The Time Dependent Schroödinger Equation

Guillermo Cabrera

December 2023

Introduction

This project aims to observe phenomena governed by the time-dependent Schrödinger equation in 1D and 2D scenarios, by way of developing numerical computer simulations. Exploring quantum phenomena like tunneling, scattering, and fluctuations, it delves into numerical methods and computational implementations for understanding particle behavior within potential landscapes.

Theoretical foundations are discussed, presenting the Schrödinger equation's continuum form and its specific parameters for this problem. Numerical approaches, including the Crank-Nicolson and ADI scheme, are detailed for efficient solving of the equation. Challenges in implementing spatially varying potentials and computing probability integrals are addressed.

Results cover convergence tests, confirming the expected accuracy of the numerical methods. Numerical experiments showcase wavefunction behaviors in scenarios like barrier scattering, well dispersion, and the double-slit experiment, revealing quantum phenomena influenced by potential shapes.

The project offers valuable computational strategies for comprehending quantum mechanics in diverse potential landscapes.

The 1D Shrödinger Equation

Review of Theory

First of all, let's take a look at the mathematics and theory behind this project.
The non-dimensional continuum Schrödinger equation used is as follows,

$$i\Psi_t(x,t) = \Psi_{xx} + V(x,t)\Psi,$$

Where $\Psi(x,t)$ is complex, and solved within the following domain,

$$0 \le x \le 1, \qquad 0 \le t \le t_{max},$$

subject to initial and boundary conditions

$$\Psi(x,0) = \Psi_0(x)$$

$$\Psi(0,t) = \Psi(1,t) = 0.$$

The initial data function $\Psi_0(x)$ in general can be complex, but only real cases are considered in this project. The time-independent potential is used in order to make the analysis more clean, and eliminate practical complexities.

Note that Ψ can be expressed in terms of its real and imaginary parts,

$$\Psi = \Psi_{Re} + \Psi_{Im}$$

Using the probability density $P(x,t)$ as a diagnostic quantity, $\rho = |\Psi|^2 = \Psi\Psi^*$

$$P(x,t) = \int_0^x |\Psi|^2 dx.$$

A family of exact solutions for this Schrödinger equation is

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

where m is a positive integer.

Numerical Approach

The continuum was discretized by introducing a level parameter l and a ratio λ between the temporal and spatial grid spacings.

$$\lambda = \frac{\Delta t}{\Delta x}$$

so,

$$nx = 2^l + 1$$

$$\Delta x = 2^{-l}$$

$$\Delta t = \lambda \Delta x$$

$$nt = round(t_{max}/\Delta t) + 1$$

Then, the Crank-Nicolson discretization approach was applied, this method is unconditionally stable and second order accurate in both space and time, combining implicit and explicit approaches to provide both stability and accuracy. Applying this method to this shröndinger equation yields the following,

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

$$n = 1, 2, ...nt - 1$$

$$j = 1, 2, ...nx - 1$$

Now, in order to apply this approach effectively, method of lines (MOL) was used to turn the problem into a linear system, this involved doing some algebra to find the coefficients of ψ for each combination of temporal and spatial step components. This resulted in a linear system with sparse matrices A and B,

$$A\psi_j^{n+1} = B\psi_j^n$$

Where the right hand side is known and solved explicitly, and ψ_j^{n+1} is solved for implicitly.

$$\psi_j^{n+1} = A^{-1}(B\psi_j^n)$$

The capitalizing of the sparsity structure of A is what allows for efficiency in computation, specially when dealing with smaller grid space, bigger time-spans and thus large matrices.

Implementation

Implementation of the 1D shrödinger equation was reasonably straightforward at first, the set up of the initial settings for the equation was simple and required the some use of some for loops and conditional statements.

Something tricky was understanding how to implement the potential term in the sparse matrices, considering that it is space dependant, I had to figure out a way to have it change spatial iteration, and the way I did that was by using a for loop to to change its value for each row on the sparse matrix, as shown bellow I would have a different term on the main diagonal "d" for each row on the matrix.

```
  for j = 1 : nx
    d(j, 1) = (1i*dt^-1 - dx^-2 - (v(j))/2);
   end
```

Another thing that was tricky was calculating the discrete running integral of probability, the goal was to apply the trapezoidal rule to compute the probability density at each j value, for each time step. At first I didn't understand how to go about this, but then I realized that I was the sum of the probability at each grid space, through the time evolution, that when properly normalized would equal to one for the spatial grid end point.

```
% Calculating probability using trapesoidal formula (O(h^2))
prob = zeros(nt, nx);
for n = 1 : nt
    for j = 1 : nx - 1
        prob(n, j+1) = prob(n, j) + (1/2)*(psisquared(n, j) + psisquared(n, j + 1))*(x(j+1) - x(j));
    end
end
```

Results

Convergence Tests:

After implementing the 1D Shrödinger equation, I performed a series of convergences tests in order to verify that my method and implementation worked as expected.

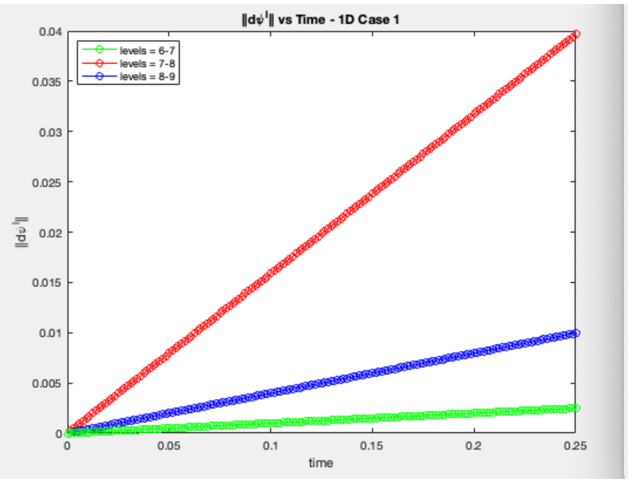
Case 1: Convergence test results given "Exact Family" initial condition

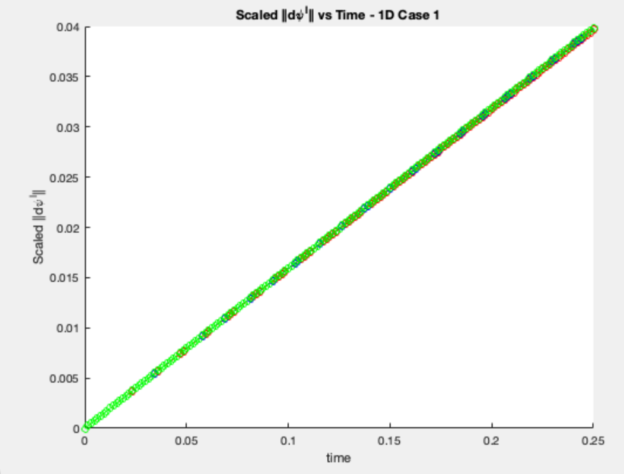
$$\psi(x,0) = sin(m\pi x), \qquad m = 3$$

This first test involves using the solution differences of subsequent level parameters denoted by $d\psi^l$, then plotting its RMS value against time (done for all tests).

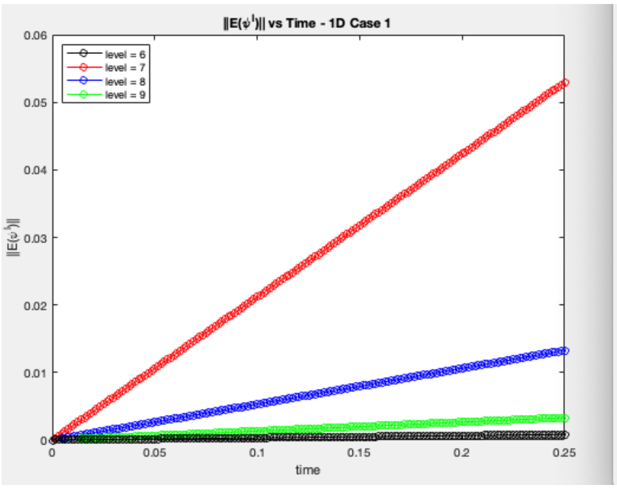
$$d\psi^l = \psi^{l+1} - \psi^l$$

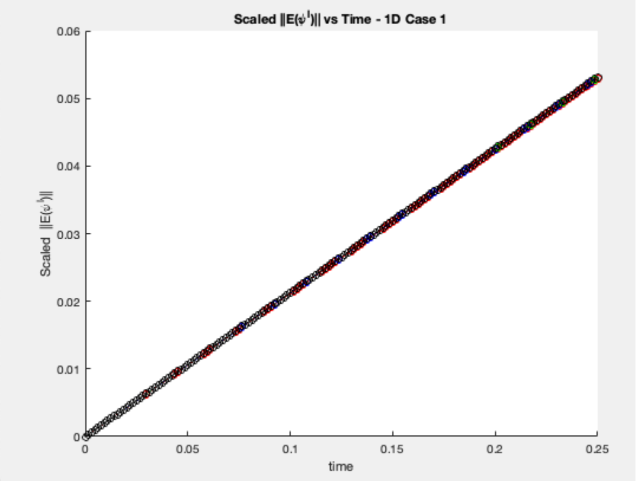
In order to demonstrate the expected $O(h^2)$ order accuracy for this method, the calculations were scaled re-scaled accordingly.





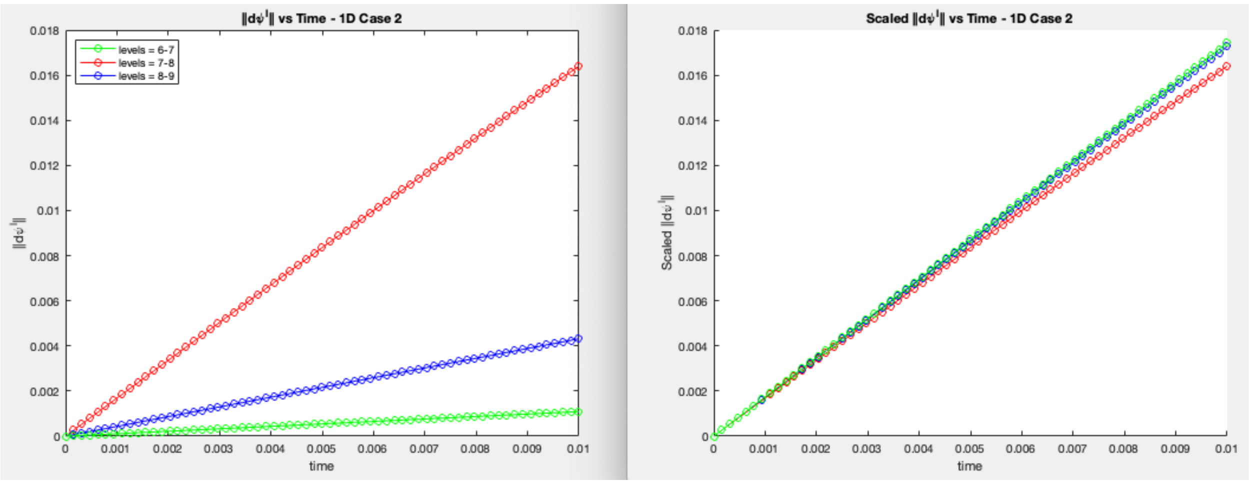
Since we already know the exact solution, we can compute the actual solution errors, so we do same procedure done above but replace ψ^{l+1} with ψ_{exact}, denoting the error as $E(\psi^l)$





Case 2: Initial condition is a "Boosted Gaussian"

$$\psi(x,0) = e^{ipx}e^{-((x-x_0)/\delta)^2}, \qquad x_0 = 0.5, \quad \delta = 0.075, \quad p = 0$$



Numerical Experiments

Two experiments were performed to gain some insights about the solutions behaviour given specific situations, as well as further verifying that the implementation worked correctly.

In these experiments, the reasonable assumption was made that

$$\overline{P}(x_2) - \overline{P}(x_1) = x_2 - x_1, \qquad x_2 > x_1$$

for a free particle ($V = 0$ and sufficiently long times). Where $\overline{P}$ is the normalized temporal average of the probability density (due to solving the Shrödinger equation on the unit interval).

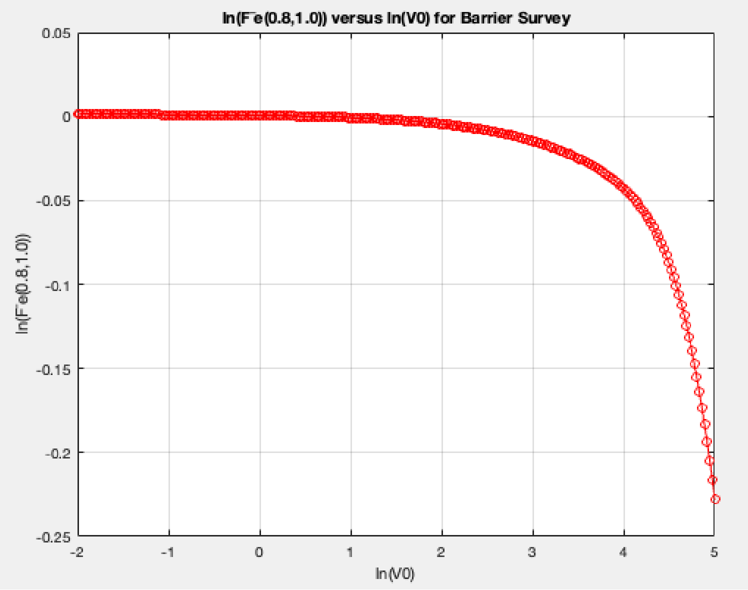
So for the general case of non-zero potential, we can define the excess (or ratio) in fractional probability that the particle spends in a given interval relative to a free particle as

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

Since the solution for the experiments result spanned orders of magnitude, this quantities natural logarithm was calculated in the results.

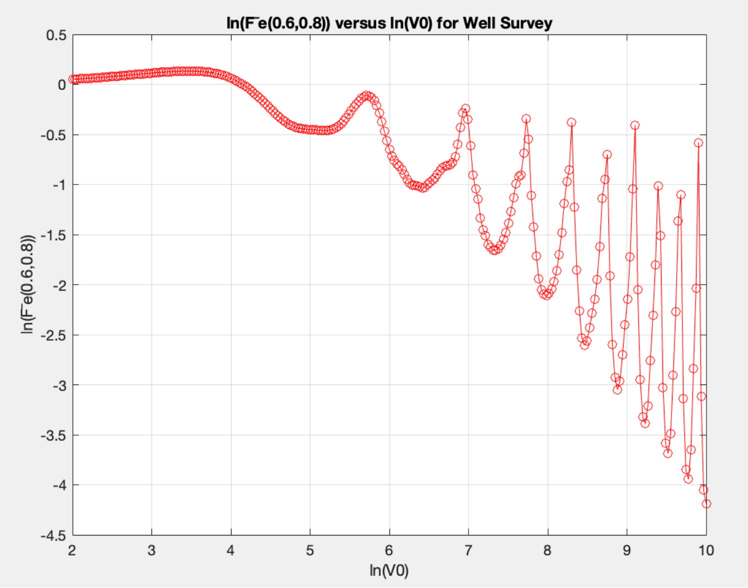
The Barrier Survey Experiment involves investigating a particle that starts on the left hand side of a potential barrier, and determining the relationship between $ln(\overline{F}_e(x_1, x_2))$ and $ln(V_0)$ where V_0 is the height of the potential (the control parameter) while x_1 and x_2 is an interval to the right of the barrier.

The following graph demonstrates the results of this experiment,



As we can see, as the potential barrier's height increases, the particle spends less and less time than I free particle would on the right side of the barrier, this makes sense because the particle is less and less likely to have penetrate, higher and higher energy barriers, as one would expect. But we can also observe that significant decay occurs after the potential has already increased substantially, this slow initial decay points to the phenomenon of quantum tunneling, and that as the barrier height increases, the likelihood of tunneling decreases (slowly for low potentials).

The Well Survey Experiment considers a particle scattering off a potential well, and again we analyze the relationship of $ln(\overline{F}_e(x_1, x_2))$ and $ln(V_0)$, where x_1 and x_2 span the width of the barrier.



As seen above, the results again demonstrate that as the potential increases (becomes more negative), the particle spends less and less time than a free particle would within the well. We can conclude that perhaps this is to do with the tunneling effect, which decreases as V_0 increases. The fluctuations we see may be due to may be simply do to the probabilistic nature of quantum mechanics, leading to variations in the likelihood of tunneling at different instances.

The 2D Shrödinger Equation

Review of Theory and Numerical Approach

Most of the theory discussed above applies in this case, as we are simply adding a another spatial dimension, so the focus of this section will be on the main differences and challenges that come from dealing with a 2D problem.

When multiplying by $-i$ the non-denationalized continuum equation is now

$$\psi_t = i(\psi_{xx} + \psi_{yy}) - iV(x,y)\psi$$

Solved in the domain,

$$0 \leq x \leq 1, \qquad ,0 \leq y \leq 1, \qquad 0 \leq t \leq t_{max},$$

subject to initial and boundary conditions

$$\Psi(x,y,0) = \Psi_0(x,y)$$
$$\Psi(0,y,t) = \Psi(1,y,t) = \Psi(x,0,t) = \Psi(x,1,t) = 0.$$

A family of exact solutions is given by,

$$\psi(x,y,t) = e^{i(m_1+m_2)^2\pi^2 t} sin(m_x \pi x) sin(m_y \pi y)$$

where m_x and m_y are positive integers.

Discretization and ADI Scheme

We let the temporal to spatial mesh spacings to be equal,

$$\lambda = \frac{\Delta t}{\Delta x} = \frac{\Delta t}{\Delta y}$$

The Alternating Direction Implicit (ADI) method is applied to the continuum Shrödinger equation shown above, the ADI method essentially alternately treats each spatial direction implicitly while discretizing the other directions explicitly. The main benefit of this method in contrast with the Crank Nicolson scheme is that it allows for tridiagonal matrix structures instead of pentadiagonal structures, this reduces the computational burden significantly. This method is also unconditionally stable and second order accurate in space and time.

Defining the difference operators ∂^h_{xx} and ∂^h_{yy} by

$$\partial^h_{xx} u^n_{i,j} = \frac{u^n_{i+1,j} - 2u^n_{i,j} u^n_{i-1,j}}{\Delta x^2}$$

$$\partial^h_{yy} u^n_{i,j} = \frac{u^n_{i,j+1} - 2u^n_{i,j} u^n_{i,j-1}}{\Delta y^2}.$$

The ADI discretization results in two discrete equations is as follows,

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} = \left(1 + i\frac{\Delta t}{2}\partial^h_{xx}\right)\left(1 + i\frac{\Delta t}{2}\partial^h_{yy} - i\frac{\Delta t}{2}V_{i,j}\right)\psi^n_{i,j}$$

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{yy} + i\frac{\Delta t}{2}V_{i,j}\right)\psi^{n+1}_{i,j} = \psi^{n+\frac{1}{2}}_{i,j}$$

$$i = 2, 3, ..., nx - 1, \qquad , j = 2, 3, ..., ny - 1, \qquad , n = 2, 3, ..., nt - 1$$

For this case, because there are two equations to be solved, we have an intermediate term $\psi^{n+\frac{1}{2}}_{i,j}$ that will be used to solve for $\psi^{n+1}_{i,j}$. Again using MOL we can create a linear,

$$A\psi^{n+\frac{1}{2}}_{i,j} = BC\psi^n_{i,j}$$

$$D\psi^{n+1}_{i,j} = \psi^{n+\frac{1}{2}}_{i,j}$$

where A, B, C, and D are tridiagonal matrices, and again capitalizing on the sparsity structure we get,

$$\psi^{n+1}_{i,j} = D^{-1}(A^{-1}(BC\psi^n_{i,j}))$$

Implementation

I tricky part of implementation was once again figuring out how the potential would be implemented into the problem, for the 1D case, the potential was indexed for each column, that lead me to think that for the 2D case I would have to create multiple matrices in order to implement the (nx x ny) grid of potentials (the potential landscape).

Initially I implemented this by defining a 3D array that stored each tridiagonal matrix, but this proved ineffective as it was the main issue that caused my code take a very long time to run (over 30 minutes for level 9 mesh). By debugging I found that issue with using that approach is that sparsity structure structure is lost when creating the 3D array, so I fixed this by instead creating a cell of sparse matrices that varied in one dimension of the potential.

```
% Initializing an empty cell array to store sparse matrices
Cs = cell(1, nx);
dlC = (1i*dt*(dy^-2)/2) * ones(nx, 1);
duC = dlC;
dlC(nx-1) = 0;
duC(2) = 0;
```

```
for i = 1 : nx
   dC = (1 - 1i*dt*dy^(-2) - 1i*dt*(v(i,:).')/2);
   dC(1) = 1;
   dC(nx) = 1;
   Cs{i} = spdiags([dlC dC duC], [-1 0 1], nx, nx);
end
```

Another part I found challenging was understanding how I would actually go about computing the solution, at first I was confused as to how each sparse matrix described above would act on ψ at time t, another matrix, then finally, I realized that each sparse matrix is meant to a specific spatial component independently. So to implemented this, I used for loops to get each sparse matrix act on its appropriate component, to do this made an intermediate term X since matrices B and C acted on different components.

```
% Computing solution using ADI Scheme
for n = 1 : nt - 1
    for i = 1 : nx
        X(i, :) = reshape(Cs{i} * squeeze(psi(n, i, :)), 1, []);
    end
    for j = 1 : ny
        f(:, j) = B*X(:, j);
    end
    % Boundary Conditions
    f(1,:) = 0;
    f(:, 1) = 0;
    f(nx, :) = 0;
    f(:, ny) = 0;
    for j = 1 : ny
        psi_nhalf(:, j) = A \ f(:, j);
    end
    for i = 1 : nx
        psi(n+1, i, :) = Ds{i} \ psi_nhalf(i, :).';
    end
end
```

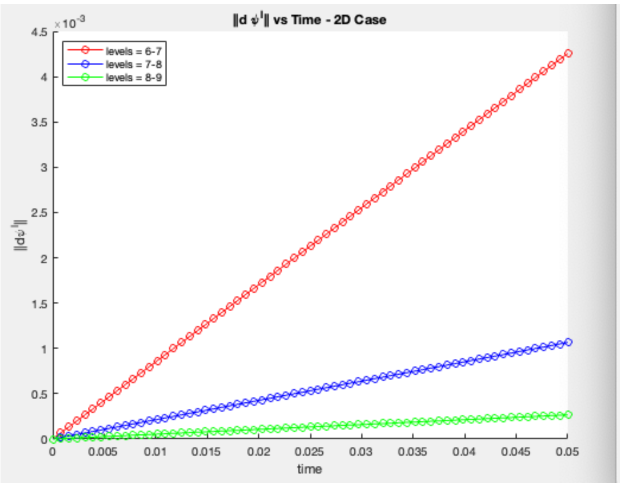
Results

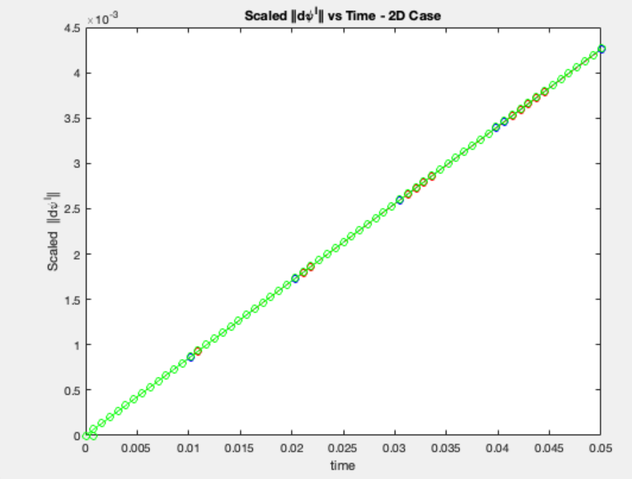
Convergence Tests

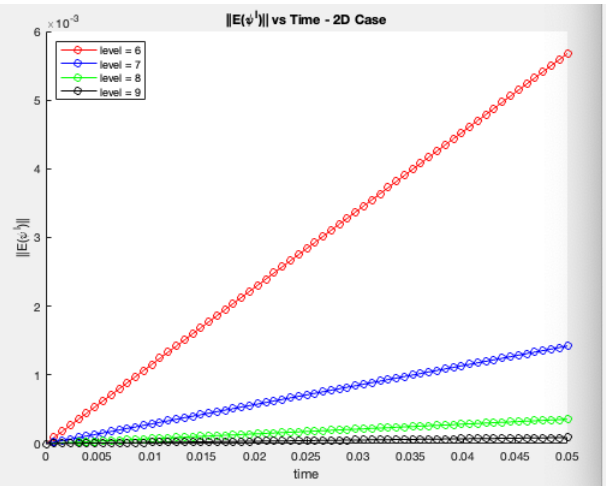
Given ”Exact Family” initial conditions,

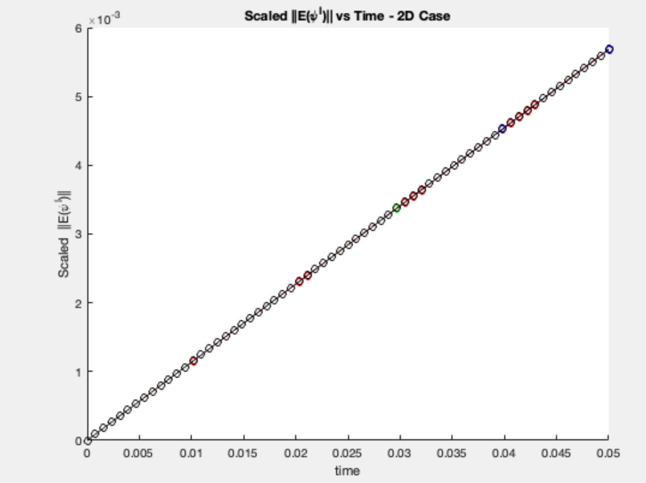
$$\psi(x, y, 0) = sin(m_1 \pi x) sin(m_2 \pi y), \qquad m_1 = 2, \quad m_2 = 3$$

The results bellow demonstrate expected second order accuracy of the ADI method.







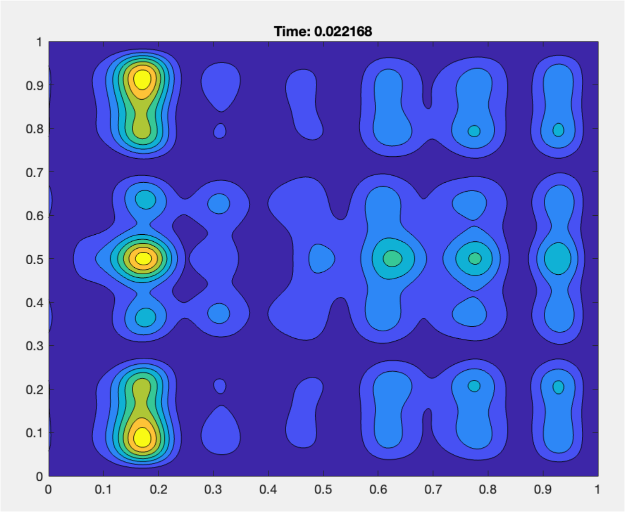


Experiments

To further test the implementation, and to observe the equations behaviour, as series of scenarios were generated. All of the scenarios bellow use the ”Boosted Gaussian” as the initial condition, and the contour plot of the probability density $|\psi(x, y)|^2$ is what is being plotted against time.

1. Scattering off a rectangular barrier.
2. Scattering off a rectangular well.
3. Scattering through a double slit.

In case 1, it is observed how for a tall barrier, no part of the wave function visibly penetrates it. But for a short barrier, part of it tunnels through as expected. A very similar behaviour is seen for case 2. For case 3, the famous experiment is indeed portrayed as one can see the diffraction pattern formation after the wave function scatters through the double slits (as shown in the figure bellow).



I also explored what would happen for different potential shapes, this may be useful in understanding the behaviour of a wavefunction given a specific energy landscape of the particle one may be working. The AVI movies are included with the project for reference.

4. Dispersion within a deep circular well.
5. Dispersion within a shallow circular well.
6. Dispersion from centre of Gaussian barrier.
7. Dispersion from centre of deep Gaussian well.
8. Dispersion from centre of shallow Gaussian well.
9. Scattering through a double slit though two Double Slit Barriers (extra).

Overall Conclusion

In exploring the time-dependent Schrödinger equation in both 1D and 2D scenarios, this project has uncovered interesting observations of quantum phenomena. Theoretical foundations, numerical methodologies, and computational implementations have provided a comprehensive understanding of particle behavior within potential landscapes.

The rigorous numerical approaches, such as the Crank-Nicolson and ADI methods, showcased stability and accuracy, confirmed through convergence tests. The experimental analyses revealed intricate behaviors like tunneling, scattering, and interference, emphasizing the significant influence of potential shapes on wavefunction propagation.

Despite implementation challenges in handling spatially varying potentials and computing probability integrals, the project successfully portrayed wavefunction dynamics in diverse scenarios, shedding light on fundamental quantum phenomena.

This exploration serves as a step in better understanding quantum mechanics, deploying valuable computational and numerical strategies and insights along the way.

The help of AI was used to learn more about certain topics, as well as for guidance in some aspects of the code, for example, I searched up how I would go about coding the meshgrid function to create a 3D array, AI helped in demonstrating how the function worked as well as all of its inputs and outputs.