Time Dependent Schrodinger Equation

Adrian Wu

1 Introduction

In the realm of quantum mechanics, the Schrodinger equation stands as a cornerstone, providing a mathematical framework to describe the behavior of particles at the smallest scales. This project delves into the task of solving the one-dimensional (1D) and two-dimensional (2D) time-dependent Schrodinger equations using the Crank-Nicolson approach and the ADI technique. After solving them, we will perform convergence testing on them. Additionally, we will try to make animations of the 2D solution as it interacts with different potentials, similar to those that we study.

2 One-dimensional time dependent Schrodinger equation

2.1 Review of Theory and Numerical Approach

The continuum equation after non-dimensionalization is

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

where $\psi(x,t)$ is the wavefunction and is complex. We can rewrite the wavefunction in terms of its real and imaginary parts:

$$\psi = \psi_{Re} + i\psi_{Im}$$

The domain we are interested in is

$$0 \le x \le 1$$

$$0 \le t \le t_{max}$$

with initial conditions

$$\psi(x,0) = \psi_0(x)$$

and boundary conditions

$$\psi(0,t) = \psi(1,t) = 0$$

We can define a useful diagnostic quantity, $P(x,t)$, as

$$P(x,t) = \int_0^x \psi(\tilde{x},t)\psi^*(\tilde{x},t)d\tilde{x} \tag{2}$$

where we are summing up the probability density $\rho = |\psi|2 = \psi\psi^*$ and expect $P(1,t) = 1$ if the wavefunction is properly normalized. In fact, even if it is not, we would still expect $P(1,t)$ to be conserved to the level of solution error. We will not need to ensure normalization in our implementation.

This integral can be approximated using the trapezoidal formula. Recall, for n approximate values of f_i at values of x, we can approximate

$$\int_{x_1}^{x_n} f(x)dx \approx \frac{1}{2}\sum_{i=1}^{n-1}(f_i + f_{i+1})(x_{i+1} - x_i)$$

We can also define something known as the temporal average of $P(x,t)$:

$$\bar{P}_j = \frac{\sum_{n=1}^{n_t} P_j^n}{n_t}$$

where we just relabeled $P(x,t) = P(x_j, t^n) = P_j^n$. We want $\bar{P}_j$ to be properly normalized such that $\bar{P}_{n_x} = 1$, and will do this by dividing

$$\bar{P}_j = \frac{\bar{P}_j}{\bar{P}_{n_x}}$$

Now, given two values x_1, x_2, we can interpret

$$\bar{P}(x_2) - \bar{P}(x_1)$$

as the fraction of time our quantum particle spends within $x_1 \leq x \leq x_2$, where $\bar{P}(x) = \bar{P}(x_j) = \bar{P}_{x_j}$. For a free particle in zero potential, we would expect, for sufficiently long time, we see

$$\bar{P}(x_2) - \bar{P}(x_1) \rightarrow x_2 - x_1$$

The fraction of time the particle spends in an interval is the size of the interval, due to the fact that we are solving the Schrodinger equation between $0 \leq x \leq 1$. We can then define another quantity known as the excess fractional probability that the particle spends in a given interval as

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

as well as the natural log of this

$$\ln\left(\bar{F}_e(x_1, x_2)\right) = \ln\left(\frac{\bar{P}(x_2) - \bar{P}(x_1)}{x_2 - x_1}\right) \tag{3}$$

The family of exact solutions to (1) is

$$\psi(x,t) = e^{-im^2\pi^2 t} sin(m\pi x) \tag{4}$$

where m is a positive integer.

2.1.1 Crank-Nicolson discretization

To solve our differential equation, we must discretize the continuum domain by first introducing a discretization level, ℓ and a radio of time to space mesh, λ, which is defined as

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

so that the ratio between them is kept constant across all levels. The mesh spacings are then defined as follows

$$\begin{aligned} n_x &= 2^\ell + 1 \\ \Delta x &= \frac{1}{2^\ell} \\ \Delta t &= \lambda \Delta x \\ n_t &= \text{round}\left(\frac{t_{max}}{\Delta t}\right) + 1 \end{aligned}$$

When we apply the Crank-Nicholson discretization approach to (1) and its initial and boundary conditions, we obtain

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

with the spatial and time indexes ranging from

$$j=2,3,...,n_x-1$$
$$n=1,2,...,n_t-1$$

and subject to boundary conditions

$$\psi_1^{n+1}=\psi_{n_x}^{n+1}=0$$
$$n=1,2,...,n_t-1$$

and initial conditions

$$\psi_j^1=\psi_0(x)$$
$$j=1,2,...,n_x$$

We can rewrite the eq. (5) with the advanced time steps on one side, and the previous time steps on the other, so that we can better solve this equation.

$$\frac{i\psi_j^{n+1}}{\Delta t}+\frac{1}{2}\left(\frac{\psi_{j+1}^{n+1}-2\psi^{n+1}{}_j+\psi_{j-1}^{n+1}}{\Delta x^2}\right)-\frac{1}{2}V_j^{n+\frac{1}{2}}\psi_j^{n+1}=\frac{i}{\Delta t}\psi_j^n-\frac{1}{2}\left(\frac{\psi_{j+1}^{n}-2\psi_j^{n}+\psi_{j-1}^{n}}{\Delta x^2}\right)+\frac{1}{2}V_j^{n+\frac{1}{2}}\psi_j^n$$
$$\left(\frac{1}{2\Delta x}\right)\psi_{j+1}^{n+1}+\left(\frac{i}{\Delta t}-\frac{1}{\Delta x^2}-\frac{1}{2}V_j^{n+\frac{1}{2}}\right)\psi_j^{n+1}+\left(\frac{1}{2\Delta x^2}\right)]\psi_{j-1}^{n+1}=\frac{i}{\Delta t}\psi_j^n-\frac{1}{2}\left(\frac{\psi_{j+1}^{n}-2\psi_j^{n}+\psi_{j-1}^{n}}{\Delta x^2}\right)+\frac{1}{2}V_j^{n+\frac{1}{2}}\psi_j^n$$
$$c_j^+\psi_{j+1}^{n+1}+c_j^0\psi_j^{n+1}+c_j^-\psi_{j-1}^{n+1}=f_j$$

where we have rewritten the discretization in the form $\underline{\underline{A}}\vec{\psi}^{n+1}=\vec{f}$ where the diagonals of $\underline{\underline{A}}$ are give by

$$c_j^+=\frac{1}{2\Delta x}$$
$$c_j^0=\frac{i}{\Delta t}-\frac{1}{\Delta x^2}-\frac{1}{2}V_j^{n+\frac{1}{2}}$$
$$c_J^-=\frac{1}{2\Delta x^2}$$
$$f_j=\frac{i}{\Delta t}\psi_j^n-\frac{1}{2}\left(\frac{\psi_{j+1}^{n}-2\psi_j^{n}+\psi_{j-1}^{n}}{\Delta x^2}\right)+\frac{1}{2}V_j^{n+\frac{1}{2}}\psi_j^n$$

The left hand side coefficients will be used in setting up the tridiagonal system later on. The upper and lower diagonals will be c_j^+, c_j^- and the middle diagonal will be c_j^0. The right hand side will be solved each iteration.

2.1.2 Initial Conditions and Potential Types

There will be two types of initial conditions we will consider. The first is the exact family, which takes the form

$$\psi(x,o)=\sin(m\pi x)$$

where we can specify m. The second type is the boosted gaussian, taking the form

$$\psi(x,0)=e^{i\rho x}e^{-\left(\frac{x-x_0}{\delta}\right)^2}$$

where x_0, δ, ρ can be altered as well. We will also consider two types of potentials. The first is no potential, where

$$V(x) = 0$$

and the second is a rectangular barrier or well, taking the form

$$V(x) = \begin{cases} 0 & \text{for } x \leq x_{\min} \\ V_c & \text{for } x_{\max} \leq x \leq x_{\min} \\ 0 & \text{for } x \geq x_{\max} \end{cases}$$

where $x_{\min}, x_{\max}$, and V_c will be specified before solving.

2.2 Implementation

We begin by implementing a solution to eq. (5) in **sch_1d_cn.m** using the coefficients $c_j^+,\ c_j^0,\ c_j^-,\ f_j$. After setting up the initial conditions and potential, it sets up a tridiagonal system (copied from the tutorial) based on those coefficients. Once the sparse matrix is created, we iterate over all the time steps. At each iteration, we solve for f_j according to the formula we found above. Then we left divide A by f to find the ψ^{n+1} terms. Afterwards, the imaginary and real parts of ψ are calculated. $P(x,t)$ is found by iterating over all spatial points, and then using the trapezoidal approximation for the integral. This function also handles different initial conditions and potentials, depending on the input parameters.

sch_1d_lvl_err.m is used for convergence testing by calculating $d\psi^\ell = \psi^{\ell+1} - \psi^\ell$ for a given range $[\ell_{min}, \ell_{max}]$. It first determines the exact integer levels to iterate over by looking at $\ell_{min} : \ell_{max} - 1$. The extra -1 term is because if we have 4 levels of interest, say (6,7,8,9), then we can only calculate the 3 differences $\Delta_{6,7}, \Delta_{7,8}, \Delta_{8,9}$. Next, we solve the Schrodinger eq. using **sch_1d_cn.m** on the lowest level. This determines the coarsened time and space meshing that we will compare each level to. Finally, we iterate over the range of levels specified earlier. At each step, we determine the solution to the Schrodinger equation and compare it to the previous step's solution. Then we scale it accordingly $(4^0, 4^1, 4^2, ...)$ depending on the level.

sch_1d_exact_err.m is used for convergence testing but with replacing $\psi^{\ell+1}$ with ψ_{exact}. Thus we are finding $||E(\psi^\ell)||_2 = ||\psi_{\text{exact}} - \psi^\ell||_2$. First we find the coarsened space and time mesh by solving the Schrodinger equation at ℓ_{min} using **sch_1d_cn.m**. Then this is used to calculate the exact solution. We again iterate over the levels $[\ell_{min}, \ell_{max}]$, and determine the difference between the solution at that level and the exact solution, and then scale the errors accordingly.

ctest_1d.m is a script that performs and plots the convergence tests outlined in the project 2 description. For each type of convergence plot, it calls one of the two functions that determine solution error and the plots it. It plots all the types of convergence tests listed in the problem, such as (exact family, zero potential) and (boosted gaussian, zero potential). It will call **sch_1d_exact_err.m** and **sch_1d_lvl_err.m** to obtain the solution arrays containing the scaled root mean square errors, and the plots them.

In **well_survey.m** solves the 1D Schrodinger equation for the scenario where the particle starts on the let of a barrier of height V_0 and travels towards the right. We then plot $\ln(\bar{F}_e(x_1, x_2))$ versus $\ln(V_0)$, defined in eq. (3).

2.3 Results

Below are the plots from the convergence testing:

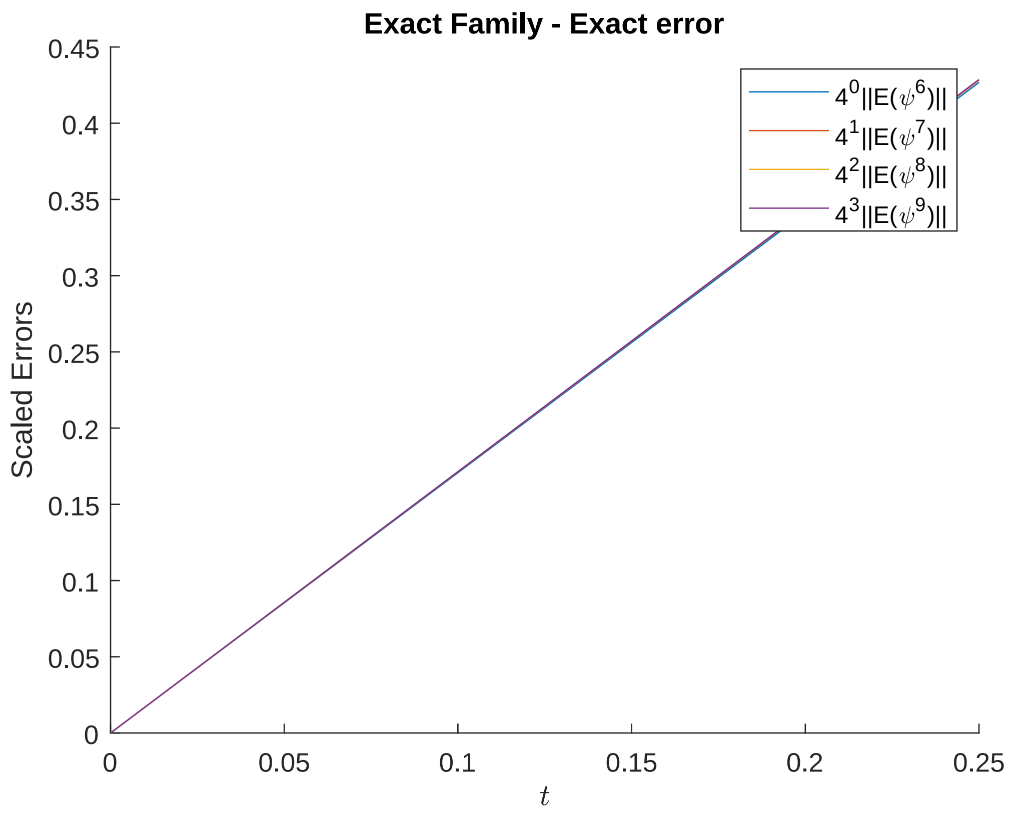

Figure 1: Errors between the computed solution to the 1D Schrodinger equation and the exact solution, for the case where the initial data was the exact family: $\psi(x, 0) = \sin(m\pi x)$, with potential $V(x) = 0$. Errors are scaled according to their level. We see good agreement between these levels $\ell = 6, 7, 8, 9$ as all lines overlap.

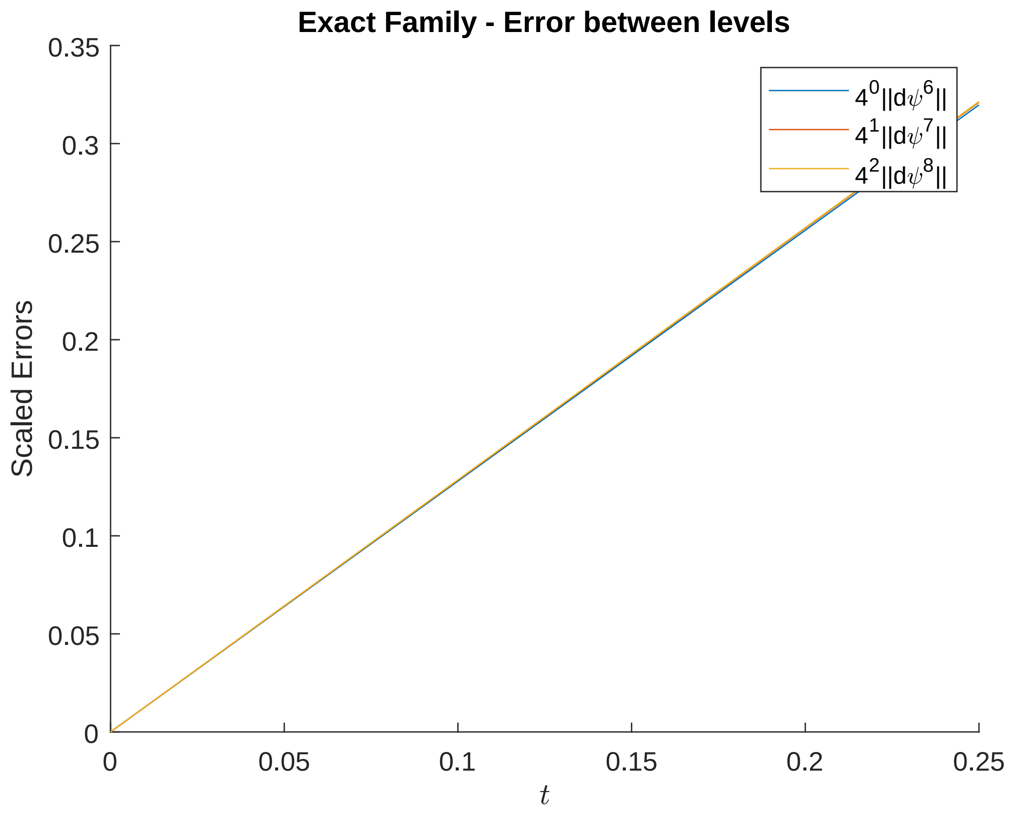

Figure 2: Errors between solutions to the 1D Schrodinger equation done at different levels. Initial data was the exact family: $\psi(x,0) = \sin(m\pi x)$, with potential $V(x) = 0$. Errors are scaled according to their level, and we see good agreement with the overlapping as well. The notation $4^2||d(\psi^8)||$ means the difference between the $\ell = 8$ and $\ell = 9$ solutions was taken ans scaled by 16.

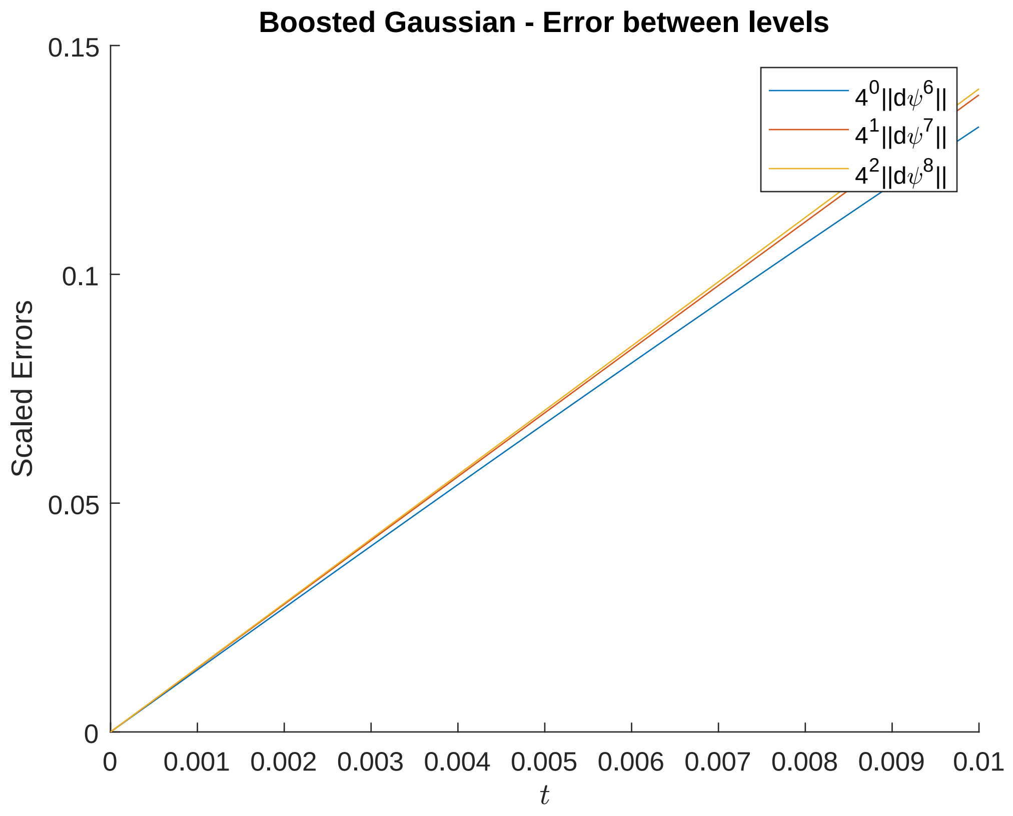

Figure 3: Errors between solutions to the 1D Schrodinger equation done at different levels. Initial data was the boosted gaussian: $\psi(x,0) = e^{i\rho x}e^{-\left(\frac{x-x_0}{\delta}\right)^2}$. The notation $4^2||d(\psi^8)||$ means the difference between the $\ell = 8$ and $\ell = 9$ solutions was taken ans scaled by 16. We see good agreement for $||d(\psi^8)||$ and $||d(\psi^7)||$, but $||d(\psi^6)||$ differs by a noticeable amount. This may suggest that there is higher accuracy at those higher ℓ.

Below are the plots for the well and barrier survey:

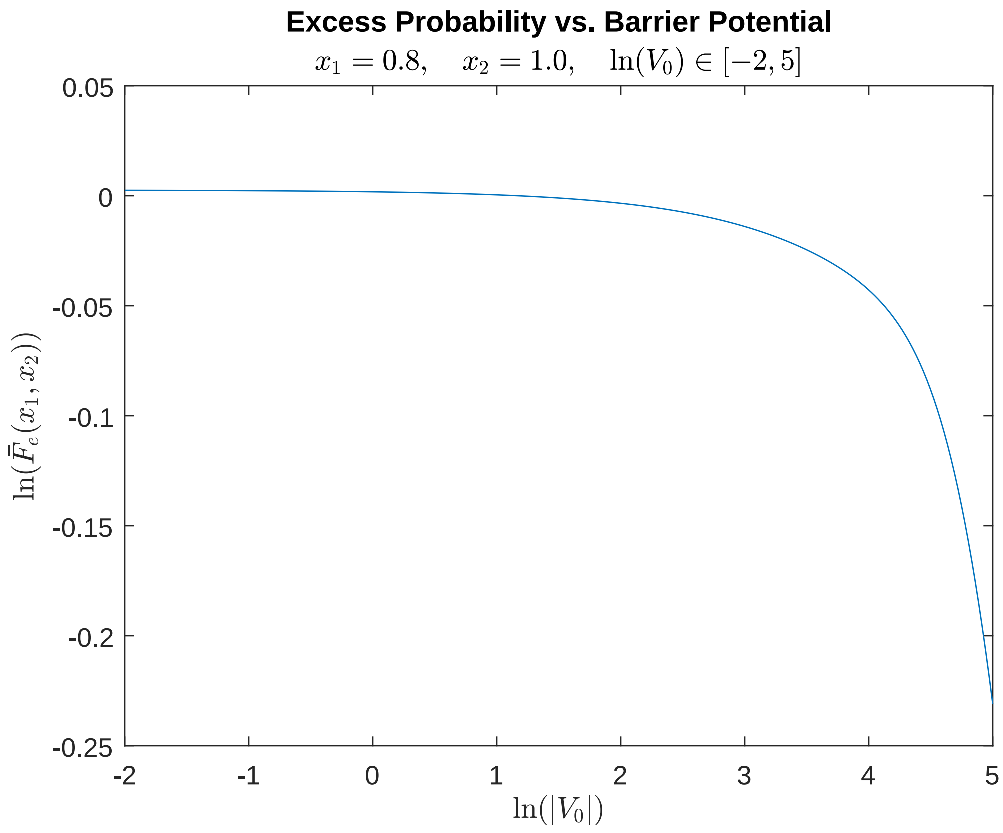

Figure 4: The excess fractional probability $(\bar{F}_e)$ of the scattering of a particle off a potential barrier. This $\bar{F}_e$ can be interpreted as the amount of time a particle in a non-zero potential spends in the region $[x_1, x_2]$ compared to a free particle. The particle begins at $x = 0.4$, and is moving rightward towards the barrier located at $[0.6, 0.8]$. We want to look at the probability that the particle makes it past this barrier, so we calculate $\bar{F}_e(0.8, 1)$. For small V_0, the barrier is low and the particle makes it through nearly as often as a free particle does (since $\ln(1) = 0$). As V_0 increases we see a rapid decrease in the probability the particle can make it past.

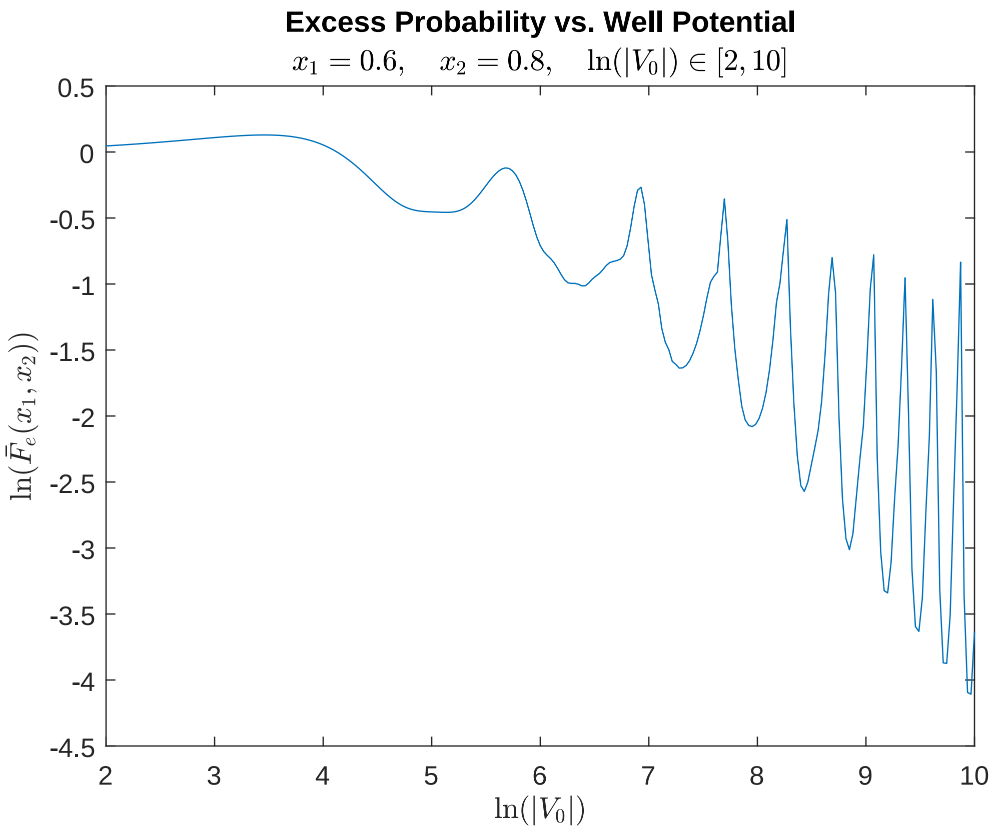

Figure 5: The excess fractional probability ($\bar{F}_e$) of the scattering of a particle off a potential well. The particle begins stationary at $x = 0.4$ with a potential well located at $[0.6, 0.8]$. Initially, with a shallow well, the particle behaves similar to a free particle. But as the well deepens, we begin to see an overall decrease in the likelihood of the particle being in that region. In fact, as $\bar{F}_e$ decreases, we see oscillatory behaviour with period starting at about $\ln(|V_0|) \approx 0.75$ but decreasing each oscillation, finishing around $\ln(|V_0|) \approx 0.25$. Each time it oscillates, $\bar{F}_e$ shoots up multiple orders of magnitude, almost reaching its original free particle behaviour. I am not sure if this is modelling real world physical behaviours, or if our approximations are breaking down. Perhaps the depth of the well is quantized, and at certain V_0 values, the particle treats the well differently than at other values. There are distinct peaks in the oscillatory decrease of $\bar{F}_e$, so perhaps those values of V_0 are special in some way.

2.4 Discussion

We have implemented a computational approximation of the solution of the 1D Schrodinger equation using the Crank-Nicolson discretization approach. Our solutions show good convergence overall, and our simulations gave us some insight behind the underlying physics.

3 Two-dimensional time dependant Schrodinger equation

3.1 Review of Theory and Numerical Approach

In this problem we are asked to solve the two-dimensional Schrodinger equation using the ADI technique discussed in class. The non-dimensionalized continuum equation is

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

We can multiply through by $-i$ to get

$$\psi_t = i(\psi_{xx} + \psi_{yy}) - iV(x,y)\psi \tag{6}$$

This form is very similar to the diffusion equation with an imaginary diffusion constant and source term. We will solve eq. (6) on the domain

$$0 \le x \le 1$$
$$0 \le y \le 1$$
$$0 \le t \le t_{\max}$$

subject to initial conditions

$$\psi(x,y,0) = \psi_0(x,y)$$

and boundary conditions

$$\psi(0,y,t) = \psi(1,y,t) = \psi(x,0,t) = \psi(x,1,t) = 0$$

A family of exact solutions is given by

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

where m_x and m_y are positive integers.

3.1.1 Discretization and the ADI scheme

We will discretize the continuum domain by introducing a level, ℓ. We will also introduce a constant that will maintain the ratio between time and space mesh spacing. This constant is, again, λ and is defined as

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

Then we can determine the time and space mesh grids as follows

$$\begin{aligned} n_x = n_y &= 2^\ell + 1 \\ \Delta x = \Delta y &= 2^{-\ell} \\ \Delta t &= \lambda \Delta x \\ n_t &= \text{round}(t_{\max}/\Delta t) + 1 \end{aligned}$$

We also define the action of the difference operators ∂_h^{xx} and ∂_h^{yy} on some $u_{i,j}^n$ as

$$\partial_{xx}^h u_{i,j}^n \equiv \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} \tag{7}$$

$$\partial_{yy}^h u_{i,j}^n \equiv \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} \tag{8}$$

Using eq. (6) and the difference operators, we define the ADI discretization as two separate steps, as discussed in class. The first step is to find the expansion about the imaginary point:

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} = \left(1 + i\frac{\Delta t}{2}\partial^h_{xx}\right)\left(1 + i\frac{\Delta t}{2}\partial^h_{yy} - i\frac{\Delta t}{2}V_{i,j}\right)\psi^n_{i,j} \tag{9}$$

and step 2 involves using the $\psi^{n+\frac{1}{2}}_{i,j}$ solution to find the advanced time solution, $\psi^n_{i,j}$. It can be written as follows

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{yy} + i\frac{\Delta t}{2}V_{i,j}\right)\psi^{n+1}_{i,j} = \psi^{n+\frac{1}{2}}_{i,j} \tag{10}$$

both over the indices

$$\begin{aligned} i &= 2, 3, ..., n_x - 1 \\ j &= 2, 3, ..., n_y - 1 \\ n &= 1, 2, ..., n_t - 1 \end{aligned}$$

Equations (9) and (10) subject to initial conditions

$$\psi^1_{i,j} = \psi_0(x_i, y_j)$$

and the boundary conditions

$$\psi^n_{1,j} = \psi^n_{n_x,j} = \psi^n_{i,1} = \psi^n_{i,n_y} = 0$$

In order to implement this, we can rewrite eq. (9) by applying the difference operators to ψ, which gives us

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

Now let us define a new matrix A such that

$$A^n_{i,j} = \psi^n_{i,j} + i\frac{\Delta t}{2}\left[\frac{\psi^n_{i,j+1} - 2\psi^n_{i,j} + \psi^n_{i,j-1}}{\Delta y^2}\right] - i\frac{\Delta t}{2}V_{i,j}\psi^n_{i,j}$$

and now our equation becomes much simpler because we can apply the difference operator ∂^h_{xx} to this new matrix A to obtain

$$\begin{aligned} \left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} &= \left(1 + i\frac{\Delta t}{2}\partial^h_{xx}\right)A^n_{i,j} \\ &= A^n_{i,j} + i\frac{\Delta t}{2}\left(\frac{A^n_{i,j+1} - 2A^n_{i,j} + A^n_{i,j-1}}{\Delta x^2}\right) \end{aligned}$$

We can then apply ∂^h_{xx} and rearrange this in the same form as the 1D case, where we have the advance times on the left hand size, and the current times on the right, to get

$$\psi^{n+\frac{1}{2}}_{i,j} - i\frac{\Delta t}{2}\left(\frac{\psi^{n+\frac{1}{2}}_{i+1,j} - 2\psi^{n+\frac{1}{2}}_{i,j} + \psi^{n+\frac{1}{2}}_{i-1,j}}{\Delta x^2}\right) = A^n_{i,j} + i\frac{\Delta t}{2}\left(\frac{A^n_{i+1,j} - 2A^n_{i,j} + A^n_{i-1,j}}{\Delta x^2}\right)$$

$$\left(\frac{-i\Delta t}{2\Delta x^2}\right)\psi^{n+\frac{1}{2}}_{i+1,j} + \left(\frac{\Delta t}{\Delta x^2} + 1\right)\psi^{n+\frac{1}{2}}_{i,j} + \left(\frac{-i\Delta t}{\Delta x^2}\right)\psi^{n+\frac{1}{2}}_{i-1,j} = A^n_{i,j} + i\frac{\Delta t}{2}\left(\frac{A^n_{i+1,j} - 2A^n_{i,j} + A^n_{i-1,j}}{\Delta x^2}\right)$$

This is now in the form of

$$c^+\psi_{i+1,j}^{n+\frac{1}{2}} + c^0\psi_{i,j}^{n+\frac{1}{2}} + c^-\psi_{i-1,j}^{n+\frac{1}{2}} = f$$

where

$$\begin{aligned} c^+ &= \frac{-i\Delta t}{2\Delta x^2} \\ c^0 &= \frac{\Delta t}{\Delta x^2} + 1 \\ c^- &= \frac{-i\Delta t}{\Delta x^2} \end{aligned}$$

That gives us the solution for step 1, and to get step 2, we use the second discretization formula given in eq. (10), and apply ∂_{xx}^h, rearranging to get

$$\begin{aligned} \left(1 - i\frac{\Delta t}{2}\partial_{yy}^h + i\frac{\Delta t}{2}V_{i,j}\right)\psi_{i,j}^{n+1} &= \psi_{i,j}^{n+\frac{1}{2}} \\ \psi_{i,j}^{n+1} - \frac{i\Delta t}{2}\left(\frac{\psi_{i,j+1}^{n+1} - 2\psi_{i,j}^{n+1} + \psi_{i,j-1}^{n+1}}{\Delta y^2}\right) + i\frac{\Delta t}{2}V_{i,j}\psi_{i,j}^{n+1} &= \psi_{i,j}^{n+\frac{1}{2}} \\ \left(1 + \frac{i\Delta t}{\Delta y^2} + \frac{i\Delta t}{2}V_{i,j}\right)\psi_{i,j}^{n+1} + \left(-\frac{i\Delta t}{2\Delta y^2}\right)\psi_{i,j+1}^{n+1} + \left(-\frac{i\Delta t}{2\Delta y^2}\right)\psi_{i,j-1}^{n+1} &= \psi_{i,j}^{n+\frac{1}{2}} \end{aligned}$$

where we once again have rearranged it in the form

$$c^+\psi_{i,j+1}^{n+1} + c^0\psi_{i,j}^{n+1} + c^-\psi_{i,j-1}^{n+1} = f$$

where

$$\begin{aligned} c^+ &= -\frac{i\Delta t}{2\Delta y^2} \\ c^0 &= 1 + \frac{i\Delta t}{\Delta y^2} + \frac{i\Delta t}{2}V_{i,j} \\ c^- &= -\frac{i\Delta t}{2\Delta y^2} \end{aligned}$$

This can now be used to solve for the advanced time step solution. We will once again set up a tridiagonal system, like in problem 1, and then use left division to solve for $\psi_{i,j}^{n+1}$.

3.2 Implementation

sch_2d_adi.m solves the 2D Schrodinger equation using the above method. It takes in some parameters specifying how to solve it (max integration time, level, initial data, ...) and uses the ADI two step technique to solve it. It iterates over all time steps. At each step, it loops over all the j indices, and at each j, it solves for all columns of i. It does this by performing step 1, or eq. (9), of our ADI technique. Afterwards, it loops over all the i indices, and at each i, it solves for all rows of j. It does this part by using step 2, or eq. (10). Before doing this, it determines what type of initial condition and what potential to use based on the inputs given.

sch_2d_exact_err.m is a function that calculates the error between the exact solution and the computed one over a range of levels. It will determine how many levels it needs to solve for, then will solve the 2D Schrodinger equation at the lowest level using **sch_2d_adi.m**. This gives us the time and space mesh spacings that will be used to compare different levels. Then, the exact solution is calculated, and the solution at each of the different levels is compared to this exact solution. Their root mean square difference is stored and scaled according to the level. Finally, it returns the time mesh and the errors at each time.

sch_2d_lvl_err.m is a function that calculates the error between solutions at adjacent levels. It takes some parameters that tell it what range of levels to do this for, as well as the type of solution to compare (initial condition, potential, etc.). Then, it calculates the solution at the lowest level using **sch_2d_adi.m** to get the time mesh and solution at that level. It will then compare the solution at a higher level to that of a previous one by looping over the range of desired levels. The root mean square of each level is scaled and stored.

ctest_2d.m performs convergence testing and plots it. Depending on the convergence test type, it will use both **sch_2d_exact_err.m** as well as **sch_2d_lvl_err.m**. It calls them using parameters given in the problem handout that correspond to a comparison to the exact solution with the right initial condition. It calls those functions, then uses their outputs to plot using a loop. The bulk of the work is done in each of those functions, all this script does is call them and plot the results.

sch_2d_animate.m is a function that creates an animation of the 2D Schrodinger equation's time evolution. It starts by solving the equation with the given initial conditions and potential using **sch_2d_adi.m**. Then it animates it using almost the same code as the galaxy simulation project, except instead of just plotting at each time step, we use a contour plot so that the values of ψ can be seen. I chose to plot $||\psi||$ because it seemed to give me the most consistent result. When I plotted the imaginary or real parts, I ran into problems with the colouring of the contours, and found the 2-norm to be the most consistent, and look the best. As to what a physical interpretation of this might be; perhaps we want to show the effect of both the real and imaginary parts of the solution, so plotting just one or the other is not sufficient.

sch_2d_sim.m is a script that allows users to choose which animation to create by changing the value of "experiment." A user can choose to simulate a barrier, well, single slit, or double slit experiment. Based on the experiment chose, the initial conditions and potential will change. The script then calls **sch_2d_animate.m** to animate the solution based on these inputs.

3.3 Results

Below are the results from our convergence testing:

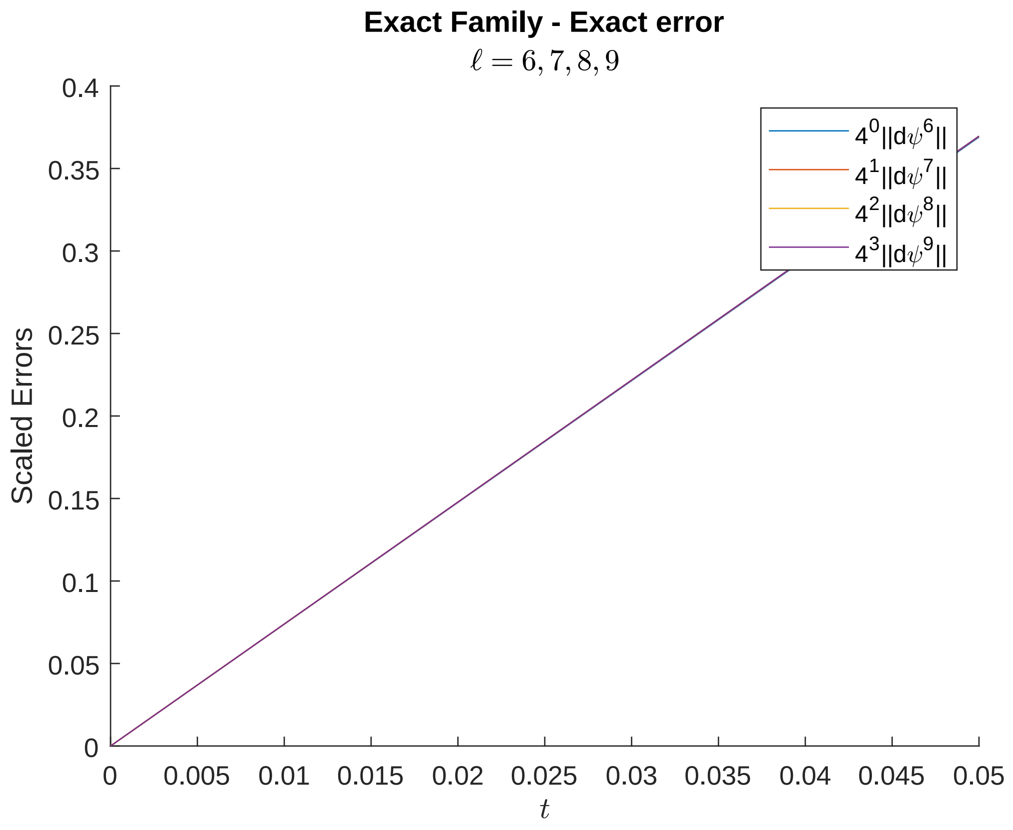

Figure 6: Errors between solutions to the 2D Schrodinger equation and the exact solution. Initial data was the exact family: $\psi(x, 0) = \sin(m\pi x)$, with potential $V(x) = 0$. The differences in the solutions was calculated across both spatial dimensions. Errors are scaled according to their level, and we see good agreement of the different levels across the entire range of time.

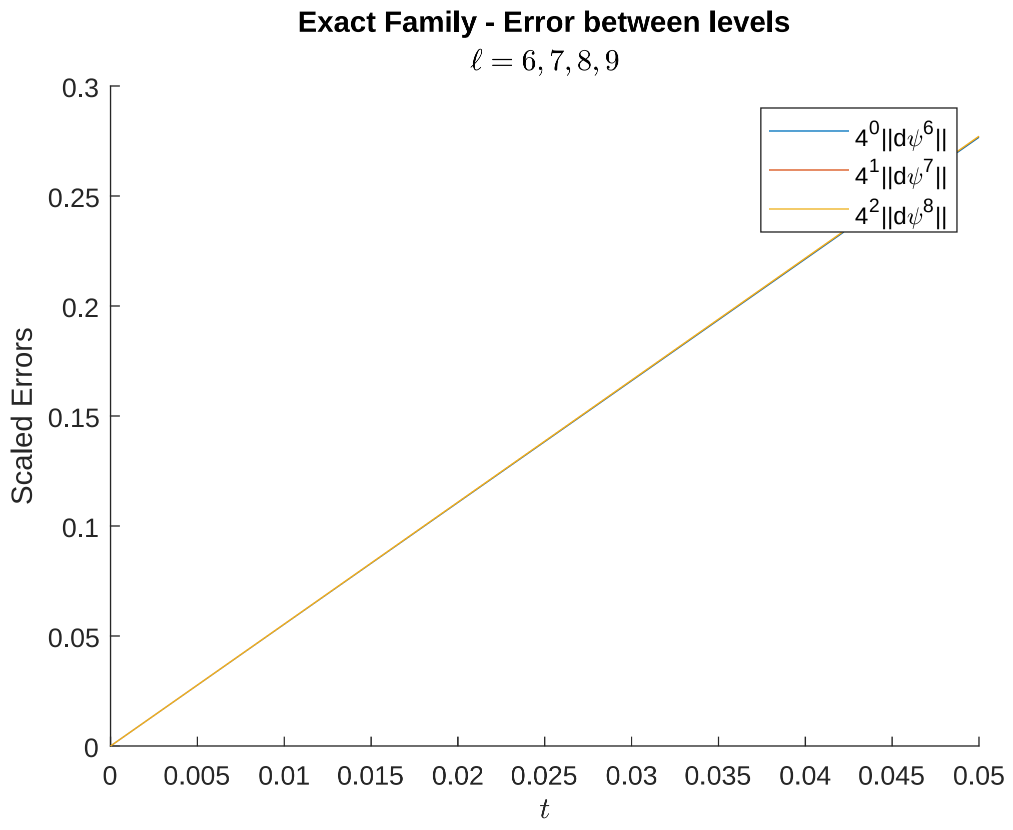

Figure 7: Errors between solutions to the 2D Schrodinger equation done at different levels. Initial data was the exact family: $\psi(x, 0) = \sin(m\pi x)$, with potential $V(x) = 0$. The differences in the solutions was calculated across both spatial dimensions. Errors are scaled according to their level, and we see good agreement between different levels across the entire range of time.

Animations were created for the different cases specified in the project description, and have been included. Gaussian initial data was used for all the scenarios created. Below is an interpretation of those results.

(a) Scattering off a **rectangular barrier**: Two cases are considered; a low potential barrier, and a high potential barrier. In the case of the low barrier, the animation shows that the particle is able to get over the wall, and it even bounces off the other side. For the case of the high barrier, the particle is unable to make it over the wall, and just bounces off of it. In this scenario, the particle is stuck between the high barrier and the edge of the grid. It ends up bouncing back and forth, leaving behind what looks like an interference pattern. A similar pattern is seen in the low barrier case, where a smaller interference pattern is seen. The highest possible V_0 that the particle to overcome seemed to be around $V_0 \leq 1 \times 10^6$. Anything below that, the particle was able to make it over.

(b) Scattering off a **rectangular well**: Two cases are again considered; a shallow well a deep potential well. For the shallow well, similar to the low barrier, the particle is able to get past it quite easily. It hits the other wall, and bounces back, creating what looks like an interference pattern. Then it continues to just oscillate back and forth within the grid. The deep well initially acted just like the high barrier; the particle was unable to make it past the deep well, and would just bounce back and forth in that region it started in. Seeing this, I decided to leave a path for the particle to take at the top of the well. The well only goes from $0 \leq y \leq 0.8$, so the particle found its way to the other side of the well. Intuitively, I did not expect the deep well to behave like the high barrier. I assumed that the particle would sort of "fall" into the well, like a person would when walking into a hole. If I were to consider the scenario of light shining over a deep well, I would not expect the light to get into or illuminate the bottom of the well. Perhaps in some sense that is what is going on here - the light is unable to reach the bottom of the well because it is so deep. For a shallower well, it is able to reach the bottom, as it is similar to the limiting case where you just shine light on flat ground.

(c) Scattering through a **single slit**: The particle begins at the bottom of the screen and is aligned with a very tall barrier that has a small hole in the middle. As it travels upwards, it reaches this hole and passes through it, creating the expected diffraction pattern. As it passes the hole, the wavefunction widens, and as it travels down to the other side of the grid, we see a large central conglomeration, and smaller conglomerations beside it. Once it hits the wall and bounce off, we see the iconic diffraction pattern we expected - one central maxima and many smaller minima.

(d) Scattering through a **double slit**: The particle begins at the bottom of the screen and is aligned with a very tall barrier that has 2 small holes near the middle. As it travels upwards, it reaches them and a small chunk passes through, Once it passes, we see the constructive and destructive interference. As it approaches the other side of the grid, it begins to form into a diffraction pattern, where the central 3 peaks are larger, and they get smaller as they get farther from the center. We have also obtained the expected result for our simulation of the double slit experiment.

3.4 Discussion

We have implemented a computational solution to the 2D time dependant Schrodinger equation using the ADI technique. Our solution shows good convergence overall, and animations display real world physical behaviour, which is good.

Instead of making that matrix A as talked about in the explanation for the numerical approach to solving the discretization, what is actually implemented is a different method. Since I was unable to get the previously described method to work, I ended up just "brute forcing" the solution, and expanding out all components in terms of $\psi^n_{i,j}$. This proved to be a lengthy expansion, since there were a lot of cross terms when both operators $\partial^h_{xx}, \partial^h_{yy}$ acted on $\psi^n_{i,j}$, and the indices would be complicated, but it worked out in the end, thankfully. I chose to explain the more sophisticated method of computing the discretization above, but I will now explain what is implemented in the code. The expansion is again for step 1 of the 2D Schrodinger equation discretization, and begins as:

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

We will still define a new matrix A such that

$$A^n_{i,j} = \psi^n_{i,j} + i\frac{\Delta t}{2}\left[\frac{\psi^n_{i,j+1} - 2\psi^n_{i,j} + \psi^n_{i,j-1}}{\Delta y^2}\right] - i\frac{\Delta t}{2}V_{i,j}\psi^n_{i,j}$$

and now we apply the partial operator to it to get

$$\begin{aligned}\left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} &= \left(1 + i\frac{\Delta t}{2}\partial^h_{xx}\right)A^n_{i,j}\\ &= A^n_{i,j} + i\frac{\Delta t}{2}\partial^h_{xx}A^n_{i,j}\\ &= A^n_{i,j} + i\frac{\Delta t}{2}\partial^h_{xx}\left[\psi^n_{i,j} + i\frac{\Delta t}{2}\left[\frac{\psi^n_{i,j+1} - 2\psi^n_{i,j} + \psi^n_{i,j-1}}{\Delta y^2}\right] - i\frac{\Delta t}{2}V_{i,j}\psi^n_{i,j}\right]\\ &= A^n_{i,j} + i\frac{\Delta t}{2}\left[\partial^h_{xx}\psi^n_{i,j} + \frac{i\Delta t}{2\Delta y^2}\left(\partial^h_{xx}\psi^n_{i,j+1} - 2\partial^h_{xx}\psi^n_{i,j} + \partial^h_{xx}\psi^n_{i,j-1}\right) - \frac{i\Delta t}{2}V_{i,j}\partial^h_{xx}\psi^n_{i,j}\right]\\ &= A^n_{i,j} + i\frac{\Delta t}{2}\left[\frac{\psi^n_{i+1,j} - 2\psi^n_{i,j} + \psi^n_{i,-1j}}{\Delta x^2} + i\frac{\Delta t}{2\Delta y^2}(\gamma) - i\frac{\Delta t}{2}V_{i,j}\frac{\psi^n_{i+1,j} - 2\psi^n_{i,j} + \psi^n_{i-1,j}}{\Delta x^2}\right]\end{aligned}$$

where the part in the brackets is

$$\gamma = \frac{\psi^n_{i+1,j+1} - 2\psi^n_{i,j+1} + \psi^n_{i-1,j+1}}{\Delta x^2} - 2\frac{\psi^n_{i+1,j} - 2\psi^n_{i,j} + \psi^n_{i-1,j}}{\Delta x^2} + \frac{\psi^n_{i+1,j-1} - 2\psi^n_{i,j-1} + \psi^n_{i-1,j-1}}{\Delta x^2}$$

Some more tedious algebra will yield us the expansion for f in our discretization solution. But even from this form, we can gleam some important information about the structure of f. Notably, there are terms will all possible indices $(\psi_{i,j}, \psi_{i-1,j-1}, \psi_{i-1,j}, \psi_{i+1,j-1}, ...)$, but they can all be grouped together. Take for example the entire

$$\frac{\psi^n_{i+1,j+1} - 2\psi^n_{i,j+1} + \psi^n_{i-1,j+1}}{\Delta x^2}$$

term can be grouped together as it appears 3 times; once in the beginning of the bracket, one in the end, and once in γ. This gives us a coefficient of

$$\left(1 - i\frac{\Delta t}{2}V_{i,j} - 2i\frac{\Delta t}{2\Delta y^2}\right)$$

Additionally, all ψ terms than have an index of $j-1$ or $j+1$ only appear once in their respective fractions. This is because the partial operator $\partial_x x^h$ was applied to them only once in the entire expansion. This makes our lives much easier, as we can assign variables to things that are grouped together in the implementation. The rest of the calculation is just algebra and expanding the terms, so I will skip over it. The main difficulty I faced during this was figuring out the right indices to use when accessing different parts of ψ in the code. At first, I had a for loop to go through both the x an y dimensions each time step, but then I realized this had already been solved in the tutorial, so I basically copied it, and just indexed the array at the right spots for the advanced and previous time steps.

Doing all the algebra gives the ADI update to f as follows:

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} = w_1\psi_1 + w_2\psi_2 + w_3 + \psi_3 + w_4 + \psi_4$$

where each term is defined as:

$$\begin{aligned}
w_1 &= \left(1 - i\frac{\Delta t}{\Delta x^2}\right)\left(1 - i\frac{\Delta t}{\Delta x^2} - i\frac{\Delta t}{2}V_{i,j}\right) \\
w_2 &= \left(i\frac{\Delta t}{2\Delta x^2}\right)\left(1 - i\frac{\Delta t}{\Delta x^2} - i\frac{\Delta t}{2}V_{i,j}\right) \\
w_3 &= i\frac{\Delta t}{2\Delta x^2}\left(1 - i\frac{\Delta t}{\Delta x^2}\right) \\
w_3 &= -\left(\frac{\Delta t}{2\Delta x^2}\right)^2 \\
\psi_1 &= \psi^n_{i,j} \\
\psi_2 &= \psi^n_{i-1,j} + \psi^n_{i+1,j} \\
\psi_3 &= \psi^n_{i,j-1} + \psi^n_{i,j+1} \\
\psi_4 &= \psi^n_{i-1,j-1} + \psi^n_{i+1,j-1} + \psi^n_{i-1,j+1} + \psi^n_{i+1,j+1}
\end{aligned}$$