Simulation of Electron 2D Double Slit Experiment Using Alternating Direction Implicit Matrix Method on the Schrodinger Equation

[redacted]

(Dated: June 4, 2020)

In this paper, I describe and implement a numerical algorithm to solve the time-dependent Schrodinger equation in 2 dimensions. I use an alternating direction implicit method to simulate a Gaussian electron wavepacket in a box, incident on a screen with two slits. I show that the simulation matches theoretical predictions of the classical photon double slit experiment.

The double slit experiment is a canonical, well-known setup for the exploration of quantum particle-wave dualistic effects. In a common version, an electron beam fired at a sheet with two slits produces an interference pattern on a screen behind, demonstrating that the electron has wavelike properties. Quantum mechanical formalism tells us that this wavelike property can be encapsulated by describing the electron as a wavefunction, governed by the time evolution of the Schrodinger equation. The probability density function is given by the amplitude squared of the wavefunction.

In this paper, I describe the implementation of a computational simulation of the double slit experiment. I use a finite difference method with appropriate boundary and initial conditions, to evolve the 2 dimensional Schrodinger equation forwards in time. I sum the probability density of the electron on the screen over time, to display the interference bands. I show that these are consistent both with analytical computation and experimental results.

I. BACKGROUND

The double slit experiment was first performed by Thomas Young in 1801, demonstrating the wavelike properties of light. [4] A beam of light incident on two slits produced an interference pattern of fringes, rather than two lines in line with the slits.

In 1924, de Broglie's theory of matter waves argued that all particles behaved like waves. In 1927, Davisson and Germer used electrons in the double slit experiment to demonstrate this wavelike nature of matter. [4]

A schematic of the experimental setup is shown in Figure 1. Similar to the case with light, an electron beam produces an interference pattern on a screen, when passing through two slits.

[redacted]

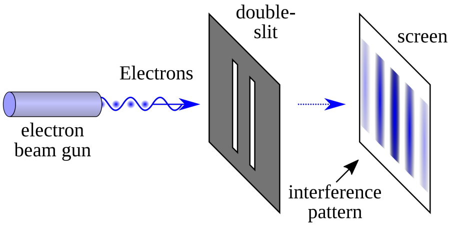

Figure 1: A schematic of the double slit experiment performed with an electron beam. [5]

II. ANALYTICAL SOLUTION

Treating the electron as a classical wave with wavelength λ_e, we can get an approximate solution to the interference pattern produced on the screen. We consider the approximation of the Fraunhofer conditions, [2] which assumes that the incident light is a plane wave.

Denoting the distance from the slits to the screen as L, the distance between slits as d and the slit width as s, the intensity as a function of angle away from normal θ, is proportional as

$$I(\theta) \propto \cos^2\left[\frac{\pi d \sin\theta}{\lambda_e}\right] \ \mathrm{sinc}^2\left[\frac{\pi s \sin\theta}{\lambda_e}\right] \quad .$$

Using De Broglie's equation to find the electron wavelength, we find that

$$\lambda_e = \frac{h}{m_e v_e}.$$

Using these expressions, and the parameters of the experiment, we can graph the light intensity on the screen.

III. PREPARATION OF COMPUTATIONAL ALGORITHM

A. Finite Difference Equation

The fundamental dynamical equation of quantum mechanics is the Schrodinger equation. It governs the time evolution of a wavefunction in a potential. In two spatial dimensions, the equation is

$$i\hbar\frac{\partial}{\partial t}\Psi(x,y,t) = \frac{-\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right) + V(x,y,t)\Psi(x,y,t). \tag{1}$$

Naively, one might expect that this equation can be discretized and solved by Euler stepping forwards in time. The standard discretization, using the centered second-order difference for the spatial derivatives, is

$$\begin{aligned} i\hbar\frac{\psi^{n+1}_{i,j}-\psi^{n}_{i,j}}{\Delta t} = \frac{-\hbar^2}{2m}&\left[\frac{\psi^{n}_{i+1,j}+\psi^{n}_{i-1,j}-2\psi^{n}_{i,j}}{(\Delta x)^2}\right. \\ &\left.+\frac{\psi^{n}_{i,j+1}+\psi^{n}_{i,j-1}-2\psi^{n}_{i,j}}{(\Delta y)^2}\right] + V_{i,j}\psi^{n}_{i,j}. \end{aligned} \tag{2}$$

However, such a method is unstable, requiring very small time increments to yield accurate results. CITE

A modification which is more stable is Alternating Direction Implicit (ADI) method. In this method, at alternating steps, one of the spatial derivatives is evaluated at the "next" time step, rather than the current time step. [1][6] In other words, the finite difference approximation for the x derivative for the first step is

$$\frac{\partial^2}{\partial x^2}\psi \approx \frac{\psi^{n+1}_{i+1,j}+\psi^{n+1}_{i,j}-2\psi^{n+1}_{i,j}}{\Delta x}$$

while the finite difference approximation for the y derivative remains evaluated at the current time step,

$$\frac{\partial^2}{\partial y^2}\psi \approx \frac{\psi^{n}_{i,j+1}+\psi^{n}_{i,j-1}-2\psi^{n}_{i,j}}{\Delta x}.$$

Then, at the next step, the y derivative is evaluated at the next time step, and the x derivative is evaluated at the current time step. This alternating patterns repeats for the entirety of the simulation.

The finite difference equation for even time steps is

$$\begin{aligned} i\hbar\frac{\psi^{n+1}_{i,j}-\psi^{n}_{i,j}}{\Delta t} = \frac{-\hbar^2}{2m}&\left[\frac{\psi^{n+1}_{i+1,j}+\psi^{n+1}_{i-1,j}-2\psi^{n+1}_{i,j}}{(\Delta x)^2}\right. \\ &\left.+\frac{\psi^{n}_{i,j+1}+\psi^{n}_{i,j-1}-2\psi^{n}_{i,j}}{(\Delta y)^2}\right] \\ &+\left[V_{i,j}\psi^{n}_{i,j}+V_{i,j}\psi^{n+1}_{i,j}\right] \end{aligned} \tag{3}$$

whereas for odd time steps it is

$$\begin{aligned} i\hbar\frac{\psi^{n+2}_{i,j}-\psi^{n+1}_{i,j}}{\Delta t} = \frac{-\hbar^2}{2m}&\left[\frac{\psi^{n+2}_{i,j+1}+\psi^{n+2}_{i,j-1}-2\psi^{n+2}_{i,j}}{(\Delta y)^2}\right. \\ &\left.+\frac{\psi^{n+1}_{i+1,j}+\psi^{n+1}_{i-1,j}-2\psi^{n+1}_{i+1,j}}{(\Delta x)^2}\right] \\ &+\left[V_{i,j}\psi^{n}_{i,j}+V_{i,j}\psi^{n+1}_{i,j}\right]. \end{aligned} \tag{4}$$

In these equations, terms proportional to ψ^{n+1} and ψ^{n} both appear with different spatial indices and in multiple places, meaning that the equations cannot be solved explicitly. Instead, they can be cast into matrix form, and solved using standard linear algebra techniques.

For simplicity, we will now set $\hbar = m = 1$, and set $\Delta x = \Delta y$. We begin with the even steps, going from n to $n+1$.

$$\begin{aligned} \psi^{n+1}_{i,j}-\psi^{n}_{i,j} = \frac{i\Delta t}{2(\Delta x)^2}&\left[\psi^{n+1}_{i+1,j}+\psi^{n+1}_{i-1,j}-2\psi^{n+1}_{i,j}\right. \\ &\left.+\psi^{n}_{i,j+1}+\psi^{n}_{i,j-1}-2\psi^{n}_{i,j}\right] \\ &-i\Delta t\left[V_{i,j}\psi^{n}_{i,j}+V_{i,j}\psi^{n+1}_{i,j}\right] \end{aligned} \tag{5}$$

Collecting terms of the same spatial indices on either side, and letting $\alpha = i\Delta t/(2\Delta x^2)$, $\beta = i\Delta t$, we obtain

$$\begin{aligned} &\psi^{n+1}_{i,j}(1+2\alpha+\beta V_{i,j})-\alpha\left[\psi^{n+1}_{i+1,j}+\psi^{n+1}_{i-1,j}\right] \\ =&\psi^{n}_{i,j}(1-2\alpha-\beta V_{i,j})+\alpha\left[\psi^{n}_{i,j+1}+\psi^{n}_{i,j-1}\right] \end{aligned} \tag{6}$$

Each side can now be conveniently collecting into a matrix equation with non-zero elements on the major diagonal and upper and lower minor diagonals. This form of matrix is a sparse block-tridiagonal matrix, which is computationally easy to deal with.

As a final notational simplification, we define

$$\tilde{V}^{+}_{i,j} = 1+2\alpha+\beta V_{i,j}$$

$$\tilde{V}^{-}_{i,j} = 1-2\alpha-\beta V_{i,j}$$

The left hand side can be expressed as

$$\begin{pmatrix} \tilde{V}^{+}_{0,j} & -\alpha & 0 & 0 & ... \\ -\alpha & \tilde{V}^{+}_{1,j} & -\alpha & 0 & ... \\ 0 & -\alpha & \tilde{V}^{+}_{2,j} & -\alpha & ... \\ .. & ... & ... & ... & \end{pmatrix}\begin{pmatrix} \psi^{n+1}_{0,j} \\ \psi^{n+1}_{1,j} \\ \psi^{n+1}_{2,j} \\ ... \end{pmatrix} \tag{7}$$

while the right hand side can be expressed as

$$\begin{pmatrix} \tilde{V}^{-}_{i,0} & \alpha & 0 & 0 & ... \\ \alpha & \tilde{V}^{-}_{i,1} & \alpha & 0 & ... \\ 0 & \alpha & \tilde{V}^{-}_{i,2} & \alpha & ... \\ .. & ... & ... & ... & \end{pmatrix}\begin{pmatrix} \psi^{n}_{i,0} \\ \psi^{n}_{i,1} \\ \psi^{n}_{i,2} \\ ... \end{pmatrix}. \tag{8}$$

Note, crucially, that the ordering of the vectors representing the 2 dimensional wave function are different. In the matrix expression for the left hand side, the vector consists of blocks of constant j with varying i, whereas for the right hand side, it is flipped - blocks of constant i with varying j. To make these expressions agree, they need to be rearranged. Fortunately, this is easy to do in code.

Further, similar expressions can be derived for the odd steps, going from $n+1$ to $n+2$

The left hand side reads

$$\begin{pmatrix} \tilde{V}_{i,0}^{+} & -\alpha & 0 & 0 & ... \\ -\alpha & \tilde{V}_{i,1}^{+} & -\alpha & 0 & ... \\ 0 & -\alpha & \tilde{V}_{i,2}^{+} & -\alpha & ... \\ .. & ... & ... & ... & \end{pmatrix} \begin{pmatrix} \psi_{i,0}^{n+2} \\ \psi_{i,1}^{n+2} \\ \psi_{i,2}^{n+2} \\ ... \end{pmatrix} \tag{9}$$

And the right hand side is

$$\begin{pmatrix} \tilde{V}_{0,j}^{-} & \alpha & 0 & 0 & ... \\ \alpha & \tilde{V}_{1,j}^{-} & \alpha & 0 & ... \\ 0 & \alpha & \tilde{V}_{2,j}^{-} & \alpha & ... \\ .. & ... & ... & ... & \end{pmatrix} \begin{pmatrix} \psi_{0,j}^{n+1} \\ \psi_{1,j}^{n+1} \\ \psi_{2,j}^{n+1} \\ ... \end{pmatrix} \tag{10}$$

B. Boundary and Initial Conditions

Treating the inner wall with slits as a region of large finite potential, we demand that $V = V_0$ at some fixed x_{slit}, except for $y_{\text{slit 1,bottom}} \leq y \leq y_{\text{slit 1,top}}$ and $y_{\text{slit 2,bottom}} \leq y \leq y_{\text{slit 2,top}}$.

$$\begin{aligned} V(x_{\text{slit}}, 0 \leq y \leq y_{\text{slit 1,bottom}}, t) &= V_0 \\ V(x_{\text{slit}}, y_{\text{slit 1,top}} \leq y \leq y_{\text{slit 2,bottom}}, t) &= V_0 \\ V(x_{\text{slit}}, y_{\text{slit 2,top}} \leq y \leq y_{\max}, t) &= V_0 \end{aligned}$$

The initial condition used is a travelling Gaussian wavepacket, localised near the left edge of the box. A 2 dimensional Gaussian distribution is used, momentum in the $\hat{x}$ direction.

The explicit form of the wavepacket is

$$\Psi(x, y, 0) = A \exp\left\{ ik_x x - \frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2} \right\}$$

Here, the constant A is for normalisation, to ensure that the wavefunction satisfies the probability-density axiom of quantum mechanics. It is computed numerically, by performing the integral

$$\int_X \int_Y |\Psi|^2 dy dx = 1$$

and solving for A.

IV. IMPLEMENTATION

In Python 3.7, a 3D `numpy` array was set up, with axes for x, y, t. The value of $\Psi(x, y, 0)$ was computed by looping through the each point of the 2D array at time 0, and computing the Gaussian wavefunction value. Then, the wavefunction is normalised as described above, computing the normalisation constant by numerical trapezoidal integration.

The 4 requisite arrays are initialised as `scipy sparse` arrays, thus bringing substantial memory-usage savings, since each row only contains at most 3 non-zero elements.

A time loop is then set up, and within the loop, two steps are executed. The wavefunction matrix at time n is reshaped into a vector appropriately, and the right-hand side of the first step (Equation 8) is evaluated by matrix multiplication. The vector is then reordered, and the linear system of the left hand side is solved to obtain the wavefunction vector of $n+1$ (Equation 8) using the `scipy.sparse.linalg.spsolve` function. A similar process is repeated for the second step, solving the left hand side by matrix multiplication (Equation 10) and then solving for the wavefunction vector at $n+2$ (Equation 9). Finally, the wavefunction is reshaped back into a matrix and stored.

The probability density is computed by taking the norm squared of the wavefunction at all times, and plotted as a heatmap.

A cross-section of the probability density, at different distances away from the wall with the slits, is also considered, to serve as a virtual "screen" to compare with the photon double slit experiment.

V. RESULTS

A snapshot of the probability density some time after arriving at the slit is shown in Figure 2. A large portion of the wavefunction reflects off, but the portion going through the slits forms an interference pattern.

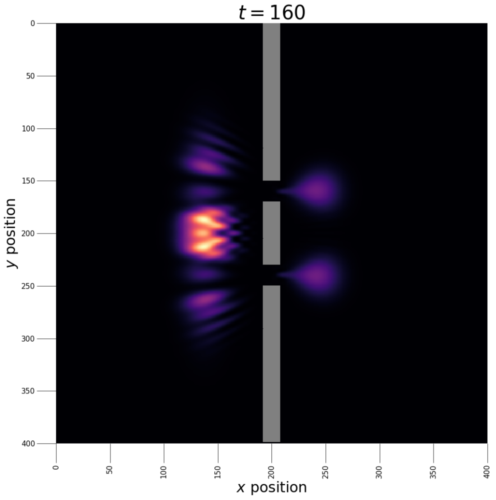

Figure 2: Probability density in the cell at $t = 160$.

Zooming in on the region after the slit allows us to see this interference pattern much more clearly.

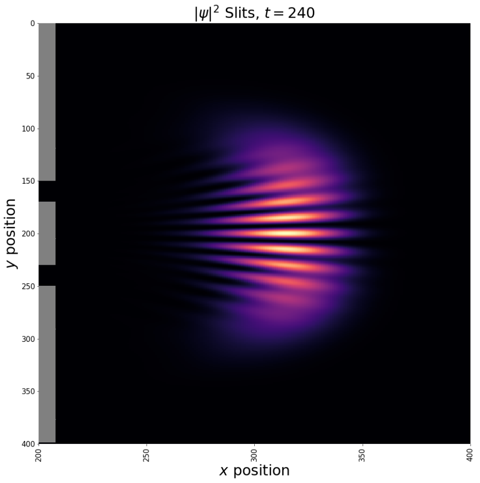

Figure 3: Probability density in the right half of the cell, at $t = 245$.

Slicing this 2 dimensional picture near the peak, at $x = 320$, we can see that the pattern of a usual double slit experiment is replicated. The classically predicted pattern and the simulated pattern are shown in Figure 4.

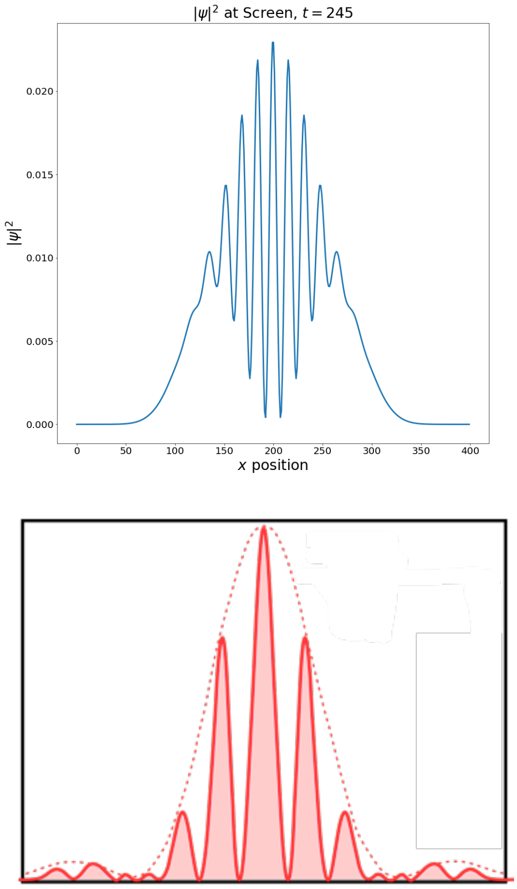

Figure 4: Top: Probability density $x = 320, t = 245$. Bottom: Classically predicted double slit interference pattern at screen.[3]

A set of regular time slices of the simulation is shown in the Appendix.

Some other cases of the experiment were also simulated. First, a single-slit variation was simulated. Figure 5 shows the diffraction pattern on the screen as expected, compared to the classical predicted pattern.

A variation using five slits was performed. Here, as expected, the simulation showed sharper, more intense peaks. The simulated pattern and the classically predicted pattern are shown in Figure 6.

Similar to the two slit case, a zoomed-in heatmap for the five slit case is shown in Figure 7.

The results generally qualitatively agree with the classical predictions in terms of the shape, number of peaks and relative intensity of peaks, and modulation by a single-slit envelope. However, certain aspects are not well replicated, including the small secondary maxima. This may have to do with the discrepancy in comparing a slice of probability density from the simulation, with a classically predicted graph as a function of angle from normal. Additionally, this may be a result of inadequately fine

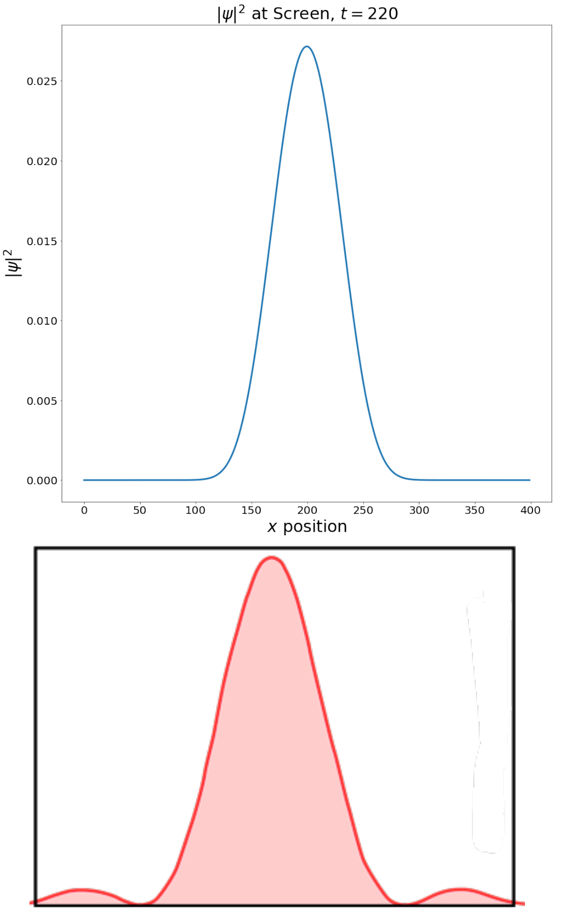

Figure 5: Top: The simulated probability density of a single slit simulation, at an x slice. Bottom: The classically predicted probability density of a single slit simulation, at an x slice.[3]

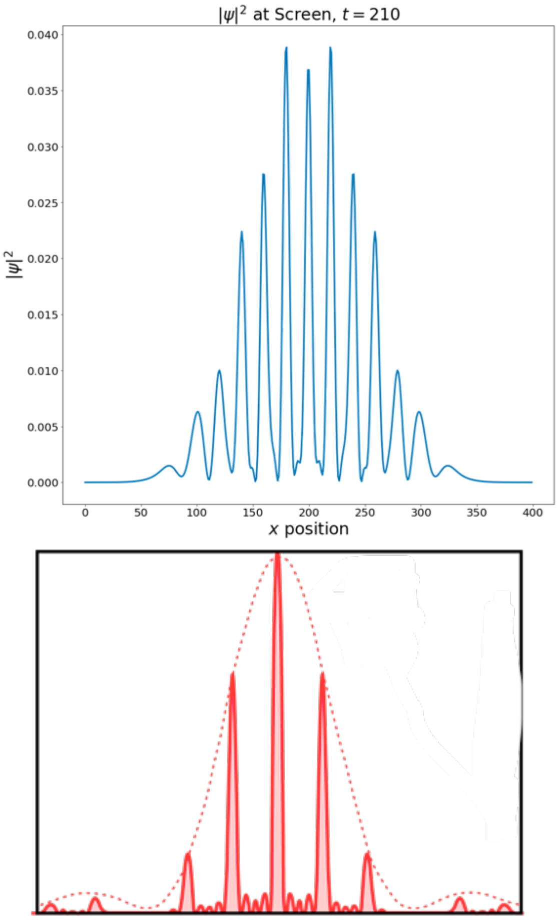

Figure 6: Top: The simulated probability density of a five slit simulation, at an x slice. Bottom: The classically predicted probability density of a five slit simulation, at an x slice.[3]

spatial resolution of the simulation.

VI. ANALYSIS

The ADI technique was found to be exceedingly stable, even for relatively large time steps.

Computationally, the most expensive operation by far was solving the linear systems to obtain the next wavefunction vector. Using the Python `timeit` function, the approximate time taken for the linear system inversion was $150ms$ for a spatial grid size of 400×400. This was an order of magnitude larger than the next most time-intensive operation, of taking a dot product. The approximate times taken are shown in Figure 8.

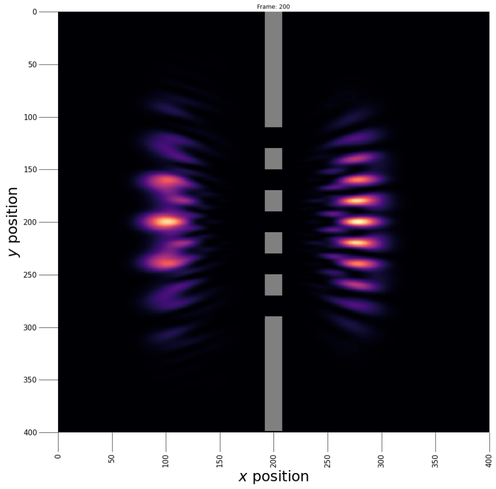

Figure 7: The simulated probability density of a five slit simulation, as a two dimensional heatmap, zoomed in after the slits.

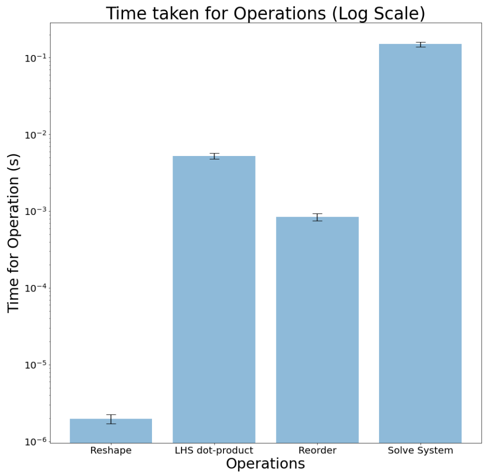

Figure 8: Approximate time taken per operation, in seconds, with a logarithmic scale

VII. CONCLUSION

An obvious direction for further improvement is to optimise the linear system solving, by better exploiting the matrix tridiagonal structure, or by using more efficient libraries. This would allow simulation at finer spatial resolution at lower cost, allowing for high-fidelity simulations.

Another area for improvement is increasing the order of the algorithm, to reduce errors further.

An important and challenging extension to this method of solving the Schrodinger equation would be extending it to 3D, since that would involve careful thought about how to arrange the matrices and wavefunction vectors.

This program and technique can be generalised to other potentials easily. It can be a useful teaching instrument in introductory quantum mechanics, so students can quickly visualise the solutions to physical experiments.

VIII. ACKNOWLEDGEMENTS

I would like to thank Professor Blas Cabrera and my TA, Stanislav Fort, for their support in this project.

REFERENCES

[1] *Alternating direction implicit method*. In: *Wikipedia*. Page Version ID: 960497360. June 3, 2020. URL: https://en.wikipedia.org/w/index.php?title=Alternating_direction_implicit_method&oldid=960497360 (visited on 06/04/2020).

[2] Carl R. Nave. *Fraunhofer Diffraction*. HyperPhysics. 2017. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/fraungeo.html#c1 (visited on 06/04/2020).

[3] Carl R. Nave. *Multiple Slit Diffraction*. HyperPhysics. 2017. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/mulslid.html (visited on 06/04/2020).

[4] *Double-slit experiment*. In: *Wikipedia*. Page Version ID: 958936183. May 26, 2020. URL: https://en.wikipedia.org/w/index.php?title=Double-slit_experiment&oldid=958936183 (visited on 06/04/2020).

[5] NekoJaNekoJa and Johannes Kalliauer. *File:Double-slit.svg*. Aug. 5, 2017. URL: https://commons.wikimedia.org/wiki/File:Double-slit.svg (visited on 06/04/2020).

[6] *quantum mechanics - Are there simple ways to numerically solve the time-dependent Schrödinger equation?* Computational Science Stack Exchange. Library Catalog: scicomp.stackexchange.com. URL: https://scicomp.stackexchange.com/questions/10876/are-there-simple-ways-to-numerically-solve-the-time-dependent-schr%c3%b6dinger-equati (visited on 06/04/2020).

IX. APPENDIX

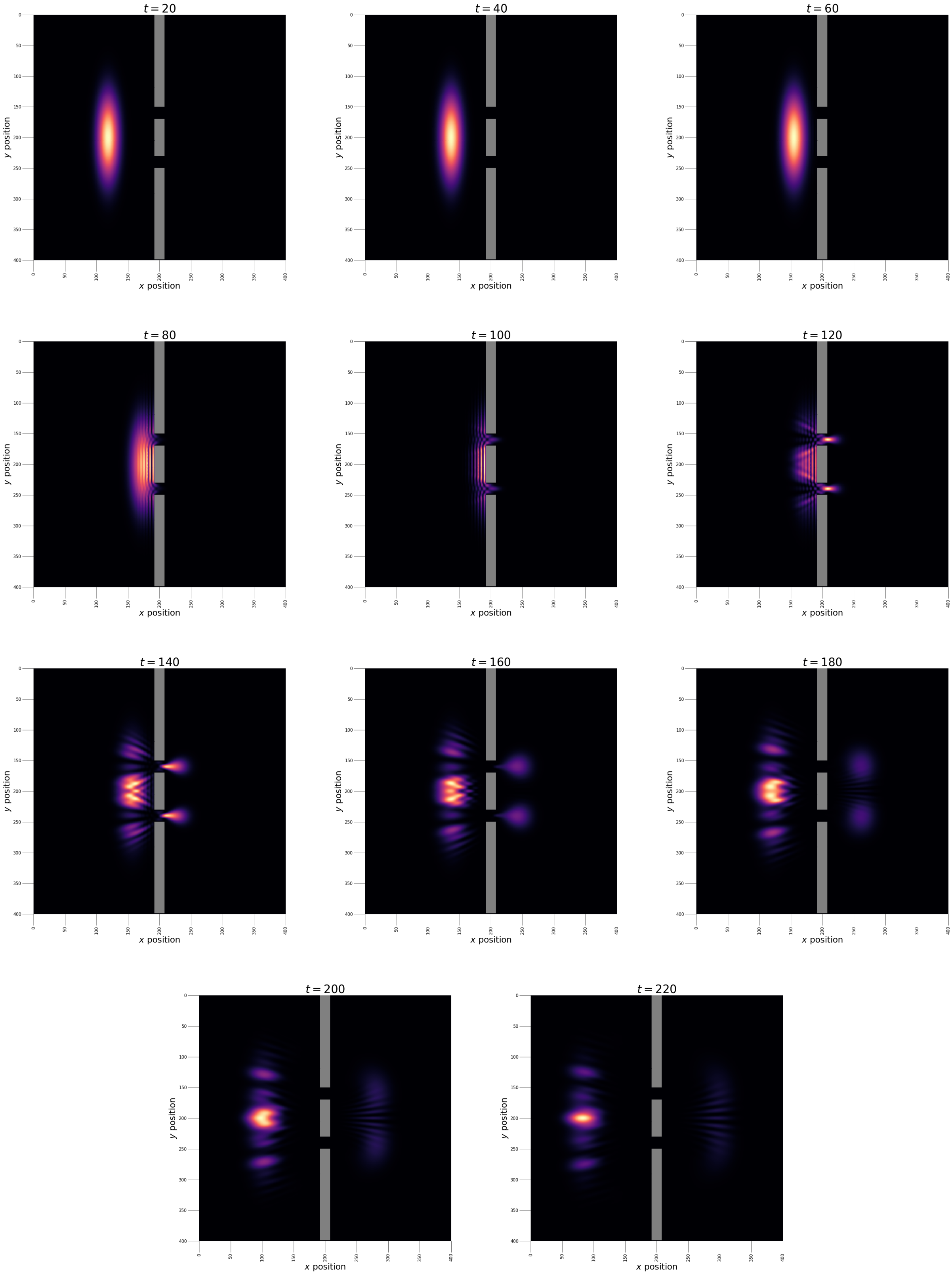

Figure 9: Time series of 2 dimensional heatmaps of probability density $|\psi|^2$. Brighter regions correspond to higher density. The grey region is the double slit potential barrier.