Simulating the EPRB experiment

Author: A.H. van Zessen (s1918141)

Supervisors: Prof. Dr. H. De Raedt
Prof. Dr. R. Timmermans

July 15, 2014

Contents

1 Introduction

Quantum Mechanics (QM) is a successful and proven theory originating from the early 20$^{\text{th}}$ century. It provides us with a much deeper understanding of the physical phenomena at the atomic and subatomic scales. QM has been verified experimentally to an incredible degree of precision.

Quantum measurement paradox

There are conceptual problems with QM however. The experimental fact that each observation yields a definite outcome together with the impossibility to describe single events from theory is called the quantum measurement paradox [1]. It is the most central and fundamental problem in QM.

The famous double-slit experiment (see Fig. 1) illustrates this manifestation of the quantum measurement paradox in an elegant way. Single electrons are shot through two slits one-by-one and arrive at a detection screen. Only after shooting a large amount of electrons at the screen an interference pattern emerges. The interference pattern can be explained by QM, but it cannot give us a description on the level of single events. What it provides us is a description of the statistical average over many events.

Event-by-event simulation

This is where the physics of QM stands now. We are unable to say anything useful about single events that occur on the quantum level. This is where my bachelor thesis will be

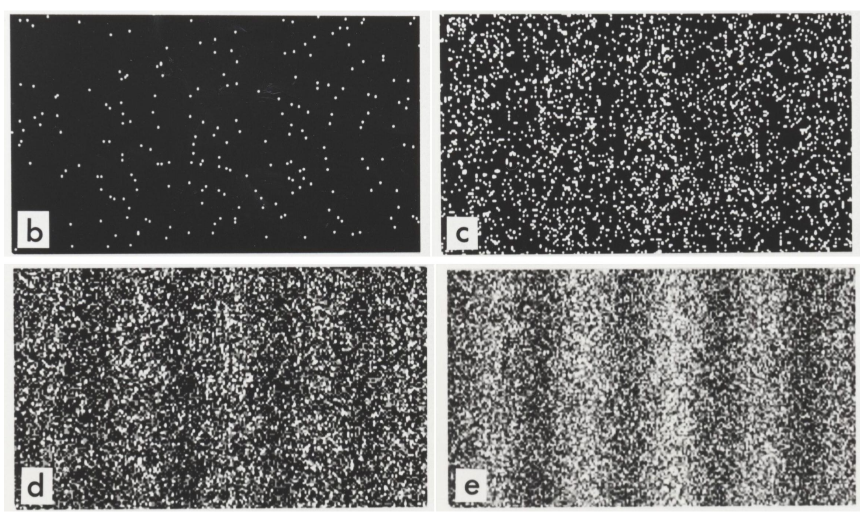

Figure 1: An interference pattern emerges after shooting many electrons on the screen.

going in a different direction. By using a computational procedure I show that event-by-event simulation of quantum phenomena in a deterministic way is possible and apply this without resorting to any concept of QM! I show the feasibility of event-by-event simulation, in particular the Einstein-Podolsky-Rosen-Bohm (EPRB) experiment, and what kind of implications this has. It is important to realize what is presented here is not about the interpretation or extension of QM. Event-by-event simulations are procedures, not (yet) a theory. The fact that it is possible to simulate quantum phenomena on the level of single events does not mean that QM is wrong or incomplete. It does not have anything to say on this level of detail [2].

The philosophy behind event-by-event simulations is as follows [3]:

1. Reproduce event-by-event the results of quantum theory.
2. Do not rely on a single concept of quantum theory.
3. Satisfy Einstein's criteria of local causality.[1]
4. One-to-one correspondence with the experimental setup.
5. Provide a simple, rational and realistic picture of the processes.
6. Do not require arguments that contradict elementary logic and common sense.

We can no longer distinguish between the data of the simulation or data of the experiment if we can construct an algorithm that satisfies these conditions. An important notice on this philosophy is that we could easily perform a simulation by solving the Schrödinger equation with pseudo-random numbers. But following this course of action will not lead to new knowledge of how the probability distribution is formed from single events. Performing the simulation without solving the Schrödinger equation is the real challenge.

Third methodology

One of the reasons that computer simulation is widely regarded as complementary to theory and simulation, is the success of the Metropolis Monte Carlo method for simulating statistical physics [5]. The similarity between the Monte Carlo simulations and the simulations presented here are that in both instances the procedure is repeated many times in order to obtain the distribution of an unknown probabilistic entity. If we are to accept that computer simulation is indeed a third methodology of physics, the realization of simulating quantum phenomena in an event-by-event way should be no surprise.

[1] For a comprehensive discussion about Einstein's locality versus Bell's locality see [4]

2 EPRB experiment with photons

Einstein, Podolsky and Rosen, known collectively as EPR, originally designed a thought experiment to illustrate that the description of the wave function is incomplete [6]. Later it was reformulated in terms of spin by Bohm and this form of the experiment became known as the EPRB experiment [7]. The EPRB experiment illustrates quantum entanglement and heaps of discussion started about what the measurement of entangled particles means and the interpretation of QM.

Experimental setup

In Figure 2 you see a typical setup of an EPRB experiment with photons. The source emits pairs of photons with opposite but otherwise random polarization. Each photon travels to an observation station where it is detected. The two observation stations are separated spatially and temporally such that the detection of particles at station 1 or 2 can not have a causal effect on the detection of particles at the other station. Before a photon arrives at one of the stations, it passes through an electro-optic modulator that rotates the polarization of the photon by an angle based on a pseudo-random voltage. Each electro-optic modulator is connected to an independent random number generator. After the polarizer the photon will be observed at one of the two detectors, which is regarded as an event. The clock of the station assigns a time-tag to each event. This will discretize time in intervals, where the width of the interval is determined by the time-tag resolution τ [9].

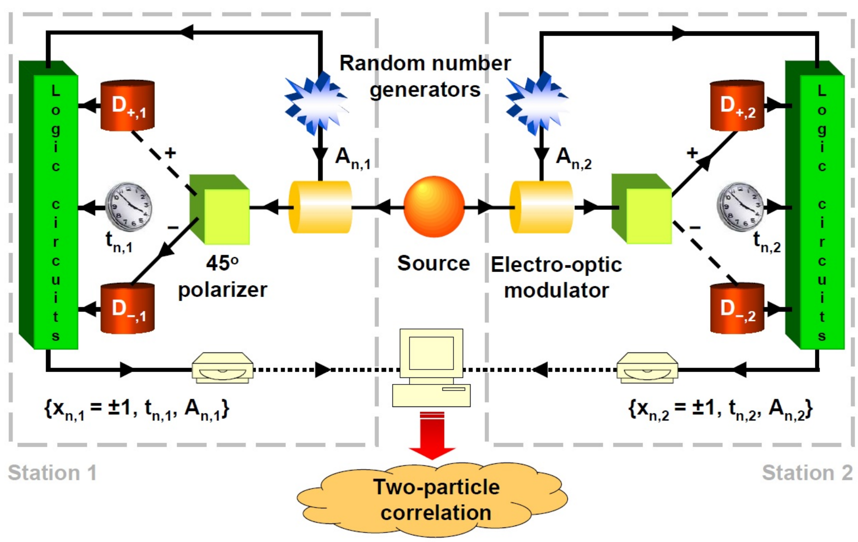

Figure 2: Schematic diagram of a real EPRB experiment with photons [8].

Data gathering

The data of the nth event at station $i = 1, 2$ consist of $x_{n,i} = \pm 1$, specifying at which detector the photon arrives, the time tag $t_{n,i}$, indicating the time of the event, and a two-dimensional unit vector $\mathbf{a}_{n,i}$ which carries the information of the polarization by the electro-optic modulator. Now the complete set of data of the experiment can be written as

$$\Upsilon_i = \{x_{n,i}, t_{n,i}, \mathbf{a}_{n,i} | n = 1 \ldots, N\} \tag{1}$$

The two sets of data $\{\Upsilon_1, \Upsilon_2\}$ can be analyzed long after the collection of the data. By using a time window W, coincidences are identified by comparing time differences $\{t_{n,1} - t_{m,2} | n, m = \ldots, N\}$ [9]. The symbol $\sum'$ is introduced to indicate that all events that satisfy $\mathbf{a}_i = \mathbf{a}_{n,i}$ are summed over. For each pair of directions $\{\mathbf{a}_1, \mathbf{a}_2\}$ of the electro-optic modulators, the number of coincidences between detectors $D_{x,1}(x = \pm 1)$ at station 1 and detectors $D_{y,2}(y = \pm 1)$ at station 2 is given by [8]

$$C_{xy} = C_{xy}(\mathbf{a}_1, \mathbf{a}_2) = \sum_{n,m=1}^{N'} \delta_{x,x_{n,1}} \delta_{y,x_{m,2}} \Theta \left(W - |t_{n,1} - t_{m,2}| \right) \tag{2}$$

Where $\Theta(t)$ is the Heaviside step function. The single-particle averages and the correlation between them are defined by

$$\begin{aligned} E_1(\mathbf{a}_1, \mathbf{a}_2) &= \frac{\sum_{x,y=\pm 1} x C_{xy}}{\sum_{x,y=\pm 1} C_{xy}} \\ E_2(\mathbf{a}_1, \mathbf{a}_2) &= \frac{\sum_{x,y=\pm 1} y C_{xy}}{\sum_{x,y=\pm 1} C_{xy}} \\ E(\mathbf{a}_1, \mathbf{a}_2) &= \frac{\sum_{x,y=\pm 1} xy C_{xy}}{\sum_{x,y=\pm 1} C_{xy}} \\ &= \frac{C_{++} + C_{--} - C_{+-} - C_{-+}}{C_{++} + C_{--} + C_{+-} + C_{-+}} \end{aligned} \tag{3}$$

Where the denominators are the sum of all the coincidences. The value of $C_{xy}(\mathbf{a}_1, \mathbf{a}_2)$ will generally depend on the time-tag resolution τ and the time window W. By tuning the time window W and filling in Eq. (3) based on the data set $\{\Upsilon_1, \Upsilon_2\}$, we are able to produce figures that agree with the two-particle correlation of the singlet state.

Quantum theory

From the axioms of QM we know that repeated measurements on a two-particle spin system yields the following statistical averages for the single-spin system

$$\begin{aligned} E_1(\mathbf{a}) &= \langle \sigma_1 \cdot \mathbf{a} \rangle \\ E_2(\mathbf{b}) &= \langle \sigma_2 \cdot \mathbf{b} \rangle \end{aligned} \tag{4}$$

And the following for the statistical average of the two-spin system

$$E(\mathbf{a}, \mathbf{b}) = \langle \sigma_1 \cdot \mathbf{a} \, \sigma_2 \cdot \mathbf{b} \rangle \tag{5}$$

Where $\sigma_i = (\sigma_i^x, \sigma_i^y, \sigma_i^z)$ are the spin-$\frac{1}{2}$ Pauli matrices for the two particles $i = 1, 2$ and $\{\mathbf{a}, \mathbf{b}\}$ are unit vectors [2].

By performing the experiment we assume that the system is represented by the singlet state $|\phi\rangle = \frac{1}{\sqrt{2}} (|H\rangle_1 |V\rangle_2 - |V\rangle_1 |H\rangle_2)$ of the two spin-$\frac{1}{2}$ particles, where H and V stand for the horizontal and the vertical polarization. Then we have the following statistical averages for the single-spin system

$$\begin{aligned} E_1(\mathbf{a}) &= 0 \\ E_2(\mathbf{b}) &= 0 \end{aligned} \tag{6}$$

And the following for the statistical average of the two-spin system

$$E(\mathbf{a}, \mathbf{b}) = -\mathbf{a} \cdot \mathbf{b} \tag{7}$$

If we now denote the orientation of the two polarizers with α and β we get

$$E(\alpha, \beta) = -\cos 2(\alpha - \beta) \tag{8}$$

When the data of the experiment is processed according to Eq. (3) with the correct time window W, the resulting figure should and will match Eq. (8) which corresponds to the theoretical outcome [9]. I elaborate more on the time window W in the discussion.

3 Simulation model

Here I show how to construct a simulation model of the experiment which satisfies the conditions mentioned in the introduction. The setup is essentially the same as the real experiment treated in the previous chapter (see Fig. 3). The algorithm requires a specification of the different components and a procedure to analyze the data [10].

Source and particles

The source emits particles that carry a vector

$$\mathbf{S}_{n,i} = (\cos(\xi_n + (i-1)\pi/2), \sin(\xi_n + (i-1)\pi/2)) \tag{9}$$

This represents the polarization of the photons that travel to station $i = 1, 2$. The chosen formula is such that the two photons have orthogonal polarizations, i.e. $\mathbf{S}_{n,1} \cdot \mathbf{S}_{n,2} = 0$. The polarization of the photons is completely determined by ξ_n, which is distributed uniformly over the interval $[0, 2\pi]$ by using pseudo-random numbers. Thus the photons are emitted in pairs with a mutually orthogonal but otherwise random polarization [11].

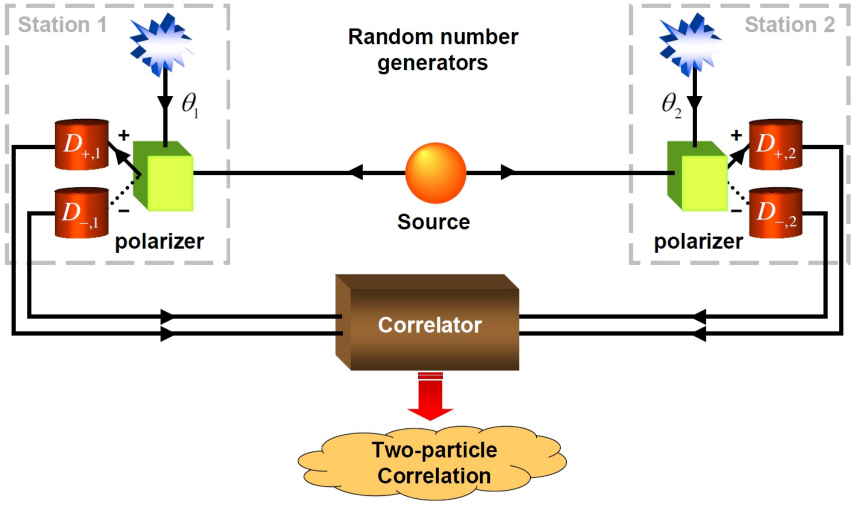

Figure 3: Schematic diagram of the EPRB simulation with randomly polarized photons [3].

Observation stations

The number of different polarization directions we call M. We then use $2M$ random numbers to fill two arrays $(\alpha_1 \ldots, \alpha_M)$ and $(\beta_1 \ldots, \beta_M)$. When the nth pair of photons leave the source we use two different pseudo-random numbers $1 \leq m, m' \leq M$ to select the angles $\theta_{n,1} = \alpha_m$ and $\theta_{n,2} = \beta_{m'}$ [3].

Polarizer

After rotating the photons by $\theta_{n,i}$ the vector describing the photons will change to

$$\mathbf{S}_{n,i} = (\cos(\xi_n - \theta_{n,i} + (i-1)\pi/2), \sin(\xi_n - \theta_{n,i} + (i-1)\pi/2)) \tag{10}$$

The polarizer at station i projects the vector onto its x-axis

$$\mathbf{S}_{n,i} \cdot \hat{\mathbf{x}}_i = \cos(\xi_n - \theta_{n,i} + (i-1)\pi/2) \tag{11}$$

Where $\hat{\mathbf{x}}_i$ is the unit vector along the x-axis of the polarizer [10]. The output of the polarizer will depend on the model used to simulate the polarizing beam splitter. I present a deterministic model and a pseudo-random model.

For the more simple deterministic model we consider the rule

$$x_{n,i} = \text{sign}(\cos 2(\xi_n - \theta_{n,i} + (i-1)\pi/2) \tag{12}$$

This rule does not comply with Malus law but it can nevertheless reproduce the correlation of the singlet state [10].

The more realistic pseudo-random model is based on the rule

$$x_{n,i} = \begin{cases} +1 & \text{if } \mathrm{r}_n \leq \cos^2(\xi_n - \theta_{n,i} + (i-1)\pi/2) \\ -1 & \text{if } \mathrm{r}_n > \cos^2(\xi_n - \theta_{n,i} + (i-1)\pi/2) \end{cases} \tag{13}$$

Where r_n are pseudo-random numbers between 0 and 1. This rule generates events such that the distribution of events complies with Malus law [11].

Time-tag model

When a photon passes through an electro-optic modulator, it experiences a retardation depending on the polarization of the photon and the rotation of the electro-optic modulator according to Maxwell's equation [12]. The amount of time the photon is delayed is missing a theoretical description [10] and in this model we assume a time delay $t_{n,i}$ which is distributed uniformly over the interval $[t_0, t_0 + T]$. Only the time difference is important, which follows from Eq. (2), thus we can set $t_0 = 0$. The time-tag for the nth event is then $t_{n,i} \in [0, T]$.

What follows is an explicit specification of T. I have used the same specification for T as De Raedt *et al.* [10], because it yields useful results

$$T(\cos(\xi_n - \theta_{n,i} + (i-1)\pi/2) = T_0(\sin 2(\xi_n - \theta_{n,i} + (i-1)\pi/2))^d \tag{14}$$

Where $T_0 = \max_\theta T(\theta)$ is the maximum time delay and defines the unit of time. This leaves d as the free parameter and it will be shown that for specific values of d the output of the model corresponds to the quantum correlation of the singlet state. The specific value of d will however depend on the polarizer used.

Data analysis

The algorithm generates the data set $\{\Upsilon_1, \Upsilon_2\}$ for fixed N and M, just as the real experiment [9]. We choose a time-tag resolution $0 < \tau < T_0$ and a coincidence window $\tau \leq W$ in order to count the coincidences [8]. The correlation count $C_{xy}(\alpha_m, \beta_{m'})$ is set to zero initially. After computing the discretized time tags $k_{n,i} = \lceil t_{n,i}/\tau \rceil$, we increment the count $C_{x_{n,1},x_{n,2}}(\alpha_m, \beta_{m'})$ when $|k_{n,1} - k_{n,2}| < k = \lceil W/\tau \rceil$. Because only then an entangled photon pair is observed as stated in the procedure in the real experiment [9].

4 Simulation results

The algorithm first generates the data set $\{\Upsilon_1, \Upsilon_2\}$ for different angles $\theta_{n,i}$ of the polarizer. Then it computes the coincidences by using Eq. (2). And finally the correlation $E(\alpha, \beta)$ is computed by using Eq. (3). The parameters used for the simulation are; $T_0 = 1, N = 10^6, M = 20, W = \tau = 0.00025$ and thus $k = 1$. The obtained results will depend on the

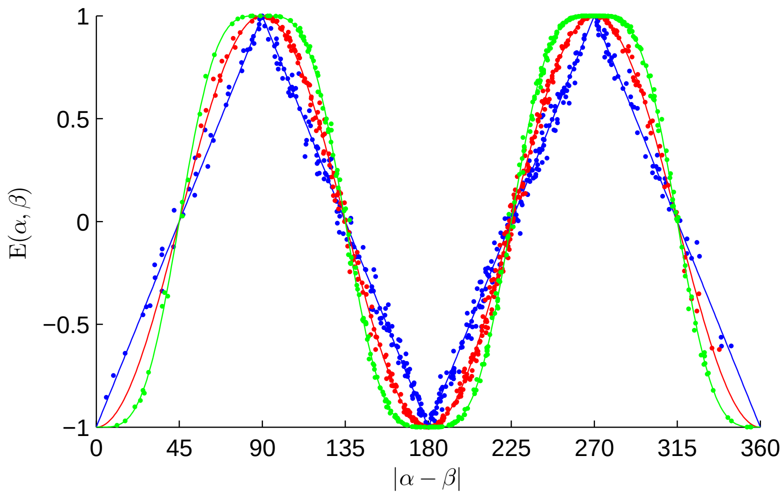

Figure 4: Simulation result with the deterministic polarizer. The solid lines correspond to the analytical expression which can be obtained by using probability theory [4]. Blue dots ($d = 0$) are the result without using a time window W. Red dots ($d = 2$) reproduces the singlet correlation exactly. Green dots ($d = 4$) shown for completeness, see discussion.

time-tag resolution τ, the time window W and the number of events N, just as the real experiment [9]. For each event pseudo-random numbers are used to select one pair of angles for the polarizers. Several simulations are performed for different values of d. $d = 0, 2, 4$ for the deterministic polarizer and $d = 0, 4, 8$ for the pseudo-random polarizer.

The result of the simulation for the correlation $E(\alpha, \beta)$ with the deterministic polarizer is shown in Fig. 4. In this figure you see a perfect agreement with Eq. (8) for $d = 2$ (shown in red). In Figure 5 the result is shown for the correlation $E(\alpha, \beta)$ with the pseudo-random polarizer. For $d = 4$ (shown in red) you can see the desired agreement with Eq. (8). The results of the other values of d will be discussed in the next section.

The algorithm used is programmed in MATLAB. The code is given in the appendix.

5 Discussion

As seen in the figures the results are in excellent agreement with quantum theory when using a particular value of d. Not using a time window W (effectively the same as putting $d = 0$) yields results that disagree with QM. The fact that using a time window W is essential for the real experiment [9] means we do not have to worry too much about that. The time window W in the real experiment as well as in the simulation is necessary to distinguish between the observation of one two-particle system or two single-particle systems. By varying over different values of d means that we are able to describe more

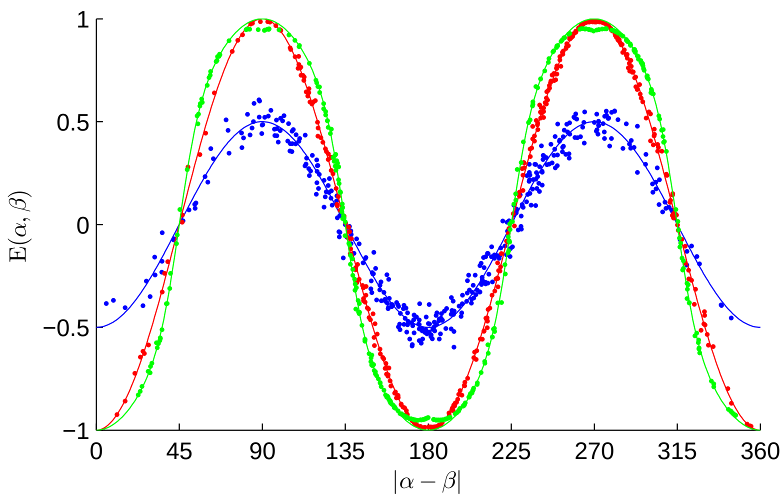

Figure 5: Simulation result with the pseudo-random polarizer. The solid lines correspond to the analytical expression which can be obtained by using probability theory [4]. Blue dots ($d = 0$) are the result without using a time window W. Red dots ($d = 4$) reproduces the singlet correlation exactly. Green dots ($d = 8$) shown for completeness, see discussion.

than QM alone and thus we can consider the result which agrees with the description of an entangled state as a special case. You could say we can go 'beyond quantum' with this classical simulation.

The model described here strictly follows the philosophy mentioned in the introduction. The great feature is the realization of reproducing the correlation for the singlet state without first solving the Schrödinger equation and providing a simple, rational and realistic picture of the processes. It does not matter which model of the polarizer is used if the goal is to reproduce the desired correlation. The pseudo-random polarizer complies with Malus law and in that sense it is more realistic. The algorithm considered here is a slimmed down version of a much broader simulation model using a deterministic learning machine (DLM) [8]. The DLM can simulate a wide range of quantum phenomena and is not restricted to the EPRB experiment or the use of photons. Based on the algorithm one can expect that by making the time-tag resolution τ and the time window W smaller we could improve the agreement with quantum theory. It is very much worth mentioning that we can obtain the analytical expressions for the different simulations by using probability theory. This is covered in great detail by De Raedt *et al.* [4] but lies beyond the scope of my thesis.

I would like to conclude my thesis with a quotation of Einstein's hope for a more complete theory of QM; "I still believe in the possibility of a model of reality, that is to say, of a theory, which represents things themselves and not merely the probability of their occurrence."

References

[1] D. Home. *Conceptual Foundations of Quantum Physics*. Plenum Press, New York, 1997.

[2] L.E. Ballentine. *Quantum Mechanics: A Modern Development*. World Scientific, Singapore, 2003.

[3] S. Zhao. "Event-based Simulation of Quantum Phenomena". PhD thesis. Rijksuniversiteit Groningen, 2009.

[4] H. De Raedt et al. "Event-by-event simulation of quantum phenomena: Application to Einstein-Podolosky-Rosen-Bohm experiments". In: *Journal of Computational and Theoretical Nanoscience* 4 (2007), pp. 957–991.

[5] D.P. Landau and K. Binder. *A Guide to Monte Carlo Simulation in Statistical Physics*. Cambridge University Press, Cambridge, 2000.

[6] A. Einstein, B. Podolsky, and N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Physical Review* 47 (1935), pp. 777–780.

[7] D. Bohm. *Quantum Theory*. Prentice-Hall, New York, 1951.

[8] F. Jin. "Towards a Corpuscular Model of Optical Phenomena". PhD thesis. Rijksuniversiteit Groningen, 2011.

[9] G. Weihs et al. "Violation of Bell's Inequality under Strict Einstein Locality Conditions". In: *Physical Review Letters* 81 (1998), pp. 5039–5043.

[10] K. De Raedt, H. De Raedt, and K. Michielsen. "A computer program to simulate Einstein-Podolsky-Rosen-Bohm experiments with photons". In: *Computer Physics Communications* 176 (2007), pp. 642–651.

[11] H. De Raedt and K. Michielsen. "Event-by-event simulation of quantum phenomena". In: *Annalen der Physik* 524 (2012), pp. 393–410.

[12] M. Born and E. Wolf. *Principles of Optics*. Pergamon, Oxford, 1964.

A MATLAB code

```
% Model for simulating an EPRB experiment
%
% Variables with A and B are for the deterministic polarizer
% and the pseudo-random polarizer respectively.
%
% The simulation runs over various values of d
for d = 0:2:8
    % Model parameters
    M = 20; % Number of angles at station 1 and 2
    N = 1e6; % Number of events generated for each pair of angles
    tau = 2.5e-4; % Time-tag resolution in units of T0 = 1
    k = 1; % Time window W = k*tau

    % Initialize matrices
    angles1 = zeros(1,M); % Angles for station 1
    angles2 = zeros(1,M); % Angles for station 2
    allangles = zeros(1,M^2); % All angle combinations
    countA = zeros(2,2,M,M); % Counts
    countB = zeros(2,2,M,M);
    totalA = zeros(1,M^2); % Total count
    totalB = zeros(1,M^2);
    EA = zeros(1,M^2); % E(a,b)
    EB = zeros(1,M^2);

    % Additional parameters for progress text and file check
    reverseStr = '';
    data_name = [num2str(d),'_',int2str(N),'_',num2str(M)];

    % Check if simulation is done before
    if exist([data_name,'.mat'],'file') ~= 2
        % Initialize tables of angles
        for i = 1:M
            angles1(i) = rand*pi;
            angles2(i) = rand*pi;
        end

        % Generate events
        for z = 1:100
            for i = 1:M^2/100
                for j = 1:N
                    % Starting polarization
                    P1 = 2*pi*rand; % Polarization of particle 1
                    P2 = P1 + 0.5*pi; % Polarization of particle 2

                    % Station 1
                    a1 = ceil(M*rand); % Select an angle
```

```
                    k1 = ceil((1 - c1^2)^(d/2)*rand/tau); % Delay time
                    % Deterministic polarizer
                    c1 = cos(2*(P1 - angles1(a1)));
                    if c1 > 0 % Check for sign
                        dpA = 2; % +1 event
                    else
                        dpA = 1; % -1 event
                    end
                    % Pseudo-random polarizer
                    if (cos(P1 - angles1(a1)))^2 > rand
                        ppA = 2; % +1 event
                    else
                        ppA = 1; % -1 event
                    end

                    % Station 2
                    a2 = ceil(M*rand); % Select an angle
                    k2 = ceil((1 - c2^2)^(d/2)*rand/tau); % Delay time
                    % Deterministic polarizer
                    c2 = cos(2*(P2 - angles2(a2))); % Check for sign
                    if c2 > 0
                        dpB = 2; % +1 event
                    else
                        dpB = 1; % -1 event
                    end
                    % Pseudo-random polarizer
                    if (cos(P2 - angles2(a2)))^2 > rand
                        ppB = 2; % +1 event
                    else
                        ppB = 1; % -1 event
                    end

                    % Count
                    if abs(k1 - k2) < k % Required rule for count
                        countA(dpA,dpB,a1,a2) =  countA(dpA,dpB,a1,a2) + 1;
                        countB(ppA,ppB,a1,a2) =  countB(ppA,ppB,a1,a2) + 1;
                    end
                end
            end
            % Progress text
            msg = sprintf('Processed %g%%', z);
            fprintf([reverseStr, msg '%%']);
            reverseStr = repmat(sprintf('\b'), 1, length(msg));
        end

        % Calculate E(a,b)
        for a2 = 1:M
            for a1 = 1:M
                allangles((a2 - 1)*M + a1) = angles1(a1) - angles2(a2);

```

```
                % Deterministic polarizer
                totalA((a2 - 1)*M + a1) = countA(1,1,a1,a2) +...
                 countA(2,2,a1,a2) + countA(1,2,a1,a2) + countA(2,1,a1,a2);
                EA((a2 - 1)*M + a1) = countA(1,1,a1,a2) +...
                 countA(2,2,a1,a2) - countA(1,2,a1,a2) - countA(2,1,a1,a2);
                if totalA((a2 - 1)*M + a1) > 0
                    EA((a2 - 1)*M + a1) = EA((a2 - 1)*M + a1)/...
                     totalA((a2 - 1)*M + a1);
                end

                % Pseudo-random polarizer
                totalB((a2 - 1)*M + a1) = countB(1,1,a1,a2) +...
                 countB(2,2,a1,a2) + countB(1,2,a1,a2) + countB(2,1,a1,a2);
                EB((a2 - 1)*M + a1) = countB(1,1,a1,a2) +...
                 countB(2,2,a1,a2) - countB(1,2,a1,a2) - countB(2,1,a1,a2);
                if totalB((a2 - 1)*M + a1) > 0
                    EB((a2 - 1)*M + a1) = EB((a2 - 1)*M + a1)/...
                     totalB((a2 - 1)*M + a1);
                end
            end
        end

        % Sort data
        [allangles, perm] = sort(allangles);
        EA = EA(perm);
        EB = EB(perm);

        save(data_name,'allangles','EA','EB')
    else
        load([data_name,'.mat'])
    end
end

makefigure
```