Student Name: David Wang
Student ID: 99030033

PHYS 410: Computational Physics - Project 2 Fall 2022

The Time Dependent Schrödinger Equation

---

1 Introduction

In this project, we seek to solve the time dependent Schrödinger equation. We first do so for the 1D case utilizing the Crank-Nicolson (CN) scheme. Then the solution for 2D is implemented using the Alternating Direction Implicit (ADI) method.

2 1D Schrödinger

The 1D continuum equation (after non-dimensionalization) is

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

We solve the equation on the domain

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

with the initial and boundary conditions

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

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

At some discretization level ℓ we define

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

$$n_x = 2^\ell + 1$$

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

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

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

Applying the Crank Nicolson discretization approach yields:

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

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

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

$$\psi_j^1 = \psi_0(x_j), \quad j = 1, 2, ..., , n_x \tag{3}$$

3 Implmentation of CN

3.1 Numerical solution

We rewrite equation (1) as

$$\left(\frac{1}{2\Delta x^2}\right)\psi_{j+1}^{n+1} + \left(\frac{i}{\Delta t} - \frac{1}{\Delta x^2} - \frac{1}{2}V_j^{n+\frac{1}{2}}\right)\psi_j^{n+1} + \left(\frac{1}{2\Delta x^2}\right)\psi_{j-1}^{n+1} = f_j^n \tag{4}$$

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

To implement the 1-D CN scheme in MATLAB, we utilize a tridiagonal matrix $\mathbf{A}$ such that

$$\mathbf{A}\psi^{n+1} = f^n \tag{5}$$

This implies that $\mathbf{A}$ is an $n_x \times n_x$ tridiagonal matrix with

$$\text{dl} = \begin{bmatrix} \frac{1}{2\Delta t} \\ \vdots \\ \frac{1}{2\Delta t} \\ 0 \end{bmatrix}, \quad \text{d} = \begin{bmatrix} 1 & & & \\ & \frac{i}{\Delta t} + \frac{1}{\Delta x^2} - \frac{1}{2}V_2^{n+\frac{1}{2}} & & \\ & & \vdots & \\ & & & \frac{i}{\Delta t} + \frac{1}{\Delta x^2} - \frac{1}{2}V_{n_x-1}^{n+\frac{1}{2}} \\ & & & 1 \end{bmatrix}, \quad \text{du} = \begin{bmatrix} 0 \\ \frac{1}{2\Delta t} \\ \vdots \\ \frac{1}{2\Delta t} \end{bmatrix}$$

where dl is the lower diagonal, d is the main diagonal, and du is the upper diagonal, such that equation (4) and the boundary conditions are satisfied.

This is implemented in sch_1d_cn.m and we iterate through each time step to update ψ^{n+1}

```
% Compute solution using Crank Nicolson scheme
f  = zeros(nx,1);
for n = 1 : nt-1
  % RHS of linear system
  f(2:nx-1) = ...
      psi(n, 3:nx) * (- 0.5/dx^2) +  ...
      psi(n, 2:nx-1) .* (1i/dt + 1.0/dx^2 + 0.5*v(2:nx-1)') + ...
      psi(n, 1:nx-2) * (-0.5/dx^2);

  f(1) = 0.0;
  f(nx) = 0.0;

  % Solve system for next time step
  psi(n+1, :) = A \ f;
end
```

3.2 1D Initial conditions

For our simulations, we utilize the variables idtype, idpar, vtype and vpar to define the following initial conditions.

idtype == 0 (Exact Family)

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

$$\text{m} = \textbf{idpar(1)}$$

idtype == 1 (Boosted Gaussian)

$$\psi(x,0) = e^{ipx} e^{-((x-x_0)/\delta)^2}$$

$$x_0 = \textbf{idpar(1)}$$
$$\delta = \textbf{idpar(2)}$$
$$p = \textbf{idpar(3)}$$

vtype == 0 (No Potential)

$$V(x) = 0$$

vtype == 1 (Rectangular Barrier or Well)

$$V(x) = \begin{cases} 0 & x < x_{min} \\ V_c & x_{min} \le x \le x_{max} \\ 0 & x > x_{max} \end{cases}$$

$$x_{min} = \textbf{vpar(1)}$$
$$x_{max} = \textbf{vpar(2)}$$
$$V_c = \textbf{vpar(3)}$$

We code the implementation for the 1D case inside sch_1d_cn.m to produce the real, imaginary, and modulus of the wave function as well as the running integral of the probability.

3.3 Level-wise Convergence Testing in 1D

We denote

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

where ψ^ℓ is the solution computed at level ℓ.

We can then compare the level-to-level RMS $l-2$ norms of $d\psi^\ell$, each scaled accordingly. For the solution to be $O(h^2)$ accurate, we expect near coincidence of the curves $4^{\ell-\ell_{min}+1} \cdot ||d\psi^\ell||_2$ This is verified in ctest_1d.m for the two tests:

Test 1

```
idpar = [3]; tmax = 0.25; lambda = 0.1; idtype = 0; vtype = 0;
vpar = []; lmin = 6; lmax = 9;
```

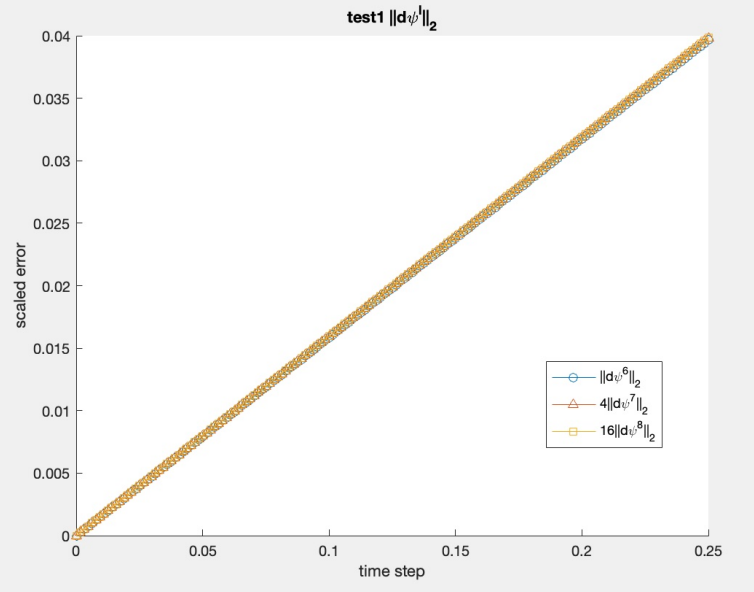

Figure 1: Plot of scaled level-to-level differences for $||d\psi^l||_2$ for levels 6 to 9 for test 1

Test 2

```
idpar = [0.5 0.075 0.0]; tmax = 0.01; lambda = 0.01;
idtype = 1; vtype = 0; lmin = 6; lmax = 9;
```

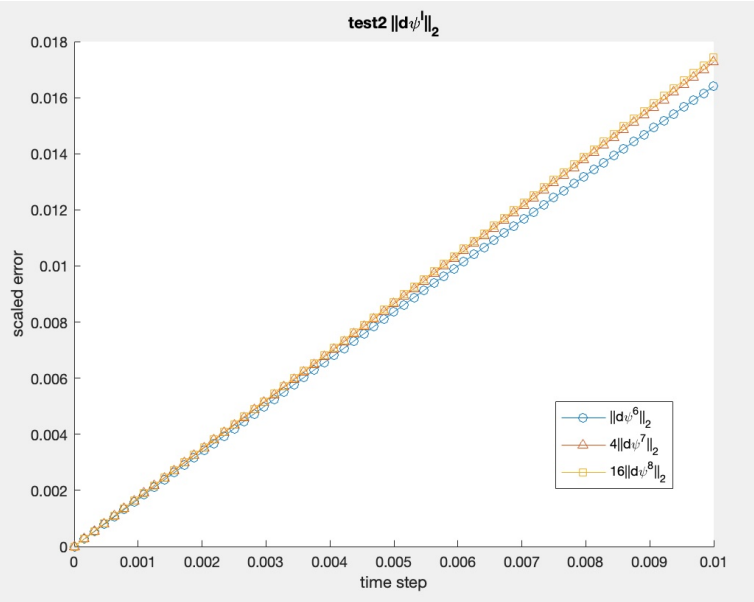

Figure 2: Plot of scaled level-to-level differences for $||d\psi^l||_2$ for level 6 to 9 for test 2

3.4 Exact Family Convergence Testing in 1D

The exact family solution when there is no potential is known as

$$\psi(x,t) = e^{im^2\pi^2 t}\sin(m\pi x) \tag{6}$$

We define

$$E(\psi^\ell) = \psi_{exact} - \psi_\ell$$

and for the solution to be $O(h^2)$ accurate, expect near coincidence of the curves $4^{\ell-\ell_{min}+1}\cdot||E(\psi^\ell)||_2$. This is verified for the parameters of test 1 in ctest_1d.m.

```
idpar = [3]; tmax = 0.25; lambda = 0.1; idtype = 0; vtype = 0;
vpar = []; lmin = 6; lmax = 9;
```

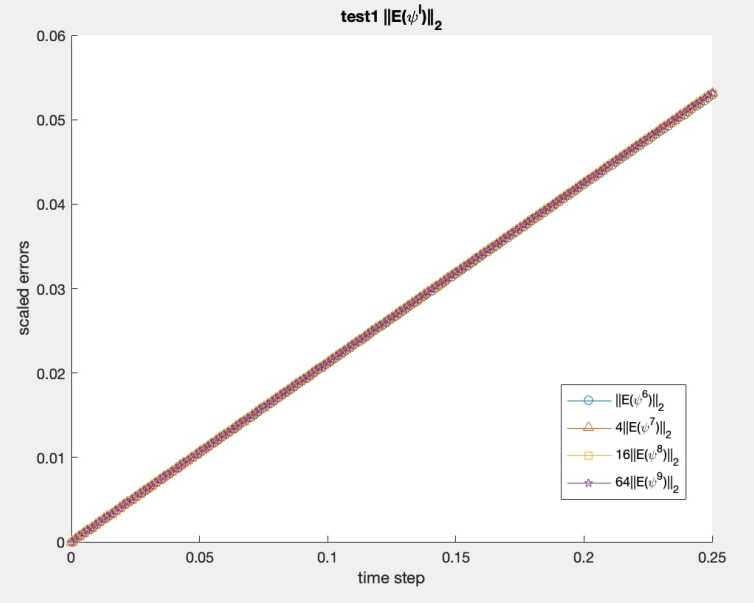

Figure 3: Plot of scaled $||E(\psi^l)||_2$ for levels 6 to 9 for test 1

3.5 Experiments

We define the excess fractional probability that a particle spends in a given spatial interval as

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

where $\bar{P}(x_i)$ is the temporal average of the discrete running integral of the probability density

$$\bar{P}_j = \frac{\sum_{n=1}^{n_t} P(x_j,t^n)}{n_t}$$

and properly normalized so that $\bar{P}_{n_x} = 1$

3.6 Barrier Survey

We use the following parameters.

```
tmax = 0.10; level = 9; lambda = 0.01; idtype = 1; vtype = 1;
idpar = [0.40, 0.075, 20.0]; vpar = [0.6, 0.8, V0]; x1 = 0.8; x2 = 1.0;
```

The experiment is performed using barrier_survey.m. The natural log of the excess fractional probability is determined for 251 uniformly spaced values points of $\ln(V_0)$ from -2 to 5 and plotted Figure (4).

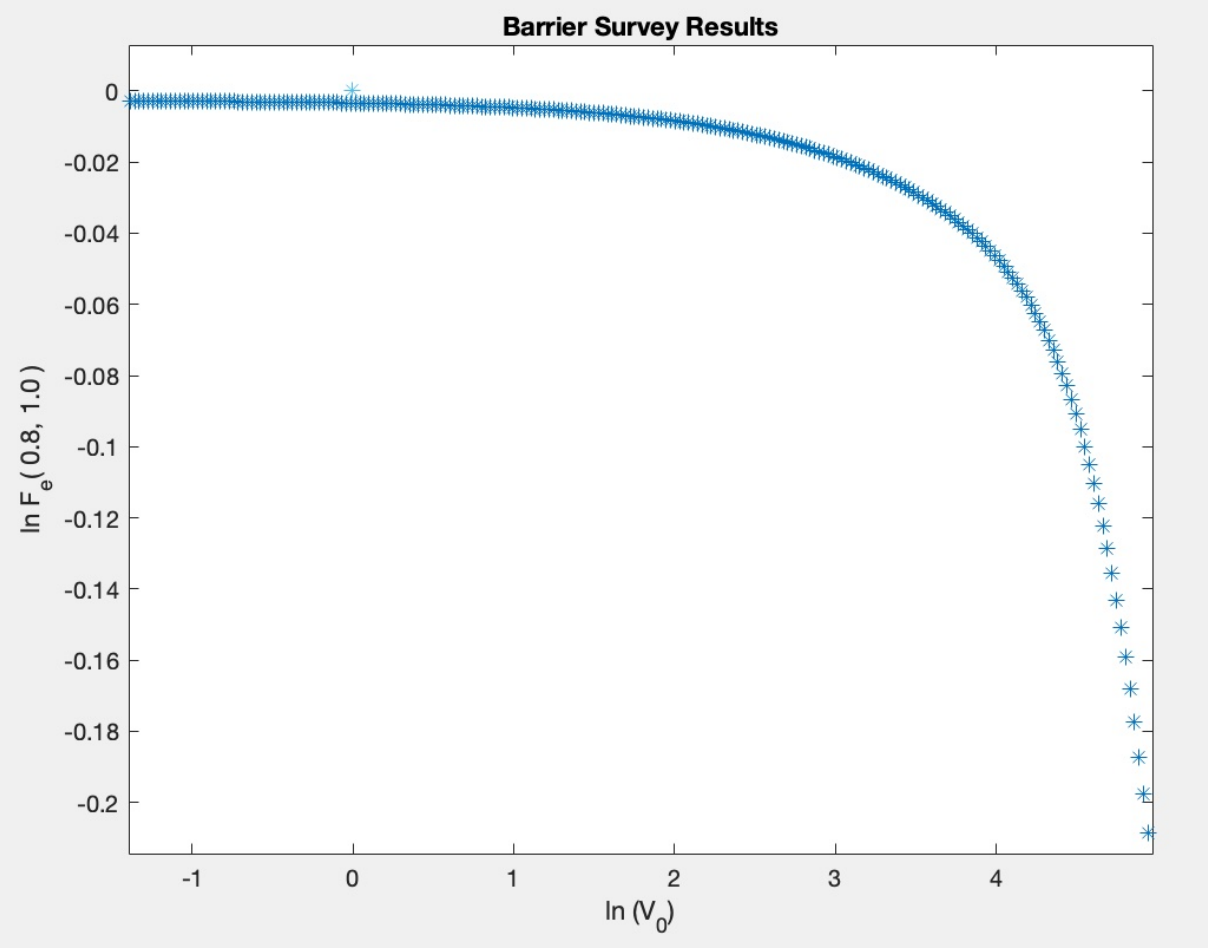

Figure 4: Plot of $\ln(F_e(0.6, 0.8))$ vs $\ln(V_0)$ of barrier

We note that the natural log of the excess fractional probability drops exponentially with the natural log of V_0. This means that with higher potential, the wave function has less proability of existing inside the barrier.

3.7 Well Survey

We use the following parameters.

```
tmax = 0.10; level = 9; lambda = 0.01; idtype = 1; vtype = 1;
idpar = [0.40, 0.075, 20.0]; vpar = [0.6, 0.8, V0]; x1 = 0.8; x2 = 1.0;
```

The experiment is performed using well_survey.m The natural log of the excess fractional probability is determined for 251 uniformly spaced values points of $\ln|V_0|$ from 2 to 10 and plotted Figure (5).

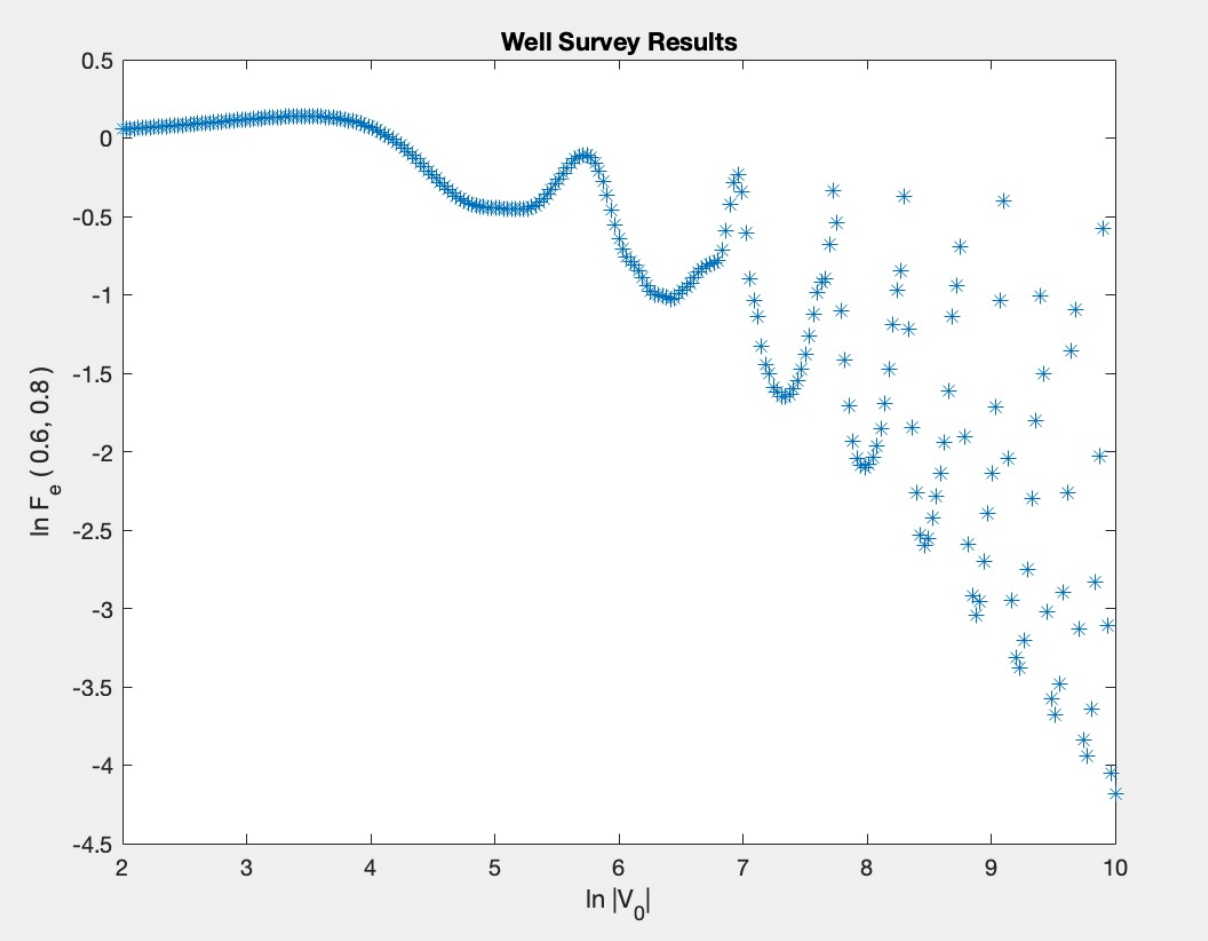

Figure 5: Plot of $\ln(F_e(0.6, 0.8))$ vs $\ln(|V_0|)$ of well

We note that the natural log of excess fractional probability shows a decreasing trend with the natural log of V_0 but also oscillates. This indicates that the permeability of the wave through the potential well is not necessarily scale with the magnitude of the potential but rather is dependent on specific potentials.

4 2D Schrödinger

The 2D continuum equation (after non-dimensionalization) is

$$\psi(x,y,t)_t = i(\psi_{xx} + \psi_{yy}) - iV(x,y)\psi \tag{7}$$

We solve this on the domain

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

with initial and boundary conditions

$$\psi(x,y,0) = \psi_0(x,y)$$
$$\psi(0,y,t) = \psi(1,y,t) = \psi(x,0,t) = \psi(x,1,t) = 0$$

At some discretization level ℓ we define

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

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

Applying the Alternating Direction Implicit method:

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} = \left(1 + i\frac{\Delta t}{2}\partial^h_{xx}\right)\left(1 + i\frac{\Delta t}{2}\partial^h_{yy} - i\frac{\Delta t}{2}V_{i,j}\right)\psi^n_{i,j} \tag{8}$$
$$i = 2,3,...,n_x - 1, \quad j = 2,3,...,n_y - 1, \quad n = 1,2,...,n_t - 1$$
$$\left(1 - i\frac{\Delta t}{2}\partial^h_{yy} + i\frac{\Delta t}{2}V_{i,j}\right)\psi^{n+1}_{i,j} = \psi^{n+\frac{1}{2}}_{i,j} \tag{9}$$
$$i = 2,3,...,n_x - 1, \quad j = 2,3,...,n_y - 1, \quad n = 1,2,...,n_t - 1$$

Where the difference operators are defined by

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

$$\partial^h_{yy}u^n_{i,j} = \left(\frac{1}{\Delta x^2}\right)u^n_{i,j+1} - \left(\frac{2}{\Delta x^2}\right)u^n_{i,j} + \left(\frac{1}{\Delta x^2}\right)u^n_{i,j-1}$$

To solve ADI numerically in matlab, we rewrite (7) and (8) as:

$$\mathbf{D^L_{xx}}\ \psi^{n+\frac{1}{2}} = \mathbf{D^R_{xx}}\ \left(\mathbf{D^R_{yy}}\ (\psi^n)^T\right)^T \tag{10}$$

$$(\mathbf{D^L_{yy}}\ \left(\psi^{n+1}\right)^T)^T = \psi^{n+\frac{1}{2}} \tag{11}$$

Which are defined below:

$\mathbf{D_{xx}^R}$ as an $n_x \times n_x$ tridiagonal matrix with

$$\mathrm{dl} = \begin{bmatrix} \frac{i}{2\Delta t \Delta x^2} \\ \vdots \\ \frac{i}{2\Delta t \Delta x^2} \\ 0 \end{bmatrix}, \quad \mathrm{d} = \begin{bmatrix} 1 \\ 1 - \frac{i}{\Delta t \Delta x^2} \\ \vdots \\ 1 - \frac{i}{\Delta t \Delta x^2} \\ 1 \end{bmatrix}, \quad \mathrm{du} = \begin{bmatrix} 0 \\ \frac{i}{2\Delta t \Delta x^2} \\ \vdots \\ \frac{i}{2\Delta t \Delta x^2} \end{bmatrix}$$

$\mathbf{D_{xx}^L}$ as an $n_x \times n_x$ tridiagonal matrix with

$$\mathrm{dl} = \begin{bmatrix} \frac{-i}{2\Delta t \Delta x^2} \\ \vdots \\ \frac{-i}{2\Delta t \Delta x^2} \\ 0 \end{bmatrix}, \quad \mathrm{d} = \begin{bmatrix} 1 \\ 1 + \frac{i}{\Delta t \Delta x^2} \\ \vdots \\ 1 + \frac{i}{\Delta t \Delta x^2} \\ 1 \end{bmatrix}, \quad \mathrm{du} = \begin{bmatrix} 0 \\ \frac{-i}{2\Delta t \Delta x^2} \\ \vdots \\ \frac{-i}{2\Delta t \Delta x^2} \end{bmatrix}$$

$\mathbf{D_{yy}^R}$ as an $n_y \times n_y$ tridiagonal matrix with

$$\mathrm{dl} = \begin{bmatrix} \frac{i}{2\Delta t \Delta y^2} \\ \vdots \\ \frac{i}{2\Delta t \Delta y^2} \\ 0 \end{bmatrix}, \quad \mathrm{d} = \begin{bmatrix} 1 \\ 1 + \frac{i}{\Delta t \Delta y^2}(1 - V_{2,j}) \\ \vdots \\ 1 + \frac{i}{\Delta t \Delta y^2}(1 - V_{n_x - 1,j}) \\ 1 \end{bmatrix}, \quad \mathrm{du} = \begin{bmatrix} 0 \\ \frac{i}{2\Delta t \Delta y^2} \\ \vdots \\ \frac{i}{2\Delta t \Delta y^2} \end{bmatrix}$$

$\mathbf{D_{yy}^L}$ as an $n_y \times n_y$ tridiagonal matrix with

$$\mathrm{dl} = \begin{bmatrix} \frac{i}{2\Delta t \Delta y^2} \\ \vdots \\ \frac{i}{2\Delta t \Delta y^2} \\ 0 \end{bmatrix}, \quad \mathrm{d} = \begin{bmatrix} 1 \\ 1 + \frac{i}{\Delta t \Delta y^2}(1 + V_{2,j}) \\ \vdots \\ 1 + \frac{i}{\Delta t \Delta y^2}(1 + V_{n_x - 1,j}) \\ 1 \end{bmatrix}, \quad \mathrm{du} = \begin{bmatrix} 0 \\ \frac{i}{2\Delta t \Delta y^2} \\ \vdots \\ \frac{i}{2\Delta t \Delta y^2} \end{bmatrix}$$

where dl is the lower diagonal, d is the main diagonal, and du is the upper diagonal.

We note that the $\mathbf{D_{xx}}$ matrices remain the same throughout the calculation, while the main diagonal of the $\mathbf{D_{yy}}$ matrix is variable depeding on V for each j.

This is implemented in code as follows

```
% % Set up tridiagonal system
dl = ones(nx,1)/dx^2 * 1i*dt/2;
d  = ones(nx,1)*-2.0/dx^2 * 1i*dt/2;
du = ones(nx,1)/dx^2 * 1i*dt/2;

d(1) = 1;
d(nx) = 1;
dl(nx-1) = 0;
du(2) = 0;

% define base matrix to build other matrices
D = spdiags([dl d du], -1:1, nx, nx);

xdiag = ones(nx,1);
xdiag(1) = 0.0;
xdiag(nx) = 0.0;
xdiag_mat = spdiags(xdiag, 0, nx, nx);
% build Dxx matrices, these are fixed for all time steps
Dxx_right = D + xdiag_mat;
Dxx_left = xdiag_mat - D;

% create V matrix with the i*dt/2 factor
V = v*1i*dt/2;

% Compute solution using ADI
for n = 1 : nt-1

  %over-ride boundary conditions to have psi always zero at end points
  psi(n,1,:) = 0.0;
  psi(n, end,:) = 0.0;
  psi(n,:,1) = 0.0;
  psi(n,:,end) = 0.0;

  psi_intermediate = zeros(nx,ny);


  for xi = 2:nx-1 % for each row of psi
      psi_i = reshape(psi(n,xi,:), [ny,1]);

      % reform Dyy_right matrix every iteration with row of V
      ydiag = ones(ny,1)-V(xi,:).';
      ydiag(1) = 0.0;
      ydiag(ny) = 0.0;
      Dyy_right = D + spdiags(ydiag, 0, ny, ny);

      psi_intermediate(xi,:)= Dyy_right * psi_i;
  end

```

```
  psi_half = zeros(nx,ny);

  for yi = 2:ny-1 % for each column of psi, solve for psi_half
      psi_j = psi_intermediate(:,yi);
      psi_half(:,yi) = Dxx_left \ (Dxx_right * psi_j);
  end


  for xi = 2:nx-1 % for each row of psi
      ydiag = ones(ny,1)+V(xi,:).';
      ydiag(1) = 0.0;
      ydiag(ny) = 0.0;
      Dyy_left = spdiags(ydiag, 0, ny, ny) - D;

      psi(n+1, xi, :) = Dyy_left \ (psi_half(xi,:).');
  end

end
```

4.1 2D Initial conditions

For our simulations, we utilize the variables idtype, idpar, vtype and vpar to define the following 2D initial conditions.

idtype == 0 (Exact Family)

$$\psi(x, y, 0) = \sin(m_x \pi x) \sin(m_y \pi y)$$

$$m_x = \textbf{idpar(1)}$$
$$m_y = \textbf{idpar(2)}$$

idtype == 1 (Boosted Gaussian)

$$\psi(x, y, 0) = e^{ip_x x} e^{ip_y y} e^{-((x-x_0)^2/\delta_x^2 + (y-y_0)^2/\delta_y^2)}$$

$$x_0 = \textbf{idpar(1)}$$
$$y_0 = \textbf{idpar(2)}$$
$$\delta_x = \textbf{idpar(3)}$$
$$\delta_y = \textbf{idpar(4)}$$
$$p_x = \textbf{idpar(5)}$$
$$p_y = \textbf{idpar(6)}$$

vtype == 0 (No Potential)

$$V(x, y) = 0$$

vtype == 1 (Rectangular Barrier or Well)

$$V(x,y) = \begin{cases} V_c & (x_{min} \leq x \leq x_{max}) \quad \textbf{and} \quad (y_{min} \leq y \leq y_{max}) \\ 0 & \textbf{otherwise} \end{cases}$$

$$x_{min} = \textbf{vpar(1)}$$
$$x_{max} = \textbf{vpar(2)}$$
$$y_{min} = \textbf{vpar(3)}$$
$$y_{max} = \textbf{vpar(4)}$$
$$V_c = \textbf{vpar(5)}$$

vtype == 2 (Double Slit)

$$V(x,y) = \begin{cases} V_c & \big[(x < x_1) \quad \textbf{or} \quad (x_2 < x < x_3) \quad \textbf{or} \quad (x > x_4)\big] \quad \textbf{and} \quad (y = j' \quad \textbf{or} \quad y = j' + 1) \\ 0 & \textbf{otherwise} \end{cases}$$

$$\text{where j' } = \frac{n_y - 1}{4} + 1$$

$$x_1 = \textbf{vpar(1)}$$
$$x_2 = \textbf{vpar(2)}$$
$$y_3 = \textbf{vpar(3)}$$
$$y_4 = \textbf{vpar(4)}$$
$$V_c = \textbf{vpar(5)}$$

We code the implementation for the 2D case inside sch_2d_adi.m to produce the real, imaginary, and modulus of the wave function in a $[n_t \times n_x \times n_y]$ grid.

4.2 Level-wise convergence in 2D

Level-wise convergence tests are performed for the 2D case with the following parameters

```
idpar = [2, 3]; tmax = 0.05; lambda = 0.05; idtype = 0;
vtype = 0; lmin = 6; lmax = 9;
```

and the scaled norms are plotted in Figure (6).

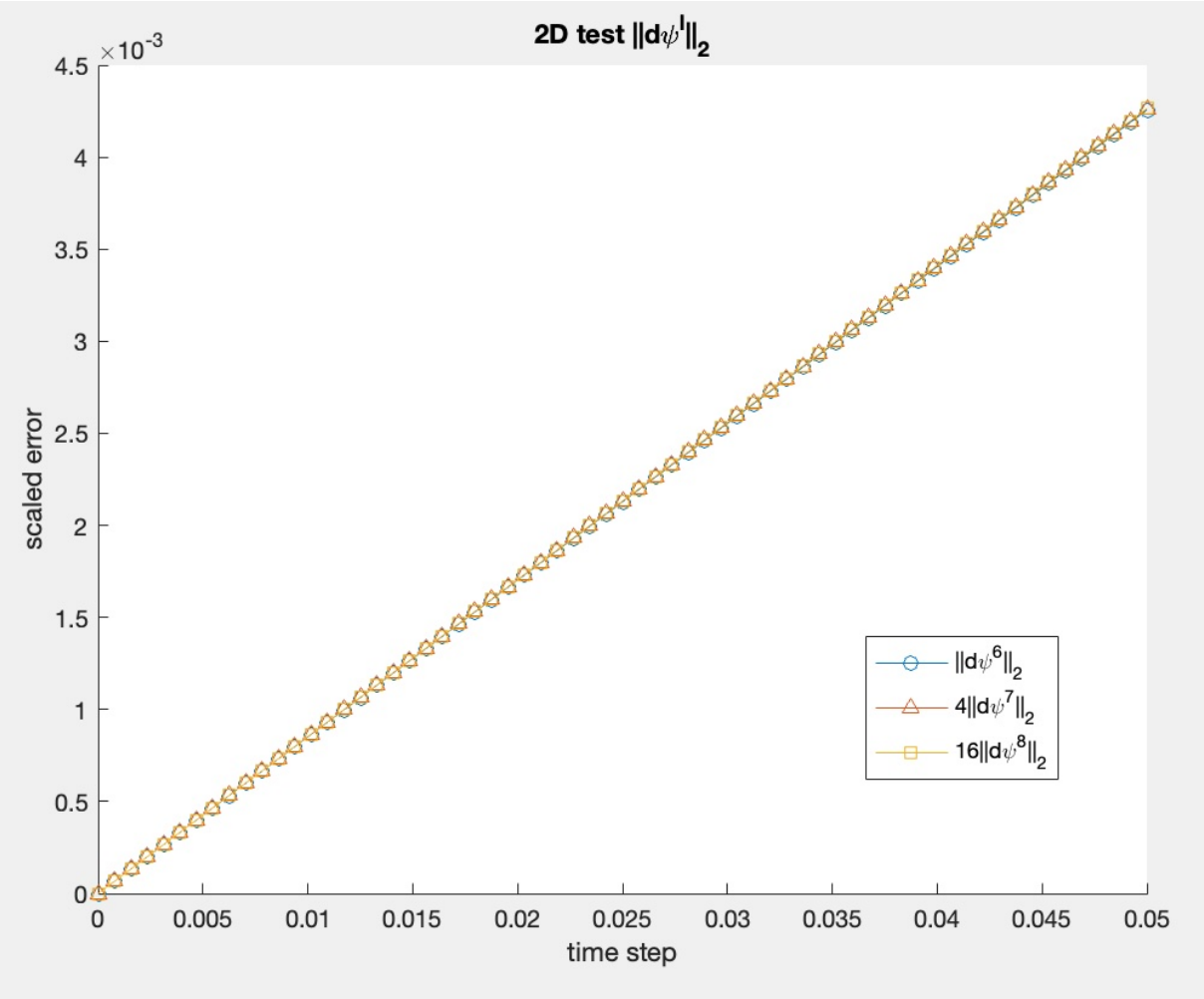

Figure 6: Plot of scaled level-to-level differences for $||d\psi^l||_2$ for level 6 to 9 for 2D

We verify that the implementation is $O(h^2)$ accurate.

4.3 Exact Family Convergence in 2D

With no potential, the exact family solutions are known as

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

Exact family convergence is performed in 2D with the following parameters

```
idpar = [2, 3]; tmax = 0.05; lambda = 0.05; idtype = 0;
vtype = 0; lmin = 6; lmax = 9;
```

and the scaled norms are plotted in Figure (7).

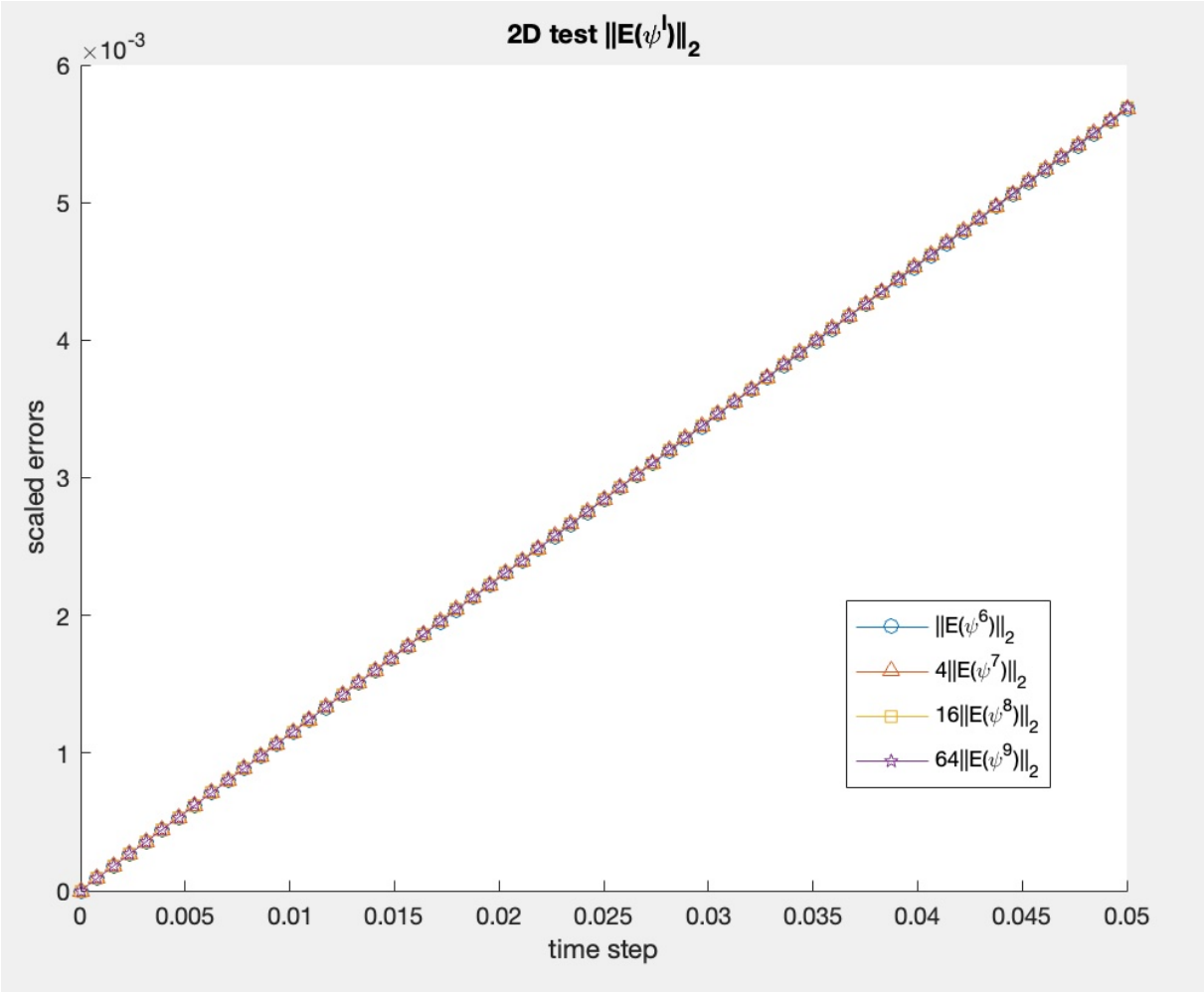

Figure 7: Plot of scaled $||E(\psi^l)||_2$ for level 6 to 9 for 2D

We verify that the implementation is $O(h^2)$ accurate.

5 Numerical Experiments in 2D

We perform various numerical experiments for the time dependent 2D Schrödinger Equation using the make_avi.m and create_animiation.m scripts and and plot napshots at various time steps for the $|\psi|$ and $\mathrm{Re}(\psi)$. The corresponding avi files are included in the project submission. All files, codes, and figures used in the project are included with the submission inside project2.zip.

5.1 Scattering Off a Rectangular Barrier

```
tmax = 0.05; lambda = 0.05; idtype = 1;
x0 = 0.8; y0 = 0.8; deltax = 0.05; deltay = 0.05; px = -5; py = -5;
idpar = [x0, y0, deltax, deltay, px, py];
vtype = 1; xmin = 0.2; xmax = 0.6; ymin = 0.2; ymax = 0.6; vc = 100000;
vpar = [xmin, xmax, ymin, ymax, vc]; level = 6;
```

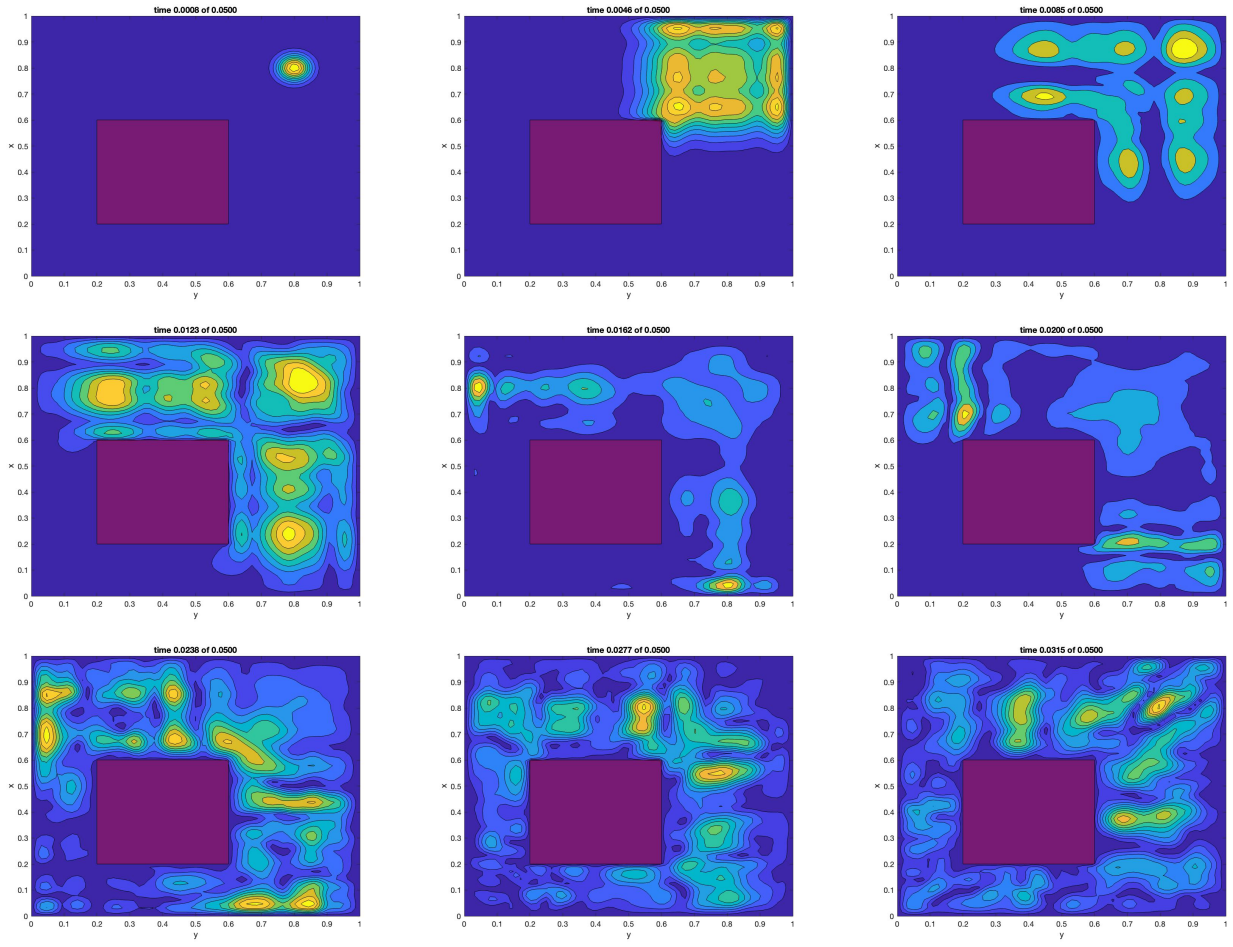

Figure 8: Scattering Off Rectangular Barrier - 2D Contour of $|\psi|$

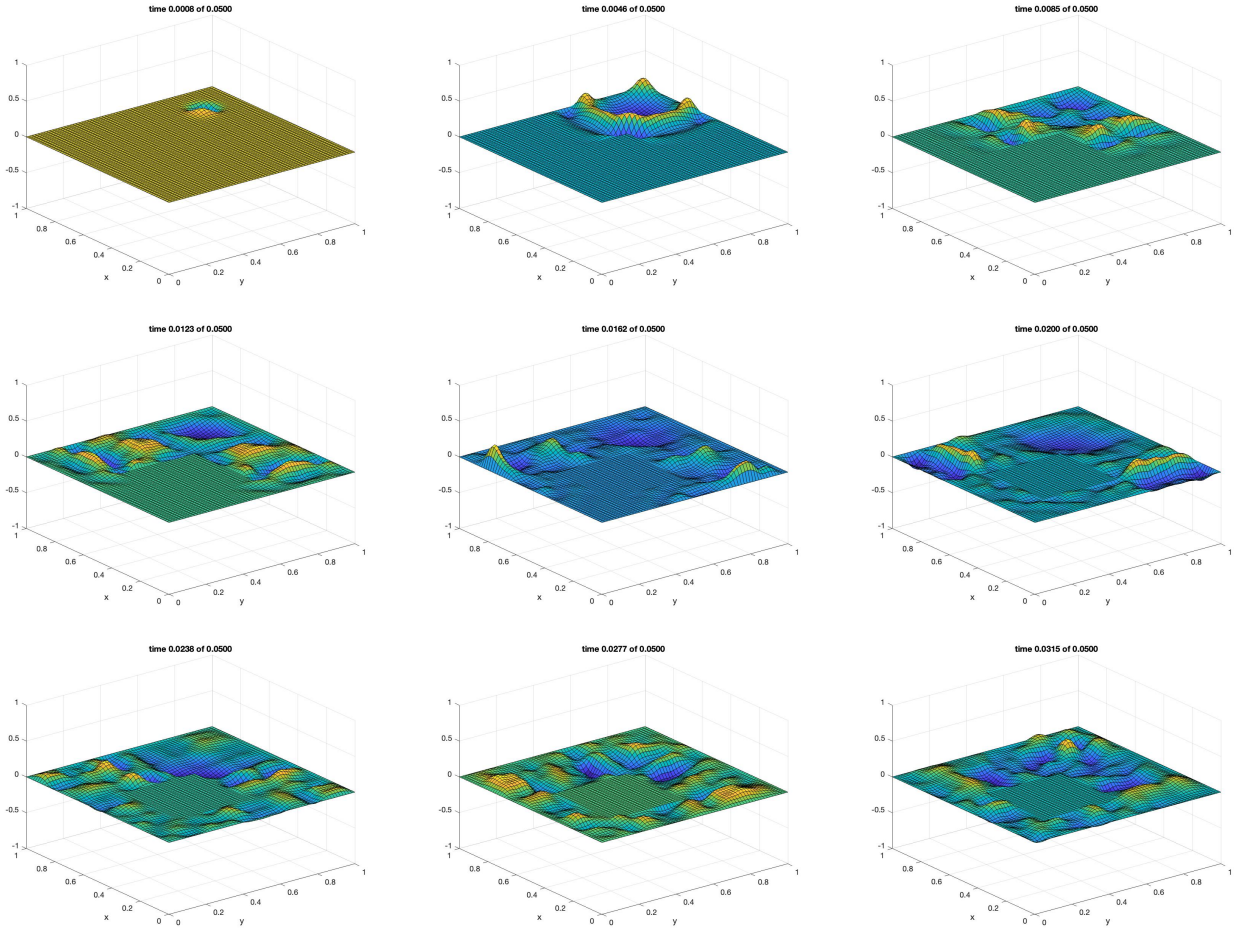

Figure 9: Scattering Off Rectangular Barrier - 3D Contour of Re(ψ)

We note that the barrier with a sufficiently high potential of 100000V, does not allow the wave to pass through, and the wave function propagates around the barrier.

5.2 Scattering Off a Rectangular Well

```
tmax = 0.05; lambda = 0.05; idtype = 1;
x0 = 0.8; y0 = 0.8; deltax = 0.05; deltay = 0.05; px = -5; py = -5;
idpar = [x0, y0, deltax, deltay, px, py];
vtype = 1; xmin = 0.2; xmax = 0.6; ymin = 0.2; ymax = 0.6; vc = -100000;
vpar = [xmin, xmax, ymin, ymax, vc]; level = 6;
```

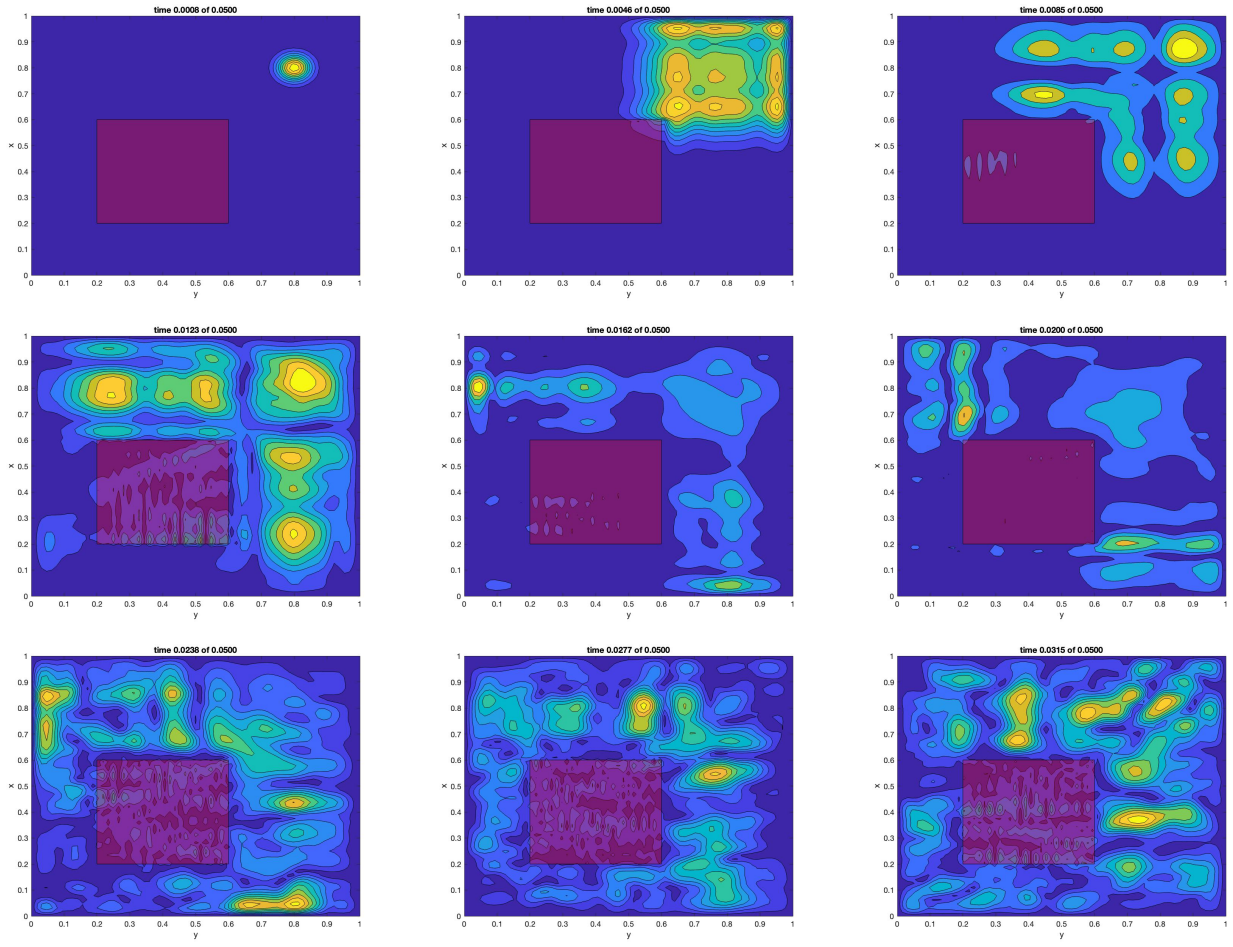

Figure 10: Scattering Off Rectangular Well - 2D Contour of $|\psi|$

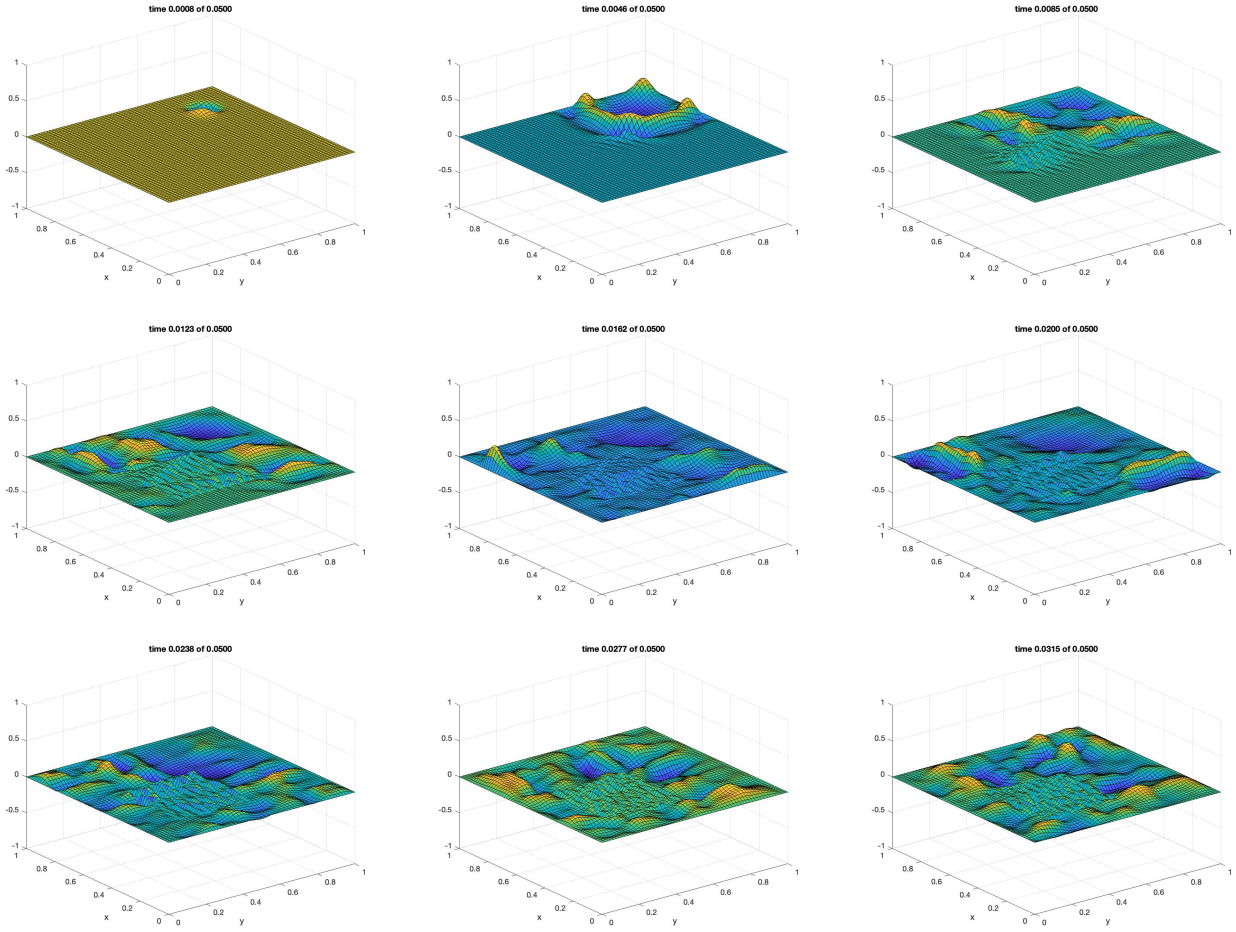

Figure 11: Scattering Off Rectangular Well - 3D Contour of Re(ψ)

We note that even with a low potential of -10000 V, wave function still is able to propagate through the well, although frequencies inside the well are much higher that those outside.

5.3 Scattering Through a Double Slit

```
tmax = 0.05; lambda = 0.05; idtype = 1;
x0 = 0.5; y0 = 0.7; deltax = 0.15; deltay = 0.08; px = 0; py = -40;
idpar = [x0, y0, deltax, deltay, px, py];
vtype = 2; x1 = 0.3; x2 = 0.35; x3 = 0.65; x4 = 0.7; vc = 100000;
vpar = [x1, x2, x3, x4, vc];
```

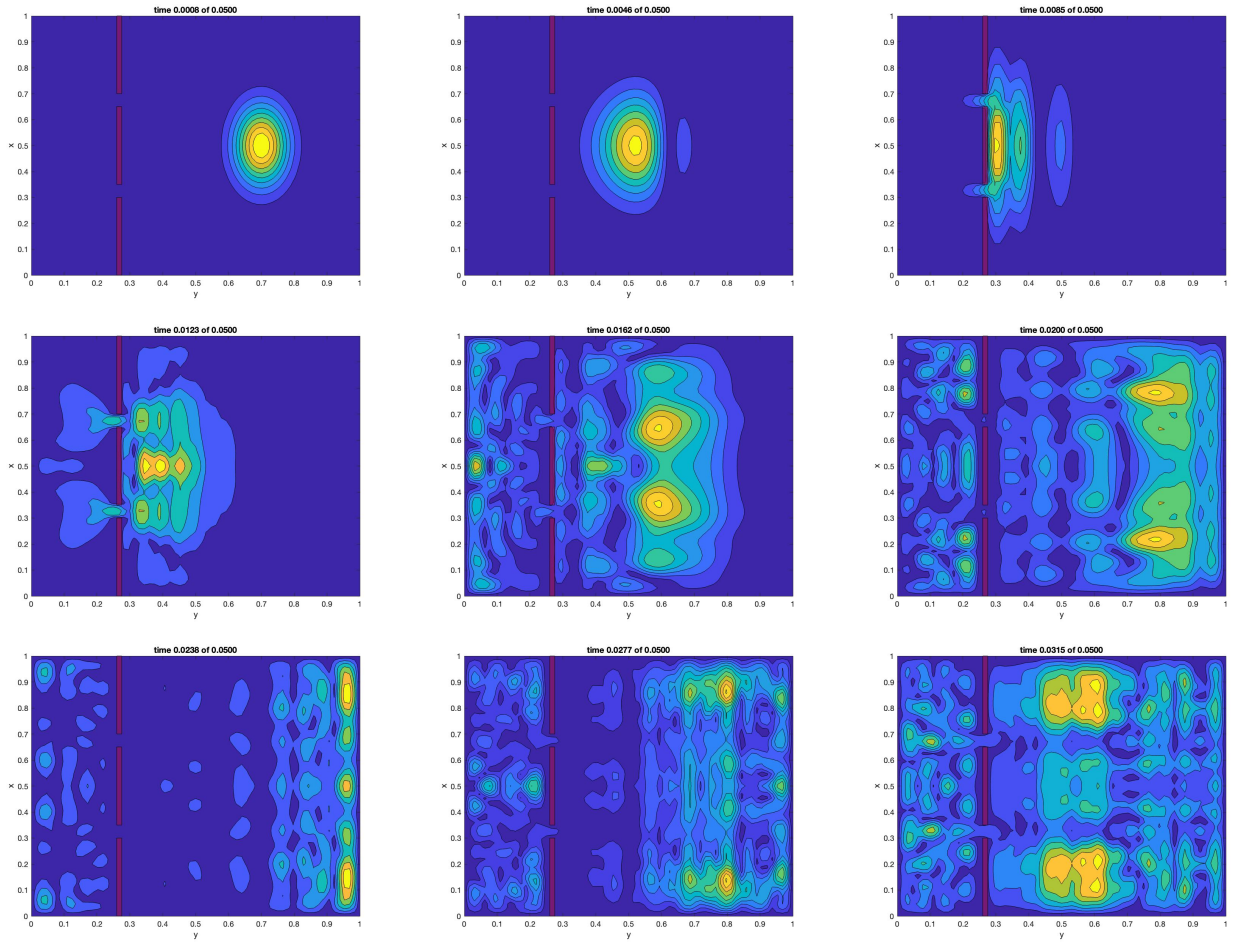

Figure 12: Scattering Through Double Slit - 2D Contour of $|\psi|$

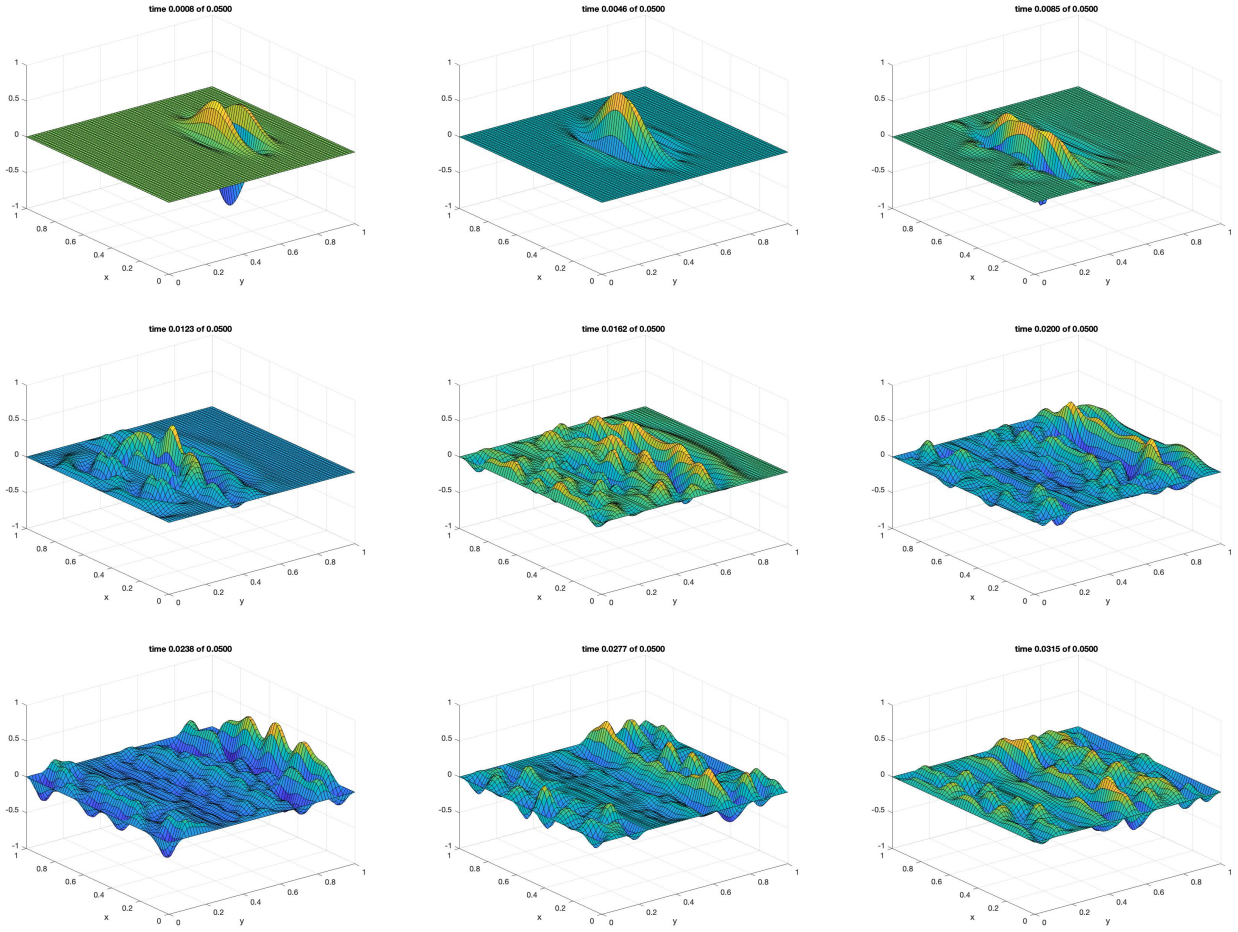

Figure 13: Scattering Through Double Slit - 3D Contour of Re(ψ)

We note that a potential of 100000 effectively acts as a barrier for the wave and only allows the wave to pass through the two slits along the barrier. Interference patterns can be clearly observed on the other side of the slit in both the 2D and 3D plots.

6 Conclusions

In this project, the time dependent Schrödinger Equation was solved, using the Crank Nicolson scheme in 1 dimension and using the Alternating Direction Implicit method in 2 dimensions. Convergence tests were performed for both cases to verify that both schemes were $O(h^2)$ accurate. Numerical experiments were performed for both cases and for the 2D case, contour and 3D plots were made in MATLAB to simulate the behaviour of the wave function at various initial conditions. Overall, the experiments and results make sense.

Improvements can be made to the ADI code for greater speed up. Currently the implementation re-creates the $\mathbf{D_{yy}}$ matrix at every time step and for every row of ψ. However, since we note that the potential stays constant through time, computational complexity can be saved by initializing all the $\mathbf{D_{yy}}$ matrices once in the begining and re-use them for each time step. This will trade off between storage space required and computational complexity of the algorithm.

Further explorations can be done to visualize the behaviour of the wave function in 2 dimensions. For example it would have been interesting to iterate through varying levels for the 2D potential well and visualize the behaviour of the wave function to better study the phenomena observed while performing the well survey.