Discrete event simulations of EPRB experiments and analysis of data processing

Jeroen Jetten

July 2, 2015

Abstract

In this paper, I will be taking a look at simulations of the Einstein-Podolsky-Rosen-Bohm experiment with photons. I will be looking at two methods specifically, one published by De Raedt et al. [11], and one published by Zhao et al. [16]. I will compare them to the predictions of quantum mechanics on variation of angle of the polarizers in the experimental setup, and to the results of the experiment done by Weihs et al. [14] with regards to the variation of the time window.

Contents

1 Introduction

It was Richard Feynman who once said *If you think you understand quantum mechanics, you do not understand quantum mechanics.* One of my motivations for choosing this research project was, if not to better understand quantum mechanics, to better understand where our understanding stops. If you understand what I am trying to say.

The original problem was first posed by Einstein, Podolsky, and Rosen about how to describe particles in quantum mechanics, arguing it is incomplete and that there must be a deeper theory to explain the world than the wave function of quantum mechanics. From this original problem an experiment was proposed and later performed using photons with orthogonal polarizations to test the predictions of quantum mechanics.

John Bell proposed in 1964 an inequality that would, if broken, prove that a class of local hidden variable theories were not able to explain quantum entanglement [1]. This idea was later applied to the photon entanglement by Clauser et al. [3].

In this paper I will be discussing two event based models that both aim to recreate the data produced by experiments aimed at disproving local hidden variable explanations for photon entanglement, where the models will be fully local non-quantum mechanical models.

2 Theory

In the following section I will first give an outline of the original thought experiment, and continue from there to set out the actual physical experiment that was used to test Einstein, Podolsky, and Rosen's thought experiment.

From there I will define the parts of the experiment that can later be found in the simulations and the general relevant equations.

2.1 EPRB gedanken experiment

In 1935, A. Einstein, B. Podolsky, and N. Rosen published a paper on quantum mechanics with the title 'Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?' in which they lay out a possible problem they have with the wave function description of reality. This is summed up by the following excerpt of their abstract:

In the case of two physical quantities described by non-commuting operators, the knowledge of one precludes the knowledge of the other. Then either (1) the description of reality given by the wave function in quantum mechanics is not complete or (2) these two quantities cannot have simultaneous reality.

They then go on to explain a thought experiment where two systems are brought to an entangled state and information about one system is gained by measuring the other giving a paradoxical result that leads them to conclude that (1) must be true. A clearer formulation of the thought experiment is given by Bohm and Aharonov as the following:

We consider a molecule of total spin zero consisting of two atoms, each of spin one-half. The wave function of the system is therefore

$$\phi = \frac{1}{\sqrt{2}}[\phi_+(1)\phi_-(2) - \phi_-(1)\phi_+(2)]. \tag{1}$$

where $\phi_+(1)$ refers to the wave function of the atomic state in which one particle (A) has spin $+\hbar/2$, etc. The two atoms are then separated by a method that does not unfluence the total spin. After they have separated enough so that they cease to interact, any desired component of the spin of the first particle (A) is measured. Then, because the total spin is still zero, it can immediatly be concluded that the same component of the other particle (B) is opposite to that of A.

If this were a classiscal system, there would be no difficulty in interpreting the above results, because all components of the spin of each particle are well defined at each instant of time. Thus, in the molecule, each component of the spin of particle A has, from the very beginning, a value opposite to that of the same component of B; and this relationship does not change when the atom disintregrates. In other words, the two spin vectors are correlated. Hence, the measurement of any component of the spin of A permits us to conclude also that the same component of B is opposite in value.[2]

What I will be using is based on the same principles as described above, but instead of using atoms in a diatomic molecule I will be using polarized light. The base setup is as follows:

1. A source creates two photons with random but orthogonal polarization.
2. Two electro-optic modulators, one for each photon, is randomly set to one of a predetermined-set of angles.
3. A polarizer sends the photon to one of two detectors based on polarization.
4. A set of detectors and sends the detection-event to the correlator together with information about which detector fired, and at what time.
5. A computer processes the information from the detectors and creates correlated pairs based on the timetag information.

A visual overview of the setup can be seen in figure 1 as taken from [10].

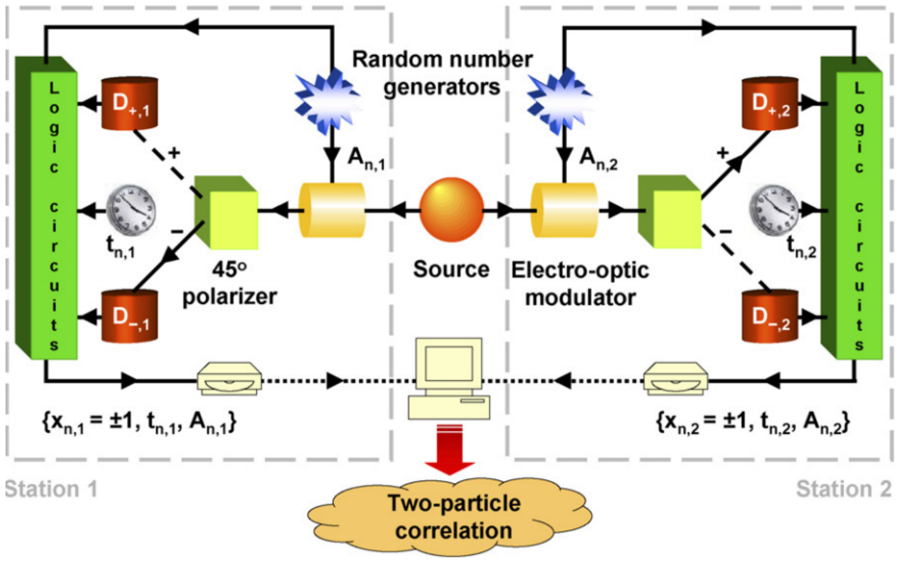


Figure 1: An overview of the simulated setup taken from [10].

2.2 Bell's inequality

In order to keep track of the registered photons I use the following function for counting coincidences:

$$C_{xy} = \sum_{n=1}^{N} \delta_{x,x_{n,1}} \delta_{y,x_{n,2}} \Theta(W - |t_{n,1} - t_{n,2}|). \tag{2}$$

where $x, y = \pm$ where $+$ is the top detector in figure 1 and Θ is a step function which is 1 if the argument is positive and zero otherwise. I can combine this function for several combinations of angles to get:

$$E(\alpha,\beta) = \frac{C_{++}(\alpha,\beta) + C_{--}(\alpha,\beta) - C_{+-}(\alpha,\beta) - C_{-+}(\alpha,\beta)}{C_{++}(\alpha,\beta) + C_{--}(\alpha,\beta) + C_{+-}(\alpha,\beta) + C_{-+}(\alpha,\beta)}, \tag{3}$$

where E is a measure of the coincidence rate or, in plain terms, how often both photons go up, or down. Adding up several of these coincidence rates for different angles leads to a Bell function[1]:

$$S(\alpha,\alpha',\beta,\beta') = E(\alpha,\beta) - E(\alpha',\beta) + E(\alpha,\beta') + E(\alpha',\beta'), \tag{4}$$

which was first formulated by Clauser et al. [3] but was later reformulated in the current form for the experiment with two photons and is often called the Bell-CHSH function.

$$S_{max} = \max_{\alpha,\alpha',\beta,\beta'} S(\alpha,\alpha',\beta,\beta') = S(\alpha,\alpha+2\theta,\alpha+\theta,\alpha+3\theta). \tag{5}$$

Because of rotational symmetry α can be chosen freely, therefore I take $\alpha = 0$ and get the following expression:

$$S(\theta) = S(0, 2\theta, \theta, 3\theta). \tag{6}$$

Varying θ gives us several possibilities for the maximum value of $S(\theta)$. I will follow others in choosing the first option giving us $S_{max} = S(0, \frac{\pi}{4}, \frac{\pi}{8}, \frac{3\pi}{8})$.

I look at two expectation values for S_{max}. In quantum theory[11] the state where both polarizations are uncorrelated gives the following limitation for the Bell-CHSH inequality;

$$\widehat{S}_{max} = \max_{\alpha,\alpha',\beta,\beta'} \widehat{S}(\alpha,\alpha',\beta,\beta') \leq 2. \tag{7}$$

The solution for the entangled(singlet) state gives us:

$$\widehat{S}_{max} = \max_{\alpha,\alpha',\beta,\beta'} \widehat{S}(\alpha,\alpha',\beta,\beta') \leq 2\sqrt{2}. \tag{8}$$

In fact it even gives a closed form expression for equation 6, namely

$$\widehat{S}(\theta) = cos(6\theta) - 3cos(2\theta). \tag{9}$$

3 Experimental data

I will mainly be comparing the data I generate with the models to the experiment done by Weihs et al. [14], where they found a violation of Bell's inequality(equation 4) of $|S| = 2.73 \pm 0.02$. This data point, where they used the angles $\alpha = 0°$ $\alpha' = 45°$ and $\beta = 22.5°$ $\beta' = 67.5°$, fits in the quantum prediction I use to compare the model to(equation 9).

Furthermore, I will compare the time tagged data to their data as provided by De Raedt et al. [11] as shown in figure 2. For comparisons I will generally use the time shifted data($\Delta = 4ns$).

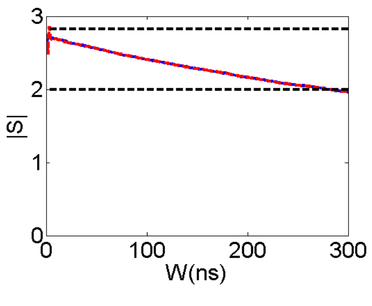
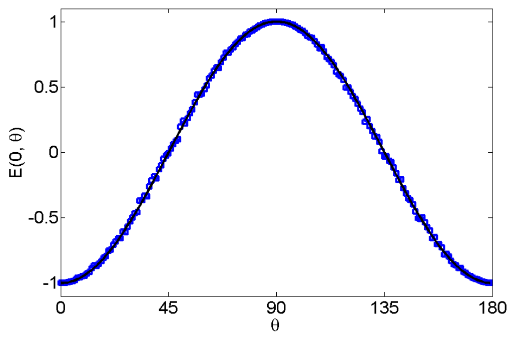
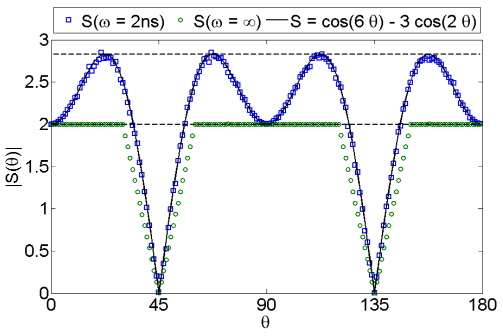


Figure 2: Red line: data with timeshift $\Delta = 0ns$; blue line: data with timeshift $\Delta = 4ns$.

4 Event based model of EPRB experiment



(a) Blue crosses are data points from the simulation calculated according to equation 3, black line is the prediction from quantum theory $E(0, \theta) = -cos(2\theta)$.



(b) Dashed lines at $|S| = 2$ and $|S| = 2\sqrt{2}$.

Figure 3

The basic layout for my simulation follows the algorithm laid out by De Raedt et al. [10]. Unlike the example program in the aforementioned paper I decided to keep setting the parameters, and doing the calculations for E(equation 3) and S (equation 4), separate from the main program loop.

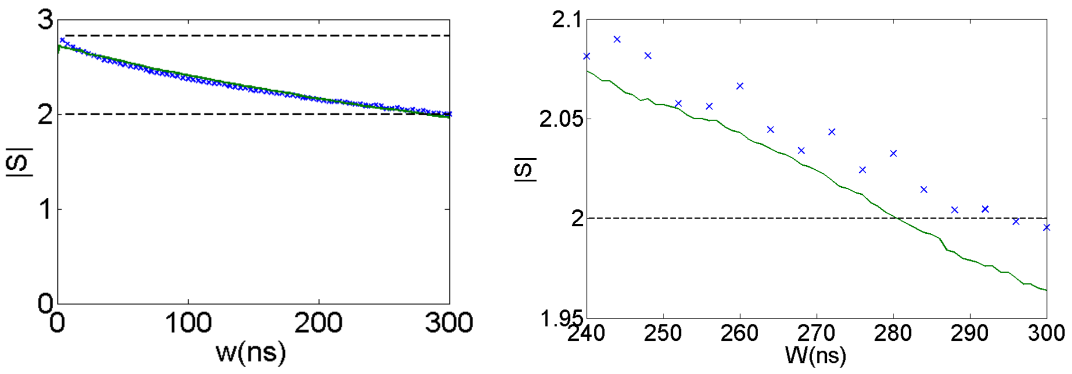
The simulation can be divided in the following steps:

1. A photon pair is created with the first photon getting a random polarization from the interval $\theta_1 \in [0, 2\pi]$ and the second polarization is determined by $\theta_2 = \theta_1 + \frac{\pi}{2}$.
2. The electro-optic modulator is simulated by the following equation: $\xi_x = \cos 2(\theta_x - \alpha)$ where α is the angle of the electro-optic modulator.
3. The detector is chosen based on the sign of ξ_x.
4. A timetag is generated according to $t_x = (1 - \xi_x^2)^{\frac{d}{2}} / \tau$ where d, τ are parameters of the simulation.
5. Then according to the variables above the coincidence bin of the photon pair is determined according to equation 2.

I considered creating separate classes for the different parts of the algorithm to both make the program more modular, and to allow reuse of the basic code. In the end I decided against it in favor of giving a better overview of the main simulation loop by keeping most of the math in one place. The actual code used to run the experiment can be found in appendix A. The commented-out lines are for this simulation and the code given is for the second model discussed in section 5.

Running the simulation to create the data sets for $E(0,\theta)$ given by equation 3 and $|S(\theta)|$ given by equation 6, I get the results as shown in figure 3. There are several parameters that can be freely chosen for the simulation that influence the time tag generation. The parameters d and τ, called tau in the code, are both mentioned in step 4 of the above list. For all simulations of the first model d is set to 2 and tau is taken to be $25 \times 10^{-5} ns^{-1}$. The parameter N controls the amount of photon pairs sent through the system and is set to 10^7 as a high enough number to smooth out the plots in figure 3 and to be small enough to quickly be run on a pc.

Variation with the time window W



(a) Plotted with a green line is the experimental data from [14] as adjusted in [11], and in blue the data from the simulation.

(b) Like 4a but zoomed in on the right end of the plot.

Figure 4

As can be seen in figure 4a, the simulated data follows the experimental data closely, albeit with two small anomalies. Firstly at $W < 6ns$ the simulation gives slightly higher values for $|S|$ but still within the predicted limit of $2\sqrt{2}$. Secondly near 300 ns, the simulated data seems to bend towards the limit $|S| = 2$ unlike the experimental data which goes below it as can be seen in figure 4b.

To understand why this must be the case with the current simulation model you can look back to figure 3b where the values for $\lim_{W\to\infty} |S| = 2$.

5 The second model

There is another model described in [16] where the algorithm to simulate the polarizer is explicitly chosen to reproduce Malus law:

$$I = I_0 \cos^2 \theta. \tag{10}$$

Because several parts of the simulation change to accommodate for this, and for completeness,I will again give the full description of the simulation where changes are in bold face.

1. A photon pair is created with the first photon getting a random polarization from the interval $\theta_1 \in [0, 2\pi]$ and the second polarization is determined by $\theta_2 = \theta_1 + \frac{\pi}{2}$.
2. The electro-optic modulator is simulated by the following equation: $\boldsymbol{\theta'_x = \theta_x - \alpha}$ where α is the angle of the electro-optic modulator.
3. The Detector is chosen based on the following **if $\boldsymbol{R \leq cos^2\theta'_x}$ the top detector is chosen otherwise the bottom one is used, where $\boldsymbol{R \in [0, 1]}$ is randomly chosen**
4. A timetag is generated according to $\boldsymbol{t_x = T_0 sin^4 2\theta'_x}$ **where $\boldsymbol{T_0 = \frac{1}{\tau}}$ is the only parameter of the simulation**.
5. Then, according to the variables above, the coincidence bin of the photon pair is determined according to equation 2.

First I try to reproduce figure 3 with the new model, the results of this can be seen in figure 6. The important difference that I was hoping for is a change in $|S|$ for the limit $W \to \infty$. However another big difference is the noise in $|S|$ with $w = 2ns$.

This noise is even more pronounced if the parameters in [16] are used where in the first paragraph of section 7 they state their parameters are the following:

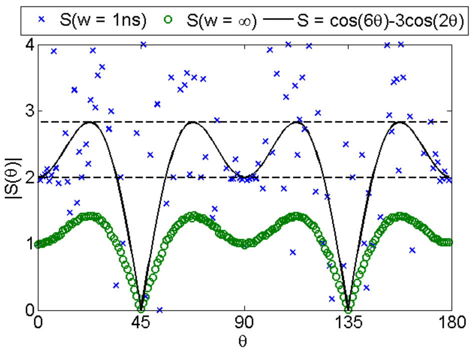
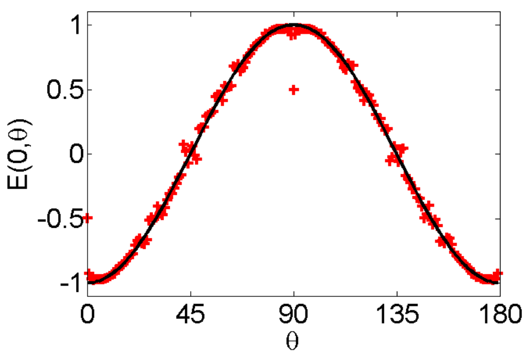


Figure 5

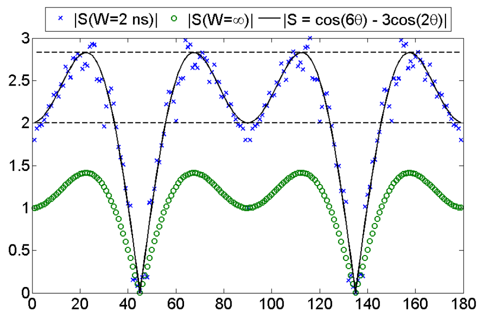
$$k = 1ns, \qquad d = 4, \qquad \frac{1}{T_0} = \tau = 0.00025. \tag{11}$$

The simulation results using these parameters can be seen in figure 5. These results deviate heavily from figure 6(left) in the paper by Zhao et al. [16].

Following figure 8 in [16] I can see that $\frac{w}{T_0}$ is an adjustable parameter that can affect the maximum value of $|S|$. In figure 6 we find $\frac{w}{T_0} = \frac{2}{1400} \approx 10^{-3}$. Results for a higher $\frac{w}{T_0}$

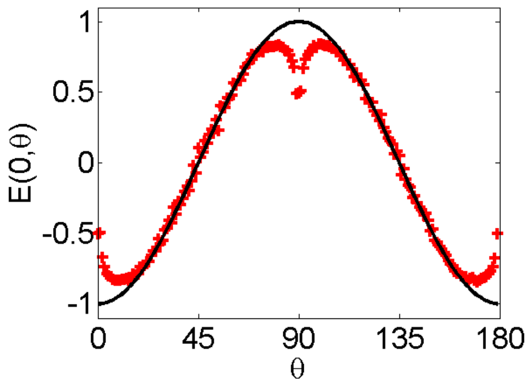


(a) Red crosses are data points from the simulation calculated according to equation 3, black line is the prediction from quantum theory $E(0,\theta) = -cos(2\theta)$.

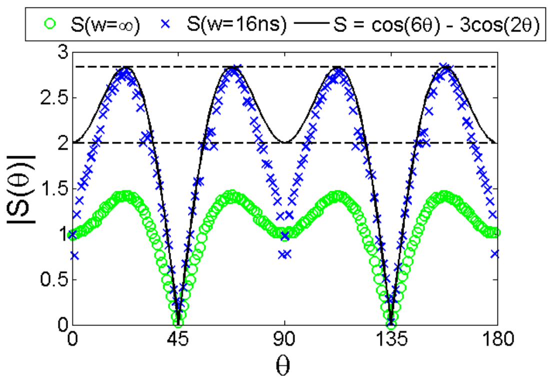


(b) Dashed lines at $|S| = 2$ and $|S| = 2\sqrt{2}$.

Figure 6: Comparison of $E(0,\theta)$ and $|S|$ for the model that recreates Malus law. Using parameters $w = 2ns$ and $T_0 = 1400ns$.

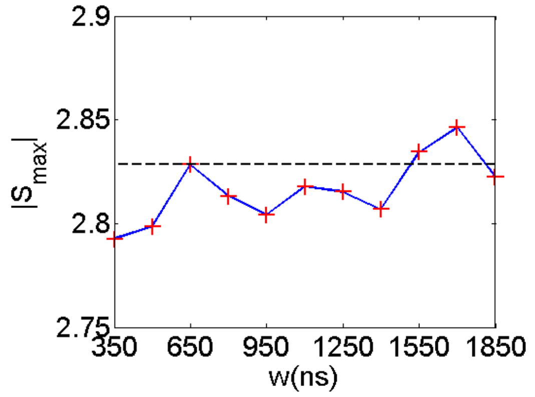


(a) Red crosses are data points from the simulation calculated according to equation 3, black line is the prediction from quantum theory $E(0,\theta) = -cos(2\theta)$.

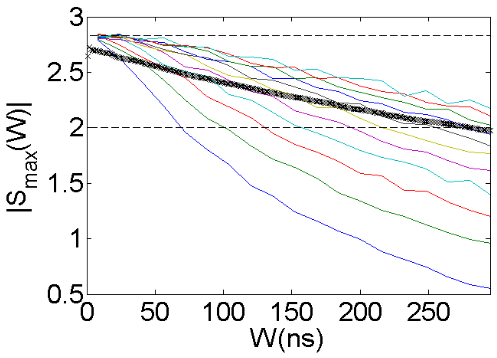


(b) Dashed lines at $|S| = 2$ and $|S| = 2\sqrt{2}$.

Figure 7: Adjusted to $\frac{w}{T_0} = 0.05\overline{3}$ by choosing $w = 16ns$ and $T_0 = 300ns$.



(a) Red pluses/Blue line: The highest value for the simulation data as a function of T_0. Black dashed line: the limit for the entangled prediction $|S_{max}| = 2\sqrt{2}$.



(b) Black crosses: experimental data. Coloured lines: data from model 2 with varying T_0 with the bottom line being $T_0 = 350ns$ and increasing with $150ns$ each line till $T_0 = 1850ns$ and a $w = 16ns$ shifted left 8ns making each datapoint the center of it's bin.

Figure 8: Influence of varying T_0 on $|S_{max}|$

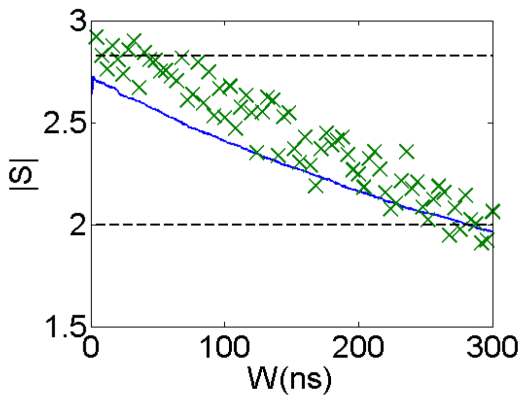
are shown in figure 7. Altough it makes $|S(\theta)|$ more well behaved around the maxima it introduces larger deviations from the prediction around $\theta = 0°, 90°$,and 180°.

Variation with the time window W

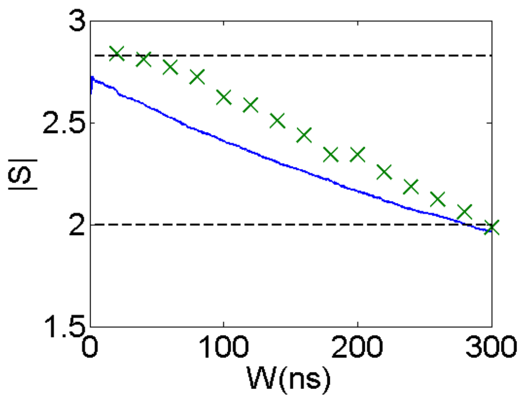
Looking at various values for T_0 in figure 8a suggests that the optimal value for T_0 will probably lie under 1500ns.

None of the lines in figure 8b seem to match the experimental data in the lower time bins. However, in the 250-300ns region the closest match seems to be $T_0 = 1400ns$ in as far as the data matches. Looking purely at the slope, the closest match seems to be $T_0 = 1850ns$ albeit shifted to the right by about 100ns. I cannot give a reasonable justification for shifting the data in such a way, therefore I will treat $T_0 = 1400ns$ as the best matching data set which can be seen in figure 9.

Comparing figure 9a with figure 9b, it becomes clear how problematic the small value for $\frac{w}{T_0}$ becomes. Increasing w in this case only achieves a smoothing out, or averaging, of the value of $|S|$. This is, in my opinion, better interpreted as hiding the underlying problem, rather than solving it.



(a) plotted with a blue line is the experimental data from [14] as adjusted in [11], and in green the data from the simulation with w = 4ns.



(b) like 9a but instead with w = 20ns.

Figure 9

6 Discussion

After looking at the two simulations, I first want to note about the simulations themselves that:

- Each component of the experiment has it's own representations in code where identical components have identical representations
- The representations are not unique to this experiment but can be used as they are to reproduce other optical experiments.
- Except for the photons moving through the system there is no exchange of information between components or between photons, i.e., each pair of photons is a seperate local event.

Looking then at how closely the data from the simulations mimics the properties of the physical experiments, it is clear that especially the first model in both figure 3 and figure 4 is in complete agreement with the experiments. The second model might offer a better match if problems with my implementation can be worked out. This to relieve the problem of the first model being bound to $|S| = 2$ for the limit $W \to \infty$.

Although Bell's inequality and consequently Bell's theorem might not be physically relevant([5] and [9]), I find it important to show how the simulated data follows the same statistics as the experimental data in comparison to Bell's inequality and theorem.

7 Acknowledgments

I would like to thank my supervisor Prof H. De Readt on suggesting the second model, and in general for introducing me to this field of science. I would also like to thank Kathryn Duffield for advising me on my English.

References

[1] On the Einstein Podolsky Rosen Paradox, J.S. Bell, Physics Vol. 1, No. 3, pp.195-200, 1964

[2] Discussion of Experimental Proof for the Paradox of Einstein, Rosen, and Podolsky, D.Bohm, and Y. Aharonov, Physical Review volume 108, November 15, 1957

[3] Proposed Experiment to Test Local Hidden-Variable Theories, J.F. Clauser, M.A. Horne, A. Shimony, and R.A. Holt, Physical Review Letters volume 23 number 15, October 13, 1969

[4] Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?, A. Einstein, B. Podolsky, and N. Rosen, Physical Review volume 47, may 15, 1935

[5] The Bell inequality cannot be validly applied to the Einstein-Podolsky-Rosen-Bohm (EPRB) experiments, Donald A. Graft, Physics Essays 22, 4 (2009), 0836-1398

[6] Einstein-Podolsky-Rosen-Bohm Experiment Using Pairs of Light Quanta Produced by Type-II Parametric Down-Conversion, T.E. Kiess, Y.H. Shih, A.V. Sergienko, and C.O. Alley, Physical Review Letters volume 71, Number 24, 13 December 1993

[7] Event-based simulation of quantum physics experiments, K. Michielsen, and H. De Raedt, International Journal of Modern Physics C, Vol. 25, No. 8 (2014)

[8] On possible experimental tests for the paradox of Einstein, Podolsky and Rosen, A. Peres, and P. Singer, Il Nuovo Cimento, v15 n6 (196003): 907-915 22 December 1959

[9] Event-by-Event Simulation of quantum Phenomena: Application to Einstein-Podolosky-Rosen-Bohm Experiments H. De Raedt, and K. Michielsen, Ann. Phys. (Berlin) 524, No.8, (2012)

[10] A computer program to simulate Einstein-Podolsky-Rosen-Bohm experiments with photons, K. De Raedt, H. De Raedt, K. Michielsen, Computer Physics Communications, Volume 176, issues 11-12, june 2007

[11] Event-by-Event Simulation of quantum Phenomena: Application to Einstein-Podolosky-Rosen-Bohm Experiments H. De Raedt, K. De Raedt, K. Michielsen, K.

Keipema, and S. Miyashita, Journal of Computational and Theoretical Nanoscience, Vol. 4, 957-991, 2007

[12] Hidden-Variable model for the Bohm Einstein-Podolsky-Rosen experiment, K. H. Schatten, Physical Review A, v48 n1 (19930701): 103-104, July 1993

[13] New Type of Einstein-Podolsky-Rosen-Bohm Experiment Using Pairs of Light Quanta Produced by Optical Parametric Down Conversionm Y.H. Shih, and C.O. Alley, Physical Review Letters volume 61, Number 26, 26 December 1988

[14] Violation of Bell's Inequality under Strict Einstein Locality Conditions, G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger, Physical Review Letters volume 81 number 23, 7 December 1998

[15] Evaluation of polarization entanglement generated by pulsed spontaneous parametric down-conversion with multi-pairs using four single-photon detectors for quantum state tomography, A. Yoshizawam, D. Fukuda, and H. Tsuchida, Optics Communications 285(2012)

[16] Event-by-Event Simulation of Einstein-Podolsky-Rosen-Bohm Experiments, S. S. Zhao, H. De Raedt, and K. Michielsen, Foundations of Physics 38(4):322-347, 0015-9018

A Code

Main.cc

```
#include "main.ih"

double CalcS(long long Data[2][2][2][2]);
double CalcE(long long Data[2][2][2][2]);
double CalcE1(long long Data[2][2][2][2]);
double CalcE2(long long Data[2][2][2][2]);

int main(int argc, char *argv[])
{
    long double angles1[M+1] = {};
    long double angles2[M+1] = {};

    long double S[TimeSteps] = {};
    long double E[TimeSteps] = {};
    long double E1[TimeSteps] = {};
    long double E2[TimeSteps] = {};

    RandomNumber rand;

    long long Ntiming[TimeSteps][2][2][2][2] = {};


    angles1[0] = 0;
    angles1[1] = pi / 4 ;
    angles2[0] = pi/8 ;
    angles2[1] = 3 * pi / 8;

    for(long long idx = 0; idx != N * M * M + 1;++idx)
    { //simulation loop
    //source
        double x1 = twopi * rand();
        double x2 = x1 + pi2;

    //station 1
        long long i1 = round(rand());

        // double c1 = cos(2 * (x1 - angles1[i1]));
        double c1 = x1 - angles1[i1];
        long long j1;
        if (rand() <= pow(cos(c1), 2))//if (c1 > 0)
            j1 = 0;
        else
            j1 = 1;

        long long k1 = pow(sin(2 * c1), 4) * Tnill;
        //long long k1 = ceil((pow((1-c1*c1), (d/2))*rand()) / tau);

    //station 2
        long long i2 = round(rand());

        //double c2 = cos(2 * (x2 - angles2[i2]));
        double c2 = x2 - angles2[i2];
        long long j2;
        if (rand() <= pow(cos(c2), 2))//if (c2 > 0)
            j2 = 0;
        else
            j2 = 1;

        long long k2 = pow(sin(2 * c2), 4) * Tnill;
        //long long k2 = ceil((pow((1-c2*c2), (d/2))*rand()) / tau);

    //count
        Ntiming[0][j1][j2][i1][i2] += 1;
        bool stop = false;
        for(size_t idx2 = 1; (idx2 != TimeSteps) && !stop;++idx2)
        {
            if(abs(k1 - k2) < abs(idx2 * k))
            {
                Ntiming[idx2][j1][j2][i1][i2] += 1;
                stop = true;
            }
        }
    }

    for(size_t idx = 0; idx != TimeSteps;++idx)
    {
        S[idx] = CalcS(Ntiming[idx]);
        E[idx] = CalcE(Ntiming[idx]);
        E1[idx] = CalcE1(Ntiming[idx]);
        E2[idx] = CalcE2(Ntiming[idx]);
    }
```

```
    cout << "W = " << k << " d = " << d << "tau = " << tau << '\n';
    cout << "W  Smax E E1 E2 \n";
    cout << "-----------------------\n";
    for(long long idxa = 0; idxa != TimeSteps;++idxa)
    {
        cout << (idxa * k) << ' ' << S[idxa] << ' '
             << E[idxa] << ' ' << E1[idxa] << ' ' << E2[idxa] << '\n';
    }
}
```

parameters.h

```
#ifndef PARAMETERS_H_
#define PARAMETERS_H_

int const M = 1; //controls the number of angles per simulation
int const Degrees = 180; //the max angle to be simulated
int const DegPerStep = 1;

const int N = 1000000; //number of photons per simulation

const long double tau = 0.00025;
const long double Tnill = 1000;

const int d = 2; //parameter to control the time tag
const int k = 2; //parameter to control the time window

int const Ntests = 1;
int const TimeSteps = 2;

const long double pi = acos(-1);//some mathematical constants
const long double twopi = pi * 2;
const long double pi2 = pi / 2;

#endif
```

CalcS.cc

```
#include "main.ih"

double CalcS(int Data[2][2][2][2])
```

```
{
    long double E00, E01, E10, E11;

    E00 = double(Data[1][1][0][0] + Data[0][0][0][0]
          - Data[1][0][0][0] - Data[0][1][0][0])
          / double(Data[1][1][0][0] + Data[0][0][0][0]
          + Data[1][0][0][0] + Data[0][1][0][0]);

    E01 = double(Data[1][1][0][1] + Data[0][0][0][1]
          - Data[1][0][0][1] - Data[0][1][0][1])
          / double(Data[1][1][0][1] + Data[0][0][0][1]
          + Data[1][0][0][1] + Data[0][1][0][1]);

    E10 = double(Data[1][1][1][0] + Data[0][0][1][0]
          - Data[1][0][1][0] - Data[0][1][1][0])
          / double(Data[1][1][1][0] + Data[0][0][1][0]
          + Data[1][0][1][0] + Data[0][1][1][0]);

    E11 = double(Data[1][1][1][1] + Data[0][0][1][1]
          - Data[1][0][1][1] - Data[0][1][1][1])
          / double(Data[1][1][1][1] + Data[0][0][1][1]
          + Data[1][0][1][1] + Data[0][1][1][1]);

    return (E00 + E10 + E11 - E01);
}

double CalcE(int Data[2][2][2][2])
{
    return double(Data[1][1][0][0] + Data[0][0][0][0]
                  - Data[1][0][0][0] - Data[0][1][0][0])
                  / double(Data[1][1][0][0] + Data[0][0][0][0]
                  + Data[1][0][0][0] + Data[0][1][0][0]);
}

double CalcE1(int Data[2][2][2][2])
{
    return double(Data[1][1][0][0] - Data[0][0][0][0]
                  - Data[1][0][0][0] + Data[0][1][0][0])
                  / double(Data[1][1][0][0] + Data[0][0][0][0]
                  + Data[1][0][0][0] + Data[0][1][0][0]);
}
```

```
double CalcE2(int Data[2][2][2][2])
{
    return double(Data[1][1][0][0] - Data[0][0][0][0]
                  + Data[1][0][0][0] - Data[0][1][0][0])
                  / double(Data[1][1][0][0] + Data[0][0][0][0]
                  + Data[1][0][0][0] + Data[0][1][0][0]);
}
```

main.ih

```
#include <iostream>
#include <cmath>
#include <random>
#include "parameters.h"
#include "randnumber.h"

using namespace std;
```

randnumber.h

```
#ifndef randnumber_H_
#define randnumber_H_

#include "main.ih"

class RandomNumber
{
    std::default_random_engine generator;
    std::uniform_real_distribution<double> distrobution;

    public:
        RandomNumber();
        double operator()();
};

inline RandomNumber::RandomNumber()
:
    distrobution(0.0, 1.0)
{}

inline double RandomNumber::operator()()
{
    return distrobution(generator);
}

#endif
```