Qeios https://doi.org/10.32388/V11ZO2.2

Computational substantial violation of the CHSH with close approximation of the respective quantum values

Han Geurdes

April 25, 2023

Abstract If a clear and valid no-go for Einsteinian hidden parameters is real, it must in no way be possible to violate the CHSH with a local hidden variables based computer simulation. In the paper we show that with the use of a a proper density and modified Glauber-Sudarshan method, it is possible to violate the CHSH. The criterion value comes close to the quantum value and is $\gtrapprox 2.4$. The proof (POC) is presented with the use of an R computer program. The important snippets of the code are discussed and, for transparency, the complete code is presented in an appendix.

Keywords Bell theorem · proper density · local function computer program · violation CHSH

1 Introduction

It is well known that entanglement was for Einstein [1] a reason to claim that quantum mechanics is incomplete and/or inseparable [2]. It is also known from letters exchanged between Einstein and Schrödinger [3] that Einstein was dissatisfied with [1]. In [4] Einstein gives a simple and readable account of his worries about the completeness of quantum mechanics. In [5, chapters 5-7] a perhaps one sided (viz. [3]) but extensive account is given of the Bohr-Einstein debates. To the author, these are the basic papers and books. Obviously, it is impossible to give proper credit to the many scientists that wrote on the topic of entanglement. The author appologizes for not including them. The paper is absolutely and in no way an overview or general review

Han Geurdes
GDS Appl Math BV Netherlads
E-mail: han.geurdes@gmail.com

Acknowledgements The author wishes to thank A. Popper for his friendly support and R. Gill for raising a point about the density leading to this version.

of the EPR debate. It only asks the question if certain inequalities are *realy* limiting to Einstein's worries about quantum theory. Nevertheless, the author wishes to explicitly mention here the analysis of Einstein's work by Nordén [6]. Nordén pointed the author at the importance of the work of Sudarshan.

1.1 Bell correlation

In 1951, David Bohm wrote an interesting chapter in his book on quantum mechanics that enabled researchers in a later stage of the developments, to employ in experiment, spin-spin entanglement [7]. The starting-point of our analysis is the Bell formula [8] to explain spin-spin entanglement with the use of hidden variables λ. The correlation formula formalized the attempt to find the possible role of local hidden variables.

$$E(a,b) = \int_{\lambda\in\Lambda} \rho(\lambda)A(a,\lambda)B(b,\lambda)d\lambda \tag{1}$$

Given the set of values Λ, this correlation function is the hidden parameters, $\lambda \in \Lambda$, expression for a pardigmatic entanglement experiment [10]:

$$[A(a)] \leftarrow\sim\cdots\sim\leftarrow\sim [S] \sim\rightarrow\sim\cdots\sim\rightarrow [B(b)] \tag{2}$$

Here, $[S]$ represents the source of entangled pairs of photons. The wavy symbols represent the photons traveling from $[S]$ in opposite directions. The a and b represent the unit length parameter vectors of the measurement instruments $[A(a)]$ and $[B(b)]$ at a sufficiently large distance $d([A],[B]) \gg 0$. In the formula of Bell (1) the hidden parameters, λ, are allowed to influence the correlation $E(a,b)$ with the restriction

$$\begin{aligned} |A(a,\lambda)| &\leq 1 \\ |B(b,\lambda)| &\leq 1 \end{aligned} \tag{3}$$

The λ in the set Λ are distributed with probability density $\rho(\lambda)$. The probability density might refer to the distribution of hidden variables. We obviously have $\int_{\lambda\in\Lambda} \rho(\lambda)d\lambda = 1$.

1.2 CHSH

In our computer program we will look at CHSH (Clauser, Horne, Shimony and Holt) inequalities [11] such as the one below. For convenience we will provide the derivation of one such an inequality from Bell's formula (1). Let us look at four pairs of parameter vectors: $(a,b),(a,c),(d,b)$ and (d,c). The pairs of

settings can be employed in the experiment (2).

$$E(a,b)+E(a,c)=\int_{\lambda\in\Lambda}\rho(\lambda)\{A(a,\lambda)B(b,\lambda)+A(a,\lambda)B(c,\lambda)\}d\lambda+ \int_{\lambda\in\Lambda}\rho(\lambda)\{A(a,\lambda)B(b,\lambda)A(d,\lambda)B(c,\lambda)\}d\lambda- \int_{\lambda\in\Lambda}\rho(\lambda)\{A(a,\lambda)B(b,\lambda)A(d,\lambda)B(c,\lambda)\}d\lambda \quad (4)$$

Hence, we can rewrite

$$E(a,b)+E(a,c)=\int_{\lambda\in\Lambda}\rho(\lambda)A(a,\lambda)B(b,\lambda)\{1+A(d,\lambda)B(c,\lambda)\}d\lambda+ \int_{\lambda\in\Lambda}\rho(\lambda)A(a,\lambda)B(c,\lambda)\{1-A(d,\lambda)B(b,\lambda)\}d\lambda \quad (5)$$

With, looking at (3), $1+A(d,\lambda)B(c,\lambda)\geq 0$ and $1-A(d,\lambda)B(b,\lambda)\geq 0$. Therefore, it can be derived that

$$B\equiv E(a,b)+E(a,c)+E(d,b)-E(d,c)\leq 2 \quad (6)$$

Obviously there are other CHSH inequalities like e.g.

$$B'\equiv E(a,b)-E(a,c)+E(d,b)+E(d,c)\leq 2 \quad (7)$$

It is noted that similar CHSH inequalities can be derived in a similar manner as provided in the above. Now the claim of CHSH and the experiment (2) is that if $E(a,b), E(a,c), E(d,b)$ and $E(d,c)$ are determined, we will get (6) if the correlation is based on (1). If one blindly trust in the derivation presented for (6) plus the numerics of the experiment, then this makes both mathematically and experimental physics sense. The question is, is it untouchably true. As an example, we now know that quantum particles can tunnel through a finite potential. That doesn't make face value sense either.

2 Modified Glauber-Sudarshan representation

In an excellent textbook of quantum optics the Glauber - Sudarshan representation is described [9, page 264-266, section 14.2]. The method is related to the experiment of Aspect [10]. The basic ingredients of this experiment are represented in (2). We can rewrite Bell's formula as [9]

$$E(\theta_1,\theta_2)=N^{-1}\int f(\lambda)S_1(\lambda,\theta_1)S_2(\lambda,\theta_2)d\lambda \quad (8)$$

The (θ_1,θ_2) [9, page 263,fig 14.1] are equivalent to the (a,b) etc setting parameters employed in formula (1). In the previous formula (8), $N=\int f(\lambda)d\lambda$ Hence, $\rho(\lambda)=f(\lambda)N^{-1}$. The Glauber-Sudarshan representation tries to find an expression for the $S_1(\lambda,\theta_1)$ and $S_2(\lambda,\theta_2)$.

2.1 Modification

Basic to our local hidden variables computer program is a modification of the Glauber-Sudarshan formulae [9, page 266]. The modification concerns the adding of functions of the variables f and ϕ like

$$\begin{aligned} \gamma_+ &= \alpha_+ \cos\theta + \alpha_- \sin\theta + f\cos\phi \\ \gamma_- &= -\alpha_+ \sin\theta + \alpha_- \cos\theta + \sin\phi \end{aligned} \tag{9}$$

In the computer program we treated ϕ as a characteristic of the measurement instrument. Both measurement instruments share the same ϕ. The use of f is explained in the section dealing with the algorithm below. The addition of ϕ and f to the Glauber-Sudarshan formulae are, in a numerical mathematics sense, sufficient to violate the CHSH.

3 Hidden variables & densities

Furthermore, next to the ϕ and f, let us consider $\alpha_+^{(X)} \in \{-1, 1\}$ with $X = A$, or $X = B$, a hidden variable inside the system and let us take $\alpha_-^{(X)} = -\alpha_+^{(X)}$. We postulate $\alpha_+^{(A)}$ on the side of Alice and $\alpha_+^{(B)}$ on the side of Bob. Let us define,

$$\int \rho(\lambda) d\lambda \rightarrow \sum_{\alpha_+^{(A)} \in \{-1,1\}} \sum_{\alpha_+^{(B)} \in \{-1,1\}} \delta(\alpha_+^{(A)}; 0)\delta(\alpha_+^{(B)}; 0) = 1 \tag{10}$$

The $\delta(\alpha; 0)$ is a function that accepts a *first* input value, e.g. $\alpha = 1$, and produces the output unity. It then compares all other input to it and giving, here for $\alpha \neq 1$, the output $= 0$. Subsequently, the $\delta(\alpha_+^{(A)}; 0)$ and $\delta(\alpha^{(B)}; 0)$ are defined with $\delta(1; 1) = 1$ and $\delta(-1; 1) = 0$, together with $\delta(-1; -1) = 1$ and $\delta(1; -1) = 0$. This may look somewhat strange. Explanation follows below the compact formulae in (11). These are

$$\begin{aligned} \delta(\alpha; 0) = (\text{when } \alpha = 1 \text{ first input}) &\begin{cases} \delta(1; 1) = 1, & \alpha = 1 \\ \delta(-1; 1) = 0, & \alpha = -1 \end{cases} \\ \text{and} & \\ \delta(\alpha; 0) = (\text{when } \alpha = -1 \text{ first input}) &\begin{cases} \delta(-1; -1) = 1, & \alpha = -1 \\ \delta(1; -1) = 0, & \alpha = 1 \end{cases} \end{aligned} \tag{11}$$

When the first value of α in $\delta(\alpha; 0) \geq 0$, equals $+1$, then the delta is conditioned by $\alpha = 1$. Here, for $\alpha, \beta \in \{-1, 1\}$, it is then defined, $r(\alpha, \beta) = (\alpha + \beta)/2$ Hence, $\delta(1; 1)$ and its value is $\delta(1; 1) = \frac{|r(1,1)|}{r(1,1)r(1,1)} = \frac{1}{1\times 1} = \frac{1}{1} = 1$. When another value $\alpha = -1$ is "input", after conditioning on $\alpha = 1$, then $\delta(-1; 1) = \frac{|r(-1,1)|}{r(-1,-1)r(1,1)} = \frac{0}{(-1)\times(+1)} = 0$. Moreover, first $\alpha = -1$ then,

$\delta(-1;-1) = \frac{|-1|}{(-1)\times(-1)} = 1$ and $\delta(1;-1) = \frac{0}{(+1)\times(-1)} = 0$. Therefore, a summation (each α in $\{-1,1\}$ value once) over $\delta(\alpha;0)$ will run like, for first $\alpha = -1$ and subsequently $\alpha = 1$,

$$\sum_{\alpha\in\{-1,1\}} \delta(\alpha;0) = \delta(-1;-1) + \delta(1;-1) = 1 + 0 = 1 \tag{12}$$

Or, the other way around, first $\alpha = 1$ and subsequently $\alpha = -1$

$$\sum_{\alpha\in\{-1,1\}} \delta(\alpha;0) = \delta(1;1) + \delta(-1;1) = 1 + 0 = 1 \tag{13}$$

The $\delta(\alpha;0) \geq 0$, is like a subroutine returning values from a given input. The returning value for a single α input in $\delta(\alpha;0)$, is always unity. Per particle pair, each pair $\delta(\alpha_+^{(A)};0)$ and $\delta(\alpha_+^{(B)};0)$ starts afresh. The summation is non-directional; i.e. $\sum_{\alpha\in\{-1,1\}} f(\alpha) = f(-1) + f(1) = f(1) + f(-1)$, for proper general f.

3.1 Kolmogoroff

Let $E \equiv \{\alpha = 1\}$ and $\overline{E} \equiv \{\alpha = -1\}$. And $E, \overline{E} \in \Sigma \subset 2^{\Omega}$, with, $\Omega = \{-1, 1\}$ and $\Sigma = \{\{-1\}, \{1\}\}$. Then, $\alpha = 1$ is determining, $P(E) = 1 \geq 0$ and $P(\overline{E}) = 0 \geq 0$. The, $P(\Omega) = 1$ and $P(E \cup \overline{E}) = P(\Omega) = P(E) + P(\overline{E}) = 1$. Note $P(\Omega) = 1$ is independent of determining α. This can be explained when $\alpha = -1$ is determining. Then $P(E) = 0 \geq 0$ and $P(\overline{E}) = 1 \geq 0$ and again $P(E \cup \overline{E}) = P(\Omega) = P(E) + P(\overline{E}) = 1$. In addition, the $P(E \cap \overline{E}) = P(\emptyset) = 0$. This concurs that either $P(E) = \delta(1;-1) = 0$ for determining $\alpha = -1$ or $P(\overline{E}) = \delta(-1;1) = 0$ for determining $\alpha = 1$. Hence, $P(E \cap \overline{E}) = P(E)P(\overline{E}) = 0$, independent determining α. The Kolmogorovian axioms are fulfilled and $\delta(\alpha;0)$ can be the "density" that generates the Kolmogorovian probabilities.

3.1.1 Variation

Obviously, there will be criticism about the variability of density with the determining first $\alpha = \pm 1$. Nevertheless, with $\delta(\alpha_+^{(X)};0)$ and $X = A$, or $X = B$, the CHSH can *formally* be derived. See the expression (24) below. Moreover, the debate is not resolved with preferred metaphysical ideology. Either proof is delivered why $\delta(\alpha_+^{(X)};0)$ cannot exist in physics or it "goes". Such a proof is e.g. proving, independently of CHSH, that particles cannot influence the density in the wings of the experiment. Face value, although it smells like metaphysics ideology, it is not entirely impossible that such an influence may occur.

3.1.2 Expectation

In the case of $\delta(\alpha;0)$ we have ambiguity in expectation values. The three Kolmogorovian axioms in section-3.1 remain valid however. It is noted that when determined by $\alpha = 1$

$$E(\alpha) = \delta(1;1) - \delta(-1;1) = 1 - 0 = 1. \tag{14}$$

In addition, when determined by $\alpha = -1$

$$E(\alpha) = -\delta(-1;-1) + \delta(1;-1) = -1 + 0 = -1 \tag{15}$$

The general average, independent of determining α, is therefore undefined. In statistics the average expected value can be undefined however, although this mostly will happen with expectation of the random variable to infinity.

In the algorithm the state of $\alpha = \alpha_+^{(A)}$ determining $\delta(\alpha_+^{(A)};0)$ is randomly generated and has either (14) or (15) for expectation. Similar claim holds for $\alpha = \alpha_+^{(B)}$ and $\delta(\alpha_+^{(B)};0)$ towards Bob.

4 Algebra of the modified representation

The A or B indication is suppressed when considered possible in the discussion below. In [9, page 266] we may read that, generally, when $|\gamma_+|^2 + |\gamma_-|^2 = |\alpha_+|^2 + |\alpha_-|^2$, we have for S

$$S(\lambda,\theta) = \frac{|\gamma_+|^2 - |\gamma_-|^2}{|\gamma_+|^2 + |\gamma_-|^2} \tag{16}$$

Obviously $|S| \leq 1$. From the modified Glauber-Sudarshan equations in (9) we can learn that

$$\begin{aligned} |\gamma_+|^2 = |\alpha_+|^2\cos^2\theta + |\alpha_-|^2\sin^2\theta + f^2\cos^2\phi + \\ 2\alpha_+\alpha_-\cos\theta\sin\theta + \\ 2\left(\alpha_+\cos\theta + \alpha_-\sin\theta\right) f\cos\phi \end{aligned} \tag{17}$$

together with

$$\begin{aligned} |\gamma_-|^2 = |\alpha_+|^2\sin^2\theta + |\alpha_-|^2\cos^2\theta + \sin^2\phi + \\ -2\alpha_+\alpha_-\cos\theta\sin\theta + \\ -2\left(\alpha_+\sin\theta - \alpha_-\cos\theta\right)\sin\phi \end{aligned} \tag{18}$$

Looking at (17) and (18) it can be concluded that

$$\begin{aligned} |\gamma_+|^2 + |\gamma_-|^2 = |\alpha_+|^2 + |\alpha_-|^2 + \\ \left\{f^2\cos^2\phi + 2\left(\alpha_+\cos\theta + \alpha_-\sin\theta\right) f\cos\phi\right. \\ \left. -2\left(\alpha_+\sin\theta - \alpha_-\cos\theta\right)\sin\phi + \sin^2\phi\right\} \end{aligned} \tag{19}$$

In order to have $|\gamma_+|^2 + |\gamma_-|^2 = |\alpha_+|^2 + |\alpha_-|^2 = 2$, with $\alpha_+ \in \{-1, 1\}$ and $\alpha_- = -\alpha_+$, it is necessary to have

$$\begin{aligned} f^2 \cos^2\phi + 2\left(\alpha_+ \cos\theta + \alpha_- \sin\theta\right) f \cos\phi \\ -2\left(\alpha_+ \sin\theta - \alpha_- \cos\theta\right) \sin\phi + \sin^2\phi = 0 \end{aligned} \tag{20}$$

This expression can be reformulated as $f^2 + 2bf + c = 0$. Therefore, if we define,

$$c \equiv \tan^2\phi - 2\left(\frac{\alpha_+ \sin\theta - \alpha_- \cos\theta}{\cos\phi}\right)\tan\phi \tag{21}$$

then

$$2b \equiv 2\left(\frac{\alpha_+ \cos\theta + \alpha_- \sin\theta}{\cos\phi}\right) \tag{22}$$

If, $b^2 - c \geq 0$, the solution f is real and gives

$$f_{1,2} = -b \pm \sqrt{b^2 - c} \tag{23}$$

This latter equation is the nucleus of the computer program. The associated Bell correlation is the formula

$$\begin{aligned} E(\theta^{(A)}, \theta^{(B)}) = \sum_{\alpha_+^{(A)} \in \{-1,1\}} \sum_{\alpha_+^{(B)} \in \{-1,1\}} \delta(\alpha_+^{(A)}; 0)\delta(\alpha_+^{(B)}; 0) \\ \times \left\{ S^{(A)}(\theta^{(A)}, \alpha_+^{(A)}, \phi)\ S^{(B)}(\theta^{(B)}, \alpha_+^{(B)}, \phi) \right\} \end{aligned} \tag{24}$$

which is the discrete equivalent of (8). Note that *formally* a CHSH, viz. (6) or (7), can be derived from (24).

The $S^{(A)}$ and $S^{(B)}$ of equation (24) are computed according to (16). The value of ϕ is a fixed parameter in the algorithm and f is computed from (20). The ϕ represents a chracteristic of the two identical-in-construction measurement instruments A and B. The $f^{(A)} = f(\theta^{(A)}, \alpha_+^{(A)}, \phi)$ and $f^{(B)} = f(\theta^{(B)}, \alpha_+^{(B)}, \phi)$ are determined from (23) and densities $\delta(\alpha_+^{(A)}; 0)$ and $\delta(\alpha_+^{(B)}; 0)$. In turn the density of e.g. $\alpha = \alpha_+^{(A)}$ is determined by the particle flying towards Alice. Similar for $\alpha = \alpha_+^{(B)}$ towards Bob. The verification of the steps in (5) formally leading to CHSH can be easily performed with (24).

5 The computer algorithm

In this section, snippets of the R-code are presented and described. The snippets are the building blocks that represent the mathematics from the previous section. The complete code is for reference presented in an appendix. Let us begin with noting that the format of the densities $\delta(\alpha_+^{(A)}; 0)$ and $\delta(\alpha_+^{(B)}; 0)$ do not need to be incorporated. Each $\alpha^{(X)+}$, with $x \in \{A, B\}$, is associated to $\delta = 1$. The snippet is something like

```
delta<-function(alpha, blFirst){
  if(blFirst){
    y<-1
  }else{
    y<-0
  }
  if(blFirst){
    blFirst<-FALSE
  }
  return(c(y,blFirst))
}
```

The blFirst is at the start of each particle pair run set to TRUE and then, after call of the delta function, immediately to FALSE.

Subsequently, defining the characteristic parameter ϕ as

$$\phi = 220.14 \times \left(\frac{\pi}{180}\right) \tag{25}$$

This value is present in both separate functions in the code representing the measurements instruments A and B (2). The numerical values are given in degrees and converted to radians in the algorithm. Furthermore, we found out via trial and error that the following $\theta^{(A(B))}$ can be used to force a CHSH violation with local means.

$$\begin{aligned} \theta^{(A)} &\in \{97.39957, 113.48717\} \times \left(\frac{\pi}{180}\right) \\ \theta^{(B)} &\in \{-82.32930, -26.37997\} \times \left(\frac{\pi}{180}\right) \end{aligned} \tag{26}$$

Looking at the expression for quantum correlation in [9, page 266, equation (14.26)] it is found that the quantum correlation $E^{(qm)}(\theta^{(A)}, \theta^{(B)}) = \cos\left[2(\theta^{(A)} - \theta^{(B)})\right]$ gives a violation for a particular Bell inequality with the $\theta^{(A(B))}$ from (26). It is:

$$\begin{aligned} &E^{(qm)}(\theta_1^{(A)}, \theta_1^{(B)}) - E^{(qm)}(\theta_1^{(A)}, \theta_2^{(B)}) \\ &+E^{(qm)}(\theta_2^{(A)}, \theta_1^{(B)}) + E^{(qm)}(\theta_2^{(A)}, \theta_2^{(B)}) \approx 2.402191 \end{aligned} \tag{27}$$

5.1 Crucial response functions

With this information let us look at the snippets of code that reflect the in-instrument processes. Let us start with the A(lice) function. In the code this sigmaA is the foundation for $\alpha_+^{(A)}$.

```
  if(thet==(113.48717*pi/180)){
    sigmaA<-(-1)
  }else{
    if(mRun%%2==0){
```

```
      sigmaA<-sign(runif(1)-0.5)
    }else{
      sigmaA<-1
    }
  }
  alpha[1]<-sigmaA
  alpha[2]<--alpha[1]
```

The 2-dim array alpha represents here alpha[1]= $\alpha_+^{(A)}$ and alpha[2]= $\alpha_-^{(A)}$. When a "wave" (2) enters the instrument the equivalent of the response is such as provided above. The runif(1) randomization is the cause that there are a couple of runs necessary to find a Bell CHSH violation. As a weak attempt at theory we tried to have only at even trials with thet=97.39957*pi/180 a random ± 1 selection. On the Bob side we have sigmaB.

```
if(thet==(-82.32930*pi/180)){
  sigmaB<-(-1)
}else{
  if(mRun%%2==0){
    sigmaB<-sign(runif(1)-0.5)
  }else{
    sigmaB<--1
  }
}
alpha[1]<-sigmaB
alpha[2]<--alpha[1]
```

The weak form of theory introduced here is on the B side for thet=-26.37997*pi/180 to have at even trials a random ± 1 selection. In both cases runif(1) is the function responsible for random number $0 < r < 1$ draws. The sign gives a -1 if that number is less than 0.5 and 1 when greater or equal than 0.5. Obviously, this cannot be the whole physics story. Hence, the number of runs. Nevertheless the claim is that with sufficient number of runs with, each, 100 pairs of "photons", the two snippets will warrant a substantial violation close to the quantum CHSH-criterium value (27). This must be completely impossible if the CHSH is really waterproof. The two snippets are computed without computational nonlocality.

5.2 Further structure: the S

Obviously there will be a sceptical look at our claim of the previous subsection. Here we will show that our $S^{(A(B))}$ functions are identical to the expression in (16). Let us look at the A(lice) side. We have the function

```
...
  n<-1
...
```

```
  phi<-220.14*(pi/180)
  while(n<2){
    #
    gamma<-fGamAlf(alpha,thet,phi)
    if(gamma%*%gamma >0 ){
      n<-n+1
    }else{
      alpha[1]<-alpha[2]
      alpha[2]<--alpha[1]
    }
    S1<-(gamma[1]**2)-(gamma[2]**2)
    if(gamma%*%gamma >0 ){
      S1<-S1/((gamma[1]**2)+(gamma[2]**2))
    }
  }
  #sumB<-0.4436177
  sumB<-0.50
  if(thet==(113.48717*pi/180)){
    S1<-S1-(0.03/sumB)
  }else{
    S1<-S1+(0.03/sumB)
  }
  return(S1/1.0124)
}
```

as part of the A side evaluation. Hopefully the reader will understand the trial and error aspect of finding the proper coefficients. The introduced fine-tuning is to take sumB (a variable only available in A btw) 0.5 and the correction factor 1.0124 in the fine-tuned output.

The B side is similar. But the fine tuning differs. We have:

```
...
  phi<-220.14*(pi/180)
  xi <- 0.4901
  sumA <- 0.968244105
...
  while(n<2){
    #
    gamma<-fGamAlf(alpha,thet,phi)
    if(gamma%*%gamma >0 ){
      n<-n+1
    }else{
      alpha[1]<-alpha[2]
      alpha[2]<--alpha[1]
    }
    S1<-(gamma[1]**2)-(gamma[2]**2)
    if(gamma%*%gamma >0 ){
```

```
      S1<-S1/((gamma[1]**2)+(gamma[2]**2))
      if(thet==(-82.32930*pi/180)){
        S2<-S1
      }else{
        S2<-S1-(xi/sumA)
      }
    }
  }
  return(S2)
```

The numbers xi and sumA are derived from trial and error experiments. It must be noted that they only are partially successful for approximation of the quantum correlation values on the B side.

We will deal with the function fGamAlf(alpha,thet,phi) later. It is based on (9) and (23).

In this part of the code it is clear that when particular gamma[1]= $\gamma_+^{(A)}$ and gamma[2]= $\gamma_-^{(A)}$ are computed according to (9) and (23), using (25), then the $S1 = S^{(A)}$ in the program has $|S1| \leq 1$. We start with n equal 1. Only if the gamma array is unequal to zero in length, do we compute $S1$. If a zero length gamma array is produced given the particular $\theta^{(A)}$ the alpha[1] and alpha[2] are interchanged. Because n is then still 1, the call of fGamAlf(alpha,thet,phi) is repeated but with the interchanged alpha. This leads in all cases to a nonzero length gamma array and therefore n becomes 2 and the $S1$ computation is completed. We note that in both the A and B case the ϕ is equal to the value in (25).

According to the statements on [9, page 264-267] our claim at the end of the previous subsection 5.1 would be impossible. It is not. In this section it is crystal clear that $|S1| \leq 1$.

5.3 The fGamAlf(alpha,thet,phi) function

Here we will show that in the computation the function fGamAlf(alpha,thet,phi) follows the requirements in the equations (9) and (23). In the program two separated functions of similar form are employed in the function for A and for B. There is absolutely no exchange of information. And we add that the latter is not needed either. On the A side we have

```
fGamAlf<-function(alpha,thet,phi){
  gamma<-array(0,2)
  c<-(tan(phi))**2
  c1<-alpha[1]*(sin(thet)/cos(phi))
  c1<-c1-(alpha[2]*(cos(thet)/cos(phi)))
  c1<-c1*tan(phi)
  c<-c-(2*c1)
  f1<-alpha[1]*cos(thet)/cos(phi)
  f2<-alpha[2]*sin(thet)/cos(phi)
```

```
b<-f1+f2
if ((b**2)-c > 0){
  s<--1
  t<-1
  if(t==1){
    f<--b+(s*sqrt((b**2)-c))
  }
  gamma[1]<-(alpha[1]*cos(thet))+(alpha[2]*sin(thet))+(f*cos(phi))
  gamma[2]<-(-alpha[1]*sin(thet))+(alpha[2]*cos(thet))+sin(phi)
}else{
  gamma[1]<-0
  gamma[2]<-0
}
return(gamma)
```

On the B side this is similar but independent & isolated from the parameters of the A side.

Let us walk through the code presented here. After the initialisation of the 2-dim gamma array the b and c from(21) are computed. The test $(b**2)-c > 0$ is to ascertain that the quadratic form in f from (20) has a real solution given in (23). The situation is such that if the condition $(b**2)-c > 0$ is not met, then, a zero length gamma array is send back via the return command. This is processed such as decribed in section 5.2. If the condition $(b**2)-c > 0$ is met, then, the f can be computed according to the famous abc formula. We have $f^2 + 2bf + c = 0$ which solution, $f = -\frac{(2b)}{2} \pm \frac{1}{2}\sqrt{(2b)^2 - 4c}$, is represented in (23). We have decided to take the + branch and therewith compute gamma[1] and gamma[2]. In the program code presented in the appendix the fCHSHA can be implemented in the $[A]$ of (2) and the fCHSHB can be implemented in $[B]$ of (2). The main (start) routine is situated in the $[S]$.

5.4 Random trials

Our computer program uses a one-step randomisation of the setting pairs $(\theta_k^{(A)}, \theta_m^{(B)})$, with $k, m \in \{1, 2\}$ but we make sure that the number of trial is a fourfold. The snippet of code to generate random pairs is given below.

```
nMax<-100
ALICE1<-c(0,0,2,2)
ALICE<-array(ALICE1,nMax)
ALICE<-(ALICE+2)/2
...
BOB1<-c(0,2,0,2)
BOB<-array(BOB1,nMax)
BOB<-(BOB+2)/2
...
sTrial<-sample(seq(1,length(BOB)))
```

```
...
  for(m in sTrial){
    thetaA<-setA[ALICE[m]]
    thetaB<-setB[BOB[m]]
    sA[m]<- fCHSHA(alpha,thetaA)
    sB[m]<- fCHSHB(alpha,thetaB)
  }
```

A four-tuple of settings is based on the elementary ALICE2=1,1,2,2 and BOB2=1,2,1,2. This is expanded to length = nMax the ALICE2 in array ALICE. And similarly for the array BOB. Subsequently the order of the presentation of both arrays is randomly permuted in the array sTrial. This gives sufficiently random pairings of settings for A and for B. In the for next loop, for(m in sTrial), the permutations are employed. The fCHSHA(alpha,thetaA) and fCHSHB(alpha,thetaB) are described in the previous sections 5.3, 5.2 and 5.1.

The computation of the E is as follows:

```
...
  norM=nMax/4
  for(m in sTrial){
    Eab[ALICE[m],BOB[m]]<-Eab[ALICE[m],BOB[m]]+(sA[m]*sB[m]/norM)
  }
  ...
```

The value norM=nMax/4 can be explained with the fact that we have 4 E elements in the computation of one B' from (7). So in fact for e.g. 100 pairs we have 25 B' computations. In the program we also demonstrate that

`sum(abs(sA)<1)=100`

and

`sum(abs(sB)<1)=100,`

meaning all values of S are $-1 \leq S \leq 1$ viz (16).

6 Result, conclusion & discussion

The claim is here that, with a local model, it is *possible* to substantially violate the CHSH inequality and come pretty close to the B' of (7) value of quantum theory. Below we report an example of a result of our computations with the algorithm. Its output is verbatim:

```
          [,1]       [,2]
[1,] 0.9546458 -0.3818026
[2,] 0.8714130  0.1616421
[1] 2.369504
[1] 100
```

```
[1] 1469
          [,1]       [,2]
[1,] 0.9999552 -0.3817306
[2,] 0.8514255  0.1690791
[1] 2.40219
[1] 100
[1] 100
```

The first matrix

$$E = \begin{pmatrix} 0.9546458 & -0.3818026 \\ 0.8714130 & 0.161642 \end{pmatrix} \tag{28}$$

is based on the computations from subsection 5.4 which refers back to the expectation in equation (24) using the modified Glauber-Sudarshan representation. The number 2.369504 represents the B' as defined in (7) applied to the E matrix from the modified Glauber-Sudarshan representation (28). The loop is such that only index number 3, i.e. (7) is outputted. The number 3 denotes that B' CHSH from (2) is violated. The output e.g. 1 instead of 3 is an indication of a violation of B in (4). In the final output the printing of the type, i.e. jout, is suppressed. It can be observed that the quantum B does not violate (6) under this conditions.

So we may note that, with Einstein local hidden parameters, the CHSH can be violated size: $B' = 2.369504$ in the output of the algorithm. The first number 100 in the output refers to nMax, the amount of "photon" pairs inspected in this particular run. Obviously this is a somewhat small number of photon pairs. However, it is the principle violation that counts. The selection of nMax=100 is simply restricted by the limited computational power of the machine. The number 1469 refers to the number of runs (in a sequence from 1 to max 1×10^5) to compute a violating B'. Each run holds 100 pairs. So there were $1469 - 1$ trials of each 100 pairs before the violating result $B' = 2.369504$ was found. The reader can experiment with the code to see that lower numbers of runs are also possible to produce a violation. Anyway, when CHSH is waterproof this type of violation must *never* occur in this size.

In the 1469-th run the A and B snippets, in particular the random parts in the ifs of section 5.1 are able to produce violating S values that generate the E in (28).

We note again that if the CHSH was really waterproof, the number of trials would be extremely large and the violation B' would be close to 2.00. Therefore we believe that (28) represents a genuine violation and contest that it *absolutely* has to be larger than $B_{min} = 1 + \sqrt{2}$ under the proper $(a, b), (a, c), (d, b)$ and (d, c) conditions. Moreover, please note that Bell's formula give rise to only $B_{min} = 2$.

The second matrix in the verbatim output is the matrix of quantum values (27).

$$E^{qm} = \begin{pmatrix} 0.9999552 & -0.3817306 \\ 0.8514255 & 0.1690791 \end{pmatrix} \tag{29}$$

The B' derived from this matrix in (29) is $B' = 2.40219$. Therefore we may note that the hidden parameter B' and the quantum B' are close. We also see from comparing (28) with (29) that the algorithm does not exactly reproduce the quantum values. This cannot be expected either. It must be noted please that because in that case the critical reader would require a complete theory whereas this reader himself would only have Bell's statistics in his hands. Further research into a proper theory might be a possibility to replace the runif(1) elements.

Nevertheless the conclusion is that the CHSH *can* be *violated* with local parameter principles. This concurs with e.g. [12], [13] and [14]. It must be stressed that it does not matter how long it takes with randomization to substantially violate and it also is quite unimportant that there is a random table in a computer. If CHSH is waterproof this kind of violations are impossible. Apparently they are possible after all. The employed densities, $\delta(\alpha_+^{(A)};0)$ and $\delta(\alpha_+^{(B)};0)$ allow a formal derivation of CHSH. The densities are Kolmogorovian and determined by the particles that fly toeards Alice or Bob. Of course, a host of objections will be encountered but if those objections remain metaphysic idealizations they are worthless.

For convenience of the reader the complete R program is included in an appendix. It must be noted that this violation can also be obtained in VBA for excel. This to avoid all kinds of needless debate about the nature of R and that the reviewer does not understand R. All these things and more already happened more than once. If the reader wants the VBA program it can be mailed asap.

Another, but related, matter is that despite the fact that the P function of the Glauber - Sudarshan representation is not everywhere positive in entanglement, this feature *not* affects the validity of the computations represented in the Appendix and explained in the above section. In this paper we only employed the expression for $\gamma_\pm^{(A)}$ and for $\gamma_\pm^{(B)}$ and modified it, (9) , with $(f^{(A)},\phi^{(A)})$ and $(f^{(B)},\phi^{(B)})$. The density here employed in the computations is positive definite. If a reader disagrees then this reader *must* point where in the formulae and/or the computations, negative probability or probability density occurs. Moreover, if a reader disagrees with the format of the densities, then the reader must show physics and mathematics proof why such a density is not possible.

With the power of argument, we reject that CHSH type of inequalities can not be violated with local hidden parameter computations. The CHSH can be formally derived from equation (24) and the definition of the densities in section-3. The key concept is that functions fCHSHA and fCHSHB are completely encapsulated, hence local in the sense of Einstein [4]. The quantum values of the respective correlations were reasonably well approximated. In order to sustain the conclusion, it is absolutely *not* necessary to do this for every possible setting. The 100 photon pair computation that violates CHSH in a local manner *and* closely approximates the quantum correlation makes, like [12], a reasonable case against the CHSH associated claim that the experimen-

tal results such as [10] rule out Einstein locality in physical reality. Of course it is not a final proof that there indeed are extra local hidden parameters such as Einstein preferred viz. [2], [3] and [4].

Declarations

The author has no conflict of interest. The work was not funded.
All data generated or analysed during this study are included in this published article.

References

1. A. Einstein, B., Podolsky, and N. Rosen, "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?," Phys. Rev. 47, 777, (1935).
2. D. Howard, Einstein on locality and separability, Stud,Hist.Phil. Sci. 16(3),171 (1985).
3. D. Howard, "Nicht sein kann was nicht sein darf, on the pre-history of EPR, 1909-1935", in: Sixty-two years of uncertainty, ed: A. Miller,61-111, Plenum press, NY, (1990).
4. A. Einstein, "Quanten-Mechanik und Wirklichkeit", Dialectica, 2, 320-324, (1948).
5. M. Jammer, "The philosophy of quantum mechanics", John Wiley & Sons Inc, Bar-Ilan university and CUNY, (1974).
6. B. Nordén, "Quantum entanglement: facts and fiction-how wrong was Einstein after all?", Q. Rev. Biophys., 49, e17, (2016)."
7. D. Bohm, "The paradox of Einstein, Rosen and Podolsky", In; Quantum theory and measurement, ed: J.A. Wheeler and W.H. Zurek, 356-368, Princeton University Press, New Jersey, (1983).
8. J.S. Bell, "On the Einstein Podolsky Rosen paradox," Physics, 1, 195, (1964).
9. D.F. Walls and G.J. Milburn, "Quantum Optics", Lectures on Quantum Optics, University Waikato & University Auckland, Auckland, New Zealand, (1994).
10. A. Aspect, P.Grangier and G.Roger, "Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment", Phys.Rev.Lett. 49, 91, (1982).
11. J.F. Clauser, M.A. Horne, A. Shimony & R.A. Holt, "Proposed experiment to test local hidden-variable theories", Phys. Rev. Lett. 23(15), 880-884, (1969).
12. H. Geurdes, "A probability loophole in the CHSH", Results in Physics, 4, 81, (2014).
13. H. Geurdes, K. Nagata, T. Nakamura & A. Farouk, "A note on the possibility of incomplete theory", arxiv 1704.00005, (2017).
14. K. Nagata, T. Nakamura and H. Geurdes, "Incompleteness in the Bell theorem using non-contextual local realistic model", Int. J. Theor. Phys. doi:10.1007/s10773-019-04312-3 in the press, (2019).

Appendix

The complete computer program is presented to enable the reader to check the locality claim. R-studio and R-software are free to obtain from the internet.

```
#CHSH
randomset<-function(){
  st<-Sys.time()
  nst<-as.integer(substring(st,18,30))
  print(nst)
```

```
  for(k in seq(1,(nst+1))){
    r<-runif(1)
  }
}
##############
##ALICE
##############
fCHSHA<-function(alpha,thet,mRun){
  fGamAlf<-function(alpha,thet,phi,mRun){
    gamma<-array(0,2)
    c<-(tan(phi))**2
    c1<-alpha[1]*(sin(thet)/cos(phi))
    c1<-c1-(alpha[2]*(cos(thet)/cos(phi)))
    c1<-c1*tan(phi)
    c<-c-(2*c1)
    f1<-alpha[1]*cos(thet)/cos(phi)
    f2<-alpha[2]*sin(thet)/cos(phi)
    b<-f1+f2
    if ((b**2)-c > 0){
      s<--1
      t<-1
      if(t==1){
        f<--b+(s*sqrt((b**2)-c))
      }
      gamma[1]<-(alpha[1]*cos(thet))+(alpha[2]*sin(thet))+(f*cos(phi))
      gamma[2]<-(-alpha[1]*sin(thet))+(alpha[2]*cos(thet))+sin(phi)
    }else{
      gamma[1]<-0
      gamma[2]<-0
    }
    return(gamma)
  }
  #
  n<-1
  #
  if(thet==(113.48717*pi/180)){
    sigmaA<-(-1)
  }else{
    if(mRun%%2==0){
       sigmaA<-sign(runif(1)-0.5)
    }else{
      sigmaA<-1
    }
  }
  alpha[1]<-sigmaA
  alpha[2]<--alpha[1]
```

```
  phi<-220.14*(pi/180)
  while(n<2){
    #
    gamma<-fGamAlf(alpha,thet,phi)
    if(gamma%*%gamma >0 ){
      n<-n+1
    }else{
      alpha[1]<-alpha[2]
      alpha[2]<--alpha[1]
    }
    S1<-(gamma[1]**2)-(gamma[2]**2)
    if(gamma%*%gamma >0 ){
      S1<-S1/((gamma[1]**2)+(gamma[2]**2))
    }
  }
  #sumB<-0.4436177
  sumB<-0.50
  if(thet==(113.48717*pi/180)){
    S1<-S1-(0.03/sumB)
  }else{
    S1<-S1+(0.03/sumB)
  }
  return(S1/1.0124)
}
###############
##BOB
###############
fCHSHB<-function(alpha,thet,mRun){
  fGamAlf<-function(alpha,thet,phi){
    gamma<-array(0,2)
    c<-(tan(phi))**2
    c1<-alpha[1]*(sin(thet)/cos(phi))
    c1<-c1-(alpha[2]*(cos(thet)/cos(phi)))
    c1<-c1*tan(phi)
    c<-c-(2*c1)
    f1<-alpha[1]*cos(thet)/cos(phi)
    f2<-alpha[2]*sin(thet)/cos(phi)
    b<-f1+f2
    if ((b**2)-c > 0){
      s<-1
      t<-1
      if(t==1){
        f<--b+(s*sqrt((b**2)-c))
      }
      gamma[1]<-(alpha[1]*cos(thet))+(alpha[2]*sin(thet))+(f*cos(phi))
      gamma[2]<-(-alpha[1]*sin(thet))+(alpha[2]*cos(thet))+sin(phi)
```

```
    }else{
      gamma[1]<-0
      gamma[2]<-0
    }
    return(gamma)
  }
  #
  n<-1
  #
  phi<-220.14*(pi/180)
  xi <- 0.4901
  sumA <- 0.968244105
  #
  if(thet==(-82.32930*pi/180)){
    sigmaB<-(-1)
  }else{
    if(mRun%%2==0){
      sigmaB<-sign(runif(1)-0.5)
    }else{
      sigmaB<--1
    }
  }
  alpha[1]<-sigmaB
  alpha[2]<--alpha[1]
  while(n<2){
    #
    gamma<-fGamAlf(alpha,thet,phi)
    if(gamma%*%gamma >0 ){
      n<-n+1
    }else{
      alpha[1]<-alpha[2]
      alpha[2]<--alpha[1]
    }
    S1<-(gamma[1]**2)-(gamma[2]**2)
    if(gamma%*%gamma >0 ){
      S1<-S1/((gamma[1]**2)+(gamma[2]**2))
      if(thet==(-82.32930*pi/180)){
        S2<-S1
      }else{
        S2<-S1-(xi/sumA)
      }
    }
  }
  return(S2)
}
#####
```

```
#main
#####
randomset()
#
nMax<-100
#ALICE<-sign(runif(nMax)-0.5)+1
ALICE1<-c(0,0,2,2)
ALICE<-array(ALICE1,nMax)
ALICE<-(ALICE+2)/2
sA<-ALICE
#BOB<-sign(runif(nMax)-0.5)+1
BOB1<-c(0,2,0,2)
BOB<-array(BOB1,nMax)
BOB<-(BOB+2)/2
#
sB<-BOB
setA<-c(97.39957,113.48717)*(pi/180)
setB<-c(-82.32930,-26.37997)*(pi/180)
Bout<-0
nTel<-0
nTelMax<-1e5
uLim<-2.33
jout<-0
#
sTrial<-sample(seq(1,length(BOB)))
blCtd<-Bout<uLim
while(blCtd==TRUE){
  Eab<-matrix(0,2,2)
  #
  nTel<-nTel+1
  for(m in sTrial){
    #print(n)
    alpha<-array(0,2)
    thetaA<-setA[ALICE[m]]
    thetaB<-setB[BOB[m]]
    sA[m]<- fCHSHA(alpha,thetaA,m)
    sB[m]<- fCHSHB(alpha,thetaB,m)
  }
  norM=nMax/4
  for(m in sTrial){
    Eab[ALICE[m],BOB[m]]<-Eab[ALICE[m],BOB[m]]+(sA[m]*sB[m]/norM)
  }
  B<-array(0,4)
  #
  B[1]<-Eab[1,1]+Eab[1,2]+Eab[2,1]-Eab[2,2]
  B[2]<-Eab[1,1]+Eab[1,2]-Eab[2,1]+Eab[2,2]
```

```
  B[3]<-Eab[1,1]-Eab[1,2]+Eab[2,1]+Eab[2,2]
  B[4]<-(-Eab[1,1])+Eab[1,2]+Eab[2,1]+Eab[2,2]
  blTST<-(B[1]>uLim)|(B[2]>uLim)|(B[3]>uLim)|(B[4]>uLim)
  #
  if(nTel>nTelMax){
    print(Eab)
    Bout<-99999
  }else{
    Bout<-max(B)
    if(blTST==TRUE){
      for (j in 1:4){
        if (B[j]==Bout){
          jout<-j
          }
        if(jout==3){
          blCtd<-FALSE
        }else{
          blCtd<-TRUE
        }
      }
    }
    if (blTST==TRUE){
      if(jout==3){
         print(Eab)
      }
    }
#
  }
  if (nTel%%100 ==0){
  print(Bout)
  }
  if (Bout==99999){
      blCtd<-FALSE
  }
}
print(Bout)
print(jout)
print(nMax)
print(nTel)
#
Eqm<-matrix(0,2,2)
Eqm[1,1]=cos(2*(setA[1]-setB[1]))
Eqm[1,2]=cos(2*(setA[1]-setB[2]))
Eqm[2,1]=cos(2*(setA[2]-setB[1]))
Eqm[2,2]=cos(2*(setA[2]-setB[2]))
if(jout==3){
```

```
   Bqm=Eqm[1,1]-Eqm[1,2]+Eqm[2,1]+Eqm[2,2]
}else{
   Bqm=Eqm[1,1]+Eqm[1,2]+Eqm[2,1]-Eqm[2,2]
}
print(Eqm)
print(Bqm)
print(sum(abs(sA)<1))
print(sum(abs(sB)<1))
#stop("end")
```