Prepared for submission.

Computational violation of the CHSH with a local model.

Han Geurdes,[a]

[a] *Institution,*
geurdes data science, C. vd Lijnstraat 164 2593 NN Den Haag, Netherlands

E-mail: han.geurdes@gmail.com

Abstract: In this paper the design and coding of a local hidden variables model is presented that violates the $|CHSH| \leq 2$ inequality. Numerically we have CHSH $\approx 1+\sqrt{2}$.

Contents

1 Introduction.

The CHSH inequality is an important element in the discussion of the existence or nonexistence of additional local hidden parameters, i.e. the completeness of quantum mechanics, [1] for entangled particle pairs. The CHSH inequality [3] is derived from Bells formula for the correlation [2], $E(a, b)$, between distant spin measurements with setting parameters a and b. Generally, Bell wrote

$$E(a, b) = \int d\lambda \rho_\lambda A_\lambda(a) B_\lambda(b) \tag{1.1}$$

In equation (1.1) we can identify the probability density of the hidden variables λ, as $\rho_\lambda \geq 0$. We have $\int d\lambda \rho_\lambda = 1$. In a local model, λ are introduced to explain the entanglement (correlation) and must have a local effect. This can be accomplished if e.g. a λ_1 is assigned to the A measurement instrument and λ_2 to the B instrument. Furthermore, the measurement functions $A_\lambda(a)$ and $B_\lambda(b)$ both project in $\{-1, 1\}$ to represent binairy spin variables (e.g. up=1, down=-1). The a and b represent unit parameter vectors in $\mathbb{R}^3$. The CHSH inequality is based on the following expression,

$$S = E(1, 1) - E(1, 2) - E(2, 1) - E(2, 2) \tag{1.2}$$

The CHSH inequality $|S| \leq 2$ was derived therof [3].

2 Preliminaries in the design

Commonly it is believed that a computer violation of the CHSH inequality $|S| \leq 2$, see (1.2), with a local model is not possible. The program must mimic an important experiment in the test of locality [4]. In this experiment strict locality conditions were closely approximated and a violation $|S| > 2$ was observed for violating setting combinations of a and b with a quantum correlation $a \cdot b$. In [5], however, the present author already showed that there is a nonzero probability that a local hidden model may violate the CHSH. Objections were raised in [6] but were answered in [7]. The present paper completes the rejection of what has been claimed in [6].

In the present paper, a local model is presented that can be implemented in a simple computer program and leads to $S \approx 1 + \sqrt{2}$ for the following violating settings. On the A side Alice has $1 = \frac{1}{\sqrt{2}}(1, 0, 1)$ and $2 = (\frac{-1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2})$ at her disposal. On the B side, Bob has $1 = (1, 0, 0)$ and $2 = (0, 0, -1)$. For the ease of the argument we inspect, $E(a, b) = a \cdot b$. A simple computation then shows that for a quantum outcome we would see $E(1, 1) = 1/\sqrt{2}$, $E(1, 2) = -1/\sqrt{2}$ while $E(2, 1) = -1/2$ and $E(2, 2) = -1/2$. Hence, looking at (1.2), $S = 1 + \sqrt{2} > 2$ is expected in an experiment. The setting parameters a and b are given a value when the A- and B-wing particle leaves the source. In flight we allow B (Bob) to change his setting. This surpasses discussions about design time information.

Needless to say that infromation hiding between Alice and Bob is the algorithmic realization of strict locality. Furthermore, in the computer simulation A doesn't know anything about B and vice versa. All computations are "encapsulated" i.e. local, despite they occur in a single loop (Appendix A).

3 Design of the algorithm based on a local model

3.1 Random sources

In the first place let us assume random sources to represent random selection of setting. We look at the randomness from the point of view of creating an algorithm. If there are N trials, i.e particle pairs, in the experiment then e.g. two independent random sources can be seen as two arrays with rank numbers from 1 to N. If $\mathcal{N}_N = (1, 2, 3, ..., N)$, then we define three random source arrays

$$\begin{aligned} \underline{\mathcal{R}}_{AS} &= \text{sample}(\mathcal{N}_N) \\ \underline{\mathcal{R}}_{B} &= \text{sample}(\mathcal{N}_N) \\ \underline{\mathcal{R}}_{C} &= \text{sample}(\mathcal{N}_N) \end{aligned} \tag{3.1}$$

As an example, suppose we have $\mathcal{N}_5 = (2, 3, 5, 1, 4)$ and so, $\mathcal{N}_{5,1} = 2$. Then in the first trial $n = 1$, the $\mathcal{N}_{5,n}$ - th element of anoter array, e.g. $q = (0.1, 0.4, -0.9, 1.2, 1.0)$ is randomly seleceted, hence, $q(n = 1) = 0.4$. In the second trial, looking at $\mathcal{N}_5$, we see, $\mathcal{N}_{5,2} = 3$ so $q(n = 2) = -0.9$, etcetera. In this way a random source $\mathcal{N}$ can be employed in a program and seen as a physical factor giving rise to randomness. The "freely tossing of a coin" is now replaced with "freely randomizing" the $\underline{\mathcal{R}}_X$ by filling it with sample($\mathcal{N}_N$). There can be no fundamental objection to this particular form of randomizing.

3.2 Design time settings

Experimentalists may claim the construction of their measuring instruments. Hence, servers in the experiment may be tuned in design time. There is no fundamental reason to reject design time to the designer of a computer experiment.

Furthermore, the designer may assume that one random source is shared by A and by S. This is the $\underline{\mathcal{R}}_{AS}$. Because there is a flow of particles between the A and the S this sharing, i.e. $\underline{\mathcal{R}}_A = \underline{\mathcal{R}}_S = \underline{\mathcal{R}}_{AS}$, cannot be prevented at run time in a real experiment. The a_n in the experiment are based on the $\underline{a}$ array. For instance $\underline{a} = (1, 2, 1, 2, 1, 2, ...)$. In design time the designer is allowed to introduce a spin-like variable $\sigma \in \{-1, 1\}$. In the sequence of trials, σ_n is selected from $\underline{\sigma} = (-1, 1, -1, 1, -1, 1, ...)$. We may note that, because of $\underline{\mathcal{R}}_A = \underline{\mathcal{R}}_S$ the relation $a_n = 1 + \frac{1}{2}(1 + \sigma_n)$ occurs. The setting a_n can be either 1 or 2 and is already presented in terms of selection unit paramter vectors in $\mathbb{R}^3$.

Note that the σ_n can be send to Bob and to Alice without any additional information conveying its meaning. So, Bob cannot derive anything from σ_n even though the designer knows the relation. This is because Bob is only active in run time, not in design time.

Finally, the source may also send a $\zeta \in \{-1, 1\}$ to both Alice and Bob. The ζ_n in the experiment is based on the $\underline{\mathcal{R}}_C = \text{sample}(\mathcal{N}_N)$ and derives from a $\underline{\zeta}$ array.

The second random source, $\underline{\mathcal{R}}_B$ is used by B exclusively, the third random source, $\underline{\mathcal{R}}_C$ is used by the source exclusively. There appears to be no physical arguments why this is a violation of locality or cannot be found in nature.

3.3 Random sources $\mathcal{R}_.$ and particles

The source sends a $\sigma \in \{-1, 1\}$ and a $\zeta \in \{-1, 1\}$. to both A and B. In a formal format,

$$[A(a_n)] \leftarrow (\sigma, \zeta)_n \leftarrow [S] \rightarrow (\sigma, \zeta)_n \rightarrow [B(b_n)]$$

Here, $[A(a)]$ represents the measuring instrument A where Alice has the a setting. This setting "runs synchronous" with σ in the source because of the "shared" random source. The particle pair source is represented by $[S]$.

The σ and ζ going into the direction of A are equal to the σ and ζ going to B. Each particle is, in the algorithm, a pair (σ, ζ). We note that ζ derives from the random source $\underline{\mathcal{R}}_C$.

3.4 A side processing of the (σ, ζ)

Firstly, let us for the ease of the presentation define a $\sigma_{A,n} = \frac{1+\sigma_n}{2}$. The σ_n at the n-th trial from the source S is a result of the synchronous $\underline{\mathcal{R}}_{AS}$.

The way the information is used remains hidden to B in order to maintain locality in the model. So we have, the setting $a_= \sigma_{A,n} + 1$. Furthermore, we define two functions $\varphi^-_{A,n} = \sigma_{A,n}$ and $\varphi^+_{A,n} = 1 - \sigma_{A,n}$. The two functions, together with ζ_n produce, in turn, a function

$$f_{\zeta_n}(a_n) = \zeta_n \varphi^+_{A,n} - \varphi^-_{A,n}$$

Note that $f_{\zeta_n} \in \{-1, 1\}$. Hence, we can store the outcome of the computations on the A side immediately in an N-size array $S_{A,n}$ for trial number n and $n = 1, 2, 3, ...N$.

3.5 B side processing of the (σ, ζ)

In the first place, let us determine with the B associated die the setting b_n. This results from the hypothetical random source $\underline{\mathcal{R}}_B$. Then, secondly and similar such as in the case of A, but of course completely hidden from A, the $(\sigma, \zeta)_n$ information from the source is processed. We have, $\sigma_{B,n} = \frac{1+\sigma_n}{2}$, then $\varphi^-_{B,n} = \sigma_{B,n}$ and $\varphi^+_{B,n} = \sigma_{B,n} + (\delta_{1,b} - \delta_{2,b})(1 - \sigma_{B,n})$. This leads to the function

$$g_\zeta(b) = \zeta\varphi^+_B + \frac{1-\zeta}{\sqrt{2}}\varphi^-_B$$

For $g_{\zeta_n}(b_n)$ we may note that it projects in the real interval $[-\sqrt{2}, \sqrt{2}]$. If $\sigma_{B,n} = 1$ then $g_{\zeta_n}(b_n) = 1$ for $\zeta_n = 1$ and $\sqrt{2}-1$ for $\zeta_n = -1$. If $\sigma_{B,n} = 0$, then $\varphi^-_{B,n} = 0$ and $g_{\zeta_n}(b_n) = \pm 1$.

Hence, in order to generate a response in $\{-1, 1\}$, a random λ_2 from the real interval $[-\sqrt{2}, \sqrt{2}]$ is uniformly drawn and $S_{B,n} = \mathrm{sgn}(g_\zeta(b) - \lambda_2)$ in the n-th trial. We note that as long as Bob doesn't know the meaning of σ_B, derived from σ and related to the $\underline{\mathcal{R}}_{AS}$, locality is warranted. Bob doesn't have access from the information of design time.

4 Correlation

After the final measurement, the correlation $E(a, b)$ is computed from counting the number of times $S_{A(a),n} = S_{B(b),n}$, i.e., $N(a, b \,|\, S_{A(a),n} = S_{B(b),n})$, and we count the number of times $S_{A(a),n} = -S_{B(b),n}$, i.e. $N(a, b \,|\, S_{A(a),n} = -S_{B(b),n})$. Hence, the expression

$$E(a, b) = \frac{N(a, b \,|\, S_{A(a),n} = S_{B(b),n}) - N(a, b \,|\, S_{A(a),n} = -S_{B(b),n})}{N(a, b \,|\, S_{A(a),n} = S_{B(b),n}) + N(a, b \,|\, S_{A(a),n} = -S_{B(b),n})} \tag{4.1}$$

This computation is also employed in the algorithm.

5 Conclusion & discussion

In the paper a simple design is given that is able to violate the CHSH inequality with numerical values close to the expected quantum mechanics. Please note that no violation of locality is employed. B doesn't know the meaning of the A-S shared information send to B. In fact, A and B process the common input (σ, ζ) differently at their side without knowing of each other's existence.

The reader kindly notes that the construction is directed to explain the outcome of the A-S-B experiment such as in Weihs's [4] and should not be confused with other locality testing experiments or related experimental set ups unequal to the scheme $A(a) \leftarrow S \rightarrow B(b)$. The present result supports the result given in [5].

In the appendix, the important loop in the R program over $n = 1, 2, 3...N$ is presented. In the explanation of entanglement with locality, three random sources $\underline{\mathcal{R}}_{AS}$, $\underline{\mathcal{R}}_C$ and $\underline{\mathcal{R}}_B$ are employed. We note that nobody knows whether or not the measuring instrument A and the particle source, S, share a random source yes or no. Moreover information from design time is not accisble in run time and there is a flow of particles between S and A.

The conceptual weakness of the computer simulation lies in the fact that, in real experiment, both Alice and Bob may change their settings when the two particles (σ, ζ)

are created and are in flight heading to their targets A and B. In our computer model, only Bob may change his setting "in flight".

Changing "in flight" setting at Bob together with no access to design time is a very strong form of information hiding between Alice and Bob. Moreover, "shared random sources" together with "meaning-hidden information transport" via the particles and "synchronized random clocks" cannot be rejected in nature beforehand, we provided another way to look at the criticism raised by Einstein [1]. As required by the author of [6] a computer simulation, be it a somewhat restricted in some details, rejects the criticism raised in [6]. We may do this because our "freezing the setting of a and b at particle creation" is a valid CHSH type of experiment. It would be strange to say that locality and causality cannot occur in an experiment where in flight changes are allowed whereas one must admit that locality and causality occurs at a somewhat more restricted experiment. Moreover, the σ in the computational model may act like a kind of clock $\sigma(t)$. The synchronization starts at creation of the particle in the source. Then the separate $\sigma(t)$ may synchronously change "in flight" until $(\sigma(t), \zeta)_n$ hits the measuring instrument.

We conclude that a local hidden variables explanation of this kind of "freeze setting at particle creation" type of experiment would not be possible if there is no probability loophole in the CHSH [5].

References

[1] A. Einstein, B. Podolsky, N. Rosen, (1935), *Can quantum-mechanical description of physical reality be considered complete, Phys. Rev.* **47** 777-780.

[2] J.S. Bell, (1964), *On the Einstein Podolsky Rosen paradox, Physics* **1** 195-200.

[3] J.F. Clauser, M.A. Horne, A. Shimony, R.A. Holt, (1969), *Proposed experiment to test local hidden-variables theories, Phys. Rev. Lett.* **23** 880-884.

[4] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter & A. Zeilinger, (1998), *Violation of Bell's inequality under strict Einstein locality conditions, Phys. Rev. Lett.* **81**, 5039-5043.

[5] J.F. Geurdes, (2014), *A probability loophole in the CHSH, Results in Physics* **4**, 81-82.

[6] R.D. Gill, (2015), *No probability loophole in the CHSHS, Results in Physics*, http://dx.doi.org/10.106/j.rinp.2015.06.002.

[7] J.F. Geurdes, (2015), *Why one can maintain that there is a probability loophole in the CHSH* arXiv:1508.04798.

Appendix A: Here the nucleus of the algorithm is shown.

```
for (n in 1:N){
#Source section
  zetah<-zeta[RC[n]]
  sygma<-sigma[RAS[n]]
#A section
  aSet<-a[RAS[n]]
  aKeep[n]<-aSet
  phiAmin<-((sygma+1)/2)
  phiAplus<-1-((sygma+1)/2)
  f<-zetah*phiAplus-phiAmin
  scoreA[aSet,n]<-f
#B section
  phiBmin<-((sygma+1)/2)
  bSet<-b[RB[n]]
  bKeep[n]<-bSet
  if(((sygma+1)/2)==1){
    phiBplus<-1
  }else{
    if(bSet==1){
      phiBplus<-1
    }
    if(bSet==2){
      phiBplus<-(-1)
    }
  }
  g<-zetah*phiBplus
  g<-g+((1-zetah)*phiBmin/sqrt(2))
  lambda_2<-runif(1)*sqrt(2)
  lambda_2<-sign(0.5 - runif(1))*lambda_2
  scoreB[bSet,n]<-sign(g-lambda_2)
}
```