilibrium. On the other hand, the double barrier (bottom panel) provides a critical test where transmission $T(E)$ and conductance G are drastically reduced due to Resonant Tunneling effects. The simulation, however, correctly captures how the noise drops to match the reduced conductance, proving that the FD Theorem holds independent of the scattering geometry or the temperature.
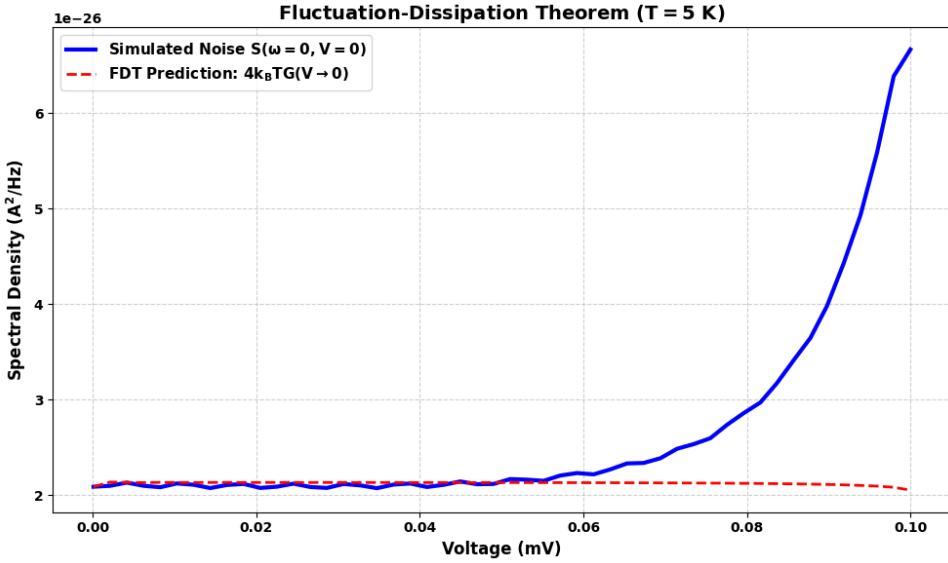
Figure 3: Verification of the FD Theorem at $T = 5$ K for a ballistic wire. The simulated noise (blue) converges exactly to the thermal prediction (red) in the limit $V \to 0$. At finite bias ($V > 1$ mV), the curves start to diverge as non-equilibrium partition (shot) noise begins to contribute.
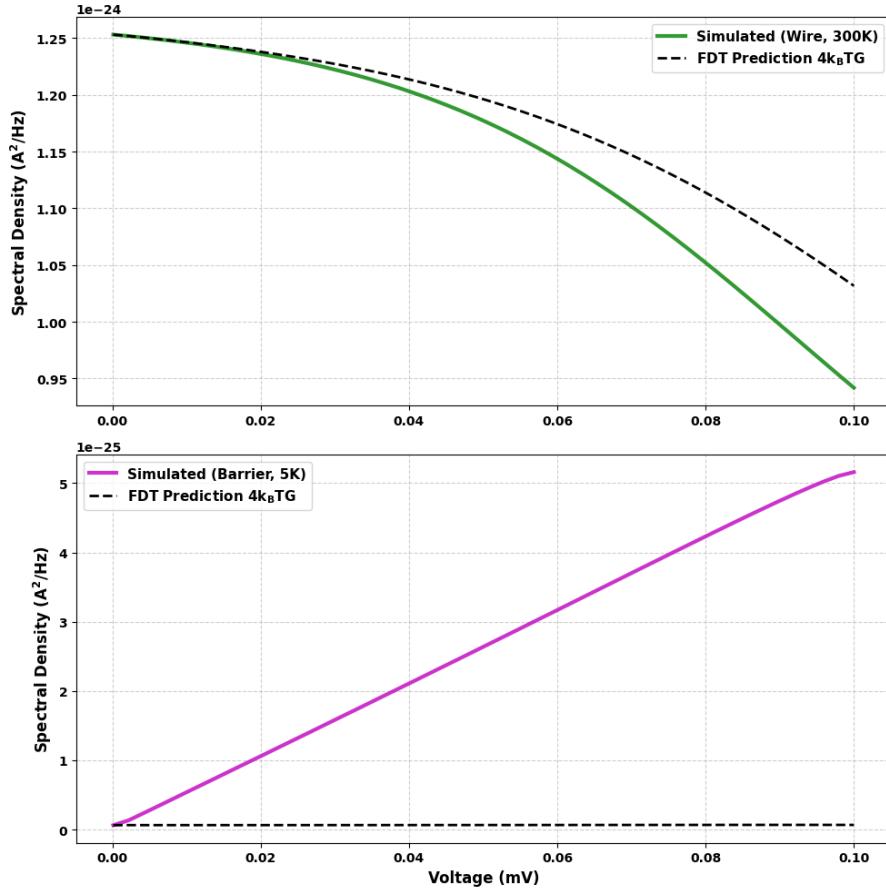
Figure 4: FD Theorem test across different physical regimes. Top Panel: Simulation at 300 K, with no barrier (ballistic wire). Bottom Panel: Simulation at low temperatures ($T = 5$ K), with a double-barrier structure (barriers of 2 nm wide, separated by a 4 nm well).

Exercise 3: Quantum high frequency

Question 3.1. Total Current at High Frequency.

Show that the particle current is insufficient at high frequencies and that the total current (particle + displacement) is required for charge conservation.

In a time-dependent scenario, the particle charge density ρ is not constant ($\partial_t \rho \neq 0$). The continuity equation states:

$$\nabla \cdot \vec{J_c} + \frac{\partial \rho}{\partial t} = 0. \tag{16}$$

This implies that the particle current divergence is non-zero ($\nabla \cdot \vec{J_c} \neq 0$), meaning the particle current entering a region does not necessarily equal the current leaving it, which leads to an accumulation of charge.

Using Gauss's Law, $\nabla \cdot \vec{E} = \rho/\epsilon$, we substitute ρ into the continuity equation:

$$\nabla \cdot \vec{J_c} + \frac{\partial}{\partial t}(\epsilon \nabla \cdot \vec{E}) = 0 \quad \Rightarrow \quad \nabla \cdot \left(\vec{J_c} + \epsilon \frac{\partial \vec{E}}{\partial t} \right) = 0.$$

We can now define the displacement current density as $\vec{J_d} = \epsilon \frac{\partial \vec{E}}{\partial t}$, and the term in the parentheses is the so-called Total Current Density:

$$\vec{J}_{total} = \vec{J_c} + \vec{J_d}. \tag{17}$$

Since $\nabla \cdot \vec{J}_{total} = 0$, the total current is divergence-free, which guarantees that the instantaneous current measured at the ammeter (outside the device) is exactly equal to the total current flowing through the active region.

3.2. Total Current Measurement as a Weak Measurement.

Consider the experimental setup in Fig. 5, where the total current measurement is done outside the active region. This measurement process can be modeled as a POVM described by a set of Kraus operators $\hat{M}_{I'_j}$ associated with a measurement outcome I'_j:

$$\hat{M}_{I'_j} = \frac{1}{\sqrt{C}} \sum_i \exp\left(-\frac{(I'_j - I_i)^2}{2\sigma^2} \right) |I_i\rangle \langle I_i|, \tag{18}$$

where $|I_i\rangle$ are the current eigenstates, σ represents the detector noise (uncertainty), and C is a normalization constant.
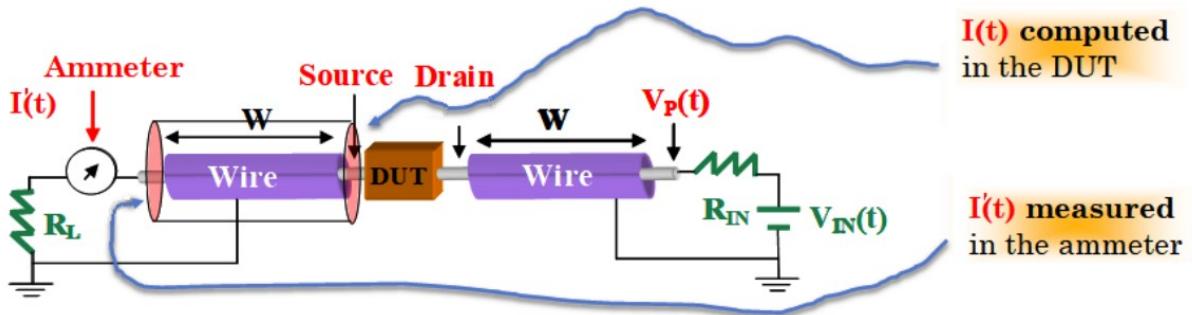
Figure 5: The prediction of the current is done by simulating ONLY the active device region, which is an open system. In the terminology of experimental laboratory, the active region is also named as the device under test (DUT). However, the real measurement of the current is done in the ammeter which is a measuring apparatus far from the active region. The relevant question is what we have to predict in the active region to be sure that it is equal to what is measured in the ammeter.

Show that the state of the quantum system in the active region, after the measurement of the total current I'_j in the ammeter, is given by:

$$\hat{M}_{I'_j} |\Psi\rangle = \sum_{i=-\infty}^{\infty} \frac{1}{\sqrt{C}} e^{-\frac{(I'_j - I_i)^2}{2\sigma^2}} c_i |I_i\rangle \tag{19}$$

The state after measurement is obtained by applying the Kraus operator $\hat{M}_{I'_j}$ to the initial wave function, which we expand in the basis of current eigenstates $|\Psi\rangle = \sum_i c_i |I_i\rangle$:

$$\hat{M}{I'j}|\Psi\rangle = \left(\sum_i \frac{1}{\sqrt{C}} e^{-\frac{(I'j - I_i)^2}{2\sigma^2}} |I_i\rangle\langle I_i|\right)\left(\sum_k c_k |I_k\rangle\right)$$

$$= \sum_i \sum_k \frac{c_k}{\sqrt{C}} e^{-\frac{(I'j - I_i)^2}{2\sigma^2}} |I_i\rangle \underbrace{\langle I_i | I_k\rangle}_{\delta_{ik}}$$

$$= \sum_i \frac{c_i}{\sqrt{C}} e^{-\frac{(I'j - I_i)^2}{2\sigma^2}} |I_i\rangle,$$

arriving to the desired result.

Discuss which values of σ, in the expression $e^{-\frac{(I'_j - I_i)^2}{2\sigma^2}}$, imply a minimal perturbation of the initial state, and which value of σ implies a maximum perturbation.

The parameter σ represents the detector uncertainty or noise. In the limit where $\sigma \to \infty$ (high detector noise), the exponential term approaches one, and the measurement operator becomes proportional to the identity operator ($\hat{M}_{I'_j} \approx \frac{1}{\sqrt{C}}\mathbb{I}$), implying a minimal perturbation of the initial state. On the other hand, when $\sigma \to 0$, the Gaussian becomes extremely sharp and forces a projection onto a specific eigenstate $|I_k\rangle$. This represents a maximum perturbation (strong measurement), resulting in the collapse of the wave function.

Show that the probability $P(I'_j)$ of getting the value $P(I'_j)$ is given by (Born law) as:

$$P(I'_j) = \langle\Psi|\hat{M}^\dagger_{I'_j}\hat{M}_{I'_j}|\Psi\rangle = \sum_{i=-\infty}^{\infty} \frac{1}{C} e^{-\frac{(I'_j - I_i)^2}{\sigma^2}} |c_i|^2 \tag{20}$$

Using the unnormalized state $\tilde{\Psi} = \hat{M}_{I'_j}|\Psi\rangle$, we calculate the probability as the norm of this state:

$$P(I'_j) = \langle\tilde{\Psi}|\tilde{\Psi}\rangle$$

$$= \left(\sum_k \frac{c_k^*}{\sqrt{C}} e^{-\frac{(I'_j - I_k)^2}{2\sigma^2}} \langle I_k|\right)\left(\sum_i \frac{c_i}{\sqrt{C}} e^{-\frac{(I'_j - I_i)^2}{2\sigma^2}} |I_i\rangle\right)$$

$$= \sum_k \sum_i \frac{c_k^* c_i}{C} e^{-\frac{(I'_j - I_k)^2}{2\sigma^2}} e^{-\frac{(I'_j - I_i)^2}{2\sigma^2}} \underbrace{\langle I_k | I_i\rangle}_{\delta_{ki}}$$

$$= \sum_i \frac{|c_i|^2}{C} \underbrace{e^{-\frac{(I'_j - I_i)^2}{2\sigma^2}} e^{-\frac{(I'_j - I_i)^2}{2\sigma^2}}}_{e^{-\frac{(I'_j - I_i)^2}{\sigma^2}}}$$

$$= \sum_i \frac{|c_i|^2}{C} e^{-\frac{(I'_j - I_i)^2}{\sigma^2}}$$

Show that $\langle\hat{I'}\rangle = \sum_{j=-\infty}^{\infty} I'_j P(I'_j) = \sum_{i=-\infty}^{\infty} I_i P(I_i) = \langle\hat{I}\rangle$.

The expectation value of the measured outcomes is given by:

$$\langle\hat{I'}\rangle = \sum_j I'_j \sum_i \frac{1}{C} e^{-\frac{(I'_j - I_i)^2}{\sigma^2}} |c_i|^2 = \sum_i |c_i|^2 \left[\frac{1}{C} \sum_j I'_j e^{-\frac{(I'_j - I_i)^2}{\sigma^2}}\right]$$

Let us define the change of variables $\alpha_j = I'_j - I_i$ (so that $I'_j = I_i + \alpha_j$). Then, we can rewrite the term in the brackets as:

$$\langle \hat{I}' \rangle = \sum_i |c_i|^2 \left[I_i \left(\frac{1}{C} \sum_j e^{-\frac{\alpha_j^2}{\sigma^2}} \right) + \left(\frac{1}{C} \sum_j \alpha_j e^{-\frac{\alpha_j^2}{\sigma^2}} \right) \right]$$

The first term inside the brackets reduces to I_i because the term in the parentheses is the normalization constant defined as 1, whereas the second term vanishes because $\alpha_j e^{-\alpha_j^2/\sigma^2}$ is an odd function summed over a symmetric range. This proves that:

$$\langle \hat{I}' \rangle = \sum_i I_i |c_i|^2 = \langle \hat{I} \rangle \tag{21}$$

Explain some arguments on why the measurement of the electrical current in the active region of Fig. 5 cannot be considered as a strong measurement, but as a weak measurement done by the ammeter. Discuss also what is the perturbation on the wave function of the active region.

The measurement cannot be considered strong because the ammeter is located far from the active region and only interacts with electrons that have already traversed the device and the connecting wires. Therefore, we can assume that, in this high-frequency regime, the detector noise σ is significantly larger than the discrete spacing between current eigenstates. This lack of resolution prevents the ammeter from observing a specific eigenvalue, resulting in the extraction of blurred, ensemble-averaged information rather than forcing a wavefunction collapse. Consequently, the Kraus operator acts almost as an identity transformation, leaving the coherent superposition within the tunneling region intact and ensuring that the perturbation to the system's state remains negligible.

Conclusions

In this work, we have moved beyond the standard mean-field description of quantum transport to analyze the intrinsic fluctuations that characterize nanoscale devices. By implementing a rigorous Landauer-Büttiker solver, we established three fundamental physical insights:

First, we confirmed that electrical conductance is not merely a geometric property but a function of the available energy states, limited by the Pauli Exclusion Principle at high bias. Second, our analysis of the Fano factor revealed the quantum nature of the charge carriers: we observed sub-Poissonian noise ($F < 1$) in the resonant regime, a direct signature of Fermi-Dirac statistics suppressing charge bunching. Conversely, we showed that thermal agitation in the leads restores random, uncorrelated transport ($F > 1$) at room temperature and high bias.

Finally, we validated the reliability of our simulation by proving the Fluctuation-Dissipation Theorem across three distinct regimes. The perfect agreement between the equilibrium noise $S(0)$ and the linear conductance $4k_B T G$ serves as a robust "stress test", confirming that a device which dissipates current (resistivity) must necessarily generate fluctuations (noise). Combined with the derivation of the total current (particle + displacement), this work provides a complete framework for modeling dynamic, high-frequency quantum transport while strictly preserving charge conservation.

A   Python Simulation Code

Below is the complete Python code used to generate all the simulations presented in this report.

```python
import numpy as np
import matplotlib.pyplot as plt

#
    --------------------------------------------------------------------------
# PHYSICAL CONSTANTS & PARAMETERS
#
    --------------------------------------------------------------------------

Dx = 1e-10               # Spatial step (1 Angstrom)
L_device = 2e-8          # Simulation Box length
m = 0.063 * 9.109e-31    # Effective electron mass (kg)
h_bar = 1.054e-34        # Reduced Planck constant ( J  s )
qe = 1.6e-19             # Electron charge (C)
Kb = 1.38e-23            # Boltzmann constant (J/K)
T = 298                  # Temperature (K)
Ef = 0.1                 # Fermi energy (eV)

# Theoretical Landauer conductance G0
G0 = qe**2 / (np.pi * h_bar)

#
    --------------------------------------------------------------------------
# HELPER FUNCTIONS
#
    --------------------------------------------------------------------------


def get_transmission(Energy, Potential):
    """
    Calculates Transmission T(E) using the Numerov Method for a given
        Potential profile.
    From HW3: RTD simulation

    Inputs:
    :Energy: Energy values (in eV)
    :Potential: Potential Energy V(x) (in Joules)
    """
    N_points = len(Potential)
    x = np.arange(1, N_points + 1) * Dx

    T_prob = np.zeros(len(Energy))

    # Numerov Constants (for Equation 1)
    A = (10 * Dx**2) / 12
    AA = Dx**2 / 12
    const_factor = (2 * m) / h_bar**2

    for i, E in enumerate(Energy):
        E_joule = E * qe

        # We ensure E > U at boundaries (leads) to avoid k being imaginary
        if E_joule <= Potential[0] or E_joule <= Potential[-1]:
            T_prob[i] = 0.0
            continue
```

```python
49
50              # Wavevectors in extremes (scattering states): k = sqrt(2m(E-V)) /
                    h_bar
51              k_left = np.sqrt(const_factor * (E_joule - Potential[0]))
52              k_right = np.sqrt(const_factor * (E_joule - Potential[-1]))
53
54              # Numerov Function: f(x) = 2m/h^2 * (V(x) - E)
55              # Equation to be solved: psi'' = f(x)psi.
56              func_numerov = const_factor * (Potential - E_joule)
57
58              # Initialize Wavefunction (first two points at right boundary)
59              # We assume outgoing plane waves on the right are prop. to exp(i *
                    k_right * x) (scattering state)
60              # This is the same as assuming a wave has successfully transmited to
                    the right side
61              psi = np.zeros(N_points, dtype=complex)
62              psi[-1] = np.exp(1j * k_right * x[-1])
63              psi[-2] = np.exp(1j * k_right * x[-2])
64
65              # Backward Integration from Right to Left (Equation 1)
66              for j in range(N_points - 3, -1, -1):
67                  term1 = (2 + A * func_numerov[j+1]) * psi[j+1]
68                  term2 = (1 - AA * func_numerov[j+2]) * psi[j+2]
69                  denom = (1 - AA * func_numerov[j])
70                  psi[j] = (term1 - term2) / denom
71
72              # Now let's match boundary conditions at the left side:
73              # Psi_left = A * exp(ikx) + B * exp(-ikx)
74              # We want T = (k_right / k_left) * |Amplitude_Transmitted|^2 / |
                    Amplitude_Incident|^2
75              #           = (k_right / k_left) * (1 / |A|^2)
76              # We need to isolate A. We do it using Psi_left and Psi'_Left = ik *
                    A * exp(ikx) -ik * B * exp(-ikx)
77              # Knowing Psi and Psi', we have two unkowns A and B --> A = ... =
                    (1/2) exp(-ikx) (Psi(x) + Psi'(x) / (ik))
78
79              # First, we get numerical values for psi and psi' in a safe point (
                    not at the edges)
80              idx_match = 5
81              psi_val = psi[idx_match]
82              # Finite difference derivative: dpsi/dx
83              d_psi = (psi[idx_match+1] - psi[idx_match]) / Dx
84
85              # Second, we get the incident amplitude with the derived formula
                    above:
86              A_inc = 0.5 * np.exp(-1j * k_left * x[idx_match]) * (psi_val + d_psi
                    / (1j * k_left))
87
88              if np.abs(A_inc) == 0:
89                  T_prob[i] = 0
90              else:
91                  T_prob[i] = (k_right / k_left) / (np.abs(A_inc)**2)
92
93          return T_prob
94
95  def compute_current(Voltages, Energies, U_base, T):
96      """
97      Computes Mean Current (I) for a given potential profile U_base and
            temperature T.
98      """
99      Currents = np.zeros(len(Voltages))
```

```python
100
101         # Define active region indices
102         N_pts = len(U_base)
103         start_active = int(0.25 * N_pts)
104         end_active = int(0.75 * N_pts)
105         points_active = end_active - start_active
106         # Linear slope drop:
107         points_slope = np.arange(points_active)
108         fraction = (points_slope / points_active)
109
110         for i, Vb in enumerate(Voltages):
111             # 1. Tilt potential
112             U_bias = U_base.copy()
113
114             # Right reservoir drops by Vb
115             U_bias[end_active:] = U_bias[end_active:] - (Vb * qe)
116
117             # Voltage drop across active region
118             voltage_drop = (Vb * qe) * fraction
119             U_bias[start_active:end_active] -= voltage_drop
120
121             # 2. Compute T(E, V)
122             Trans = get_transmission(Energies, U_bias)
123
124             # 3. Compute Current Integral
125             mu_L = Ef
126             mu_R = Ef - Vb
127
128             # LOW TEMPERATURE (step function approx)
129             if T < 1e-6:
130                 # At T~0, Fermi-Dirac becomes a step function
131                 # s.t. f_L (f_R) = 1 for E < mu_L (mu_R) and 0 otherwise
132                 # Current is integral of T(E) between mu_R and mu_L
133                 # Find indices where E is between mu_R and mu_L
134                 idx_window = np.where((Energies >= mu_R) & (Energies <= mu_L))
                        [0]
135                 if len(idx_window) > 0:
136                     integral = np.trapezoid(Trans[idx_window], Energies[
                            idx_window] * qe)
137                 else:
138                     integral = 0
139
140             # FINITE TEMPERATURE
141             else:
142                 arg_L = (Energies - mu_L) * qe / (Kb * T)
143                 arg_R = (Energies - mu_R) * qe / (Kb * T)
144
145                 # Fermi Functions
146                 fL = 1.0 / (1.0 + np.exp(arg_L))
147                 fR = 1.0 / (1.0 + np.exp(arg_R))
148
149                 integrand = Trans * (fL - fR)
150                 integral = np.trapezoid(integrand, Energies * qe)
151
152             # I = (e / pi hbar) * Integral
153             Currents[i] = (qe / (np.pi * h_bar)) * integral
154
155         return Currents
156
157     def compute_noise(Voltages, Energies, U_base, T):
158         """
```

```python
159         Computes Noise Power Spectral Density S(0)
160         for a given potential profile U_base and temperature T.
161         """
162         Noise_S = np.zeros(len(Voltages))
163
164         # Define active region indices
165         N_pts = len(U_base)
166         start_active = int(0.25 * N_pts)
167         end_active = int(0.75 * N_pts)
168         points_active = end_active - start_active
169         points_slope = np.arange(points_active)
170         fraction = (points_slope / points_active)
171
172         for i, Vb in enumerate(Voltages):
173             # 1. Tilt potential
174             U_bias = U_base.copy()
175             # Right reservoir drops by Vb
176             U_bias[end_active:] = U_bias[end_active:] - (Vb * qe)
177             # Voltage drop across active region
178             voltage_drop = (Vb * qe) * fraction
179             U_bias[start_active:end_active] -= voltage_drop
180
181             # 2. Compute T(E, V)
182             Trans = get_transmission(Energies, U_bias)
183
184             # 3. Compute Noise Integral
185             # Integrand: T * [fL(1-fR) + fR(1-fL)] - T^2 * (fL - fR)^2
186             mu_L = Ef
187             mu_R = Ef - Vb
188
189             # LOW TEMPERATURE (step function approx)
190             if T < 1e-6:
191                 idx_window = np.where((Energies >= mu_R) & (Energies <= mu_L))
                        [0]
192                 if len(idx_window) > 0:
193                     # Noise Integrand: T * (1 - T)
194                     integrand =Trans[idx_window] * (1 - Trans[idx_window])
195                     integral = np.trapezoid(integrand, Energies[idx_window] * qe
                        )
196                 else:
197                     integral = 0
198             # FINITE TEMPERATURE
199             else:
200                 arg_L = (Energies - mu_L) * qe / (Kb * T)
201                 arg_R = (Energies - mu_R) * qe / (Kb * T)
202
203                 # Fermi Functions
204                 fL = 1.0 / (1.0 + np.exp(arg_L))
205                 fR = 1.0 / (1.0 + np.exp(arg_R))
206
207                 # Noise Integrand
208                 integrand = Trans * (fL * (1 - fR) + fR * (1 - fL)) - Trans**2 *
                        (fL - fR)**2
209                 integral = np.trapezoid(integrand, Energies * qe)
210
211             # S = (2e^2 / pi hbar) * Integral
212             Noise_S[i] = (2.0 * qe**2 / (np.pi * h_bar)) * integral
213
214         return Noise_S
215
216         #
```

16

```python
       # -------------------------------------------------------------------------
217    # PART 1: LANDAUER CONDUCTANCE
218    #
       #    -------------------------------------------------------------------------
219
220    # Set up (perfect wire, no barrier)
221    xpoints = int(np.floor(L_device / Dx))
222    U_wire = np.zeros(xpoints) # Flat potential V=0 everywhere
223
224    # Energy Grid
225    E_max = 0.5
226    E_steps = 1000
227    Energies = np.linspace(0.001, E_max, E_steps)
228
229    # Voltage Sweep
230    V_max = 0.2
231    V_steps = 100
232    Voltages = np.linspace(0.001, V_max, V_steps) # Avoid dividing by zero
233
234    # Case 1: Ideal Case (T -> 0 K, No Barrier)
235    print("Simulating Case 1: T -> 0 K, No Barrier...")
236    I_ideal = compute_current(Voltages, Energies, U_wire, T=5)
237    G_ideal = I_ideal / Voltages
238
239    # Case 2: One Barrier (T -> 0 K, 0.3 eV)
240    print("Simulating Case 2: T -> 0 K, One Barrier...")
241    # Define a small barrier
242    U_barrier = np.zeros(xpoints)
243    center = xpoints // 2
244    width = int(2e-9 / Dx)
245    U_barrier[center - width//2 : center + width//2] = 0.3 * qe
246
247    I_bar = compute_current(Voltages, Energies, U_barrier, T=5)
248    G_bar = I_bar / Voltages
249
250    # Case 3: Room Temperature (T = 300 K, No Barrier)
251    print("Simulating Case 3: T = 300 K, No Barrier...")
252    I_room = compute_current(Voltages, Energies, U_wire, T=300)
253    G_room = I_room / Voltages
254
255    # Case 4: Room Temperature + Barrier (T = 300 K, One Barrier)
256    #print("Simulating Case 4: T = 300 K, One Barrier...")
257    #I_roombar = compute_current(Voltages, Energies, U_barrier, T=300)
258    #G_roombar = I_roombar / Voltages
259
260
261    # Plot G/G0 vs Voltage
262    plt.figure(figsize=(10, 6))
263    plt.axhline(y=1.0, color='k', linestyle='--', label='Theoretical $G_0$')
264    plt.plot(Voltages, G_ideal / G0, 'b-', lw=2, label='Ideal (No barrier, T~0)'
           )
265    plt.plot(Voltages, G_room / G0, 'r-', lw=2, label='No barrier (T~300K)')
266    plt.plot(Voltages, G_bar / G0, 'g-', lw=2, label='One barrier (T~0)')
267    #plt.plot(Voltages, G_roombar / G0, 'm-', label='One Barrier (T~298K)')
268
269
270    plt.xlabel('Voltage (V)')
271    plt.ylabel(r'$G/G_0$')
272    plt.legend(loc='upper right')
```

```
273  plt.grid(True)
274  plt.tight_layout()
275  plt.show()
276
277  #
       -------------------------------------------------------------------------

278  # PART 2: FANO FACTOR AND EQUILIBRIUM NOISE
279  #
       -------------------------------------------------------------------------

280
281  # Define Double Barrier Potential (0.4 eV, 2nm wide, 5nm apart)
282  xpoints = int(np.floor(L_device / Dx))
283  U_double = np.zeros(xpoints)
284  b_width = int(2e-9 / Dx)
285  well_width = int(4e-9 / Dx)
286  center = xpoints // 2
287
288  # Two barriers
289  U_double[center - well_width//2 - b_width : center - well_width//2] = 0.4 *
         qe
290  U_double[center + well_width//2 : center + well_width//2 + b_width] = 0.4 *
         qe
291
292  # Run Simulations
293  V_sweep = np.linspace(0.001, 0.4, 200)
294  E_grid = np.linspace(0.001, 0.6, 500)
295
296  print("\nSimulating Double Barrier: T = 300 K")
297  I_300 = compute_current(V_sweep, E_grid, U_double, T=300)
298  S_300 = compute_noise(V_sweep, E_grid, U_double, T=300)
299
300  print("\nSimulating Double Barrier: T = 5 K")
301  I_5 = compute_current(V_sweep, E_grid, U_double, T=5)
302  S_5 = compute_noise(V_sweep, E_grid, U_double, T=5)
303
304  # Fano Factor F = S / (2eI)
305  F_300 = S_300 / (2 * qe * I_300)
306  F_5 = S_5 / (2 * qe * I_5)
307
308  # PLOTTING: IV CURVE, NOISE, AND FANO FACTOR
309
310  fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10, 12), sharex=True)
311
312  # Plot 1: I-V Characteristic
313  ax1.plot(V_sweep, I_300 * 1e6, 'r-', linewidth=2, label='300 K')
314  ax1.plot(V_sweep, I_5 * 1e6, 'b-', linewidth=2, label='5 K')
315  ax1.set_ylabel(r'Current ($\mu$A)', fontsize=12)
316  ax1.set_title('Resonant Tunneling Transport Analysis', fontsize=14)
317  ax1.legend(loc='upper right')
318  ax1.grid(True, linestyle='--', alpha=0.7)
319
320  # Plot 2: Noise Power S(V)
321  ax2.plot(V_sweep, S_300, 'r-', linewidth=2, label='300 K')
322  ax2.plot(V_sweep, S_5, 'b-', linewidth=2, label='5 K')
323  ax2.set_ylabel(r'Noise Density $S$ ($A^2/Hz$)', fontsize=12)
324  ax2.legend(loc='upper right')
325  ax2.grid(True, linestyle='--', alpha=0.7)
326
327  # Plot 3: Fano Factor
```

```python
328  ax3.plot(V_sweep, F_300, 'r-', linewidth=2, label='300 K')
329  ax3.plot(V_sweep, F_5, 'b-', linewidth=2, label='5 K')
330  # reference line for Poissonian Noise (F=1):
331  ax3.axhline(y=1.0, color='k', linestyle='--', linewidth=1.5, label='
         Poissonian Limit (F=1)')
332
333  ax3.set_ylabel(r'Fano Factor', fontsize=12)
334  ax3.set_xlabel('Voltage (V)', fontsize=12)
335  ax3.set_ylim(0, 2.0)
336  ax3.legend(loc='upper right')
337  ax3.grid(True, linestyle='--', alpha=0.7)
338
339  plt.tight_layout()
340  plt.show()
341
342  # EXERCISE 5.3. FLUCTUATION-DISSIPATION THEOREM
343  xpoints = int(np.floor(L_device / Dx))
344  U_wire = np.zeros(xpoints)
345  U_double = U_double
346
347  V_fdt = np.linspace(0.0001, 0.002, 50)
348  E_grid = np.linspace(0.001, 0.3, 500)
349
350  T_low = 5.0
351  T_room = 300
352
353  # CASE A: No Barrier, 5 K
354  I_A = compute_current(V_fdt, E_grid, U_wire, T_low)
355  S_A = compute_noise(V_fdt, E_grid, U_wire, T_low)
356  G_A = I_A / V_fdt
357  S_th_A = 4 * Kb * T_low * G_A
358
359  # CASE B: No Barrier, 300 K
360  I_B = compute_current(V_fdt, E_grid, U_wire, T_room)
361  S_B = compute_noise(V_fdt, E_grid, U_wire, T_room)
362  G_B = I_B / V_fdt
363  S_th_B = 4 * Kb * T_room * G_B
364
365  # CASE C: Double Barrier, 5 K
366  I_C = compute_current(V_fdt, E_grid, U_barrier, T_low)
367  S_C = compute_noise(V_fdt, E_grid, U_barrier, T_low)
368  G_C = I_C / V_fdt
369  S_th_C = 4 * Kb * T_low * G_C
370
371  # Plotting
372  # PLOT CASE A:
373  plt.figure(figsize=(10, 6))
374
375  # Simulated Noise
376  plt.plot(V_fdt*1000, S_A, 'b-', lw=3, label=r'Simulated Noise $S(\omega=0, V
         =0)$')
377  # Theoretical Prediction (4kTG)
378  plt.plot(V_fdt*1000, S_th_A, 'r--', linewidth=2, label=r'FDT Prediction: $4
         k_B T G(V \rightarrow 0)$')
379
380  plt.title(r'Fluctuation-Dissipation Theorem ($T=5$ K)', fontsize=14)
381  plt.xlabel('Voltage (mV)', fontsize=12)
382  plt.ylabel(r'Spectral Density ($A^2/Hz$)', fontsize=12)
383  plt.legend(fontsize=11)
384  plt.grid(True, which='both', linestyle='--', alpha=0.6)
385
```

```
386  plt.tight_layout()
387  plt.show()
388
389  fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 10))
390
391  # PLOT CASE B:
392  ax1.plot(V_fdt*1000, S_B, 'g-', lw=3, alpha=0.8, label=r'Simulated (Wire,
         300K)')
393  ax1.plot(V_fdt*1000, S_th_B, 'k--', lw=2, label=r'FDT Prediction $4k_BTG$')
394
395  ax1.set_ylabel(r'Spectral Density ($A^2/Hz$)', fontsize=12)
396  ax1.legend(fontsize=11)
397  ax1.grid(True, linestyle='--', alpha=0.6)
398
399  # PLOT CASE C:
400  ax2.plot(V_fdt*1000, S_C, 'm-', lw=3, alpha=0.8, label=r'Simulated (Barrier,
         5K)')
401  ax2.plot(V_fdt*1000, S_th_C, 'k--', lw=2, label=r'FDT Prediction $4k_BTG$')
402
403  ax2.set_xlabel('Voltage (mV)', fontsize=12)
404  ax2.set_ylabel(r'Spectral Density ($A^2/Hz$)', fontsize=12)
405  ax2.legend(fontsize=11)
406  ax2.grid(True, linestyle='--', alpha=0.6)
407
408  plt.tight_layout()
409  plt.show()
410
411  # Convergence Check
412  print(f"Convergence Ratio (Wire, 5K):   {S_A[0]/S_th_A[0]:.4f}")
413  print(f"Convergence Ratio (Wire, 300K): {S_B[0]/S_th_B[0]:.4f}")
414  print(f"Convergence Ratio (Barrier):    {S_C[0]/S_th_C[0]:.4f}")
```

Listing 1: Python script for the simulations.