Correlations and EPR experiments, evaluation of a local polarizer model.

Pierre Leroy

Abstract: This paper studies the correlations and results produced in Bell inequalities by a local polarizer model. It shows that the local model produces correlations conforming to classical theory, and under certain conditions conforming to quantum mechanics predictions. It also shows that it can produce small amplitude violations of Bell inequalities due to stochastic variations. These amplitudes are evaluated.

Introduction.

The latest EPR experiments performed produce a violation of Bell inequalities.
These are however of low amplitude. [1] [2].
This document studies whether these amplitudes could be produced by a local model.

This is done by simulations using a flawless EPR configuration.
The results are compared to those predicted by quantum mechanics.

---

1. Correlations produced by the local model.

This first part evaluates the correlations produced by the local model.
This one is described in another document [3].

The model is with hidden variables, and uses 2 variables associated to the photon.
These are denoted p and q, with p representing the angle of polarization of the photon, and q a local physical quantity.

These two variables are the information coupling link between the two photons of a pair, making it possible to produce non-random detections correlations.

For the polarization variable p, value is imposed by the polarization produced by the parametric conversion crystal and by the polarizer used to spatially separate the photons of the pair in the case of using a collinear source.

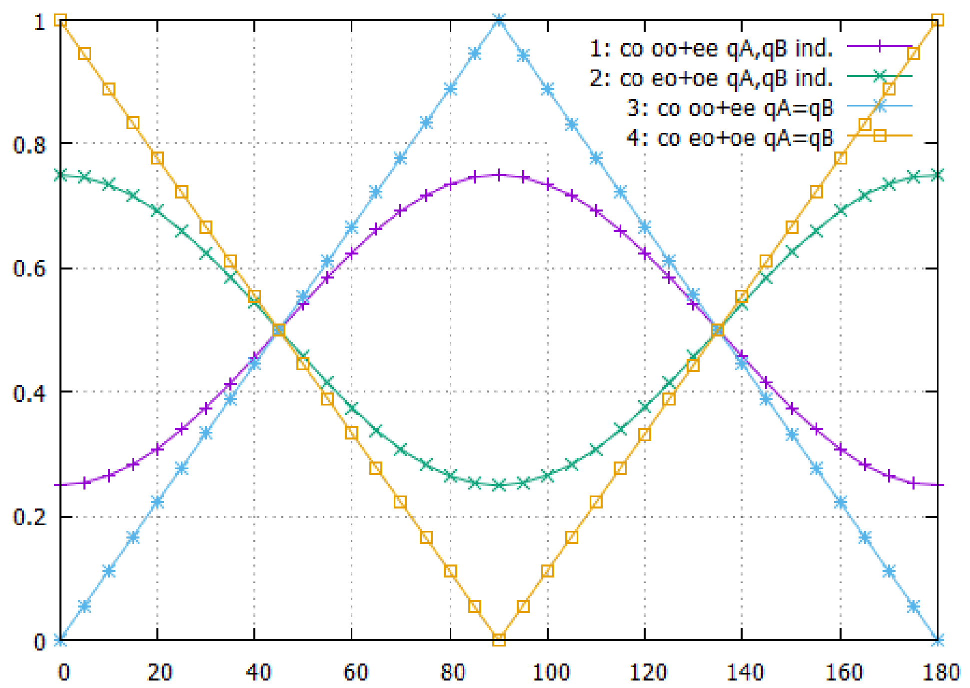
Coupling is maximum for this variable, the two photons having a constant polarization difference (PI/2).

For the variable q, the coupling can vary.
Noting:
qA = value of q for the photon going to the measuring station A (for Alice)
qB = value of q for the photon going to the measuring station B (for Bob)

Maximum coupling is achieved by setting qA = qB.
Zero coupling is achieved by initializing qA and qB with two independent random values.

The following graph shows the coupling effect produced by the variable q on the correlations, with zero and maximum coupling.
The simulation uses a pulsed source and names e and o the outputs of the polarizers.

Graph 1




```
X axis: Angle between polarizers in degrees.
Y axis: Pair detection ratio.
Curve 1: (oo + ee) / N. qA and qB are independent.
Curve 2: (oe + eo) / N. qA and qB are independent.
Curve 3: (oo + ee) / N. qA = qB.
Curve 4: (oe + eo) / N. qA = qB.
With N = oo + ee + oe + eo (total number of pairs detected)
Note: The difference in polarization between the two photons of a
pair is PI/2.
```

We see on the graph that for a zero coupling by the variable q (the values qA and qB are independent), we obtain correlation curves $\cos^2/\sin^2$ of amplitude ½.
This is consistent with the correlations predicted by Maxwell's theory between two distinct polarized sources.

For maximum coupling (qA = qB), we obtain trianglular correlations which show that the system is local.
This coupling mode is the one which produces the most detection correlations, and will be used subsequently in this document.
It assumes that the q value carried by the photons is the same if the two photons of a pair are produced by a parametric down conversion.

The correlations produced with a zero coupling by q (independent qA and qB) are described in the appendix in "Correlations produced using only p".

1.1 Influence of the detection rate on the correlations.

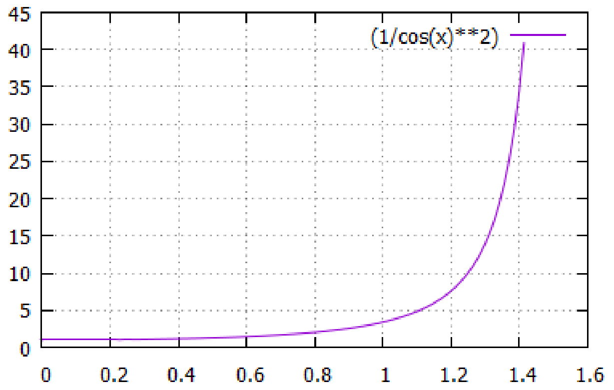
The curves of graph 1 are obtained by simulating a perfect detection of all the photons and by simulating 4 detectors.

By simulating quantum mechanics, we consider a detection probability dependent on the sensitivity of the detectors.
Non-detection is then a random phenomenon.
It is not possible to link the non detection with a property of the photon, because that will require to consider "hidden" variables not foreseen by the theory.

From this perspective, the non-detections will only have the effect of producing simple detections, or an absence of measurement (denoted uu, for **u**ndetected + **u**ndetected), but without altering the shape of the correlation curves produced by the pairs detected.
This also implies that the non-detections do not depend on the difference in angle between the polarizers.

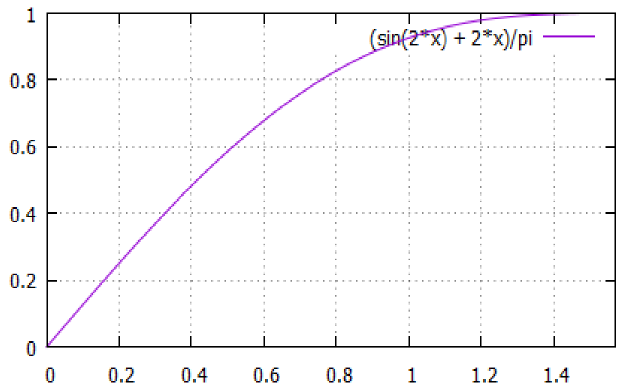
1.2 Sensitivity of detectors.

Graph 2



X axis: rp
y axis: 1/cos(rp)²

Graph 3



X axis: tg [0..PI/2]: Detector trigger threshold.
Y axis: dr: Global detection rate.

For the local model studied, we will consider that the triggering of the detector does not depend on randomness, but on local parameters associated with the photon.

Referring to the document on the polarizer [3], we will assume that the triggering depends on a state of "compression/stretching" of the wave packet of the photon, and assume that the more the photon has been "stretched", the less it will be detectable.
The amplitude of this state depends on a variable noted **rp** and is equal to 1/cos(**rp**)² (graph 2), rp being the repolarization angle that the photon undergoes while passing through the polarizer.
This amounts to assuming that the more a photon has been repolarized, the less detectable it will be.
This interpretation is arbitrary, but due to the lack of a current theoretical model linking the local polarization method to an established theory, it seems appropriate.

A trigger threshold of the detector noted **tg** in range [0..PI/2] can then be compared with |rp|, and the detection will be made if |rp| < tg. (Reversed threshold)

1.3 Global photon detection rate.

By using this detection threshold **tg** , we can define a variable denoted by **dr** (detection rate), which will determine the global rate of photon detection.
The value of **dr** as a function of **tg** (Graph 3) is equal to the following value:

$$dr = \frac{\sin(2\,tg) + 2\,tg}{\pi}$$

By using the reciprocal function tg = f (dr), we can then define the threshold value **tg** of the detector to obtain a determined detection rate **dr.**

With this detection rule, the detections are no longer random, but depend on rp, which itself depends on the angle of the polarizers.
This has the consequences that the detections and non-detections will be dependent on the angles of the polarizers, and that the correlations will depend on the detection rate dr.

2. EPR correlations:

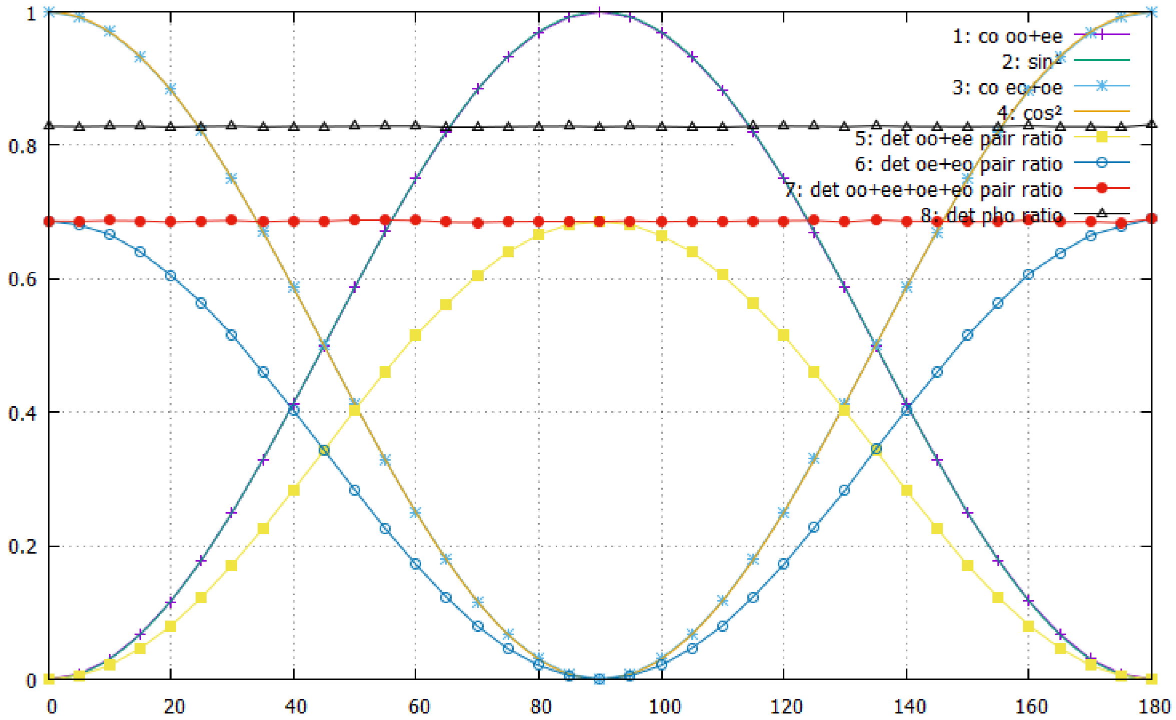
Now that it is possible to set a detection rate for the local model, and using probabilistic detection to simulate the results produced by quantum mechanics, it is possible to compare the correlation curves produced by the two models as a function of global detection rate **dr**.

The following two graphs were produced by setting the detection rate to 0.83. (Exact value is probably 2*sqrt(2) – 2).
This value has the property of producing identical correlation curves and average pair detection rate for the two models.

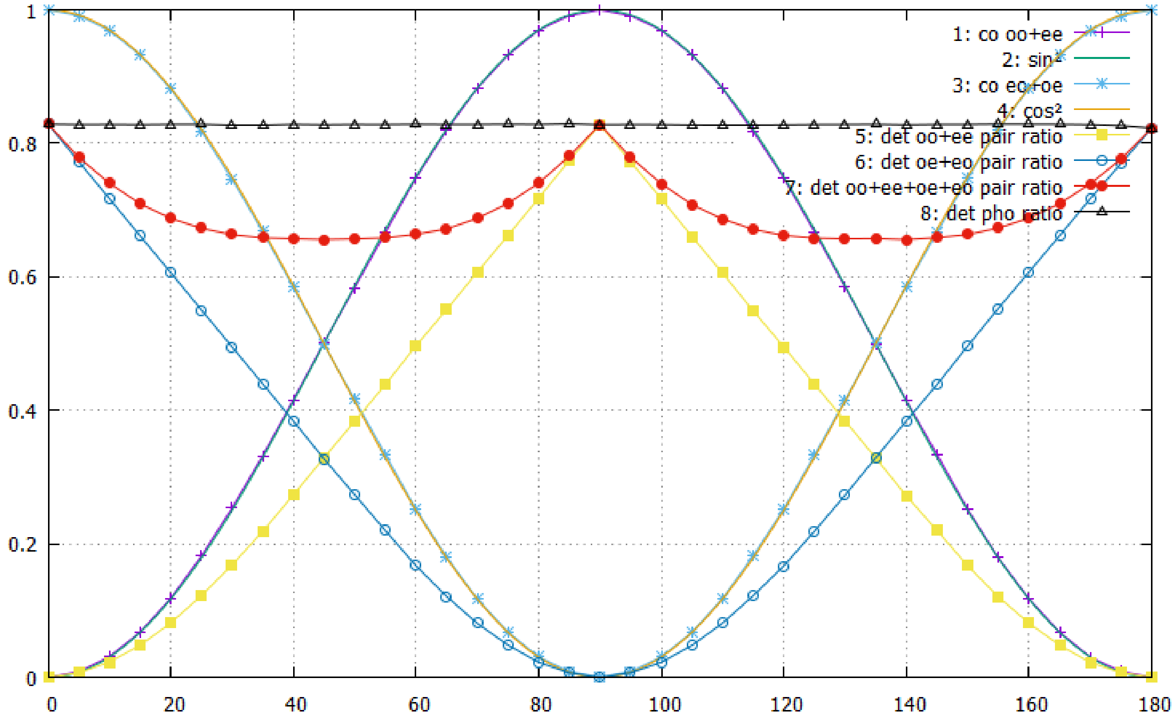
2.1 Correlations of the pairs oo + ee and eo + oe

Graph 4

QM simulation.



Local simulation (qA = qB)



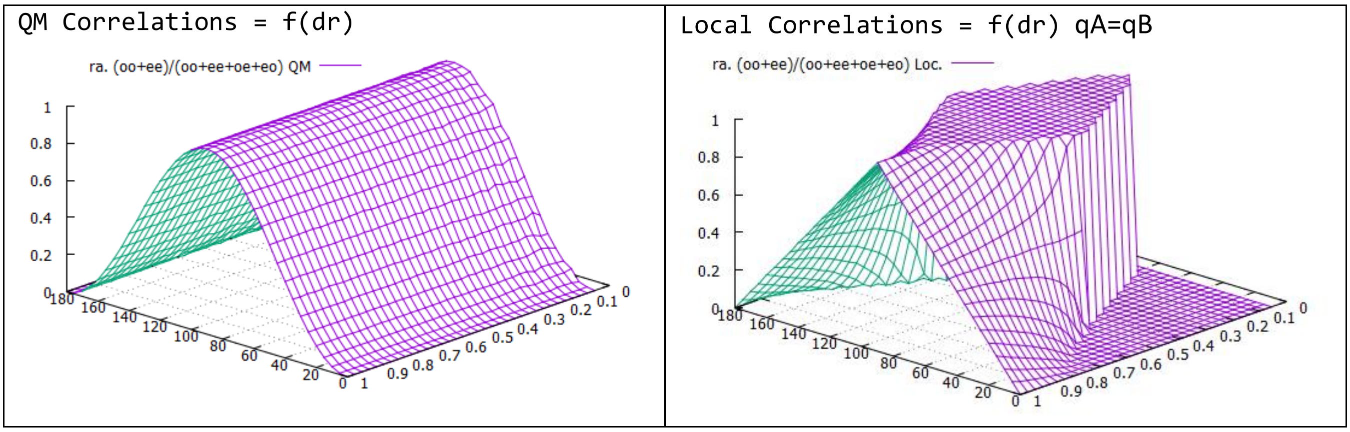
X axis: Angle between polarizers in degrees.
Y axis: Ratio.
Curve 1: (oo + ee) / (oo + ee + oe + eo)
Curve 2: reference sin² curve (merged with curve 1)
Curve 3: (oe + eo) / (oo + ee + oe + eo)
Curve 4: cos² reference curve (merged with curve 3)
Curve 5: (oo + ee) / (number of pairs emitted)
Curve 6: (oe + eo) / (number of pairs emitted)
Curve 7: (oo + ee + oe + eo) / (number of pairs emitted)
Curve 8: (number of photons detected) / (number of photons emitted)

We see on these graphs, that with a detection rate of 0.83, the two models produce identical correlation curves (curves 1 and 3).
The only difference is the detection rate of the pairs (curves 5,6,7), which is variable with the local model, with the maximums located at the angles n*PI/2.
A sensitive EPR experiment using 4 detectors and reproducing the curve on 0..PI could distinguish the two models.

2.2 Correlations as a function of the detection rate.

The following graphs display the shapes of the correlation curves by varying the overall detection rate **dr** between 0.05 and 1.

Graph 5



By simulating quantum mechanics, the shape of the correlation curves is not affected by the detection rate.

For the local model, it varies from a triangle to a shape that becomes rectangular when the detection rate decreases.
Only the rate 0.83 produces exact correlations in $\cos^2/\sin^2$.
It require a detection rate close to 1 to highlight the triangular shape indicating a local model.

3. Bell inequalities.

This part evaluates the results obtained by the local model.

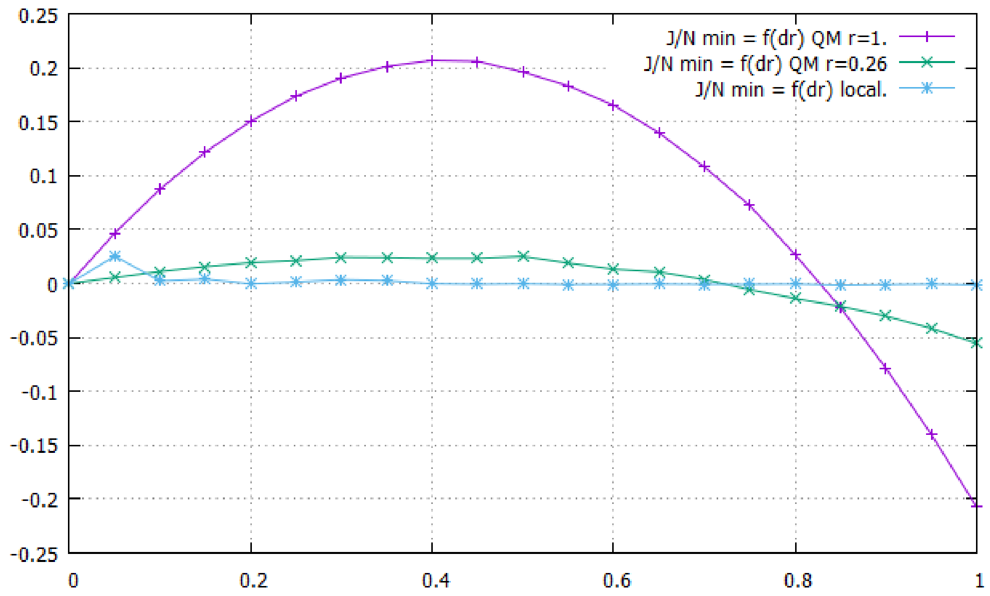
The inequality used is that of Eberhard.
As a reminder, this has the following form:

J = (a1_b2_oe + a1_b2_ou + a2_b1_eo + a2_b1_uo + a2_b2_oo) - a1_b1_oo (4 detectors)
J = (a1_b2_ou + a2_b1_uo + a2_b2_oo) - a1_b1_oo (2 detectors)

The J value must be >= for a local model.
The J/N value expresses the intensity of a violation if it is negative, with a maximum amplitude of -0.207 with QM and r = 1. (r: entanglement level)
N represents the number of measurements made for each combination of angles. (a1b1, a1b2, a2b1, a2b2)

The following graph shows the minimum J/N values that the local model and QM can produce as a function of the detection rate dr.

Graph 6




```
X axis: Global detection rate dr.
Y axis: J/N.
Note: The angles allowing to obtain the smallest J/N are adjusted with QM
and local model for each value of dr.
```

We see on this graph that the local model can produce a value of J/N around 0 whatever the level of detection.

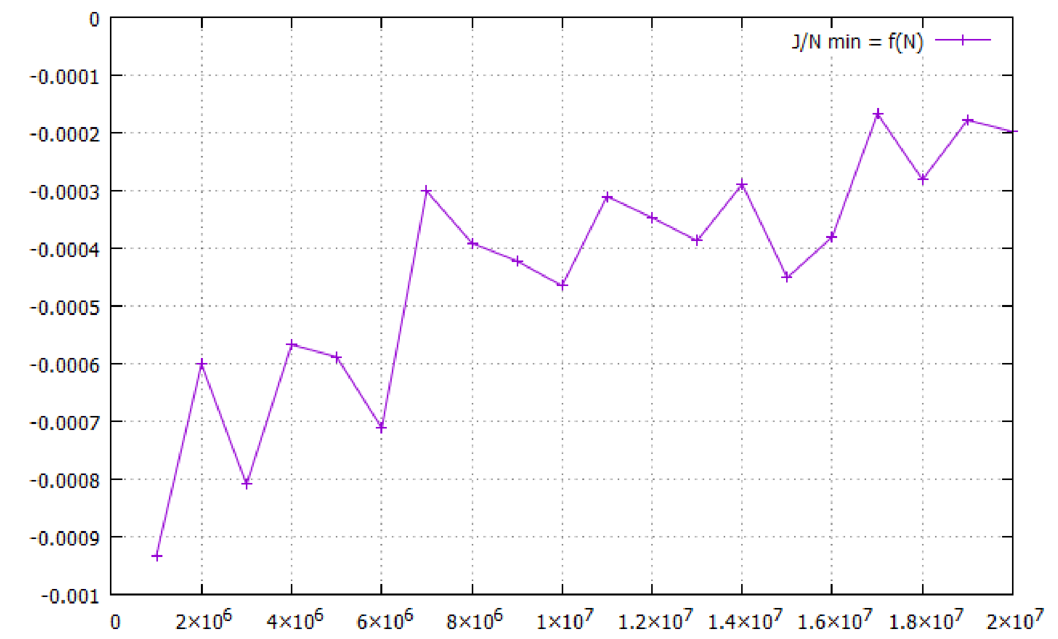
If we increase the display scale around the J/N value, we see that fluctuations around 0 can produce small amplitude violations.
This is due to the stochastic variations of the random generators used to initialize the q values of the photon during emission by the source.

There is no model to define q at the output of a parametric conversion crystal, then q is chosen with a uniform random distribution.
The value of p is fixed at 0 and PI/2 for the photons going towards Alice and Bob, simulating the separating polarizer of a collinear source.

3.1 Stochastic fluctuations.

The following graph shows the intensity of the violations that can be produced by the stochastic fluctuations.
This depends on the number of measurements used to evaluate the inequality (N), and tends towards 0 when N tends towards infinity, this minimizing the average stochastic effect.

Graph 7



X axis: N used to evaluate the inequality.
Y axis: J/N maximum intensity of the violation.
Note: Each inequality test is repeated 20 times for each value of N, and the lowest J/N value obtained is used.
N varies between 1,000,000 and 20,000,000

We notice that the convergence towards 0 is slow, and that the possible level of violation remains relatively high with values of N of 20*10^6.

3.2 Ratio of inequality violations.

This part is interested in the probability of obtaining a negative value of J/N, producing an apparent violation with the local model.

The local model being local, it cannot produce a stable violation, and the average of the J/N values during a series of tests quickly tends towards 0.
However, it can produce, depending on a given N value and the seed of the RNG generator, different amounts of positive and negative results.

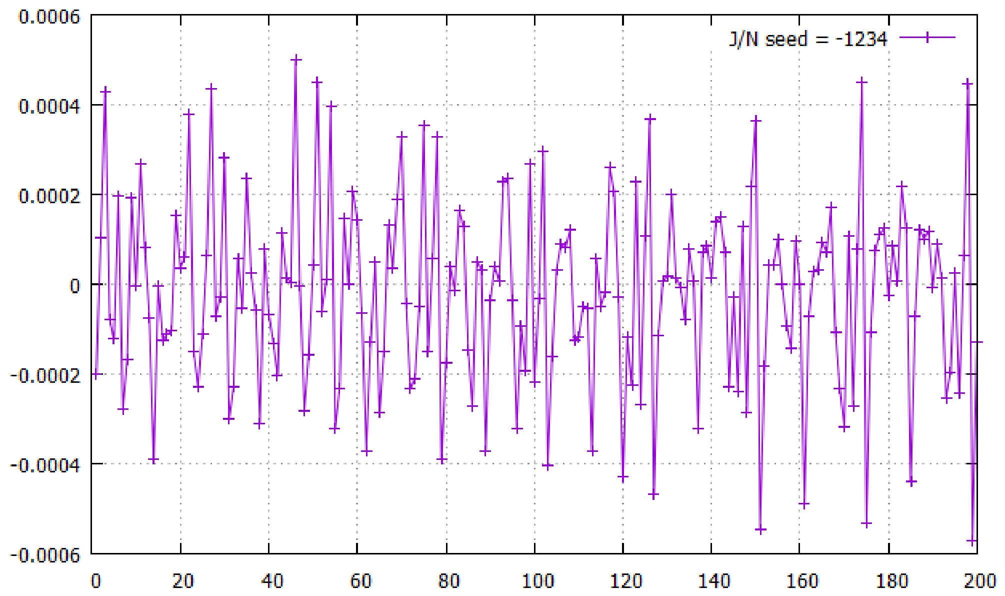
3.3 Seed effect of RNG.

The following two graphs show the number of positive and negative results obtained during a sequence of inequality tests.

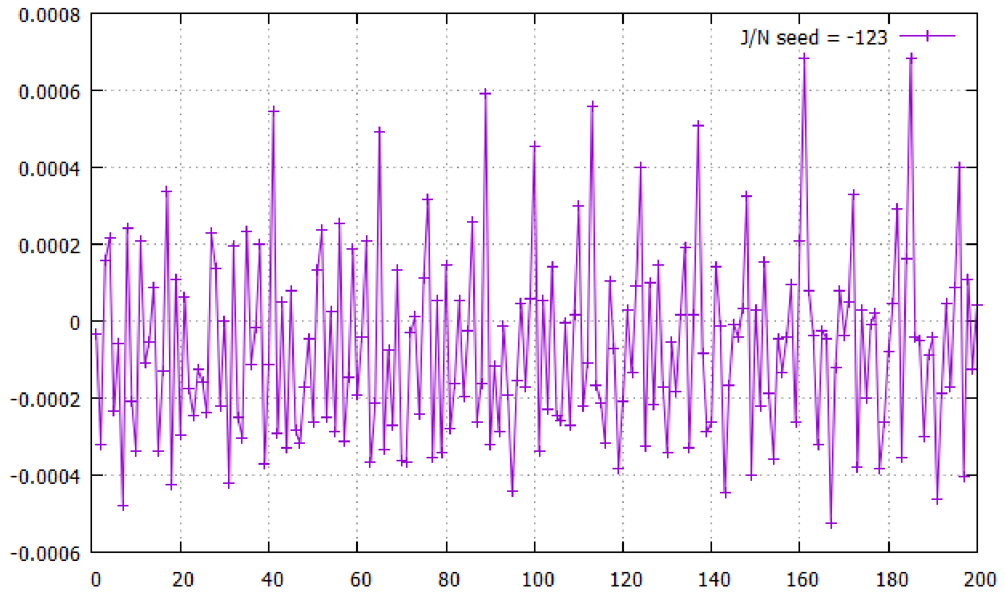
They represent 200 tests of the inequality for the same combination of angles with N = 10,000,000.
They make it possible to evaluate a term of “positivity” defining the rate of positive results obtained on the number of tests carried out.

Graphs 8a and 8b

Graph 8a



Graph 8b




```
X axis: Number of the inequality test.
Y axis: J/N.
```

We see on these graphs that the seed can significantly influence the rate of positive/negative results obtained.

With a seed RNG = -1234 (graph 8a), we obtain a positivity of 0.505, or 49.5% of the tests which produce a violation.
With a seed RNG = -123 (graph 8b), we obtain a positivity of 0.37, or 63% of the tests which produce a violation.

In the second case, we see that the results are more often negative than positive.
However, the average value of J/N nevertheless tends rapidly towards 0 because the average amplitude of negative values is lower than those of positive values.

These two graphs show the importance of reproducing an experiment a sufficient number of times to ensure the stability of a result.
One option is to subdivide a set N of measures in order to do the inequality test several times. However, this has the drawback of reducing the value of N, thus increasing the instability of the results.

4. Double detection flaw.

This part underlines an effect produced by the local model, which can generate an artificial violation of inequalities if it is not taken into account.

As indicated previously, the detections are dependent on the difference in angles between the polarizers, which implies that the non-detections are also dependent.

We can then consider an effect which occurs if during a measurement the detectors receive more than one pair emitted by the source.

In the case of reception of a single pair, if the measurement is uu (double non-detection), this can be taken into account in the measurement count.
In the case of a reception of two pairs, if the first measurement is uu, the second pair can produce a valid measurement obscuring the first pair.

The sampling is then no longer fair, because the rate of uu pairs is no longer valid, and is replaced by other measurements, depending on the difference in angle between the polarizers.

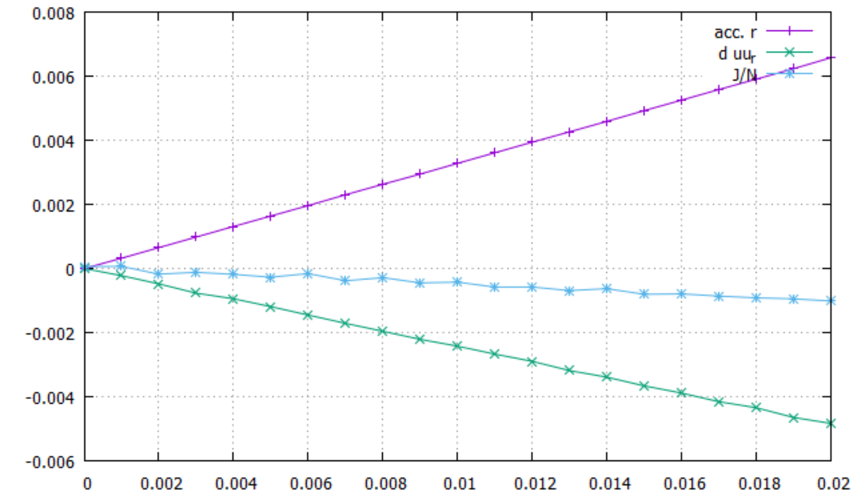
This effect alone can produce a stable inequality violation.

However, double detections also produce measurement errors by mixing the detections if the first measurement produce detections.
Some errors are undetectable, and others produce measures called "accidental".
These accidental measurements correspond to more than one detection on the same detector, or to a measurement on two detectors of the same arm if the experiment has 4 detectors.

The simulations show that if the accidental measurements are counted as a single detection, this compensates for the effect produced by the unfair sampling and that no violation is possible.
However if the accidental measurements are counted as a uu measurement and are normalized in the counters, this produces a stable violation.
It is therefore necessary to consider accidental measurements as a non-random effect, but as a possible consequence of double detections caused by multiple emissions from the source.

The following graph shows the intensity of violation produced if the accidental measurements are not counted as simple detections.

Graph 9




```
Axis x: Rate of double pairs received on the detectors. [0..2%]
Axis y: Ratio or J/N.
acc.r curve: Accidental detection rate.
d uu curve: Variation rate of uu measurements.
J/N curve: Stable J/N result of the inequality.
```

4.1 Double emissions by the source.

In order for double detections to occur, the source must emit a certain proportion of double pairs on the detection areas.
This will depend on the collection/detection surface of the photons, and on the probability that the source has to emit at least one pair.

If we consider a collinear source, the probability of producing a pair towards the detectors will depend mainly on the power of the pump beam and the thickness that it passes through in the crystal.
If we denote by pr_e the probability of producing a pair, the probability of producing two pairs will be at least $(pr\_e)^2$.
Thus, for example with a pulsed source having a probability of producing a pair for 10 emission requests, there will be a probability of producing two pairs every 100 requests, or 1%.

It should be noted that measuring the rate of double transmissions is particularly difficult if the two pairs are issued with very close times.
This is due to the fact that the detectors have a reactivation time which does not allow them to detect two temporally close photons.
The only reliable method is to evaluate it through the rate of accidental measurements produced on separate detectors using a 4 detectors configuration.

5. Summary and conclusion.

Simulations performed with the local model show that it can produce low amplitude violations of Bell inequalities with a flawless experiment.
The probability of producing these violations can be greater than ½ and can occur regardless of the sensitivity of the detectors.
The magnitude of the violation then depends on the number of measures used to calculate the result of the inequality. (N)

The simulations also highlight the effect that double detections can produce, showing that they can produce a stable violation if the accidental measurements they produce are considered as random.

This study may help in the interpretation of the results produced in future EPR experiments.
It is not valid to consider an infinitesimal probability that a local model can produce a violation.
It is in fact necessary that the intensity of the violations obtained in an experiment be of greater amplitude than those which the stochastic variations of a local model can produce in order to conclude that non-locality is necessary to explain a result. (Graph 7)

References.

[1] B. G. Christensen (2013) "Detection-Loophole-Free Test of Quantum Nonlocality, and Applications"
https://arxiv.org/abs/1306.5772
[2] M. Giustina (2015) "Significant-Loophole-Free Test of Bell's Theorem with Entangled Photons"
https://arxiv.org/abs/1511.03190
[3] P. Leroy (2022) "Malus' law photon by photon, a deterministic method"
https://vixra.org/abs/2201.0019
[4] B Dalton (2001) "Law of Malus and Photon-Photon Correlations: A Quasi-Deterministic Analyser Model"
https://arxiv.org/pdf/quant-ph/0101127.pdf

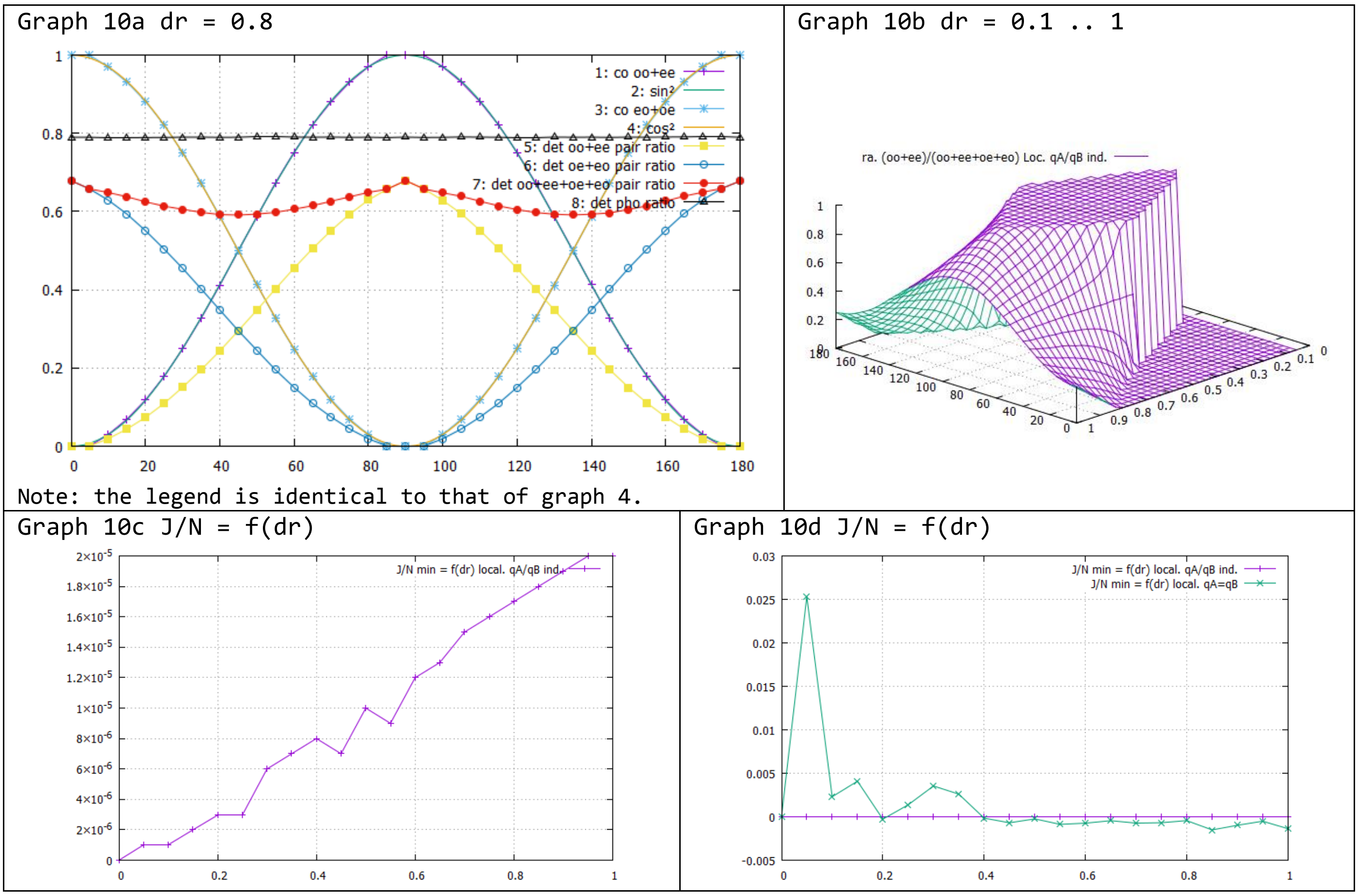
Appendix.

Correlations produced using only p.

This part is placed in the appendix because it does not seem to be able to explain the experimental results. Correlations depend only on photon polarization p, the q variables of each photon in a pair being initialized with independent random values.

The two following graphs show the correlations obtained for a detection rate 0.8 and a varying one between 0.05 and 1 (dr)

Graph 10



With a detection rate close to 0.8, the cos²/sin² correlations of amplitude 1 are also produced. (Graph 10a)
However, the pair detection rate is lower than with qA = qB. (Compared with Graph 4, curves 5,6,7).

This lower pair detection rate always makes it possible to produce a J/N value close to 0 whatever the value of dr. (Graph 10c)
However, unlike the case qA = qB, the stochastic variations become very weak and no longer produce violations. (Graph 10d)

We notice in graph 10b, that the more the detection rate approaches 1, the more the amplitude of the correlations is reduced, and produce the classic correlations provided by Maxwell's theory when the detection rate is equal to 1.

This interpretation of the correlations produced only by polarization could make it possible to eliminate the need to associate a hidden variable q with the photon.
It would then be necessary to locally generate the value of q at the level of each polarizer, this one having to have a uniform random distribution.

However, this would not make it possible to produce correlations that produce sensitive violations of inequalities, unless we suppose that the value of q can be locally produced with identical random values for qA and qB.

This could possibly be done by assuming identical spatial perturbations at the level of the two polarizers, making it possible to define q.
However, this would pose the problem for polarizations shifted in time, for which the perturbations would be different.
This would also pose the problem of distinguishing between a pair emitted by a parametric source and two distinct polarized sources, which would then produce identical correlations.

It therefore seems necessary that the variable q be attached to the photon.
In addition, as described in the polarizer document, this allows operation that does not require random sources, which also makes it possible to propagate the coupling of the correlation information through several polarizers.

A model described by B Dalton [4] uses two distinct random sources to produce the coincidence correlations.
This model has the advantage of using an established physical model based on Stokes parameters.
Perhaps it would be possible to modify it in order to remove the random sources.

Source code (C code).

The following program simulates 100 times an EPR test with two detectors.
It can be used to evaluate the magnitude of violations produced as a function of the N value used to calculate Eberhard's inequality.
The average value of the positive and negative amplitudes is displayed.
Compilation with GCC can be done with the following command:
gcc -O2 epr_eb_test.c -o epr_eb_test.exe

The code can be downloaded here: epr_eb_test.c

```
// Evaluate stochastic violations with Eberhard test for given N and RNG seed.
// A dual detectors EPR configuration is used.
#include <stdio.h>
#include <math.h>

#define PI 3.14159265358979323846
#define DEG_TO_RAD(d) (((d)*PI)/180.0)          // degree to radians conversion

// coded values for ordinary/extraordinary out
#define OUT_O 0
#define OUT_E 1

// photon data
struct pho_t
{
  double p;                                     // hidden variable 1 (polarization)
  double q;                                     // hidden variable 2 (to be defined)
  double rp_abs;                                // |rp| value used for detectors threshold.
  int pol_out;                                  // polarizer out coded id (OUT_O/OUT_E)
};

// ------------------------------------------------
// random generator, initialize q [-1..1]
// Microsoft C library RNG algorithm is used.

static unsigned long n_seed = -123;             // default seed

static void srand(int seed) { n_seed = seed; }

// return random value for q [-1..1]
static double rand_q(void)
{
  n_seed = n_seed * 214013L + 2531011L;
  return ((n_seed >> 16) & 0x7fff)*(2.0/0x7fff) - 1;
}

// --------------------------------------------------
// Polarizer local model

// Simulate source PBS output
```

```
static void emit_photon_pair(struct pho_t *a, struct pho_t *b)
{
  double q = rand_q();                          // q for Alice and Bob (qA = qB)

  a->p = 0;                                     // Alice polarisation angle
  a->q = q;                                     // Alice q

  b->p = PI/2;                                  // Bob polarisation angle
  b->q = q;                                     // Bob q
}

// Polarize photon with a_pol polarizer oriented angle
static void polarize(struct pho_t *pho, double a_pol)
{
  int o;
  double rp;                                    // repolarisation value
  double ad = pho->p - a_pol;                   // angle diff photon polarisation/polarizer
  double s = ad + (PI - acos(pho->q))*0.5;      // angle sum define out (-PI..PI+PI/2 range)

       if (s >=  PI)    { o = OUT_O; rp = ad -   PI; }
  else if (s >=  PI/2)  { o = OUT_E; rp = ad - PI/2; }
  else if (s >=  0)     { o = OUT_O; rp = ad;        }
  else if (s >= -PI/2)  { o = OUT_E; rp = ad + PI/2; }
  else                  { o = OUT_O; rp = ad +   PI; }

  // define results
  pho->pol_out = o;                             // define output
  pho->rp_abs = fabs(rp);                       // define detectability = |rp|

  // The propagation of the p and q values is not necessary for EPR simulation.
  // Code is disabled to optimize speed.
#if 0
  // update p
  if (o == OUT_O)
    pho->p = a_pol;                             // o align polarization
  else
    pho->p = fmod(a_pol + PI/2, PI);            // e align polarization

  // update q
  {
    double c = cos(rp);                         // get cos(rp)
    if (rp >= 0)
      pho->q = (pho->q + 1)/(c*c) - 1;
    else
      pho->q = (pho->q - 1)/(c*c) + 1;
  }
#endif
}

// Convert detection ratio [0..1] to trig level for detectors. [0..PI/2]
// As reciprocal function tg = f(dr) is unknown, a numerical interpolation is used.
static double sim_local_det_trig(double dr)
{
  if (dr < 1.0)
  {
    double x = 0.0;                             // tg to find with x = 2*tg
    double dx;                                  // x step
    dr *= PI;                                   // search x for dr*PI = sin(x) + x

    for (dx=PI/10; dx>0.0000001; dx/=10.0)
    {
      while (1)
      {
        double x1 = x + dx;
        double y1 = sin(x1) + x1;               // x1 = 2*tg
        if (y1 > dr)                            // if dx to big, break, reduce dx
          break;
        x = x1;
      }
    }
    return x/2;                                 // tg = x/2
  }
```

```
  return PI/2;                                   // return max value
}

// ------------------------------------------------
// EPR code

// counters for 2 detectors EPR configuration, oo pairs are detected
struct epr_ctr_t
{
  int oo;                                        // Alice detected o + Bob detected o
  int uo;                                        // Alice undetected + Bob detected o
  int ou;                                        // Alice detected o + Bob undetected
  int uu;                                        // Alice undetected + Bob undetected
};

// -----------------------------------------------
// epr test 2 detectors
static void epr_test_2d(int N, double a_pol, double b_pol, double det_trig, struct epr_ctr_t *ctr)
{
  int n;
  for (n=0; n<N; n++)
  {
    struct pho_t pa, pb;                         // Alice and Bob photons
    int a_detect, b_detect;                      // Alice and Bob detection flags (1 = detected)

    // emit + polarize
    emit_photon_pair(&pa, &pb);                  // initialize emitted photon of pair
    polarize(&pa, a_pol);                        // polarize Alice with a_pol angle
    polarize(&pb, b_pol);                        // polarize Bob with b_pol angle

    // detect alice and bob photons if pass o and trig detector
    a_detect = (pa.pol_out == OUT_O) && (pa.rp_abs < det_trig);
    b_detect = (pb.pol_out == OUT_O) && (pb.rp_abs < det_trig);

    // update detection counters
    if (a_detect)
    {
      if (b_detect)
        ctr->oo++;
      else
        ctr->ou++;
    }
    else
    if (b_detect)
      ctr->uo++;
    else
      ctr->uu++;                                 // no detection
  }
}

// ------------------------------------------------
// Eberhard inequality test

// Do test, return J/N
static double eberhard_test(int N, double det_trig, double a1, double a2, double b1, double b2)
{
  double J;
  // declare 0 initialized counters
  struct epr_ctr_t a1_b1 = { 0 };
  struct epr_ctr_t a1_b2 = { 0 };
  struct epr_ctr_t a2_b1 = { 0 };
  struct epr_ctr_t a2_b2 = { 0 };

  // do epr test for angles
  epr_test_2d(N, a1, b1, det_trig, &a1_b1);
  epr_test_2d(N, a1, b2, det_trig, &a1_b2);
  epr_test_2d(N, a2, b1, det_trig, &a2_b1);
  epr_test_2d(N, a2, b2, det_trig, &a2_b2);

  // define result
  J = (a1_b2.ou + a2_b1.uo + a2_b2.oo) - a1_b1.oo;
  return J/N;                                    // return J/N
```

```
}

// main
void main(void)
{
  // ----------------------------------
  // test configuration
  int n_test = 100;                            // count of tests
  int N = 10*1000*1000;                        // N for Eberhard inequality
  double dr = 0.92;                            // detection ratio dr used

  // Eberhard angles used for test
  double a1 = DEG_TO_RAD(161.14);
  double a2 = DEG_TO_RAD(124.78);
  double b1 = DEG_TO_RAD(60.83);
  double b2 = DEG_TO_RAD(86.40);

  // ----------------------------------
  // test init
  double det_trig = sim_local_det_trig(dr);    // define detectors threshold
  double JoN_min = 1.0;                        // min produced J/N
  double JoN_neg_sum = 0;                      // sum of < 0 J/N values
  double JoN_pos_sum = 0;                      // sum of >= 0 J/N values
  int i, n_pos = 0;                            // positive results counter
  srand(-123);                                 // init RNG seed

  // ----------------------------------
  // test run
  printf("Test started.. x%d N:%d seed:%d\n", n_test, N, n_seed);
  for (i=1; i<=n_test; i++)
  {
    double JoN = eberhard_test(N, det_trig, a1, a2, b1, b2);
    if (JoN >= 0)
    {
      n_pos++;
      JoN_pos_sum += JoN;
    }
    else
      JoN_neg_sum += JoN;

    if (JoN < JoN_min)
      JoN_min = JoN;

    // print results
    printf("Test %d/%d: posi:%.2f J/N:%.6f\n", i, n_test, (float)n_pos/i, JoN);
  }
  // display results stats
  printf("Results for N=%d\nMin J/N: %.6f Avg>0: %.6f Avg<0: %.6f\n", N, JoN_min, JoN_pos_sum/n_pos,
JoN_neg_sum/(n_test - n_pos));
}
```

Email : pierrel5@free.fr

Initial version.