Secure Communication with Entangled Particles

Team: Fuzzy Bell

Agenda

1. Motivation
2. Bell inequality
3. Quantum experiment
4. Bell's Theorem
5. Solution

1. Motivation

In our increasingly connected world more and more **sensitive data** is being exchanged.

Therefore, it is crucial to provide **secure communication**.

Since quantum states **cannot be copied**, entanglement is a possible option to solve this problem.

Furthermore, entanglement can improve communication (e.g., superdense coding)

Squashed Entanglement

- Squashed Entanglement $E_{\mathrm{sq}} = \inf \left\{ \frac{1}{2} I(A;B|E) : \rho^{\mathrm{ABE}} \text{ extends } \rho^{\mathrm{ab}} \right\}$
- Entanglement measure
- Does not increase under LOCC
- Convex
- Additive on tensor products, ...
- Definition and bounds available in terms of entropy

$$S = -\mathrm{tr}\left(\rho \ln \rho\right)$$

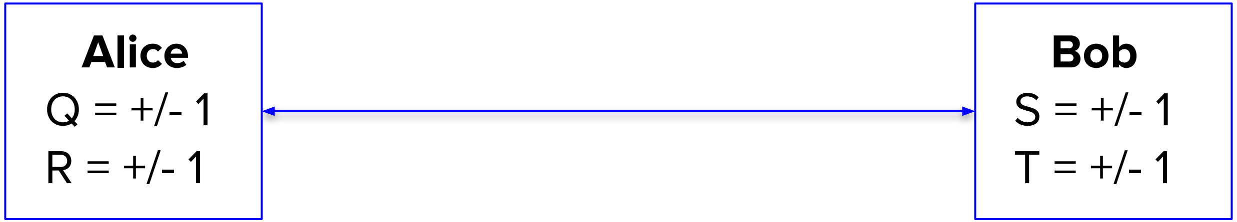
$$S = -\sum_i \lambda_i \ln \lambda_i$$

Problem with entropy

- Unknown state is given
- What is the density matrix?? Eigenvalues??
- Entropy NOT directly connected to results.

=> Find more practical measure/proof of entanglement

2. Bell inequality (experimental setup)



Alice can choose to measure either Q or R and Bob either S or T

We look at the quantity $QS + RS + RT - QT = (Q + R)S + (R - Q)T = \pm 2$

Since $R, Q = \pm 1$, either $(Q + R)S = 0$ or $(R - Q)T = 0$

2. Bell inequality (experimental setup)

Alice
$Q = \pm 1$
$R = \pm 1$

$QS + RS + RT - QT = \pm 2$

Bob
$S = \pm 1$
$T = \pm 1$

Before the measurements are performed: $Q=q$, $R=r$, $S=s$, $T=t$

$$E(QS + RS + RT - QT) = \sum p(q,r,s,t)(qs + rs + rt - qt) \leq \sum p(q,r,s,t) \times 2 = 2$$

Bell inequality

$$\boxed{E(QS) + E(RS) + E(RT) - E(QT) \leq 2}$$

3.- A quantum mechanical experiment with Charlie

Charlie prepares a quantum system of two qubits in the state:

$$|\psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

First bit to Alice

Second bit to Bob

Alice

Bob

$Q = Z_1$

$R = X_1$

$$S = \frac{-Z_2 - X_2}{\sqrt{2}} \qquad T = \frac{Z_2 - X_2}{\sqrt{2}}$$

3.- A quantum mechanical experiment with Charlie

Alice

$Q = Z_1$
$R = X_1$

Bob

$$S = \frac{-Z_2 - X_2}{\sqrt{2}} \qquad T = \frac{Z_2 - X_2}{\sqrt{2}}$$

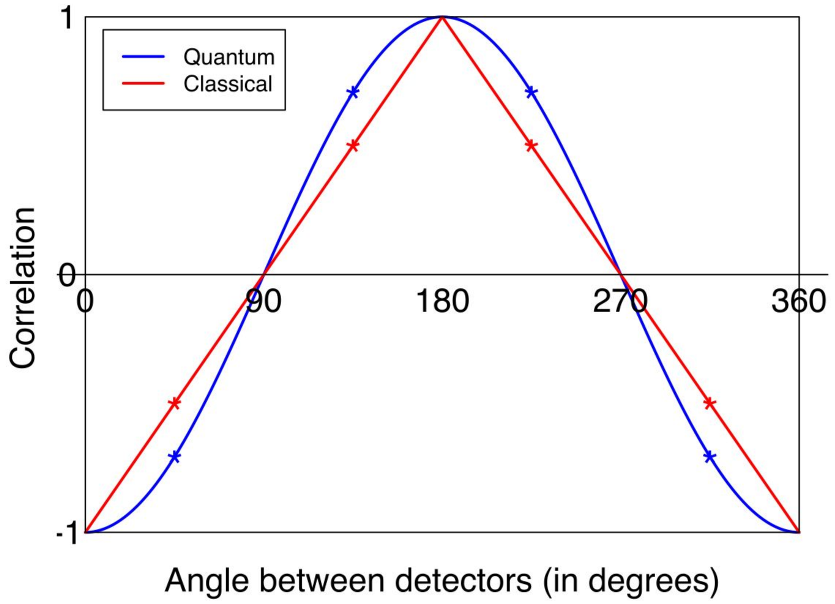
Average values: $\langle QS \rangle = \frac{1}{\sqrt{2}}$; $\langle RS \rangle = \frac{1}{\sqrt{2}}$; $\langle RT \rangle = \frac{1}{\sqrt{2}}$; $\langle QT \rangle = -\frac{1}{\sqrt{2}}$

$$\langle QS \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle = 2\sqrt{2}$$ **Quantum measurement**

4. Bell's Theorem

$E(QS) + E(RS) + E(RT) - E(QT) \leq 2$ vs. $\langle QS \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle = 2\sqrt{2}$

Bell's Theorem states that any physical theory that incorporates **local realism cannot reproduce** all the predictions of **quantum mechanical theory**.



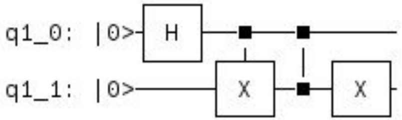
4.- Bell's Theorem

Generate Bell state 01-10

In [3]:

```
bellQubits=QuantumRegister(2)
qc=QuantumCircuit(bellQubits);
qc.h(bellQubits[0])
qc.cx(bellQubits[0],bellQubits[1])
qc.cz(bellQubits[0],bellQubits[1])
qc.x(bellQubits[1])
qc.draw()
```

Out[3]:



In [4]:

```
job=execute(qc,Aer.get_backend('statevector_simulator'))
print(job.result().get_statevector())
```

```
[ 0.         +0.j -0.70710678+0.j  0.70710678+0.j  0.         +0.j]
```

4. Bell's Theorem

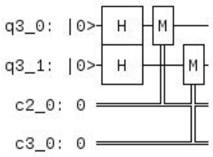
2.3562 =135 degrees

Generate random bits

In [6]:
```
# Here we generate the random control bits for A and B.
randomBit_A=ClassicalRegister(1)
randomBit_B=ClassicalRegister(1)
randomQubit=QuantumRegister(2)
qc=QuantumCircuit(randomBit_A,randomBit_B,randomQubit);


# Create the random input for A and B
qc.h(randomQubit[0])
qc.h(randomQubit[1])
qc.measure(randomQubit[0],randomBit_A)
qc.measure(randomQubit[1],randomBit_B)
qc.draw()
```

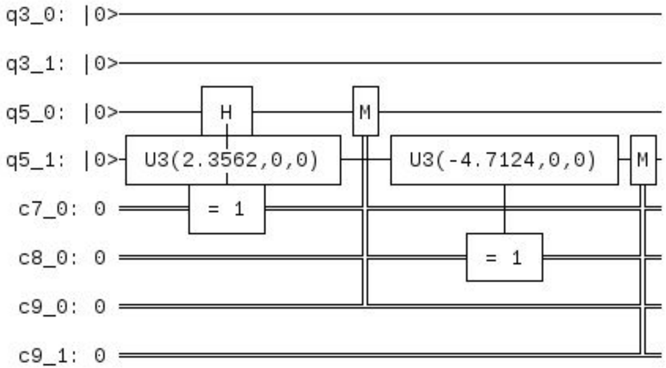
Out[6]:



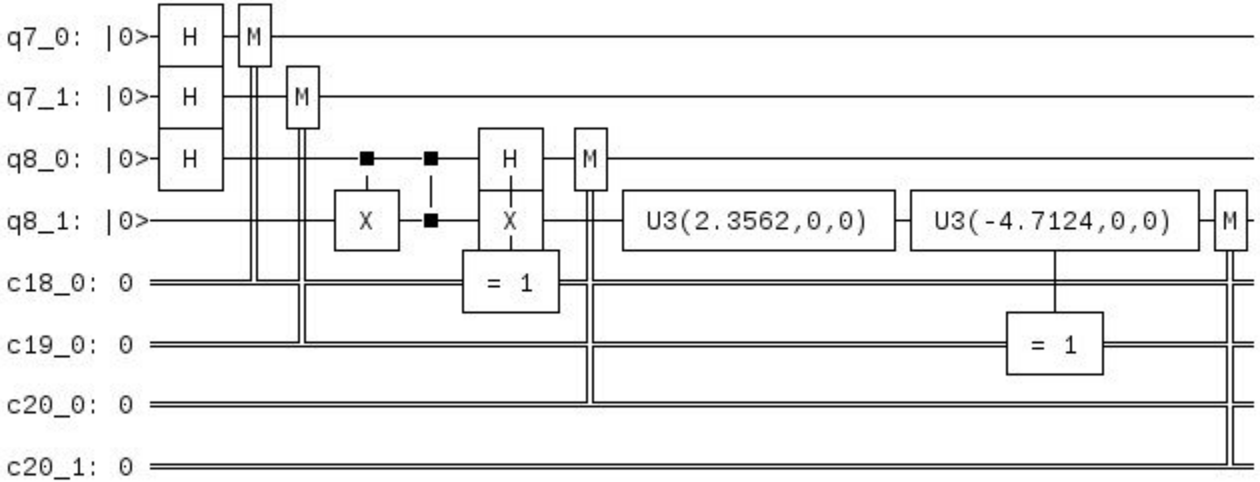
Perform random measurements

In [8]:
```
randomBit_A=ClassicalRegister(1)
randomBit_B=ClassicalRegister(1)
bellQubits=QuantumRegister(2)
resultBits=ClassicalRegister(2)
qc=QuantumCircuit(randomBit_A,randomBit_B,randomQubit,bellQubits,resultBits);
qc.h(bellQubits[0]).c_if(randomBit_A,1)
qc.u3(2.3562,0,0,bellQubits[1])
qc.u3(-2.3562*2,0,0,bellQubits[1]).c_if(randomBit_B,1)
qc.measure(bellQubits,resultBits)
qc.draw()
```

Out[8]:

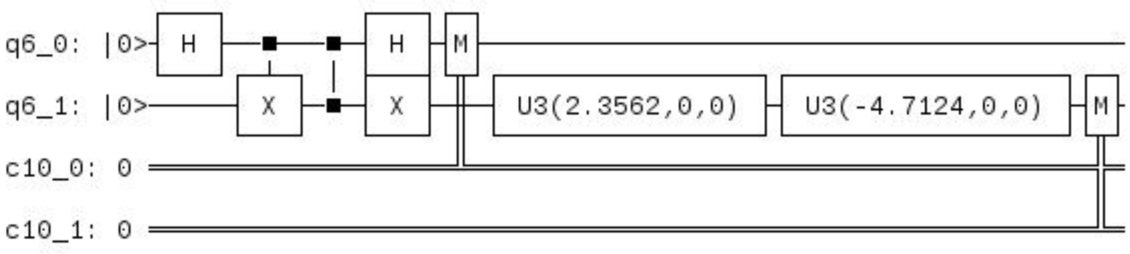


4. Bell's Theorem (Putting everything together)



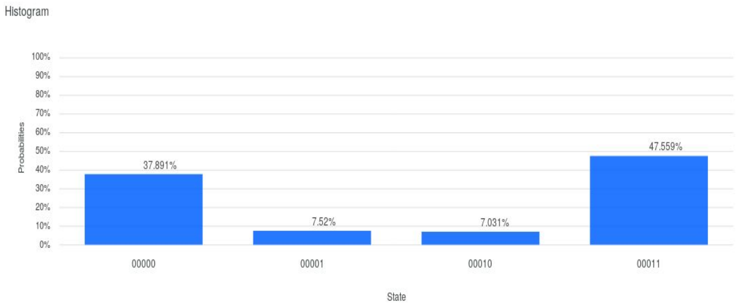
Hardware limitations: Does **not** work on ibmqx2.
Simulator OK.

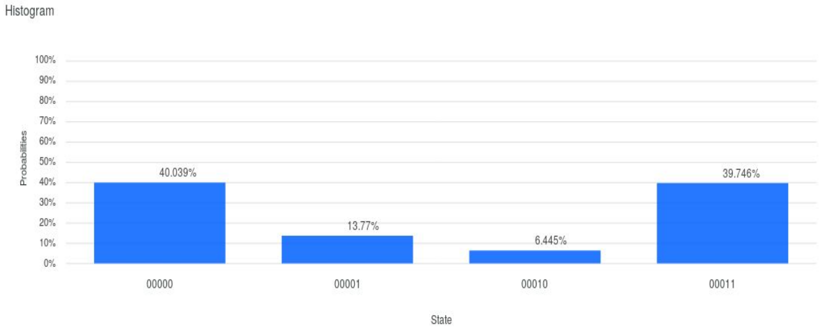
4. Simplified Circuit without randomness

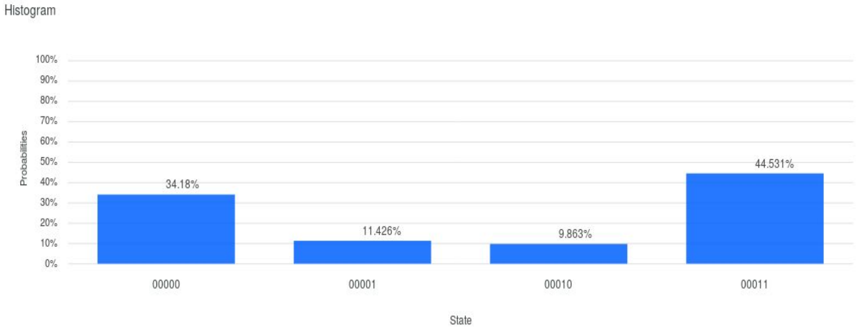


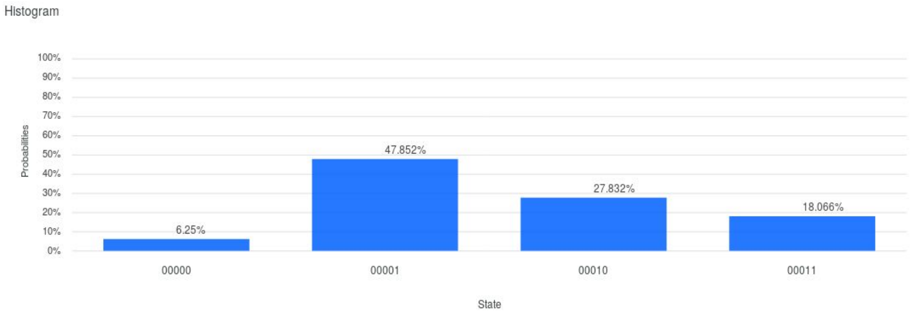
Use all 4 versions separately:
-H on/off
-U3(-4.7124,0,0) on/off

Our results (ibmqx2 hardware)









Experiment

Our result (ibmqx2 hardware)

[23]:

```
res00=[388,77,72,487]
res01=[350,117,101,456]
res10=[410,141,66,407]
res11=[64,490,285,185]

v0=(res00[0]+res00[3]-res00[1]-res00[2])/(res00[0]+res00[3]+res00[1]+res00[2])
v1=(res01[0]+res01[3]-res01[1]-res01[2])/(res01[0]+res01[3]+res01[1]+res01[2])
v2=(res10[0]+res10[3]-res10[1]-res10[2])/(res10[0]+res10[3]+res10[1]+res10[2])
v3=(res11[0]+res11[3]-res11[1]-res11[2])/(res11[0]+res11[3]+res11[1]+res11[2])

# Should be sqrt(1/2) or -sqrt(1/2).
print(v0,v1,v2,v3)
print(v0+v1+v2-v3)
```

```
0.708984375 0.57421875 0.595703125 -0.513671875
2.392578125
```

Any Questions?

Thank you
for your attention!

Sources

- https://en.wikipedia.org/wiki/Bell%27s_theorem
- https://imgur.com/gallery/DV3IJ8Y
- https://arxiv.org/pdf/quant-ph/0308088.pdf
- https://arxiv.org/pdf/1010.1750.pdf
- Quantum Computation and Quantum Information (Michael A. Nielsen and Isaac L.Chuang)

Team “Fuzzy Bell”