PHYS 410 - Project 2

Felipe Garavelli
46427951

December 4, 2025

Contents

1 Introduction

This project investigates the dynamics of quantum mechanical systems through the numerical solution of the Time-Dependent Schrödinger Equation (TDSE), implemented in MATLAB. The approach utilizes two implicit finite difference techniques. Specifically the Crank-Nicolson method for the one-dimensional case and the Alternating Direction Implicit (ADI) scheme for two spatial dimensions. While the present model operates within a non-dimensionalized framework and focuses on single-particle dynamics, it captures essential quantum phenomena such as wave packet spreading, tunneling, and interference. The primary objective of this project is to simulate and visualize the time evolution of wavefunctions interacting with various potentials, including rectangular barriers, wells, and double slits. The project also validates the accuracy of the one and two-dimensional solvers through convergence testing.

2 Theory

2.1 The Time-Dependent Schrödinger Equation

The simulations are based on the Time-Dependent Schrödinger Equation (TDSE), which describes the quantum state evolution of a particle in a potential field. The wavefunction, ψ, encapsulates the probability amplitude of finding a particle at a given position and time. For the one-dimensional case, the TDSE is given by:

$$i\psi(x,t)_t = -\psi_{xx} + V(x,t)\psi \tag{1}$$

where $V(x,t)$ represents the potential. This equation is solved on the domain $0 \leq x \leq 1$ subject to Dirichlet boundary conditions $\psi(0,t) = \psi(1,t) = 0$.

For the two-dimensional problem, the equation generalizes to include the second spatial derivative, resembling a diffusion equation with an imaginary diffusion constant:

$$i\psi(x,y,t)_t = -(\psi_{xx} + \psi_{yy}) + V(x,y)\psi \tag{2}$$

The 2D domain is defined as the unit square ($0 \leq x, y \leq 1$), with the wavefunction equalling 0 at all boundaries. In both dimensions, the potential V may represent free space ($V = 0$), rectangular barriers/wells, or complex geometries such as a double slit.

2.2 Probability Density and Diagnostics

A fundamental property of the wavefunction is the probability density, defined as $\rho = |\psi|^2 = \psi\psi^*$. To analyze the behavior of the particle, we utilize the "running integral" of the probability density, $P(x,t)$:

$$P(x,t) = \int_0^x \psi(\tilde{x},t)\psi^*(\tilde{x},t)d\tilde{x} \tag{3}$$

In a closed system with normalized initial data, $P(1,t)$ must equal 1; however, even without explicit normalization, the value $P(1,t)$ must remain conserved to the level of the solution error.

To quantify scattering effects, we define the time-averaged probability distribution $\overline{P}_j$ over the total integration time. For a specific spatial interval $[x_1, x_2]$, the excess fractional probability, $\overline{F}_e$, compares the particle's residence time in that interval against that of a free particle:

$$\overline{F}_e(x_1, x_2) = \frac{\overline{P}(x_2) - \overline{P}(x_1)}{x_2 - x_1} \tag{4}$$

Here, $\overline{P}(x)$ denotes the normalized, time-averaged probability integral. A value of $\overline{F}_e < 1$ indicates the particle spends less time in the interval than a free particle would, a characteristic metric for analyzing tunneling and scattering phenomena.

3 Numerical Approach

3.1 Discretization and Crank-Nicolson Scheme (1D)

The continuum domain is discretized by introducing a discretization level l, which defines the spatial mesh spacing $\Delta x = 2^{-l}$ and the number of spatial points $n_x = 2^l + 1$. The temporal spacing is coupled to the spatial mesh via the parameter $\lambda = \Delta t/\Delta x$. Lastly, the total number of time steps is given by $n_t = \text{round}(T/\Delta t) + 1$, where T is the total simulation time.

To solve the 1D Schrödinger equation, we employ the Crank-Nicolson method, an implicit scheme that is second-order accurate in both time and space ($O(\Delta t^2, \Delta x^2)$). The discretized equation takes the form:

$$\begin{aligned} i\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = &-\frac{1}{2}\left(\frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{\Delta x^2} + \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{\Delta x^2}\right) \\ &+ \frac{1}{2}V_j^{n+\frac{1}{2}}(\psi_j^{n+1} + \psi_j^n) \end{aligned} \tag{5}$$

for $j = 2, \ldots, n_x - 1$, and $n = 0, 1, \ldots, n_t - 1$. This formula results in a complex tridiagonal system of linear equations for the advanced time step ψ^{n+1}, which is solved efficiently using sparse matrix operations. The system is first rearranged by multiplying through by Δt and introducing the constants:

$$\alpha \equiv \frac{\Delta t}{2\Delta x^2}, \qquad \mu_j \equiv \frac{\Delta t}{2}V_j^{n+\frac{1}{2}}.$$

and moving the terms involving ψ^{n+1} to the left-hand side and those involving ψ^n to the right-hand side:

$$\alpha\psi_{j-1}^{n+1} + \big(i - 2\alpha - \mu_j\big)\psi_j^{n+1} + \alpha\psi_{j+1}^{n+1} = -\alpha\psi_{j-1}^n + \big(i + 2\alpha + \mu_j\big)\psi_j^n - \alpha\psi_{j+1}^n.$$

Equivalently, we can write it as

$$a_j\psi_{j-1}^{n+1} + b_j\psi_j^{n+1} + c_j\psi_{j+1}^{n+1} = d_j, \tag{6}$$

with

$$a_j = c_j = \alpha \qquad b_j = i - 2\alpha - \mu_j,$$

and the right-hand side

$$d_j = -\alpha\psi_{j-1}^n + (i + 2\alpha + \mu_j)\psi_j^n - \alpha\psi_{j+1}^n.$$

This is the tridiagonal system where the coefficients a_j, b_j, and c_j form the sub-diagonal, main diagonal, and super-diagonal, respectively. In this form, the system can be solved by looping over time steps and using MATLAB's built-in sparse matrix capabilities.

3.2 Alternating Direction Implicit Scheme (2D)

For the two-dimensional problem, solving a large sparse system directly becomes computationally expensive. Instead, we utilize the Alternating Direction Implicit (ADI) technique with the same discretization parameters as in the 1D case. This operator-splitting method advances the solution from t^n to t^{n+1} in two half-steps.

Defining the standard second-order central difference operators ∂_{xx}^h and ∂_{yy}^h, the ADI scheme is implemented first with an implicit step in the x-direction,

$$\left(1 - i\frac{\Delta t}{2}\partial_{xx}^h\right)\psi_{i,j}^{n+\frac{1}{2}} = \left(1 + i\frac{\Delta t}{2}\partial_{xx}^h\right)\left(1 + i\frac{\Delta t}{2}\partial_{yy}^h - i\frac{\Delta t}{2}V_{i,j}\right)\psi_{i,j}^n \tag{7}$$

followed by an implicit step in the y-direction:

$$\left(1 - i\frac{\Delta t}{2}\partial_{yy}^h + i\frac{\Delta t}{2}V_{i,j}\right)\psi_{i,j}^{n+1} = \psi_{i,j}^{n+\frac{1}{2}} \tag{8}$$

where indices i and j correspond to the x and y spatial grid points, respectively. These steps utilize the central difference approximations for the second derivatives (assuming $\Delta x = \Delta y$):

$$\partial_{xx}^h\psi_{i,j} = \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x^2}, \quad \partial_{yy}^h\psi_{i,j} = \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta x^2}.$$

Defining the discretization constants similar to the Crank-Nicolson scheme:

$$\alpha \equiv \frac{i\Delta t}{2\Delta x^2}, \qquad \mu_{i,j} \equiv i\frac{\Delta t}{2}V_{i,j},$$

we can rearrange the equations to solve for the unknown wavefunctions.

To implement the first half-step, Equation (7), the code evaluates the right-hand side (RHS) efficiently by splitting the operator product into two explicit stages.

First, we evaluate the inner operator, which accounts for the explicit diffusion in the y-direction and the interaction with the potential V. This generates an intermediate array, denoted here as $\Phi_{i,j}$:

$$\Phi_{i,j} = \alpha\psi_{i,j-1}^n + (1 - 2\alpha - \mu_{i,j})\,\psi_{i,j}^n + \alpha\psi_{i,j+1}^n.$$

Next, we apply the explicit x-direction operator to this intermediate result Φ to construct the final forcing term $d_{i,j}$:

$$d_{i,j} = \alpha\Phi_{i+1,j} + (1 - 2\alpha)\,\Phi_{i,j} + \alpha\Phi_{i-1,j}.$$

With $d_{i,j}$ constructed, the code solves the implicit tridiagonal system for the updated half-step wavefunction $\psi^{n+1/2}$:

$$-\alpha\psi_{i-1,j}^{n+\frac{1}{2}} + (1+2\alpha)\psi_{i,j}^{n+\frac{1}{2}} - \alpha\psi_{i+1,j}^{n+\frac{1}{2}} = d_{i,j}. \tag{9}$$

This two-stage approach ensures that the cross-terms and potential interactions are applied in the correct order before the implicit solver is invoked.

The second half-step is similarly rearranged. Note that the potential term $\mu_{i,j}$ appears on the implicit side for this step:

$$-\alpha\psi_{i,j-1}^{n+1} + \big(1+2\alpha+\mu_{i,j}\big)\psi_{i,j}^{n+1} - \alpha\psi_{i,j+1}^{n+1} = \psi_{i,j}^{n+\frac{1}{2}}. \tag{10}$$

Each half-step involves solving n_y tridiagonal systems of size n_x and n_x systems of size n_y, respectively. This approach significantly reduces computational complexity compared to solving the full 2D system directly, while maintaining second-order accuracy in both space and time.

3.3 Convergence Testing Methodology

To verify the accuracy of the numerical implementation, we perform convergence tests across multiple discretization levels ($l = 6$ to $l = 9$). We compute the L_2 norm of the deviation between solutions at consecutive levels, denoted as $d\psi^l$:

$$||d\psi^l||_2(t^n) = ||\psi^{l+1} - \psi^l||_2(t^n) \tag{11}$$

where the finer grid solution ψ^{l+1} is coarsened to match the ψ^l grid. Since the FDA is $O(h^2)$, we expect the error to decrease by a factor of 4 as the resolution doubles. Consequently, plotting $4^{l-l_{min}}||d\psi^l||_2$ should result in coincident curves. With an exact solution, we also validate the solver by computing the exact error norm

$$||E(\psi^l)||_2(t^n) = ||\psi_{exact} - \psi^l||_2(t^n)$$

and expect similar convergence behavior with each line being scaled by $4^{l-l_{min}}$.

4 Implementation

4.1 1D Solver Function (`sch_1d_cn`)

The `sch_1d_cn.m` function is the main script for the one-dimensional simulation. It accepts the integration time `tmax`, discretization `level`, and ratio `lambda`, along with integer flags and parameter vectors for the initial data and potential types (`idtype`, `idpar`, `vtype`, `vpar`).

First, the function establishes the spatial grid resolution $n_x = 2^l + 1$ and calculates the time step $\Delta t = \lambda\Delta x$. Then, the initial wavefunction $\psi(x, 0)$ and potential $V(x)$ are generated based on the specified types and parameters. The initial wavefunction is normalized to ensure that the total probability is unity. A crucial implementation detail for the Crank-Nicolson scheme is the construction of the complex tridiagonal matrices from Section 3.1. These

matrices represent the left-hand side and right-hand side of the discretized TDSE. These are efficiently generated using MATLAB's `spdiags` command to exploit the sparse structure of the problem.

Once initialized, the function enters the main time-stepping loop. Unlike an explicit scheme, the Crank-Nicolson method requires solving a system of equations at each step. To optimize performance, the constant tridiagonal matrix is pre-factored or solved directly using MATLAB's left division operator (\). The code utilizes MATLAB's native complex number support, ensuring that the non-conjugating transpose operator (`.'`) is used to avoid accidental complex conjugation during matrix operations.

4.2 2D Solver Function (`sch_2d_adi`)

The `sch_2d_adi.m` function extends the numerical implementation to two dimensions. While the input interface remains consistent with the 1D solver, the internal logic is restructured to accommodate the Alternating Direction Implicit (ADI) method described in Section 3.2.

The implementation avoids solving a computationally prohibitive $n_x n_y \times n_x n_y$ system. Instead, the operator is split into two half-steps. Critically, the discretization constants α and μ are implemented with the explicit inclusion of the imaginary unit i (e.g., `alpha` $= 1i \cdot \Delta t/(2\Delta x^2)$).

To strictly enforce Dirichlet boundary conditions ($\psi = 0$ at all boundaries), the solver adopts an interior-point strategy. Rather than constructing a full matrix that includes boundary nodes, the linear system is reduced to solve only for indices $2, \ldots, N-1$. The boundary values are manually clamped to zero during array initialization. This prevents numerical noise from accumulating at the edges during the intermediate half-steps, ensuring long-term stability.

The tridiagonal systems for the implicit steps are constructed using MATLAB's `spdiags` function, which minimizes memory overhead by storing only the non-zero diagonals. For the x-implicit step, the left-hand side operator is independent of the potential, allowing a single constant matrix A_x to be precomputed. In contrast, the y-implicit step incorporates the potential term $\mu_{i,j}$ on the left-hand side. Consequently, the matrix A_y is reconstructed dynamically within the solver loop to account for the spatial variation of $V(x, y)$ across different grid coordinates.

A specific constraint in the 2D implementation is memory scaling. Since the function returns the full time-history of the wavefunction (an array of size $n_t \times n_x \times n_y$), memory requirements grow cubically with resolution. The script initializes these arrays efficiently, but users must be mindful of the $O(N^3)$ scaling at high discretization levels.

5 Results: 1D Simulations

5.1 1D Convergence Analysis

To validate the accuracy of the Crank-Nicolson implementation, I performed a convergence test using the exact solution family:

$$\psi(x,t) = \sin(m\pi x)e^{-im^2\pi^2 t},$$

where m is the mode integer. This solution corresponds to a free particle in a box with zero potential ($V = 0$) and Dirichlet boundary conditions. The test parameters, listed in Table 1, were chosen to check the solver in a scenario where the numerical error can be directly compared against the analytical solution.

Parameter	**Value**	**Description**
`idtype`	0	Exact family initial data
`vtype`	0	Free space ($V = 0$)
`idpar`	[3]	Mode number $m = 3$
`tmax`	0.25	Maximum integration time
`lambda`	0.1	Ratio $\Delta t/\Delta x$
l_{min}	6	Coarsest discretization level
l_{max}	9	Finest discretization level

Table 1: Parameters for the 1D convergence test (Exact Solution Case).

The convergence behavior was analyzed using two metrics: the self-convergence norm $||d\psi^l||_2$ and the exact error norm $||E(\psi^l)||_2$. As discussed in Section 3.3, since the Crank-Nicolson scheme is second-order accurate ($O(h^2)$), we expect the error to decrease by a factor of 4 as the resolution doubles.

The results are presented in Figures 1 and 2. Figure 1 illustrates the scaled self-convergence norms as a function of time. The distinct overlap of the curves (where the error at level l is scaled by $4^{l-l_{min}}$) confirms that the solver is achieving the expected second-order convergence rate.

Figure 2 displays the scaled exact error norms as a function of time. Similar to the self-convergence test, the strict alignment of these curves further validates the correctness of the discretization and the solver implementation against the theoretical model.

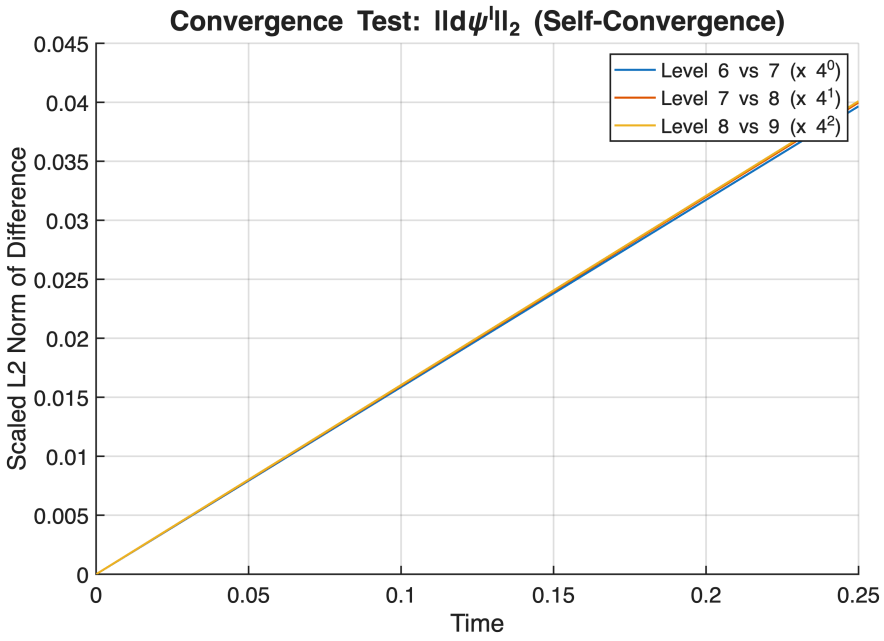

Figure 1: Scaled self-convergence norms $4^{l-6}||d\psi^l||_2$ vs. time for levels $l = 6, 7, 8$. The overlap of the curves demonstrates $O(h^2)$ convergence.

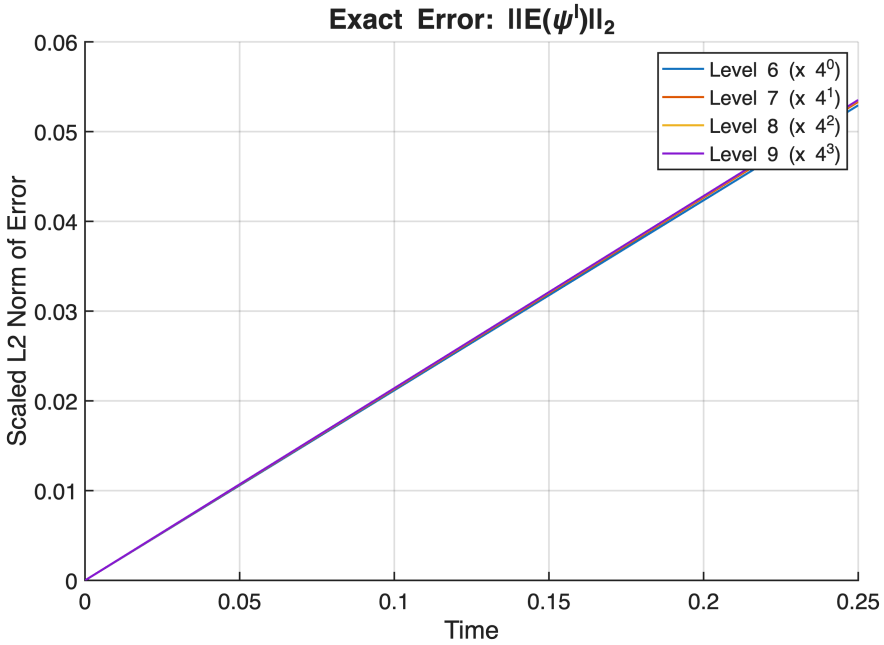

Figure 2: Scaled exact error norms $4^{l-6}||E(\psi^l)||_2$ vs. time. The agreement with the theoretical convergence rate validates the solver against the analytical solution.

5.2 Experiment 1: Barrier Survey

In this experiment, I investigated the scattering properties of a quantum particle encountering a rectangular potential barrier. The setup consists of a boosted Gaussian wave packet

initialized to the left of the barrier ($x_0 = 0.4$, $p = 20.0$). I analyzed the probability of transmission by computing the excess fractional probability from Equation (4), $\overline{F}_e$, in the region to the right of the barrier ($0.8 \leq x \leq 1.0$).

The control parameter for this survey was the barrier height, V_0. Using the script `barrier_survey`, I performed 251 simulations with $\ln(V_0)$ uniformly spaced between 2 and 5. The simulation parameters are summarized in Table 2.

Parameter	Value	Description
`tmax`	0.10	Integration time
`level`	9	Spatial discretization ($n_x = 513$)
`lambda`	0.01	Ratio $\Delta t/\Delta x$
`idtype`	1	Boosted Gaussian
`idpar`	[0.40, 0.075, 20.0]	x_0, δ, p
`vtype`	1	Rectangular Barrier
`vpar`	[0.6, 0.8, V_0]	x_{min}, x_{max}, Height
`x1, x2`	0.8, 1.0	Transmission measurement region

Table 2: Simulation parameters for the Barrier Survey experiment.

Figure 3 presents the natural logarithm of the excess fractional probability, $\ln(\overline{F}_e)$, as a function of the log-barrier height, $\ln(V_0)$.

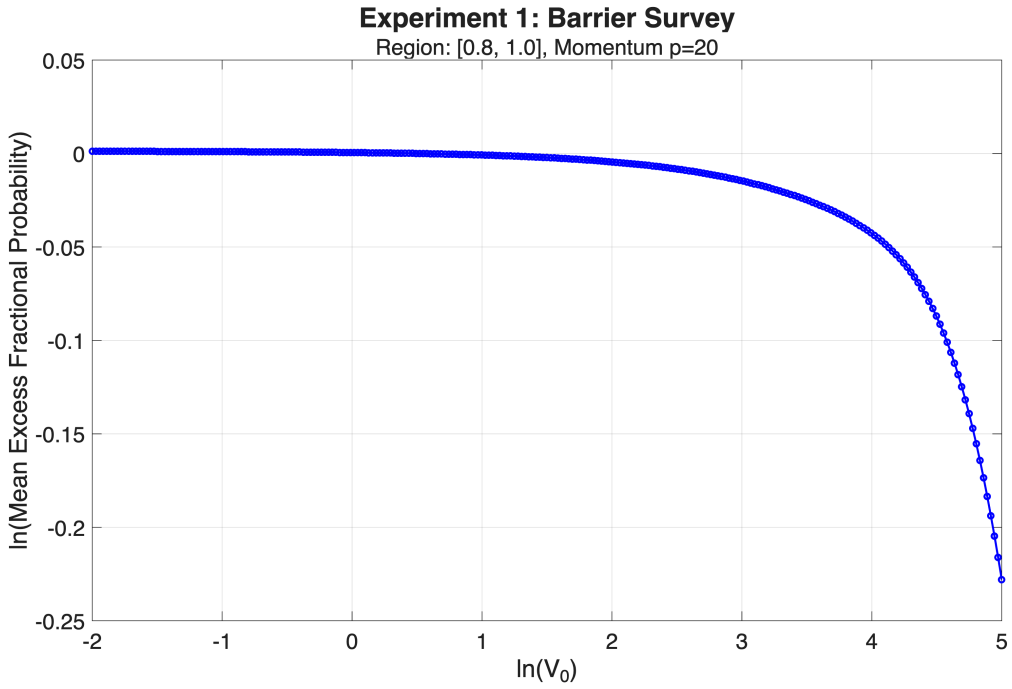

Figure 3: Log-log plot of the excess fractional probability $\overline{F}_e$ in the transmission region vs. barrier height V_0.

The experiment was conducted with a particle momentum of $p = 20$, corresponding to a non-dimensional kinetic energy of $E \approx p^2 = 400$. The barrier height V_0 varies such that

$\ln(V_0) \in [-2, 5]$, meaning V_0 ranges approximately from 0.14 to 148.

Throughout this entire range, the particle energy exceeds the barrier height ($E > V_0$). Classically, one would expect perfect transmission, which would correspond to $\ln(\overline{F}_e) \approx 0$. However, Figure 3 shows a decrease in the fractional probability as the barrier height increases.

This deviation from free-particle behavior is an effect of quantum reflection. Even though the particle has sufficient energy to surmount the barrier, the sharp discontinuity in the potential causes a portion of the wave packet to reflect. As V_0 increases (approaching the order of magnitude of E), the reflection coefficient increases, causing $\overline{F}_e$ to drop below 1 (making $\ln(\overline{F}_e)$ increasingly negative). The curve demonstrates that scattering occurs even in the absence of a classical turning point.

5.3 Experiment 2: Well Survey

In the second experiment, I examined the interaction of a spreading Gaussian wave packet with a potential well ($V_0 < 0$). Unlike the barrier survey, the particle was initialized with zero net momentum ($p = 0$). The simulation relies on the natural spreading of the wavefunction to interact with the well located at $x \in [0.6, 0.8]$.

I measured the excess fractional probability inside the well region. The control parameter was the magnitude of the well depth, $|V_0|$. Using the script `well_survey`, I performed 251 simulations with $\ln(|V_0|)$ ranging from 2 to 10. The simulation parameters are listed in Table 3. The result is shown in Figure 4 which plots the natural logarithm of the excess fractional probability against the log-magnitude of the well depth.

Parameter	**Value**	**Description**
`tmax`	0.10	Integration time
`level`	9	Spatial discretization ($n_x = 513$)
`lambda`	0.01	Ratio $\Delta t/\Delta x$
`idtype`	1	Gaussian (No boost)
`idpar`	[0.40, 0.075, 0.0]	$x_0, \delta, p = 0$
`vtype`	1	Rectangular Well
`vpar`	[0.6, 0.8, V_0]	x_{min}, x_{max}, Depth ($V_0 < 0$)
`x1, x2`	0.6, 0.8	Measurement region (Inside well)

Table 3: Simulation parameters for the Well Survey experiment .

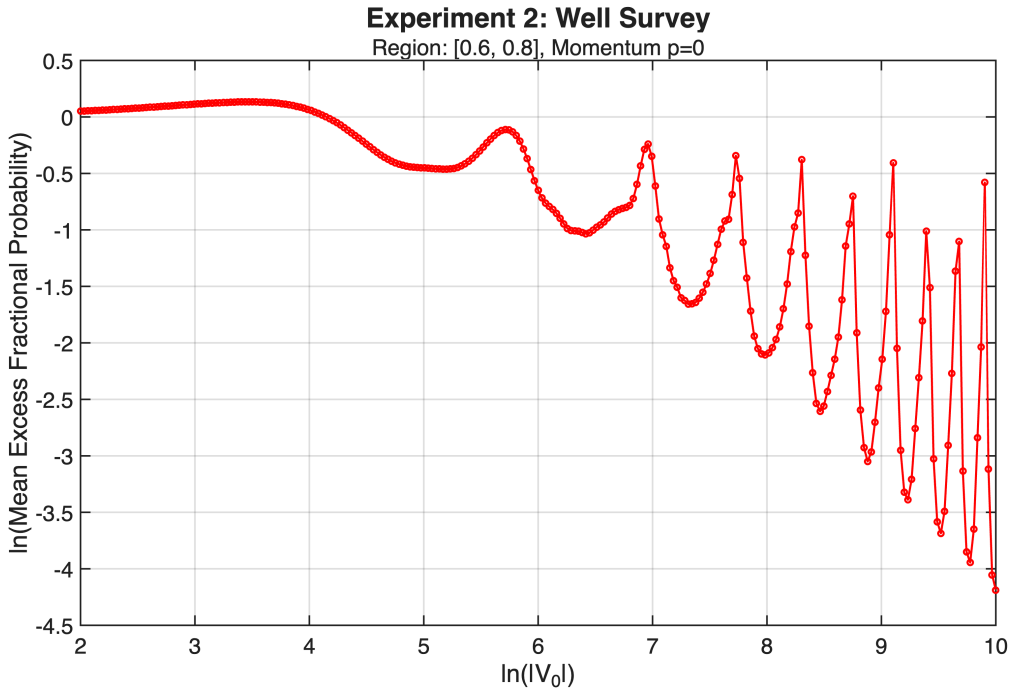

Figure 4: Log-log plot of excess fractional probability $\overline{F}_e$ inside the well vs. well depth $|V_0|$. Values less than 0 indicate the particle spends less time in the well than a free particle.

The results in Figure 4 display two distinct physical behaviors: a general downward trend and strong oscillations.

First, the overall decay of $\ln(\overline{F}_e)$ confirms the classical expectation of particle acceleration. Inside the well, the potential is negative ($V_0 < 0$), increasing the particle's kinetic energy ($K = E - V_0$). As the well deepens ($\ln(|V_0|)$ increases), the particle moves significantly faster within the region $[0.6, 0.8]$ than it would in free space, reducing its residence time and thus the integrated probability density.

Second, and more importantly, the plot exhibits prominent oscillations that grow in amplitude as the well deepens. These are a because of quantum interference and scattering resonances. The rectangular well creates two sharp impedance mismatches at $x = 0.6$ and $x = 0.8$. As the wave packet spreads into the well, it undergoes multiple internal reflections.

The wavelength of the particle inside the well depends on the depth ($\lambda \propto 1/\sqrt{|V_0|}$). As $|V_0|$ varies, the ratio of the well width to the particle's wavelength changes, cycling through conditions for constructive and destructive interference. The peaks in the graph correspond to resonant depths where the wave "traps" or builds up constructively inside the well, while the troughs correspond to destructive interference where the probability density is suppressed.

6 Results: 2D Simulations

6.1 2D Convergence Analysis

To verify the ADI solver, I performed a convergence test using the two-dimensional exact solution family defined in Eq. 12:

$$\psi(x, y, t) = e^{-i(m_x^2+m_y^2)\pi^2 t} \sin(m_x \pi x) \sin(m_y \pi y) \tag{12}$$

For this test, I utilized mode numbers $m_x = 2$ and $m_y = 3$. The ADI scheme is also second-order accurate in both space and time ($O(h^2)$). Therefore, we expect the error norms to diminish by a factor of 4 for each level increment. The test parameters are summarized in Table 4.

Parameter	Value	Description
`idtype`	0	Exact family initial data
`vtype`	0	Free space ($V = 0$)
`idpar`	[2, 3]	Modes $m_x = 2, m_y = 3$
`tmax`	0.05	Integration time
`lambda`	0.05	Ratio $\Delta t/\Delta x$
l_{min}	6	Coarsest discretization level
l_{max}	9	Finest discretization level

Table 4: Parameters for the 2D convergence test.

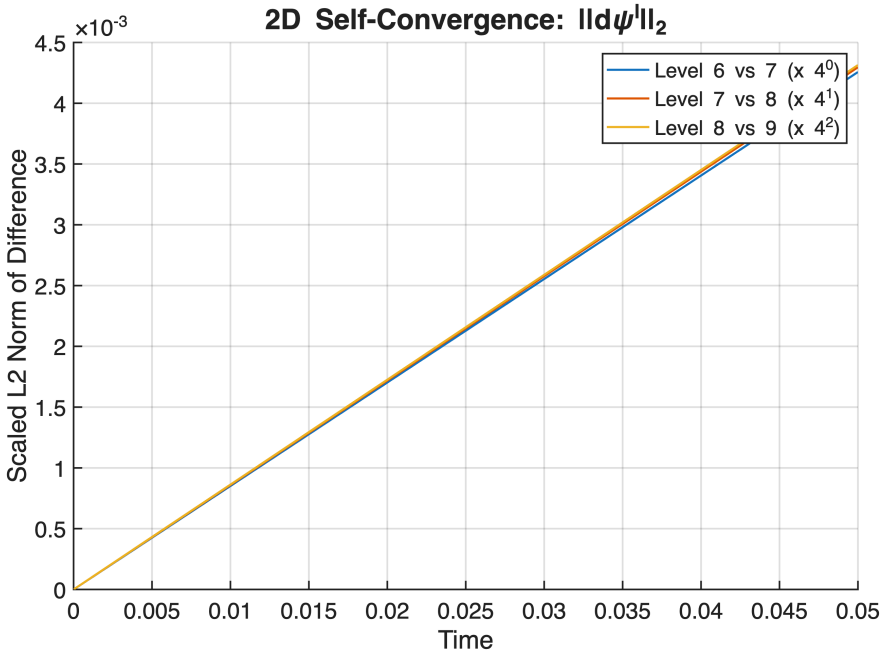

Figure 5: Scaled self-convergence norms $4^{l-6}||d\psi^l||_2$ vs. time for the 2D ADI solver. The overlap confirms $O(h^2)$ accuracy.

The results of the convergence analysis are shown in Figures 5 and 6. Figure 5 illustrates the scaled self-convergence norms, $4^{l-l_{min}}||d\psi^l||_2$. Note that for the 2D case, the L_2 norm involves a summation over the full $n_x \times n_y$ grid. The coincident curves confirm that the ADI splitting error and the spatial discretization errors are both converging at the expected second-order rate.

Figure 6 compares the numerical solution directly against the analytical function. The alignment of the scaled error norms $||E(\psi^l)||_2$ provides strict validation of the 2D solver implementation.

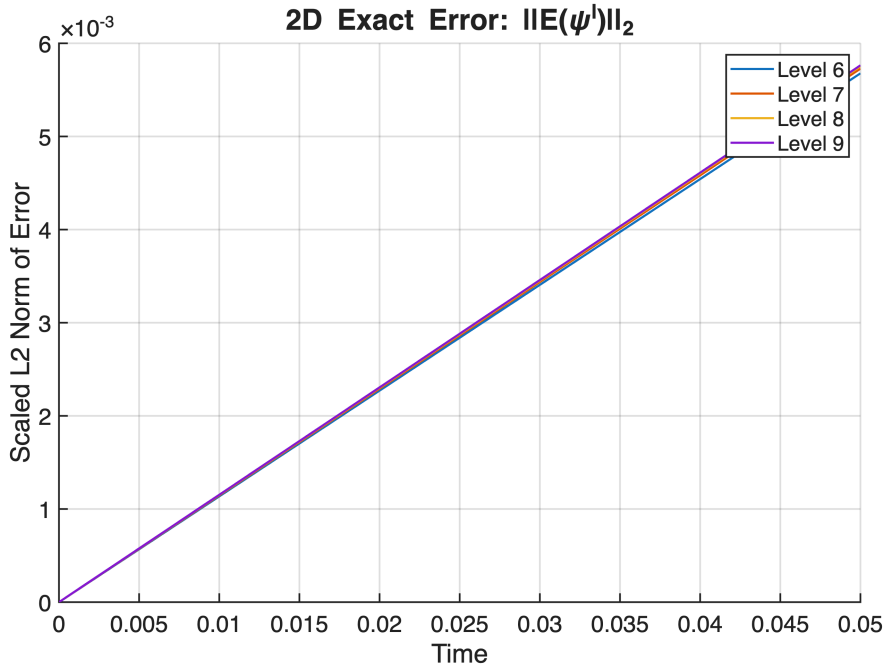

Figure 6: Scaled exact error norms $4^{l-6}||E(\psi^l)||_2$ vs. time for the 2D ADI solver.

6.2 Scattering Experiments

To visualize the dynamics of the two-dimensional Schrödinger equation, I conducted a series of numerical experiments simulating a Gaussian wave packet interacting with various potential landscapes. As suggested in the project handout, these results were visualized using filled contour plots of the probability density $|\psi|^2$ to observe time-evolution phenomena such as interference and diffraction. The solver does a good job showing these effects until the wavefunction reaches the boundaries, at which point unphysical reflections occur due to the Dirichlet boundary conditions.

6.2.1 Rectangular Barrier

In the first scenario, a Gaussian wave packet was directed toward a rectangular potential barrier of height $V_0 > E$, where E is the kinetic energy of the incoming particle. The wave packet was initialized with a boost in the x-direction to ensure it would collide with the barrier. The simulation captures the splitting of the wavefunction into transmitted and

reflected components. Upon impact, a small portion of the wave packet with energy $E < V_0$ tunnels through the barrier, while most of the packet reflects. The parameters for this simulation are listed in Table 5.

Parameter	Value	Description
`tmax`	0.015	Integration time
`level`	9	Spatial discretization
`idtype`	1	Boosted Gaussian
`idpar`	[0.25, 0.5, 0.05, 0.05, 50, 0]	$x_0, y_0, \delta_x, \delta_y, p_x, p_y$
`vtype`	1	Rectangular Barrier
`vpar`	[0.5, 0.55, 0.0, 1.0, 4000]	$x_{min}, x_{max}, y_{min}, y_{max}, V_c$

Table 5: Parameters for the 2D Rectangular Barrier experiment.

Figure 7 displays three snapshots of the interaction. The interference between the incoming tail and the reflected head of the wave packet creates a standing wave pattern (ripples) in the probability density on the incident side of the barrier. In the last snapshot, you can see the wavepacket hitting the boundary and reflecting back into the domain, which is an unphysical artifact of the Dirichlet boundary conditions.

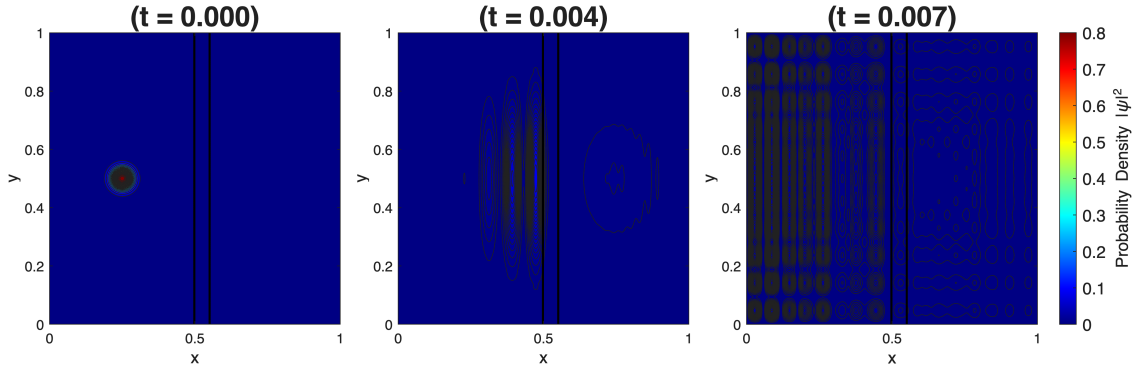

Figure 7: Time evolution of the probability density $|\psi|^2$ scattering off a rectangular barrier. Snapshots taken at $t = 0$, $t = 0.004$, and $t = 0.007$.

6.2.2 Rectangular Well

Next, I simulated scattering off a rectangular potential well ($V_0 < 0$). In this case, the classical expectation is that the particle accelerates as it traverses the well. Quantum mechanically, this manifests as a shortening of the wavelength inside the potential region. The setup parameters are provided in Table 6.

As seen in Figure 8, the impedance mismatch at the well boundaries causes partial reflections at both the entry and exit interfaces. These internal reflections lead to transient trapping effects within the well before the packet eventually transmits through.

Parameter	Value	Description
`tmax`	0.015	Integration time
`level`	9	Spatial discretization
`idtype`	1	Boosted Gaussian
`idpar`	[0.25, 0.5, 0.05, 0.05, 20, 0]	$x_0, y_0, \delta_x, \delta_y, p_x, p_y$
`vtype`	1	Rectangular Well
`vpar`	[0.5, 0.8, 0.2, 0.8, -10000]	$x_{min}, x_{max}, y_{min}, y_{max}, V_c$

Table 6: Parameters for the 2D Rectangular Well experiment.

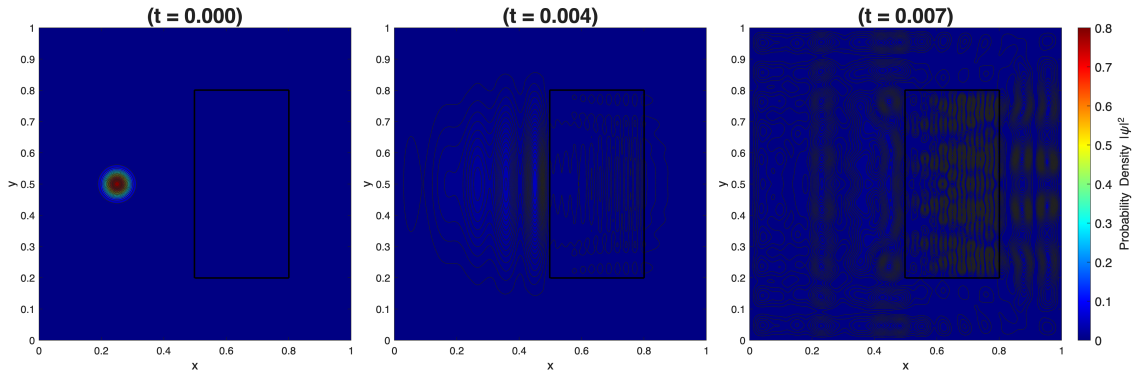

Figure 8: Snapshots of the wave packet interacting with a potential well. Note the change in wavelength and probability density inside the well region.

6.2.3 Double Slit

To demonstrate quantum interference, I implemented a double-slit apparatus. The potential was defined as a thin wall at a fixed y-index with two narrow apertures. As the Gaussian wave packet strikes the wall, the majority is reflected, but a coherent wavefront emerges from the slits.

Parameter	Value	Description
`tmax`	0.01	Integration time
`level`	9	Spatial discretization
`idtype`	1	Boosted Gaussian
`idpar`	[0.5, 015, 0.05, 0.05, 0, 20]	$x_0, y_0, \delta_x, \delta_y, p_x, p_y$
`vtype`	2	Double Slit
`vpar`	[0.3, 0.35, 0.65, 0.7, 1000]	x_1, x_2, x_3, x_4, V_c

Table 7: Parameters for the 2D Double Slit experiment.

On the far side of the barrier (Figure 9), these cylindrical waves expand and overlap. The resulting probability density displays characteristic interference fringes confirming the wave nature of the particle.

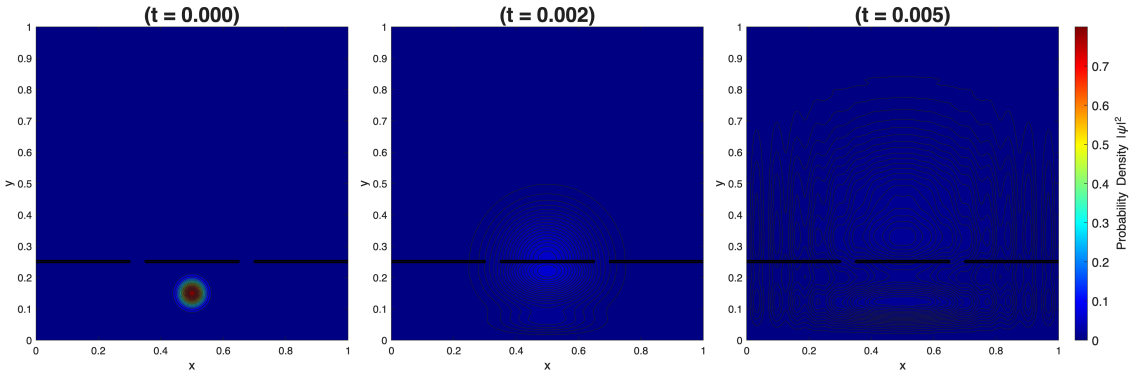

Figure 9: Diffraction through a double slit. The emerging wavefronts interfere to form a fringe pattern on the far side of the barrier.

6.2.4 Square Pillar

Finally, we extended the investigation to a custom geometry: scattering off a localized "pillar" potential. This setup models a finite obstacle placed directly in the path of the wave packet. Unlike the wall, the wave packet can flow around the object.

Parameter	Value	Description
`tmax`	0.008	Integration time
`level`	9	Spatial discretization
`idtype`	1	Boosted Gaussian
`idpar`	[0.2, 0.5, 0.05, 0.05, 50, 0]	$x_0, y_0, \delta_x, \delta_y, p_x, p_y$
`vtype`	1	Square Pillar
`vpar`	[0.5, 0.55, 0.475, 0.525, 1000]	$x_{min}, x_{max}, y_{min}, y_{max}, V_c$

Table 8: Parameters for the 2D Pillar experiment.

Figure 10 reveals the diffraction of the wavefunction around the edges of the pillar. Behind the obstacle, the wavefronts that wrap around the left and right sides converge, creating a complex interference wake.

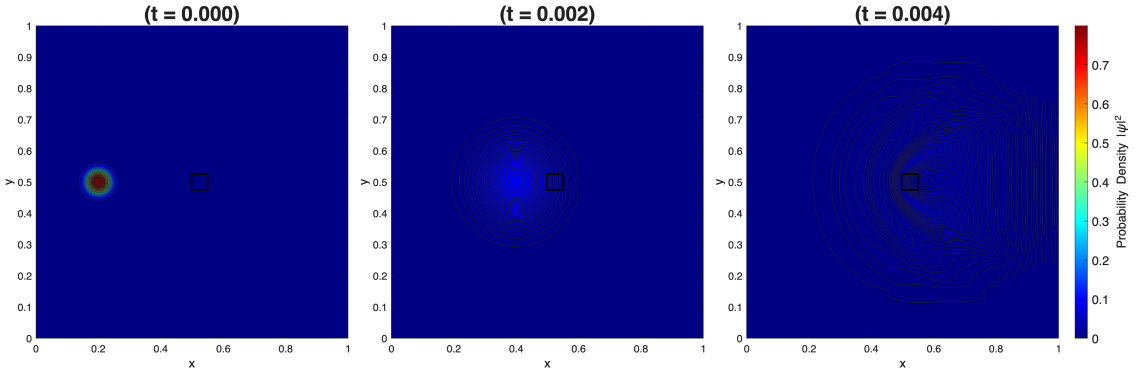

Figure 10: Scattering off a square pillar potential. The wavefunction diffracts around the obstacle, creating an interference wake downstream.

7 Conclusion

This project successfully implemented and verified numerical solvers for the TDSE in both one and two dimensions. By utilizing the Crank-Nicolson method for the 1D case and the ADI scheme for the 2D case, the simulations achieved the expected unconditional stability and $O(h^2)$ convergence rates. The numerical experiments accurately reproduced fundamental quantum phenomena, including wave packet spreading, quantum reflection above potential barriers, and the emergence of scattering resonances within potential wells.

A challenge encountered in this project was implementing the two step ADI method and solving the resulting tridiagonal systems efficiently. Remembering that the imaginary unit was correctly incorporated into the discretization constants brought some difficulty. That simple mistake led to unstable behavior. Additionally, interpreting the "excess fractional probability" metric initially proved counter-intuitive, particularly in the 1D barrier survey where the potential was infact lower than the particle energy since energy, $E \approx p^2$, was greater than the barrier height, V_0. This led to confusion when analyzing the results.

7.1 Use of AI

I used ChatGPT and Gemini as a guide to help me understand the numerical schemes and clarify the underlying quantum mechanics. All code was written by me. AI was primarily used for formatting tables in my LaTeX report and generating consistent MATLAB plots to ensure a uniform visual style. ChatGPT was also helpful in debugging my ADI solver, specifically identifying a missing imaginary unit i in the discretization constants which had me stuck for hours. It also assisted in making the figures for the 2D experiments after generating the movies.

8 Files

The following MATLAB files were created for this assignment:

- **Solvers:** `sch_1d_cn.m`, `sch_2d_adi.m`
- **Problem 1:** `ctest_1d.m`, `barrier_survey.m`, `well_survey.m`
- **Problem 2:** `ctest_2d.m`, `movies.m`