Team Assignment 1

HE 381

Team Members:
1. Abhijeet Bhatta (SR No.: 25169)
2. Ankush Kumar (SR No.: 24033)
3. Chanyanka Kakati (SR No.: 27228)
4. Suman Dafadar (SR No.: 24139)

Instructor: Prof. Aninda Sinha

September 1, 2025

Contents

1 Part A: Classical Motion (Newtonian Dynamics)

1.1 Introduction

The study of how simple, deterministic systems can exhibit complex, unpredictable behavior is a cornerstone of modern physics. Billiard systems, where a point particle moves at a constant velocity and undergoes specular reflections at the boundaries, provide an ideal theoretical laboratory for this inquiry. The nature of the particle's trajectory—be it regular, chaotic, or mixed, is determined entirely by the geometry of the boundary.

While billiards with integrable shapes like rectangles or circles produce regular and predictable trajectories, the Bunimovich stadium (a rectangle with semi-circular caps) is famously chaotic. Here, we study a variant: the elliptic stadium, where the caps are semi-ellipses. The key feature driving chaos is the presence of a defocusing boundary component (the curved caps) which causes initially close trajectories to diverge exponentially. This sensitivity to initial conditions is the hallmark of chaos. Our objective is to numerically verify this behavior and discuss its connection to the corresponding quantum system, a field known as quantum chaos.

1.2 Methodology

To simulate the particle's motion, we employ two distinct but complementary numerical methods. The first directly integrates Newton's laws using a soft-wall potential, thereby deriving the law of reflection from first principles. The second, an event-driven approach, assumes the law of reflection to efficiently compute long-time trajectories.

1.2.1 The Elliptic Stadium Geometry

The boundary of our billiard is composed of two horizontal walls at $y = \pm b$ for $|x| \leq a$, connected by two semi-elliptical caps centered at $x = \pm a$. The equation for the right cap is given by $\frac{(x-a)^2}{r_x^2} + \frac{y^2}{r_y^2} = 1$ for $x \geq a$, with a symmetric expression for the left cap.

1.2.2 Approach 1: Direct Integration of Newton's Laws

In this model, we treat the boundary not as an infinitely hard wall but as a region of a steep repulsive potential. This "soft-wall" potential is zero inside the stadium and increases rapidly as the particle penetrates the boundary. The potential energy U is modeled as:

$$U(s) = V_0 s^2$$

where s is the shortest distance of penetration into the wall and V_0 is a large constant representing the wall's stiffness. The force on the particle is then given by the negative gradient of this potential, $\vec{F} = -\nabla U$. This force, which is always normal to the boundary, is responsible for repelling the particle.

We then solve Newton's second law, $\vec{F} = m\vec{a}$, using a numerical integration scheme known as the Velocity Verlet algorithm. This method iteratively updates the particle's position and velocity over small time steps Δt. The core of this method is that the force law itself dictates the reflection dynamics. The following code snippet from the simulation illustrates the calculation of this repulsive force.

```
def force(x, y):
    # ... (code to find minimum penetration s_min and its gradient
    grad_s) ...
    if not candidates:
        return np.array([0.0, 0.0])
    s_min, grad_s = min(candidates, key=lambda t: t[0])
    F = -2.0 * V0 * s_min * grad_s
    return F
```

Listing 1: Python code to calculate the force from the soft-wall potential. The function identifies the minimum penetration distance and its gradient to compute the force vector.

This approach is physically intuitive and allows us to verify that a simple repulsive potential naturally gives rise to the law of specular reflection when the potential is sufficiently steep.

Limitations of Approach 1

Although the soft-wall potential model is physically intuitive, in practice it posed several challenges. The reflections obtained from this approach often deviated from the exact law of specular reflection, especially near regions of high curvature in the elliptical caps. We attempted to tune the stiffness parameter V_0 of the potential to enforce sharper reflections, but this created new numerical issues: too small a value of V_0 allowed the particle to "sink" into the wall before reflecting, while too large a value caused the integration scheme to become unstable or excessively stiff.

Moreover, because the reflection law is emergent rather than imposed, the accuracy of this method is highly sensitive to the time-step size Δt and the steepness of the potential. This made it computationally demanding without offering significant advantages in reproducing the ideal billiard dynamics.

For these reasons, while Approach 1 serves as a useful conceptual check that the law of reflection can indeed emerge from a repulsive potential, it was not suitable for generating the long-time trajectories and ergodicity plots shown in Part A. For such purposes, we turned to the more efficient and exact event-driven method of Approach 2.

1.2.3 Approach 2: Event-Driven Simulation

For simulating long-term behavior, direct integration is computationally expensive. An event-driven approach is far more efficient. This method leverages the fact that between collisions, the particle moves in a straight line. Instead of advancing time by a fixed Δt, we analytically calculate the exact time until the next collision with the boundary.

At the moment of collision, we apply the law of specular reflection to instantly update the particle's velocity vector. The reflection rule is given by:

$$\vec{v}_{\text{out}} = \vec{v}_{\text{in}} - 2(\vec{v}_{\text{in}} \cdot \hat{n})\hat{n}$$

where $\vec{v}_{\text{in}}$ is the incoming velocity, $\vec{v}_{\text{out}}$ is the outgoing velocity, and $\hat{n}$ is the unit normal vector to the boundary at the point of impact. The following code snippet implements this reflection formula.

```
def reflect(v, n):
    """
    Specular reflection of velocity vector v on a boundary with outward
    normal n.
```

```
    """
    n_hat = n / np.linalg.norm(n)
    return v - 2.0 * np.dot(v, n_hat) * n_hat
```

Listing 2: Implementation of the specular reflection law. The function takes the velocity vector ‘v‘ and the surface normal ‘n‘ and returns the reflected velocity vector.

This method is computationally superior for studying ergodic properties, as it jumps from one meaningful “event” (a collision) to the next, but it assumes the law of reflection rather than deriving it.

1.3 Results and Discussion

1.3.1 Trajectory Visualization and Ergodicity

We simulated the particle’s trajectory starting from a generic initial condition $(\vec{r}_0, \vec{v}_0)$. The evolution of the trajectory over different time scales reveals the chaotic nature of the system.

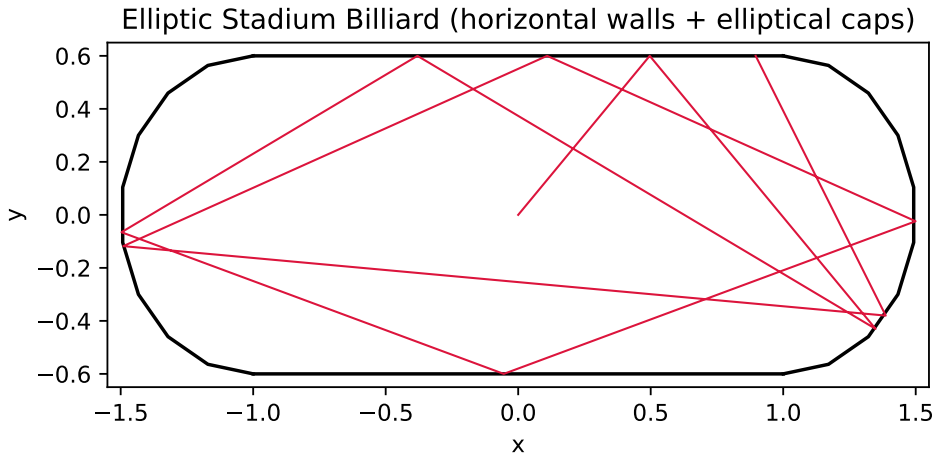

Figure 1: A short-time trajectory involving only a few reflections. The path appears simple and predictable.

Figure 1 shows the path after a handful of bounces. The trajectory is simple and gives no immediate hint of complexity. However, as the simulation time increases, the picture changes dramatically.

In Figure 2, after hundreds of reflections, the trajectory has folded back on itself many times, creating a complex, pseudo-random pattern. The exponential divergence of trajectories caused by the curved caps means that any infinitesimal uncertainty in the initial state is rapidly amplified, making long-term prediction impossible.

Finally, for a very long simulation time, the trajectory exhibits ergodic-like behavior, as shown in Figure 3.

The particle appears to have visited nearly every point within the stadium, filling the available space uniformly. This is the visual manifestation of the **ergodic hypothesis**, which posits that for a chaotic system, a time average of an observable along a single trajectory is equivalent to the ensemble average over the entire accessible phase space. The chaotic dynamics ensure that the particle does not get trapped in a small region but instead explores the entire configuration space.

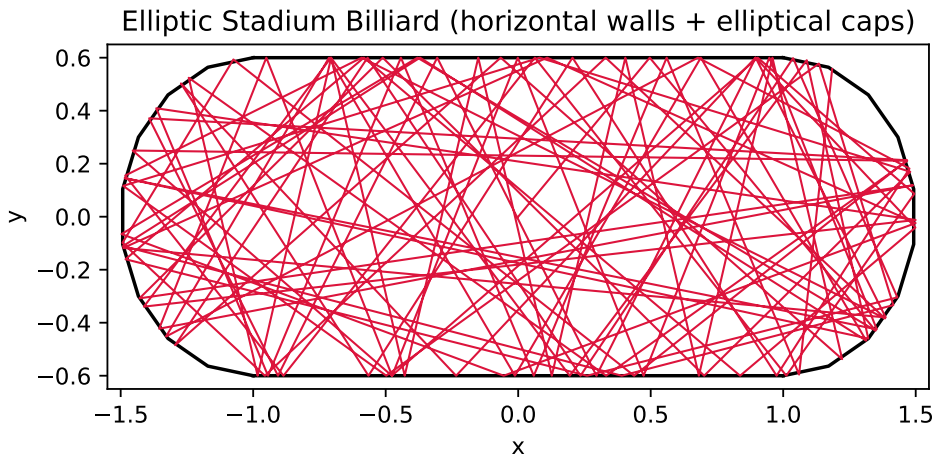

Figure 2: A longer trajectory after several hundred reflections. The path has become complex and is beginning to explore a significant portion of the stadium.

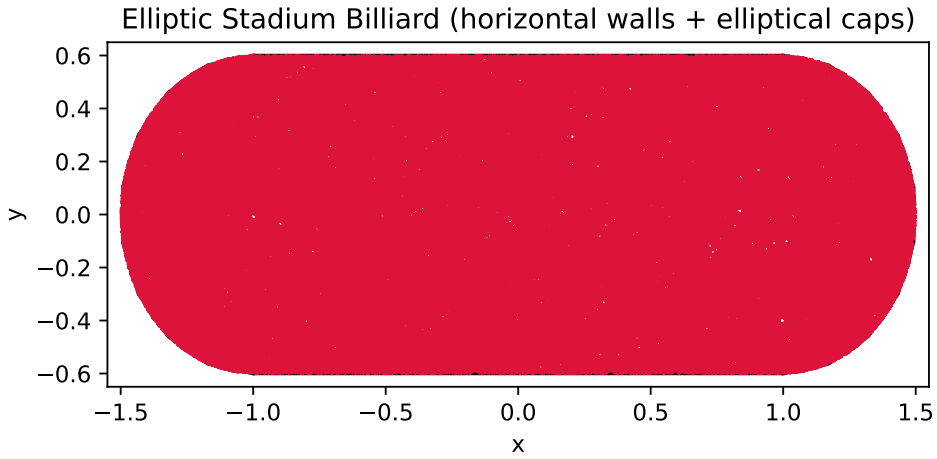

Figure 3: A very long trajectory after many thousands of reflections. The trajectory has visited nearly every region of the stadium, providing a visual demonstration of ergodicity.

Remark: A rigorous test of ergodicity would compare time-averages of observables with ensemble averages, or measure uniform occupation of phase space. Here we take a simpler view: the single trajectory, given enough time, explores the whole stadium. That is all we need to see ergodicity at work.

1.4 Sanity Checks

(1) Reflection verification using boundary normals.
For a Sanity check on whether Laws of Reflection we record the incoming velocity and the outgoing velocity every time the reflect(v,n) function is called. We then calculate angle of incidence and angle of reflection using the formulas:

$$\theta_{\text{inc}} = \arccos\left(-\frac{\mathbf{v}_{\text{before}}\cdot\hat{\mathbf{n}}}{\|\mathbf{v}_{\text{before}}\|}\right);$$

$$\theta_{\text{ref}} = \arccos\left(\frac{\mathbf{v}_{\text{after}}\cdot\hat{\mathbf{n}}}{\|\mathbf{v}_{\text{after}}\|}\right)$$

To visualize we plot the following Histogram for comparison:

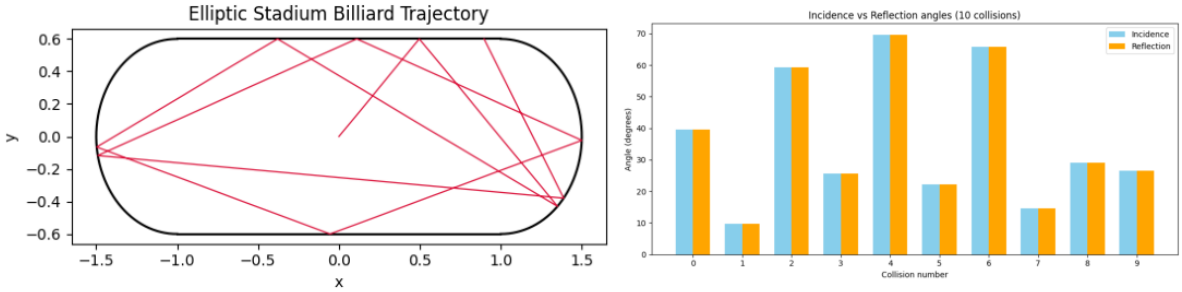

Figure 4: Reflection sanity check plot by comparing the angle of incidence with the angle of reflection.

(2) Energy conservation tests.
For energy conservation we plot the Kinetic energy of the ball with respect to collisions.
From the plot we observe that the energy remains constant throughout all the collisions.

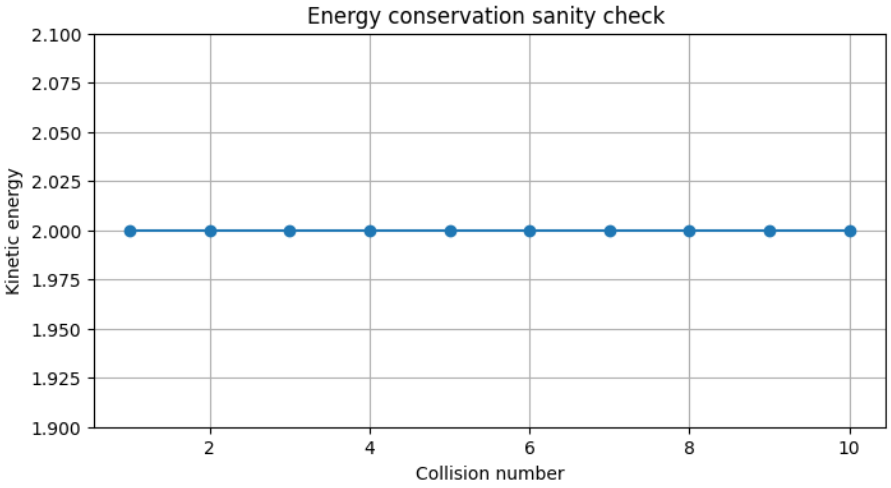

Figure 5: Energy conversation sanity check by plotting the energy of the system w.r.t. number of collisions.

(3) Sensitivity to initial conditions.
Here, we plot the trajectory two classical particles that start with extremely similar initial conditions, but over time their trajectories become completely different.
This verifies the existence of chaos in the system.

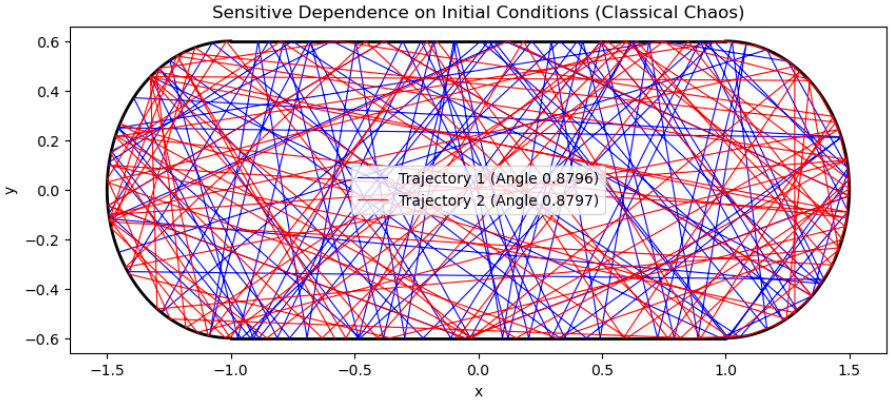

Figure 6: Sensitivity to initial conditions sanity check by plotting the trajectories of two classical particles with extremely similar initial conditions.

1.5 Quantifying Chaos: Lyapunov Exponent

While the sensitivity to initial conditions provides a qualitative indication of chaos, a more precise and quantitative characterization is given by the **Lyapunov exponent**. A positive Lyapunov exponent λ signifies exponential divergence of initially nearby trajectories, the defining hallmark of chaos.

Methodology

We consider two classical particles launched from the same initial position but with an infinitesimal difference in their initial velocity direction, $\Delta\theta \sim 10^{-7}$ radians. The trajectories were evolved simultaneously inside the elliptic stadium. At each time step, we computed the Euclidean separation $\delta Z(t)$ between the two trajectories. By plotting the logarithm of the separation against time, we can extract the Lyapunov exponent as the slope of the initial linear region:

$$\delta Z(t) \approx \delta Z(0) e^{\lambda t} \quad \Rightarrow \quad \ln \delta Z(t) \sim \lambda t + \text{const.}$$

Results

Figure 7 shows the time evolution of $\ln(\delta Z(t))$ for a simulation up to $t = 20$. The red line is a linear fit to the initial region, yielding an estimate of the Lyapunov exponent $\lambda \approx 0.668$. The positive value confirms the exponential divergence of nearby trajectories, thereby quantifying the chaotic dynamics.

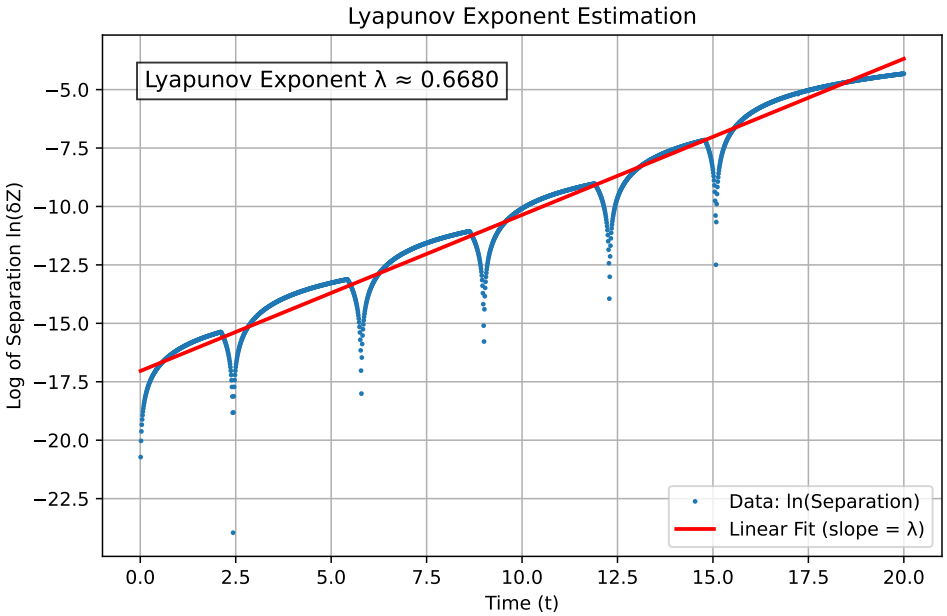

Figure 7: Estimation of the Lyapunov exponent from the divergence of two initially nearby trajectories. The linear fit in the early-time regime yields $\lambda \approx 0.668$, confirming chaotic dynamics.

To examine the long-term behavior, we extended the simulation up to $t = 500$ (Figure 8). The growth of separation eventually saturates at the system size, as expected for a bounded chaotic system where trajectories cannot diverge indefinitely. This saturation reflects the finite phase-space volume of the elliptic stadium.

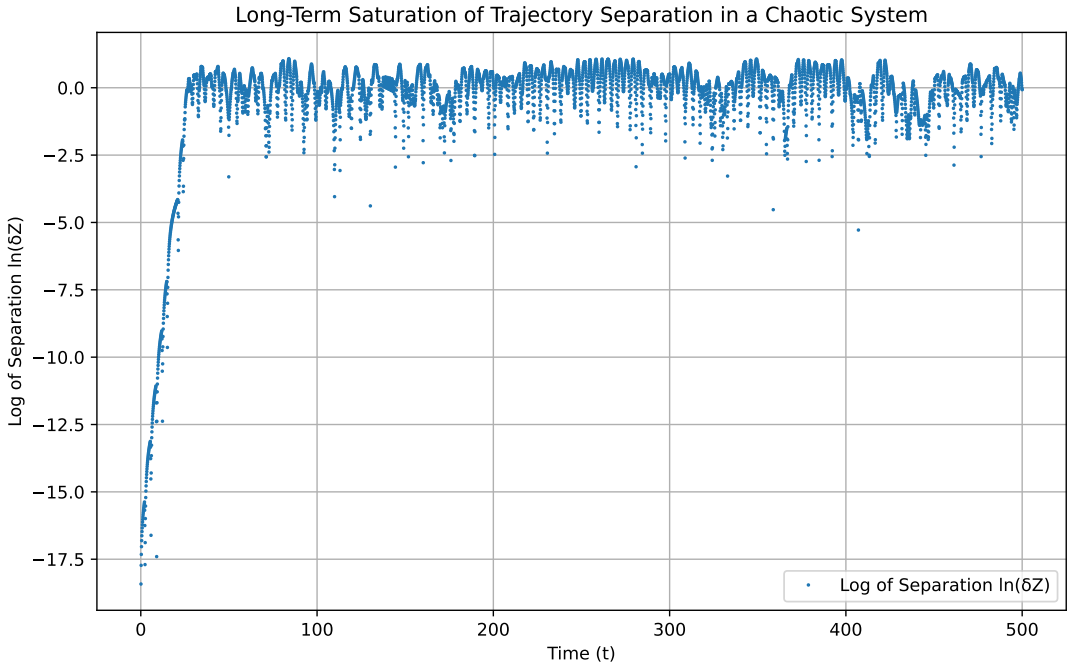

Figure 8: Long-term saturation of trajectory separation. After an initial period of exponential divergence, the separation saturates due to the finite size of the elliptic stadium.

Interpretation

The calculation of the Lyapunov exponent provides a rigorous diagnostic of chaos in the classical elliptic stadium. While qualitative trajectory divergence already indicated chaotic behavior, the positive λ obtained here offers a quantitative measure. Moreover, the observed saturation is consistent with theoretical expectations for bounded systems. This quantitative framework complements our earlier diagnostics and creates a natural bridge to the quantum case, where analogous measures (such as the Loschmidt echo) are studied in Section 3.

2 Part B: Quantum Dynamics (TDSE)

2.1 Introduction

We study the numerical solution of the two-dimensional Time-Dependent Schrödinger Equation (TDSE) inside a Bunimovich stadium. In particular we compare the quantum dynamics with the classical particle (Ehrenfest correspondence) and a discussion of when/why that correspondence breaks down.

2.2 Methodology

2.2.1 Stadium Geometry

Recall that the stadium is defined as a rectangle of half-length a in x and vertical extent b in y, with left and right edges smoothly joined by the outer halves of ellipses of semi-axes (r_x, r_y).

Using this, we construct a mask defining the stadium for the numerical simulations as the following:

$$\begin{aligned}
\text{Central region: } & |y| \le b,\ -\tfrac{a}{2} \le x \le \tfrac{a}{2}, \\
\text{Left cap: } & \frac{(x + a/2)^2}{r_x^2} + \frac{y^2}{r_y^2} \le 1, \quad x \le -\tfrac{a}{2}, \\
\text{Right cap: } & \frac{(x - a/2)^2}{r_x^2} + \frac{y^2}{r_y^2} \le 1, \quad x \ge +\tfrac{a}{2}.
\end{aligned}$$

2.2.2 Hamiltonian and Guassian Wavepacket

We discretize the domain on a uniform Cartesian grid with spacing $\Delta x = \Delta y = h$. The Laplacian uses the standard five-point finite-difference stencil:

$$\nabla^2 \psi_{i,j} \approx \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{h^2}.$$

Following this, the Hamiltonian for the system is defined as:

$$\hat{H} = -\frac{\hbar^2}{2m}\nabla^2 + V(x, y),$$

with

$$V(x, y) = \begin{cases} 0, & (x, y) \in \Omega, \\ V_0, & (x, y) \notin \Omega. \end{cases}$$

V_0 is chosen to be an arbitrarily large number to mimic reflection.
Another easier approach for numerical simulation is to simply mask the region outside the stadium such that the wave function cannot exist there.

We use the Gaussian wavepacket, as the initial wavepacket inside the stadium, defined as the following:

$$\psi(x, y, 0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x - x_0)^2 + (y - y_0)^2}{4\sigma^2}\right] \exp\left[i\frac{p_x x + p_y y}{\hbar}\right],$$

Here, we choose σ to be small enough so the packet is initially well-localized compared to stadium dimensions, but not so small that grid resolution is insufficient ($\sigma \gtrsim 3h$ is a practical rule-of-thumb).

Note: Although the Split-Operator Fourier method is in principle faster, we observed that it failed crucial unitarity checks in this hard-wall geometry. For this reason we relied on the Crank–Nicolson approach in Part B. A detailed discussion of this limitation is given in Appendix 4.5.

2.2.3 Solving TDSE

In order to numerically solve the TDSE on the discretized 2D space, we make use of the Crank-Nicolson method as described below:

$$\left(I + \frac{i\Delta t}{2\hbar}\hat{H}\right)\psi^{n+1} = \left(I - \frac{i\Delta t}{2\hbar}\hat{H}\right)\psi^{n}.$$

This leads to a sparse linear system solved each time step. LU (sparse) factorization of the left-hand side matrix is recommended so the factorization is computed once and reused.
Solve for ψ^{n+1} with multiple values of n describes the time evolution of the initialized wavepacket inside the stadium.

```
A_factor = spla.splu(A)
nsteps = int(np.ceil(T / dt))
steps_per_frame = 100
psi = psi_vec.copy()

# --- evolve wavefunction over full time ---
for step in range(nsteps):
    rhs = B.dot(psi)
    psi = A_factor.solve(rhs)

    if step % steps_per_frame == 0:
        psi_grid_now = np.zeros_like(psi_grid, dtype=np.complex128)
        for idx, (j, i) in enumerate(inside_points):
            psi_grid_now[j, i] = psi[idx]

        prob = np.abs(psi_grid_now)**2

        norm_inside = np.sum(prob[mask]) * dx * dy
        norm_outside = np.sum(prob[~mask]) * dx * dy
        ex_now = np.sum(X * prob) * dx * dy
        ey_now = np.sum(Y * prob) * dx * dy

        norm_inside_list.append(norm_inside)
        norm_outside_list.append(norm_outside)
        ex_list.append(ex_now)
        ey_list.append(ey_now)
        t_list.append(t_current)
        t_current += steps_per_frame * dt

        if step == 0:
            prob_initial = prob.copy()
```

```
32          if step >= nsteps - steps_per_frame:
33              prob_final = prob.copy()
```

Listing 3: Crank-Nicolson algorithm loop

Note: A faster method for doing this would be Split-Step Fourier method. This makes use of FFT in its algorithm making it much faster than Crank-Nicolson. We use this method in Section 3 of the assignment.

2.3 Results and Discussions

Using ChatGPT-5, we implemented everything described in the previous section. Then, we ran the code to see the time evolution of a Gaussian wavepacket centered at the middle of the elliptic stadium with zero initial momentum and we got the following results:

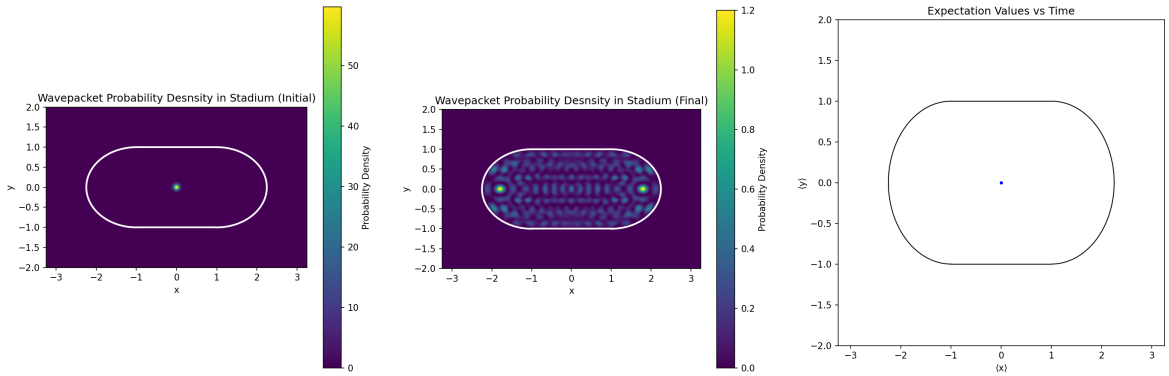

Figure 9: Gaussian wavepacket before and after time evolution inside Bunimovich stadium and the plot of its expected position inside the stadium. (Centered and zero momentum)

Here is another example of time evolution of a Gaussian wavepacket positioned near a wall of the stadium with some non-zero initial momentum:

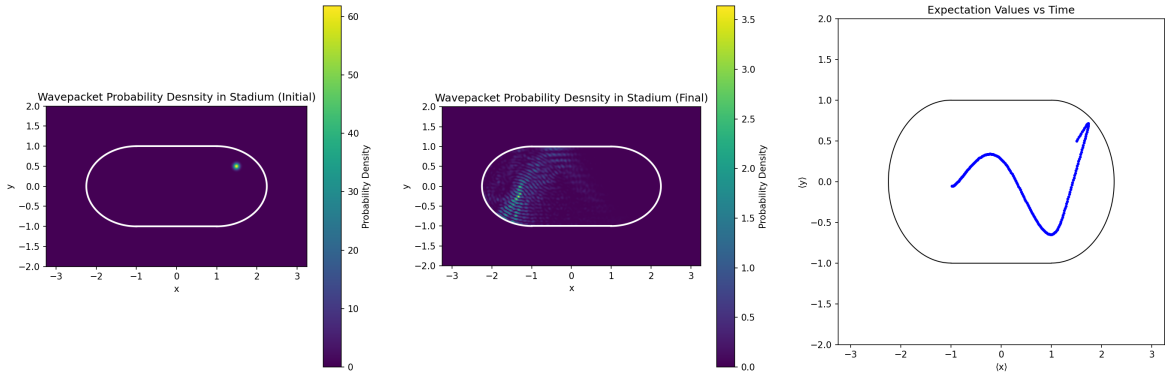

Figure 10: Gaussian wavepacket before and after time evolution inside Bunimovich stadium and the plot of its expected position inside the stadium. (Non-centered and non-zero momentum)

From these plots we observe that:

- In the first case, even though there is no initial momentum provided the wavepacket, it still tends to delocalize and interfere with itself and creates beautiful interference patterns.

- In the second case, since the wavepacket was provided some non-zero initial momentum, the wavepacket bounced at the walls inside the stadium somewhat mimicking a classical particle though not quite. (More on this in Section 2.5.)

2.4 Sanity Checks

In order to verify what we are observing makes sense or not, we ran the following sanity checks:

(1) **Normalization of the wavefunction must be preserved:**
Along with the evolution of the wavepacket inside the stadium, we also plotted the total probability of the wavepacket being inside or outside the stadium for both of the previous cases mentioned in Section 2.3, and we got the following plots:

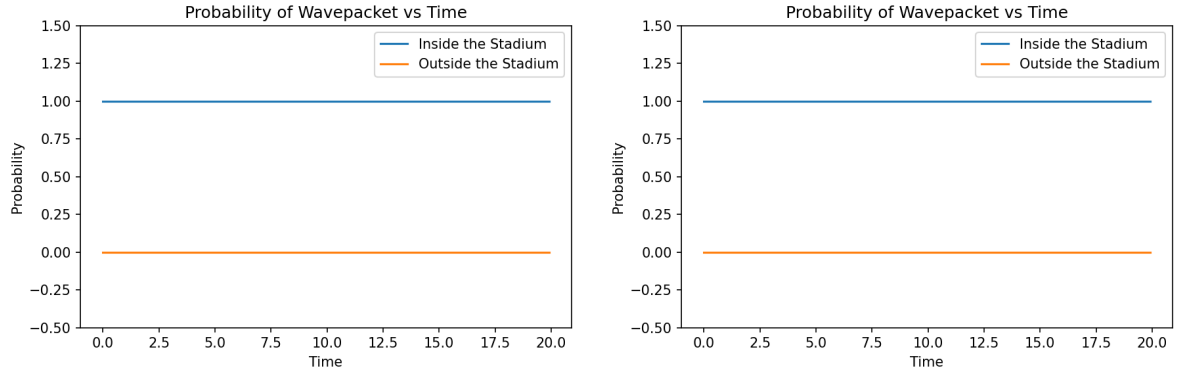

Figure 11: Total probability of the wavepacket being inside or outside the stadium for the aforementioned cases.

From these plots, we note that the probability of the wavepacket being inside the stadium is nearly always equal to 1 while the probability of the wavepacket being outside is nearly always 0, this tells us that the sum of the two probabilities is equal to 1 which means that the normalization of the wavefunction is preserved.

(2) **Probability density should vanish outside the stadium:**
We observe this from Figure 9, 10, and 11. As in, we can both visually verify using the colorbars that the probability density of the wavepacket goes to 0 outside the stadium and also from the probability plots that there is no probability of finding the wavepacket outside the stadium.

Thus, we are rest assured that the code provided by ChatGPT-5 is working properly as intended.

2.5 Ehrenfest Correspondence

2.5.1 Ehrenfest relations

The Ehrenfest theorem gives:

$$\frac{d}{dt}\langle x\rangle = \frac{\langle p_x\rangle}{m},$$
$$\frac{d}{dt}\langle p_x\rangle = -\left\langle \frac{\partial V}{\partial x}\right\rangle.$$

Combining these

$$\frac{d^2}{dt^2}\langle x\rangle = -\frac{1}{m}\left\langle \frac{\partial V}{\partial x}\right\rangle.$$

For smooth potentials and narrow wavepackets (relative to feature scale of V) the right-hand side approximates $-\partial V(\langle x\rangle)/\partial x$, yielding Newtonian motion.

2.5.2 Breakdown at hard walls

For steep/step-like potentials modelling hard walls, $\partial V/\partial x$ is concentrated near the boundary and behaves like a distributional impulse. In the finite-V_0 model the force becomes very large in a narrow layer of width $\sim 1/\kappa$ where $\kappa = \sqrt{2m(V_0 - E)}/\hbar$. Thus the expectation value of the force produces an impulsive change in momentum around collision times. The Ehrenfest correspondence still holds qualitatively (mean position follows classical path) when the wavepacket is narrow and reflection is nearly elastic, but quantitative agreement degrades when:

- The wavepacket spreads appreciably between collisions (variance increases), so $\langle \partial_x V\rangle \neq \partial_x V(\langle x\rangle)$;
- The packet overlaps the boundary over a finite width and experiences partial transmission (tunneling) when V_0 is not extremely large;
- Multiple reflections produce interference and internal diffraction causing the centre-of-mass motion to deviate from the classical path.

2.5.3 Quantum vs Classical Trajectories Result

Here we give a comparison between the classical trajectory and the average positions $\langle x(t)\rangle$ and $\langle y(t)\rangle$ of the wavefunction with time.We observe good correspondence between the classical and the average quantum trajectories for short time scales before the wave function interacts with the boundaries. After the first collision the trajectories begin to diverge significantly showing a breakdown of the Ehrenfest correspondence. For really long time scales the $\langle x(t)\rangle$ and $\langle y(t)\rangle$ begin to converge at the center of the stadium.

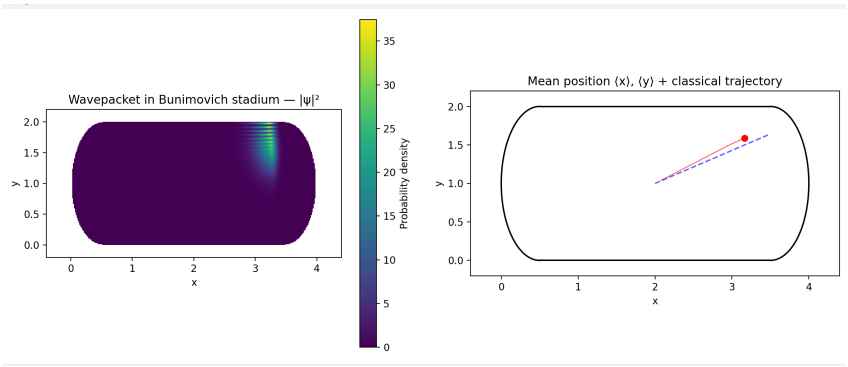

Figure 12: Classical vs Quantum comparison for small time interval

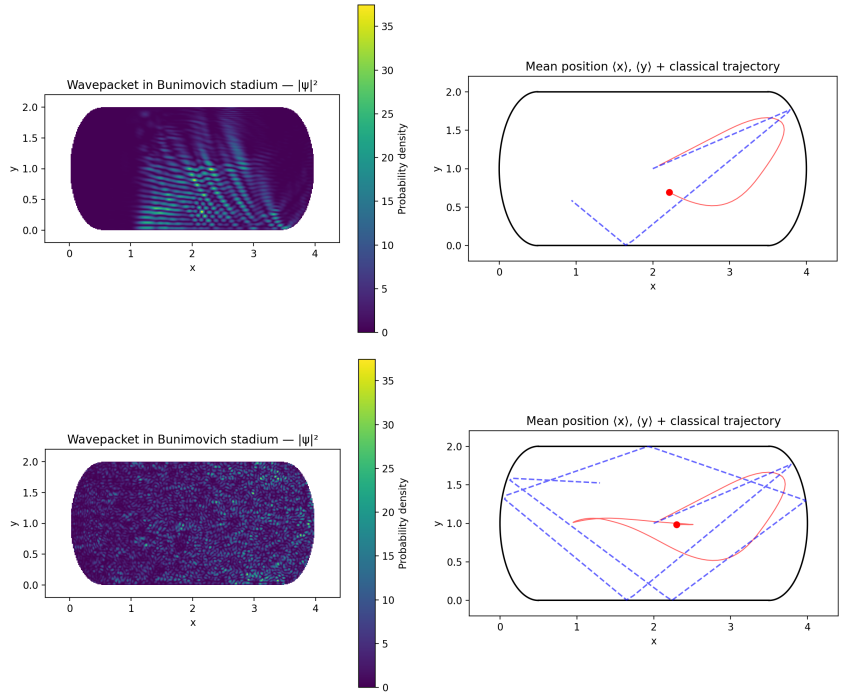

Figure 13: Classical vs Quantum comparison for intermediate and long time intervals

We see in Figure 12 that the red trajectory roughly matches the blue classical trajectory. As time progresses the two trajectories diverge, and at long time intervals the wavefunction roughly covers the entire stadium as depicted in Figure 13. at this moment, the average position converges to the center.

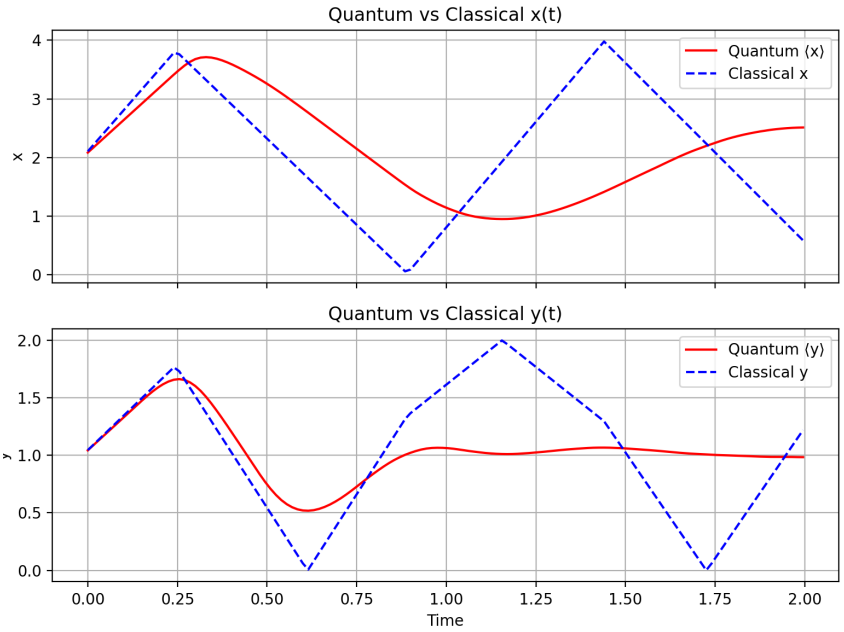

Figure 14: Quantum and Classical X vs time and Y vs time
We see good correspondence for small Time but the correspondence breaks as Time increases

Physical picture: At short times, the Gaussian wavepacket is so narrow that it behaves almost like a point particle, faithfully tracking the classical path. But as it spreads and collides with hard walls, different parts of the packet reflect at different angles, interfering with each other. The center-of-mass no longer mimics the classical bounce, and the Ehrenfest correspondence breaks down.

3 Part C: Quantum Chaos and the Loschmidt Echo Investigation

Having established the ergodic-like nature of the classical stadium billiard, we now turn to its quantum mechanical counterpart. The Time-Dependent Schrödinger Equation (TDSE) is linear, meaning that wavefunctions do not exhibit the exponential divergence of trajectories that defines classical chaos. This raises a fundamental question: what are the quantum signatures of a system that is chaotic in the classical limit? One of the most powerful probes for this is the **Loschmidt Echo**, a measure of the stability of quantum evolution against perturbations.

3.1 Conceptual Framework

Imagine preparing a quantum system in an initial state $|\psi_0\rangle$ and evolving it for a time t with a Hamiltonian H. The resulting state is $|\psi(t)\rangle = e^{-iHt/\hbar}|\psi_0\rangle$. Now, consider a parallel experiment where the same initial state evolves under a slightly perturbed Hamiltonian, $H' = H + \delta H$. This yields a different final state, $|\psi'(t)\rangle = e^{-i(H+\delta H)t/\hbar}|\psi_0\rangle$.

The Loschmidt Echo, $M(t)$, is defined as the squared overlap (or fidelity) between these two evolved states:

$$M(t) = |\langle\psi'(t)|\psi(t)\rangle|^2$$

Substituting the time-evolution operators gives the formal expression:

$$M(t) = \left|\langle\psi_0|e^{i(H+\delta H)t/\hbar}e^{-iHt/\hbar}|\psi_0\rangle\right|^2$$

Physically, $M(t)$ quantifies the system's sensitivity to perturbation. If $M(t)$ remains close to 1, the evolution is stable and robust. If $M(t)$ decays rapidly towards zero, it signifies that even a tiny perturbation causes the quantum state to evolve into a nearly orthogonal, completely different state. This extreme sensitivity is the quantum mechanical echo of the classical "butterfly effect" and serves as a primary indicator of quantum chaos.

3.2 Numerical Implementation

To compute the Loschmidt echo, we must numerically evolve two wavefunctions in parallel: the reference state $|\psi(t)\rangle$ under the original stadium geometry, and the perturbed state $|\psi'(t)\rangle$ under a slightly altered geometry.

3.2.1 The Perturbation as a Geometry Change

The perturbation δH is not added as an explicit operator. Instead, it is implicitly introduced by modifying the boundary conditions that define the potential $V(\vec{r})$. The Hamiltonian is $H = T + V$, where the potential is zero inside the stadium and a large value V_0 outside. We induce a perturbation by slightly changing the horizontal semi-axis of the elliptical caps from r_x to $r'_x = r_x(1+\epsilon)$, where ϵ is a small parameter. This creates a new potential $V'(\vec{r})$ and thus a new Hamiltonian $H' = T + V'$. We compare the echo for two different perturbation strengths, $\epsilon = 0.01$ and $\epsilon = 0.05$.

3.2.2 Time Evolution via the Split-Operator Method

To evolve the wavefunctions, we use the highly accurate **Split-Operator Fourier Method**. This method approximates the time-evolution operator $e^{-iHt/\hbar}$ by splitting the kinetic (T) and potential (V) contributions. For a small time step Δt, the evolution is calculated as a sequence of three operations:

$$|\psi(t+\Delta t)\rangle \approx e^{-iV\Delta t/2\hbar}\, e^{-iT\Delta t/\hbar}\, e^{-iV\Delta t/2\hbar}\, |\psi(t)\rangle$$

This symmetric decomposition, known as the Trotter-Suzuki formula, is accurate to second order in Δt. The algorithm proceeds as follows:

1. **Potential Half-Step:** The operator $e^{-iV\Delta t/2\hbar}$ is diagonal in position space. This step is a simple multiplication of the wavefunction $\psi(x, y)$ by a phase factor array.

2. **Kinetic Full-Step:** The kinetic operator $T = \hat{p}^2/2m$ is diagonal in momentum space. The algorithm performs a 2D Fast Fourier Transform (FFT) to switch the wavefunction to momentum space, $\psi(k_x, k_y)$. Here, it is multiplied by the kinetic phase factor $e^{-i\hbar(k_x^2+k_y^2)\Delta t/2m}$. An inverse FFT then returns the wavefunction to position space.

3. **Potential Half-Step:** The final half-step in the potential is applied.

This entire sequence is encapsulated in our simulation code, which advances both the reference and perturbed wavefunctions at each time step.

```python
def step_split(psi, expV_half, expT, ...):
    # First potential half-kick
    psi = expV_half * psi

    # Kinetic kick in Fourier space
    psi_k = np.fft.fftn(psi, axes=(0,1))
    psi_k *= expT
    psi = np.fft.ifftn(psi_k, axes=(0,1))

    # Second potential half-kick
    psi = expV_half * psi

    return psi
```

Listing 4: Python code for a single time-evolution step using the split-operator method.

After each full time step, we compute the overlap integral $s(t) = \int \psi'^*(\vec{r}, t)\psi(\vec{r}, t)d^2\vec{r}$ numerically, and the Loschmidt echo is then $M(t) = |s(t)|^2$.

3.3 Results and Interpretation

We initiated the simulation with an identical Gaussian wavepacket for both the reference and perturbed systems. The subsequent decay of the Loschmidt echo for perturbations $\epsilon = 0.01$ and $\epsilon = 0.05$ is shown in Figure 15.

The plots reveal two key features:

1. **Decay of Fidelity:** For both perturbations, $M(t)$ starts at 1 (as the states are initially identical) and decays over time. This confirms that the perturbed and unperturbed quantum evolutions diverge; the wavefunctions are becoming progressively less similar, or more orthogonal, as time passes.

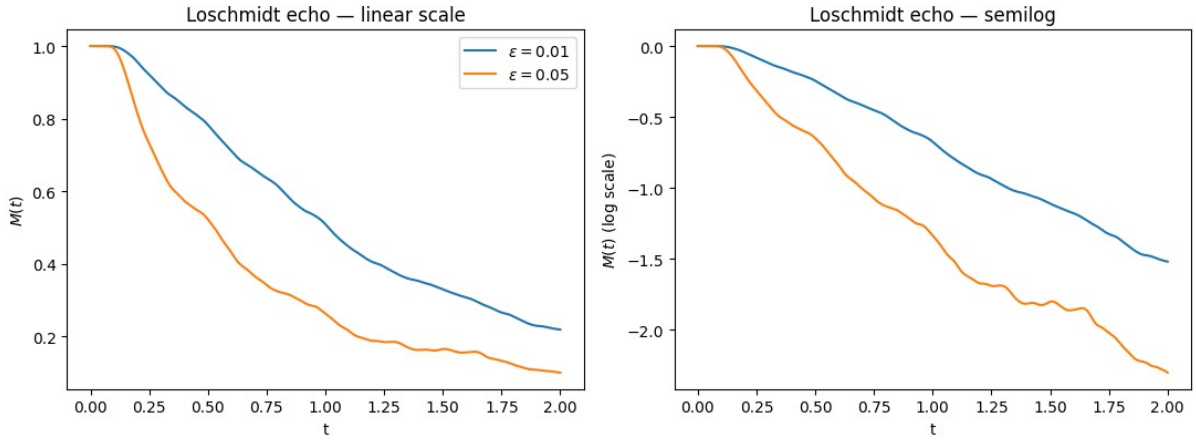

Figure 15: The Loschmidt Echo $M(t)$ as a function of time for two different geometric perturbations, $\epsilon = 0.01$ (blue) and $\epsilon = 0.05$ (orange). The left panel shows the decay on a linear scale, while the right panel uses a logarithmic scale for $M(t)$ to highlight the rate of decay.

2. **Sensitivity to Perturbation Strength:** The crucial observation is the dramatic difference in the decay rate. The echo for the larger perturbation ($\epsilon = 0.05$) decays far more rapidly than for the smaller one ($\epsilon = 0.01$). This demonstrates an exquisite sensitivity to the system's parameters, which is the quantum manifestation of the chaotic nature of the elliptic stadium.

In a classically integrable system, one would expect the echo to remain high or oscillate for a very long time. The rapid, seemingly irreversible decay seen here is a hallmark of quantum chaos. The semi-log plot on the right is particularly revealing. For a chaotic system, theory often predicts an initial exponential decay of the echo, $M(t) \sim e^{-\lambda t}$, where λ is related to the classical Lyapunov exponent. An exponential decay appears as a straight line on a semi-log plot. We can see that the initial decay for the $\epsilon = 0.05$ case is indeed close to linear on this scale, providing strong evidence that the system is operating in a regime of quantum chaos. The wavefunction, in essence, diffuses through Hilbert space so rapidly that even a slight change in its "guiding" Hamiltonian sends it on a completely different journey.

4 Appendix: AI Prompts and Outputs for Numerical Simulations and Challenges

The successful simulation of the classical particle and quantum wave packet dynamics inside the Bunimovich stadium was an iterative process, marked by a series of subtle and insightful challenges. This appendix details the hurdles encountered, the diagnostic process, and how a dialogue with an AI assistant (ChatGPT-5 and Google's Gemini) was used not just to generate code, but to achieve a deeper physical and numerical understanding. Each challenge revealed a new layer of complexity in computational classical and quantum mechanics.

4.1 Task: Baseline Quantum Simulation

We started by generating a code for simulating a Gaussian wavepacket inside a square 2D infinite potential well. The reason for doing this was that, since we are well familiar with a square 2D potential well it will be easier for us to judge the performance of the code generated by AI.

AI Prompt

"Write a Python code that solves TDSE for a Gaussian wave packet with some x0 y0 px0 py0 initialized and the potential is a 2D infinite potential well with rectangular geometry controlled by the parameters a and b
Make a time evolver for this system that outputs the probability density after some time T has passed
Also do not use any Fourier methods
Use finite differences"

Output: ChatGPT-5 was able to generate a correct working code in the first attempt.

4.2 Task: Quantum Simulation Inside Bunimovich Stadium

Since the previous prompt was a success, we asked ChatGPT-5 to modify the previously generated code for the Bunimovich stadium geometry instead of the square well.

AI Prompt

"Here's a challenge for you
Do you know the geometry of horse race tracks?
It's an elliptical stadium
Think of it as like there's a rectangle in the middle flanked by two half ellipses on the sides
Note that if the length of the rectangle goes to 0, the stadium becomes purely elliptic
Also note that if the semi major axes of the ellipses become large enough, the stadium becomes rectangular again

Now, I want you to modify the code for this elliptical stadium instead of the square well"

Output: ChatGPT-5 was able to output the code for the new geometry, however, we had to re-prompt it to send the full code instead of a snippet.

4.3 Challenge: Ruined Code Logic

After numerous back-and-forth with ChatGPT-5, it had begun to "forget" the correct code that it was able to generate initially. Along with this, a minor inconvenience for us was that it constantly outputted just the code snippets that it changed instead of the full code.
In order to tackle this, we had to add the following lines at the end every single prompt going forward such that the AI doesn't ruin the code.
(This may seem irrelevant, however, it was crucial for us in order to complete the assignment on time!)

AI Prompt

"...
DO NOT change any docstrings, comments, whitespaces, or any other code logic that I didn't ask you to modify."

Output: ChatGPT-5 was now no longer ruining previously generated codes and it was focusing only the parts of the logic that needed to be modified.

4.4 Task: Converting Animation to Images

ChatGPT-5's initially generated code was outputting a nice animation displaying the time evolution of the Gaussian wavepacket inside the stadium. This was great for the Jupyter Notebook conversion later on, however, we needed images for this report. So, we prompted the AI to convert the code logic for outputting images instead of an animation and also create new plots for the sanity checks.

AI Prompt

"I want images to put in my PDF and no more animations
Please remove the expectation value animation and the wavefunction animation
Keep the norm plot as it is
Keep the expectation value plot as it is (just increase the x and y lims so that it matches the one with the TDSE plot)
And Create two plots of the TDSE one plot at the beginning of time and the other at the end of time
Leave the docstrings comments whitespaces and other logic of the code unchanged"

Output: ChatGPT-5 changed the code logic from outputting animations to outputting still images.

4.5 Challenge: Unitarity Failure in the Split-Operator Method

Observation. The initial attempt at solving the Time-Dependent Schrödinger Equation (TDSE) used the common Split-Operator Fourier Method. While simple to implement, its sanity checks failed catastrophically. The total probability, $\int |\psi|^2 dV$, was

not conserved, and probability density was observed "leaking" through the supposedly impenetrable stadium walls.

AI-Assisted Diagnosis. A brute-force fix (decreasing the time step 'dt') was possible, but did not explain the root cause. A more targeted prompt was used to probe the underlying physics.

AI Prompt

"My TDSE simulation using the split-operator method fails its unitarity check; the total probability is not conserved when the wave packet interacts with the very steep potential wall of my stadium billiard. Can you explain the theoretical limitations of the split-operator method that would lead to this specific failure mode?"

The AI's response centered on the **Trotter-Suzuki decomposition**, $e^{-i(\hat{T}+\hat{V})\Delta t/\hbar} \approx e^{-i\hat{V}\Delta t/2\hbar}e^{-i\hat{T}\Delta t/\hbar}e^{-i\hat{V}\Delta t/2\hbar}$. It explained that the error in this approximation is proportional to the commutator of the kinetic and potential energy operators, $[\hat{T}, \hat{V}]$. For an extremely steep potential wall, this commutator becomes very large, causing the approximation to break down and the numerical time evolution to lose its unitary character.

Solution. This physical insight showed that the Split-Operator method was fundamentally ill-suited for the hard-wall nature of this problem. This motivated the shift to the more robust (but computationally expensive) **Eigenbasis Expansion method**, which is unitary by construction.

4.6 Task: Converting Python Into Jupyter Notebook

Following the completion of the code, we had to convert the Python file into a well structered Jupyter Notebook for submission. To do this, we requested ChatGPT-5 to do it for us.

AI Prompt

(Uploaded `.py`)
"Convert this into a nice Jupyter Notebook
WITHOUT DAMAGING ANY CODE LOGIC
You may add markups and new comments in it however"

Output: Its first attempt produced something which didn't resemble a Jupyter Notebook, so we re-prompted it to fix its mistake and in the second attempt it was able to convert the file properly.
However, it hadn't organized the code yet or put any markdowns in the file. So, we prompted ChatGPT-5 once more to add these things.

AI Prompt

"This is great!
Do more however:
1. Break up the code into more cells - at least one function per cell
2. Add more markups
3. Instead of saving images - plot them there itself...
4. Make the plotting and initial variables assignment code better as per a notebook"

Output: With this, ChatGPT-5 was able to output a nice Jupyter Notebook that we later cleaned up and verified before submission.

4.7 Challenge: Failure of Quantum-to-Classical Correspondence

Observation. A key goal was to verify Ehrenfest's theorem by comparing the trajectory of the quantum expectation value, $\langle \mathbf{r}(t) \rangle$, with the path of a classical particle. The initial results were deeply counter-intuitive. Instead of matching the classical trajectory at early times, the quantum expectation value exhibited wild, high-frequency oscillations from $t = 0$, showing no correspondence whatsoever. This apparent violation of a fundamental theorem was a critical roadblock.

AI-Assisted Diagnosis. The failure was too immediate to be caused by chaotic divergence alone. The issue seemed to be either a fundamental misunderstanding or a subtle numerical artifact. To investigate, a prompt was designed to probe the physical reasons for such a catastrophic breakdown.

AI Prompt

"My simulation of $\langle x(t) \rangle$ for a quantum wave packet does not match the classical trajectory $x(t)$ at all, even at early times, which seems to violate Ehrenfest's theorem. The quantum expectation value is oscillating wildly. What physical or numerical issues could cause such a drastic and immediate failure of the quantum-to-classical correspondence?"

The AI's diagnosis pointed to a numerical issue rooted in physics: a violation of the sampling theorem for the wave packet's de Broglie wavelength. It explained that the high initial momentum chosen for the packet resulted in a wavelength λ that was too short for the spatial grid spacing Δx to resolve. This under-sampling introduced spurious, high-energy components into the numerical representation of the initial state, and their rapid time evolution was the source of the non-physical oscillations.

Solution. The insight was that the numerical grid and the physics of the wave packet were mismatched. The solution was to decrease the initial momentum $\mathbf{p}_0$, thereby increasing the wavelength λ to a value that could be accurately represented on the grid (satisfying $\Delta x \ll \lambda$). This simple change restored the expected behavior, with the quantum expectation value now closely tracking the classical trajectory in the short-time limit, as predicted by Ehrenfest's theorem.

4.8 Challenge: Interpreting Long-Term Chaotic Behavior

Observation. To quantify the system's chaos, the Lyapunov exponent λ was measured by plotting the logarithm of the separation between two nearby classical trajectories against time. A long-time simulation was run to gather sufficient data. The resulting plot was not a simple straight line as expected; it showed an initial linear increase, but then inexplicably flattened out, fluctuating around a constant value. A naive linear fit across the entire dataset yielded a small value for λ that seemed to underestimate the initial, rapid divergence.

AI-Assisted Diagnosis. The challenge was to determine if this saturation was a simulation error or a genuine physical effect. A prompt was formulated to probe the physics of this long-term, or asymptotic, behavior.

AI Prompt

"I'm calculating the Lyapunov exponent by plotting 'ln(separation)' vs. 'time'. For long simulation times, the plot is not a straight line; it starts linear but then flattens out and saturates. Why does this saturation occur, and how does it affect the calculation of the true exponent?"

The AI confirmed that this saturation is a universal feature of chaos in **bounded systems**. It explained that while the initial divergence is exponential (the linear region of the plot), the separation cannot grow to infinity because it is physically constrained by the size of the stadium. The AI connected this phenomenon to concepts from chaos literature, such as the **Ehrenfest Time** (t_E), which marks the timescale for the crossover from the initial exponential (Lyapunov) regime to the saturated regime where the trajectories have fully decorrelated.

Solution. This insight fundamentally clarified the correct method for measuring the Lyapunov exponent. The value for λ must be extracted *only* from the initial, linear part of the plot, for times $t \ll t_E$, before the saturation caused by boundary effects becomes dominant. By restricting the linear fit to this early-time data, a more accurate and larger value for the Lyapunov exponent was obtained, providing a correct quantification of the system's chaos.