Evolution of a Simulated Quantum Particle

Campbell Timms

Aims

The idea for this project is to create a simulation of a quantum particle using the Schrödinger equation in free space. This project was completed computationally using numerical values plotted on an array and calculated. The aim was achieved, although with limitations, due to the approximations and discrete nature of the values used. The project resulted in a simulation that requires the particle to be smaller than the space it occupies and a high enough resolution for the calculations to be performed.

Background

To achieve the aim of this project and numerically predict the time-evolution of a wavefunction in two dimensions, we have to first define the wavefunction at t=0, and write code to solve the time-dependent Schrödinger equation and predict what the wavefunction is at later times.

The time-dependent Schrödinger equation has the form:

$$i\,\hbar\partial_t\psi = \frac{-\hbar^2}{2m}\,\nabla^2 + V\psi$$

(1) Where V is a potential, ψ is the wavefunction, ∇^2 denotes the Laplacian, m is the mass and $\hbar$ and i are both constants

Computers only store and handle discrete amounts of information. Before we discuss solving the equation, we need to represent space and the wavefunction on discrete grids (i.e. arrays).

Now that a wavefunction is defined at t=0, the next step is to solve the Schrödinger equation and predict the form of the wavefunction at later times.

A formal solution to the free-space equation is

$$e^{\frac{-i\hbar\Delta t}{2m}\nabla^2}\psi = e^{\frac{-i\Delta t}{\hbar}\left[\frac{-\hbar^2}{2m}\nabla^2\right]}\psi(t=0)$$

(2)

The operator $e^{\frac{-i\hbar\Delta t}{2m}\nabla^2}$ looks difficult, but there is a trick to evaluate it easily with Fourier transforms:

$$e^{\frac{-i\hbar\Delta t}{2m}\nabla^2}\psi(t) = F^{-1}\left[e^{\frac{-i\hbar\Delta t}{2m}q^2}F[\psi]\right]$$

(3) Where $F[\psi]$ and F^{-1} is the Fourier transformation and the inverse Fourier transformation operators, while q is the magnitude of the 2D reciprocal space vector.

Results/Discussion

What units did you use for all the parameters? What are the units for q? What are the units for sigma?

The parameters used were N, L, σ, k, $\hbar$, Δt and m. N refers to the size of the array used which graphically is the resolution of the image. For a fairly detailed image without it being too large to execute the program, I opted for a 1000x1000 image of 1,000,000 pixels. For all the units of the parameters I decided to use SI units in order to add some baring of the values used and some real-world constants like the mass of an electron. [2] L is a length in real space for which will set the values to within the range of 0 – L with L being in metres for SI units. Sigma (σ) appears in the definition of the Gaussian which we used to create the initial wavefunction.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

(4) Where $\frac{1}{\sigma\sqrt{2\pi}}$ is the normalisation constant N and μ is the expected value or more useful for us is the centre of the bell curve. Sigma (σ) in the terms of the Gaussian refers to the standard deviation and controls the width of the curve. [1]

In order to find the units to Sigma (σ) we must first solve for σ in terms of x and μ.

$$\sigma = \pm\sqrt{\frac{-2\ln(\frac{f(x)}{N})}{(x-\mu)^2}}$$

(5)

Now if we do some diminution analysis, replacing our parameters with our desired SI units we can find the units of σ in SI units. Both *x* and μ are space dimensions and have units of metres (m), so σ becomes:

$$\sigma = \pm\sqrt{\ln(m)}$$

(6)

So σ has units of the square root of the natural log of metres. This might seem a bit obscure but since sigma is mainly used in its variance from σ^2 in the Gaussian (4), the square root disappears and the units are the natural log of metres.

A few other of the parameters we used also map directly to SI units in the time dependant Schrödinger equation (1). Time which appears as ∂_t or when solving discretely $\partial_t \rightarrow \Delta t$ maps to seconds (s), while m which is the mass uses kilograms (kg). It should also be noted that h-bar appears in the time dependent Schrödinger Equation as well which has units of Joule seconds (J s).

Now all the parameters are defined with their SI units we can find the units of q which appears in equation (3). In the associated lab notes with this project [3] the q value is described as the "magnitude of the 2D reciprocal space vector". The magnitude of a 2 dimension vector is typically described by the following:

$$\|\boldsymbol{a}\| = \sqrt{a_1^2 + a_2^2}$$

(7) The magnitude of a two-dimensional vector $\boldsymbol{a} = (a_1, a_2)$ [4]

So per the definition we should expect the units to be $\frac{1}{m}$, which is inverse space or in SI unites inverse metres.

If we rearrange equation (3) we can see that:

$$F\left[\mathrm{e}^{\frac{-\mathrm{i}\hbar\Delta\mathrm{t}}{2\mathrm{m}}\nabla^2}\psi(t)\right] = F\left[\mathrm{e}^{\frac{-\mathrm{i}\hbar\Delta\mathrm{t}}{2\mathrm{m}}\mathrm{q}^2}\psi(t)\right]$$

(8)

We can see that the left-hand side of the equation looks similar to the right hand side where the only difference is the q^2 factor appears instead of the Laplacian. Removing the other variables and constants gives:

$$F[\nabla^2\psi(x,t)] = F[q^2\psi(x,t)]$$

(9)

This gives us a clearer image that this q^2 value is a result of the Laplacian acting on the wavefunction. Added also is the wavefunction being a function of both space and time, which $\psi(x,t)$ now reflects.

Since the wavefunction we are acting upon is the Gaussian (4) which has no time element and we take the Fourier transformation of the function we obtain:

$$\nabla^2\psi(x,t) = -\|\boldsymbol{q}\|^2\,\mathrm{F}[\psi(x,t)]$$

(10)

In this case $\|\boldsymbol{q}\|^2 = q^2$ and the brackets refer to the magnitude of a 2D vector like that is described by equation (7) above. This vector appears when taking the Fourier transformation of the Laplacian as a reciprocal space is used where the Laplacian is proportional to the negative of the squared magnitude of its Fourier transform in reciprocal space. This is seen as our $-\|\boldsymbol{q}\|^2$ factor. This is because differentiation has to be changed for a Fourier transform in place of a multiplication by the wave vector in reciprocal space. Since q is a magnitude of a wave vector based off the Gaussian (4), this shows that q has the units of space. With q being defined as the magnitude of the wave vector in reciprocal space, in regular space this would mean it would have units of $\frac{1}{m}$ in SI units.

Table 1: A table detailing all the values used for the following simulation in figure 1 and their associated units as discussed.

Symbol:	**Definition:**	**Value:**
N	Grid length	1000 cells
L	Real space constant	2 m
σ	Wavefunction width	$0.1\sqrt{\ln(\mathrm{m})}$
k	Spatial movement constant	100
$\hbar$	Reduced planks constant	$1.0545\text{x}10^{-34}$ J s
m	Mass	$0.00054858\text{x}1.6605\text{x}10^{-27}$ kg
Δt	Change in time	20 s

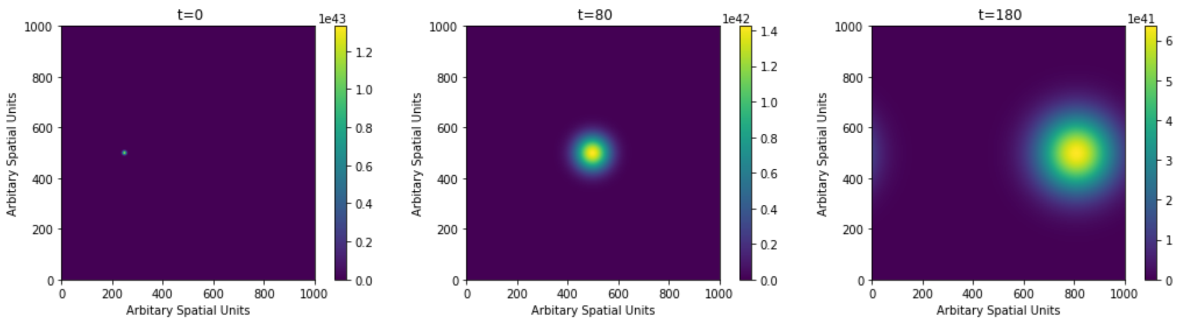

Figure 1: Evolution of a quantum particle over time at t = 0s, 80s and 180s with the above parameters in table 1. This particle exists in a space of arbitrary size and length.

What happens when the expanding wavefunction becomes larger than the size of the array? Is this physical?

By keeping the above settings and allowing the quantum particle to expand over a greater time changing Δt = 100s ,we can see what happens when the expanding wavefunction increases larger than the size of the array.

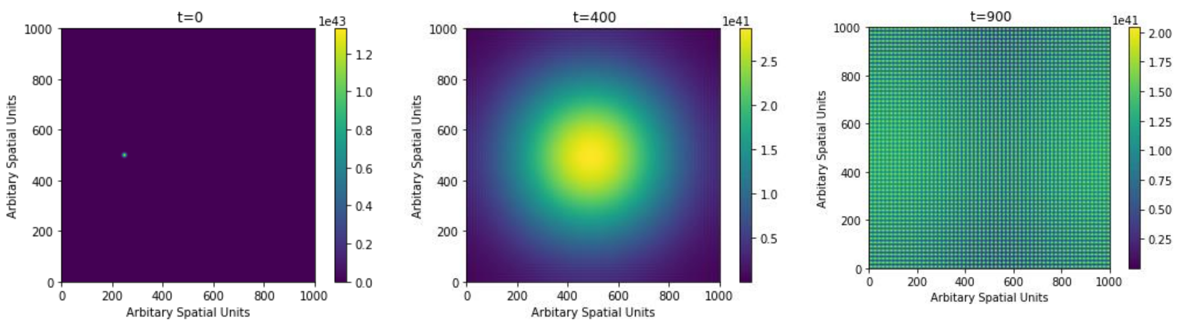

Figure 2: Evolution of a quantum particle over time at t = 0s, 400s and 900s with the above parameters in table 1, except for Δt.

As we can see in figure 2 in a particle which expands greater than the size of the array we start to see this repeating pattern. Why is this?

In setting up this wavefunction we gave it some movement in the space. As we can see in figure 1 as it evolves over time it also moves from left to right. In fact if we allow it to continue we can see that it exits the right side of the array and continues from the left side. In figure 3 below it is shown more clearly where at t=250s the particle is continuing from the left side of the array.

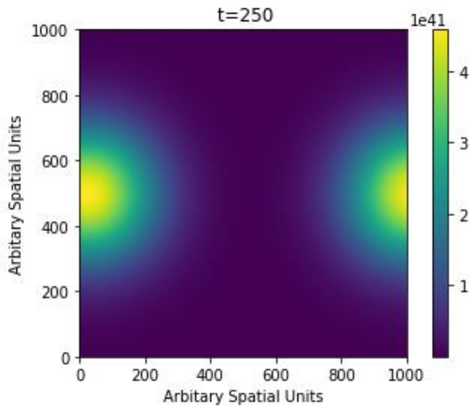

Figure 3: Evolution of a quantum particle over time at t = 250s with the above parameters in table 1.

So we can see that the boundaries of this grid do not act as a wall but exist as something that cosmologists call a closed universe. [6] This is the idea where if you continue going in the same direction you will eventually end up where you started. This theory of the universe is based on space curvature and is outside the scope of this report although the principle still applies for our simulation.

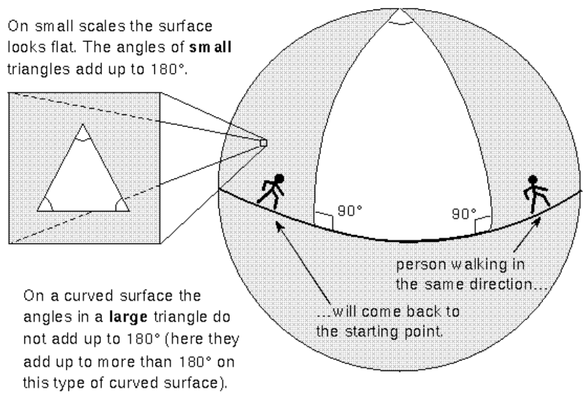

Figure 4: A diagram showing the features of how curved surface universes work unlike flat ones. [8]

Now that we have established what happens when things go past the boundaries of our array we can now see what happens when our expanding particle becomes much larger than the size of our array. It follows that the edge of the particle as it propagates outwards will eventually pass the array boundaries and appear on the other side of the array. There is an issue with this however because there is the other side of the particle which is also doing the same thing so the particle starts to interact with itself.

While I have been calling this quantum particle a particle, that is incorrect because more formally they are wave-particles and also exhibit wavelike properties. This was first shown by the Davisson–Germer[9] experiment showing electrons also behave similar to photons in the double slit experiment.

It is these wavelike properties that cause the grid like pattern to occur because it allows the wave-particle to interfere with itself and give this interference pattern once it is large enough.

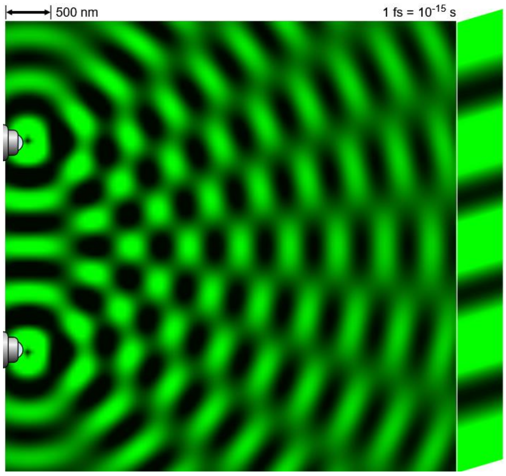

Figure 5: A diagram showing the interference of two photons which show wavelike properties similar to a quantum particle and to the right the interference pattern which we see displayed by a flat surface. [10]

To address if this physical, we unfortunately cannot get a quantum particle to interact with itself as it expands as we cannot create a closed universe, let alone one small enough for our particles. On the other hand, the current consensus among cosmologists is that we live in a flat universe, although that could change. [7] What do find is that quantum particles can interfere with each other with one unique form which is a property of quantum mechanics called quantum entanglement, which was first coined by Schrödinger in 1935. [11] Although quantum entanglement is a unique mechanism, it is its role on other quantum particles called quarks which make up protons, neutrons and electrons that exist in a 'stew' of entangled quarks which are constantly interfering with each other as they're held together by gluons. [12]

What boundary conditions are assumed by the discrete Fourier Transform? How do they affect the simulation? What happens if the number of pixels in the array is too small?

The discrete Fourier transform is defined by the following:

$$F_n = \sum_{k=0}^{N-1} f_k \, e^{2\pi\, i\, n\, k/N}$$

(11) Where $f_k == f(t_k)$ and $t_k = \Delta k$ *with* $k = 0, \ldots, N-1$ and $i = \sqrt{-1}$ [13]

For the discrete Fourier transform (DFT) to be made, the function f_k has to be periodic which means that the function repeats itself. For a wavefunction, this is no problem due to its wavelike properties. This also assumes that the function's boundaries are at 0 outside the range of the period taken.

The other important part about the function being periodic is that you need a reasonable change in the variable Δk, which changes the function from being continuous to discrete. If this interval is not periodic, which in a small enough part of the wavefunction is true, leakage occurs. [14]

Leakage is where data within one interval Δk leaks and effects neighbouring bin intervals. This then carries on as a numerical error in the following data. While this error cannot be exactly eliminated as our functions don't work in integer intervals, its effect should be small with a high enough resolution. This brings us to our next condition.

The other condition for us is that the DFT is finite and is determined by the number of elements in our array. If there aren't enough elements or points to work on then the DFT breaks or causes artificial patterns or artefacts.

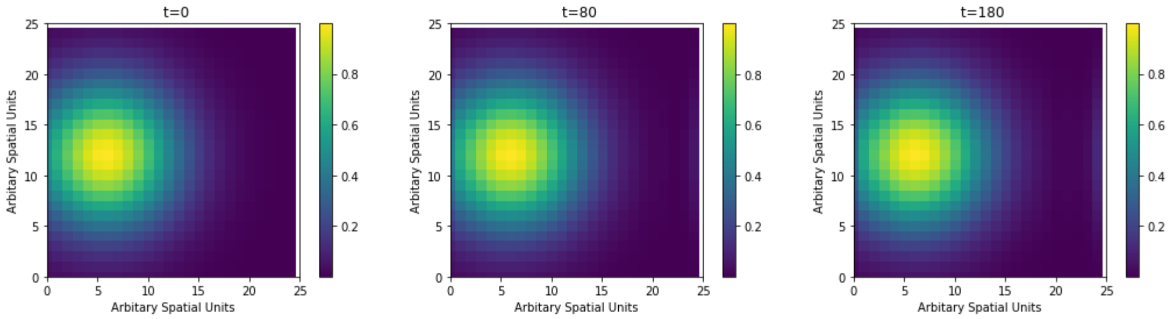

Figure 6: Evolution of a quantum particle over time at t = 0s, 80s and 180s with the above parameters in table 1, except for N=25.

As we can see in figure 6, where N=25 the simulation breaks and it does not expand over time as the DFT does not have a large enough data set. The particle is moving although very slowly and if we let it run for a long time again we can see the artificial interference patterns that it creates.

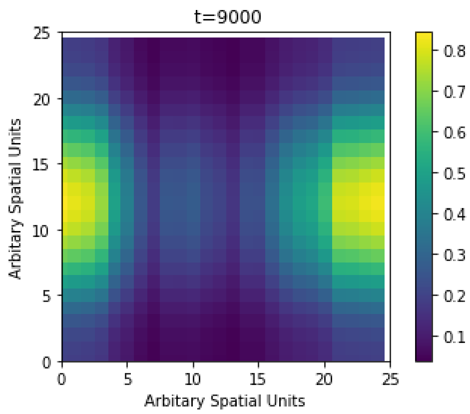

Figure 7: Evolution of a quantum particle over time at t = 9000s with the above parameters in table 1, except for N=25.

The resolution breaks the simulation as any approximation errors are carried through and are amplified later. Not to mention the lack of ability to make inferences based on the result as the approximations are too high.

If you look closely at figure 2, even with the high resolution at t=400 you can begin to see the fringing occur. This is not due to interference as mentioned before as it is still within the boundaries of the array, but is instead because of the discrete nature of the DFT with the numerical errors due to approximations, leakage and a poor discrete interval Δk.

As mentioned throughout the discussion, the accuracy of our simulation and results are limited by the discrete nature of our numerical calculations and the difference from a physical space to our simulation. For the most part the errors should be negligible, but only when the resolution of the array is high such as N=1000 and if the size of the particle remains within the size of the space.

The other errors which cannot be avoided computationally are transforming our theoretical equations and variables into ones that have to be discrete for the program to computationally give a numerical result. New equations which work on discrete sets of data and the methods used means there are artefacts and small errors when making assumptions about a value at a particular point.

Conclusions

The program that was created simulates the time evolution of an expanding quantum particle with the main result of this being shown in figure 1. However, the program is not without its limitations. We have also shown that due to the type of array we have and the numerical method used, for the best results we need to keep the size of the particle within the size of the array and have a high resolution. In the future we could possibly add a second quantum particle and see how they interact with each other. Another idea is changing the boundary conditions of the array and possibly giving it an infinite potential like the infinite square well or just a variable potential to see how quantum particles diffract or even tunnel. Lastly, we could add a function which controls the direction of motion and change the axis the particle moves along into a two dimensional grid for it to move around.

Appendix (printed code)

```
@author: Campbell Timms
"""

import numpy as np
import matplotlib.pyplot as plt

#Define values

N = 1000
# N = Number of pixels/grid size
L = 2
# L = real space constant
sigma = 0.1
#sigma value found in the Gaussian
delta_t = 20
#change in time
m = (0.0005485803*1.660566*10**-27)
#m-mass electron = 0.0005485803 (amu) * 1 amu =    1.660566E-27     kg
hbar = 1.054571817*10**-34
#constant hbar: 1.054 571 817...x 10-34 J s or 6.582 119 569...  x 10-
16 eV s
k = 100
#quickness in space
nplot = 10
#number of plots produced by code

xyarrays = np.mgrid[:N,:N]
x = xyarrays[0]
y = xyarrays[1]
X = (L)*(x - x[N-1,N-1]/2)
Y = (L)*(y - y[N-1,N-1]/2)

# Define the wave function as a Gaussian
r = np.sqrt(X**2+Y**2)
gaussian = np.exp(-r**2/(2*sigma**2))
wavefunction = np.array(gaussian, dtype=complex)

# Normalize the wavefunction
A_wavefunction = np.sum((wavefunction)**2)
N_constant = 1/np.sqrt(A_wavefunction)
N_wavefunction = wavefunction*N_constant
```

```
#Take the Fourier transform of the wavefunction at t=0.
F_wavefunction = np.fft.fft2(N_wavefunction)

xyarrays = np.mgrid[:N,:N]
q_x = xyarrays[0]
q_y = xyarrays[1]
q_X = (q_x - q_x[N-1,N-1]/2)
q_Y = (q_y - q_y[N-1,N-1]/2)

shift = N//2
q_xshift = np.roll( q_X, int(-shift), 0)
q_yshift = np.roll( q_Y, int(-shift), 1)

#Use the q-space arrays to make a complex array that stores the values
of the free space equation
qr2 = np.sqrt(q_xshift**2 + q_yshift**2)
Schro_time = np.exp((1j*hbar*delta_t/(2*m))*qr2**2)
qspace = np.array(Schro_time, dtype=complex)

# Multiply the Fourier transform of the wave function by the phase
factor
F_wavefunctiont = F_wavefunction*Schro_time

# Take the inverse Fourier transform.
wavefunctiont = np.fft.ifft2(F_wavefunctiont)

# Initialise the wave function to a Gaussian form times a linear phase
term
x0 = N/2
x_move = Y+x0
int_gaus = N_constant*np.exp((-(x_move**2+X**2)/2*sigma**2)+1j*k*Y)

#Evolve this wave function in time and plot the absolute value of the
wave function at several times
t=0

for i in range(nplot):
    delta_t=20
    delta_t = t + delta_t*i

    int_gaus = N_constant*np.exp((-(x_move**2+X**2)/2*sigma**2+1j*k*Y))
    Schro_time = np.exp((1j*hbar*delta_t/(2*m))*qr2**2)
    F_wavefunction_new = np.fft.fft2(int_gaus)
    F_wavefunctiont_new = F_wavefunction_new*Schro_time
    new_wavefunctiont = np.fft.ifft2(F_wavefunctiont_new)

    plt.imshow(abs(new_wavefunctiont))
    plt.title('' + ' ' + 't=' + str(delta_t))

    plt.xlim(0,N)
    plt.xlabel('Arbitary Spatial Units')

    plt.ylim(0,N)
    plt.ylabel('Arbitary Spatial Units')

    plt.colorbar()
    plt.show()
```

References

[1] Weisstein, E. W. (n.d.). *Gaussian Function*. Mathworld.wolfram.com. https://mathworld.wolfram.com/GaussianFunction.html

[2] *Atomic Weights and Isotopic Compositions for All Elements*. (2013). NIST (National Insitute of Standards and Technology). https://www.nist.gov/pml/atomic-weights-and-isotopic-compositions-relative-atomic-masses

[3] *PHYS2175 Quantum Physics - Computational Lab Part 1*. (2023).

[4] *Magnitude of a vector definition - Math Insight*. (2019). Mathinsight.org. https://mathinsight.org/definition/magnitude_vector

[5] *1.17: Quantum Mechanics and the Fourier Transform*. (2019, March 25). Chemistry LibreTexts. https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Quantum_Tutorials_(Rioux)/01%3A_Quantum_Fundamentals/1.17%3A_Quantum_Mechanics_and_the_Fourier_Transform

[6] Letzter,LiveScience, R. (n.d.). *Is the Universe a Giant Loop?* Scientific American. https://www.scientificamerican.com/article/is-the-universe-a-giant-loop/

[7] *How do we know the universe is flat? Discovering the topology of the universe*. (n.d.). Phys.org. https://phys.org/news/2017-06-universe-flat-topology.html

[8] *Cosmology*. (2023). Ufl.edu. https://users.astro.ufl.edu/~guzman/ast1002/class_notes/Ch17/Ch17.html

[9] Davisson, C. J., & Germer, L. H. (1928). Reflection of Electrons by a Crystal of Nickel. *Proceedings of the National Academy of Sciences of the United States of America*, *14*(4), 317–322. https://doi.org/10.1073/pnas.14.4.317

[10] *Wave Interference*. (2019). Colorado.edu. https://phet.colorado.edu/sims/html/wave-interference/latest/wave-interference_en.html

[11] Schrödinger, E., 1935. “Discussion of Probability Relations Between Separated Systems,” *Proceedings of the Cambridge Philosophical Society*, 31: 555–563; 32 (1936): 446–451.

[12] Pappas, S. (n.d.). *Scientists See Quantum Intersference between Different Kinds of Particles for First Time*. Scientific American. Retrieved April 4, 2023, from https://www.scientificamerican.com/article/scientists-see-quantum-interference-between-different-kinds-of-particles-for-first-time/

[13] Weisstein, E. W. (n.d.-a). *Discrete Fourier Transform*. Mathworld.wolfram.com. https://mathworld.wolfram.com/DiscreteFourierTransform.html

[14] Weisstein, E. W. (n.d.-c). *Leakage*. Mathworld.wolfram.com. https://mathworld.wolfram.com/Leakage.html