

Tarea 4 Tecnologías Cuánticas

Juan Artigas, Antonia Dias, Lukas Wolff, Patricio Palacios, Manuel Tagle, Benjamin Tapia

1. Scripts de Bob y Alice

Se han desarrollado dos scripts, `alice_choices.py` y `bob_choices.py`, que generan las elecciones de bases para Alice y Bob, respectivamente. Estos scripts seleccionan específicamente los ángulos necesarios para el test CHSH: 0° y 45° para Alice, y $22,5^\circ$ y $-22,5^\circ$ para Bob. Para asegurar la reproducibilidad de los resultados, se utilizan semillas distintas: 1001 para Alice y 2002 para Bob.

Los archivos funcionan de la siguiente manera:

2. Script del Árbitro

El árbitro ejecuta el script `referee_quantum.py`, que implementa el test de Bell usando el criterio CHSH (Clauser-Horne-Shimony-Holt). El proceso se desarrolla de la siguiente manera:

2.1. Generación de Resultados Cuánticos

Para cada par de mediciones con ángulos θ_A (Alice) y θ_B (Bob), el script simula un estado entrelazado Bell $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ mediante la función `sample_joint()`, que:

1. Calcula la correlación teórica: $E = \cos(2(\theta_A - \theta_B))$
2. Define las probabilidades cuánticas para los cuatro resultados posibles:

$$P(+1, +1) = P(-1, -1) = \frac{1 + E}{4} \quad (1)$$

$$P(+1, -1) = P(-1, +1) = \frac{1 - E}{4} \quad (2)$$

3. Muestrea aleatoriamente un resultado según estas probabilidades

2.2. Cálculo de Correlaciones

El script calcula cuatro correlaciones específicas $E(a, b)$ para las combinaciones de bases del test CHSH:

- $E(a, b)$: bases $a = 0^\circ$ y $b = 22,5^\circ$
- $E(a, b')$: bases $a = 0^\circ$ y $b' = -22,5^\circ$
- $E(a', b)$: bases $a' = 45^\circ$ y $b = 22,5^\circ$
- $E(a', b')$: bases $a' = 45^\circ$ y $b' = -22,5^\circ$

Para cada combinación de bases, la correlación se calcula como:

$$E(a, b) = \langle A_k \cdot B_k \rangle = \frac{1}{N} \sum_{k=1}^N A_k \cdot B_k$$

donde A_k y B_k son los resultados (± 1) de Alice y Bob para el par k , y N es el número de mediciones para esa combinación específica de bases.

2.3. Parámetro CHSH

El parámetro CHSH se calcula como:

$$S = E(a, b) + E(a, b') + E(a', b) - E(a', b')$$

Interpretación del parámetro CHSH:

- $|S| \leq 2$: Compatible con teorías de variables ocultas locales (límite clásico)
- $2 < |S| \leq 2\sqrt{2} \approx 2,828$: Violación de desigualdades de Bell, evidencia de entrelazamiento cuántico
- $|S| > 2\sqrt{2}$: Imposible según la mecánica cuántica (límite de Tsirelson)

3. Resultados

3.1. Bases Iniciales

Sobre las bases iniciales de Alice y Bob, se obtiene una correlación de 2.77 lo que indica que se cumple el límite inferior de 2 y el superior de 2.828, lo que indica que tales bases se encuentran entrelazadas.

A continuación se realizó el análisis sobre un conjunto de bases, donde se estudiara cómo se ven afectadas las correlaciones y el parámetro CHSH.

4. Descripción detallada del programa

Implementamos la simulación de correlaciones cuánticas asociadas al estado de Bell Φ^+ mediante tres módulos fundamentales: generación de bases locales (Alice y Bob) y cálculo de correlaciones por parte de un árbitro. El código se desarrolló en Python utilizando `numpy` y `pandas` para el manejo de datos, y `matplotlib` para la visualización.

4.1. Generación de bases locales

Alice (función `generar_bases_alice_df`) Este módulo corresponde al **Script A**. Se reciben como entrada el número de pares N , las bases de Alice (por defecto $a = 0^\circ$ y $a' = 45^\circ$) y una semilla para el generador aleatorio (`seed=1001`). El programa genera un archivo `Alice.choices.csv` con las columnas:

`pair_id, Alice_basis`

donde cada fila corresponde a la elección aleatoria de una de las dos bases con probabilidad 50 %.

Bob (función `generar_bases_bob_df`) Este módulo corresponde al **Script B**. Es análogo al anterior, pero para Bob, con bases por defecto $b = 22,5^\circ$ y $b' = -22,5^\circ$ y semilla `seed=2002`. El resultado se guarda en `Bob.choices.csv`.

4.2. Árbitro cuántico y cálculo de correlaciones

Fusión de datos El árbitro, implementado en la función `arbitro_bell_df`, recibe los dos archivos de elecciones y realiza un `merge` por `pair_id`, validando que no existan duplicados ni pares faltantes.

Regla cuántica Para cada fila k , el árbitro evalúa los ángulos locales $\theta_A^{(k)}$ y $\theta_B^{(k)}$ y calcula:

$$E_k = \cos \left[2(\theta_A^{(k)} - \theta_B^{(k)}) \right].$$

Con este valor se construye el vector de probabilidades para los cuatro posibles resultados conjuntos $(A, B) \in \{\pm 1\}$:

$$P(A, B) = \frac{1}{4} (1 + AB E_k).$$

La función `sample_joint` realiza un muestreo estocástico de esta distribución, asignando a cada par de bases un resultado (A_k, B_k) . De esta manera se generan los archivos `Alice_results.csv` y `Bob_results.csv`, que incluyen tanto la base elegida como el resultado obtenido.

Cálculo de correlaciones El árbitro identifica las cuatro combinaciones experimentales (a, b) , (a, b') , (a', b) y (a', b') a partir de los valores únicos de bases utilizados. Para cada subconjunto, se calcula el promedio:

$$E(\theta_A, \theta_B) = \langle AB \rangle = \frac{1}{M} \sum_{k=1}^M A_k B_k,$$

donde M es el número de pares en esa combinación.

Parámetro de Bell-CHSH Finalmente, se calcula el parámetro

$$S = E(a, b) + E(a, b') + E(a', b) - E(a', b').$$

El código imprime los valores de cada correlación y de S , permitiendo contrastar con los límites teóricos: $S \leq 2$ para modelos locales clásicos y $S \leq 2\sqrt{2}$ para la mecánica cuántica.

4.3. Optimización de bases

El bloque `optimizar_bases_chsh` implementa un barrido estocástico de ángulos iniciales y separaciones relativas de las bases de Alice y Bob. En cada iteración se:

1. Generan ángulos (a, a') y (b, b') dentro de un rango prefijado.
2. Se construyen las elecciones de Alice y Bob mediante las funciones previas.
3. Se evalúa el árbitro con las nuevas bases y se calcula el valor de S .
4. Se registra el conjunto de bases y correlaciones obtenidas.

El programa guarda en `resultados_optimizacion.csv` el historial de iteraciones, y en `Alice_choices_maxS.csv` y `Bob_choices_maxS.csv` los experimentos asociados al valor máximo de S encontrado.

5. Análisis de resultados y visualización

Además de la simulación de correlaciones y del cálculo del parámetro S , se desarrolló una función `analizar_resultados` destinada a examinar de manera sistemática los resultados de la optimización de bases. Esta función recibe el `DataFrame` completo con las iteraciones (`resultados_df`) y produce un conjunto de métricas y visualizaciones.

5.1. Identificación de la configuración óptima

El primer bloque determina el índice de la iteración que alcanzó el valor máximo de S :

$$S_{\max} = \max_i S_i.$$

Se imprimen las bases asociadas (a, a', b, b') y los valores de correlación E medidos en dicha iteración. Esto permite verificar qué configuraciones de ángulos producen la violación más pronunciada de la desigualdad de Bell.

5.2. Estadísticas generales

El programa resume la cantidad de iteraciones que superan el límite clásico ($S > 2$) y cuántas se acercan al máximo cuántico ($S > 2,8$). Estas estadísticas permiten cuantificar la robustez de la violación y evaluar la frecuencia con la que se alcanzan valores cercanos al límite teórico de Tsirelson ($S = 2\sqrt{2}$).

5.3. Gráficas generadas

Se generan automáticamente y se guardan las siguientes figuras:

- **Correlaciones para S_{\max} (correlations_S_max.pdf):** Gráfico de barras comparando correlaciones medidas y teóricas en la iteración que alcanzó el máximo S .

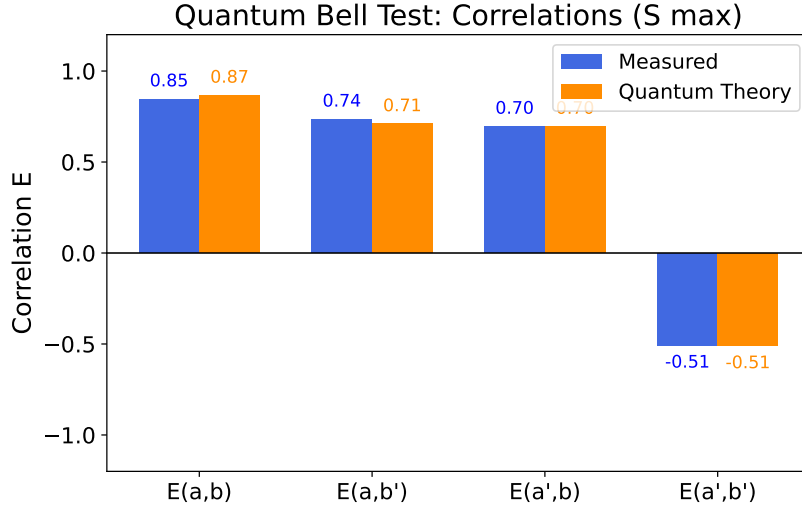


Figura 1: Correlaciones medidas vs. teóricas en la iteración óptima (S_{\max}).

- **Valor CHSH de S (CHSHvalue_S_max.pdf):** Gráfico de una barra única mostrando el valor de S medido en el caso óptimo, con líneas de referencia en $S = 2$ (límite clásico) y $S = 2\sqrt{2}$ (máximo cuántico).

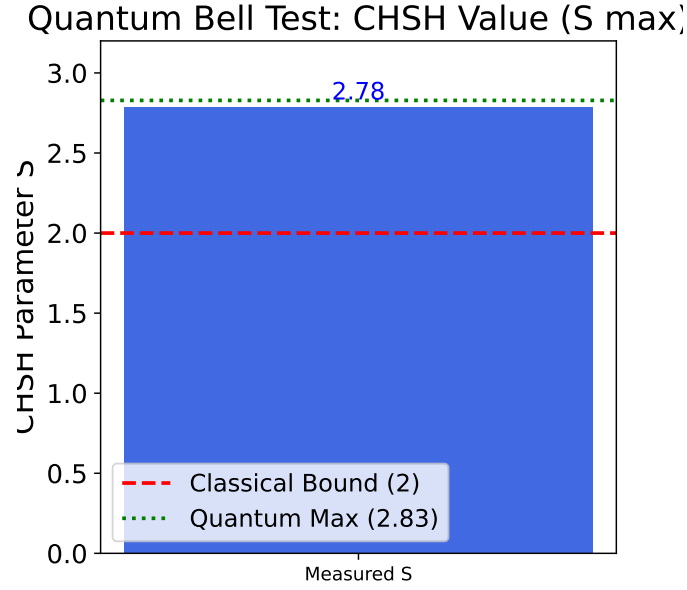


Figura 2: Valor de S en el caso óptimo, con límites clásico y cuántico.

- **Distribución de S (Distribution.S.pdf):** Histograma con todas las iteraciones (10000 realizaciones), mostrando cuántas veces se alcanzan valores cercanos a los límites clásico y cuántico. Este gráfico ilustra la dispersión estadística de los resultados.

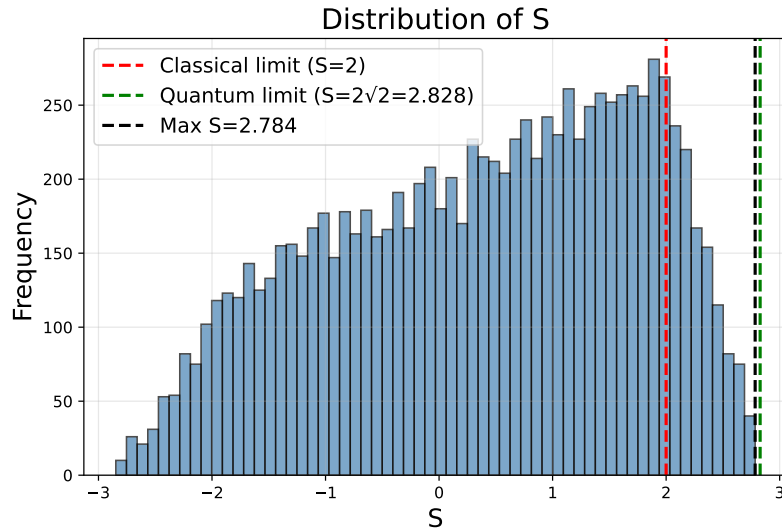


Figura 3: Distribución estadística de los valores de S a lo largo de todas las simulaciones.

- **Evolución temporal de S (AllModels.S.pdf):** Diagrama de dispersión que representa los valores de S obtenidos en cada iteración. Se resalta con un asterisco rojo la iteración con $S_{\text{máx}}$, esto indica en que numero de iteracion lo alcanzamos, no sera siempre la misma iteracion debido a la naturaleza probabilista que introdujimos.

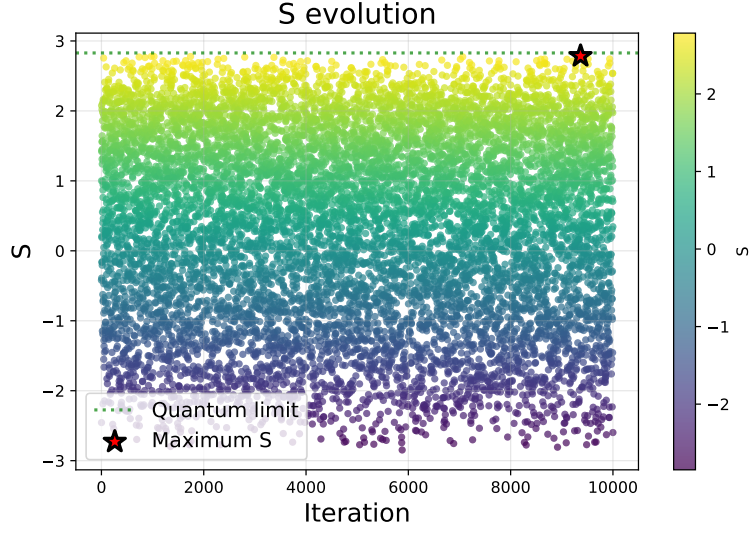


Figura 4: Evolución de los valores de S en cada iteración de la optimización.

- **Correlaciones en función de S (E.S.pdf):** Cuatro diagramas de dispersión, uno para cada correlación ($E(a,b)$, $E(a,b')$, $E(a',b)$ y $E(a',b')$), contra los valores de S . Esto permite estudiar la dependencia de las correlaciones elementales en la magnitud de la violación.

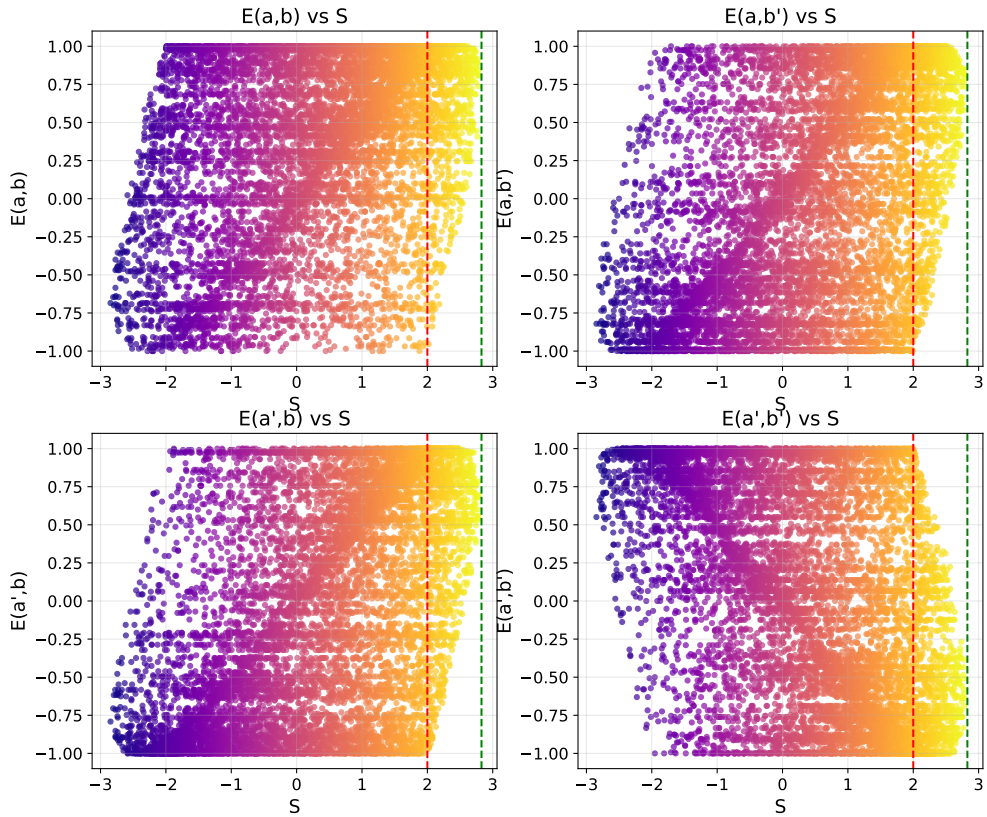


Figura 5: Relación entre cada correlación elemental E y el valor de S alcanzado.

4. Conclusión del análisis

La función `analizar_resultados` complementa el proceso de simulación al proporcionar un conjunto de indicadores gráficos y numéricos que permiten no solo identificar la mejor configuración de ángulos, sino también evaluar la distribución estadística de las violaciones de Bell en un gran número de simulaciones. Los resultados visuales confirman que el programa implementado reproduce la predicción cuántica del estado de Bell Φ^+ , alcanzando valores de S consistentes con el límite teórico $2\sqrt{2}$.

6. Notas

- Es importante notar que para todas las bases se mantuvieron las bases tanto para Alice y Bob, así como, el número de mediciones (1000).
- Todo el código fuente y los scripts pueden ser encontrados en el siguiente link