FINITE DIFFERENCE METHOD FOR THE 1D AND 2D TIME-DEPENDENT SCHRODINGER EQUATION

Seth Gibson
Professor Xianping Li

MAT 494

ASU Spring 2022

ABSTRACT. This project uses the finite difference method to solve the time-dependent Schrodinger equation under different time-independent potential functions for a linear domain and a square domain. I derive and write the code from first principles and show the plotted MATLAB results with the code.

1 The Problem

The manifold domain $\Omega = [0,1] \times [0,1] \subset \mathbb{C}^2$.

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle . \tag{1}$$

The probability of the wavefunction $\Psi(t)$ is

$$P(t) = |\langle \Psi(t)|\Psi(t)\rangle| \in [0,1]. \tag{2}$$

$\hat{H}$ is the (hermitian) Hamiltonian operator, which is the kinetic energy operator $\hat{T} = \frac{\hat{p}}{2m} = -\frac{\hbar^2}{2m}\nabla^2$ plus some potential V.

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = (\hat{T} + V) |\Psi(t)\rangle$$

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = -\frac{\hbar^2}{2m}\nabla^2 |\Psi(t)\rangle + V |\Psi(t)\rangle$$

In one dimension,

$$i\hbar\frac{d}{dt}\left|\Psi(x,t)\right\rangle = \left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x)\right]\left|\Psi(x,t)\right\rangle \tag{3}$$

In two dimensions,

$$i\hbar\frac{d}{dt}\left|\Psi(x,y,t)\right\rangle = \left[-\frac{\hbar^2}{2m}\left(\frac{d^2}{dx^2} + \frac{d^2}{dy^2}\right) + V(x)\right]\left|\Psi(x,y,t)\right\rangle \tag{4}$$

The goal is to find a hermitian, unitary operator $\hat{U}$ such that,

$$\begin{aligned} \hat{U}\hat{U}^\dagger = \hat{U}\hat{U}^{-1} = \mathbf{I} \\ \text{and } \left|\Psi(x,y,t)\right\rangle = \hat{U}\left|\Psi(x,y,t_0)\right\rangle \end{aligned} \tag{5}$$

in a discrete form, which will iterate through the wavefunction evolution from an initial time t_0 to each later time step.

2 Discretization (Crank-Nicolson Scheme)

I used the central difference in time method (theta-method with $\theta = \frac{1}{2}$) since its solution is unconditionally stable.

The space and time is discretized with $l, m, n \in \mathcal{Z}$ (integers),

$$\begin{aligned} t \rightarrow t_l &= [1, N_t], \\ x \rightarrow x_m &= [1, N], \\ y \rightarrow y_n &= [1, N], \\ \text{and solution: } \Psi &\rightarrow \psi. \end{aligned}$$

The space and time discrete differentials are $\delta x = \delta y = \frac{1}{N}$, and $\delta t = \frac{1}{N_t}$, fixed during the run-time.

2.1 One-Dimension

For 1D, the temporal discretization is

$$i\hbar\frac{\psi^{l+1} - \psi^l}{\delta t} = \frac{1}{2}\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi + V\psi\right]^{l+1} + \frac{1}{2}\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi + V\psi\right]^{l}. \tag{6}$$

Rearranging for convenience,

$$\psi^{l+1} + \frac{i\delta t}{2\hbar}\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi + V\psi\right]^{l+1} = \psi^l - \frac{i\delta t}{2\hbar}\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi + V\psi\right]^{l}. \tag{7}$$

Now, central difference discretization of space is

$$\psi_m^{l+1} + \frac{i\delta t}{2\hbar}\left[-\frac{\hbar^2}{2m}\frac{\psi_{m+1}^{l+1} - 2\psi_m^{l+1} + \psi_{m-1}^{l+1}}{\delta x^2} + V_m\psi_m^{l+1}\right]$$
$$= \psi_m^l - \frac{i\delta t}{2\hbar}\left[-\frac{\hbar^2}{2m}\frac{\psi_{m+1}^{l} - 2\psi_m^{l} + \psi_{m-1}^{l}}{\delta x^2} + V_m\psi_m^{l}\right]. \tag{8}$$

Simplifying,

$$\psi_m^{l+1} - \frac{i\hbar\delta t}{4m\delta x^2}\left(\psi_{m+1}^{l+1} - 2\psi_m^{l+1} + \psi_{m-1}^{l+1}\right) + \frac{i\delta t}{2\hbar}V_m\psi_m^{l+1}$$
$$= \psi_m^l + \frac{i\hbar\delta t}{4m\delta x^2}\left(\psi_{m+1}^{l} - 2\psi_m^{l} + \psi_{m-1}^{l}\right) - \frac{i\delta t}{2\hbar}V_m\psi_m^{l}. \tag{9}$$

Setting $b = \frac{\hbar\delta t}{4m\delta x^2}$ and $a_m = \frac{\delta t}{2\hbar}V$,

$$\psi_m^{l+1} - ib\psi_{m+1}^{l+1} + 2ib\psi_m^{l+1} - ib\psi_{m-1}^{l+1} + ia_m\psi_m^{l+1}$$
$$= \psi_m^l + ib\psi_{m+1}^l - 2ib\psi_m^l + ib\psi_{m-1}^l - ia_m\psi_m^l. \tag{10}$$

Simplifying,

$$-ib\psi_{m-1}^{l+1} + (1 + 2ib + ia_m)\psi_m^{l+1} - ib\psi_{m+1}^{l+1} = +ib\psi_{m-1}^l + (1 - 2ib - ia_m)\psi_m^l + ib\psi_{m+1}^l. \tag{11}$$

In full matrix form,

$$\left(\mathbf{I} + \begin{pmatrix} (2ib + ia_1) & -ib & 0 & \cdots & 0 \\ -ib & (2ib + ia_2) & -ib & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -ib & (2ib + ia_N) \end{pmatrix}\right)\begin{pmatrix} \psi_1^{l+1} \\ \psi_2^{l+1} \\ \vdots \\ \psi_N^{l+1} \end{pmatrix}$$
$$= \left(\mathbf{I} - \begin{pmatrix} (2ib + ia_1) & -ib & 0 & \cdots & 0 \\ -ib & (2ib + ia_2) & -ib & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -ib & (2ib + ia_N) \end{pmatrix}\right)\begin{pmatrix} \psi_1^{l} \\ \psi_2^{l} \\ \vdots \\ \psi_N^{l} \end{pmatrix} \tag{12}$$

Calling the assembly matrix $\mathbf{A}$, we can write the iterative equation as

$$\begin{pmatrix} \vdots \\ \psi^{l+1} \\ \vdots \end{pmatrix} = \frac{\mathbf{I} + \mathbf{A}}{\mathbf{I} - \mathbf{A}}\begin{pmatrix} \vdots \\ \psi^{l} \\ \vdots \end{pmatrix}. \tag{13}$$

Clearly, $\frac{\mathbf{I}+\mathbf{A}}{\mathbf{I}-\mathbf{A}}$ is unitary since

$$\frac{\mathbf{I} + \mathbf{A}}{\mathbf{I} - \mathbf{A}}\left(\frac{\mathbf{I} + \mathbf{A}}{\mathbf{I} - \mathbf{A}}\right)^\dagger = \mathbf{I}, \tag{14}$$

and thus satisfies equation (5) in a discrete form.

2.2 Two-Dimensions

Following from equation (4) and discretizing as equation (7),

$$\begin{aligned}\psi^{l+1} + \frac{i\delta t}{2\hbar}\left[-\frac{\hbar^2}{2m}\left(\frac{d^2}{dx^2}+\frac{d^2}{dy^2}\right)\psi + V\psi\right]^{l+1} \\ = \psi^{l} - \frac{i\delta t}{2\hbar}\left[-\frac{\hbar^2}{2m}\left(\frac{d^2}{dx^2}+\frac{d^2}{dy^2}\right)\psi + V\psi\right]^{l}.\end{aligned} \tag{15}$$

Now, skipping to a form as in equation (9),

$$\begin{aligned}\psi^{l+1}_{m,n} - \frac{i\hbar\delta t}{4m\delta x^2}\left((\psi^{l+1}_{m+1,n} - 2\psi^{l+1}_{m,n} + \psi^{l+1}_{m-1,n}) + (\psi^{l+1}_{m,n+1} - 2\psi^{l+1}_{m,n} + \psi^{l+1}_{m,n-1})\right) + \frac{i\delta t}{2\hbar}V_{m,n}\psi^{l+1}_{m,n} \\ = \psi^{l}_{m,n} + \frac{i\hbar\delta t}{4m\delta x^2}\left((\psi^{l}_{m+1} - 2\psi^{l}_{m,n} + \psi^{l}_{m-1,n}) + (\psi^{l+1}_{m,n+1} - 2\psi^{l+1}_{m,n} + \psi^{l+1}_{m,n-1})\right) - \frac{i\delta t}{2\hbar}V_{m,n}\psi^{l}_{m,n}.\end{aligned} \tag{16}$$

Replacing with a and b and simplifying,

$$\begin{aligned}\psi^{l+1}_{m,n} - ib\left(\psi^{l+1}_{m+1,n} - 4\psi^{l+1}_{m,n} + \psi^{l+1}_{m-1,n} + \psi^{l+1}_{m,n+1} + \psi^{l+1}_{m,n-1}\right) + ia_{m,n}\psi^{l+1}_{m,n} \\ = \psi^{l}_{m,n} + ib\left(\psi^{l+1}_{m+1,n} - 4\psi^{l+1}_{m,n} + \psi^{l+1}_{m-1,n} + \psi^{l+1}_{m,n+1} + \psi^{l+1}_{m,n-1}\right) - ia_{m,n}\psi^{l}_{m,n}.\end{aligned} \tag{17}$$

$$\begin{aligned}\left(1 + 4ib + ia_{m,n}\right)\psi^{l+1}_{m,n} - ib\psi^{l+1}_{m+1,n} - ib\psi^{l+1}_{m-1,n} - ib\psi^{l+1}_{m,n+1} - ib\psi^{l+1}_{m,n-1} \\ = \left(1 - 4ib - ia_{m,n}\right)\psi^{l}_{m,n} + ib\psi^{l}_{m+1,n} + ib\psi^{l+1}_{m-1,n} + ib\psi^{l}_{m,n+1} + ib\psi^{l}_{m,n-1}.\end{aligned} \tag{18}$$

We will reshape our 2D wavefunction to stay a vector,

$$\begin{pmatrix} \psi_{1,1} \\ \psi_{2,1} \\ \vdots \\ \psi_{N,1} \\ \psi_{1,2} \\ \psi_{2,2} \\ \vdots \\ \vdots \\ \psi_{N-1,N} \\ \psi_{N,N} \end{pmatrix} \tag{19}$$

We form an assembly matrix **B** to act appropriately on this wavefunction by using the Kronecker product.

$$\mathbf{B} = \mathbf{A} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A} \tag{20}$$

This will naturally form block matrices, where **B** has dimensions $N^2 \times N^2$.

Thus, the iterative equation has the form equation as

$$\begin{pmatrix} \psi_{1,1}^{l+1} \\ \psi_{2,1}^{l+1} \\ \vdots \\ \psi_{N,1}^{l+1} \\ \psi_{1,2}^{l+1} \\ \psi_{2,2}^{l+1} \\ \vdots \\ \vdots \\ \psi_{N-1,N}^{l+1} \\ \psi_{N,N}^{l+1} \end{pmatrix} = \frac{\mathbf{I} + \mathbf{B}}{\mathbf{I} - \mathbf{B}} \begin{pmatrix} \psi_{1,1}^{l} \\ \psi_{2,1}^{l} \\ \vdots \\ \psi_{N,1}^{l} \\ \psi_{1,2}^{l} \\ \psi_{2,2}^{l} \\ \vdots \\ \vdots \\ \psi_{N-1,N}^{l} \\ \psi_{N,N}^{l} \end{pmatrix} \tag{21}$$

3 Initial Condition and Various Potential Functions

I decided to the use the "free particle" as the initial condition of the form

$$\begin{aligned} |\Psi(x, y, t_0 = 0)\rangle &= Ae^{2\pi i(x+y)}, \\ \text{where A satisfies } \Sigma P = 1 = A^2 &\iint_\Omega e^{-2\pi i(x+y)} e^{2\pi i(x+y)} dxdy, \\ A &= 1 \end{aligned} \tag{22}$$

Then, I abruptly apply potentials $V(x, y)$, one Gaussian,

$$V(x, y) = 1000e^{-25x^2} e^{-25y^2} \tag{23}$$

and box barriers or wells,

$$V(x, y) = \begin{cases} 100000 & ; \text{for } x, y \in [\frac{1}{3}, \frac{2}{3}] \\ 0 & ; \text{else} \end{cases} \tag{24}$$

$$V(x, y) = \begin{cases} 0 & ; \text{for } x, y \in [\frac{1}{3}, \frac{2}{3}] \\ 100000 & ; \text{else} \end{cases} \tag{25}$$

4 MATLAB Results

One-Dimensional

```
% Constants
hbar = 1; % Planck's
m = 1; % Particle mass

% Domain
L = 1;
Nx = 1000;
Nt = 1e5;
dx = 1/Nx;
dt = 1/Nt;
x = linspace(0, L, Nx)';

% Initial Wavefunction
u0 = sqrt(2).*sin(1*pi*x(:));
% u0 = 0.00001*abs(exp(1i*(pi*x(:) + pi/2)));

% Potential Function
V = -1e4.*exp(-(x - L/2).^2/(2*dx)); % Gaussian

% V = 50*(x(:) - 0.5).^2; % Harmonic

% V = 0*x(:)./x(:); % Square Well
% for j = 1:Nx/3
%     V(j) = -2e4;
%     V(Nx - j) = -2e4;
% end

% V = 2e4*x(:)./x(:); % Square Barrier
% for j = 1:Nx/3
%     V(j) = 0;
%     V(Nx - j) = 0;
% end

% V = 1e5*((x(:)+1).^(-12) - (x(:)+1).^(-6)); % Lennard-Jones

% V = -1./(x(:)+0.001); % Electrical

plot(x,V)

% Assembly Matrix
a = dt.*V/(2*hbar);
b = dt*hbar/(4*m*dx*dx);

u = zeros(Nx, Nt);
u(:, 1) = u0;

I = zeros(Nx, Nx);
I(1, 1) = 1;
I(Nx, Nx) = 1;
A = zeros(Nx, Nx);
A(1, 1) = a(1) + 2*b;
A(1, 2) = -b;
```

```
A(Nx, Nx-1) = -b;
A(Nx, Nx) = a(Nx) + 2*b;

for j = 2:Nx-1
    A(j, j-1) = -b;
    A(j, j) = a(j) + 2*b;
    A(j, j+1) = -b;

    I(j, j) = 1;
end

% Unitary Matrix
U = (I - 1i.*A)/(I + 1i.*A);

for t = 1:Nt
    u(:, t+1) = U*(u(:, t)./(sqrt( abs( u(:, t)'*u(:, t) ) )));
end

plot(x, abs(u(:, 1)).^2, x, real(u(:, 1)), 'm', x, imag(u(:, 1)), 'c');

for t = 1:Nt/100
%     plot(x, real(u(:, t)), x, imag(u(:, t)));
    plot(x, abs(u(:, t)).^2, 'b', x, V(:)/4e5, 'k', x, real(u(:, t))/50, ...
        'm', x, imag(u(:, t))/50, 'c');
    axis([0 1 -0.03 0.03]);
    pause(0.001)
end
```

The dark blue line is the probability of the wavefunction. The magenta line is the real component of the function and the cyan is the complex.

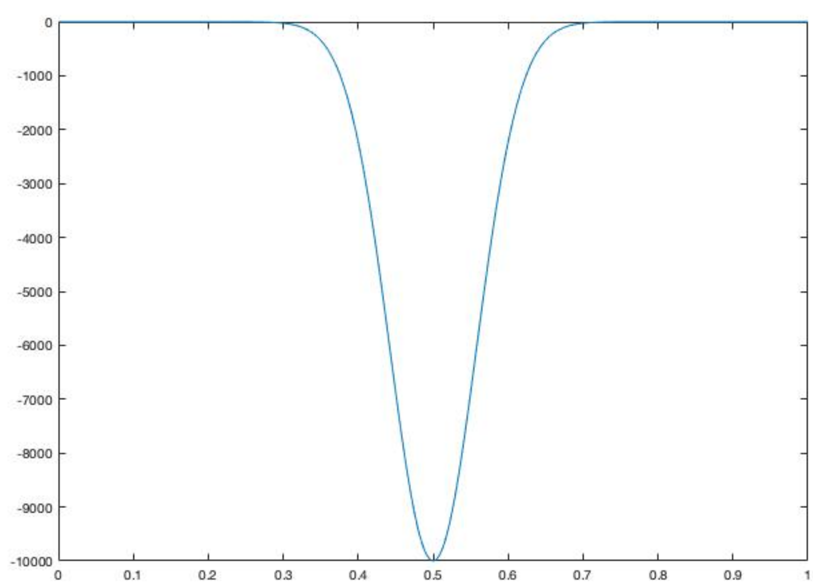


Figure 1: 1D Gaussian Well

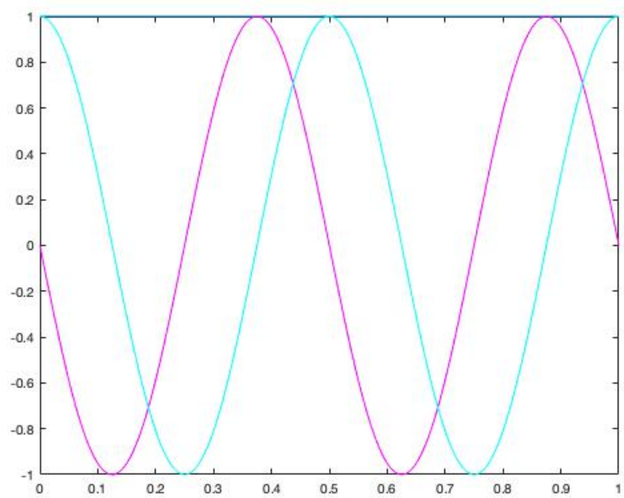
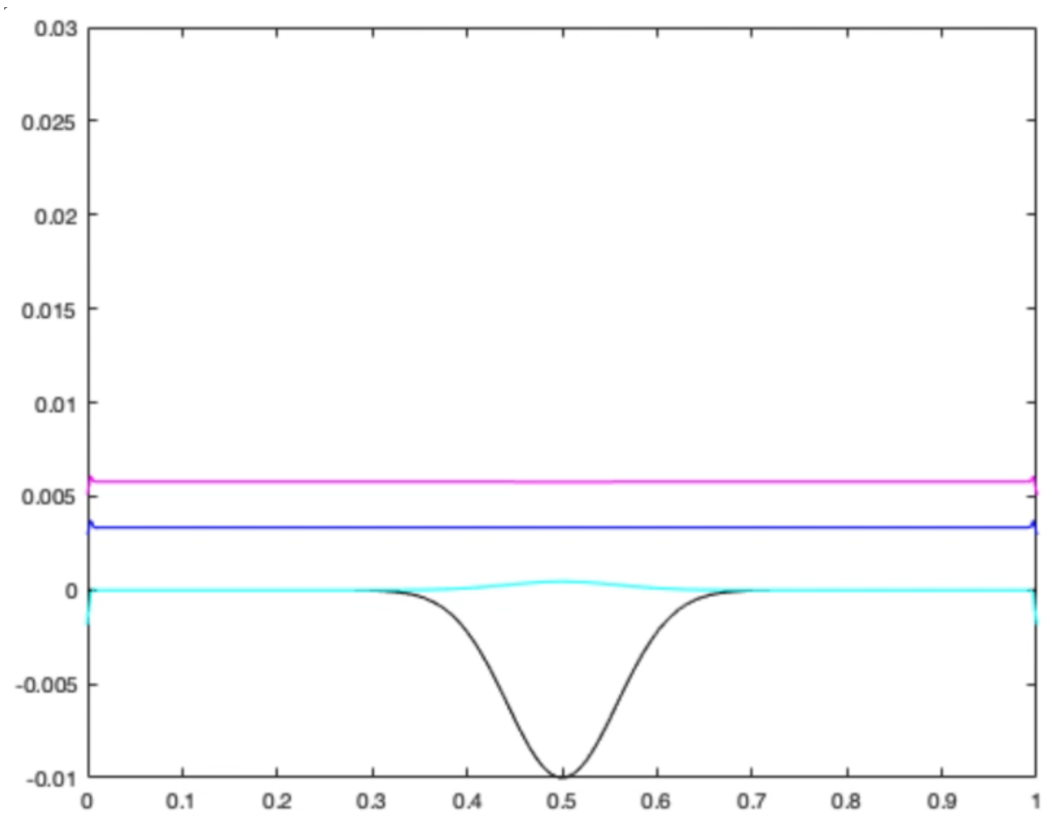
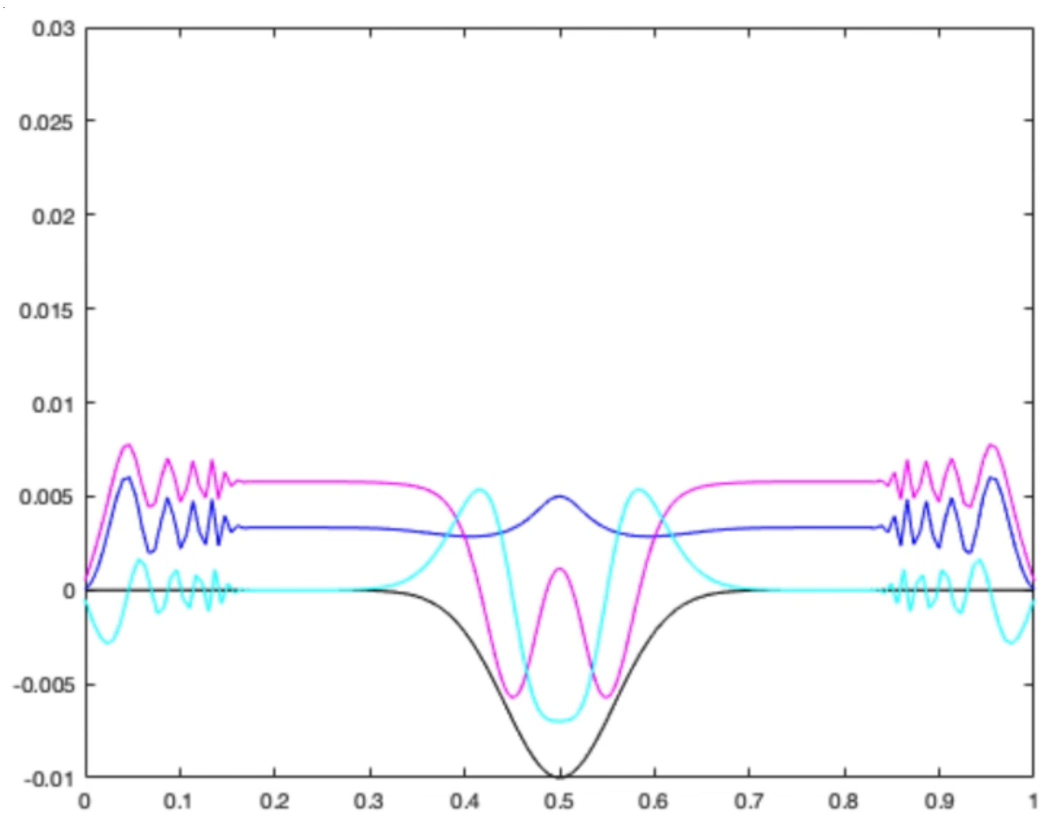


Figure 2: Initial Wavefunction


0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

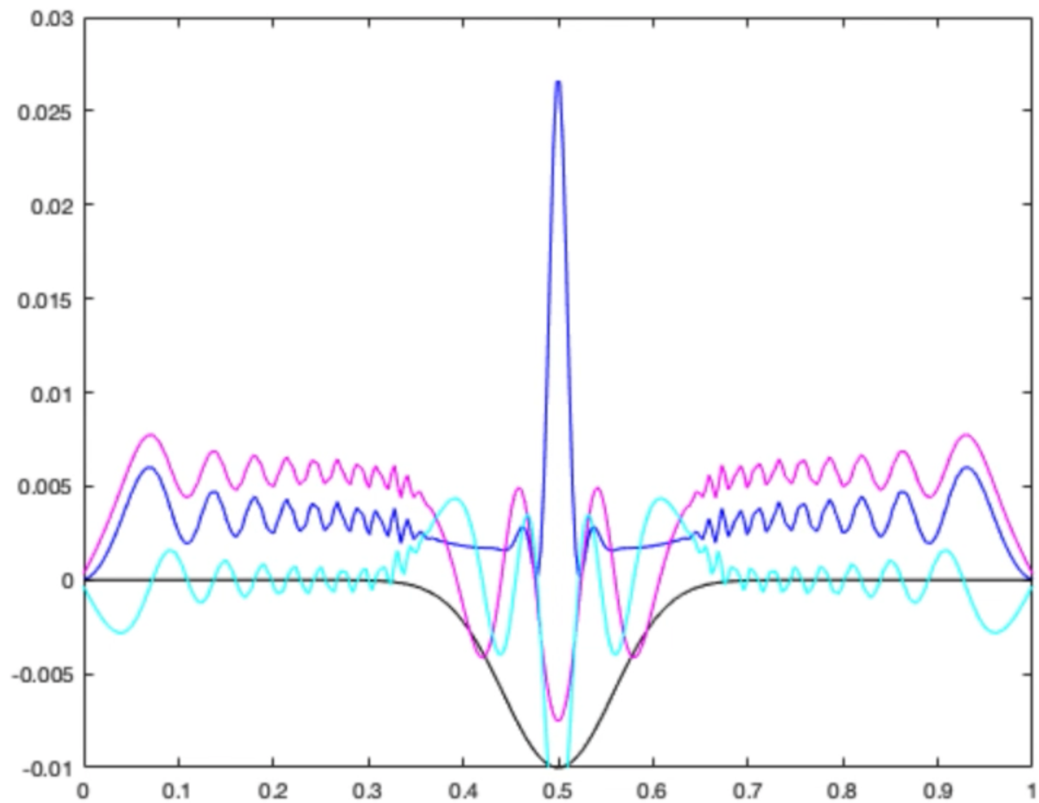


Figure 3: Wavefunction

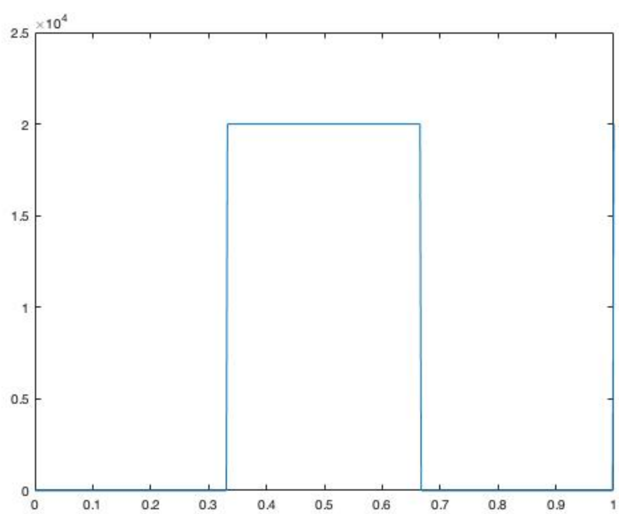
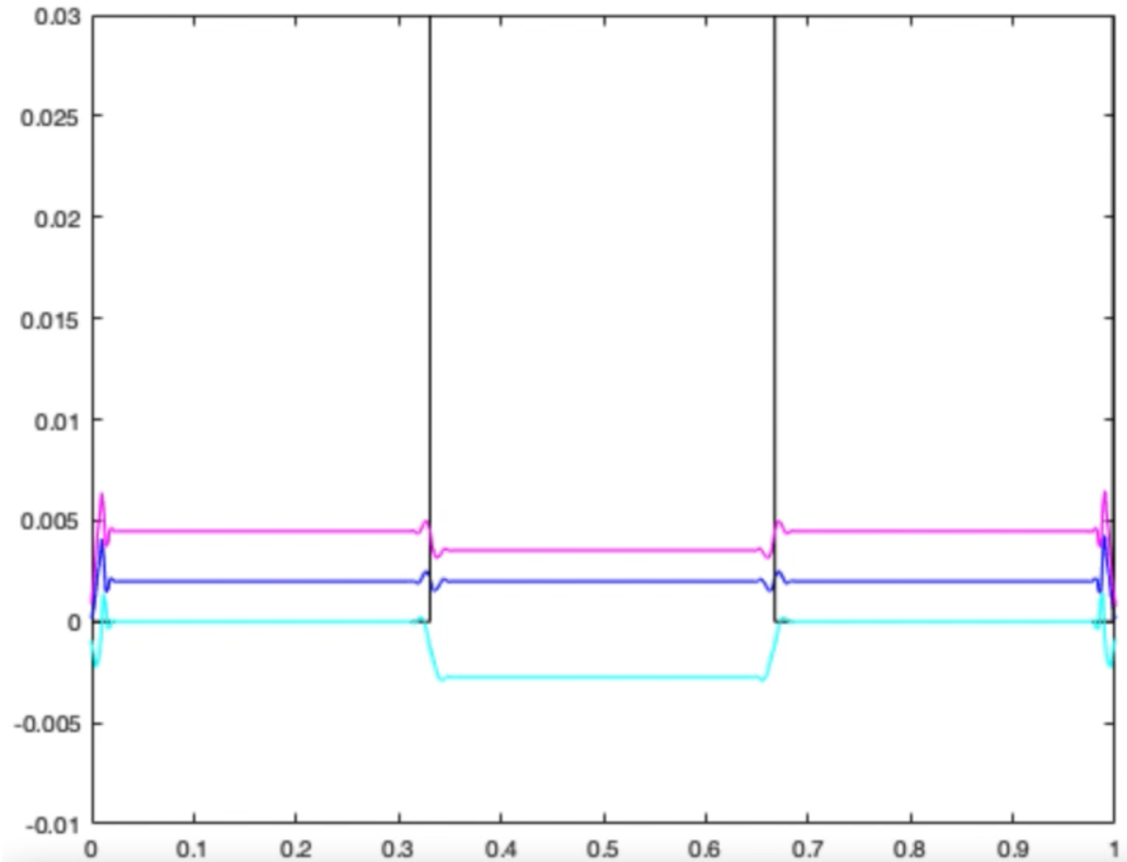
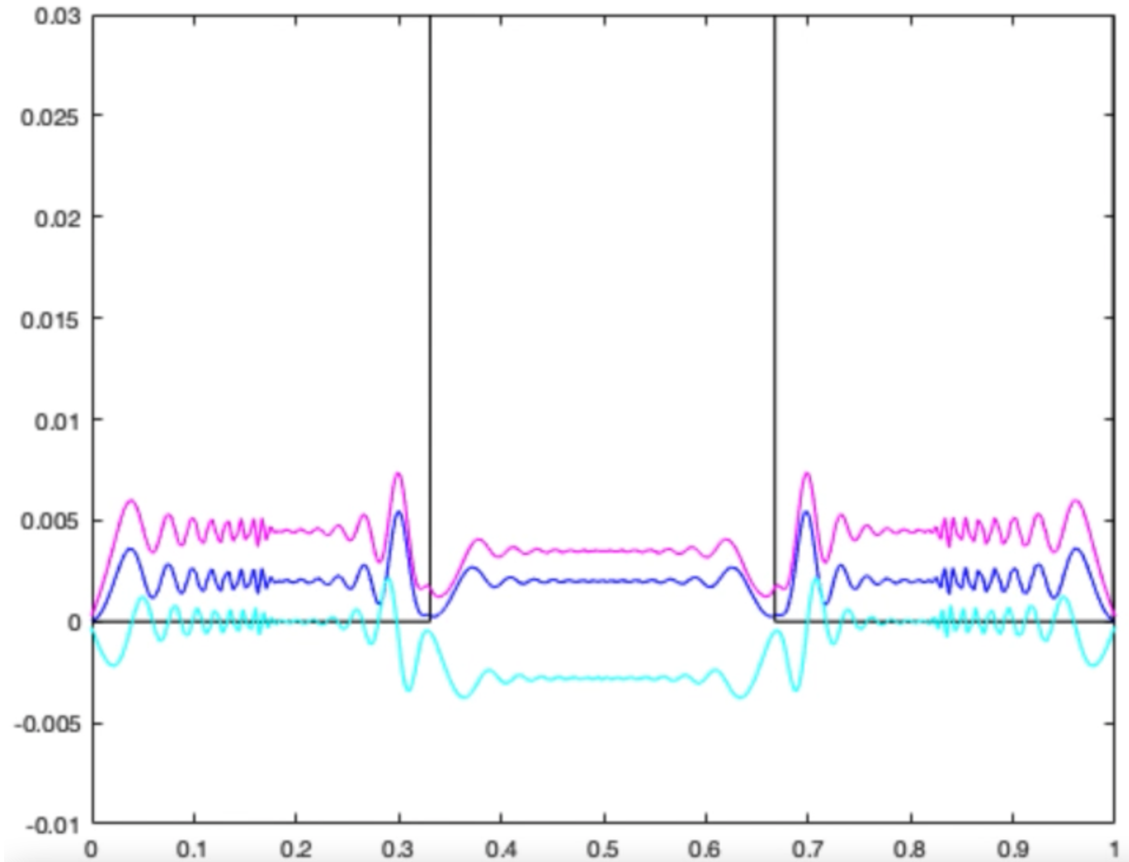


Figure 4: 1D Box Barrier




0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

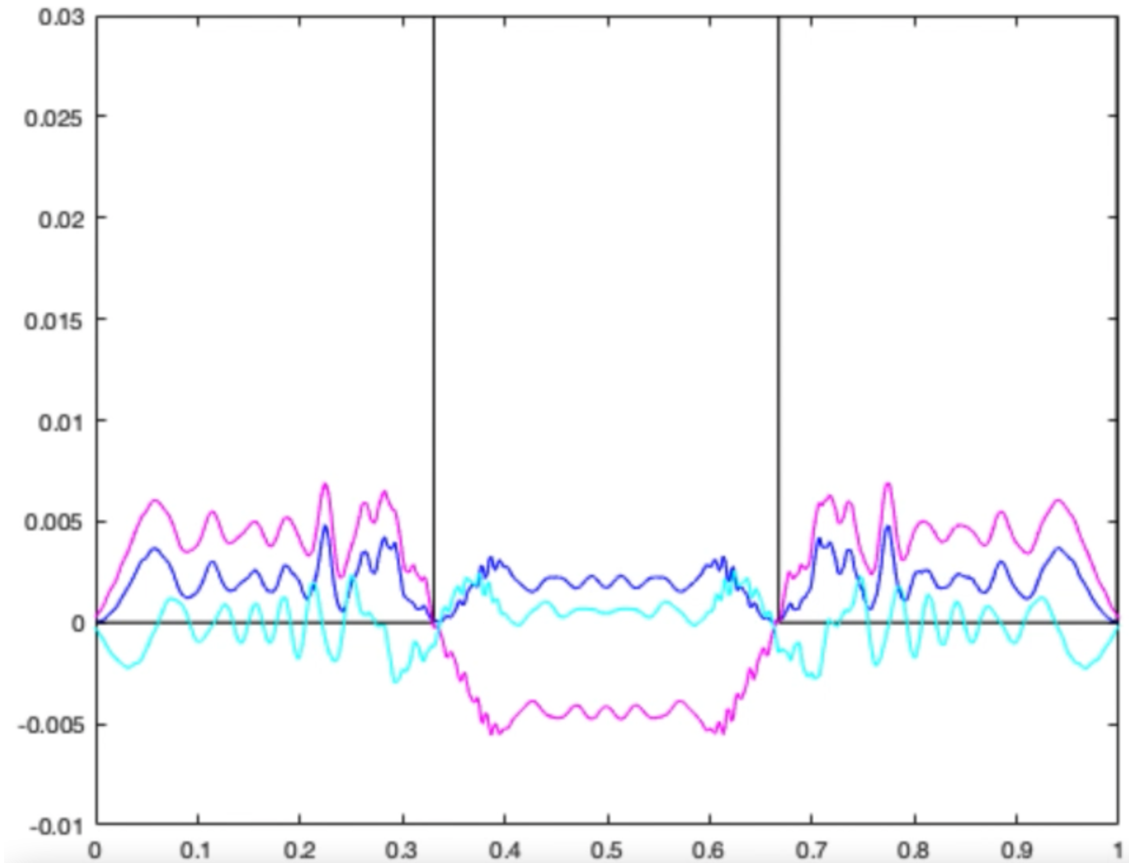


Figure 5: Wavefunction

Two-Dimensional

```
% Constants
hbar = 1; % Planck's
m = 1; % Particle mass

% Domain
L = 1;
N = 100;
Nt = 1e3;
dx = 1/N;
dt = 1/Nt;
x = linspace(0, L, N)';
y = linspace(0, L, N);

% Initial Wavefunction
% u0 = sqrt(2).*(sin(pi*x) + sin(pi*y));
u0 = exp(1i*(4*pi*(x+y) + pi/2));

```

```
% Potential Function
V = 1e4-1e4.*exp(-(x - L/2).^2/(2*(1/50))).*exp(-(y - ...
    L/2).^2/(2*(1/50))); % 2D Gaussian

% V = zeros(N,N); % 2D Square Well
% for j = 1:N/3
%     V(j,:) = 1e7;
%     V(:,j) = 1e7;
%     V(N - j+1,:) = 1e7;
%     V(:,N - j+1) = 1e7;
% end

surf(x,y,V);

% Assembly Matrix
a = dt.*V/(2*hbar);
b = dt*hbar/(4*m*dx*dx);

u0 = reshape(u0, [N*N, 1]);
u = zeros(N*N, Nt);
u(:, 1) = u0;

I = zeros(N, N);
I(1, 1) = 1;
I(N, N) = 1;
A = zeros(N, N);
A(1, 1) = a(1,1) + 4*b;
A(1, 2) = -b;
A(N, N-1) = -b;
A(N, N) = a(N,N) + 4*b;

for j = 2:N-1
    A(j, j-1) = -b;
    A(j, j) = a(j,j) + 2*b;
    A(j, j+1) = -b;

    I(j, j) = 1;
end

% Kronecker Product
B = kron(A, I) + kron(I, A);

% Unitary Matrix
U = (kron(I, I) - 1i.*B)/(kron(I, I) + 1i.*B);

for t = 1:Nt
    u(:, t+1) = U*(u(:, t)./(sqrt(abs(u(:, t)'*u(:, t)))));
end

for t = 1:Nt/6
    u1 = u(:,t);
%     plot(x, real(u(:, t)), x, imag(u(:, t)));
%     plot(x, abs(u(:, t)).^2, 'b', x, V(:)/1e4, 'k', x, real(u(:, t)), ...
    'm', x, imag(u(:, t)), 'c');
    u1 = reshape(u1, [N, N]);
    surf(x, y, abs(u1).^2);
zlim([0 0.05])
    hold on
```

```
%     surf(x, y, V/1e6, 'FaceColor','r', 'FaceAlpha',0.5)
    hold off
    pause(0.01)
end
```

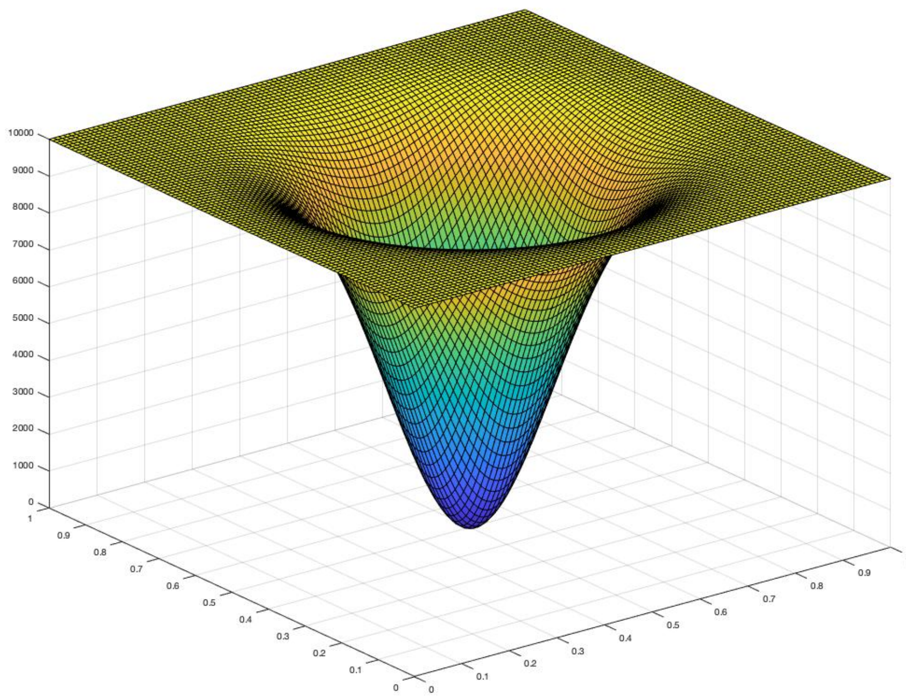
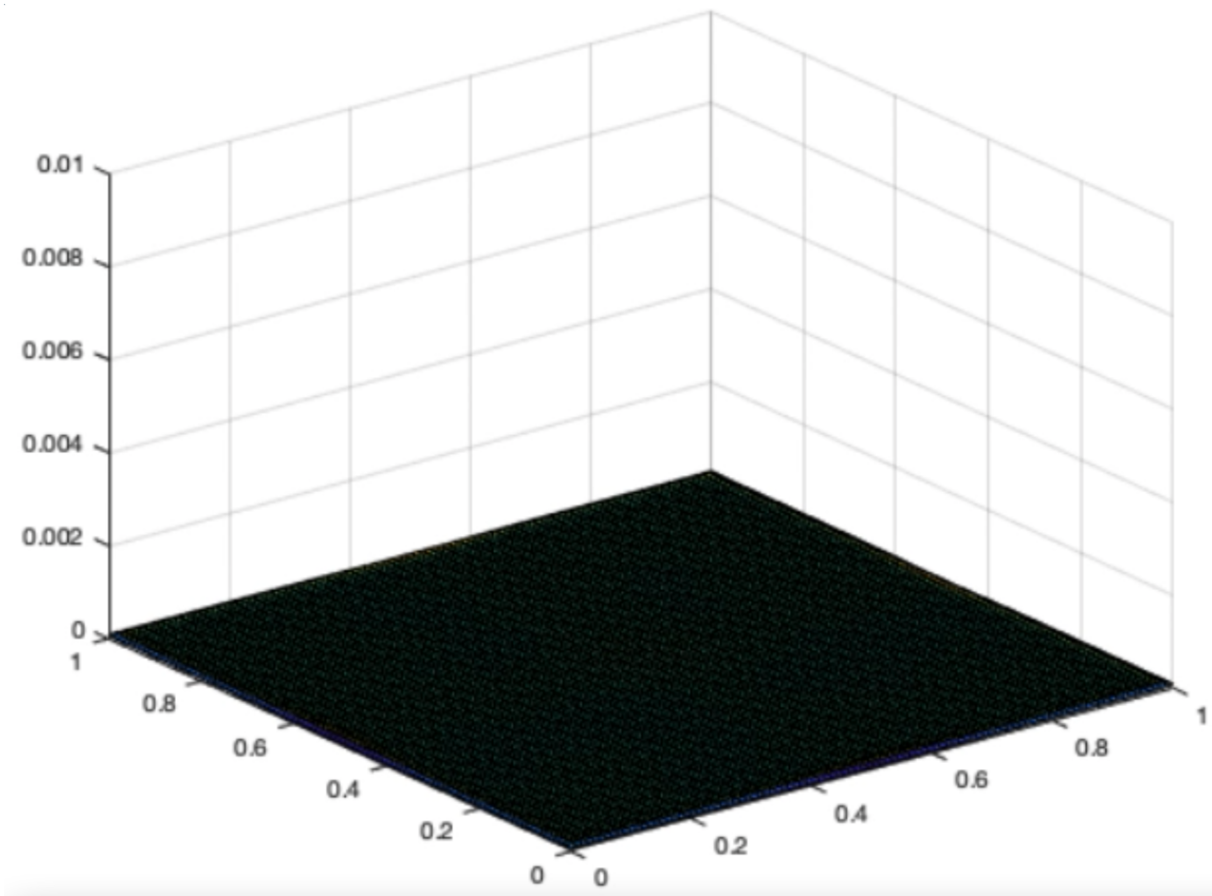
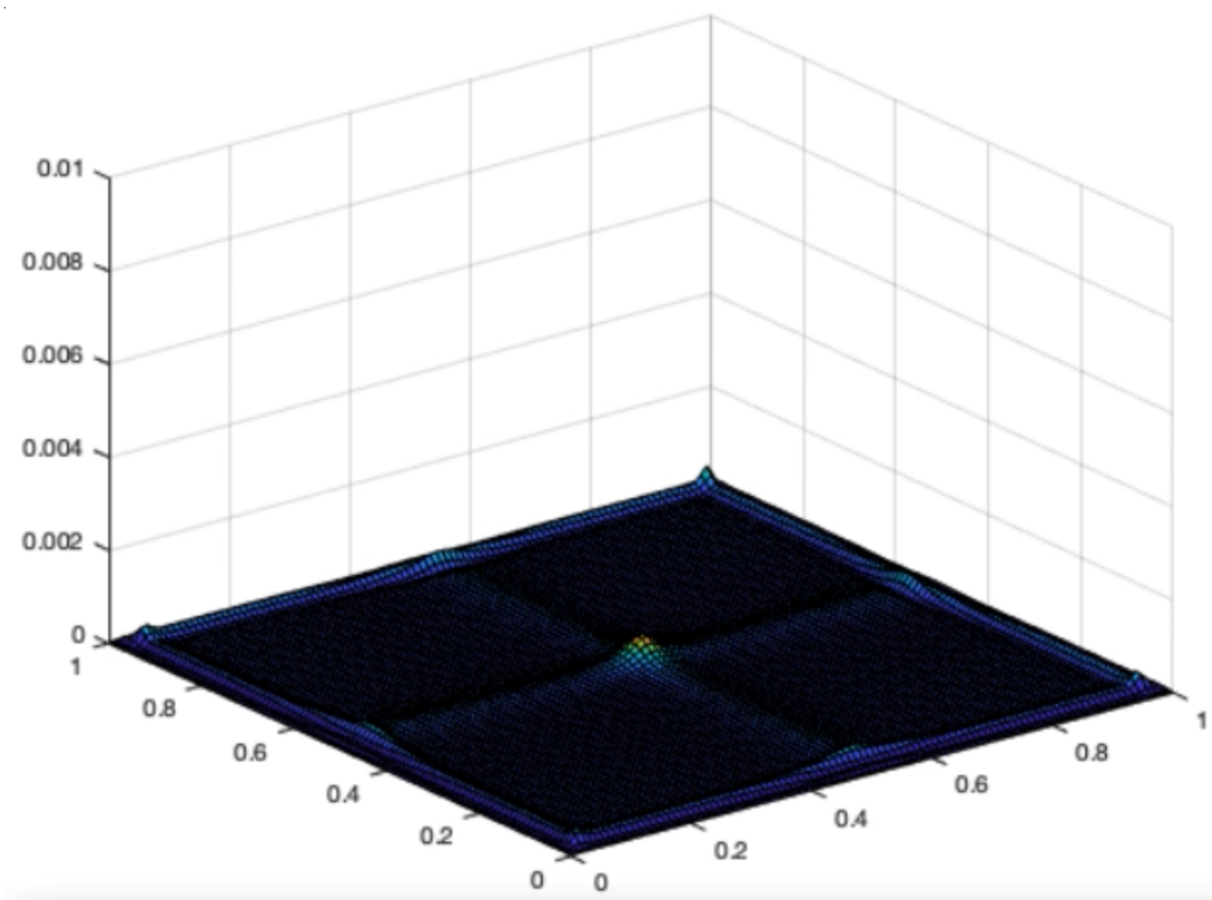
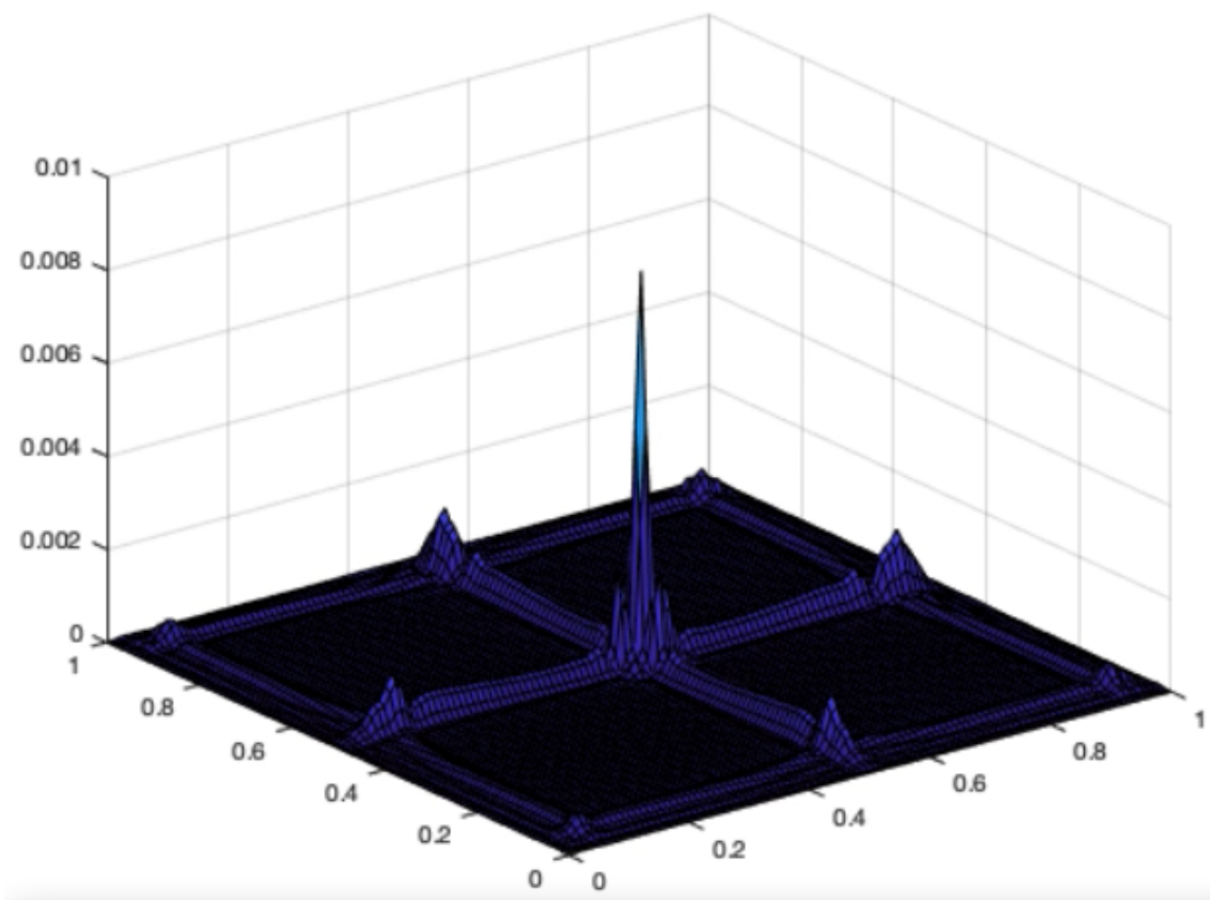


Figure 6: 2D Gaussian Well


0.01
0.008
0.006
0.004
0.002
0
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


0.01
0.008
0.006
0.004
0.002
0
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


0.01
0.008
0.006
0.004
0.002
0
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

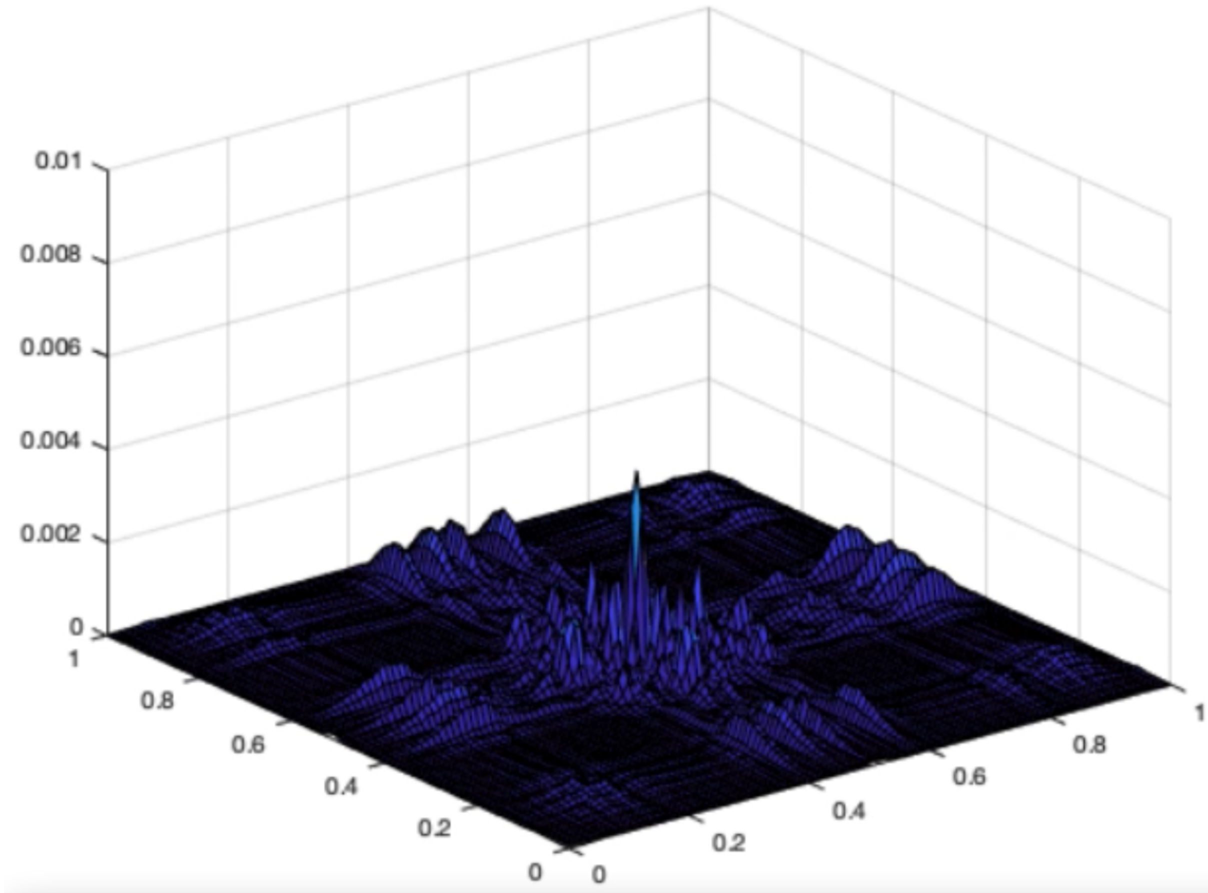


Figure 7: Wavefunction

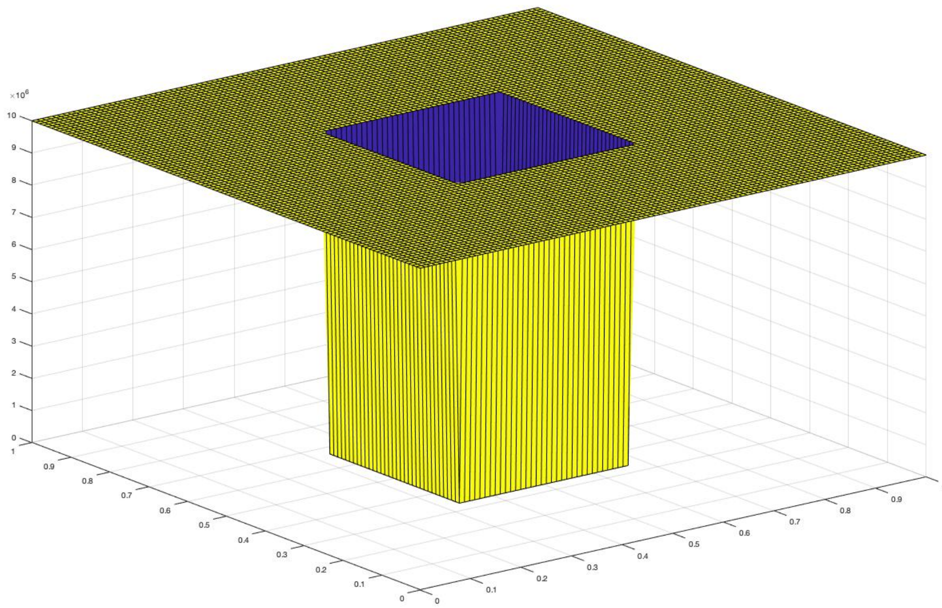
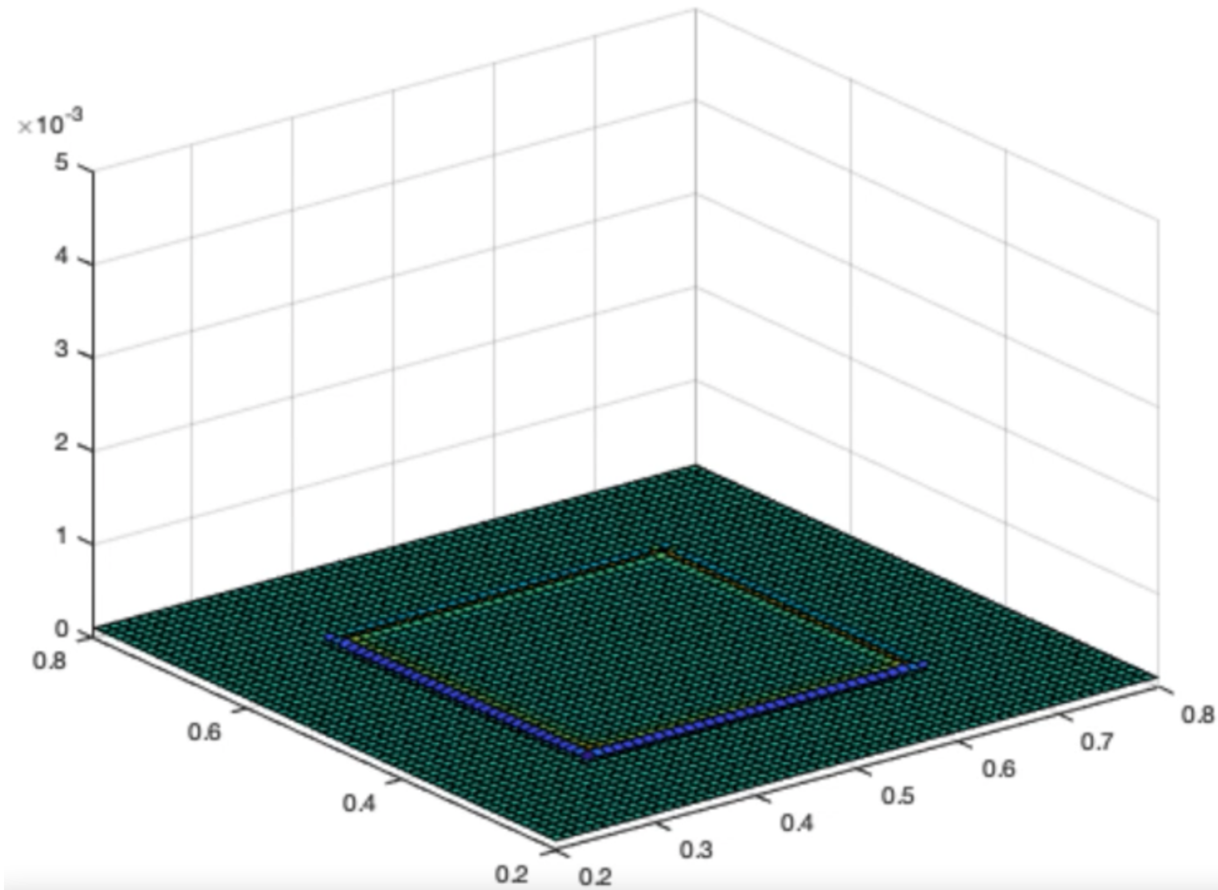
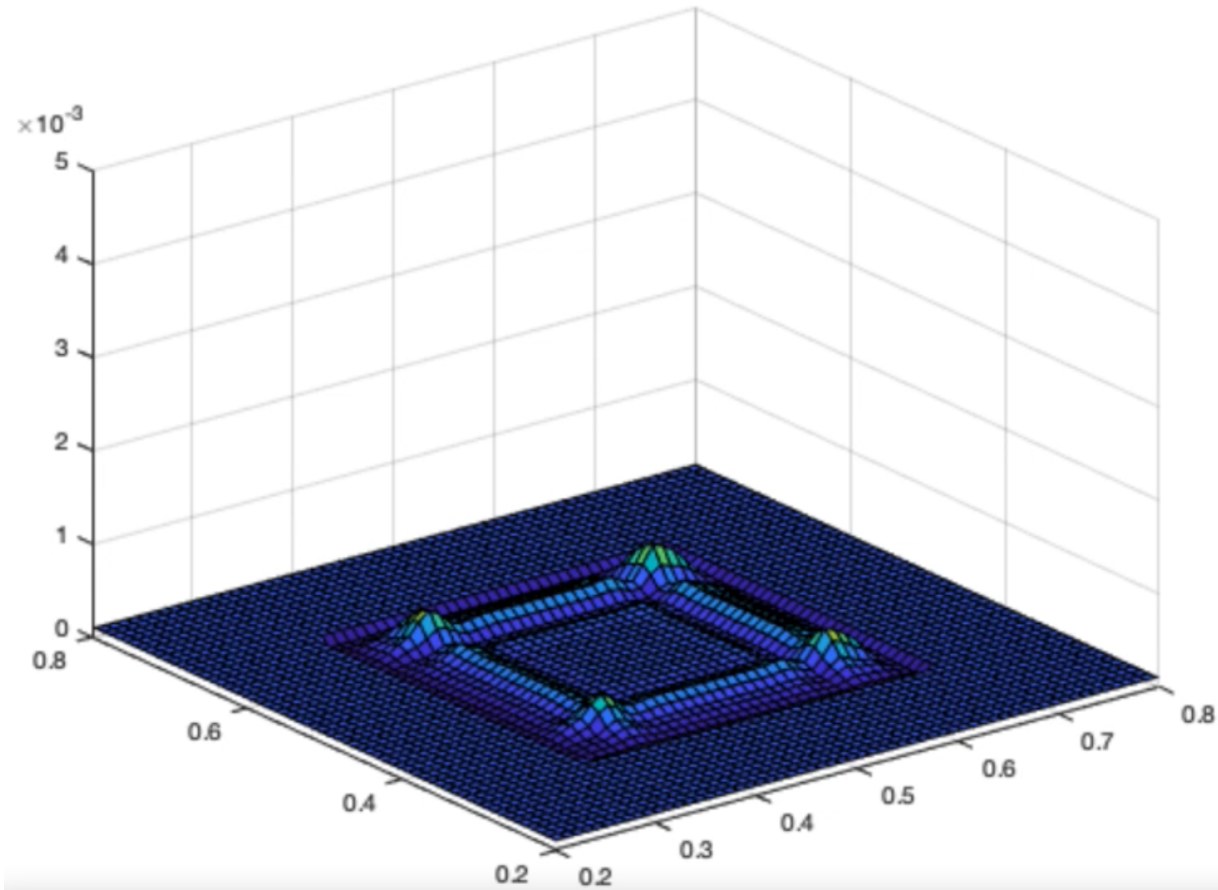


Figure 8: 2D Box Well


×10^-3
5
4
3
2
1
0
0.8
0.6
0.4
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8


×10^-3
5
4
3
2
1
0
0.8
0.6
0.4
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8

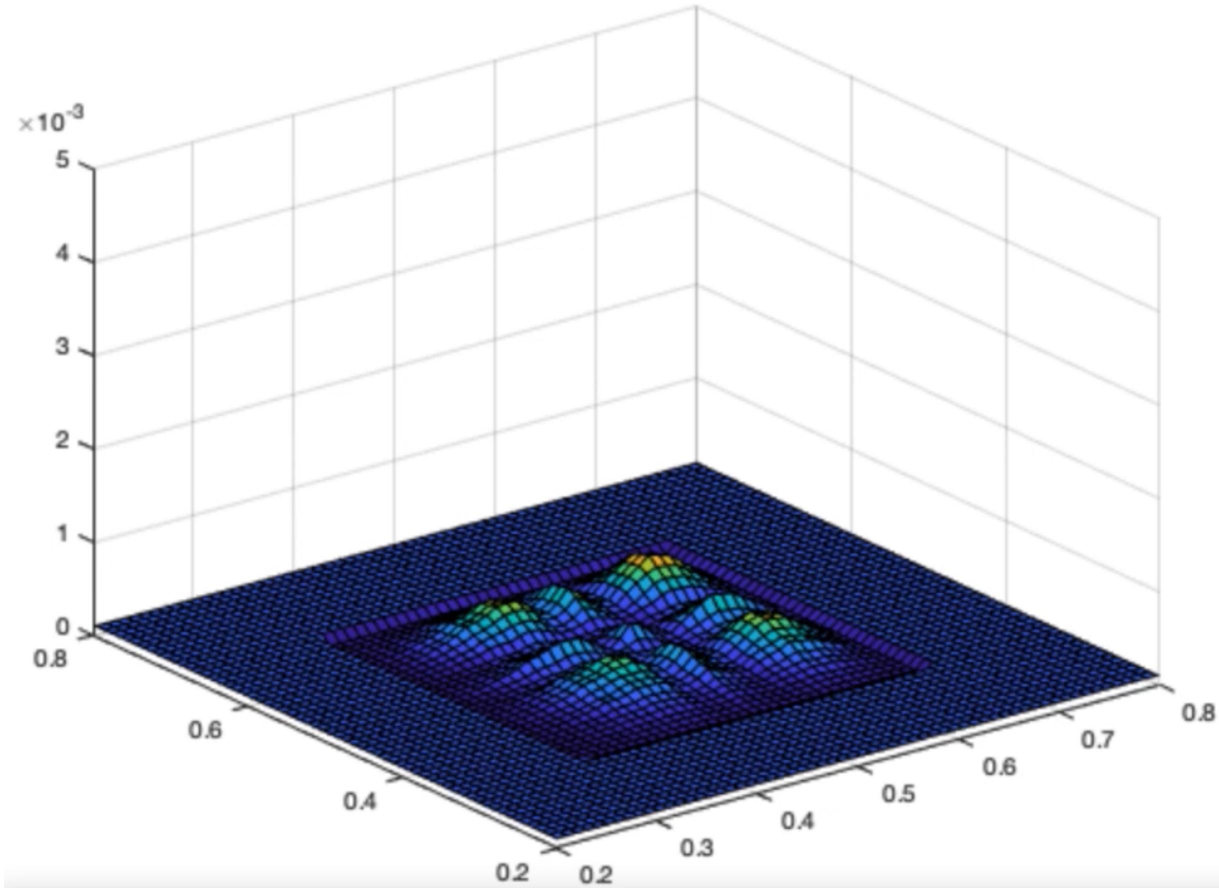


Figure 9: Wavefunction

5 Reference Links

(1) https://asu.instructure.com/courses/107001/files/folder/Lecture%20notes

(2) https://en.wikipedia.org/wiki/Kronecker_sum_of_discrete_Laplacians

(3) Quantum Mechanics Concepts and Applications, Second Edition, Nouredine Zettili