Canadian Journal of Pure and Applied Sciences
Vol. 15, No. 3, pp. 5333-5340, Oct 2021
Online ISSN: 1920-3853; Print ISSN: 1715-9997
Available online at www.cjpas.net

THE SHAPE OF A PHOTON

Christopher C. O'Neill
University College Dublin (UCD), Belfield, Dublin, Ireland

ABSTRACT

This paper deals with partial confirmation of the shape of a photon via a new kind of mathematics, where imaginary numbers are privileged over real numbers when an absolute value is given, by the way of the XNOR logic gate. The research begins with polynomials and then progresses from a planar wave form to a final and more accurate radial wave form. This research study reveals the mechanism behind the results of the Double Slit experiment.

Keywords: Photon shape, logic operators, complex numbers.

INTRODUCTION TO MULTI-DIMENSIONAL SPACE

Imaginary numbers are invaluable in many areas of mathematics, physics, and engineering. But in general, they are abstract entities with no real-world analog. This, however, is not true, in the strange world of Quantum Mechanics, where imaginary numbers take on a real and – in some sense – measurable significance. Recent developments in our understanding of imaginary numbers have led people to equate the Real number system with the XOR logic gate and the Imaginary number system with XNOR (Gode, 2014; Hayes, 2018; O'Neill, 2021).

In quantum mechanics, imaginary numbers are used to explain and represent the fact that a particle's momentum and position are two different and interlinked properties of the particle, for instance,

$$\Psi(r,t) = Ae^{i(kr-\omega t)}$$

In this most simple example of a plane wave, we see the variable '*i*' as an exponent value. Knowing the momentum of a particle exactly, in this equation, results in its position being everywhere at once, and vice versa. To avoid running into these infinities, these two properties can only ever be known approximately. This level of uncertainty is a consequence of the Schrödinger Equation and the Heisenberg Uncertainty Principle and is a direct consequence of the imaginary number or '*i*'.

To demonstrate this relationship between momentum and position, the imaginary part of the wave function must be raised to its power. This cancels out the imaginary part of the equation to give the absolute value. A similar result can be achieved with XNOR (!Δ) and XOR (Δ). This is because XOR and XNOR are noncommutative, !Δ.Δ + Δ.!Δ= 0, just like the Quaternions and Octonions. By privileging the XNOR value, we can obtain a Real value for Complex numbers. Note that this value is different to the absolute square mod and leads to different physical results. Therefore, it is important not to get them confused with one another. The equation must be run twice; once in XNOR and then in XOR, and the results of both equations are summed together.

Using a simple polynomial example, we obtain

$$(x!\Delta + y\Delta)(x!\Delta + y\Delta)$$
$$a: -x!\Delta^2 - 2x!\Delta y\Delta - y\Delta$$
$$b: x!\Delta^2 + 2x!\Delta y\Delta + y\Delta$$
$$a + b = 0$$

Using this polynomial equation, it is possible to graph various surfaces in both XOR and XNOR spaces. The XOR universe has three spatial dimensions. The fourth spatial dimension is the XNOR imaginary axis. When the three coordinates of the real axes are combined with the fourth imaginary axis, they always cancel each other out. This cancellation results in the flat plane that denotes the particle being in all positions or momentums at once.

Plotting the above example in dimensions 'Δ, Δ, !Δ' gives the result shown in Figure 1. Whereas plotting 'Δ, Δ, Δ, !Δ', which is a fourth-dimensional coordinate, results in the flat plane shown in Figure 2.

This achieves the same results as the complex quantum mechanical equation with a minimum of effort. These

Corresponding author e-mail: chris.ozneill@gmail.com

new multiplicative rule sets are called ‘Dimensional Gate Operators’ (DGO) because they express dimensionality through operators based on the logic gates.

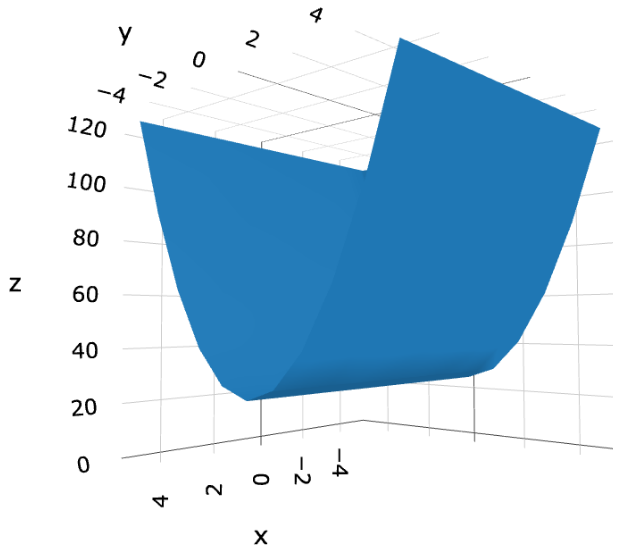


Fig. 1. The quadratic curve $(\Delta z + \Delta y + !\Delta x)^3$.

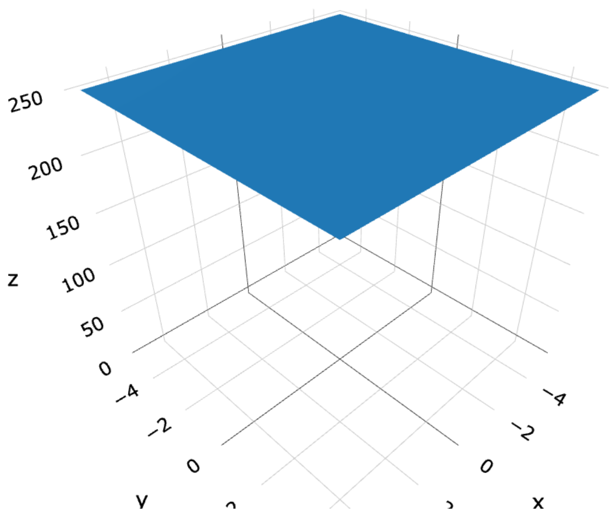


Fig. 2. Infinite plane $(\Delta w + \Delta z + \Delta y + !\Delta x)^3$.

A Better Example and Interference

When graphing the position or momentum of a particle, it is more accurate to use waveforms than mere polynomials. This waveform represents the exact momentum of a particle at time t. Once again, applying the XNOR and XOR gates to this returns the position value for this wave function (Fig. 3). Since it is possible to graph the XOR momentum-space component, it is a trivial matter to calculate the XNOR position-space component, which will simply be its inverse. This scenario is shown in Figure 4.

Both wave functions have so far been graphed separately. It is possible now to graph them simultaneously, see Figure 5. This is something which is ordinarily forbidden in quantum mechanics.

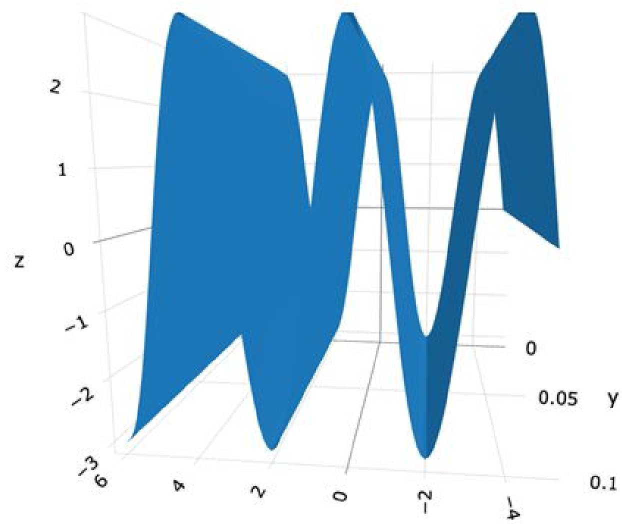


Fig. 3. Cos(Δx) waveform for the momentum of a particle.

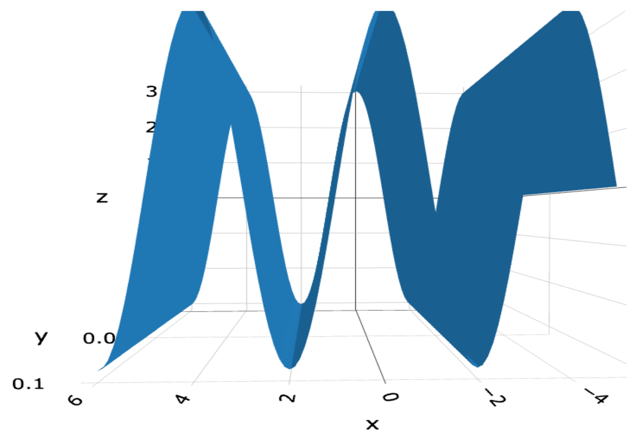


Fig. 4. Cos($!\Delta x$) waveform for the position of a particle.

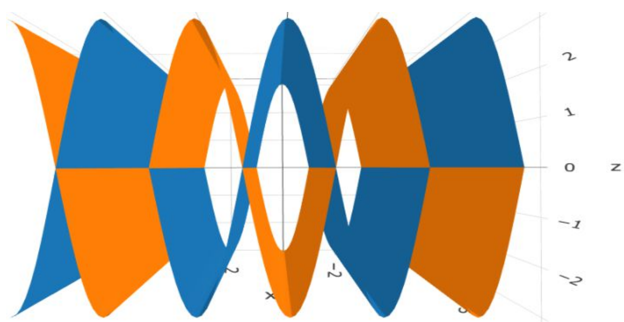


Fig. 5. The exact momentum and position-state superimposed.

From this perspective, it becomes apparent why the Heisenberg Uncertainty Principle exists. The two wave functions cancel out and what is left is the infinite value plane in the fourth dimension (Fig. 6). This cancellation is like that of orthogonality in linear algebra. Although this waveform is only a toy model, it reveals how particles look, when viewed from both sides of the XOR and XNOR axes simultaneously.

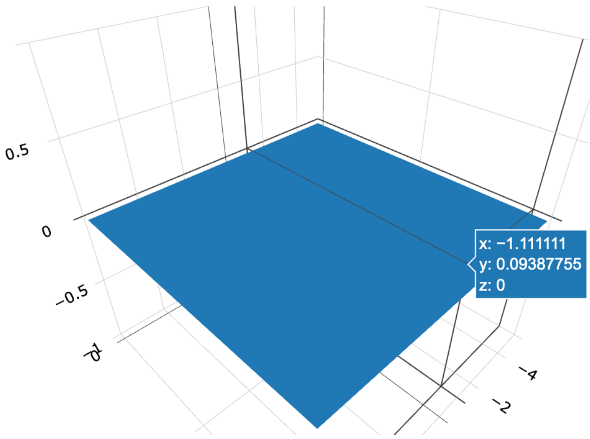


Fig. 6. The cancelled out infinite plane.

A More Complex Example

So far, a plane sine wave function has been employed to describe fundamental particles. If a truer understanding is to be had, it will be necessary to progress to a more accurate model of the photon. For this, the author has chosen initially the following Mexican Hat function:

$$z = \frac{\sin\left(\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}}$$

This function $z(x, y)$ produces the graph in XOR shown in Figure 7.

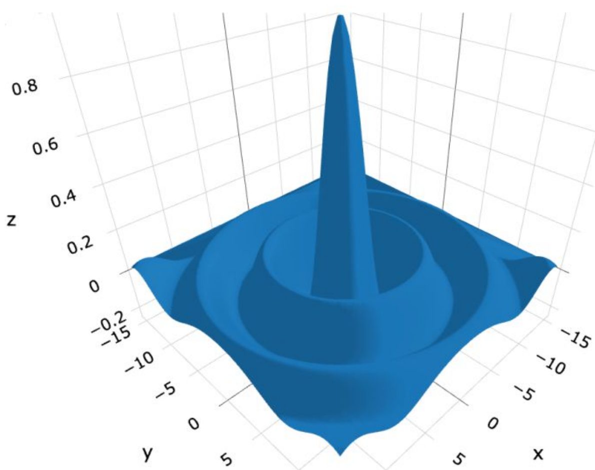


Fig. 7. The Mexican Hat graph in XOR.

For practical reasons this model is too simple, as it does not include the discrete nature of a particle that we might expect to see. This will be remedied in the next section. For now, consider Figure 8, or how the graph looks like in XNOR (see in Appendix 1 for the computer code).

Notice how this 'wave function' – although it no longer looks like a wave – curves downwards at the bottom. This characteristic is suggestive of hyper-dimensionality and has been observed in other higher-dimensional algebraic graphs (https://chart-studio.plotly.com/~robotwax/931/#/). There are also visible series of strange lines in Figure 8. These lines represent the points at which the square roots of the x^2 and y^2 are equal to zero. These points go to infinity or are otherwise undefined. Their cross shape is the result of the traces of the matrix division/multiplication (depending on the used function). By subtracting one infinity from another any real number (real value) can be produced. In this case, the value of '1' is provided. Adding in the value of –1 produces an entirely different result, which will be examined later in more detail.

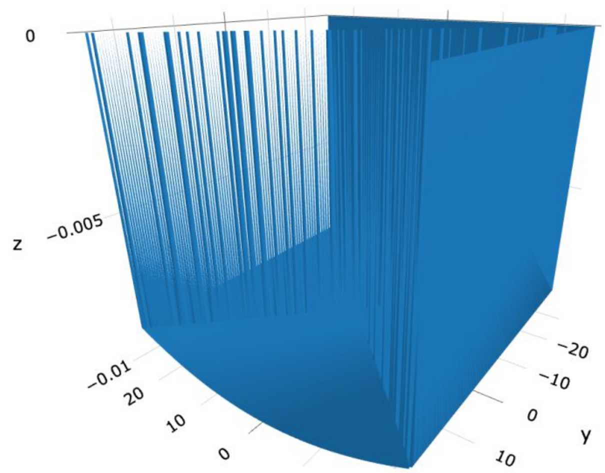


Fig. 8. The XNOR Mexican Hat graph.

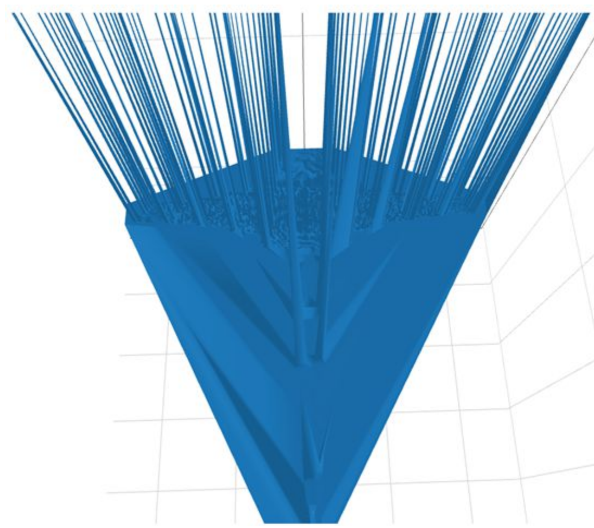


Fig. 9. The Parametric Mexican Hat graph with XOR as the x-axis and XNOR as the y-axis.

The Parametric Mexican Hat graph with XOR as the x-axis and XNOR as the y-axis is shown in Figure 9 (see in

Appendix 2 for the computer code, where the interference pattern is also shown). Another way to view these data is to set the results of our XOR calculation as our *y*-axis and the results of our XNOR calculation as our *z*-axis. This means that the two separate results of the XOR and XNOR calculation become a new coordinate set, which obviously has its advantages, when visualizing 4-dimensional space.

The waveform completely cancels out in 4D space, leaving only the infinite values. This reveals that the Mexican Hat diagram is not a valid waveform for the photon. An even more complex is therefore needed.

An Even More Complex Example

Asides from some complications with *sin*, *cos*, and exponential function, the mathematics in XNOR and its confluence with XOR is simple and straight forward enough. The function for a discrete wave packet in a 3D XOR state is

$$z = 2\exp\left(-1.2\sqrt{x^2+y^2}\right)\cos\left(7\sqrt{x^2+y^2}\right)\cos(x)$$

Plotting this results in the image shown in Figure 10 (see in Appendix 3 for the computer code).

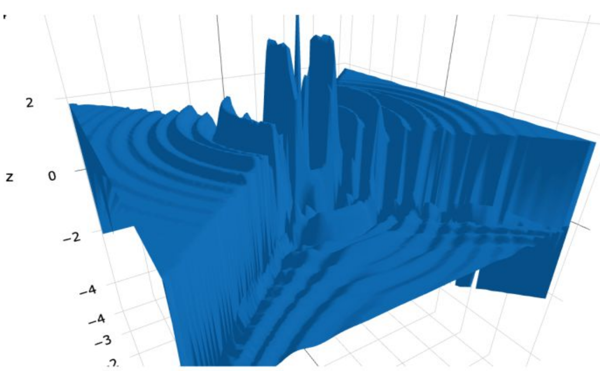


Fig. 10. The XNOR exponential function.

Again, we see this cross-shape, where the x^2 and y^2 components are undefined. Previously this value had been replaced with an '1', but now a value of '– 0.8' is more expedient. This flattens the graph down to a much more reasonable result (Fig. 11).

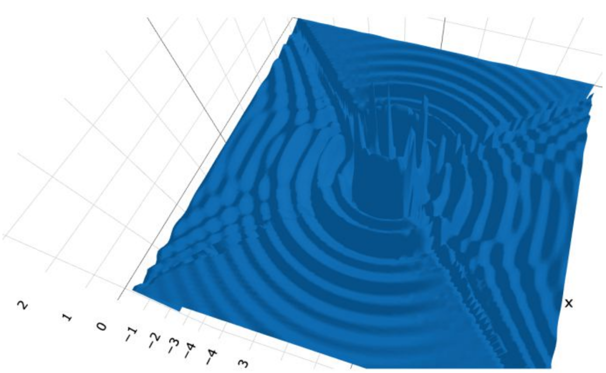
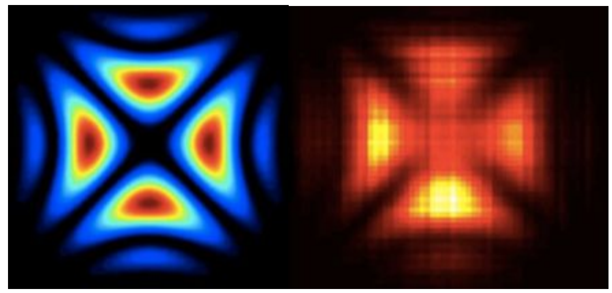


Fig. 11. The exponential XNOR graph.

Interestingly, this XNOR wave function for the photon looks very similar to the theoretical predictions based on the Schrödinger equation (Fig. 12b), as well as experimental imaging evidence carried out by physicists at the University of Warsaw (Fig. 12a). (Available at: https://cosmosmagazine.com/physics/what-shape-is-a-photon/). Graphing Figure 11 in purely XNOR gives the result shown in Figure 13.



(a) (b)

Fig. 12. (a) The theoretical predictions based on the Schrödinger equation and (b) the experimental imaging carried out by physicists at the University of Warsaw.

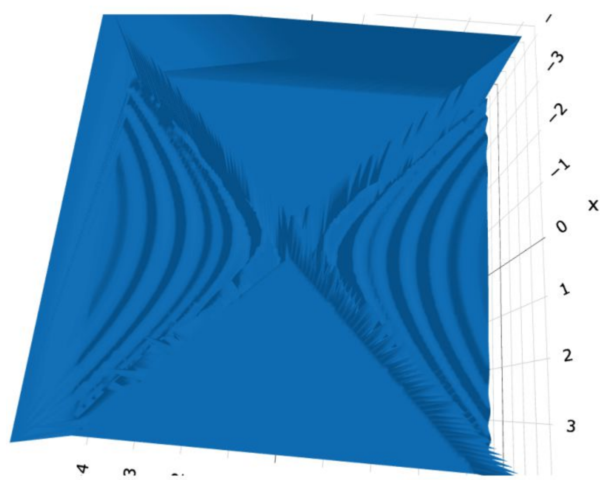


Fig. 13. The other XNOR exponential graph.

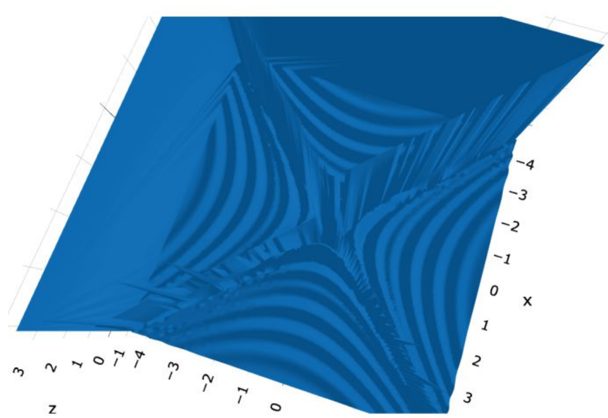


Fig. 14. The photon graph rotated by 90 degrees and superimposed.

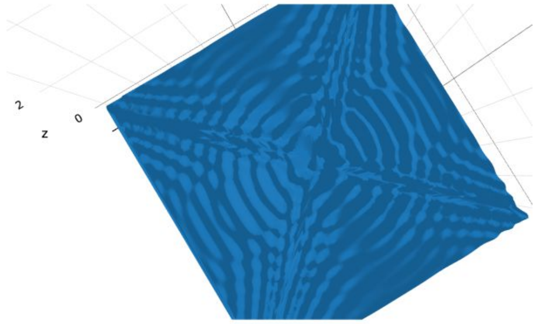


Fig. 15. The result of summation of the graphs shown in Figures 10 and 14.

With this more complex example, only half of the wave function cancels itself out. This promises this result is at least half right. To create a full XNOR momentum-space waveform, it will be necessary to graph a spherical wave front and then transform into XNOR. In leu of that, Figure 13 can be rotated by 90 degrees and summed with the original graph that is shown in Figure 14. This produces the result even closer to the result obtained at the University of Warsaw. Summing the graph shown in Figure 14 with the original flattened graph (Figure 10) results in Figure 15 (see in Appendix 4 for the code).

The Flattened Mexican Hat

Let us examine what happens with the Parametric Mexican Hat graph (see Fig. 9), when our undefined variable is set to –1. The result shown in Figure 16 is the decidedly flattened (and elongated) Mexican Hat. This is reminiscent of the theoretical Alcubierre warp drive (Fig. 17), which operates by compressing space time in front of a spacecraft and expanding the space behind it.

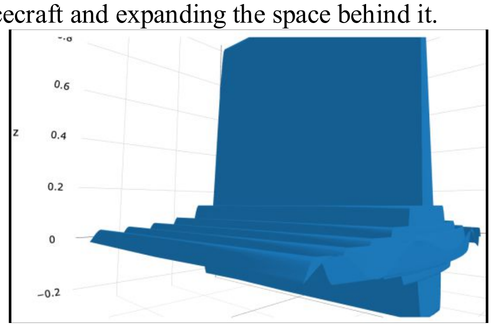


Fig. 16. The flattened Mexican Hat graph.

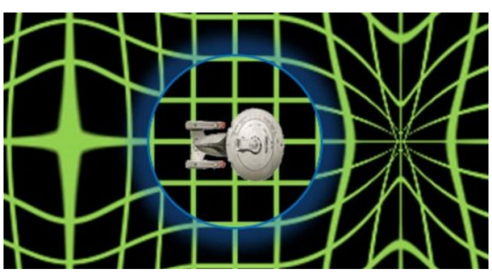


Fig. 17. The Alcubierre warp drive (Image source: https://www.universetoday.com/77005/astronomy-without-a-telescope-warp-drive-on-paper/).

Transformations of this kind can either be done on the object in question (in this case a spacecraft) or on the coordinate system surrounding them (in this case space time). These transformations are logically considered equivalent. However, notice that in the case of the flattened Mexican Hat diagram, no such transformational algorithm has been applied. Instead, the shape simply *falls out* of the XNOR and XOR data, with the XNOR axis privileged in the *y*-axis and the inclusion of the variable (–1). This means that, according to the Dimensional Gate Operator hypothesis, this is truly the momentum-state information of our particle. This is what the wave function of the photon looks like when it is travelling at light speed. This is a remarkable confirmation of the concepts set out early on in this paper, in the opinion of the author.

If indeed this is an accurate depiction of the motion of the particle, then there are numerous conclusions that can be drawn from it. First, notice the similarity between this graph and the Alcubierre warp drive. This drive is said to function on negative matter, or exotic matter. Some researchers have postulated that Dark Energy and Dark Matter may be a type of negative matter that is driving the expansion of the Universe (Farnes, 2018). If this is correct, then the arithmetic of the Dark Matter Universe is in XNOR. When a positive number is added to a positive in XOR a positive value is obtained. But doing the same thing in XNOR results in a negative. This is a good description for negative matter and since the momentum-state of the particle appears to be represented in XNOR, it would appear all matter has a dark matter component to it. This can potentially give us a way to test dark matter theory. Make note that we say 'appears', as the momentum-state is the second derivative of speed and XNOR has, as yet, no such equivalent definition.

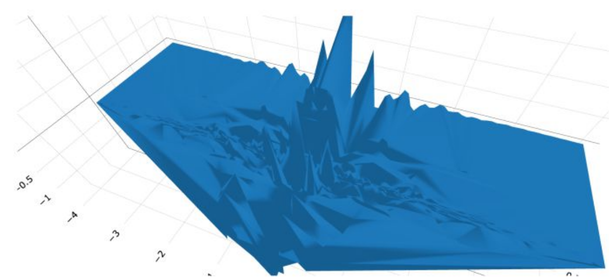


Fig. 18. 4-D XNOR space.

As mentioned earlier, the distortion of space time by any negative energy is equivalent to distorting the metric coordinate system. Notice how there is no distortion of the metric in the 'Flattened Mexican Graph' (Fig. 16). This implies that what is being depicted in this figure is more akin to a traditional Doppler Effect than space time warping. Precisely the same perspective can be applied to the more complex wave function of the photon (Fig. 18). Figure 18 shows a slice of what a photon looks like in 4-D

XNOR space. It has a pressure front and a wake. The fact that this can be derived without recourse to relativistic motion, suggests that it is entirely non relativistic in character.

CONCLUSION

Using the XNOR logic gates in place of imaginary numbers can reproduce some of the results shown in the Warsaw experimentation. This suggests a relationship between the mathematics of XNOR space and particle physics. The method therefore provides good insight into these quantum systems. It also hints at the mechanism behind the results of the Double Slit experiment.

REFERENCES

Alcubierre, M. 1994. The warp drive: Hyper-fast travel within general relativity. Classical and Quantum Gravity. 11(5):L73. DOI: https://doi.org/10.1088/0264-9381/ 11/5/001.

Farnes, JS. 2018. A unifying theory of dark energy and dark matter: Negative masses and matter creation within a modified ΛCDM framework. Astronomy and Astrophysics. 620:A92-112. https://doi.org/10.1051/0004-6361/201832898.

Gode, DB. 2014. Complex number theory without imaginary number (i). Open Access Library Journal. 1(7):e856-869. DOI: https://doi.org/10.4236/ oalib. 1100856.

Hay, MA. 2016. Recursive distinctioning, tetracoding and the symmetry properties of chiral Tetrahedral molecules. Journal of Space Philosophy. 5(2):28-55. http:// keplerspaceinstitute.com/wpcontent/uploads/2017/11/JSP -Fall-2016-11_Hay-Final.pdf.

Merli, PG., Missiroli, GF. and Pozzi, G. 1976. On the statistical aspect of electron interference phenomena. American Journal of Physics. 44(3):306-307. DOI: https://doi.org/10.1119/1.10184.

O'Neill, CC. 2021. Reimagining complex numbers. Canadian Journal of Pure and Applied Sciences. 15(2):5261-5268. DOI:https://doi.org/10.13140 /RG.2.2. 26666.44480/1.

Received: June 29, 2021; Revised: August 17, 2021;
Accepted: Sept 24, 2021

Appendix 1.
The Python code used in the generation of the kind of graph shown in Figure 8.

```
t = np.linspace(-35, 35, 300)
c = []
lo = []
p = []
for x in t:
for y in t:
    s = math.sin(math.sqrt(x**2 + y**2))/(math.sqrt(x**2 + y**2))
    a = (x**2)*-1
    b = (y**2)*-1
if b < 0:
        a1 = a + (b*-1)
elif b > 0:
        a1 = a - b
else:
        a1 = 1
try:
        c1 = (math.sqrt(a1))*-1
exceptValueError:
        c1 = 1
if c1 == 0:
        c1 = 1
try:
        g = math.sqrt(a1)*-1
exceptValueError:
        g = 1
gb = math.sin(g/c1)*-1
    r = s + gb
p.append(gb)
c.append(x)
lo.append(y)
data = [go.Mesh3d(
            x = c,
            y = lo,
            z = p
        ),
    ]
layout = go.Layout(dict(title='Sine',
titlefont= {"size": 14},
font={'color':'black'},
paper_bgcolor= 'white',
plot_bgcolor= "white",
hovermode='closest',
xaxis={"range":[-15, 15]}))
figure = dict(data=data, layout = layout)
iplot(figure)
```

Appendix 2.
The computer code for the graph shown in Figure 9. If c1==-1 and [x, y, z] == [c, p, lo], the graph in Figure 16 (the squashed Mexican Hat graph) is obtained:

```
t = np.linspace(-15, 15, 100)
c = []
lo = []
p = []
for x in t:
for y in t:
    s = math.sin(math.sqrt(x**2 + y**2))/(math.sqrt(x**2 + y**2))
    a = (x**2)*-1
    b = (y**2)*-1
if b < 0:
        a1 = a + (b*-1)
elif b > 0:
        a1 = a - b
else:
        a1 = -1
try:
        c1 = (math.sqrt(a1))*-1
exceptValueError:
        c1 = -1
if c1 == 0:
        c1 = -1
try:
        g = math.sqrt(a1)*-1
exceptValueError:
        g = -1
gb = math.sin(g/c1)*-1
    r = s + gb
p.append(gb)
c.append(x)
lo.append(s)


data = [go.Mesh3d(
            x = c,
            y = lo,
            z = p
        ),
    ]
layout = go.Layout(dict(title='Photon XOR XNOR',
titlefont= {"size": 14},
font={'color':'black'},
paper_bgcolor= 'white',
plot_bgcolor= "white",
hovermode='closest',
xaxis={"range":[-1, 1]}))
figure = dict(data=data, layout = layout)
iplot(figure)
```

Also, steps can be taken to extract the data from Figure 9, which results in the interference pattern shown in Figure 19 below.

The reader should note however that this is not a new interference pattern. It is simply the Mexican Hat diagram viewed in a different way. However, the ‘spikes’ shown in Figure 9 would produce the correct, although inverted, interference result (Merli *et al*., 1976).

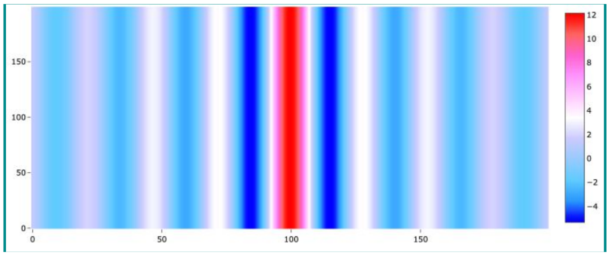


Fig. 19. The interference pattern.

Appendix 3.

The computer code for the graph shown in Figures 10 is below. To produce the graph shown in Figure 11, simply change ‘a1 = 1’ to ‘a1 = -1’ or minus phi.

```
t = np.linspace(-4, 4, 100)
c = []
lo = []
p = []
ee = []
fe = []
pv = []
uu = []

met=-1.2

tet= 10

for x in t:
for y in t:
    s = 2*math.exp(met*math.sqrt(x**2+y**2))*math.cos(((tet*math.sqrt(x**2+y**2))))*math.cos(x**2+y**2)
    a = (x**2)*-1
    b = (y**2)*-1
if b < 0:
      a1 = a + (b*-1)
elif b > 0:
      a1 = a - b
else:
      a1 = 1
try:
      c1 = (math.sqrt(a1))*-1
exceptValueError:
      c1 = 1
    tet2 = ((met*c1)*-1)
if tet2 <= 0:
      tet3 = math.exp(tet2)
elif tet2 > 0:
      tet3 = (math.exp(tet2))-tet2
gb = ((2*tet3)*-1)*(math.cos(((tet*c1)*-1))*-1)
    gb1 = (math.cos(c1)*-1)
    gb2 = (gb*gb1)*-1
    m = s + gb2
if gb2 < 0:
      r = (s+(gb2*-1))
elif gb2 > 0:
      r = (s-gb2)
qq = s + r
uu.append(m)
p.append(r)
pv.append(qq)
ee.append(s)
fe.append(gb2)
c.append(x)
lo.append(y)
data = [go.Mesh3d(
        x = c,
        y = lo,
        z = pv
      ),
    ]
layout = go.Layout(dict(title='Photon XNOR',
titlefont= {"size": 14},
font={'color':'black'},
paper_bgcolor= 'white',
plot_bgcolor= "white",
hovermode='closest'))
figure = dict(data=data, layout = layout)
iplot(figure)
```

Appendix 4.

Change the ‘*z*’ parameter to ‘o11’, ‘o13’, or ‘o14’ to produce the graphs shown in Figures 13, 14, and 15, respectively.

```
ou9 = np.array(fe).reshape(100, 100)
o10 = np.rot90(ou9, 1)
o11, o12 = np.ravel(ou9), np.ravel(o10)
o13 = [o11[i]+o12[i] for i in range(len(pv))]
o14 = [pv[i]+o12[i] for i in range(len(pv))]
data = [go.Mesh3d(
        x = c,
        y = lo,
        z = o13
      ),
    ]
layout = go.Layout(dict(title='Photon XNOR',
titlefont= {"size": 14},
font={'color':'black'},
paper_bgcolor= 'white',
plot_bgcolor= "white",
hovermode='closest'))
figure = dict(data=data, layout = layout)
iplot(figure)
```