

Violating the CHSH Inequality: Empirical Evidence of Quantum Nonlocality (PHY 3035 Quantum Mechanics Honors)

Damien Koon

April 2025

Abstract

The Clauser–Horne–Shimony–Holt (CHSH) inequality provides an experimentally testable framework for Bell’s theorem, and led to John Clauser being awarded the 2022 Nobel Prize in Physics for his work in quantum information. This work replicates and extends these pivotal experiments to explore quantum nonlocality in diverse computational environments. This paper implements three CHSH experiments: a noiseless simulation using Qiskit’s StateVector simulator, a noisy simulation to emulate decoherence effects inherent in quantum computing, and execution on IBM quantum hardware. In each case, entangled Bell states are prepared and measured with optimized settings, leading to the demonstration of violations of the classical bound, $|S| \leq 2$. Our results empirically validate Bell’s theorem, and additionally highlight the resilience of quantum correlations under noise. The paper adopts a pedagogical framework while replicating the CHSH experiment to serve as a practical guide for educators and researchers seeking to reproduce this foundational quantum experiment.

Contents

Introduction	3
Qubits	3
Gates, Phase, and the Bloch Sphere	4
Gates	4
Phase	8
Bloch Sphere	9
Quantum Circuits, Measurement, and Entanglement	10
Quantum Circuits	10
Measurement	11
Bell States and Entanglement	12
CHSH Inequality	14
Bell's Theorem	14
Expectation Values	14
The CHSH parameter, S	16
Tsirelson's bound	18
CHSH Experiment	21
Aer Simulator	21
Qasm Simulator	21
Results	24
Statevector Simulator	25
Results	26
Execution on IBM Quantum Hardware	27
Results	28
Conclusion	29
Bibliography	30

Introduction

Proving Bell's theorem with the CHSH experiment is a foundational result in Quantum Information theory. To begin, this paper will first begin by introducing the required background knowledge adopted from the formalism of quantum information as described by Nielsen and Chuang [1], then it will explain and derive the CHSH inequality, including Tsirelson's bound, and then it will go through the quantum code to experimentally reproduce the results of the CHSH experiment on IBM hardware[9].

Qubits

Qubits, or "quantum bits," are quantum representations of information. Physically, a qubit can be made from any quantum particle that has two distinct states. For example a photon of light being polarized either horizontally or vertically or an electron being spin up or spin down. Physically, qubits are usually either spin up/spin down instead of being in a "zero/one" state.

$$|0\rangle = |\uparrow\rangle \quad |1\rangle = |\downarrow\rangle$$

A singular qubit can be either zero or one (or both!), and represented mathematically by

$$|\psi\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |\psi\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Similarly to classical bits, qubits can be either 0 or 1, however, they can also leverage superposition and be in both states simultaneously

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Where the square of $\frac{1}{\sqrt{2}}$ is the probability of each state, so it's 50% $|0\rangle$ and 50% $|1\rangle$ The qubit will stay in this superposition until interaction with the environment or measurement, then it will collapse into either $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |1\rangle$

$$P(|0\rangle) = \left(\frac{1}{\sqrt{2}}\right)^2 = 0.5 \quad P(|1\rangle) = \left(\frac{1}{\sqrt{2}}\right)^2 = 0.5$$

A more general way to represent qubits is like this, where $|\alpha|^2$ and $|\beta|^2$ are the probabilities of being in the respective states.

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$$

So this is how a single qubit can be represented, but how do we consider a multi-qubit system? To add new qubits to a system, they need to be tensored together.

A 5-qubit system can be represented inside of a ket read right to left. Thus, the first qubit in this system is 1.

$$|\psi\rangle = |01011\rangle$$

This 5-qubit system is quite messy to represent with matrices and can be done with the tensor product.

$$|01011\rangle = |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle$$

The tensor product takes the possible states and expands it to show every possible permutation of the states, for example the two qubits tensored together

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \beta\gamma|10\rangle + \alpha\delta|01\rangle + \beta\delta|11\rangle$$

By tensoring these qubits together, we have collected all possible outcomes of the qubits and their probabilities, where now $|\alpha\gamma|^2$ is the probability for measuring $|00\rangle$

Gates, Phase, and the Bloch Sphere

Gates

In classical computing, there are logic gates, which operate on binary inputs. For example, the AND gate and the OR gate for two binary inputs.

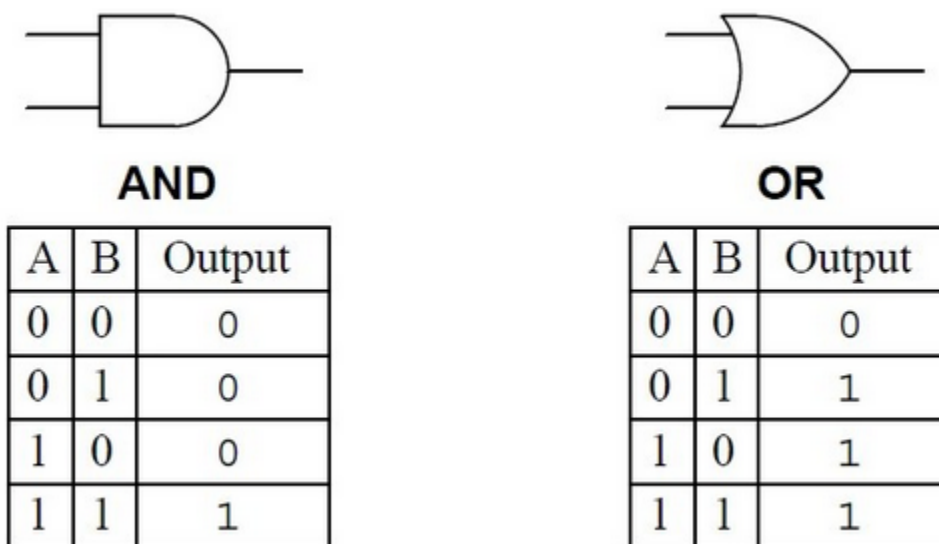


Figure 1: AND and OR Logic Gates
[4]

This leads into the idea of quantum logic gates, or simply qubit gates.[1]

Before going into qubit gates, it is of note that it is also of note that each of these gates are Hermitian unitary operators.

To define Hermitian matrices, we must first define the matrix operation, U^\dagger , pronounced "U dagger." To perform this operation, you must take the transpose of the matrix and then apply the complex conjugate, or vice versa.

$$(U^T)^* = (U^*)^T = U^\dagger$$

For a unitary matrix:

$$U^\dagger U = I$$

For a Hermetian matrix:

$$U = U^\dagger$$

All quantum gates are unitary operators, however not all of the gates are Hermitian.

Now to explain qubit gates, I will show various examples of how they change qubits. However, for the CHSH experiment, we will focus on the H, CNOT, and RY gates.

We will start with the Pauli matrices, X, Y, Z (sometimes $\sigma_x, \sigma_x, \sigma_z$). Each of these gates apply to a single qubit.

The Pauli-X Gate:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The X gate is both unitary and Hermitian, but what does it do?

$$XX^\dagger = I \quad X = X^\dagger$$

The X gate can be used to change a $|0\rangle$ to a $|1\rangle$ or a $|1\rangle$ to a $|0\rangle$.

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle.$$

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 + 0 \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle.$$

The Pauli-Y Gate:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$YY^\dagger = I \quad Y = Y^\dagger$$

The Y gate is noticeably different with the addition of the imaginary number, i . We will later see how this can be visualized with the Bloch sphere. For now, we will just see what happens when we apply it to $|0\rangle$ and $|1\rangle$

$$Y|0\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 1 + (-i) \cdot 0 \\ i \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix} = i|1\rangle.$$

$$Y|1\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \cdot 0 + (-i) \cdot 1 \\ i \cdot 0 + 0 \cdot 1 \end{bmatrix} = \begin{bmatrix} -i \\ 0 \end{bmatrix} = -i|0\rangle.$$

The Pauli-Z Gate:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$ZZ^\dagger = I \quad Z = Z^\dagger$$

$$Z|0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 + 0 \cdot 0 \\ 0 \cdot 1 + (-1) \cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle.$$

$$Z|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 0 \cdot 1 \\ 0 \cdot 0 + (-1) \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = -|1\rangle.$$

The three Pauli matrices are the simplest of the Quantum Gates and they aid in understanding how a qubit can be transformed. We will later see how these qubits geometrically on the Bloch sphere, but, before that we will finish introducing the gates that will be needed for the CHSH experiment.

The Hadamard Gate:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The Hadamard Gate is central to entanglement, because it creates an equal superposition state from a basis state

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + (-1) \cdot 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle).$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 + (-1) \cdot 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle).$$

We originally start with $|0\rangle$, but after applying the Hadamard gate, we have an equal superposition where the qubit is 50% in $|0\rangle$ and 50% in $|1\rangle$. For the CHSH experiment, we will not be using $H|1\rangle$, but it is instructive.

The Controlled X Gate:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

So far, we have only considered single qubit gates, but the CNOT is a two qubit gate. Consider the CNOT gate to be an X gate on a target qubit, but it only activates if the second, "control" qubit is a 1.

An example will not be show using matrix multiplication. Visual examples will be shown in the later section, Quantum Circuits.

Looking forward to entanglement, what happens if the control qubit is in an equal superposition of $|0\rangle$ and $|1\rangle$? As we will soon see, the CNOT gate and the Hadamard gate are central to the entanglement between two qubits.

The RY gate — Rotation operator gate about y-axis

$$RY(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

It is of note that the explicit matrix form is not always needed. The RY gate can also be represented by

$$e^{-i\frac{\theta}{2}Y}$$

Which can be expanded out by using the Taylor series

$$\begin{aligned} e^{-i\frac{\theta}{2}Y} &= \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)Y \\ &= \cos\left(\frac{\theta}{2}\right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i\sin\left(\frac{\theta}{2}\right) \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \end{aligned}$$

You might recall that we have the Y gate, which is a rotation of π radians about the y-axis.

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

So what happens if we plug in π to the RY gate?

$$RY(\pi) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

This looks similar to the Y gate, but it's missing the i term! Why is that? This is due to the idea of global and relative phase. The Y gate represents a rotation by π radians about the y-axis, but with an additional global phase of -i

$$RY(\pi) = -iY$$

The Pauli-Y gate is defined with a conventional global phase to satisfy anti-commutativity along with the Pauli-X and Pauli-Z operators. This is not highly relevant, and the key takeaway should be that $RY(\pi)$ and Y perform the same physical rotation, but differ by a global phase of -i.

Phase

First let us take a base qubit, $|0\rangle$, and then let us apply a Hadamard gate to split it into a superposition:

$$|\psi\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Now that we have an equal superposition, let's apply a Z gate and see what happens.

$$Z\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$Z|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

This is very similar to just applying a Hadamard gate to $|0\rangle$, but it has changed $|1\rangle$ to $-|1\rangle$. Recall Euler's identity and note that we can rewrite this like this.

$$Z|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi}|1\rangle)$$

This is the idea of relative phase. Recall that the Z gate rotates the qubit around the z-axis by π , so what if we want to rotate it by some angle ϕ ?

A general form:

$$|\psi\rangle = \alpha|0\rangle + e^{i\phi}\beta|1\rangle$$

But why is the $e^{i\phi}$ only on $|1\rangle$? Let's try putting it on both the 0 and 1 kets.

$$|\psi\rangle = e^{i\phi}\alpha|0\rangle + e^{i\phi}\beta|1\rangle = e^{i\phi}(\alpha|0\rangle + \beta|1\rangle)$$

The global phase, $e^{i\phi}$, has no observable consequences and doesn't change the probabilities of getting a measurement result and is considered to be physically irrelevant, so it can be left off.[1]

$$e^{i\phi}(\alpha |0\rangle + \beta |1\rangle) = \alpha |0\rangle + \beta |1\rangle$$

Note that if both phases are different, it just reduces to the relative phase.

$$|\psi\rangle = e^{i\theta}\alpha |0\rangle + e^{i\phi}\beta |1\rangle = \alpha |0\rangle + e^{i(\phi-\theta)}\beta |1\rangle$$

Also note that the probabilities do not change since $|e^{i\phi}|^2 = 1$

The phase factor on the 1 ket instead of the 0 ket by standard convention. It can be rewritten to be on the 0 ket instead if needed

$$|\psi\rangle = \alpha |0\rangle + e^{i\phi}\beta |1\rangle = e^{i2\pi}\alpha |0\rangle + e^{i\phi}\beta |1\rangle = e^{i(2\pi-\phi)}\alpha |0\rangle + e^{i\phi}\beta |1\rangle$$

Bloch Sphere

We need more ways to represent qubits other than Dirac notation. It would be beneficial to have a geometric representation, so we can better visualize changes to a qubit. Thus, the physicist, Felix Bloch, introduced a sphere on which qubits can be represented as a point on the surface.

Higher vertically means that the qubit is more likely to be $|0\rangle$ and vice versa for $|1\rangle$

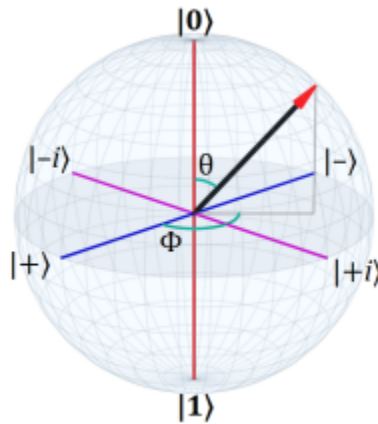


Figure 2: The Bloch Sphere
[5]

But what are these different states along the equator?

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \quad |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$$

There are in between $|0\rangle$ and $|1\rangle$, so that would imply that the states are equally probable. The only difference between them is their phase.

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi}|1\rangle) & |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi}|1\rangle) \\ | +i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{2}}|1\rangle) & |-i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{3\pi}{2}}|1\rangle) \end{aligned}$$

Quantum Circuits, Measurement, and Entanglement

Quantum Circuits

So far, we have discussed gates, phase, and the Bloch sphere, but each of these discussions has only involved a single qubit. How can we apply a gate to a single qubit in a multi-qubit system, and how can we represent this system graphically?

Before we introduce quantum circuits, let us recall the AND and OR gates from the previous section, and lets create a classical circuit.

The Classical Circuit in Figure 3 takes 3 inputs, A, B, and C and provides 1 output, Q, after passing A, B, and C through 2 AND gates and 1 OR gate.

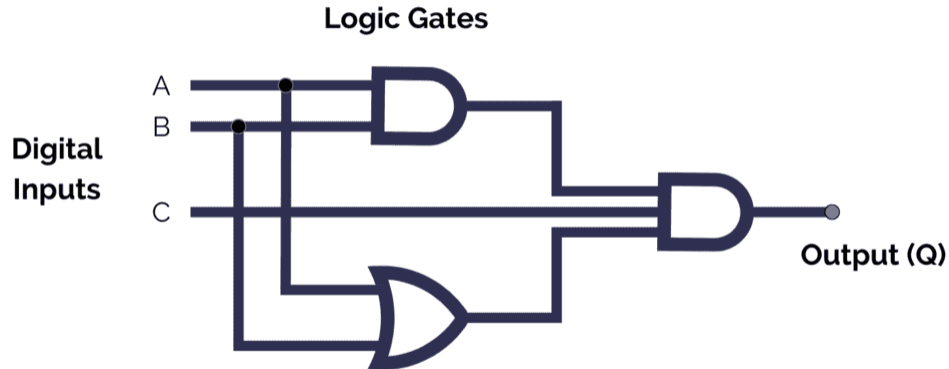
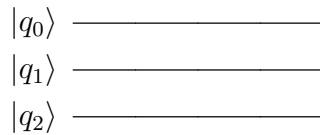


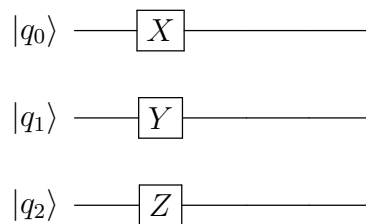
Figure 3: Classical Circuit
[6]

This idea from classical computing leads into the idea of the quantum circuit — a graph with lines as "wires" which read left to right as different points in time with each qubit in a system having a separate wire.

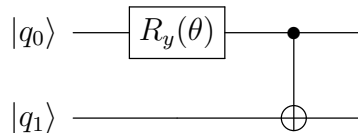
To draw the base diagram, first consider a 3 qubit system without any gates applied



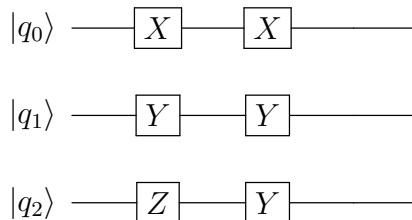
Now what if we want to apply an X gate to qubit 0, a Y gate to qubit 1, and a Z gate to qubit 2?



Quantum circuits are intuitive and very useful for visualizing a qubit system. Let's explore a few more examples.



This system has an RY gate applied on qubit 0, and a CNOT gate controlled by qubit 0 and targeted to qubit 1.

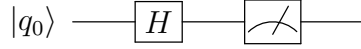


This system has two of each Pauli gate applied to its respective qubit. Recall that the X, Y, and Z gates are both unitary, so this circuit is equivalent to a circuit without any gates applied.

Measurement

We briefly mentioned measurement previously when discussing the probabilities, but now we can see how it is implemented in a quantum circuit.

We have created a quantum circuit with one qubit with a Hadamard gate applied, and then after it is applied, the qubit is measured.



Recall that the Hadamard gate takes a basis qubit, and splits it into equal probabilities. Thus, if it is measured, the qubit will have a 50/50 chance to collapse into either state.

During this experiment, measurement will be handled by the backend. However, it is important to understand what is happening at a fundamental level.

Consider the X-basis states, $|+\rangle$ and $|-\rangle$, and a qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, such that $|\alpha|^2 + |\beta|^2 = 1$

By the Born rule[1], the probability of the qubit being in the $|+\rangle$ is

$$\langle\psi|+\rangle \langle+|\psi\rangle = |\langle+|\psi\rangle|^2 = \frac{\langle 0| + \langle 1|}{\sqrt{2}}(\alpha|0\rangle + \beta|1\rangle) = \frac{\alpha + \beta}{\sqrt{2}}$$

Where the probability is

$$p(+)=\left|\frac{\alpha + \beta}{\sqrt{2}}\right|^2 = \frac{|\alpha + \beta|^2}{2}$$

And similarly for $|-\rangle$

$$\langle\psi|-\rangle \langle-|\psi\rangle = |\langle-|\psi\rangle|^2 = \frac{\langle 0| - \langle 1|}{\sqrt{2}}(\alpha|0\rangle + \beta|1\rangle) = \frac{\alpha - \beta}{\sqrt{2}}$$

Where the probability is

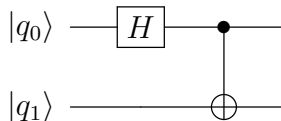
$$p(-)=\left|\frac{\alpha - \beta}{\sqrt{2}}\right|^2 = \frac{|\alpha - \beta|^2}{2}$$

So if $\alpha = 1, \beta = 0$, an equal probability emerges.

$$p(+)=\frac{|1+0|^2}{2} = 1/2 \qquad p(-)=\frac{|1-0|^2}{2} = 1/2$$

Bell States and Entanglement

As we saw from the Hadamard gate, it takes a basis state, and splits them into a superposition. So what if we take two qubits, apply a Hadamard gate, and then apply a CNOT gate?



This two qubit circuit is the classical example of entanglement and is referred to as a Bell state. More specifically, this is a "maximally entangled two-qubit Bell state."

This Bell state can be mathematically represented by

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

There are 4 different maximally entangled Bell states, but $|\Phi^+\rangle$ is the most common and will be used in this CHSH experiment.

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) & |\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) & |\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned}$$

But what does it mean for two qubits to be entangled?

First, consider a two qubit system

$$|\psi_0\rangle = \alpha |0\rangle + \beta |1\rangle$$

$$|\psi_1\rangle = \gamma |0\rangle + \delta |1\rangle$$

And as we saw from the previous example earlier in the introduction,

$$|\psi_0\rangle \otimes |\psi_1\rangle = \alpha\gamma |00\rangle + \beta\gamma |10\rangle + \alpha\delta |01\rangle + \beta\delta |11\rangle$$

now compare this to the entangled Bell state, $|\Phi^+\rangle$

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Let's try to match the coefficients of the original two qubit system to the Bell state. I.E. we need to find coefficients such that

$$\alpha\gamma = 0 \quad \alpha\delta = \frac{1}{\sqrt{2}} \quad \beta\gamma = \frac{1}{\sqrt{2}} \quad \beta\delta = 0$$

It is impossible to match these coefficients from the sub system to the system of a Bell state. This implies non-separability of the qubits in the Bell state. This is the core idea of entanglement.

CHSH Inequality

The CHSH inequality is a more robust version of Bell's inequality. Both are used to prove Bell's theorem, however Bell's inequality requires a noiseless environment.

The section will outline the theory behind the CHSH experiment, including derivations of the CHSH inequality, how the expectation values are calculated, and the true upper bound for the CHSH parameter, Tsirelson's bound.

Bell's Theorem

Bell's theorem states that no local hidden variable theory can reproduce all of the predictions of quantum mechanics. [2]

A local hidden variable theory assumes that the universe operates with "hidden variables" which cannot be measured and there is no way for us to obtain information about them, with the additional principle of locality stating that objects must be influenced only by its immediate surroundings.

The principle of locality evolved from the theories of classical physics, and is central to special and general relativity. One of Einstein's primary postulates is one of locality — No information or causal influence can travel faster than the speed of light.

Einstein personally advocated for local realism, which states that there is no instantaneous action at a distance and that physical properties exist independently of observation. However, the CHSH experiment is evidence against this idea of local realism.

The CHSH inequality provides a practical way to test Bell's theorem by bounding correlations between two spatially separated systems under the assumptions of a local hidden variable theory. [3]

Expectation Values

We will soon see that the CHSH parameter is composed of expectation values and is written as

$$S = E(A_0B_0) - E(A_0B_1) + E(A_1B_0) + E(A_1B_1)$$

Where A_i and B_j are the measurement settings for two qubits, which we will name Alice and Bob respectively. So compute S , we must first calculate these individual expectation values.

However, we must note the correlation between A_0 and B_0 to the Z-basis and A_1 and B_1 to the X-basis

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Where the eigenstates are $|0\rangle$ and $|1\rangle$, with eigenvalues of ± 1

$$\det(Z - \lambda I) = 0 = (1 - \lambda)(-1 - \lambda) \implies \lambda = \pm 1$$

For $\lambda = +1$:

$$(Z - I)\mathbf{v} = 0$$

$$\begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \rightarrow v_2 = 0$$

$$\mathbf{v} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

For $\lambda = -1$:

$$(Z + I)\mathbf{v} = 0$$

$$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \rightarrow v_1 = 0$$

$$\mathbf{v} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Where the eigenstates are $|+\rangle$ and $|-\rangle$, with eigenvalues of ± 1

$$\det(X - \lambda I) = 0 = \lambda^2 - 1 \implies \lambda = \pm 1$$

For $\lambda = +1$:

$$(X - I)\mathbf{v} = 0$$

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \implies v_1 = v_2$$

$$\mathbf{v} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |+\rangle$$

For $\lambda = -1$:

$$(X + I)\mathbf{v} = 0$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \implies v_1 = -v_2$$

$$\mathbf{v} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = |-\rangle$$

To optimize the violation of the CHSH inequality, we want to misalign Alice and Bob's measurement bases by 45° on the Bloch sphere. The Z and X bases are orthogonal, but optimal angles are adjusted with the RY gate to maximize correlations.

The expectation value $E(A_i B_j)$ is computed by

$$E(A_i B_j) = \langle \psi | A_i \otimes B_j | \psi \rangle$$

Where $|\psi\rangle$ is the shared entangled state, $|\Phi^+\rangle$

However, in practice the expectation values are derived from repeated trials where Alice and Bob record their outcomes, a_k, b_k for the measurement settings A_i, B_j .

$$E(A_i B_j) = \frac{1}{N} \sum_{k=1}^N a_k b_k$$

Such that $a_k, b_k \in \{\pm 1\}$ are the outcomes for the total number of trials.

If both of the outcomes are ± 1 , then the outcomes agree, and are positively correlated. However, if one correlation is $+1$ and the other is -1 , then the outcomes disagree, and are negatively correlated. Thus we can write the expectation values as such

$$E(A_i, B_j) = \frac{N_{++} - N_{+-} - N_{-+} + N_{--}}{N_{total}}$$

The CHSH parameter, S

The CHSH parameter¹, S, is given by

$$S = E(A_0 B_0) - E(A_0 B_1) + E(A_1 B_0) + E(A_1 B_1)$$

Where $E(A_i B_j)$ represents the expectation values of the product of the measurement outcomes for two qubits, of which we will name Alice and Bob.

This expression arises by strategic choice. It is designed to maximize violation of the CHSH inequality and to simplify experimentation from Bell's original inequality, which was ineffective against experimental noise. [3]

¹Throughout this paper, S is referred to using the terms "parameter" and "value" interchangeably. However, "parameter" is the more accurate term.

To proceed with the derivation of the inequality, assume that outcomes are predetermined by a hidden variable, λ . Locality ensures that outcomes only depend on local settings, and realism assumes that outcomes exist independently outside of measurement.

For a fixed value of λ , with binary outcomes,

$$A_0(\lambda), A_1(\lambda), B_0(\lambda), B_1(\lambda) \in \{\pm 1\}$$

Then the CHSH value becomes

$$S(\lambda) = A_0(\lambda)B_0(\lambda) - A_0(\lambda)B_1(\lambda) + A_1(\lambda)B_0(\lambda) + A_1(\lambda)B_1(\lambda)$$

$$S(\lambda) = A_0(\lambda)[B_0(\lambda) - B_1(\lambda)] + A_1(\lambda)[B_0(\lambda) + B_1(\lambda)]$$

Recall that each value can only take on the values of ± 1 , so it is quite easy to tabulate all of the possible combinations of values and evaluate from there.

All possible permutations				
B_0	B_1	$B_0 - B_1$	$B_0 + B_1$	$S(\lambda)$
+1	+1	0	+2	$+2A_1$
+1	-1	+2	0	$+2A_0$
-1	+1	-2	0	$-2A_0$
-1	-1	0	-2	$-2A_1$

Thus, in all cases and for all hidden variables, λ , $|S(\lambda)| \leq 2$ holds. Due to the absolute value, the bound $S \leq -2$ is also valid, but $S \geq 2$ is more commonly used

This inequality is experimentally testable and can be violated with a Bell state, but how can we maximize this violation to avoid experimental noise? Looking ahead and knowing that the value $|S| \geq 2$ will be violated, then what is the maximum possible violation?

Tsirelson's bound

If we find that the CHSH value is greater than 2, we have proven the contrapositive of the assumptions used in deriving the CHSH inequality. This will be done in the next section, but what is the maximum possible violation within the realm of quantum mechanics?

This maximum value is known as the Tsirelson bound, named after the mathematician Boris S. Tsirelson. In his 1980 paper, Tsirelson derived the the upper bound for the CHSH and highlighted the differences in Bell's 1964 framework of local hidden variable theories. [7] [2]

Using a slightly different notation than before, we will use the operator norm of the CHSH operator, T . This is not the same as S , but it is directly related by the expectation value.

$$S = \langle A_0 \otimes B_0 \rangle - \langle A_0 \otimes B_1 \rangle + \langle A_1 \otimes B_0 \rangle + \langle A_1 \otimes B_1 \rangle$$

$$S = \langle \psi | T | \psi \rangle$$

$$T = A_0 \otimes B_0 - A_0 \otimes B_1 + A_1 \otimes B_0 + A_1 \otimes B_1$$

As an intermediate step

$$T = A_0 \otimes (B_0 - B_1) + A_1 \otimes (B_0 + B_1)$$

$$T^2 = (A_0 \otimes (B_0 - B_1) + A_1 \otimes (B_0 + B_1))^2$$

$$T^2 = (A_0 \otimes (B_0 - B_1))^2 + (A_1 \otimes (B_0 + B_1))^2 + A_0 A_1 \otimes ((B_0 - B_1)(B_0 + B_1)) + A_1 A_0 \otimes ((B_0 + B_1)(B_0 - B_1))$$

First consider the square terms and recall that $A_i^2 = B_j^2 = I$

$$(B_0 - B_1)^2 = 2I - (B_0 B_1 + B_1 B_0)$$

$$(B_0 + B_1)^2 = 2I + (B_0 B_1 + B_1 B_0)$$

Thus,

$$(A_0 \otimes (B_0 - B_1))^2 = A_0^2 \otimes (B_0 - B_1)^2 = I \otimes (2I - (B_0 B_1 + B_1 B_0))$$

$$(A_1 \otimes (B_0 + B_1))^2 = A_1^2 \otimes (B_0 + B_1)^2 = I \otimes (2I + (B_0 B_1 + B_1 B_0))$$

Adding them together gives,

$$I \otimes [2I - (B_0B_1 + B_1B_0) + 2I + (B_0B_1 + B_1B_0)] = I \otimes 4I = 4I$$

Now consider the cross terms, and remember that terms are not necessarily commutative.

$$(B_0 - B_1)(B_0 + B_1) = B_0B_1 - B_1B_0 = [B_0, B_1]$$

and similarly,

$$(B_0 + B_1)(B_0 - B_1) = B_1B_0 - B_0B_1 = [B_1, B_0] = -[B_0, B_1]$$

Thus the cross terms become

$$\begin{aligned} A_0A_1 \otimes [B_0, B_1] + A_1A_0 \otimes (-[B_0, B_1]) &= A_0A_1 \otimes [B_0, B_1] - A_1A_0 \otimes [B_0, B_1] \\ &= [A_0A_1 - A_1A_0] \otimes [B_0, B_1] = [A_0, A_1] \otimes [B_0, B_1] \end{aligned}$$

Combining all of the terms,

$$T^2 = 4I + [A_0, A_1] \otimes [B_0, B_1]$$

Notice that if the commutators are zero, then we see the classical bound, $T = 2I$

To evaluate these commutators requires the Simultaneous diagonalization theorem [1], which leads into the commutation relations for the Pauli matrices as stated by Nielsen and Chuang. This results will not be fully shown here.

Recall that the values A_0, B_0 correlate to Z, and the values A_1, B_1 correlate the X. Then we can find the commutation relations for Z and X.

$$ZX = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

$$XZ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

$$[Z, X] = ZX - XZ = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix}.$$

Compare with $2iY$:

$$2iY = 2i \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix}.$$

$$[Z, X] = 2iY$$

$$[A_0, A_1] = [Z, X] = 2iY \quad [B_0, B_1] = [Z, X] = 2iY$$

$$T^2 = 4I + [A_0, A_1] \otimes [B_0, B_1] = 4I + 2iY \otimes 2iY = 4I - 4(Y \otimes Y)$$

Since $Y^2 = I$, the eigenvalues are ± 1 , thus it follows that

$$T^2 = 4I \pm 4I$$

ignoring $T^2 = 0$,

$$||T^2|| \leq 8 \implies ||T|| \leq 2\sqrt{2}$$

$$|S| \leq 2\sqrt{2}$$

With quantum mechanics, we find the maximum value for the CHSH value is $2\sqrt{2}$. This is the theoretical limit, even with higher-dimensional entanglement, no quantum state or measurement strategy will exceed this. Physically it demonstrates that quantum correlations are stronger than classical, however still bounded by $2\sqrt{2}$

CHSH Experiment

Qiskit, or the Quantum Information Software Kit, is an open source software development kit (SDK) created by IBM researchers to execute programs on quantum computers. Qiskit has various tools for creating and manipulating quantum information.

Qiskit allows usage of their prototype quantum devices on the IBM Quantum Platform or by using locally run simulators. In this section there are 3 different programs. Two using the Aer simulator, and one that is compatible with IBM quantum hardware.

Aer Simulator

To recreate the CHSH experiment, I first used the Aer simulator module with Qiskit.[8]. Two simulator backends were used, the Qasm Simulator which simulates a noisy quantum circuit, and the Statevector Simulator, which is an ideal quantum circuit simulator.

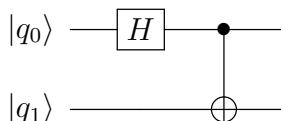
We will skip over some of the Qiskit and Aer simulator setup. Thus, the code in this paper is pseudo-code that directly relates to the CHSH experiment. The complete code can be found on <https://github.com/acroscopic/CHSH/>

Qasm Simulator

The first step was to create a bell state between two qubits, Alice, $q[0]$ and Bob, $q[1]$. 2 classical bits are also created for storing the results of measurement.

```
1 q = QuantumRegister(2, 'q')
2 c = ClassicalRegister(2, 'c')
3
4 def bell():
5     qc = QuantumCircuit(q, c)
6     qc.h(q[0])
7     qc.cx(q[0], q[1])
8
9     return qc
```

Recall the graphical visualization of this bell state. The only difference between this circuit and the code, is that two classical bits are created for storing the results after measurement.

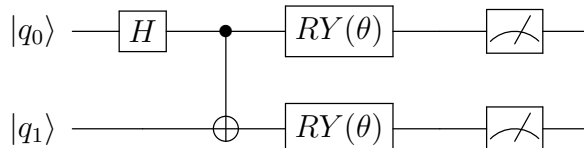


The next step is to measure the qubits and store them in the classical bits. However, before measurement, the RY gate is applied to the measured qubit. The angle at which the RY gate is applied will be discussed shortly.

```

1 def measure(qc, angle, qubit, cbit):
2     qc.ry(-angle, qubit)
3     qc.measure(qubit, cbit)
4     return qc

```



Now we need to figure out a way to calculate the expectation values. Recall that the individual expectation values are calculated by

$$E(A_i, B_j) = \frac{N_{++} - N_{+-} - N_{-+} + N_{--}}{N_{total}}$$

```

1 def expectation(counts):
2     total = sum(counts.values())
3     count_00 = counts.get('00', 0)
4     count_11 = counts.get('11', 0)
5     count_01 = counts.get('10', 0)
6     count_10 = counts.get('01', 0)
7     E = (count_00 + count_11 - count_01 - count_10) / total
8     return E

```

You might notice that $count_01 = counts.get('10')$ and $count_10 = counts.get('01')$. This is due to Qiskit inherently ordering bitstrings from right to left instead of left to right.

Now that the setup is done, it is time to implement the CHSH experiment directly.

```

1 def CHSH():
2     backend = Aer.get_backend('qasm_simulator')
3     # Uses the noisy simulator backend
4
5     S = 0 # initializes the CHSH value
6
7     # These angles provide the maximum violation of the CHSH
8     # inequality, known as Tsirelson's bound
9     angles = {
10         'A0': np.pi / 2,
11         'A1': 0,
12         'B0': np.pi / 4,
13         'B1': -np.pi / 4
14     }
15     measurement_settings = [
16         ('A0', 'B0'),
17         ('A0', 'B1'),
18         ('A1', 'B0'),
19         ('A1', 'B1'),
20     ]
21     for alice, bob in measurement_settings:
22         qc = bell()
23
24         # Apply Alice's A0 and A1 measurements on the 0th qubit and
25         # store it in the 0th bit for each measurement setting
26         qc = measure(qc, angles[alice], 0, 0)
27
28         # Apply Bob's B0 and B1 measurements on the 1st qubit and
29         # store it in the 1st bit for each measurement setting
30         qc = measure(qc, angles[bob], 1, 1)
31
32         # 10 million iterations for the expectation values
33         transpiled_qc = transpile(qc, backend)
34         job = backend.run(transpiled_qc, shots=10000000)
35         result = job.result()
36         counts = result.get_counts()
37
38         # Calculate the expectation value for each measurement setting
39         E = expectation(counts)
40
41         #  $S = E(A_0 B_0) - E(A_0 B_1) + E(A_1 B_0) + E(A_1 B_1)$ 
42         if alice == 'A0' and bob == 'B1':
43             S -= E
44         else:
45             S += E
46     return S

```

Results

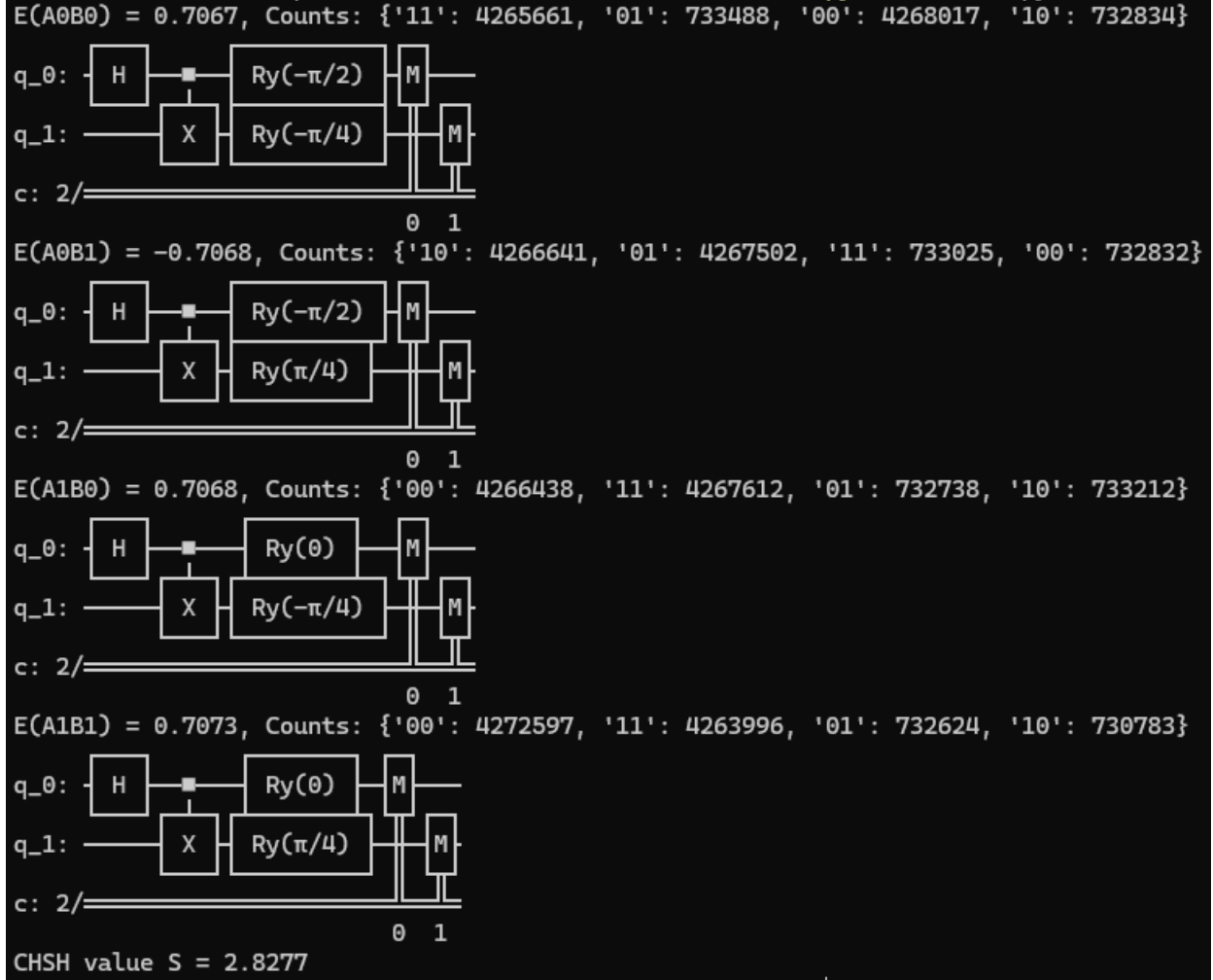


Figure 4: Terminal output from running Aer_Qasm.py

After running the code, with a total of 10 million iterations per expectation value, we get a final CHSH value of 2.8277, which is just under the maximum bound of $2\sqrt{2} \approx 2.828...$

However, since the Qasm simulator is a noisy quantum circuit simulator, values above Tsirelson's bound can be achieved. This would be a major alarm, but this is an issue inherent in the Qasm simulator, and with finite sampling in general. If you use a higher shot count, the values get closer to the bound $2\sqrt{2}$.

CHSH value $S = 2.8288$

Figure 5: Noise-induced violation of Tsirelson's bound

Statevector Simulator

Previously, we used the Qasm Simulator within Aer, however due to finite sampling, we got values that were above the maximum violation of the CHSH inequality, Tsirelson's bound.

Let's now recreate the results from the Qasm Simulator, but with the StateVector simulator, which is a noiseless backend.

The code is largely the same, however it removes the need for finite sampling due to the shots count needed for calculating the expectation values.

The begins by initializing the backend, CHSH values, angles, and the measurement settings. Functionally, this is the same as Aer_Qasm.py.

```
1 backend = Aer.get_backend('statevector_simulator')
2
3 S = 0
4
5 angles = {
6     'A0': np.pi / 2,    # X-basis measurement
7     'A1': 0,             # Z-basis measurement
8     'B0': np.pi / 4,    # rotated Z-basis measurement
9     'B1': -np.pi / 4    # rotated Z-basis measurement
10 }
11 measurement_settings = [
12     ('A0', 'B0'),
13     ('A0', 'B1'),
14     ('A1', 'B0'),
15     ('A1', 'B1'),
16 ]
```

Next, we loop through the measurement settings, create a Bell state, and apply the measurement rotations.

```
1 for alice, bob in measurement_settings:
2     # Initialize circuit with 2 qubits (no classical registers)
3     qc = QuantumCircuit(2)
4
5     # Create Bell state between Alice and Bob
6     qc.h(0)
7     qc.cx(0, 1)
8
9     # Apply measurement rotations
10    qc.ry(-angles[alice], 0) # Alice's basis
11    qc.ry(-angles[bob], 1)   # Bob's basis
```

Now that we have prepared the system, it's time to make a measurement and find the expectation values. This is done by simulating the statevectors for each measurement settign, and the squaring the probability amplitude.

```

1 # Simulate to get the statevector
2 transpiled_qc = transpile(qc, backend)
3 result = backend.run(transpiled_qc).result()
4 statevector = result.get_statevector()
5 # Calculate probabilities by squaring the probability amplitude
6 probs = np.abs(statevector)**2
7 # Extract probabilities for each outcome
8 # "little-endian" ordering: |q1 q0>
9 prob_00 = probs[0] # |00>
10 prob_01 = probs[1] # |01>
11 prob_10 = probs[2] # |10>
12 prob_11 = probs[3] # |11>
13 # Compute expectation value
14 # recall negative and positive correlations
15 E = (prob_00 + prob_11) - (prob_01 + prob_10)

```

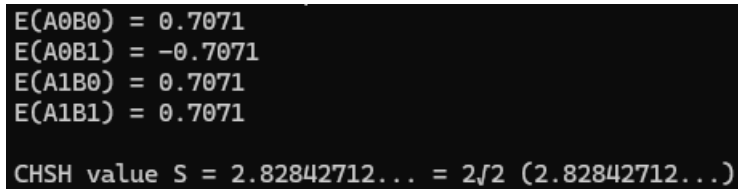
All that remains is compute the CHSH value and compare the results to the Qasm simulator.

```

1 def CHSH():
2     if alice == 'A0' and bob == 'B1':
3         S -= E
4     else:
5         S += E
6     print(f"E({alice}{bob}) = {E:.4f}")
7     return S
8 S_exact = CHSH()
9 print(f"\nCHSH value S = {S_exact:.8f}... = 2sqrt(2) ({2*np.sqrt(2):.8f}...)")

```

Results



```

E(A0B0) = 0.7071
E(A0B1) = -0.7071
E(A1B0) = 0.7071
E(A1B1) = 0.7071

CHSH value S = 2.82842712... = 2√2 (2.82842712...)

```

Figure 6: Terminal output from running Aer_StateVector.py

The Aer statevector simulator has provided a noiseless environment to run the CHSH experiment. We can see that the value obtain is identical to $2\sqrt{2}$. However running the CHSH experiment on real hardware does not have the luxury of a noiseless environment. To run on IBM hardware, the code must be modified so that the violation can still be clearly observed even with experimental noise. To do this, we will iterate through equally spaced angles to be used in the RY gate. The more iterations would prove more of a statistical significance, provide a high maximum violation.

Execution on IBM Quantum Hardware

Previously, the section of code was ran using the Aer simulator and did not interface with any quantum hardware. It is highly relevant to empirically replicate the results of the CHSH experiment using a real quantum computer [9]. The setup is different, but the theory behind it is not.

Since there is a lot of setup for IBM hardware, more Qiskit specific aspects of the code will be shown. However, this is still pseudo-code, and the full version can be found on <https://github.com/acroscopic/CHSH/>

First, we need to set up the type of backend we want to use. We will filter for hardware with 127 qubits, which is the modern "Eagle" processor-based systems via IBM Quantum [9].

```
1 backend = service.least_busy(simulator=False, operational=True,
    min_num_qubits=127)
```

For this version of the code instead of running millions of calculations to find the expectation values, the angle for the RY gate will be parametrized between equally spaced values over a full 0 to 2π rotation. It can do it this way because of the difference in calculating the expectation values.

```
1 theta = Parameter("$\\theta$")
2
3 qc = QuantumCircuit(2)
4
5 qc.h(0)
6 qc.cx(0, 1)
7
8 qc.ry(theta, 0)
9
10 number_of_phases = 20
11 phases = np.linspace(0, 2 * np.pi, number_of_phases)
```

One of the biggest difference between the previous Aer simulator code and this version, is that expectation values are written in terms of the observables as Pauli matrices, the X and Z gates. Where X and Z are tensored together. This is just an alternative method to calculate the same expectation values as before.

```
1 # S = <ZZ> - <ZX> + <XZ> + <XX>
2 observable = SparsePauliOp.from_list([
3     ("ZZ", 1), # E(A_0 B_0)
4     ("ZX", -1), # - E(A_0 B_1)
5     ("XZ", 1), # E(A_1 B_0)
6     ("XX", 1) # E(A_1 B_1)
7 ])
```

Since we are running this code on hardware, we need to modify our data to fit the specific hardware

```

1 target = backend.target
2 pm = generate_preset_pass_manager(target=target, optimization_level
  =3)
3 qc_isa = pm.run(qc)
4 isa_observable = observable.apply_layout(layout=qc_isa.layout)
5 estimator = Estimator(mode=backend)
6 pub = (
7     qc_isa, # remapped ISA circuit
8     [[isa_observable]], # remapped observables
9     individual_phases # Parameter values to test
10 )

```

And then finally, we submit the job to queue and wait to receive the results.

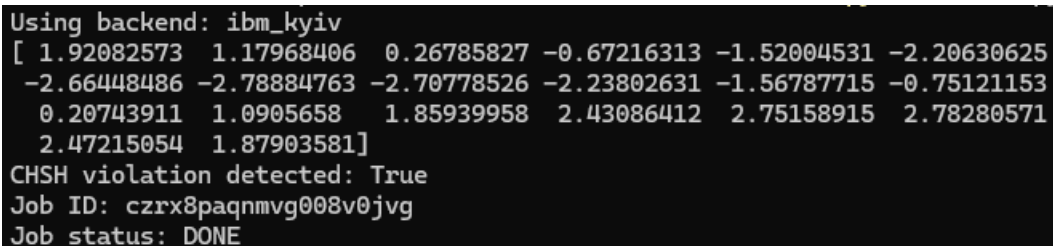
```

1 job = estimator.run(pubs=[pub])
2 result = job.result()
3
4 S_values = result[0].data.evs[0] # Expectation values for the
  observable
5 print(S_values)
6
7 violation = np.any(np.abs(S_values) > 2)
8 print(f"CHSH violation detected: {violation}")
9
10 print(f"Job ID: {job.job_id()}") # Unique identifier for job
11 print(f"Job status: {job.status()}") # Should be 'DONE' if
  successful

```

Results

After running the code on IBM's quantum computer in Ukraine, Kyiv, we get multiple CHSH values corresponding to the different angles of the RY gates, many of which violate the classical bound $|S| \leq 2$.



```

Using backend: ibm_kyiv
[ 1.92082573  1.17968406  0.26785827 -0.67216313 -1.52004531 -2.20630625
 -2.66448486 -2.78884763 -2.70778526 -2.23802631 -1.56787715 -0.75121153
  0.20743911  1.0905658  1.85939958  2.43086412  2.75158915  2.78280571
  2.47215054  1.87903581]
CHSH violation detected: True
Job ID: czrx8paqnmvg008v0jvg
Job status: DONE

```

Figure 7: Terminal output from running CHSH_experiment.py

Conclusion

After executing the CHSH experiment on both a noiseless and a noisy Aer Simulators and on IBM hardware, we observed the CHSH parameter, $S > 2$, violating the classical bound of $|S| \leq 2$. This empirical violation of the CHSH inequality provides strong evidence against local hidden variable theories, which assume that outcomes are determined by hidden properties independent of distant measurements.

By violating this inequality, we have empirically validated Bell's theorem, demonstrating that non-local realism theories can fully describe quantum correlations. More specifically, the results reject local realism, which is the joint assumptions of locality and realism. Quantum mechanics inherently incorporates non-local correlations through entangled states, as seen by the Bell state used in this experiment.

This violation implies that quantum mechanics exhibits non-locality. While this does not imply instantaneous communication due to the no-signaling theorem, it confirms that quantum systems cannot be described by the local deterministic methods from classical physics without violating this empirical evidence.

The noisy simulator and hardware results highlight the fragility of quantum correlations in practical settings. Despite this, the measured violations consistently surpass $S = 2$, further reinforcing the necessity of quantum mechanics being an accurate representation of nature.

These findings underscore the necessity of embracing quantum theory and its inherently non-classical framework to be able to describe the fundamental aspects of nature. The results align with the Copenhagen interpretation, and the violation challenges many classically held intuitions about causality and locality.

In summary, this experiment shows that quantum mechanics is not solely a statistical tool, but a fundamental framework to describe reality.

Bibliography

References

- [1] Nielsen, M. A., & Chuang, I. L. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [2] Bell, J. S. "On the Einstein Podolsky Rosen Paradox." *Physics Physique Fizika*, vol. 1, pp. 195–200, 1964. https://cds.cern.ch/record/111654/files/vol1p195-200_001.pdf
- [3] Clauser, J. F., Horne, M. A., Shimony, A., & Holt, R. A. "Proposed Experiment to Test Local Hidden-Variable Theories." *Physical Review Letters*, vol. 23, no. 15, pp. 880–884, 1969. <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.23.880>
- [4] Toheeb Olaide/Class Notes, "logic-gates-symbol-csc-classnotesng.jpg," classnotes.ng, Date, <https://classnotes.ng/lesson/logic-gate>
- [5] Wyrđ Smythe/Logos Con Carne, "bloch-hdr.png," logosconcarne.com, March 15, 2021, <https://logosconcarne.com/2021/03/15/qm-101-bloch-sphere/>
- [6] Liam Cope, Logic Circuit: Definition, Examples, Types and FAQs, January 2025, , <https://engineerfix.com/electrical/plc/logic-circuit/>
- [7] Tsirelson, B. S. "Quantum Generalizations of Bell's Inequality." *Letters in Mathematical Physics*, vol. 4, pp. 93–100, 1980.
- [8] IBM, *Qiskit: An Open-Source Framework for Quantum Computing*, 2025, 2.0, <https://qiskit.org>
- [9] IBM, *IBM Quantum Computing*, 2025, <https://quantum-computing.ibm.com>