Simulation of the Time-Dependent Schrödinger Equation using Crank-Nicolson and ADI Discretization Methods

Melvin Mathews

December 2, 2020

1 Introduction

In this project the Time-Dependent Schrödinger Equation (TDSE) was investigated. The differential equation describing the TDSE (after non-dimensionalization) is as follows:

$$i\frac{d\psi}{dt} = -\nabla^2\psi + V\psi \tag{1}$$

For the purposes of this project, only the 1 and 2 dimensional TDSEs will be considered so Equation 1 can be reduced into Equations 2 and 3 representing the 1 dimensional TDSE and the 2 dimensional TDSE respectively.

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

$$i\psi(x,y,t)_t = -(\psi_{xx} + \psi_{yy}) + V(x,y,t)\psi \tag{3}$$

The second section of this document describes the Crank-Nicolson (CN) discretization approach being applied to simulate the 1D TDSE (Equation 2) while the third section discusses the application of the Alternating Direction Implicit (ADI) scheme to simulate the 2D TDSE (Equation 3). These sections will also describe the convergence tests applied to verify implementation and several experiments used to characterize the behaviour of the TDSE.

2 One Dimensional TDSE and Crank-Nicolson Discretization

The domain is defined as:

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

with initial conditions and boundary conditions:

$$\psi(x,0) = \psi_0(x) \tag{4}$$

$$\psi(0,t) = \psi(1,t) = 0 \tag{5}$$

2.1 Crank-Nicolson Discretization

The CN discretization method builds on the standard discretized time and spatial relations defined below.

$$\begin{aligned} \lambda &= \frac{\Delta t}{\Delta x} \\ n_x &= 2^l + 1 \\ \Delta x &= 2^{-l} \\ \Delta t &= \lambda \Delta x \\ n_t &= round(t_{max}/\Delta t) + 1 \end{aligned} \tag{6}$$

The CN discretization approach can now be applied to Equations 2, 4 and 5 on this discretized domain.

$$\begin{aligned} i\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = -\frac{1}{2}\left(\frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{\Delta x^2} + \frac{\psi_{j+1}^{n-1} - 2\psi_j^{n-1} + \psi_{j-1}^{n-1}}{\Delta x^2}\right) + \\ \frac{1}{2}V_j^{n+1}\left(\psi_j^{n+1} + \psi_j^n\right), \quad j = 2,3,...,n_x - 1 \quad n = 1,2,...,n_t - 1 \end{aligned} \tag{7}$$

$$\psi_1^{n+1} = \psi_{n_x}^{n+1} = 0, \quad n = 1,2,...,n_t - 1 \tag{8}$$

$$\psi_j^1 = \psi_0(x_j), \quad l = 1,2,...,n_x \tag{9}$$

These relations can now be implemented in code.

2.2 Implementation and Relevant Files

2.2.1 Implementation

The implementation closely follows the theory discussed in the previous subsection. The domain is first discretized and the selection of the appropriate initial data type and potential shape is made. The implemented initial data types are the exact family of solutions seen in Equation 10 and the boosted Gaussian family of solutions seen in Equation 11.

$$\psi(x,0) = sin(m\pi x) \tag{10}$$

where m is parameterized.

$$\psi(x,0) = e^{ipx} e^{-((x-x_0)/\gamma)^2} \tag{11}$$

where x_0, γ, and p are parameterized. The potential shape options are either no potential:

$$V(x) = 0 \tag{12}$$

or a rectangular barrier well:

$$V(x) = \begin{cases} 0 & for \quad x < x_{min} \\ V_c & for \quad x_{min} \leq x \leq x_{max} \\ 0 & for \quad x > x_{max} \end{cases} \tag{13}$$

with x_{min}, x_{max}, and V_c parameterized. Next, a tridiagonal system is setup and solved. Equation 7 can be rearranged to form a relation purely relating a future time step to a previous time step of ψ. This can then be used to generate a tridiagonal system as is seen below. The notation used matches the notation of the code to some degree.

$$A\psi_j^{n+1} = B\psi_j^n \tag{14}$$

$$A = \begin{pmatrix} d_1 & -\alpha & & 0 \\ -\alpha & \ddots & \ddots & \\ & \ddots & \ddots & -\alpha \\ 0 & & -\alpha & d_1 \end{pmatrix}, \quad B = \begin{pmatrix} d_2 & \alpha & & 0 \\ \alpha & \ddots & \ddots & \\ & \ddots & \ddots & \alpha \\ 0 & & \alpha & d_2 \end{pmatrix} \tag{15}$$

Where α, d_1, and d_2 can be defined as:

$$\alpha = \frac{i\Delta t}{2\Delta x^2}, \quad d_1 = 1 + 2\alpha + V_j, \quad d_1 = 1 - 2\alpha + V_j \tag{16}$$

Left division is then used to solve this system for the future time step of ψ.

2.2.2 Relevant Files

- `sch_1d_cn.m`: contains the implementation of the CN FDA for the 1D TDSE
- `convergence_testing_exact.m`: contains convergence testing and error values of the CN FDA for the 1D TDSE initialized with the exact family of solutions
- `convergence_testing_boosted.m`: contains convergence testing of the CN FDA for the 1D TDSE initialized with the Boosted Gaussian family of solutions
- `experiment_well.m`: contains behaviour characterization of the CN FDA for the 1D TDSE as the well depth potential is increased.

- `experiment_barrier.m`: contains behaviour characterization of the CN FDA for the 1D TDSE as the barrier height potential is increased.

2.3 Convergence Testing

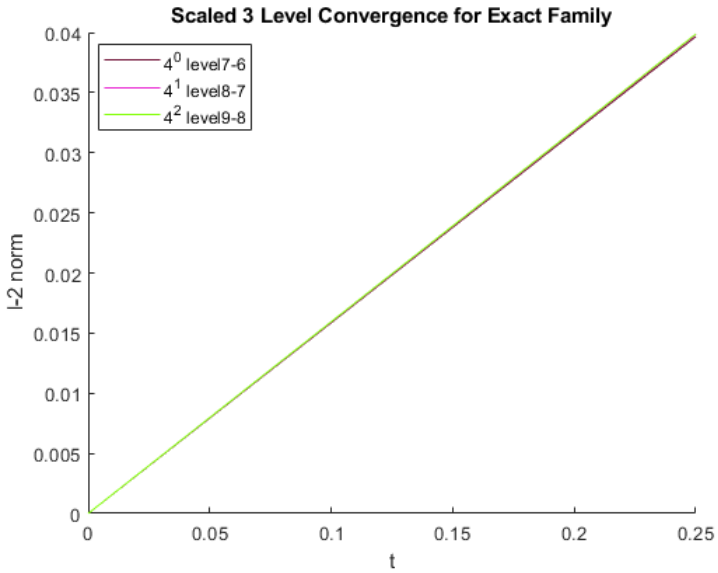
Convergence tests were performed for both the exact and boosted Gaussian families of solutions, both with no potential defined. Levels 6, 7, 8, and 9 were used to test for convergence. The l-2 norm, or root mean square (RMS) value, was computed for the difference between consecutive input levels to reduce the dimensional complexity of the problem.

$$||d\psi^l||_2(t^n) = ||\psi^{l+1} - \psi^l||_2(t^n) \tag{17}$$

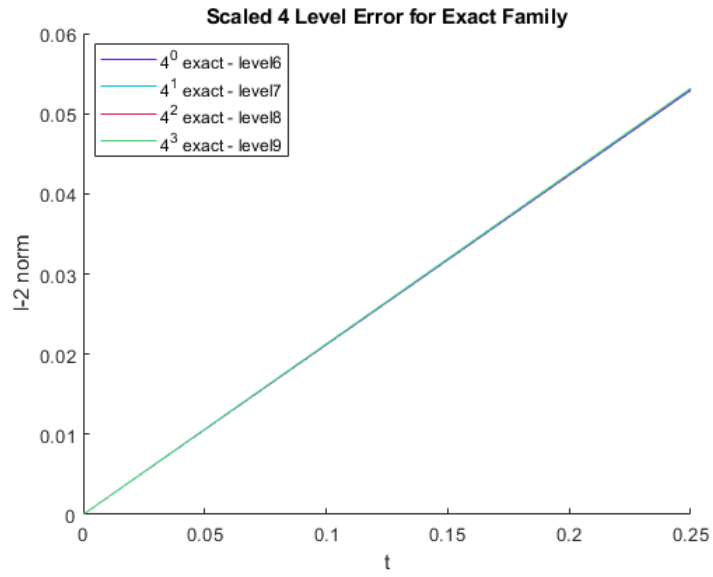
For the exact family of solutions the error was calculated in a similar way:

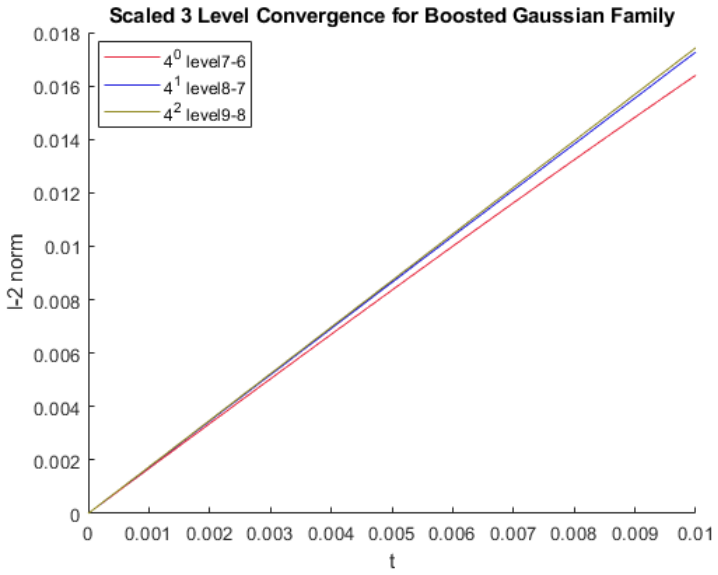
$$||E(\psi^l)||_2(t^n) = ||\psi_{exact} - \psi^l||_2(t^n) \tag{18}$$

As the CN FDA is $O(h^2)$ accurate, a factor n^2 increase in discretization can be accounted for by a 4^n decrease in error. The following plots illustrate this property was achieved. Specific parameters for the generations of these plots can be found in the appropriate files.



It is important to observe the near coincident linear lines in all three plots. This provides us with confidence in the correctness of our implementation. The linear nature of the plots can be attributed to the fact that the l-2 norm is spatially independent and error increases as the FDA is performed away from initial conditions. It is also interesting to see that the 3 level convergence for the





boosted Gaussian family of solutions has markedly worse solution error when compared to the exact family of solutions as it is plotted for less than half the time interval.

2.4 Numerical Experiments

The fraction of time a given quantum particle spends in a spatial interval $x_1 < x < x_2$ is given by:

$$\bar{P}(x_2) - \bar{P}(x_1) \tag{19}$$

Where $\bar{P}$ represents the normalized cumulative probability function. The non-normalized version of this is an output of the `sch_1d_cn` function. To gain a more meaningful value, this expression can be divided by the interval to get F_e, a function describing the *excess* fractional probability for a particle to remain in an interval relative to a completely free particle.

$$F_e(x_1, x_2) = \frac{P(x_2 - P(x_1)}{x_2 - x_1} \tag{20}$$

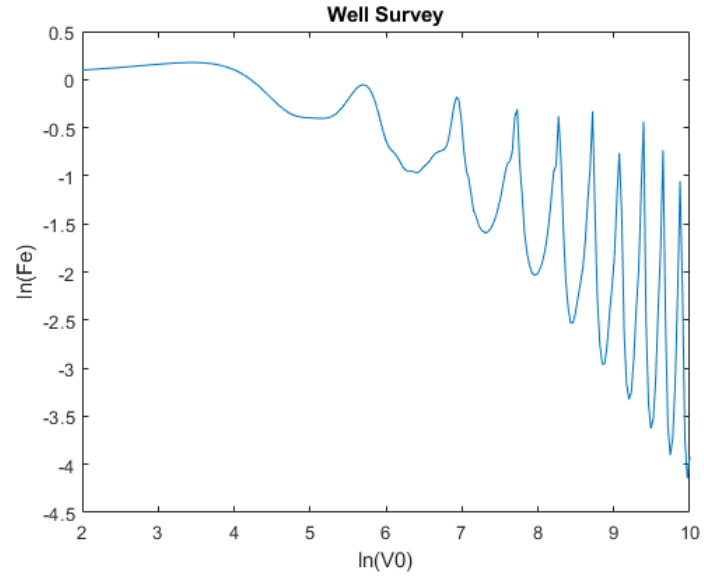
Two experiments were conducted using this value. The first of which describes particle behaviour around a potential barrier while the second describes particle behaviour around a potential well. For both experiments, $ln(F_e)$ was observed with respect to $ln(V_0)$ which represented the height or depth of the given potential.

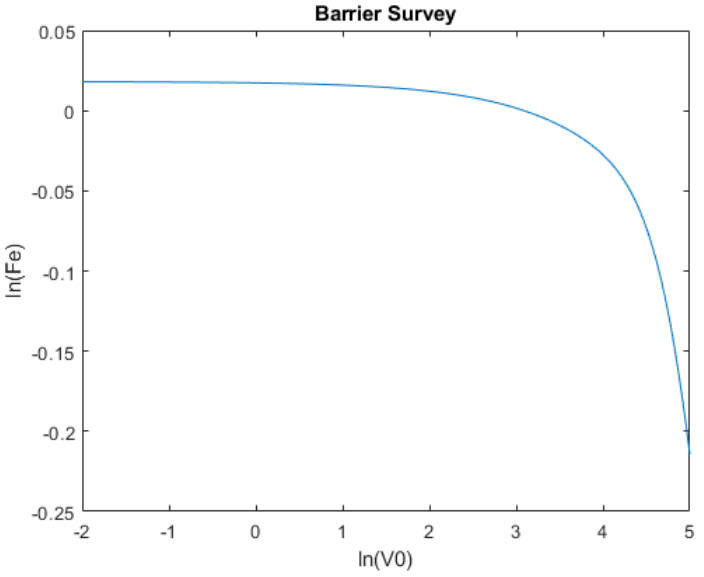
2.4.1 Well Survey

For this experiment a potential well ($V_c \leq 0$) was constructed around the region $0.6 \leq x \leq 0.8$ and the fractional probability as per Equation 20 was calculated for the region within this barrier. The graph below was generated as the depth of this barrier was swept. It can be observed that for a shallow well, the particle acts much like a free particle but then exhibits a periodic decrease in fractional probability as the depth of the well is increased. This periodic behaviour can be attributed to certain well depths causes resonance and causes full transmission out of the well. At these heights, a particle would be more likely to exit the well than remain within it. The low points on the graph below correspond to such well heights.

2.4.2 Barrier Survey

For this experiment, a potential barrier ($V_c \geq 0$) was constructed around the region $0.6 \leq x \leq 0.8$ and the fractional probability as per Equation 20 was calculated for the region $0.8 \leq x \leq 1.0$. The boosted Gaussian was used as the initial state of the particle with a x_0 value of 0.4; "before" the barrier. The graph below was generated as the height of this barrier was swept. It can be observed that the particle spends approximately the same amount of time as a free particle when the height of the barrier is low but then this fractional probability exponentially decays as the barrier height is increased. This is expected as it describes how it will be more "difficult" for a particle to overcome or tunnel through a higher barrier.





3 Two Dimensional TDSE and ADI Discretization

The domain is defined as:

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

with initial conditions and boundary conditions:

$$\psi(x, y, 0) = \psi_0(x, y) \tag{21}$$

$$\psi(0, y, t) = \psi(1, y, t) = \psi(x, 0, t) = \psi(x, 1, t) = 0 \tag{22}$$

3.1 Alternating Direction Implicit Scheme

For this scheme, we are once again required to discretize the domain as in Equation 4. The only difference being that there is now a second dimension y that is discretized in the same way as x. The ADI scheme can now be applied and is defined as:

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{xx}\right)\psi^{n+\frac{1}{2}}_{i,j} = \left(1 + i\frac{\Delta t}{2}\partial^h_{xx}\right)\left(1 + i\frac{\Delta t}{2}\partial^h_{yy} - i\frac{\Delta t}{2}V_{i,j}\right)\psi^n_{i,j} \tag{23}$$

$$\left(1 - i\frac{\Delta t}{2}\partial^h_{yy} + i\frac{\Delta t}{2}V_{i,j}\right)\psi^{n+1}_{i,j} = \psi^{n+\frac{1}{2}}_{i,j} \tag{24}$$

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

With the partial differential operators defined as:

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

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

The key advantage of this approach is that it allows for the solution of a multi-dimensional PDE by considering each dimension independently.

3.2 Relevant Files and Implementation

3.2.1 Implementation

The implementation closely follows the theory discussed in the previous subsection. The domain is first discretized and the selection of the appropriate initial data type and potential shape is made. The implemented initial data types are the exact family of solutions seen in Equation 25 and the boosted Gaussian family of solutions seen in Equation 26.

$$\psi(x, y, 0) = \sin(m_x\pi x)\sin(m_y\pi y) \tag{25}$$

where m_x and m_y are parameterized.

$$\psi(x, y, 0) = e^{ip_x x}e^{ip_y y}e^{-((x-x_0)/\gamma_x)^2}e^{-((y-y_0)/\gamma_y)^2} \tag{26}$$

where x_0, γ, and p are parameterized. The potential shape options are either no potential:

$$V(x, y) = 0 \tag{27}$$

a rectangular barrier well:

$$V(x, y) = \begin{cases} V_c & for \quad x_{min} \leq x \leq x_{max} \; and \; y_{min} \leq y \leq y_{max} \\ 0 & otherwise \end{cases} \tag{28}$$

with x_{min}, x_{max}, y_{min}, y_{max}, and V_c parameterized. A third potential was defined to simulate a double slit. The specifications for this potential can be observed in the code. Next, Equation 23 is broken down into 3 operators. The first operator is defined by the brackets surrounding the partial differential operator on y. The second is the partial differential operator on x on the right hand side of the equation. The third is the partial differential operator on x on the left hand side of the equation. This left hand side operator can be converted into a tridiagonal matrix exactly as shown in section 2.2.1. The right hand side requires some simplification before the entire system (Equation 23) can be solved. To do this, a sequential approach is taken in which the partial differential operator on y is first calculated on the current ψ. Next, the entire first operator is computed, followed by the second operator applied to this result. The final result is a column vector that can be used to left-divide by the right hand side operator to give the half time step increment. Now Equation 24 can be solved by left dividing this half time step ψ with the left hand side operator in Equation 24. This results in the full time step increment of ψ.

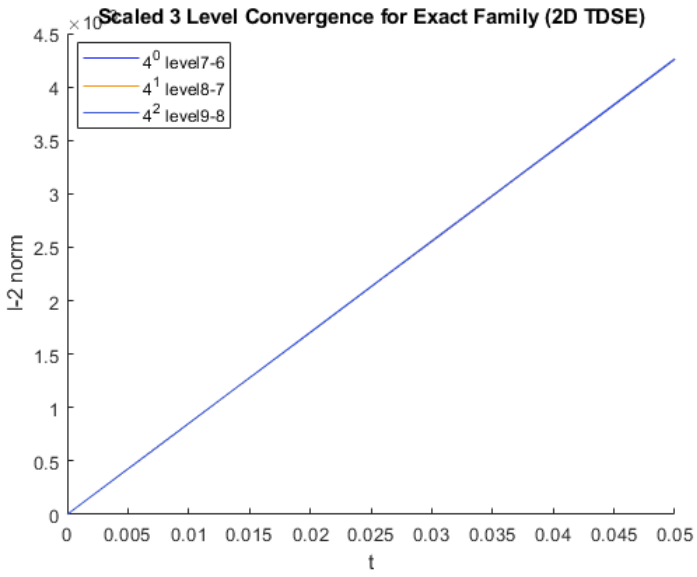
3.2.2 Relevant Files

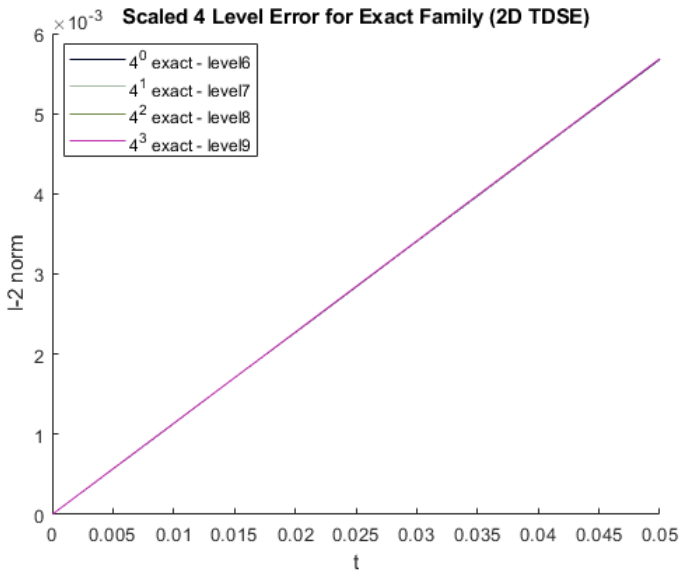
- `sch_2d_adi.m`: contains the implementation of the ADI FDA for the 2D TDSE
- `convergence_testing_exact.m`: contains convergence testing and error values of the ADI FDA for the 2D TDSE initialized with the exact family of solutions
- `convergence_testing_boosted.m`: contains convergence testing of the ADI FDA for the 2D TDSE initialized with the Boosted Gaussian family of solutions
- `experiment_well.m`: contains behaviour characterization of the ADI FDA for the 2D TDSE interacting with a well.
- `experiment_barrier.m`: contains behaviour characterization of the ADI FDA for the 2D TDSE interacting with a barrier.
- `experiment_double_slit.m`: contains behaviour characterization of the ADI FDA for the 2D TDSE passing through a double slit.

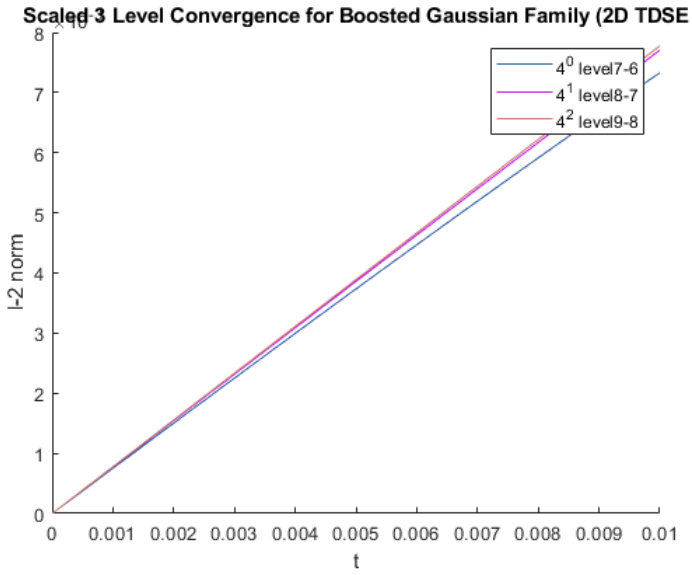
3.3 Convergence Testing

Convergence tests were performed for both the exact and boosted Gaussian families of solutions, both with no potential defined. Levels 6, 7, 8, and 9 were used to test for convergence. The l-2 norm, or root mean square (RMS) value, was computed in much the same way as defined in section 2.3 with the key difference being that the RMS was calculated in 2 dimensions rather than 1.

As the CN FDA is $O(h^2)$ accurate, a factor n^2 increase in discretization can be accounted for by a 4^n decrease in error. The following plots illustrate this property was achieved. Specific parameters for the generations of these plots can be found in the appropriate files.







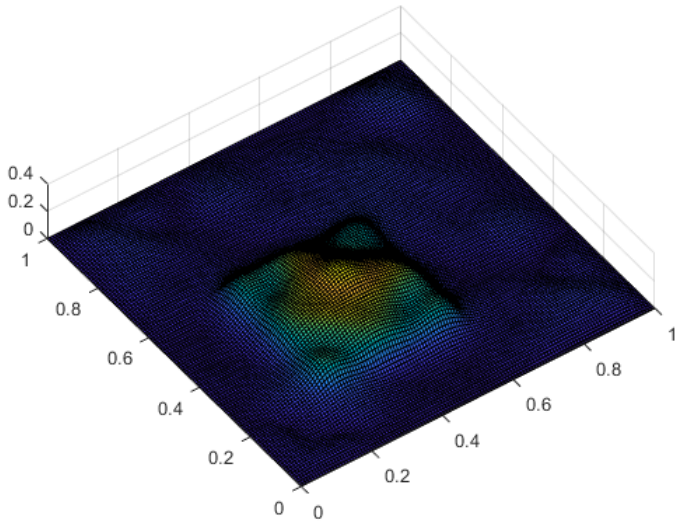
It is important to observe that all the observations made of the 1D TDSE convergence tests can be made here as well. This provides us with confidence in the correctness of our implementation.

3.4 Numerical Experiments

A series of experiments were performed with the 2D TDSE simulation developed with the ADI Method. The exact parameters for each test can be found in the respective files. Different parameters to the Boosted Gaussian were modified to observe desired behaviour.

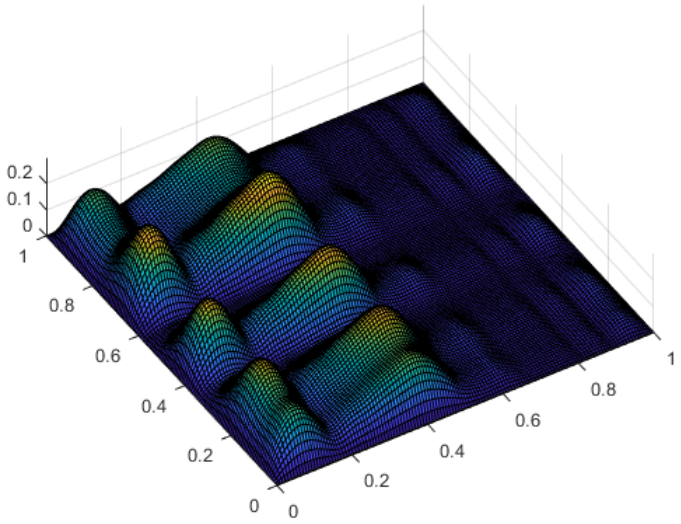
3.4.1 Well Behaviour

In this experiment we observed the behaviour of the 2D TDSE with a well. The relevant files are `experiment_well.m` and `experiment_caught_in_well.m` and a clearer description of the behaviour can be found in the respective .avi file in the results folder of this submission. It is clear to see that the wave function is confined to the well. The caught in well file explores the behaviour of the TDSE originating outside of the well while the image below describes the TDSE originating inside the well and is generated by the `experiment_well.m` file.



3.4.2 Barrier Behaviour

In this experiment we observed the behaviour of the 2D TDSE with a barrier. The relevant files are `experiment_barrier.m` and `experiment_on_barrier.m` and a clearer description of the behaviour can be found in the respective .avi file in the results folder of this submission. It is clear to see that the wave function does not tunnel over a sufficiently high barrier. The on barrier file explores the behaviour of the TDSE originating on top of the barrier while the image below describes the TDSE originating away from the barrier and is generated by the `experiment_barrier.m` file.



3.4.3 Double Slit Behaviour

In this experiment we observed the behaviour of the 2D TDSE with a double slit. It is clear to see the areas of destructive and constructive behaviour that might be expected. A clearer description of the behaviour can be found in the respective .avi file in the results folder of this submission and is generated by the `experiment_double_slit.m` file.