

Numerically simulating the 2D many-slit experiment using the Crank-Nicolson method

Henrik Haug

Axl H Kleven

Live Ljungqvist Storborg

(Dated: September 22, 2025)

We have simulated the many-slit experiment for 0, 1, 2, and 3 slits in 2D. The simulation was made using the Crank-Nicolson method with the Schrödinger equation in combination with Dirichlet boundary conditions. We have studied the probability as a function of time, and illustrated how the wave function behaves before, during, and after passing the slits. The simulation has proven to conserve probability to an order of 10^{-14} over a total time $T = 0.010$ with time steps $\Delta t = 2.5 \cdot 10^{-5}$. When applying a double slit, the simulation produced illustrations clearly showing diffraction and interference, illustrating the wave-nature of quantum particles. When simulating the many-slit experiment by measuring the probability of finding the particle at a position $x = 0.8$ and time $t = 0.002$, we found expected probabilities similar to a classical wave experiencing diffraction and interference when passing through similar slits.

I. INTRODUCTION

The double-slit experiment is a well-known experiment, proving that particles can exhibit wave-like properties. As seen in FIG 1, electrons can experience a diffraction and interference (with itself) according to a classical wave. These wave properties can be explained using quantum mechanics. The Schrödinger equation provides an equation describing the time evolution of a quantum particle in a given potential. It does so by evolving the wave function, which can be used to find the probability density of a particle at a given position and time.

In this article we will be numerically solving the Schrödinger equation of a particle passing through a varying number of slits. We will mainly focus on the computational aspect of solving the Schrödinger equation itself. As such, we will be using dimensionless variables. Further, we will be using the Crank-Nicolson method for discretizing and solving the wave function, as the method is probability conserving. We will be using a simple gaussian wave packet to describe the particle, with a simple and easily modifiable setup of the slits.

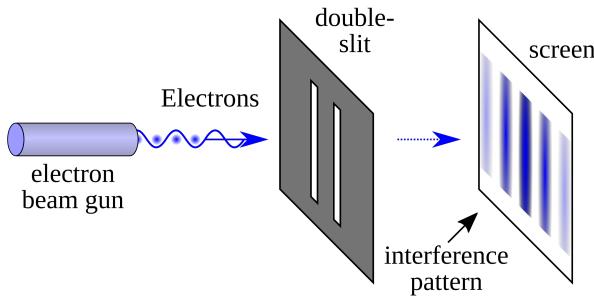


FIG. 1. Figure from [1]. Electrons are sent through two small slits, where they experience a diffraction pattern similar to a wave, with their final positions being measured on a screen behind the slits.

To test our simulation, we will be exploring the conservation of probability, and visually studying the evolution of the probability over time. Finally, we will use our simulation to predict the probability of finding the particle on a screen behind the slit at a given time. We will do so for a single, double, and triple-slit, to further explore how the amount of slits affect the probability of where we can expect to find the particle.

II. METHOD & THEORY

A. The Schrödinger equation

When considering a quantum system, the position of a particle no longer evolves exactly, as it would in a classical system. The particle can be detected at a position according to a probability, defined by the wave function Ψ . The wave function contains all the information about the quantum state of a given particle. The time evolution of the wave function is described by the Schrödinger equation (SE), as shown in equation (1)

$$i\hbar \frac{d}{dt} \Psi(x, y, t) = \hat{\mathbf{H}} \Psi(x, y, t). \quad (1)$$

Where $i = \sqrt{-1}$ is the imaginary number, \hbar is the reduced Planck's constant, $\Psi(x, y, t)$ is the time dependent wave function in the xy -plane, and $\hat{\mathbf{H}}$ is the Hamiltonian operator. The probability of finding the particle at a position (x, y) , at a time t , is defined by the Born rule, shown in equation (2). The Born rule states that the probability can be found by taking the absolute value squared of the wave function.

$$P(x, y; t) = |\Psi(x, y, t)|^2 = \Psi^*(x, y, t)\Psi(x, y, t). \quad (2)$$

The Hamiltonian can be expressed as

$$\hat{\mathbf{H}} = \hat{\mathbf{T}} + \mathbf{V}, \quad (3)$$

where \mathbf{V} is the potential energy containing all the information about the external environment and $\hat{\mathbf{T}}$ is the kinetic energy operator

$$\hat{\mathbf{T}} = \frac{\hat{\mathbf{p}}^2}{2m} = -\frac{\hbar^2}{2m} \left(\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2} \right). \quad (4)$$

Defined using the kinetic energy $K = \frac{p^2}{2m}$ from classical physics, where $p = mv$ is the momentum, using the velocity v . The mass of the particle is denoted m .

Seeing as this is a numerical project, we will be scaling away all dimensionful variables, giving a ‘bare’ SE as seen in equation (5)

$$i \frac{\delta \mathbf{u}}{\delta t} = -\frac{\delta^2 \mathbf{u}}{\delta x^2} - \frac{\delta^2 \mathbf{u}}{\delta y^2} + \mathbf{v}(x, y) \mathbf{u}, \quad (5)$$

where all the variables are dimensionless, which is practical for numerical projects. This is the equation we will solve numerically, where \mathbf{u} now represents the wave function. As such, our probability from the Born rule now being

$$p(x, y; t) = |\mathbf{u}(x, y, t)|^2 = \mathbf{u}^*(x, y, t) \mathbf{u}(x, y, t), \quad (6)$$

where the total probability in the xy -plane at a time t should always be equal to 1.

1. Discretization, boundary conditions and notation

In the context of a numerical project, we need to discretize our simulation and define boundaries. We will be using

$$\begin{aligned} x &\in [0, 1], \\ y &\in [0, 1], \\ t &\in [0, T], \end{aligned} \quad (7)$$

where T is a variable, defining the total time. For the x and y directions, we will be using M points of a step size h , not to be confused with Planck’s constant. For the time t , we will use a step size Δt . The step sizes will be determined according to the accuracy needed.

For the entirety of this project we will be assuming Dirichlet boundary conditions in the xy -plane, which are specified by

$$\begin{aligned} \mathbf{u}(x = 0, y, t) &= 0, \\ \mathbf{u}(x = 1, y, t) &= 0, \\ \mathbf{u}(x, y = 0, t) &= 0, \\ \mathbf{u}(x, y = 1, t) &= 0. \end{aligned} \quad (8)$$

These boundary conditions will also be helpful for our Crank-Nicolson method, described in section II B.

To simplify our notation, we define

$$\mathbf{u}(x, y, t) \rightarrow u(ih, jh, n\Delta t) \equiv u_{ij}^n, \quad (9)$$

where i and j denote the indexes of x and y respectively, and n is the time-index. Thus, we can define a matrix \mathbf{U}^n , with matrix elements u_{ij}^n . This also applies for the potential

$$\mathbf{v}(x, y) \rightarrow v(ih, jh) \equiv v_{ij}, \quad (10)$$

which can be used as matrix elements of a matrix \mathbf{V} in a similar manner.

B. Crank-Nicolson method

The Crank-Nicolson method is a numerical method used to solve partial differential equations (PDEs). The method is implicit in nature and is known for being unconditionally stable, making it effective for solving equations over long time intervals. The method also has a second order accuracy, with a conservation of probability. This makes it a good fit for solving the SE.

The method consists of a weighted sum of an implicit and an explicit method: The backward Euler method and the forward Euler method, respectively. The method takes the weighted average of these two methods to achieve a higher accuracy. To establish the method, we begin by defining the following equation

$$\frac{\delta u(x, y, t)}{\delta t} = F(x, y, t). \quad (11)$$

Using this, the forward difference becomes

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^n, \quad (12)$$

and the backwards difference becomes

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^{n+1}. \quad (13)$$

A linear combination of these two produces

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \theta F_i^{n+1} + (1 - \theta) F_i^n \quad (14)$$

where $\theta \in [0, 1]$ defines the weights. These weights control the contribution of the forward difference (maximized for $\theta = 0$) or the backward difference (maximized for $\theta = 1$). The Crank-Nicolson method uses $\theta = \frac{1}{2}$.

1. Application to the wave function

Applying the discretizations described in the previous section II A 1, we can rewrite our bare SE from equation (5), according to the Crank-Nicolson approach. We get equation (15). The derivation of this equation is shown

in Appendix A.

$$\begin{aligned}
& u_{ij}^{n+1} - r [u_{i+1,j}^{n+1} - 2u_{ij}^{n+1} + u_{i-1,j}^{n+1}] \\
& - r [u_{i,j+1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j-1}^{n+1}] + \frac{i\Delta t}{2} v_{ij} u_{ij}^{n+1} \\
& = u_{ij}^n + r [u_{i+1,j}^n - 2u_{ij}^n + u_{i-1,j}^n] \\
& + r [u_{i,j+1}^n - 2u_{ij}^n + u_{i,j-1}^n] - \frac{i\Delta t}{2} v_{ij} u_{ij}^n. \quad (15)
\end{aligned}$$

Where $r \equiv \frac{i\Delta t}{2h^2}$. When taking the Dirichlet boundary conditions from equations (8) into account, we can rewrite equation (15) as follows

$$\mathbf{A}\mathbf{u}^{n+1} = \mathbf{B}\mathbf{u}^n. \quad (16)$$

Here, the vectors \mathbf{u}^n and \mathbf{u}^{n+1} are column vectors containing all the u_{ij}^n values on the xy -grid at time step n or $n+1$. The vectors do however ignore the boundary points, and as such only consider the internal u_{ij}^n values. The vectors on row-form take the shape shown in equation (17).

$$\mathbf{u}^n = [(u_{1,1}^n, u_{2,1}^n, \dots, u_{M-2,1}^n), (u_{1,2}^n, u_{2,2}^n, \dots, u_{M-2,2}^n), \dots]. \quad (17)$$

As such the vectors will be of length $(M-2)^2$, and the matrices \mathbf{A} and \mathbf{B} will be of sizes $(M-2)^2 \times (M-2)^2$. The contents with an example of these matrices are shown in Appendix B.

We solve this equation by first performing the matrix multiplication $\mathbf{B}\mathbf{u}^n = \mathbf{b}$. Then we solve the matrix equation $\mathbf{A}\mathbf{u}^{n+1}$, finding the unknown \mathbf{u}^{n+1} . The matrices \mathbf{A} and \mathbf{B} will be sparse matrices, which we can utilize in our code to improve efficiency using the Armadillo [2] library.

C. The wave packet

We will be using a Gaussian wave packet that is not normalized on the form of equation (18).

$$u(x, y, t=0) = e^{-\frac{(x-x_c)^2}{2\sigma_x^2} - \frac{(y-y_c)^2}{2\sigma_y^2} + ip_x x + ip_y y}. \quad (18)$$

Here, x_c and y_c are the coordinates of the center of the initial wave packet. The initial widths of the wave packet are σ_x and σ_y , and p_x and p_y are the momenta of the wave packet. The directions are denoted by x and y .

The normalization of the probability is an important aspect of the project, as the sum of probability over all xy should equal 1. Therefore, we will implement numerical methods for normalizing, such that

$$\sum_{i,j} u_{ij}^{0*} u_{ij}^0 = 1. \quad (19)$$

This means that the total probability at the first time step is initially set to 1. The Crank-Nicolson method conserves probability, providing a useful test case for validating the code.

D. The double-slit

We wish to first study the double-slit, and then later study different shapes, numbers, and positions of the slits. This will allow us to both test that the simulation works according to expectation, but also explore the behaviour of the system.

The slits are modeled as a strong potential v_0 where we want a ‘wall’, and the potential is set to 0 everywhere else. We need v_0 in combination with the width of the wall, to be large enough to avoid tunneling. The value of v_0 is found through experimentation.

For the dimensions of the wall we start with:

- Wall thickness: $x = 0.02$.
- Wall center: $x = 0.5, y = 0.5$.
- Length of the slit itself: $y = 0.05$.
- Length of the wall between the slits: $y = 0.05$.

These initial values are chosen, as they represent a simple double-slit, which should produce an expected result of a wave pattern using a high enough v_0 .

E. Testing and using the simulation

As mentioned, the Crank-Nicolson method conserves probability. As such we can test our simulation by looking at the probability as a function of time. We do this both with and without the double-slits, in order to test if and how the probability deviates from 1 over time. We want the particle to ‘hit’ the center of the double-slits. As such we will initiate the particle in the center on the y -axis, with a velocity only in the x -direction. This applies for all testing and simulating. We will vary the particle’s ‘height’ (σ_y), which scatters the probability over a larger area. This increases visibility in the plots and tests the model’s accuracy with a less concentrated probability.

Continuing with the double-slits, we wish to see that the simulation produces the expected diffraction of the wave function through the double-slit, and that the wave function evolves as expected in time. To do this, we will make a colourmap of the probability at different times during the simulation. We want to study what the probability of finding the particle initially looks like, and how it evolves over time, with special interest in how the slits affect the probability. We will here use a larger σ_y to increase the visibility of the particle over time.

Finally, we wish to use our simulation to study the detection probability on a fixed screen, simulating the double-slit experiment. The position and time of the screen are determined from the previous test, looking for an interference pattern behind the slits. We wish to find

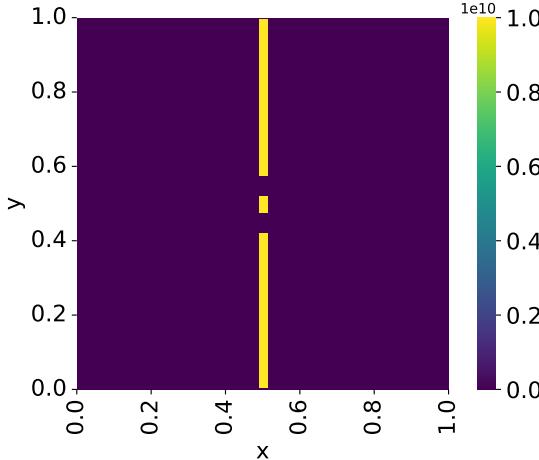


FIG. 2. The double-slit. All slit variations are made in a similar fashion, varying only in the amount of slits used.

the probability along the screen for single, double, and triple-slits. Following the simulation of the experiment, we assume a 100% chance of finding the particle along the screen, and normalize the result accordingly. On the screen we expect a diffraction pattern, but we expect the diffraction pattern to become more and more clear as we increase the slits, with different positions for odd and even slits. This behavior mirrors that of classical waves, serving both as a test and as a practical application of the simulation. By simply adjusting the initial position of the particle, the slits, and the screen's position, the simulation can predict the probability distribution for where the particle is likely to be found.

F. Tools

For code collaboration, we have used GitHub, and our code can be found by following this link [3]. Our code is mainly written in C++, where we have used the additional Armadillo [2] library for linear algebra operations. For creating figures we have used Python, with the following libraries: matplotlib [4], numpy [5], pandas [6], and seaborn [7].

III. RESULTS & DISCUSSION

We initialize the potential using the parameters described in II D. The potential for a double slit can be visualized as seen in FIG 1. We have not defined the potential outside the boundary conditions, which is visible on the borders of the figure.

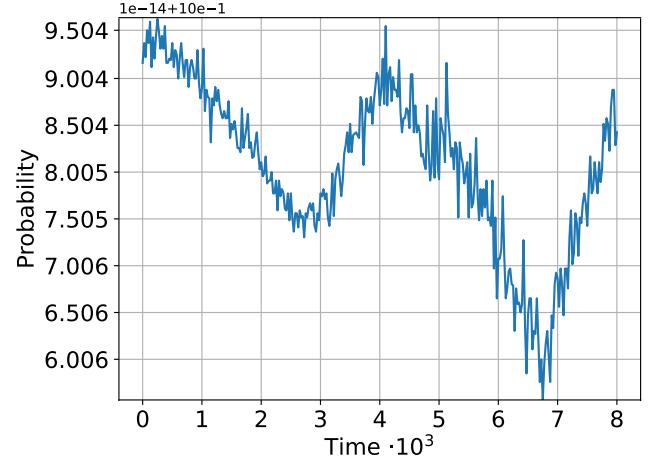


FIG. 3. The total probability of finding the particle over a total time $T = 0.008$ using a timestep $\Delta t = 2.5 \cdot 10^{-3}$, using no potential.

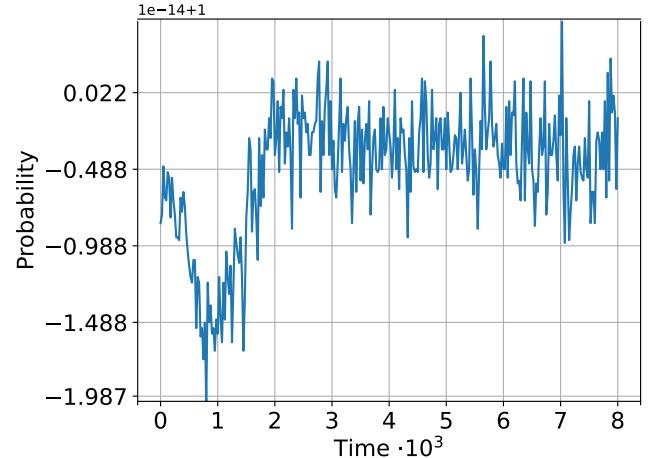


FIG. 4. The total probability of finding the particle over a total time $T = 0.008$ using a timestep $\Delta t = 2.5 \cdot 10^{-3}$, using a double-slit potential.

A. Conservation of probability

For testing the conservation of probability, we define the particle using $x_c = 0.25, \sigma_x = 0.05, p_x = 200, y_c = 0.5, \sigma_y = 0.05, p_y = 0$. We define the simulation using $h = 0.005, \Delta t = 2.5 \times 10^{-5}, T = 0.008$, with no potential ($v_0 = 0$). This provides a probability as a function of time, as seen in FIG 3.

Turning on the double-slit, using $v_0 = 10^{10}$, and using $\sigma_y = 0.1$, we re-run the previous simulation. This gives a probability as a function of time as seen in FIG 4.

In both FIG 3 and FIG 4, we can see that the probability of finding the particle is always approximately 1. The deviations from 1 are of order 10^{-14} , which is generally the order of floating point errors. As such, the

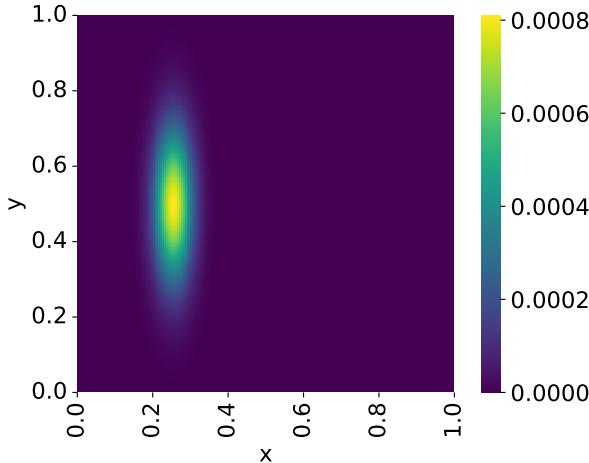


FIG. 5. Probability of finding the particle at $t = 0$, using a double-slit.

Crank-Nicolson method appears correctly implemented. Further we can see that the probability using a potential appears more stable than the probability without a potential, especially after $\sim t = 0.002$. This is unexpected as we would expect the two cases to be equally stable if the method is working optimally.

B. Evolution in time

We change our parameters to $T = 0.2$, $\sigma_y = 0.2$, keeping all other parameters the same as in the previous section IIIA. For the double-slit, the probability of finding the particle at times $t \in \{0, 0.001, 0.002\}$ are visualised in FIG 5, 8, 11. These probabilities are calculated through the real and imaginary parts of the wave function. The probabilities can also be visualized, and are presented respectively in FIG 6, 9, 12, and FIG 7, 10, 13. We selected the times through experimentation, focusing on the interaction with the slit.

We can see that the probability of finding the particle starts out as a tall oval shape, due to increasing σ_y . When the particle ‘hits’ the slit in FIG 8, we can see a clear wave behavior of the probability. Parts of the probability passes through both the slits, while the rest is reflected back in the form of a wave, where there are stripes of high probability, and stripes of very low probability. After passing through the slits, the probability now shows a clear interference pattern, similar to a classical wave. This is as expected. We can see that at a time $t = 0.002$, we could expect to find the particle on a screen placed along $x = 0.8$. We will use this later.

We can see that the real and imaginary parts look very similar to the probability, but they are more clearly wave-like. Where the probability only has clear stripes around the slit itself, the real and imaginary parts show these stripes everywhere. The stripes indicate the tops and

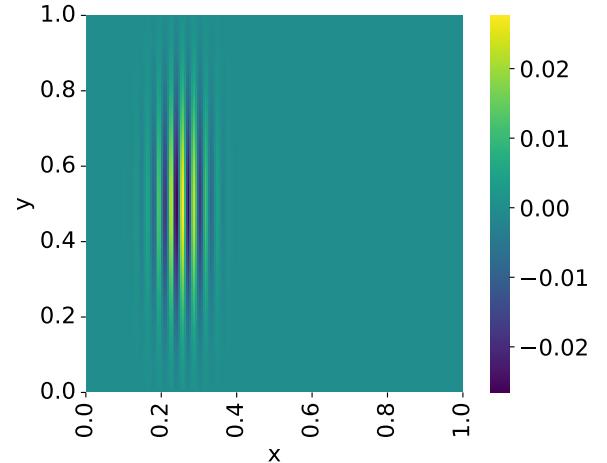


FIG. 6. Real part of the wave function of the particle at $t = 0$, using a double-slit.

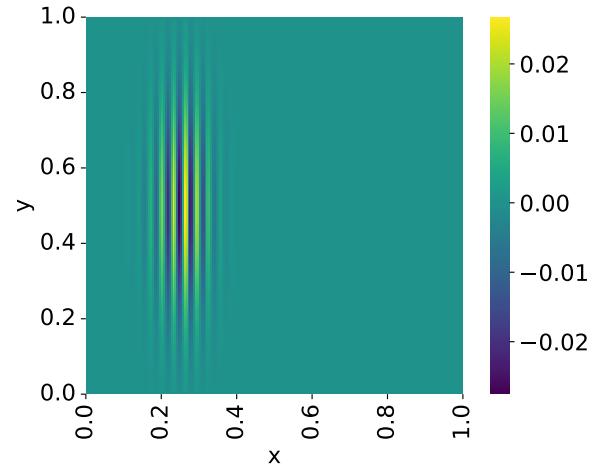


FIG. 7. Imaginary part of the wave function of the particle at $t = 0$, using a double-slit.

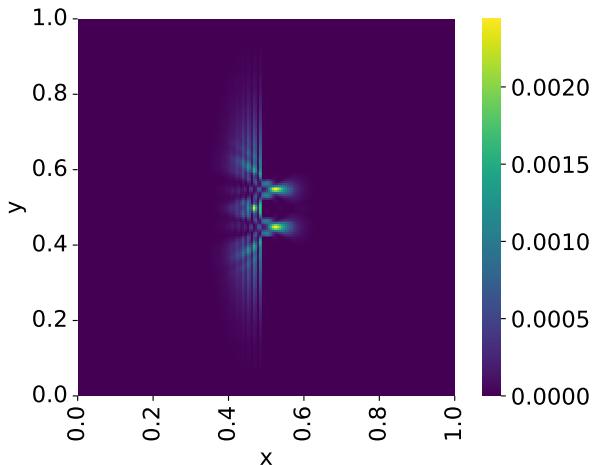


FIG. 8. Probability of finding the particle at $t = 0.001$, using a double-slit.

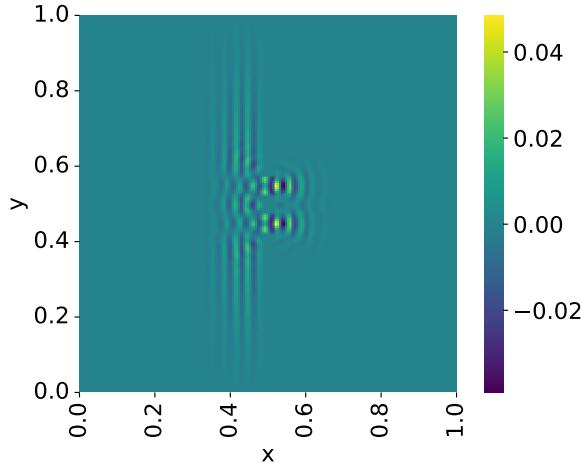


FIG. 9. Real part of the wave function of the particle at $t = 0.001$, using a double-slit.

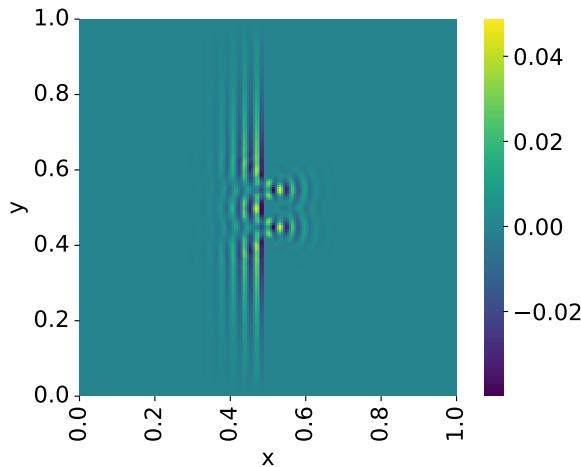


FIG. 10. Imaginary part of the wave function of the particle at $t = 0.001$, using a double-slit.

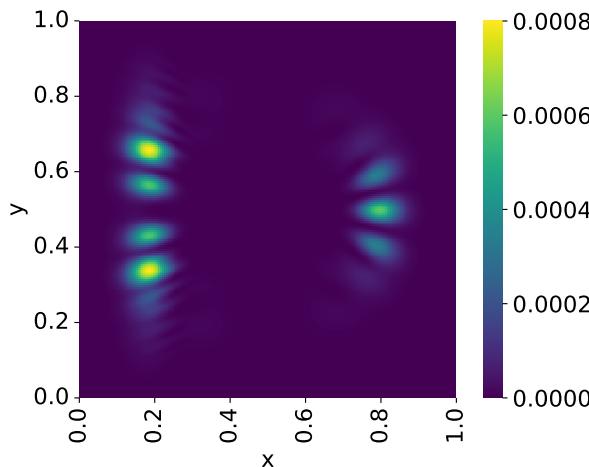


FIG. 11. Probability of finding the particle at $t = 0.002$, using a double-slit.

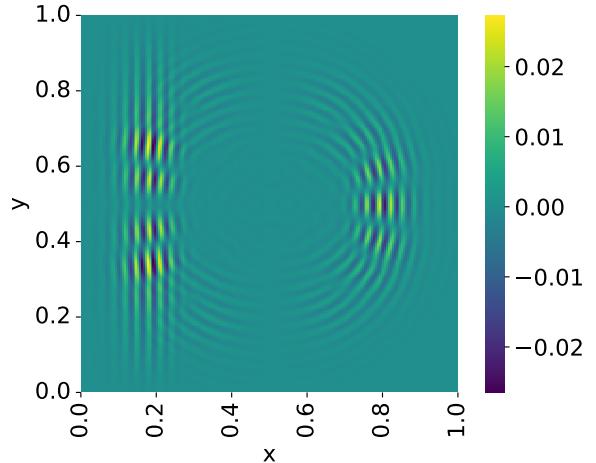


FIG. 12. Real part of the wave function of the particle at $t = 0.002$, using a double-slit.

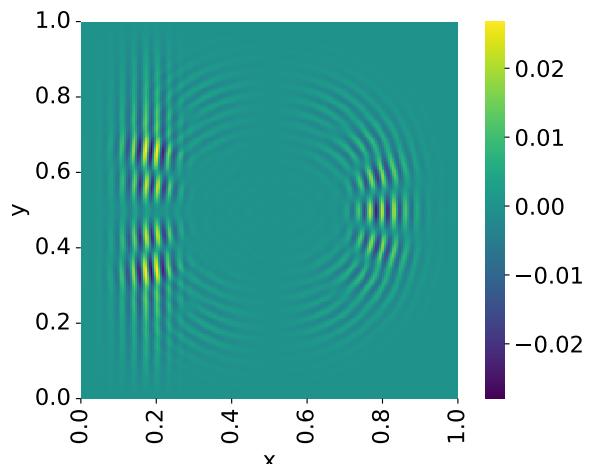


FIG. 13. Imaginary part of the wave function of the particle at $t = 0.002$, using a double-slit.

bottoms of waves. Further, the diffraction and wave patterns are more clearly shown in the real and imaginary parts. This makes them interesting to study, as the wave nature of particles is more clearly shown.

An animation of these probabilities up to $T = 0.0010$ can be found on our GitHub [3]. This animation shows the same elements as the figures; the wave nature of the probability.

C. Detection on a screen

Using the same parameters as in the previous section III B, we wish to study the probability of finding the particle on a screen behind the slits. The screen is placed at $x = 0.8$, $t = 0.002$, following FIG 11. We use a single, double, and triple slit, respectively, in FIG 14, 15, 16.

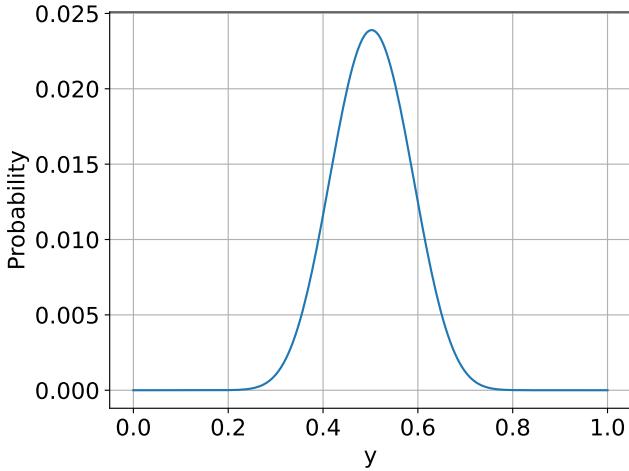


FIG. 14. Probability of finding the particle along a screen at $x = 0.8, t = 0.002$, using a single-slit.

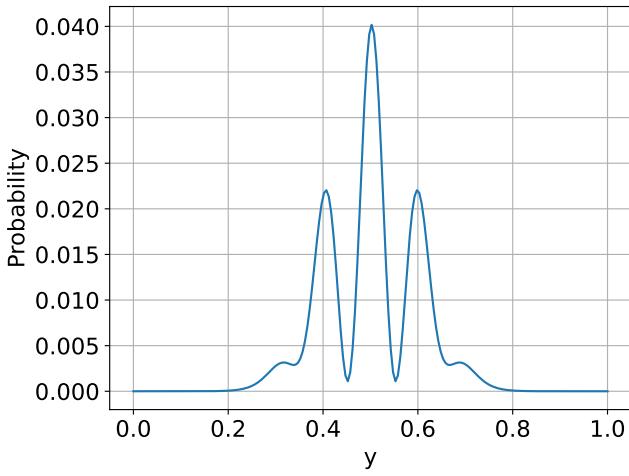


FIG. 15. Probability of finding the particle along a screen at $x = 0.8, t = 0.002$, using a double-slit.

For a single-slit, we get what appears to be a gaussian function. This is expected according to a classical wave, as we expect no interference, but still diffraction around the edges of the slit. For the double-slit we get an interference pattern, although we would expect a greater amount of smaller peaks further along the y -axis. This is likely because the wave spreads out in a circular pattern, while our screen is flat and captures the wave at only a single moment in time. Looking at FIG 11, we could clearly expect the lowest peaks being higher if the screen was measuring over a time interval. The same argument can be made for the triple-slit case.

Continuing, we can see that the single and double-slit produce clear peaks on the center of the screen. For the triple-slit, we see however a clear gap on the center of the screen. This is a clear proof of interference from diffraction, as both the single and triple slits have an

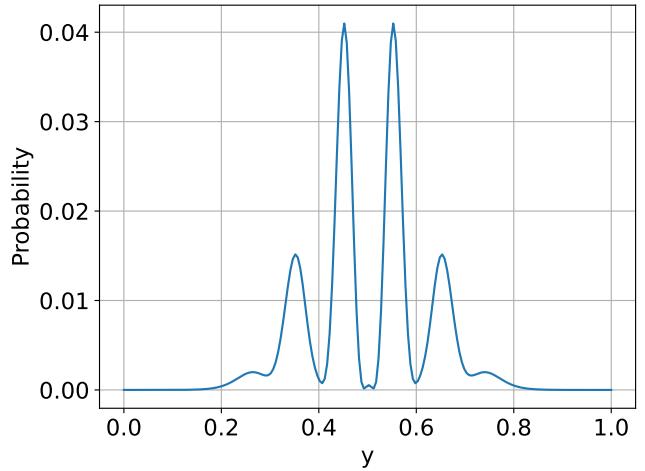


FIG. 16. Probability of finding the particle along a screen at $x = 0.8, t = 0.002$, using a triple-slit.

opening on the center of the potential. The double slit does however not have an opening on the center, but rather openings above and below the center, yet we see a clear peak in probability on the center of the screen. This follows what we saw in the previously shown illustrations.

D. Further studies

This study has not attempted to compare the numerical results with experimental nor analytical results beyond visual agreement of the diffraction pattern. A clear next step for further study of the simulation is to compare the numerical results to experimental results. Further studies could explore the boundaries of the simulation: adding a more realistic potential, exploring new compositions of the slits, and also expanding the simulation to three dimensions. This will allow for a more versatile simulation with a better understanding of its limitations and usages.

IV. CONCLUSION

We have used the Crank-Nicolson method applied to the Schrödinger equation with Dirichlet boundary conditions to study the many-slit experiment in 2D. For testing the conservations of probability over time, we used 0 and 2 slits. And later we used 1, 2, and 3 slits to illustrate the wave nature of a quantum particle. We have seen that the Crank-Nicolson method conserves probability to an order of 10^{-14} over a total time $T = 0.010$ using a timestep $\Delta t = 2.5 \cdot 10^{-5}$. Visually, we have shown how the wave function experiences diffraction and interference similar to a classical wave. We have simulated the many-slit experiment for 1, 2, and 3 slits, using a screen placed behind the slits at $x = 0.8$, measuring at time $t = 0.002$.

We saw that the probability of finding the particle on the screen behind the slits take an expected distribution according to the diffraction and interference of classical waves. Further study of the model with direct compar-

isons to experimental results or further applications of the model with different potentials could prove useful to explore the limitations of the model.

- [1] Wikipedia, Double-slit experiment (2024), https://en.wikipedia.org/wiki/Double-slit_experiment. Last accessed 7th of december 2024.
- [2] R. C. Conrad Sanderson, Armadillo: a template-based c++ library for linear algebra., Journal of Open Source Software **1**, 7 (2016).
- [3] H. Haug, A. H. Klevan, and L. L. Storborg, Project 5, <https://github.uio.no/heh/FYS3150/tree/main/Project5>.
- [4] J. D. Hunter, Matplotlib: A 2d graphics environment, Computing in Science & Engineering **9**, 90 (2007).
- [5] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Array programming with NumPy, Nature **585**, 357 (2020), <https://doi.org/10.1038/s41586-020-2649-2>.
- [6] T. pandas development team, pandas-dev/pandas: Pandas (2020), <https://doi.org/10.5281/zenodo.3509134>.
- [7] M. L. Waskom, seaborn: statistical data visualization, Journal of Open Source Software **6**, 3021 (2021).

Appendix A: Discretizing the wave function according to the Crank-Nicolson method

Using $\theta = \frac{1}{2}$, our Crank-Nicolson method becomes

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{1}{2} (F_{ij}^{n+1} + F_{ij}^n). \quad (\text{A1})$$

We wish to apply this to our SE.

Using our ‘bare’ SE (5), we can define

$$\begin{aligned} \frac{\delta \mathbf{u}}{\delta t} &= -i \left[-\frac{\delta^2 \mathbf{u}}{\delta x^2} - \frac{\delta^2 \mathbf{u}}{\delta y^2} + \mathbf{v} \mathbf{u} \right] \\ &= -i \left[-\frac{\delta^2}{\delta x^2} - \frac{\delta^2}{\delta y^2} + \mathbf{v} \right] \mathbf{u}, \\ \frac{\delta u_{ij}^n}{\delta t} &= -i \left[-\frac{\delta^2}{\delta x^2} - \frac{\delta^2}{\delta y^2} + v_{ij}^n \right] u_{ij}^n, \end{aligned} \quad (\text{A2})$$

where we know the definition of the discretized double derivative from equation A3

$$\begin{aligned} \frac{\delta^2 u_{ij}^n}{\delta x^2} &= \frac{u_{i+1,j}^n - 2u_{ij}^n + u_{i-1,j}^n}{h^2}, \\ \frac{\delta^2 u_{ij}^n}{\delta y^2} &= \frac{u_{i,j+1}^n - 2u_{ij}^n + u_{i,j-1}^n}{h^2}. \end{aligned} \quad (\text{A3})$$

We will keep this in mind, and apply it later for practical reasons, as these are long expressions. For the same reason, we define

$$\nabla^2 = \frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2}. \quad (\text{A4})$$

We continue by remembering equation (11), such that

$$\begin{aligned} F_{ij}^n &= \frac{\delta u_{ij}^n}{\delta t}, \\ F_{ij}^{n+1} &= \frac{\delta u_{ij}^{n+1}}{\delta t}. \end{aligned}$$

With this, we can fill in equation (A1) and simplify.

$$\begin{aligned} \frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} &= \frac{1}{2} (F_{ij}^{n+1} + F_{ij}^n) \\ &= \frac{1}{2} (-i [-\nabla^2 + v_{ij}] u_{ij}^{n+1} - i [-\nabla^2 + v_{ij}] u_{ij}^n) \end{aligned}$$

Where we want all the u_{ij}^n on the left side, and u_{ij}^{n+1} on the right side.

$$\begin{aligned} u_{ij}^{n+1} + \frac{i\Delta t}{2} [-\nabla^2 + v_{ij}] u_{ij}^{n+1} &= u_{ij}^n - \frac{i\Delta t}{2} [-\nabla^2 + v_{ij}] u_{ij}^n \\ u_{ij}^{n+1} + \left[-rh^2 \nabla^2 + \frac{i\Delta t}{2} v_{ij} \right] u_{ij}^{n+1} &= u_{ij}^n - \left[-rh^2 \nabla^2 + \frac{i\Delta t}{2} v_{ij} \right] u_{ij}^n. \end{aligned} \quad (\text{A5})$$

Where we have defined $r = \frac{i\Delta t}{2h^2}$. Filling in for ∇^2 we get our final expression in equation (A6).

$$\begin{aligned} u_{ij}^{n+1} - r [u_{i+1,j}^{n+1} - 2u_{ij}^{n+1} + u_{i-1,j}^{n+1}] \\ - r [u_{i,j+1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j-1}^{n+1}] + \frac{i\Delta t}{2} v_{ij} u_{ij}^{n+1} \\ = u_{ij}^n + r [u_{i+1,j}^n - 2u_{ij}^n + u_{i-1,j}^n] \\ + r [u_{i,j+1}^n - 2u_{ij}^n + u_{i,j-1}^n] - \frac{i\Delta t}{2} v_{ij} u_{ij}^n. \end{aligned} \quad (\text{A6})$$

Appendix B: Matrices

Where we define

The matrices when using $(M - 2) = 3$ will look according to equation (B1)

$$\mathbf{A} = \begin{bmatrix} a_0 & -r & 0 & -r & 0 & 0 & 0 & 0 & 0 \\ -r & a_1 & -r & 0 & -r & 0 & 0 & 0 & 0 \\ 0 & -r & a_2 & 0 & 0 & -r & 0 & 0 & 0 \\ -r & 0 & 0 & a_3 & -r & 0 & -r & 0 & 0 \\ 0 & -r & 0 & -r & a_4 & -r & 0 & -r & 0 \\ 0 & 0 & -r & 0 & -r & a_5 & 0 & 0 & -r \\ 0 & 0 & 0 & -r & 0 & 0 & a_6 & -r & 0 \\ 0 & 0 & 0 & 0 & -r & 0 & -r & a_7 & -r \\ 0 & 0 & 0 & 0 & 0 & -r & 0 & -r & a_8 \end{bmatrix}$$

$$a_k = 1 + 4r + \frac{i\Delta t}{2}v_{ij},$$

$$b_k = 1 - 4r - \frac{i\Delta t}{2}v_{ij}. \quad (\text{B2})$$

$$\mathbf{B} = \begin{bmatrix} b_0 & r & 0 & r & 0 & 0 & 0 & 0 & 0 \\ r & b_1 & r & 0 & r & 0 & 0 & 0 & 0 \\ 0 & r & b_2 & 0 & 0 & r & 0 & 0 & 0 \\ r & 0 & 0 & b_3 & r & 0 & r & 0 & 0 \\ 0 & r & 0 & r & b_4 & r & 0 & r & 0 \\ 0 & 0 & r & 0 & r & b_5 & 0 & 0 & r \\ 0 & 0 & 0 & r & 0 & 0 & b_6 & r & 0 \\ 0 & 0 & 0 & 0 & r & 0 & r & b_7 & r \\ 0 & 0 & 0 & 0 & 0 & r & 0 & r & b_8 \end{bmatrix}. \quad (\text{B1})$$

Where k is the index along the diagonal of the matrix.