# Bohmian Barrier Simulation

This folder contains a full Python-based pipeline for simulating, visualizing, and exporting movies of Bohmian trajectories interacting with a potential barrier.

---

## Folder Contents

| File / Folder | Description |

|-------------------------------|-------------|

| `bohmian\_viewer\_export.py` | Generates quantum potential (Q), plots wavefunction and Bohmian particles, and exports each time frame as a PNG to `exports/`. |

| `bohmian\_movie\_maker.py` | Assembles PNG frames into a `.mp4` movie using OpenCV. |

| `bohmian\_simulation.mp4` | Output movie file generated by the movie maker script. |

| `bohm\_trajectories\_calculator.py` | Generates Bohmian particle trajectories. |

| `bohmian\_viewer.py` | (Optional) Interactive viewer for step-through exploration (play/pause) of the simulation. |

| `tdse\_visualizer.py` | Visualizer for time-dependent Schrödinger equation (TDSE) wavefunction output. |

| `tdse\_solver.py` | TDSE solver script to evolve the initial state. |

| `output/` | Stores simulation results (`psi\_data.npz`, `bohm\_trajectories.npz`, etc.). |

| `exports/` | Stores exported PNG frames for movie creation. |

| `poetry.lock`, `pyproject.toml` | Package and dependency info (Poetry environment). |

In the first instance, simply watch the file bohmian\_simulation.mp4. This shows the result of the full simulation, including the evolution of the wavefunction, the quantum potential, and the motion of Bohmian particles. No further action is required unless you wish to change the parameters of the simulation.

If you do want to change the setup, begin by editing and running tdse\_solver.py. This script solves the time-dependent Schrödinger equation (TDSE) and allows you to modify key features of the system. You can change the height and width of the potential bump (e.g. a Gaussian barrier), the initial position and velocity of the wavefunction, and the form of the initial state itself. By default, the wavefunction is initialized as a coherent state, which is appropriate for propagation in a parabolic (harmonic) background.

After solving the TDSE, you can inspect the time-evolved wavefunction by running tdse\_visualizer.py. This allows you to verify that the wavefunction behaves as expected before proceeding.

Once the wavefunction is correct, you need to generate Bohmian particle trajectories. This is done by running bohm\_trajectories\_calculator.py. Before you do this, make sure the flag append\_trajectories is set to False. This will initialize a fresh set of trajectories. Run the script once to generate, for example, 3000 particles.

If you want to add more particles (to improve statistical resolution), set append\_trajectories = True and run the script again. This will add new particles to the existing dataset without overwriting it. You can repeat this process to accumulate as many particles as you like.

To view the full system — including particle trajectories, wavefunction amplitude, and the quantum potential — run bohmian\_viewer.py. This is an interactive viewer with a time slider and playback controls.

If you would like to export a full movie of the simulation:

1. Run bohmian\_viewer\_export.py. This will loop through all time steps and save a separate PNG image for each frame into the exports/ folder.
2. Then run bohmian\_movie\_maker.py. This script uses OpenCV to compile the PNG frames into a single .mp4 movie file named bohmian\_simulation.mp4.

You can now view the updated movie, or repeat the process with different parameters.