Numerical Simulation of Quantum Particle Diffraction

Cecilie S. Rønnestad and Vaitian L. Marianayagam

(Dated: December 12, 2024)

We have simulated the evolution of a quantum particle travelling through one, two, and three slits. We have implemented a detector screen to explore the detection probability of the particle. The wave function showed interference patterns in accordance with observational data. The total probability remained constant over time, with a tolerance of 10^{-13}, specifically for the no slit and double-slit case.

I. INTRODUCTION

The double-slit experiment is one of the most well-known experiments within quantum mechanics. It provides valuable understanding of the wave-particle duality of matter by showing that particles can produce wave-like interference patterns, or behave like individual particles, depending on the experimental setup.

We want to simulate variations of the single-particle double-slit experiment (inside a box) by numerically solving the time-dependent Schrödinger equation in two dimensions. Due to the equation's complexity, we need to rely on numerical methods for solving it. We will discretize the Schrödinger equation using the Crank-Nicolson method. This implicit finite-difference scheme is widely used in numerical physics due to its combination of stability and accuracy.

In Section II we introduce the time-dependent Schrödinger equation and discretise it in two spatial dimension and one temporal dimension by applying the Crank-Nicolson method. We express the equation in matrix form and formulate an algorithm for solving it. In the end we introduce some important parameters for the simulation. Our results, which are focused on quantum probabilities, are presented and discussed in Section III. Finally, in Section IV we give a brief summary of our main results. The code used to create all our results is available at our GitHub repository[1].

II. METHODS

Solving the Schrödinger Equation

Quantum mechanical objects are described mathematically by their quantum state $|\Psi\rangle$, often represented by its complex valued wave function $\Psi(\vec{r},t)$. The state contains all available information about the object. Notably, by the Born rule, the probability of observing the particle in a volume d^3r at time t is $|\Psi(\vec{r},t)|^2 d^3r$. The total probability is required to be 1: $\int |\Psi|^2 d^3r = 1$. If a wave function is normalised at any point in time, it will necessarily be normalised at any other point in time.

The time evolution of the state is governed by the Schrödinger equation,

$$i\hbar\frac{\partial}{\partial t}\Psi(\vec{r},t) = \hat{H}\Psi(\vec{r},t),$$

where i is the imaginary unit and $\hbar$ is the reduced Planck's constant. $\hat{H}$ denotes the Hamiltonian operator,

$$\hat{H} = -\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r},t),$$

where m is the particle mass, ∇^2 is the Laplacian operator, $\nabla\cdot\nabla = \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\dots\right)$, and $V(\vec{r})$ is the potential subjected on the object.

Our simulations will be done in two spatial dimensions, and this will be assumed in the following. To make it easier to work with, we will rid the equation of it's dimensionality, giving us the following dimensionless Schrödinger equation:

$$i\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + v(\vec{r})u. \tag{1}$$

We will not include a rigorous derivation, but notice that $u(x,y,t)$ plays the role of Ψ.

TABLE I. Summary of notation used in the report.

Symbol	Description
x, y	Continuous variables in the domain $[0,1]$.
t	Continuous variable in the domain $[0,T]$.
h	Step size for x and y.
$x_i = ih$	Discretised x, with $i = 0,1,\dots,M-1$.
$y_j = jh$	Discretised y, with $j = 0,1,\dots,M-1$.
$t_n = n\Delta t$	Discretised t, with $n = 0,1,\dots,N_t-1$.
u_{ij}^n	Discretised solution $u(ih, jh, n\Delta t)$.
U^n	Matrix with elements u_{ij}^n.
v_{ij}	Discretised potential $v(ih, jh)$.

To solve Equation (1) numerically, we must discretise it, and choose boundary conditions and an initial wave function. The notation used when discretising can be seen in Table I. We will consider a free particle inside an impenetrable square wall of size 1×1. This implies Dirichlet boundary conditions, where $u(x,y,t)=0$ when either x or y is 0 or 1. We discretise both x and y into M evenly spaced values on $[0,1]$. The area inside the wall

[1] https://github.uio.no/vaitianm/quantum_diffraction.git

can now be represented as an $(M-2)\times(M-2)$ coordinate matrix. We will use the notation $u(x_i, y_j, t_n) \equiv u^n_{ij}$. The normalisation condition of u is then

$$\sum_{i,j} u^{0*}_{ij} u^0_{ij} = 1, \tag{2}$$

where we are interpreting $u^*_{ij}u_{ij}$ as the probability of the particle being observed on the grid cell i, j with area h^2, $h \equiv 1/M$, rather than a probability density.

To further discretise Equation (1), we will use the Crank-Nicholson approach: The partial time derivative is approximated by the simple finite difference,

$$\frac{\partial u}{\partial t} \to \frac{u^{n+1} - u^n}{\Delta t},$$

and the spatial second derivatives are approximated as the sum of the forward and backward differences, weighted equally:

$$\nabla^2 u \to \frac{1}{2}\left[F^{n+1}_{ij} + F^n_{ij}\right]$$

where

$$F^{n+1}_{ij} = \frac{u^{n+1}_{i+1,j} - 2u^{n+1}_{i,j} + u^{n+1}_{i-1,j}}{h^2} + \frac{u^{n+1}_{i,j+1} - 2u^{n+1}_{i,j} + u^{n+1}_{i,j-1}}{h^2}$$

and

$$F^n_{ij} = \frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{h^2} + \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{h^2}.$$

The same treatment is given to the potential term:

$$v(\vec{r})u \to \frac{1}{2}\left[v_{ij}u^{n+1}_{ij} + v_{ij}u^n_{ij}\right]$$

Applying these transformations on (1) together with some simple algebraic manipulation and introducing the constant $r \equiv \frac{i\Delta t}{2h^2}$, yields

$$\begin{aligned} &u^{n+1}_{ij} - r\left[u^{n+1}_{i+1,j} - 2u^{n+1}_{ij} + u^{n+1}_{i-1,j}\right] \\ &- r\left[u^{n+1}_{i,j+1} - 2u^{n+1}_{ij} + u^{n+1}_{i,j-1}\right] + \frac{i\Delta t}{2} v_{ij} u^{n+1}_{ij} \\ &= u^n_{ij} + r\left[u^n_{i+1,j} - 2u^n_{ij} + u^n_{i-1,j}\right] \\ &+ r\left[u^n_{i,j+1} - 2u^n_{ij} + u^n_{i,j-1}\right] - \frac{i\Delta t}{2} v_{ij} u^n_{ij}. \end{aligned} \tag{3}$$

By expressing u_{ij} as a column vector with elements u_k, and taking the Dirichlet boundary conditions into account, Equation (3) can be expressed in matrix form as

$$A\vec{u}^{n+1} = B\vec{u}^n. \tag{4}$$

Assuming u to be non-zero on an $N \times N$ grid, the conversion from indices i, j to k is done in column-major order by the relation

$$k = Nj + i, \tag{5}$$

as illustrated in Figure 1.

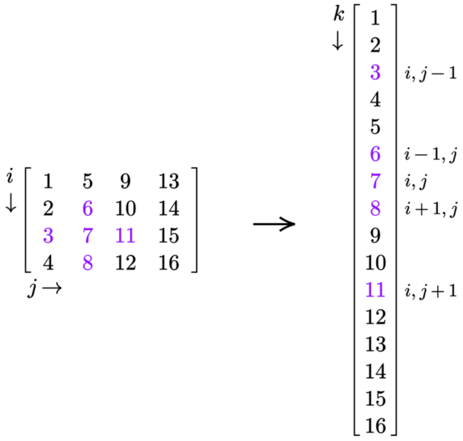

FIG. 1. A 4×4 matrix flattened according to Equation (5). Notice the placement of adjacent elements after the transformation. Figure taken from [1].

To retrieve Equation (3) from (4), we see that the diagonal elements a_k of A and b_k of B must be

$$\begin{aligned} a_k &= 1 + 4r + \frac{i\Delta t}{2} v_{ij}, \\ b_k &= 1 - 4r - \frac{i\Delta t}{2} v_{ij}. \end{aligned} \tag{6}$$

Furthermore, notice that $u_{i\pm1,j} = u_{k\pm1}$, so to encode the terms $-r\left[u^{n+1}_{i+1,j} + u^{n+1}_{i-1,j}\right]$ from the left hand side of the equation and $+r\left[u^n_{i+1,j} + u^n_{i-1,j}\right]$ from the right, the sub- and super-diagonal of B and A must be $\pm r$, respectively. However, $u_{k=nN-1}$ and $u_{k=nN}$, where n is an integer, will represent a grid point next to the boundary, and as such $u_{i+1,j} = 0$ when $k = nN$, and $u_{i-1,j} = 0$ when $k = nN + 1$. This will correspond to a zero at every N'th element in the sub- and super-diagonal. Lastly, from Equation (5), we see that $u_{i,j\pm1} = u_{k\pm N}$. Thus, diagonals N and $-N$ (taking the principal diagonal to be number 0) must similarly be filled with $\pm r$ to include the terms $\pm r\left[u_{i,j+1} + u_{i,j-1}\right]$. These diagonals do not need any zeroes spread throughout, since all the boundary adjacent points will be at the beginning and end of the matrix, so the points will naturally be excluded. An example of the matrices A and B in the case of $N = 3$

Algorithm 1 Solving the Schrödinger equation, Crank-Nicholson scheme

```
procedure C-N_Solver($A, B, \vec{u}, T, dt$)
    $t_N = T/dt$                                ▷ Number of time steps
    for $t_n = 1, 2, \ldots, t_N$ do
        $\vec{b} \leftarrow B\vec{u}$
        find $\vec{x} | A\vec{x} = \vec{b}$
        $\vec{u} \leftarrow \vec{x}$
```

can be seen below.

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

$$B = \begin{bmatrix} b_0 & r & 0 & r & 0 & 0 & 0 & 0 & 0 \\ r & b_1 & r & 0 & r & 0 & 0 & 0 & 0 \\ 0 & r & b_2 & 0 & 0 & r & 0 & 0 & 0 \\ r & 0 & 0 & b_3 & r & 0 & r & 0 & 0 \\ 0 & r & 0 & r & b_4 & r & 0 & r & 0 \\ 0 & 0 & r & 0 & r & b_5 & 0 & 0 & r \\ 0 & 0 & 0 & r & 0 & 0 & b_6 & r & 0 \\ 0 & 0 & 0 & 0 & r & 0 & r & b_7 & r \\ 0 & 0 & 0 & 0 & 0 & r & 0 & r & b_8 \end{bmatrix}$$

For the initial wave function, $u(x, y, t = 0)$, we will use a normalised Gaussian wave packet,

$$u(x_i, y_j, t = 0) = ce^{-\frac{(x_i - x_c)^2}{2\sigma_x^2} - \frac{(y_j - y_c)^2}{2\sigma_y^2} + ip_x x_i + ip_y y_j}, \quad (7)$$

choosing $c \in \mathbb{R}$ such that the condition from Equation (2) is met.

With A, B and u in place, the simulation algorithm is neatly summarized in Algorithm 1. The input values T and dt are total time of simulation and step size in time. Notice that A is a sparse matrix, so to solve $A\vec{x} = \vec{b}$, a specialised solver for sparse matrix equation will be much faster. We have used the superLU solver [2], based on LU decomposition. At the end of the loop at every time step, we saved either the wave function at that time, or the deviation of total probability from 1.

Simulations

In our simulations we will have our particle inside a box. To make this box, we will use the parameters given in Table II.

Slit setup. Single-, double- and triple-slit. Deviation

TABLE II. Parameters for the slit setup.

Parameter	Value
Wall thickness in x-direction	0.02
Wall centre in x-direction	0.5
Length of wall piece between slits	0.05
Slit aperture in y-direction	0.05
Symmetry around y	0.5

Our most important simulations parameters are given in Table III, with values given for the simulation without any slits. For our first double-slit case, we will set v_0 to $1 \cdot 10^{1}0$ and $\sigma_y = 0.10$. For both these cases, we will plot the probability deviation from 1. For our last simulations, we will use $T = 0.002$, $\sigma_y = 0.10$ and $v_0 = 1 \cdot 10^{1}0$.

TABLE III. Simulation parameters for the box without any slits.

Parameter	Value	Description
h	0.005	Step size for x and y.
Δt	$2.5{\cdot}10^{-5}$	Step size for t.
T	0.008	Total time of simulation.
x_c	0.25	x-value of the centre of the initial wave function.
σ_x	0.05	Std in x-direction of initial wave function.
p_x	200	Momentum in x-direction.
y_c	0.5	y-value of initial wave funtion centre.
σ_y	0.05	Std in y-direction of initial wavee function.
p_y	0	Momentum in y-direction.
v_0	0	Potential inside the central barrier.

Tools

All plots in this report were created using the Python library `matplotlib` [3]. In the plotting process, we also used the Python library `numpy` [4] to create arrays. In addition, we used the C++ library `armadillo` [5] for all vector operations and to store data in matrix form.

III. RESULTS AND DISCUSSION

Deviation of Total Probability

A plot of the deviation of the total probability from 1.0 as a function of time can be seen in Figure 2. The simulation is run for $t \in [0, 0.008]$ with no barrier in the box, and with a double slit. There does not seem to be a systematic increase in deviation, but rather unpredictable fluctuations around 0. The fluctuations of

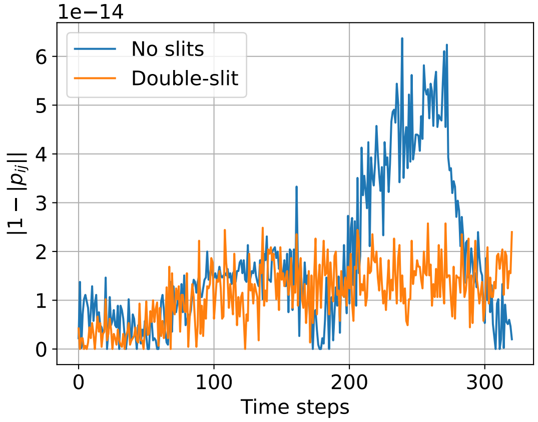

FIG. 2. Deviation from 1 of total probability at every time step.

the total probability is on the order of 10^{-14}. Since the machine precision of the C++ double is 10^{-15}, this is probably the source of the deviation. Using a tolerance of 10^{-13}, the simulation is thus in accordance with the theoretical time independence of normalisation. There is no reason to believe that the deviation will be any larger in the rest of the simulations, especially as the simulation times will be shorter.

Time Evolution of the Wave Function

Figure 3 shows the time evolution of u_{ij}^n in a double slit configuration through three still pictures at different times. Note that this is the square root of the probability function p_{ij}^n. An animation can be seen in the GitHub repository[2]. The wave function shows a clear interference pattern after going though the slits. The reflected part also shows an interference pattern. The wavelike nature of the wave function is even clearer seen in Figures 4 and 5. We can see how the interference pattern comes from superposition of peaks and troughs. Even though the magnitude of the wave function, and hence, the probability function, initially is Gaussian, the wave function itself is a sinusoidal wave packet in a Gaussian envelope. This can be seen in the initial expression, Equation (7), which can be rewritten using Euler's formula, to a sinusoidal term multiplied by a Gaussian. We see from the figures that it is the Born rule that makes the probability function look like a split beam, rather that overlapping waves. Now, translating back to the probability function, we can imagine there to be a detector screen in place along the y-axis at $x = 0.8$, and that it detects a particle at $t = 0.002$. Figure 6 shows the probability of detecting the particle at different points along the y-axis, for both single, double, and triple slit configurations. The single slit configuration does not produce an interference pattern, while the two others does. The single slit probability function has a single maximum. All the probability functions are symmetrical around the center. This is expected, as the simulated system is completely symmetrical around $y = 0.5$.

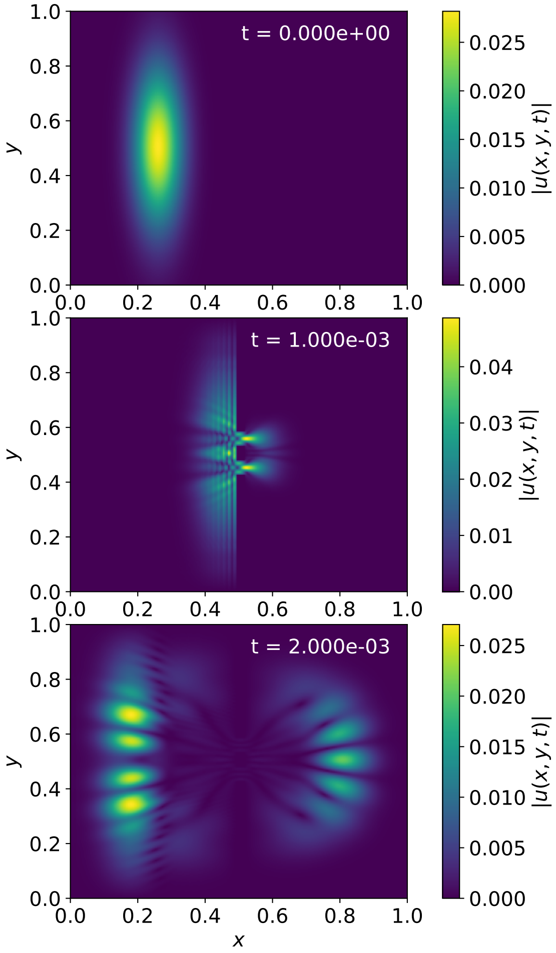

FIG. 3. Magnitude of $u(x, y, t)$ at three points in time with the double slit configuration.

Comparing this to experimental data, we can consider Figure 7. It shows the result of detecting single photons from a continuos He-Ne laser beam going through either single, double or triple slits. A, B and C are single slit configurations, AB, AC and BC are double slit configurations, and ABC is a triple slit configuration. The empirical result also shows single slit probabilities with a single central extremal point, and interference patterns from the double and triple slits. The exact pattern varies with slit dimensions and distance between the detector and the slits. The initial conditions are also different, as the experiment was done with a continuos beam, and not single particles. The AB and BC patterns shows a high

[2] https://github.uio.no/vaitianm/quantum_diffraction/blob/6d59bbe60ab2097780768e14bf466d083019ec5a/animation/animation.gif

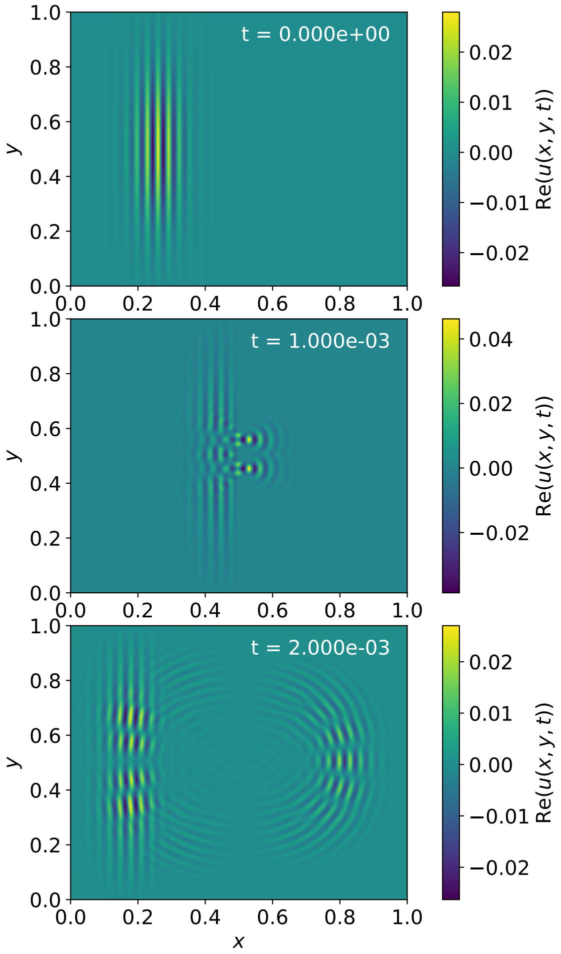

FIG. 4. Real part of $u(x, y, t)$ at three points in time with the double slit configuration.

central part, and five maxima, like our double slit result.

IV. CONCLUSION

Our simulations demonstrate the evolution of the two-dimensional Schrödinger wave function over time. The total probability remains effectively conserved within machine precision, and any minor deviations observed can be attributed to numerical limitations rather than a fundamental departure from theoretical expectations. By examining single, double, and triple slit configurations, we observe that the single slit setup produces a straightforward diffraction pattern with a single maximum, whereas introducing multiple slits leads to interference patterns featuring multiple distinct maxima. These simulated results qualitatively match empirical findings in empirical experiments. The close resemblance to experimental interference outcomes—despite differences in initial conditions and experimental arrangements—further validates the computational approach.

[1] Kvellestad, A. (2024). Lecture notes FYS3150 - Computational Physics, Fall 2024. https://github.com/anderkve/FYS3150/blob/master/lecture_notes/2024/lecture_notes.pdf

[2] Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., & Liu, J. W. H. (1999). A supernodal approach to sparse partial pivoting. SIAM Journal on Matrix Analysis and Applications, 20(3), 720–755. https://portal.nersc.gov/project/sparse/xiaoye-web/simax-29176.pdf

[3] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering, 9(3)*, 90-95.

[4] Harris, C. R., Millman, K. J., van der Walt, S. J. et al. (2020). Array programming with NumPy. *Nature 585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

[5] Curtin, R & Sanderson, C. (2019). Practical Sparse Matrices in C++ with Hybrid Storage and Template-Based Expression Optimisation. *Mathematical and Computational Applications*, Vol. 24, No. 3.

[6] Sinha, U., Couteau, C., Jennewein, T., Laflamme, R., & Weihs, G. (2008). Testing Born's Rule in Quantum Mechanics with a Triple Slit Experiment. arXiv:0811.2068 https://arxiv.org/pdf/0811.2068

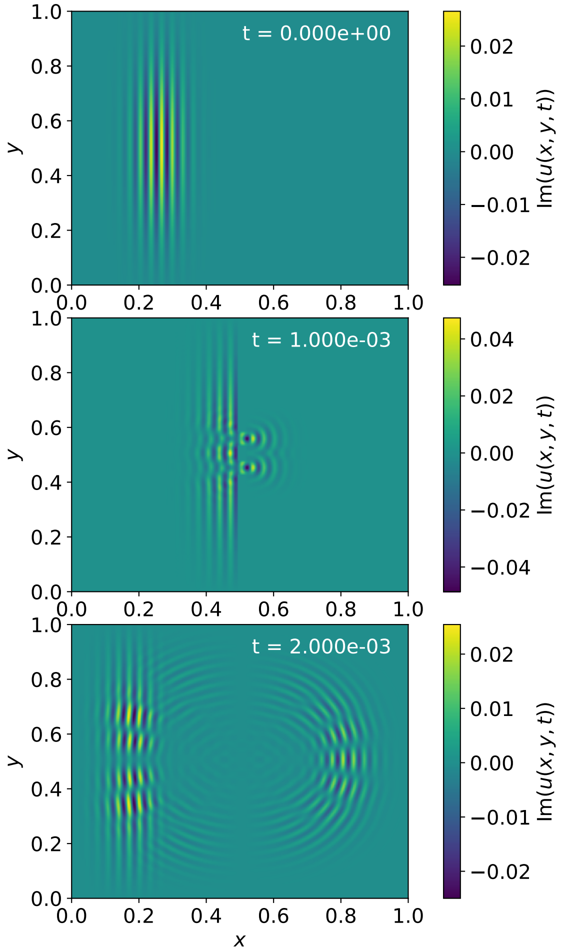

FIG. 5. Imaginary part of $u(x,y,t)$ at three values of t, with the double slit configuration.

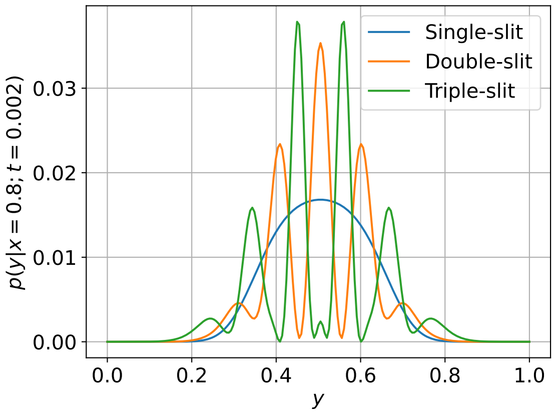

FIG. 6. Probability of detecting the particle along the y axis, given that it was detected at $x = 0.8$ at $t = 0.002$.

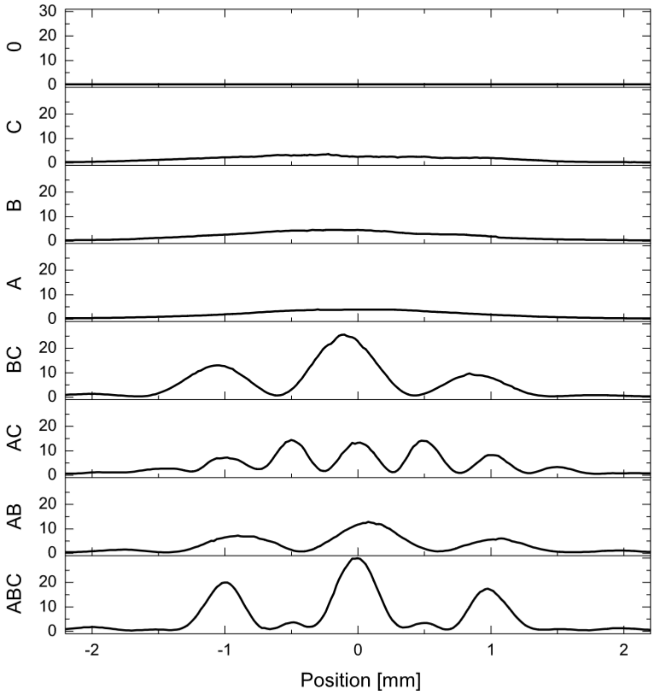

FIG. 7. Empirical results of the singe/double/triple slit experiment with photons. A, B and C are single slit configuration, AC and AB are double slit configurations, and ABC is the triple slit configuration. The y-axis is number of detected photons divided by 10^3. The Figure is taken from [6].