# Entangled photons: Simulation of Bell's inequality

March - September 2021

Institut de Ciències del Cosmos de la Universitat de Barcelona (ICCUB)

Universitat de Barcelona

Facultat de Física

Robert Benassai Dalmau

Pràctiques en empresa

Supervisor: Bruno Juliá-Díaz

# Contents

# 1 Introduction

This report contains all the information about my experience during the course *Pràctiques en empresa*, carried out between March and September 2021. My work was supervised by Dr. Bruno Juliá Díaz and assessed by the rest of the members of the project group: Martí Pedemonte Bernat, Dr. Jose M. Gómez Cama and Abel Rojo Francàs.

## 1.1 The Project

After talking to Dr. Bruno Julià about my interest in his supervision, he offered me to work either in one of Quantumlab's divulgation projects or in a group focused in building an experiment to break Bell's inequality using entangled photons. The objective of the latter was to use the experiment as a divulging and teaching tool for UB students and physics enthusiasts in events such as science fairs. This project is handled by ICCUB and it has been ongoing for around to years now. I considered it would give me extra knowledge and experience I would otherwise not acquire in the degree, so I took the offer.

The group was formed by the members I mentioned earlier. Dr. Jose M. Gómez and Martí Pedemonte were in charge of building and designing the experiment: designing the circuit to record the coincidences, gathering the necessary components and placing accurately the optical devices. My role in the group was to create a program that would allow the user to learn the basics of the experiment and would simulate the classical and quantum theories so he or she could see why the quantum theory works and the Hidden Variable Theory (HVT) does not.

## 1.2 Meetings and format of work

In order to keep up with the progress of my code and Martí's work in the experiment, all the group met up every Monday through video call (due to the pandemic) and both of us exposed our progress. On my side, I described what I had been working on the last week, the problems I had encountered and the objectives for next week's meeting. I was then given feedback and advice. In addition, every time I did an important change in the code, I merged the progress to Quantumlab's repository on Github. This way of working allowed me to have a backup online and my supervisor (and the other project members) to see the state of the program at any time.

When I joined the group, we did a couple of meetings so I could consolidate the theoretical basis of Bell's inequality. I read the two papers the project tried to replicate ([1] and [2]) and asked the doubts I had in the following meeting. Once I had enough knowledge about the experiment itself and the physics behind it I started working on the code.

My work was fully autonomous and there was no fixed schedule, so I could work without pressure and adapt the time spent and my weekly targets to my needs. I kept track of the hours spent with an Excel sheet where, every time I finished coding, I would write the amount of time I had been working.

## 1.3 Main objectives

Before starting with the work, I exposed how I imagined the program would look like and we set some objectives in order to establish an orientative path for me to follow, which would still give me freedom when coding:

1. The program has to simulate the quantum and classical (HVT) theories. The program must show and calculate the respective S in each case.
2. The user has to be able to set up the parameters of the simulation.
3. The program also has to involve some graphical widget in order to picture the basics of the experiment, ideally a simple animation.
4. It would be useful that the program could manage real data from the actual readings in the experiment and showcase them in some manner.

The order of the objectives listed above is from the most to the least important, so I first started with writing a functional code which simulated both experiments accurately and then I moved on to learning Kivy and building the actual user interface (UI). After I had a solid UI, I added some features which were not within the objectives at first, like 2D and 3D plots or a detailed description of the experiment's components.

# 2 Quantum mechanics behind Bell's inequality

Before getting into my work, a glimpse of physical context needs to be portrayed in order to better understand the whole project.

Entanglement, quantum mechanics and Bell's inequality are usually explained using the spin of a particle (usually an electron), although, as the experiment uses photons, these topics will be treated using the polarisation instead.

According to classical optics, when a linear polarizer receives a photon with a certain polarisation, the polarizer doesn't allow past any of the components which are not parallel to its axis. This translates to the intensity of the resulting beam being reduced as dictated by Malus's law:

$$I = I_0 \cos^2 \theta \tag{1}$$

Where $\theta$ is the angle between the incident polarization and the polarizer.

Quantum mechanics describes this process rather differently. It considers that the polarization is defined as a combination of the vertical ($|V\rangle$) and horizontal ($|H\rangle$) stationary states, which means that a photon will have a given wave function $\Psi$ which will be the given combination of $|H\rangle$ and $|V\rangle$ . Using the parallelism with the spin, the measurement process, instead of being a Stern-Gerlach experiment, is just the passing of a photon through a polarizer oriented in a given direction. This will cause the wave function to essentially collapse in the vertical or horizontal states. In the vertical state, the photon will pass through, while, if the polarization is horizontal, it will not. Each case has a given probability to happen given by $|\langle V(H)|\,|\Psi\rangle|^2$.

As with the electrons, if two or more photons are considered, an entangled state can exist. In the experiment, the entangled photons generated have the following wave function:

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|V\rangle_s |V\rangle_i + |H\rangle_s |H\rangle_i) \tag{2}$$

Where the subindexes $i$ and $s$ mean idler and signal respectively, which are just two terms to describe the two resulting photons from the downconversion of the original (pump) photon. In this case, this state cannot be factored into a simple product of signal and idler states. This means that the state of one particle cannot be specified without making reference to the other particle. If one of the photons is detected to be vertical or horizontal, the other one will have the same polarization as the first one. This is a result classical mechanics can't predict.

# 3 The Hidden Variable Theory (HVT).

## 3.1 An overview

A hidden variable theory is a proposal to explain a quantum behaviour in a deterministic, thus classical, way. These kind of theories state that the aleatory nature of quantum mechanics is due to the lack of information of a certain variable, which doesn't allow the observer to predict with full certainty the outcome of a measurement on a given particle.

In 1964, Bell discovered that quantum mechanics and local hidden variable theories would never be in agreement. He stated an inequality which HVTs must follow but quantum mechanics violate. This meant that the debate between the deterministic and non-deterministic sides would end once and for all if the inequality was broken. In our experiment, the objective is to break this inequality and prove, like John Clauser and Stuart Freedman did in 1972, that quantum mechanics is not incomplete and that local HVTs are not correct.

### 3.2 The considered HVT

To replicate Mitchell's paper [2], the same HVT has been used. It is simple but it agrees pretty well with the quantum prediction, as shown in figure 7. The considered HVT allows the photon through (equivalent to the vertical quantum state) following the probability:

$$P_V(\gamma, \lambda) = \begin{cases} 1 & |\gamma - \lambda| \leq \pi/4 \\ 1 & |\gamma - \lambda| \geq 3\pi/4 \\ 0 & otherwise \end{cases} \tag{3}$$

Where $\lambda$ and $\gamma$ are the polarization and polarizer angles respectively. In essence, when a photon meets a polarizer set to an angle $\gamma$, it will always register as $V_\gamma$ if $\lambda$ is closer to $\gamma$ than to $\gamma + \pi/2$.

Note that this is indeed a hidden variable theory. One pair of photons will have the same polarization angle $\lambda$, but this will change in an unknown way (could be random or yet to be discovered) in the next pair, as the polarization in the incident beam is circular. In that sense, the polarization angle is a hidden variable, it is not known beforehand. This theory is also local, as the outcome of the measurement is known at the same place it occurs.

## 4 The experiment

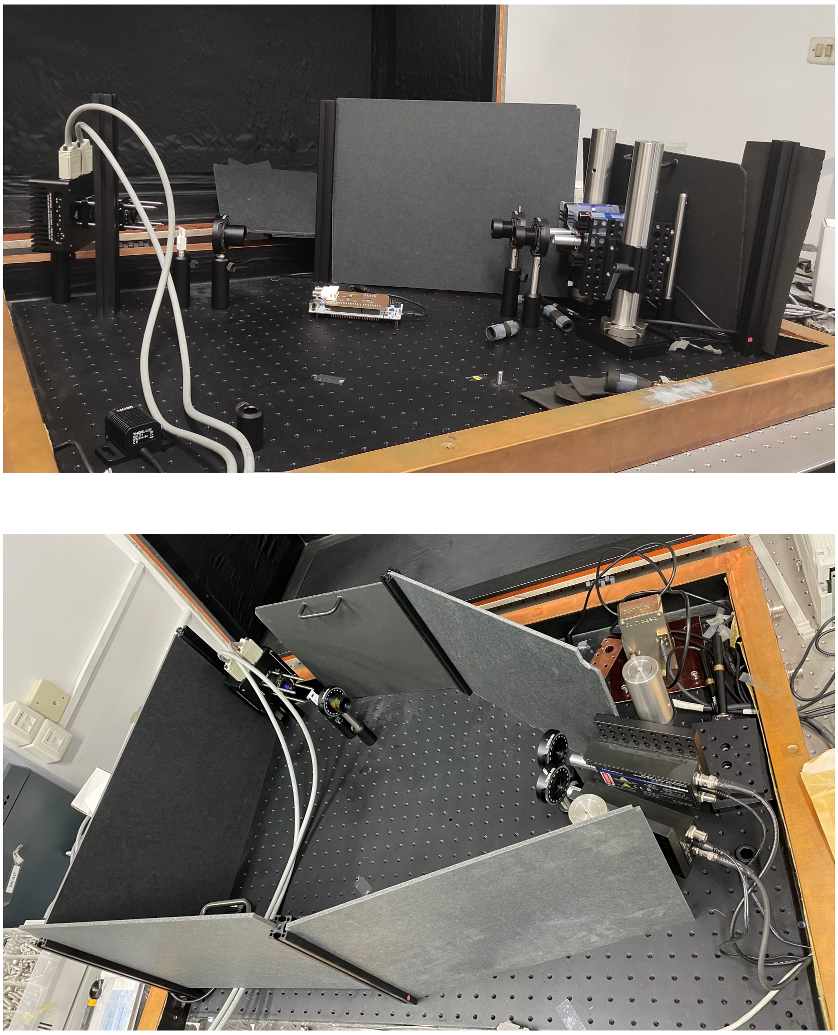


***Figure 1:*** *Pictures of the experiment taken at the laboratory.*

## 4.1 The process of creating a pair of entangled photons

In this section, an idea of how the experiment works will be presented.

Firstly a diode laser produces violet photons which are passed through a blue filter, a linear polarizer and then a quartz birefringent plate whichs gives the beam's polarization a certain phase. Then, the photons encounter two beta barium borate (BBO) crystals which are in charge of downconverting an original photon (pump) into the two less energetic ones (idler and signal) by means of a type I spontaneous parametric downconversion (SPD). Each crystal can only support downconversion of one pump polarization, the other one passes through unchanged. That is why there are two BBO crystals, one rotated $90^o$ respect from the other.

As the two resulting photons come from the same pump parent, their combined energies and momentum must match the original ones. In addition, they are produced barely at the same time and have the same polarization as well. By an appropriate placement of the detectors, these can measure photons with the same wavelength, which doubles the parent one. Knowing this, an entangled pair of photons will fire the detectors almost at the same time with the same energy.

Before the detectors, there is one polarizer in front of each of the single-photon counting modules (SPCMs), which is in charge of making a measurement of the corresponding photon's polarization. A computer counts the coincident (two simultaneous detections) and single firings to then compute the S to check Bell's Inequality.

## 4.2 Expected quantum probabilities

Let's consider the two measuring polarizers, with axis angles respect to the vertical direction $\alpha$ and $\beta$, and an entangled state with the wave function shown in equation (2). For a polarizer rotated by a certain angle $\alpha$, we can change to the rotated basis using:

$$|V_\alpha\rangle = \cos\alpha\,|V\rangle - \sin\alpha\,|H\rangle \tag{4}$$

$$|H_\alpha\rangle = \sin\alpha\,|V\rangle + \cos\alpha\,|H\rangle \tag{5}$$

Adding the polarization angle of the linear polarizer prior to the downconversion ($\theta_l$) and the given phase added by the birefringent quartz plate ($\phi$), the wave function looks like:

$$|\psi_{DC}\rangle = \cos\theta_l\,|H\rangle_s\,|H\rangle_i + \exp{(i\phi)}\sin\theta_l\,|V\rangle_s\,|V\rangle_i \tag{6}$$

The different probabilities of getting vertical-vertical (VV), horizontal-vertical (VH or HV), or horizontal-horizontal (HH) results in the measurements are the following:

$$P_{VV}(\alpha,\beta) = \left|\langle V_\alpha|_s\,\langle V_\beta|_i\,|\psi_{DC}\rangle\right|^2 = \sin^2\alpha\sin^2\beta\cos\theta_l + \cos^2\alpha\cos^2\beta\sin^2\theta_l + \frac{1}{4}\sin 2\alpha\sin 2\beta\sin 2\theta_l\cos\phi \tag{7}$$

$$\begin{aligned} P_{VH}(\alpha,\beta) = \left|\langle V_\alpha|_s\,\langle H_\beta|_i\,|\psi_{DC}\rangle\right|^2 = |-\sin\alpha\cos\beta\cos\theta_l + \exp{[i\phi]}\cos\alpha\sin\beta\sin\theta_l|^2 = \\ \sin^2\alpha\cos^2\beta\cos^2\theta_l + \cos^2\alpha\sin^2\beta\sin^2\theta_l - \frac{1}{4}\sin 2\alpha\sin 2\beta\sin 2\theta_l\cos\phi \end{aligned} \tag{8}$$

Where $P_{HV}$ is the same as $P_{VH}$ changing $\beta$ for $\alpha$ and $P_{HH}$ is not of interest because the horizontal photons will not be detected.

### 4.3 The calculation of Bell's inequality

Using photons instead of electrons to produce the entanglement implies that Bell's inequality has the following form for classical systems:

$$|S| \leq 2 \tag{9}$$

Where S is calculated using:

$$S \equiv E(a,b) - E(a,b') + E(a',b) + E(a',b') \tag{10}$$

Here $a$, $a'$, b and $b'$ are four different polarizer angles. Each $E(\alpha,\beta)$ contains four $N(\alpha,\beta)$ (where $\alpha = a$ or $a'$ and $\beta = b$ or $b'$) which are the amount of coincidences the experiment detects for each pair of angles:

$$E(\alpha,\beta) = \frac{N(\alpha,\beta) + N(\alpha_\perp,\beta_\perp) - N(\alpha,\beta_\perp) - N(\alpha_\perp,\beta)}{N(\alpha,\beta) + N(\alpha_\perp,\beta_\perp) + N(\alpha,\beta_\perp) + N(\alpha_\perp,\beta)} \tag{11}$$

Note that $\alpha_\perp$ and $\beta_\perp$ correspond to $\alpha + 90°$ and $\beta + 90°$ respectively. Equations (10) and (11) imply that, to determine $S$, sixteen different measurements need to be taken.

### 4.4 Calculating the error of S

According to the mentioned paper [1], The error from S corresponds to the standard deviation:

$$\sigma_S = \sqrt{\sum_{i=1}^{16} N_i \left(\frac{\partial S}{\partial N_i}\right)^2} \tag{12}$$

despite the amount of partial derivatives to be calculated (16), note that the partial derivatives done respect to the **positive** $N_i$ in the numerator in equation (11) have the following form:

$$\frac{\partial E}{\partial N_+} = \frac{2[N(\alpha,\beta_\perp) + N(\alpha_\perp,\beta)]}{(N(\alpha,\beta) + N(\alpha_\perp,\beta_\perp) + N(\alpha,\beta_\perp) + N(\alpha_\perp,\beta))^2} \tag{13}$$

while the partial derivatives respect to the $N_i$ which are **subtracting** in the numerator are as follows:

$$\frac{\partial E}{\partial N_-} = \frac{-2(N(\alpha,\beta) + N(\alpha_\perp,\beta_\perp))}{(N(\alpha,\beta) + N(\alpha_\perp,\beta_\perp) + N(\alpha,\beta_\perp) + N(\alpha_\perp,\beta))^2} \tag{14}$$

Using these properties, the calculus of the error in S is much simpler, because using just two formulas, the sixteen partial derivatives can be obtained. I noticed this characteristic when I had to calculate the error in the program and it simplified the code to a couple of simple loops.

## 5 The Program

Although my job was also to find the probabilities, the S and its error, without a doubt, the hardest and most challenging part of the work I have done is building and coding the actual program.

## 5.1 Structure

The program is structured in four main parts:

1. Quantum experiment
2. HVT experiment
3. Informative screen
4. Real data loading

Which are organised into three separate tabs. At the top of the screen, as shown in figure 2, there are the two polarizers in the left (which rotate according to the angles selected by the user) and a model of the experiment created in OneShape3D[1]. When the experiment is ran, a small animation triggers, representing a blue laser being split into two red ones which then pass through the polarizers and reach the detectors.

## 5.2 Quantum experiment

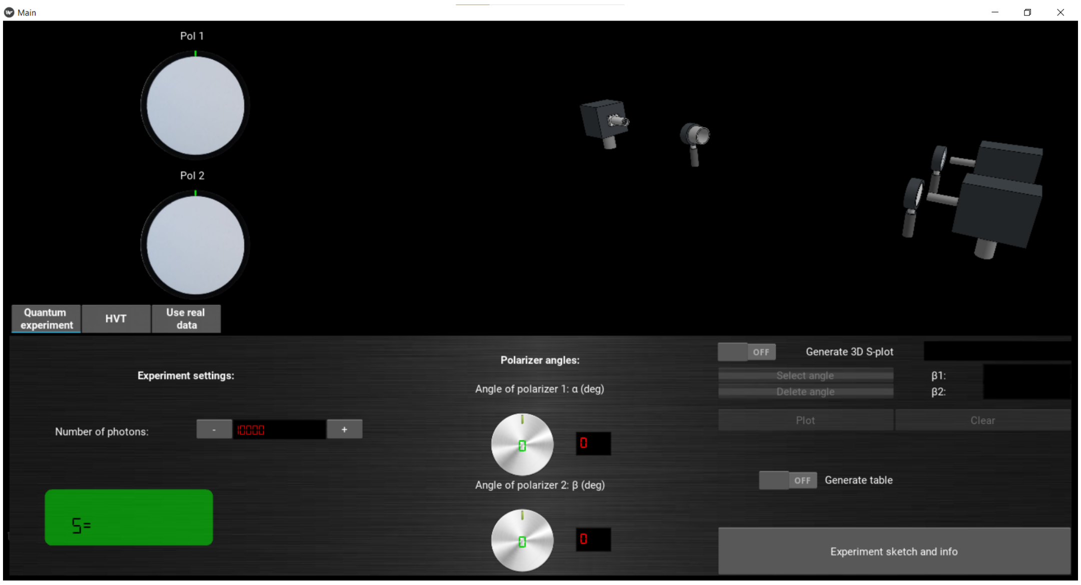


***Figure 2:*** *Quantum tab*

The "quantum experiment" tab allows the user to simulate the real experiment. The program uses the probabilities calculated in equations (7)-(8) and the input angles selected by the user to calculate the S and its error according to equations (10) and (12). The user can also input the number of photons thrown in order to increase the precision in the calculation of the S.

On the right side of the screen there are the graph settings, which, when activated, generate a 3D plot of the S. The user has to input the start and finish angles $(\beta_1, \beta_2)$ of the second polarizer, while the angle of the first one $(\alpha)$ will swipe through the 0 to 360° range, creating an $(\alpha, \beta, S)$ 3D coordinate space where $\alpha$ and $\beta$ are the angles of the first and second polarizers respectively.

[1] This is an online free program for professional 3D modelling and design. I had to learn the basics to achieve a simple model of the experiment.

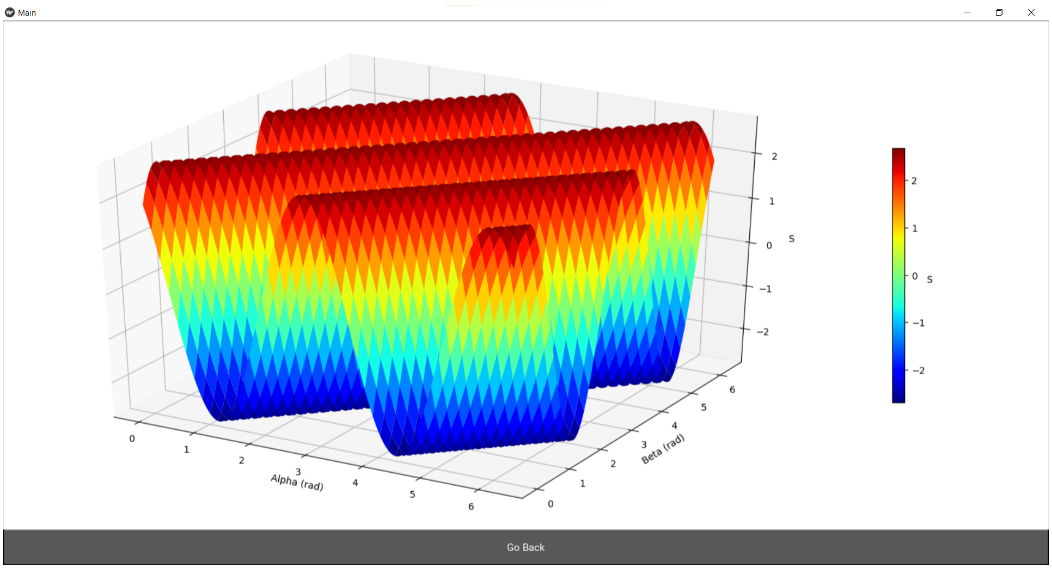
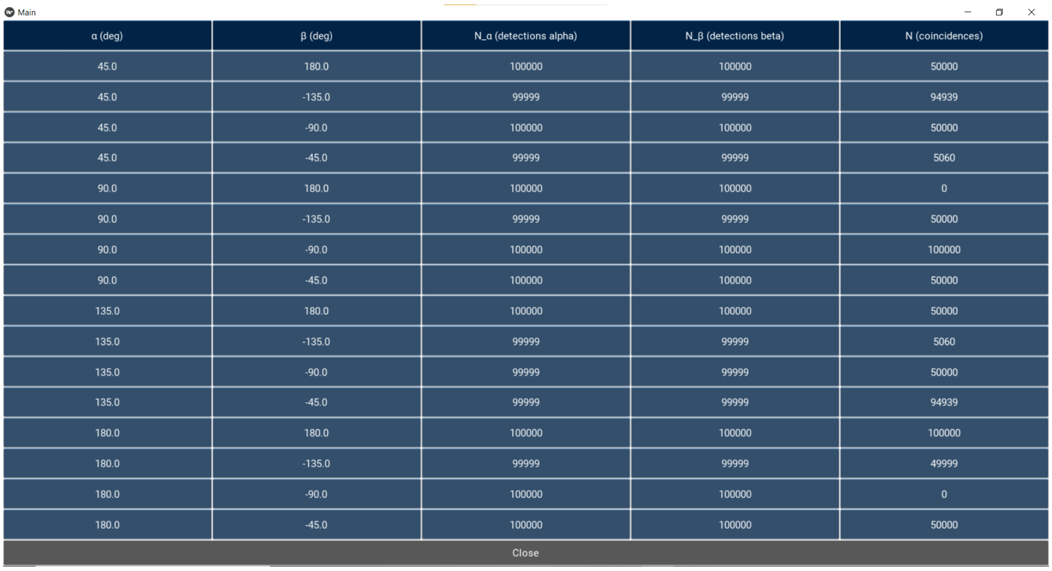


***Figure 3:*** *3D plot of the S. Input angles used: 0 and 360.*

In figure 3 the sinusoidal periodicity of the S value can be observed. In addition to this, it can be clearly seen that it exceeds the classical limit ($|S| \leq 2$).

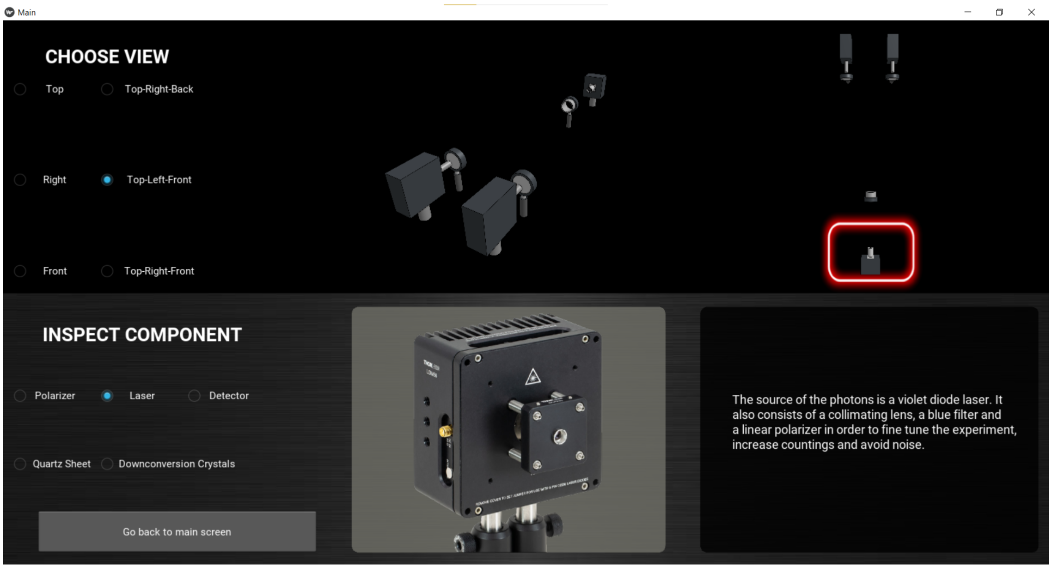
Just below the graph settings in figure 2, the user can also generate a table showing the number of coincidences and single detections for each of the sixteen angles needed to calculate the last $S$.



| α (deg) | β (deg) | N_α (detections alpha) | N_β (detections beta) | N (coincidences) |
|---|---|---|---|---|
| 45.0 | 180.0 | 100000 | 100000 | 50000 |
| 45.0 | -135.0 | 99999 | 99999 | 94939 |
| 45.0 | -90.0 | 100000 | 100000 | 50000 |
| 45.0 | -45.0 | 99999 | 99999 | 5060 |
| 90.0 | 180.0 | 100000 | 100000 | 0 |
| 90.0 | -135.0 | 99999 | 99999 | 50000 |
| 90.0 | -90.0 | 100000 | 100000 | 100000 |
| 90.0 | -45.0 | 100000 | 100000 | 50000 |
| 135.0 | 180.0 | 100000 | 100000 | 50000 |
| 135.0 | -135.0 | 99999 | 99999 | 5060 |
| 135.0 | -90.0 | 99999 | 99999 | 50000 |
| 135.0 | -45.0 | 99999 | 99999 | 94939 |
| 180.0 | 180.0 | 100000 | 100000 | 100000 |
| 180.0 | -135.0 | 99999 | 99999 | 49999 |
| 180.0 | -90.0 | 100000 | 100000 | 0 |
| 180.0 | -45.0 | 100000 | 100000 | 50000 |

***Figure 4:*** *Table for angles* $\alpha = 45°$ *(pol 1) and* $\beta = 180°$ *(pol 2).*

## 5.3 Informative Screen



○ Top ○ Top-Right-Back

○ Right ● Top-Left-Front

○ Front ○ Top-Right-Front

○ Polarizer ● Laser ○ Detector

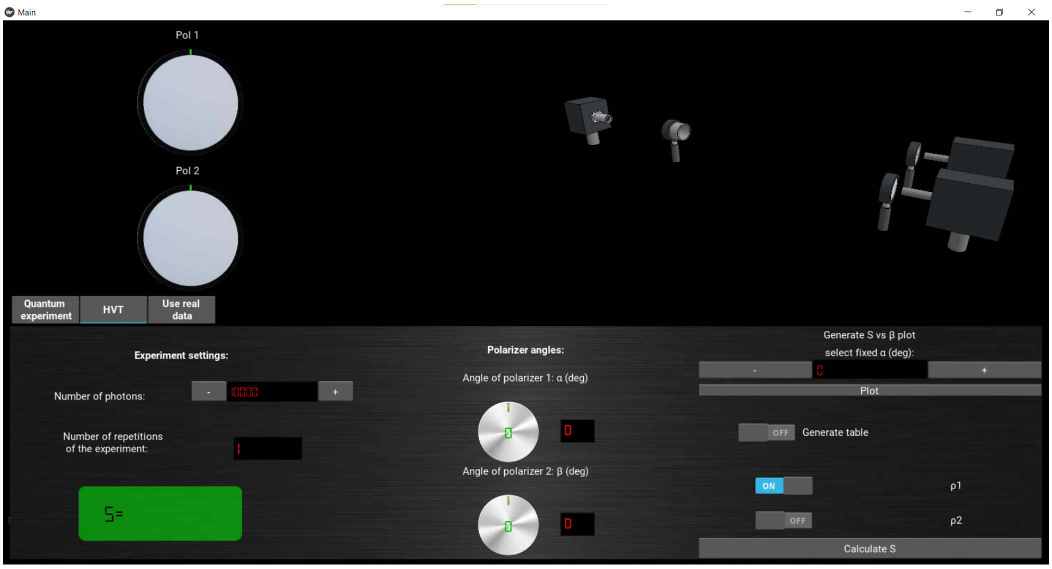
○ Quartz Sheet ○ Downconversion Crystals

The source of the photons is a violet diode laser. It also consists of a collimating lens, a blue filter and a linear polarizer in order to fine tune the experiment, increase countings and avoid noise.

***Figure 5:*** *Informative Screen.*

This part of the program is accessed by pressing the “Info Screen” button at the bottom right corner of the quantum tab (see figure 2). Here, the user can, at the top half of the screen, select different angles from which to see the 3D model and, at the bottom part, choose amongst the most relevant components of the real experiment. When a part is selected, a picture of it and its description will appear in the grey and black squares respectively (see Figure 5: selected component: laser, picture and description). The red frame at the top right of the screen will move towards the selected item on the top view of the model in order to let the user know where each part is located in the actual experiment. The role of this screen is to give the user a basic but more in-depth understanding, component-wise, of the real project.

## 5.4 HVT tab



Number of photons: 10000

Number of repetitions of the experiment: 1

S=

Angle of polarizer 1: α (deg) 0

Angle of polarizer 2: β (deg) 0

select fixed α (deg): 0

OFF Generate table

ON ρ1

OFF ρ2

***Figure 6:*** *HVT tab.*

In this tab, the user can repeat the experiment using the HVT theory described in equation (3) or using a probability which agrees with Manus' law and is proportional to $\cos^2 \alpha$. In order to calculate the error of S, the user must change the "number of repetitions of the experiment". If a number greater than 1 is selected, the program will compute the standard deviation and show it alongside the S in the green rectangle.

The only difference between this tab and the quantum one in terms of functionality is that the graph is now two dimensional instead of 3D because the computing time of the calculation of the S in the classical experiment is greater than in the case before.

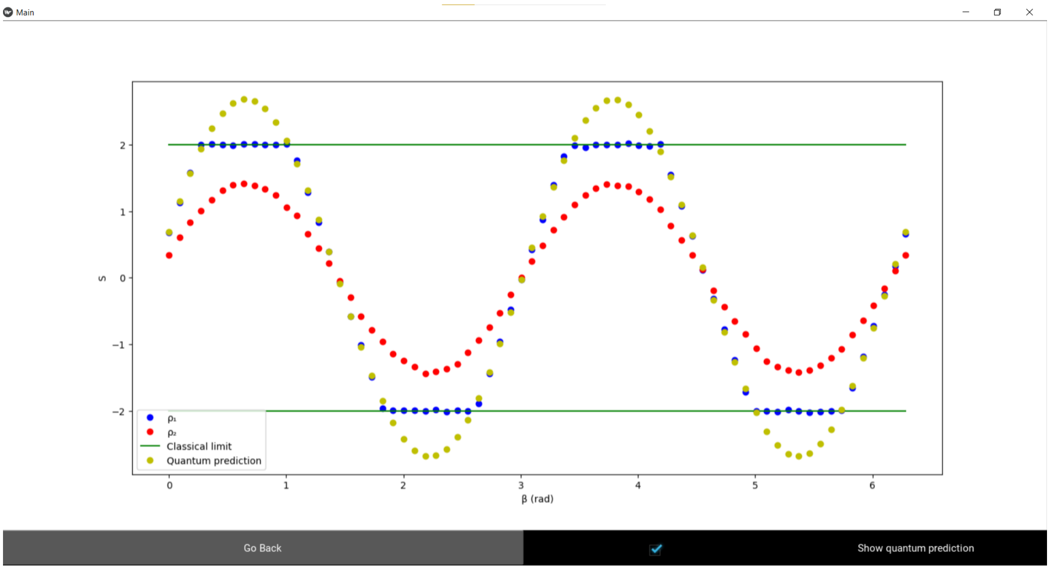
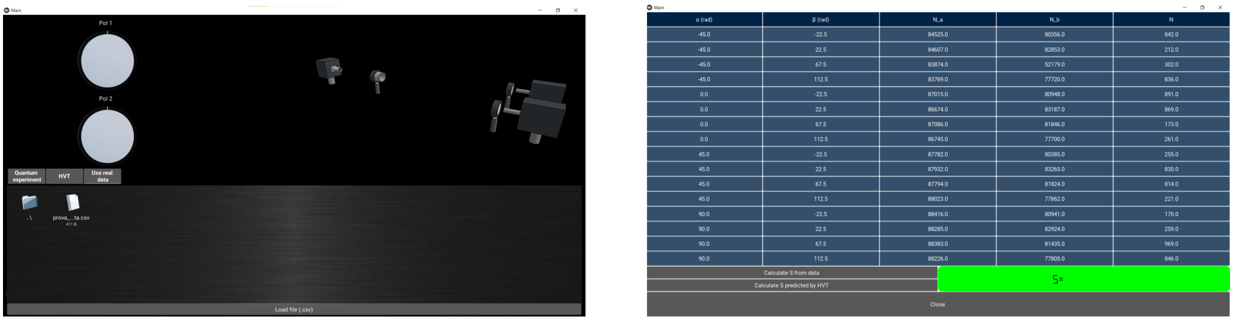


***Figure 7:*** *S in function of $\beta$ using the two available probabilities (HVT and Manus' law), showing the quantum prediction (yellow dots) and classical limits (green lines).*

Once the user has selected a fixed $\alpha$ for the first polarizer, the graph is plotted using the probability selected. Inside the plot, the quantum prediction can be shown selecting the bottom right checkbox. This graph allows the user to see the differences between the HVT and quantum theories clearly. It is notable that the theory differs clearly in the quantum region of the $S$ ($|S| > 2$) but agrees well with the quantum prediction inside the classical boundaries ($|S| < 2$).

## 5.5 Data Screen



***(a)*** *Navigation tab*

| α (rad) | β (rad) | N_a | N_b | N |
|---|---|---|---|---|
| -45.0 | -22.5 | 84525.0 | 80356.0 | 842.0 |
| -45.0 | 22.5 | 84607.0 | 82853.0 | 212.0 |
| -45.0 | 67.5 | 83874.0 | 52179.0 | 302.0 |
| -45.0 | 112.5 | 83769.0 | 77720.0 | 836.0 |
| 0.0 | -22.5 | 87015.0 | 80948.0 | 891.0 |
| 0.0 | 22.5 | 86674.0 | 83187.0 | 869.0 |
| 0.0 | 67.5 | 87086.0 | 81846.0 | 173.0 |
| 0.0 | 112.5 | 86745.0 | 77700.0 | 261.0 |
| 45.0 | -22.5 | 87782.0 | 80385.0 | 255.0 |
| 45.0 | 22.5 | 87932.0 | 83265.0 | 830.0 |
| 45.0 | 67.5 | 87794.0 | 81824.0 | 814.0 |
| 45.0 | 112.5 | 88023.0 | 77862.0 | 221.0 |
| 90.0 | -22.5 | 88416.0 | 80941.0 | 170.0 |
| 90.0 | 22.5 | 88285.0 | 82924.0 | 259.0 |
| 90.0 | 67.5 | 88383.0 | 81435.0 | 969.0 |
| 90.0 | 112.5 | 88226.0 | 77805.0 | 846.0 |

***(b)*** *Table of the selected data*

***Figure 8***

In this tab the user will be able to navigate inside the computer folders and use a *.csv* file with the results of a real measure done by the actual experiment. Note that **only** *.csv* files can be selected in order to avoid undesired errors. When the file is selected, a table similar to the one shown in Figure 4 will appear. In addition to this, the user will be able to calculate the S from the real data and compare it to the one predicted by HVT using the same angles from the file.

# 6 Code overview

The goal of this section is not to fully describe the code, but to give an idea of the main structure of it. The detailed description of the functions created would not give any useful insight to understand how the program works apart from the amount of work which went into the development of it. The language used in the project was Python and a library called Kivy. The latter is mostly used to make simple computer and cellphone apps, and is especially useful and practical in order to create a simple but solid user interface (UI).

The program runs on three different files:

1. **The Kivy file** (*entangled.kv*): this file is in charge of building the "structure" of the graphics and putting the different widgets in the correct place in the screen. It essentially generates the UI creating a certain type of layout (grid-like for example) where to locate the different widgets (buttons, knobs, images, animations, etc.) also defined in this file.

2. **The main file** (*entangled.py*): where the functions of the different screens are defined and the program is actually ran. It is more like the main stem of the program because it connects every file with one another. The "status" of the widgets defined in the Kivy file can be read, used and modified. For example, if the state of a button is "pressed" the program can run a function to make something happen on screen.

3. **The experiment file** (*entangledexp.py*): this python file is similar to a library. It will be imported in the main file where the calculations are done. It stores the functions to compute the S and its errors in the HVT and quantum tabs. This file contains every function responsible of simulating the experiment mathematically, using the expressions described in section 4, and it is mostly used when the variables are set and the experiment is ran.

## 6.1 Main file structure

The main file is divided into five important classes. I have used *ScreenManager* which is a Kivy tool that allows the program to move around many screens. For example, when the "info screen" button is pressed, *ScreenManager* sets the active screen to *InfoScreen.* This allowed the different classes to use functions between each other.

Five screen classes were created:

1. *EntangledScreen*: Contains all the functions that operate in the main screen. Includes the "quantum", "HVT", and "real data" tabs.

2. *InfoScreen*: This is the class which has all the information about the components and controls the informative screen. The information is read from a text file and stored in a dictionary, where the keys are the component's index.

3. *GraphScreen*: In this screen the 2D HVT and 3D quantum plots are shown. The class manages the creation of the canvas for the graph and the graph itself using a Kivy extension which emulates Matplotlib.

4. *TableScreen*: This is simply a grid layout where the numbers from the calculation of the S are shown as a table (See figure 4).

5. *DataScreen*: This screen is effectively almost the same as *TableScreen* using the data from a real experiment.

### 6.2 “Library file”: Main functions

In this file, as I mentioned earlier, the functions related to the calculations of the S are stored so they can be used when imported by the main file.

The quantum S is the most simple to calculate. The function takes as input the number of photons to be considered and the angles of the two polarizers (all of the variables are given by the user). Then, to determine the amount of VV, VH, HV and HH pairs of photons, the program just multiplies the probability of each event given by equations (7) and (8) times the total amount of photons. Then the results are put in a list. After this, another function gets that list and calculates the S and its error using equations (10) to (14).

The classical S using HVT is more interesting though (from a physics point of view at least). Here, the actual experiment is done. The function generates an amount of photons specified by the user with random polarizations and then gets the probability for each one to either pass through the polarizer or not using equation (3). Note that there is no conceptual difference between the simulation and the real experiment here (if the HVT was to be correct). A random number (polarization) is filtered through a function (polarizer) and a reading (measurement) is done. After this process, the results are passed to another function that computes the S and its error. The latter is calculated when the experiment is set to be repeated more than once. In this case, the standard deviation will be computed and shown besides the S on screen.

The computing time is a very important factor here. In the quantum function, there are no loops (it just consists of a few multiplications), but in the HVT function, the actual experiment is repeated $4N_{reps} \times N_{photons}$. This is because the program has to generate a photon and then pass it through the polarizers with angles $\alpha$ and $\beta$ $N_{photons}$ times. Then it has to repeat this process for the 4 different angles required in equations (10) and (11). Last but not least, to compute the error of S (standard deviation), this process needs to be carried out $N_{reps}$ more times. As this is a function that requires multiple iterations, it takes more time to run than its quantum counterpart.

The main problem here was that, if no further actions were taken, the screen would freeze and be unresponsive until all the iterations were completed. This is not a big problem if the function is ran just once, but, when the “HVT plot” is required, the computing time grows considerably to around fifteen to twenty seconds, as the S needs to be computed nearly 100 times. To have the program unresponsive for so long would be detrimental to the user’s experience, so, to avoid long waiting times, a multiprocessing tool called *ProcessPoolExecutor* from python has been used. This allows the program to still be able to function while the graph is being calculated and plotted. In addition to this, an informative message is shown on screen to let the user know that the process is ongoing.

## 7 Development proposals

The objectives set at the beginning of the project have been achieved and I am very satisfied with the final version of the program considering the limited time I disposed of to learn the language and produce the code. The tool created will help to better understand and picture Bell’s inequality and the experiment carried out. As it stands, the current program limits to the simulation and explanation of the experiment, but it could be extended and developed in a way that it is another part of the experiment itself.

In one of the meetings during the project’s development, the idea of making the experiment interactive by means of the code I was developing was proposed. The plan would have been to use a Raspberry Pi, which is a tiny affordable computer, to rotate the polarizers and set the parameters, essentially automatising the whole experiment. At that time, it would have been an impossible undertaking because the experiment was not fully developed yet and I did not have the knowledge nor time to develop such a demanding task. To add to the problems, extra hardware would have been needed to rotate the polarizers and the idea was discarded.

Now that the program is complete and solid, such an undertaking, planned and discussed properly, would be more plausible for another student to tackle, and it would be an important evolution to the existing program.

## 8 Main challenges and conclusions

The main challenge I faced in this project was the lack of knowledge in object oriented programming and the Kivy library, which is very extense and intricate because has its own language. During my work I was fully autodidact, and I progressively learned from how to put a button in a black screen to making a fully functional program.

To be honest, this training has been very enriching to me. On the one hand, I have learned to code more professionally and rigorously and I have dug into object oriented programming. It is important to consider that I already knew the basics of Python from the “Informàtica” subject in the degree, but it was mainly focused in solving problems and working with files instead of using classes and defining objects.

This project also evidenced how important programming efficiency is when large numbers of iterations are taken into consideration. A well optimised program will run fast and fluently, and I am grateful to have had to learn how to utilise the threading and multiprocessing tools. All in all, it has improved my coding style and understanding of how a computer deals with different workload demandings.

On the other hand I have progressively learned how to use Kivy, which was frustrating at first but fulfilling in the end. After completing this training, I have acquired the essential skills to be able to develop a user interface in this platform. This effectively enlarges my skill set which can be unexpectedly useful in a world where apps are crucial for companies.

In the physics point of view, I have also discovered more deeply Bell’s inequality and photon entanglement, which was what dragged me into the project at first. This has fulfilled my initial needs to learn more physics by doing this project.

In conclusion, all the objectives initially set have been accomplished and more features have been integrated to the program. In addition to that, I am proud of having overcome all the setbacks and having worked around some structural inconveniences (which there are) in the Kivy language, which required a deeper understanding on how the library operates. All in all, I have extended my knowledge and CV, which is always helpful, and the topic has ignited my interest in quantum mechanics.

## References

[1] Entangled photon apparatus for the undergraduate laboratory, Dietrich Dehlinger and M. W. Mitchell, June 2002

[2] Entangled photons, nonlocality and Bell inequalities in the undergraduate laboratory. Dietrich Dehlinger, M. W. Mitchell, May 2002.