HOW-TO-ENTANGLE-CRAYTONS

A tutorial by

BARRY SCHWARTZ,

prepared and placed in the public domain
in the year 2023, and last revised
Tue Oct 3 06:12:34 PM UTC 2023

Containing also a

PROOF

(without recourse to quantum mechanics)
of the correlation coefficient
of a two-channel Bell test experiment.

With special thanks to an anonymous person
for their scientific curiosity.
In our times, scientific method
has been displaced by scientific authority,
and to encounter actual curiosity is rare.

1. Magically Entangled Craytons. What follows are instructions on how to write a program that, on an ordinary computer, will quantum-entangle two variables so that there is action at a distance between them when they are printed out. Normally such a program would require a quantum computer, but the program *you* write will be magical.

I will myself, in the process of instructing you, write a magical C program that does the same thing. Obtaining a C program from the tutorial itself is part of the magic of using `CWEB` to write these instructions.

That the programs we write will be magical is guaranteed us by no less than the 2022 Nobel Prize winners in Physics. Thousands of papers have been published and thousands of volumes printed. Jillions of public dollars have been spent. Experiment after experiment after experiment has been conducted. So what occurs in our programs *must* be nothing less than the magic of entanglement and quantum non-locality (that is, action at a distance).

2. First, a preliminary. We will need a way to pick arbitrary numbers between zero and one, without showing much bias. The method need not be fancy. It will not matter whether zero or one are themselves included. The following algorithm, consisting of a global variable and a function returning a floating point number, will suffice on most modern computers. (Or you could just use your programming language's "random number" facilities.)

```
⟨ arbitrary numbers between zero and one 2 ⟩ ≡
  int a_global_variable = 12345;
  double number_between_zero_and_one( )
  {
    int i = a_global_variable * 75;
    while (i > 65537) i = i − 65537;
    a_global_variable = i;
    return ((1.0 * i)/65538.0);
  }
```

This code is cited in sections 5 and 8.

This code is used in section 21.

3. Another preliminary: the value of π. This is available to C programmers on POSIX platforms as `M_PI`, but I think perhaps not in the C standard itself. (An old FORTRAN trick, by the way, is to use $4.0 * atan(1.0)$.)

```
#define PI 3.14159265358979323846264338327950288419716939937510582097494459
```

4. Now to the magical program itself. The first things we need are the magical variables. These will be of a type called **crayton**, whose value is either *updown* or *sideways*. How to define such a type in your language will vary, but here is one way to define it in C.

```
⟨ the crayton type 4 ⟩ ≡
  typedef enum {
    updown, sideways
  } crayton;
```

This code is used in section 21.

5. We need a special source of **crayton** variables. It produces two **crayton** at a time, one *updown* and the other *sideways*. Which of the two **crayton** is which, however, is sometimes *updown-sideways*, sometimes the *sideways-updown*, without bias. Lack of bias is ensured by use of ⟨ arbitrary numbers between zero and one 2 ⟩.

In the C code, the two **crayton** will be returned in the C version of a record structure. This pair of **crayton** variables will be the pair the program magically entangles.

```
⟨ the crayton source 5 ⟩ ≡
  typedef struct {
    crayton k1;
    crayton k2;
  } crayton_pair;
  crayton_pair crayton_source( )
  {
    crayton_pair pair;
    if (number_between_zero_and_one( ) < 0.5) {
      pair.k1 = updown;
      pair.k2 = sideways;
    }
    else {
      pair.k1 = sideways;
      pair.k2 = updown;
    }
    return pair;
  }
```

This code is used in section 21.

6. As with many a magic trick, we need mirrors. What we need here is the digital equivalent of a device made from a kind of mirror that breaks a beam of light into two beams. But this mirror is also the digital equivalent of a polarizing filter. From the foregoing description it may seem complicated, but in fact the type for the entire mess is just a floating point number capable of representing an angle in radians. The angle is simply how much someone has rotated the angle of the filter. We need a pair of these filters, so let us call the type a **cray_ban**. For our magic, we will use a pair of polarized **cray_ban**.

```
⟨ the cray_ban type 6 ⟩ ≡
  typedef double cray_ban;
```

This code is used in section 21.

7. Such a magic trick as ours also needs smoke. In this case, the smoke comprises classical physics done, by doctors of philosophy in physics or mathematics, so shockingly incorrectly that you go psychosomatically blind. Once you are blinded, the doctors of philosophy can implant illusions into your brain. However, there is not space here for phony mathematics, so we refer you to the quantum physics literature instead.

Having dived into the literature (or better yet *not* having dived into the literature, but merely imagined yourself having done so), please leave yourself a chance to recover your vision. You may need as medicaments the following reminders:

- Let a and b represent propositions, and $a\wedge b$ their logical conjunction. The *definition* of their conditional probability is
$$P(a|b) = P(a\wedge b) \,/\, P(b)$$
This definition is *purely mathematical* and is complete in itself. Nevertheless, if you have read the "smoke" literature, you will have seen that none other than John Stewart Bell, Fellow of the Royal Society, redefined the conditional probability as follows:
$$P(a|b) = \begin{cases} P(a\wedge b) \,/\, P(b) & \text{pretty much never} \\ P(a) & \text{if local causality, beables, socks, heart attacks, } \lambda\text{, etc.} \end{cases}$$
Most individuals seriously familiar with mathematics will recognize this as (literally) a license to declare "proved" any nonsense one wishes, such as $1 = 0$ and $E = mc^9$. The concussion of a Fellow of the Royal Society proudly displaying such a license is what rendered you psychosomatically blind.
- The claim that quantum physics is "irreducible" to classical physics, though usually assumed to be a claim about physics, is actually the *mathematical* claim—and an alarming one—that a quantum physics problem, written in logically equivalent form but in a mathematics other than that of quantum physics, cannot exist, cannot be solved, or will come to a different result! For, once put in the form of a word problem, physics becomes applied mathematics, and "classical physics" becomes merely the application of any and all mathematics to the reasonable solution of such word problems. Despite public address systems blaring pronouncements through billows of smoke, nothing resembling a smidgen of proof of such "irreducibility" has ever been produced. The quantum physics literature, however, does employ *sheer incompetence* in mathematics (similar to John Bell's described above) *to give the impression* of such "irreducibility." The practitioner of such incompetence might merely give up short of a solution, proclaiming, "That is all that can *possibly* be done. Now please run experiments showing *these* are not the results obtained empirically." The encounter of scientists not even *trying* to solve problems causes temporary shriveling of the hypothalamus. Blindness is merely a portion of the psychosomatic injury.

8. A **cray_ban** does not deal with a beam of light, but instead with a **crayton**. It decides which of two ways to send a **crayton** (we will number the ways $+1$ and -1) according to an algorithm that depends on ⟨ arbitrary numbers between zero and one 2 ⟩. Students of optics may recognize this algorithm as the *Law of Malus*, but here we will call it the *Law of Logodaedalus*, because that sounds more magical.

⟨ the Law of Logodaedalus 8 ⟩ ≡
```
int law_of_logodaedalus(cray_ban angle, crayton crayton_that_will_be_sent)
{
  double x;
  int i;
  if (crayton_that_will_be_sent ≡ updown) x = sin(angle);
  else x = cos(angle);
  if (number_between_zero_and_one( ) < x * x) i = +1;
  else i = −1;
  return i;
}
```
This code is used in section 21.

9. In what follows I show what one event of the experiment looks like. There is the one **crayton** source and there are the two **cray_ban**, set respectively to their angles. The source generates a **crayton** pair. Each **crayton** in the pair is put respectively through one of the **cray_ban**. Data is recorded. You can return the data in a record, as in the following C code, or do whatever you prefer.

⟨ an experimental event 9 ⟩ ≡

```
typedef struct {
  crayton_pair pair;
  int way_k1_was_sent;
  int way_k2_was_sent;
} event_data;
event_data experimental_event(cray_ban angle1, cray_ban angle2)
{
  event_data data;
  data.pair = crayton_source( );
  data.way_k1_was_sent = law_of_logodaedalus(angle1, data.pair.k1);
  data.way_k2_was_sent = law_of_logodaedalus(angle2, data.pair.k2);
  return data;
}
```

This code is used in section 21.

10. One wishes to run a series of events, all with one particular pair of **cray_ban** angles, and count the different types of coincidence. For this there is a new record type, the **series_data**, containing the total number of events and the number of each type of event. (The total number of events will equal the sum of the other fields.)

You do not have to use a record type, of course. This is just one way to represent the information.

(By using records, I am avoiding more C-specific features that have no use being in a tutorial such as this one. Anyway, I like using records.)

⟨ the **series_data** type 10 ⟩ ≡

```
typedef struct {
  cray_ban angle1;
  cray_ban angle2;
  int number_of_events;
  int number_of_updown_sideways_plus_plus;
  int number_of_updown_sideways_plus_minus;
  int number_of_updown_sideways_minus_plus;
  int number_of_updown_sideways_minus_minus;
  int number_of_sideways_updown_plus_plus;
  int number_of_sideways_updown_plus_minus;
  int number_of_sideways_updown_minus_plus;
  int number_of_sideways_updown_minus_minus;
} series_data;
```

This code is used in section 21.

11. Thus a series of n events may be run as follows. And it so happens that the **crayton** pairs will be magically entangled!

```
⟨ a series of n experimental events 11 ⟩ ≡
  series_data experimental_series(cray_ban angle1, cray_ban angle2, int n)
  {
    series_data sdata;
    sdata.angle1 = angle1;
    sdata.angle2 = angle2;
    sdata.number_of_events = n;
    sdata.number_of_updown_sideways_plus_plus = 0;
    sdata.number_of_updown_sideways_plus_minus = 0;
    sdata.number_of_updown_sideways_minus_plus = 0;
    sdata.number_of_updown_sideways_minus_minus = 0;
    sdata.number_of_sideways_updown_plus_plus = 0;
    sdata.number_of_sideways_updown_plus_minus = 0;
    sdata.number_of_sideways_updown_minus_plus = 0;
    sdata.number_of_sideways_updown_minus_minus = 0;
    for (int i = 0; i ≠ n; i = i + 1)      /* Do n times. */
    {
      event_data edata = experimental_event(angle1, angle2);
      if (edata.pair.k1 ≡ updown) {
        if (edata.way_k1_was_sent ≡ +1) {
          if (edata.way_k2_was_sent ≡ +1) {
            sdata.number_of_updown_sideways_plus_plus =
                sdata.number_of_updown_sideways_plus_plus + 1;
          }
          else {
            sdata.number_of_updown_sideways_plus_minus =
                sdata.number_of_updown_sideways_plus_minus + 1;
          }
        }
        else {
          if (edata.way_k2_was_sent ≡ +1) {
            sdata.number_of_updown_sideways_minus_plus =
                sdata.number_of_updown_sideways_minus_plus + 1;
          }
          else {
            sdata.number_of_updown_sideways_minus_minus =
                sdata.number_of_updown_sideways_minus_minus + 1;
          }
        }
      }
      else {
        if (edata.way_k1_was_sent ≡ +1) {
          if (edata.way_k2_was_sent ≡ +1) {
            sdata.number_of_sideways_updown_plus_plus =
                sdata.number_of_sideways_updown_plus_plus + 1;
          }
          else {
            sdata.number_of_sideways_updown_plus_minus =
                sdata.number_of_sideways_updown_plus_minus + 1;
```

```
        }
      }
      else {
        if (edata.way_k2_was_sent ≡ +1) {
          sdata.number_of_sideways_updown_minus_plus =
              sdata.number_of_sideways_updown_minus_plus + 1;
        }
        else {
          sdata.number_of_sideways_updown_minus_minus =
              sdata.number_of_sideways_updown_minus_minus + 1;
        }
      }
    }
  }
  return sdata;
}
```

This code is used in section 21.

12. Proof of Entanglement. The "smoke" mentioned earlier contains some techniques for "showing" *absence* of entanglement—which then *fail*, thereby supposedly proving entanglement by inference. Actually the techniques temporarily wither the audience's hypothalamuses, and the audience members are being mind-controlled, as if in a Philip K. Dick novel. However, *you* do not practice mind-control (I hope), and *our* task is different: we must *positively demonstrate* entanglement. Thus we will do nothing less than show that our experiment is empirically consistent with a formula from quantum mechanics: the correlation coefficient for our experimental arrangement. According to the 2022 Nobel Prize winners in Physics, and practically the entire field of quantum physics, this would be impossible unless the **crayton** pairs were entangled. The entanglement, then, *must* be so. Thus the **crayton** pairs were indeed entangled.

So, then, you ask, what is a *correlation coefficient*? It is a value between -1 and $+1$ that gives some idea how interrelated are two functions or sets of data. It is a notion familiar in the field of statistics, but also in the theory of waves, where it indicates the capacity of two waves (if superposed) to form different interference patterns. For this experiment, we want the correlation coefficient comparing the "way the **crayton** was sent" of the two **crayton** in the pair. So let us begin.

Assume the two **cray_ban** settings are ϕ_1 and ϕ_2. The formula from quantum mechanics is then

$$\begin{aligned} \textit{correlation coefficient} &= -\cos\{2(\phi_1 - \phi_2)\} \\ &= -\{\cos^2(\phi_1 - \phi_2) - \sin^2(\phi_1 - \phi_2)\} \end{aligned}$$

or the same formula with the sign reversed, because a correlation coefficient has arbitrary sense. One must be sure to be consistent, but otherwise whether there is a minus sign or a plus sign in front of the whole thing does not matter. I choose the minus sign because, after I stop dissembling and present my own derivation of the correlation coefficient, it will have the minus sign due to how I formulated the Law of Logodaedalus.

The formula itself makes it evident that only the size of the angle between ϕ_1 and ϕ_2 matters, not the direction of the subtraction. It is also clear that the formula is *invariant with rotations of the* **cray_ban** *pair*—it does not matter what the particular angles are, but only what they are relative to each other. Some might also notice that there is a resemblance to the Law of Logodaedalus—this is not accidental, but let us not go into the details. The resemblance is important in the study of optics.

13. What experts in quantum physics tell us is: if we run four series of the experiment, using settings I will list below, and get approximately the results predicted by quantum mechanics, then we will have proved that our **crayton** pairs were entangled.

Actually they do not know about the **crayton** specifically, but only about other objects they do not know how to test this with, so they have invented other tests, such as shouting "LOOK THAT WAY!" and running out of the room. But *we* have the **crayton** and so can run the test. The experts may object, of course. They always object. But let us proceed, nonetheless.

The settings and corresponding correlation coefficients are as follows:

$$\phi_1, \phi_2 = \begin{cases} 0,\ \pi/8 & -1/\sqrt{2} \approx -0.70711 \\ 0,\ 3\,\pi/8 & +1/\sqrt{2} \approx +0.70711 \\ \pi/4,\ \pi/8 & -1/\sqrt{2} \approx -0.70711 \\ \pi/4,\ 3\,\pi/8 & -1/\sqrt{2} \approx -0.70711 \end{cases}$$

14. Now we are going to do some clever stuff. We are going to use the data we have collected, together with the Law of Logodaedalus, to compute the correlation coefficient empirically. More specifically, we are going to use *frequencies of the recorded events* to get estimates of trigonometric functions of ϕ_1 and ϕ_2, which we will then use to compute an approximation of $-\{\cos^2(\phi_1 - \phi_2) - \sin^2(\phi_1 - \phi_2)\}$.

15. Obtaining the frequencies is a simple matter of computing ratios. Given a **series_data** record *sdata*, simply do a bunch of divisions, after converting the integers to rational numbers (floating point, in the C program):

⟨ frequencies of events 15 ⟩ ≡

```
double freq_of_updown_sideways_plus_plus =
    (1.0 * sdata.number_of_updown_sideways_plus_plus) / sdata.number_of_events;
double freq_of_updown_sideways_plus_minus =
    (1.0 * sdata.number_of_updown_sideways_plus_minus) / sdata.number_of_events;
double freq_of_updown_sideways_minus_plus =
    (1.0 * sdata.number_of_updown_sideways_minus_plus) / sdata.number_of_events;
double freq_of_updown_sideways_minus_minus =
    (1.0 * sdata.number_of_updown_sideways_minus_minus) / sdata.number_of_events;
double freq_of_sideways_updown_plus_plus =
    (1.0 * sdata.number_of_sideways_updown_plus_plus) / sdata.number_of_events;
double freq_of_sideways_updown_plus_minus =
    (1.0 * sdata.number_of_sideways_updown_plus_minus) / sdata.number_of_events;
double freq_of_sideways_updown_minus_plus =
    (1.0 * sdata.number_of_sideways_updown_minus_plus) / sdata.number_of_events;
double freq_of_sideways_updown_minus_minus =
    (1.0 * sdata.number_of_sideways_updown_minus_minus) / sdata.number_of_events;
```

This code is used in section 19.

16. From the Law of Logodaedalus, it is possible to use these frequencies to estimate products of the squares of cosines and sines of ϕ_1 and ϕ_2. I leave it as an exercise for the reader to convince themselves of this fact. Here is not a good place to do a proof, and it is good exercise for what Hercule Poirot called "the little gray cells." It is intuitive once visualized, if a person be capable. (Not everyone is. There are people, including the late famous neurologist Oliver Sacks, who cannot even recognize a human face—whereas Sacks's mother, a surgeon, could do a three-dimensional drawing without reference. One must allow for individual variation.) Thus:

⟨ estimates of certain products 16 ⟩ ≡

```
double estimate_of_cos2_phi1_cos2_phi2 =
    freq_of_updown_sideways_minus_plus + freq_of_sideways_updown_plus_minus;
double estimate_of_cos2_phi1_sin2_phi2 =
    freq_of_updown_sideways_minus_minus + freq_of_sideways_updown_plus_plus;
double estimate_of_sin2_phi1_cos2_phi2 =
    freq_of_updown_sideways_plus_plus + freq_of_sideways_updown_minus_minus;
double estimate_of_sin2_phi1_sin2_phi2 =
    freq_of_updown_sideways_plus_minus + freq_of_sideways_updown_minus_plus;
```

This code is cited in section 17.

This code is used in section 19.

17. The following angle-difference identities may be found in reference books, such as the *CRC Handbook of Mathematical Sciences*:

$$\cos(\alpha - \beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$$
$$\sin(\alpha - \beta) = \sin\alpha\cos\beta - \cos\alpha\sin\beta$$

We can obtain estimates of the terms on the right side by taking square roots of the results from ⟨ estimates of certain products 16 ⟩. There are, of course, *two* square roots, one positive and one negative, and so we must know which one to use. However, all of our ϕ_1, ϕ_2 settings are for angles in Quadrant I, and therefore only positive square roots will be needed. Thus:

```
⟨ estimates of the angle-difference functions 17 ⟩ ≡
  double estimate_of_cos_phi1_minus_phi2 =
      sqrt(estimate_of_cos2_phi1_cos2_phi2) + sqrt(estimate_of_sin2_phi1_sin2_phi2);
  double estimate_of_sin_phi1_minus_phi2 =
      sqrt(estimate_of_sin2_phi1_cos2_phi2) - sqrt(estimate_of_cos2_phi1_sin2_phi2);
```

This code is used in section 19.

18. Finally, then, one can estimate the correlation coefficient:

```
⟨ estimate of the correlation coefficient 18 ⟩ ≡
  double estimate_of_correlation_coefficient =
      -((estimate_of_cos_phi1_minus_phi2 * estimate_of_cos_phi1_minus_phi2) -
      (estimate_of_sin_phi1_minus_phi2 * estimate_of_sin_phi1_minus_phi2));
```

This code is used in section 19.

19. There follows a C function that puts together these calculations and turns a **series_data** record into an estimate of a correlation coefficient. Put the operations together similarly, in whatever language you are using.

```
⟨ correlation coefficient estimate function 19 ⟩ ≡
  double correlation_coefficient_estimate(series_data sdata)
  {
    ⟨ frequencies of events 15 ⟩
    ⟨ estimates of certain products 16 ⟩
    ⟨ estimates of the angle-difference functions 17 ⟩
    ⟨ estimate of the correlation coefficient 18 ⟩
    return estimate_of_correlation_coefficient;
  }
```

This code is used in section 21.

20. Next is a procedure that will print out that estimate, along with the nominal value. You will want something similar, but how to print out data varies greatly from one programming language to another. Although I know many programming languages, in this tutorial I cannot help you much. In any case, some of these languages cannot make up their mind how to do output, and every one of them sucks at it.

```
⟨ printing out the correlation coefficient estimate 20 ⟩ ≡
  void print_correlation_coefficient_estimate(series_data sdata)
  {
    printf("cray_ban␣angle1␣␣␣␣␣␣%4.1f␣deg\n", sdata.angle1 * 180.0/PI);
    printf("cray_ban␣angle2␣␣␣␣␣␣%4.1f␣deg\n", sdata.angle2 * 180.0/PI);
    printf("nominal␣corr␣coef␣␣␣␣%+8.5f\n", -cos(2.0 * (sdata.angle1 - sdata.angle2)));
    printf("measured␣corr␣coef␣␣␣%+8.5f\n", correlation_coefficient_estimate(sdata));
  }
```

This code is used in section 21.

21. Finally I will put together my C program, and you can put together your program.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
  ⟨ arbitrary numbers between zero and one 2 ⟩
  ⟨ the crayton type 4 ⟩
  ⟨ the crayton source 5 ⟩
  ⟨ the cray_ban type 6 ⟩
  ⟨ the Law of Logodaedalus 8 ⟩
  ⟨ an experimental event 9 ⟩
  ⟨ the series_data type 10 ⟩
  ⟨ a series of n experimental events 11 ⟩
  ⟨ correlation coefficient estimate function 19 ⟩
  ⟨ printing out the correlation coefficient estimate 20 ⟩
  int main( )
  {
    int n = 10000;      /* Each series will contain n experimental events. */
    series_data sdata1 = experimental_series(0.0, PI/8.0, n);
    series_data sdata2 = experimental_series(0.0, 3.0 * PI/8.0, n);
    series_data sdata3 = experimental_series(PI/4.0, PI/8.0, n);
    series_data sdata4 = experimental_series(PI/4.0, 3.0 * PI/8.0, n);
    printf("\n");
    print_correlation_coefficient_estimate(sdata1);
    printf("\n");
    print_correlation_coefficient_estimate(sdata2);
    printf("\n");
    print_correlation_coefficient_estimate(sdata3);
    printf("\n");
    print_correlation_coefficient_estimate(sdata4);
    printf("\n");
    return 0;
  }
```

22. On POSIX systems, the C program has to be linked with a library of mathematical functions, `-lm`. Also, this probably does not matter, but I wrote the program for a C standard that is not expected to be approved until 2024. The program should be legal under the older standards, but more likely to provoke warnings from your compiler.

23. When I compile and run my program, I obtain the following as my output:

```
cray_ban␣angle1␣␣␣␣␣␣␣0.0␣deg
cray_ban␣angle2␣␣␣␣␣␣22.5␣deg
nominal␣corr␣coef␣␣␣␣-0.70711
measured␣corr␣coef␣␣␣-0.71180

cray_ban␣angle1␣␣␣␣␣␣␣0.0␣deg
cray_ban␣angle2␣␣␣␣␣␣67.5␣deg
nominal␣corr␣coef␣␣␣␣+0.70711
measured␣corr␣coef␣␣␣+0.70980

cray_ban␣angle1␣␣␣␣␣␣45.0␣deg
cray_ban␣angle2␣␣␣␣␣␣22.5␣deg
nominal␣corr␣coef␣␣␣␣-0.70711
measured␣corr␣coef␣␣␣-0.70859

cray_ban␣angle1␣␣␣␣␣␣45.0␣deg
cray_ban␣angle2␣␣␣␣␣␣67.5␣deg
nominal␣corr␣coef␣␣␣␣-0.70711
measured␣corr␣coef␣␣␣-0.70148
```

Thus is entanglement proven! I have entangled **crayton** pairs on ordinary computer hardware. No quantum computer was necessary. Each **crayton** in a pair settled into its individual state non-locally upon measurement of the other.

24. Okay, I Lied. There was actually no entanglement. I am sure there is no entanglement anywhere in the universe. The very notion is as silly as a perpetual motion machine. Entanglement is the wrongest wrong thing there has ever been in the history of physics.

Your program demonstrates that quantum theorists have been *just plain wrong* in their reasoning. One of the root causes is that license John Stewart Bell gave theorists to declare anything "true" that suited their fancy. They have used that license freely. Any attempt to declare their math illegitimate is immediately canceled by the license Bell gave them. The declarer will be slammed with a GISH GALLOP of "*Local causality, beables, socks, heart attacks, loopholes, dichotomic variables, imported hay, Fantastic Voyage, Final Four, ... ! Did you hear me? Tick, tick, tick. I said* LOCAL CAUSALITY HAY SOCKS LOOPHOLES!"

This deluge will serve two purposes. The first is to prevent publication of any paper the declarer may present. The second is, so to speak, to damage the declarer's hypothalamus. The victim, if not hurt too badly, will run off like a dog with tail between legs. I, personally, am psychiatrically disabled and have been rendered temporarily bedridden by such a barrage, despite that what I had said concerned *an electronic signal processing problem*, and so had no quantum mechanics in it whatsoever.

I had thought to say more about what has been perpetrated, but words escape me. Papers and books promoting "entanglement" and "non-locality" are simply worthless. They have no use except as paper pulp. Instead I will do another thing the SMOKE-&-MIRRORS magicians claim cannot be done: for the sake of those capable of reading the mathematics, I will derive the correlation coefficient of our experiment, but using classical physics instead of quantum mechanics.

25. Actually the correlation coefficients for experiments such as this one were derived long ago using the classical theory of wave mechanics! If you assume waves are assemblages of particles, then some "quantum" phenomena can easily be explained classically. There are also other ways in which such "quantum" phenomena might be explained classically as wave phenomena, while letting the waves be continuous substance, as they traditionally have been imagined. For instance, if photodetectors randomly generate photoelectrons in proportion to light intensity, this explains the "quantum" phenomena. (This explanation is due to A. F. Kracklauer.) However, in counterargument: "LOCAL CAUSALITY HAY SOCKS LOOPHOLES!"

If you see what I mean.

The same counterargument will apply to the derivation below. However, at least the derivation will not depend upon wave theory. It will employ more fundamental mathematics.

26. What we are looking for is the correlation coefficient of the "way sent" values $+1$ and -1. The definition of the correlation coefficient is the covariance over the product of the standard deviations. That denominator is merely a normalization, to put the correlation coefficient between -1 and $+1$, and the "way sent" values were chosen so that no such normalization was necessary. Thus the correlation coefficient is equal to the covariance. Call the correlation coefficient ρ and the two "way sent" values τ_1 and τ_2, and let E represent an *expectation*—that is, an average weighted by a probability density function (pdf). Then

$$\rho = E(\tau_1 \tau_2)$$

for some pdf we have to determine. That is, the correlation coefficient is a very carefully weighted average of the products of "way sent" values.

27. I studied the problem casually for some 20 years before finally figuring out how to determine the pdf. But then I decided that determining the pdf was not necessary, after all!

Yes, I had derived the correlation coefficient by determining a pdf, but I shall not reproduce that derivation for you, because it is too complicated. You need an education in electronic signal processing theory to understand it, and even then it makes one's head feel as if it were a muddled fruit at the bottom of a cocktail shaker. (Which school always did to me, in any case. I hate school.) The derivation probably still has bugs in it, the way a computer program that is too complicated seems never to have all the bugs cleaned out. They can be cleaned out, certainly, but the effort is not worth it. It is better to find a new approach.

28. The following much simpler derivation starts by deriving what the SMOKE-&-MIRRORS CLUB seems to believe is all classical physics is capable of deriving: a particular function of the two **cray_ban** settings that is *not* a function of their difference. *This* derivation is tedious but straightforward.

Let k_1 and k_2 represent the **crayton** pair, and ϕ_{01} and ϕ_{02} the **cray_ban** settings. Then the Law of Logodaedalus is

$$\begin{aligned}
P(\tau_1 = +1 \,|\, k_1 = \mathit{updown}) &= \sin^2(\phi_{01}) \\
P(\tau_1 = -1 \,|\, k_1 = \mathit{updown}) &= \cos^2(\phi_{01}) \\
P(\tau_2 = +1 \,|\, k_2 = \mathit{updown}) &= \sin^2(\phi_{02}) \\
P(\tau_2 = -1 \,|\, k_2 = \mathit{updown}) &= \cos^2(\phi_{02}) \\
P(\tau_1 = +1 \,|\, k_1 = \mathit{sideways}) &= \cos^2(\phi_{01}) \\
P(\tau_1 = -1 \,|\, k_1 = \mathit{sideways}) &= \sin^2(\phi_{01}) \\
P(\tau_2 = +1 \,|\, k_2 = \mathit{sideways}) &= \cos^2(\phi_{02}) \\
P(\tau_2 = -1 \,|\, k_2 = \mathit{sideways}) &= \sin^2(\phi_{02})
\end{aligned}$$

The Law of Logodaedalus is obviously consistent, in that

$$\begin{aligned}
P(\tau_1 = +1 \,|\, k_1 = \mathit{updown}) + P(\tau_1 = -1 \,|\, k_1 = \mathit{updown}) &= \sin^2(\phi_{01}) + \cos^2(\phi_{01}) = 1 \\
P(\tau_2 = +1 \,|\, k_2 = \mathit{updown}) + P(\tau_2 = -1 \,|\, k_2 = \mathit{updown}) &= \sin^2(\phi_{02}) + \cos^2(\phi_{02}) = 1 \\
P(\tau_1 = -1 \,|\, k_1 = \mathit{sideways}) + P(\tau_1 = +1 \,|\, k_1 = \mathit{sideways}) &= \sin^2(\phi_{01}) + \cos^2(\phi_{01}) = 1 \\
P(\tau_2 = -1 \,|\, k_2 = \mathit{sideways}) + P(\tau_2 = +1 \,|\, k_2 = \mathit{sideways}) &= \sin^2(\phi_{02}) + \cos^2(\phi_{02}) = 1
\end{aligned}$$

This smidgen of mathematical consistency is present even in the SMOKE-&-MIRRORS literature.

29. The **crayton** source also gives us opportunity to define some probabilities:

$$\begin{aligned}
P(k_1 = \mathit{updown} \wedge k_2 = \mathit{sideways}) = P(k_1 = \mathit{sideways} \wedge k_2 = \mathit{updown}) &= 1/2 \\
P(k_1 = \mathit{updown} \wedge k_2 = \mathit{updown}) = P(k_1 = \mathit{sideways} \wedge k_2 = \mathit{sideways}) &= 0
\end{aligned}$$

These add up to one and so also are consistent. For simplicity, from now on we will write down only k_1, given that the value of k_2 is immediately deducible. Including it in the calculations merely adds tedium. Thus the equations above become simply

$$P(k_1 = \mathit{updown}) = P(k_1 = \mathit{sideways}) = 1/2$$

30. Suppose we want to calculate the joint probability $P_1 = P(k_1 = \mathit{updown} \wedge \tau_1 = +1 \wedge \tau_2 = +1)$. One does it by using the definition of the conditional probability—the actual definition, not the John Stewart Bell definition:

$$\begin{aligned} P_1 &= P(k_1 = \mathit{updown} \wedge \tau_1 = +1 \wedge \tau_2 = +1) \\ &= P(k_1 = \mathit{updown})\, P(\tau_1 = +1 \wedge \tau_2 = +1 \,|\, k_1 = \mathit{updown}) \\ &= P(k_1 = \mathit{updown})\, P(\tau_1 = +1 \,|\, k_1 = \mathit{updown})\, P(\tau_2 = +1 \,|\, k_1 = \mathit{updown}) \\ &= P(k_1 = \mathit{updown})\, P(\tau_1 = +1 \,|\, k_1 = \mathit{updown})\, P(\tau_2 = +1 \,|\, k_2 = \mathit{sideways}) \\ &= \frac{1}{2} \sin^2(\phi_{01}) \cos^2(\phi_{02}) \end{aligned}$$

A person might notice I assumed

$$P(\tau_1 = +1 \wedge \tau_2 = +1 \,|\, k_1 = \mathit{updown}) = P(\tau_1 = +1 \,|\, k_1 = \mathit{updown})\, P(\tau_2 = +1 \,|\, k_1 = \mathit{updown})$$

without proof, but this was because I am old and tired and get senior discounts. I did not invoke "LOCAL CAUSALITY HAY SOCKS LOOPHOLES!" Seriously, though, the two **cray_ban** operate independently and that is the intuition here. This is different from what John Bell attempted, which was to construct an explicit causal chain (by abusing conditional probability notation), whack the audience with a stun weapon—blinding them—then impress upon them the illusion an explicit causal chain is the *only* form in which classical physics can be expressed.

If John Bell had been correct about that, then Johannes Kepler was not doing classical physics when he observed that planets moved in ellipses, nor was Isaac Newton when he formulated his Law of Universal Gravitation. But really that is beside the point, because thosed are *empirical laws*, not derived theories. As I said earlier, the SMOKE-&-MIRRORS crowd are actually distracting you from *this* fact: in the context at hand, "classical physics" means *any* mathematics that is not quantum mechanics, *if* employed to reach the same result as quantum mechanics. Their actual claim, *sotto voce*, is that no mathematics but quantum mechanics can get the job done.

It is a ludicrous claim. It would have been laughed out of the room so long ago that the Tortoise had not yet caught up with the Hare, had the claim been voiced out loud. It is so ridiculous a claim that it could not have been kept secret. Thus, indeed, it is not so much that the claim is kept *sotto voce* as that its believers do not, in fact, see that it is what they believe. Their cortexes are screaming and their hypothalamuses are pulsating. They *think* they are saying things that make sense. They spend too much time amidst their own psychosomatic barrages.

Part of the reason for me writing this program as *instructions* on how to write a program, rather than as merely a program for others to compile and run, is so SMOKE-&-MIRRORS CLUB members can sooth their throbbing brains by writing *their own programs*. So they can experience the truth firsthand, and as recreation rather than hard work. I encourage them to pick up computer and bow to play a soothing **crayton** lullaby, according to the sheet music herein, but each playing the music in their unique style.

31. By the previous calculation, and then by similar ones (though actually by symmetry considerations), a table can be constructed:

$$P(k_1 = \textit{updown} \wedge \tau_1 = +1 \wedge \tau_2 = +1) = \frac{1}{2}\sin^2(\phi_{01})\cos^2(\phi_{02})$$
$$P(k_1 = \textit{updown} \wedge \tau_1 = +1 \wedge \tau_2 = -1) = \frac{1}{2}\sin^2(\phi_{01})\sin^2(\phi_{02})$$
$$P(k_1 = \textit{updown} \wedge \tau_1 = -1 \wedge \tau_2 = +1) = \frac{1}{2}\cos^2(\phi_{01})\cos^2(\phi_{02})$$
$$P(k_1 = \textit{updown} \wedge \tau_1 = -1 \wedge \tau_2 = -1) = \frac{1}{2}\cos^2(\phi_{01})\sin^2(\phi_{02})$$
$$P(k_1 = \textit{sideways} \wedge \tau_1 = +1 \wedge \tau_2 = +1) = \frac{1}{2}\cos^2(\phi_{01})\sin^2(\phi_{02})$$
$$P(k_1 = \textit{sideways} \wedge \tau_1 = +1 \wedge \tau_2 = -1) = \frac{1}{2}\cos^2(\phi_{01})\cos^2(\phi_{02})$$
$$P(k_1 = \textit{sideways} \wedge \tau_1 = -1 \wedge \tau_2 = +1) = \frac{1}{2}\sin^2(\phi_{01})\sin^2(\phi_{02})$$
$$P(k_1 = \textit{sideways} \wedge \tau_1 = -1 \wedge \tau_2 = -1) = \frac{1}{2}\sin^2(\phi_{01})\cos^2(\phi_{02})$$

32. By adding the probabilities of mutually exclusive propositions in that table, one deduces

$$\begin{aligned} &P(\tau_1 = +1 \wedge \tau_2 = +1) \\ &\quad = P(\tau_1 = -1 \wedge \tau_2 = -1) \\ &\quad = \frac{1}{2}\sin^2(\phi_{01})\cos^2(\phi_{02}) + \frac{1}{2}\cos^2(\phi_{01})\sin^2(\phi_{02}) \end{aligned}$$

$$\begin{aligned} &P(\tau_1 = +1 \wedge \tau_2 = -1) \\ &\quad = P(\tau_1 = -1 \wedge \tau_2 = +1) \\ &\quad = \frac{1}{2}\sin^2(\phi_{01})\sin^2(\phi_{02}) + \frac{1}{2}\cos^2(\phi_{01})\cos^2(\phi_{02}) \end{aligned}$$

33. Now suppose we want to find an "expectation" $E'(\tau_1\tau_2)$ *not* as a function of a difference, such as $\phi_{01} - \phi_{02}$, but *instead* as a function of particular given values ϕ_{01} and ϕ_{02}. This, I believe, is a problem the SMOKE-&-MIRRORS CLUB has mistaken for the real one. But its solution will lead *so quickly* to the real answer (in terms of a difference between angles) that ... you have to see it to believe it. One wonders not so much *how* they missed the solution, but whether some of them *saw* it but dismissed it as unpublishable, because "LOCAL CAUSALITY HAY SOCKS LOOPHOLES!" That is, they knew if they submitted a paper they would be bombarded with psychic energy weapons and have their careers severely damaged.

34. To write this new "expectation" $E'(\tau_1\tau_2)$ (call it ρ') as an integral weighted by a pdf would be excessive. It can be written as a sum:

$$\begin{aligned}\rho' &= E'(\tau_1\tau_2)\\ &= (+1)(+1)P^{++} + (+1)(-1)P^{+-} + (-1)(+1)P^{-+} + (-1)(-1)P^{--}\\ &= P^{++} - P^{+-} - P^{-+} + P^{--}\end{aligned}$$

where

$$\begin{aligned}P^{++} &= P(\tau_1 = +1 \wedge \tau_2 = +1)\\ P^{+-} &= P(\tau_1 = +1 \wedge \tau_2 = -1)\\ P^{-+} &= P(\tau_1 = -1 \wedge \tau_2 = +1)\\ P^{--} &= P(\tau_1 = -1 \wedge \tau_2 = -1)\end{aligned}$$

Substituting the calculated expressions for the probabilities gives

$$\begin{aligned}\rho' &= -\{\cos^2(\phi_{01})\cos^2(\phi_{02}) - \cos^2(\phi_{01})\sin^2(\phi_{02}) - \sin^2(\phi_{01})\cos^2(\phi_{02}) + \sin^2(\phi_{01})\sin^2(\phi_{02})\}\\ &= -\{\cos^2(\phi_{01}) - \sin^2(\phi_{01})\}\{\cos^2(\phi_{02}) - \sin^2(\phi_{02})\}\\ &= -\cos(2\phi_{01})\cos(2\phi_{02})\end{aligned}$$

where the last step is by a double-angle identity found in reference books. This result is, I believe, what SMOKE-&-MIRRORS members commonly believe is the best classical physics can achieve.

35. This result has the wrong form, so it simply *cannot* be the correct solution! And, indeed, it gives incorrect results. If you plug in the angles $\phi_{01} = \pi/4$ and $\phi_{02} = \pi/8$, for instance, you will get zero instead of the correct value, $-1/\sqrt{2}$. But now, with this result that obviously, at a glance, cannot be correct, you can derive an "inequality" and win a Nobel Prize. I am pretty sure this is one route, at least, by which the so-called "CHSH inequality" can be derived. I do not wish to damage my cerebral cortex by looking into the matter more deeply. The "CHSH inequality" is complete garbage.

36. But suppose that, instead of publishing an "inequality" and winning a Nobel Prize, we consider only the special case $\phi_{02} = 0$. Then

$$\rho' = -\cos(2\phi_{01}) = -\cos\{2(\phi_{01} - \phi_{02})\}$$

and it *does* have the correct form. Yes, it is valid to include zero in an expression.

And now let us give the name $\Delta\phi$ to any angle whatsoever, and include zero in the expression once more, by adding $0 = \Delta\phi - \Delta\phi$ to $\phi_{01} - \phi_{02}$:

$$\rho' = -\cos(2\{(\phi_{01} + \Delta\phi) - (\phi_{02} + \Delta\phi)\})$$

And then let us call $\phi_{01} + \Delta\phi$ by the name ϕ_1, and $\phi_{02} + \Delta\phi$ by the name ϕ_2, and also (because it has the correct form) rename ρ' as simply ρ:

$$\begin{aligned}\rho &= -\cos\{2(\phi_1 - \phi_2)\} \\ &= -\{\cos^2(\phi_1 - \phi_2) - \sin^2(\phi_1 - \phi_2)\}\end{aligned}$$

Having done these things, we have derived, using only classical physics, the same correlation coefficient quantum mechanics gives. All we have done is take the *incomplete* solution SMOKE-&-MIRRORS members know how to derive, restrict it to a special case, and then discovered (by careful consideration of the number zero) that this restriction makes the solution complete! In so doing, we have shown that there is no entanglement, no non-locality, no quantum weirdness whatsoever. Einstein, Podolsky, and Rosen were correct in 1935. The 2022 Nobel Prize in Physics was awarded for research done so badly it ought to be regarded as *pseudoscience*. A perpetual motion machine, as I implied before, would be just as deserving of a prize.

This affair demonstrates that prizes in science are unethical. Who will volunteer to torpedo the 2022 Nobel Prize in Physics, even though this *must* be done, eventually? Besides, to give out a huge cash prize for what ought to be a humble profession of scientific method is *an insult*. But, of course, it is not, in fact, a humble profession of scientific method. Today's science is often a profession of career advancement through citation churning. Journals do not advocate for their scientific value, but their "impact factor." So perhaps a Nobel Prize is fitting, in this instance.

The number of paper retractions due is staggering. Nevertheless, expect instead professors occupying university administration offices, standing on the roofs with megaphones, shouting "LOCAL CAUSALITY HAY SOCKS LOOPHOLES!"

37. There is a simple interpretation for this classical derivation, an interpretation I worked into simulations slightly more complicated than the one this tutorial describes. Actually, those simulations existed many weeks before this derivation, so served as immediate evidence of the proof's validity. In the simulations, the equivalents of a **cray_ban** are constantly rotating on axles, in unison. This is as if $\Delta\phi$ were allowed to increase freely over time. Although the proportions of "who gets sent which way" change as $\Delta\phi$ changes, the correlation coefficient stays fixed with the relative angle. There is no entanglement, there is no non-locality, there is nothing weird whatsoever. Members of the SMOKE-&-MIRRORS CLUB were always deploying psychosomatic weaponry rather than presenting facts.

38. However, there is also a much deeper interpretation: *any* angle may be labeled zero, as long as it is *the same* angle on both **cray_ban** in the pair.

To make this so was the main goal of the pdf in my original proof, where I achieved the goal by making the probability density uniform with respect to one of the two angular settings. This approach might seem obvious to we who do not confuse probability with "randomness" or with physical substance. Nevertheless, the approach is overcomplicated. Instead, simply set the angle, once and for all time, to zero. Then introduce $0 = \Delta\phi - \Delta\phi$.

39. The Unlicense.

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <https://unlicense.org>

40. Index.

⟨ a series of n experimental events 11 ⟩ Used in section 21.
⟨ an experimental event 9 ⟩ Used in section 21.
⟨ arbitrary numbers between zero and one 2 ⟩ Cited in sections 5 and 8. Used in section 21.
⟨ correlation coefficient estimate function 19 ⟩ Used in section 21.
⟨ estimate of the correlation coefficient 18 ⟩ Used in section 19.
⟨ estimates of certain products 16 ⟩ Cited in section 17. Used in section 19.
⟨ estimates of the angle-difference functions 17 ⟩ Used in section 19.
⟨ frequencies of events 15 ⟩ Used in section 19.
⟨ printing out the correlation coefficient estimate 20 ⟩ Used in section 21.
⟨ the Law of Logodaedalus 8 ⟩ Used in section 21.
⟨ the **cray_ban** type 6 ⟩ Used in section 21.
⟨ the **crayton** source 5 ⟩ Used in section 21.
⟨ the **crayton** type 4 ⟩ Used in section 21.
⟨ the **series_data** type 10 ⟩ Used in section 21.