



**Departamento de  
Física de la  
Materia Condensada  
Universidad Zaragoza**



---

TRABAJO DE FIN DE GRADO EN FÍSICA

## **Redes neuronales cuánticas**

---

AUTOR:

Lucas Tristán Artigas

DIRECTORES:

David Zueco Láinez  
Sebastián Roca Jerat

Departamento de Física de la Materia Condensada  
Facultad de Ciencias, Universidad de Zaragoza

Julio 2024



*“What is college? Stop going until we figure it out”*  
- John Mulaney, Kid Gorgeous at Radio City

Gracias a David, por ser un referente tanto a nivel profesional como personal.

Gracias a Sebas, por enseñarme a *discutir* sobre física.

Gracias a Pilar e Iván, por haber hecho el proceso mucho más ameno.

Gracias a David y Jesús, por reíros de mis náuticos en un bar.

Gracias a mi familia, por ser la columna vertebral que sostiene mi vida.

Gracias a mi novia, por enseñarme lo que significa el amor.

Gracias a Guillermo “Maverick” Pascua, por ser el mejor *wingman* que he podido tener.

Gracias a Dompal, por ser la inspiración para salir de mi zona de confort.

Gracias a Rafa “Larry” Bes, por reafirmarme en todos los momentos de duda.

Gracias a todos mis amigos que han hecho este viaje posible: Abella, María, Sara, Andrea, Gabri, Diego, Luis, Fer, Carrasco, Kumar, María, Irene...

Gracias a la física, por haberme forzado a conocerme más de lo que jamás hubiera imaginado.

Os estoy eternamente agradecido.

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Objetivos y Estructura de la Memoria</b>	<b>2</b>
<b>3</b>	<b>Algoritmos Cuánticos Variacionales (VQAs)</b>	<b>3</b>
3.1	Variational Quantum Eigensolver (VQE) . . . . .	3
<b>4</b>	<b>Modelos Long-Range (LRH)</b>	<b>7</b>
4.1	Introducción de los Modelos Long-Range . . . . .	7
4.2	Modelo Long-Range de Ising Cuántico . . . . .	8
<b>5</b>	<b>Estados Coherentes Generalizados (GCS)</b>	<b>9</b>
5.1	Introducción del <i>ansatz</i> GCS . . . . .	9
5.2	Implementación del <i>ansatz</i> GCS en un circuito cuántico . . . . .	10
<b>6</b>	<b>Cálculo analítico de <math>\langle H \rangle</math></b>	<b>12</b>
6.1	Cálculo de $\langle \sigma_n^\alpha \rangle$ . . . . .	12
6.2	Cálculo de $\langle \sigma_n^z \sigma_m^z \rangle$ . . . . .	14
<b>7</b>	<b>Resultados</b>	<b>16</b>
7.1	Estudio Previo . . . . .	16
7.2	Resolución del modelo long-range de Ising cuántico . . . . .	20
<b>8</b>	<b>Conclusión</b>	<b>24</b>
<b>9</b>	<b>Bibliografía</b>	<b>25</b>
<b>A</b>	<b>Cálculo de <math>\lambda</math></b>	<b>26</b>
<b>B</b>	<b>Código</b>	<b>27</b>
<b>C</b>	<b>Especificaciones del dispositivo</b>	<b>28</b>

# 1 Introducción

Los problemas de optimización, generalmente presentados como la minimización de una función sujeto a unas ligaduras, son omnipresentes en el mundo actual. El interés en dichos problemas reside en su variedad de aplicaciones: desde reducir el tiempo de envío de un paquete hasta minimizar el riesgo en una inversión financiera.

En el ámbito de la física, el problema de los muchos cuerpos cuántico (“quantum many-body problem”) juega un papel central en el estudio del comportamiento de la materia. La obtención de la energía del estado fundamental del sistema, que no es sino la minimización del valor esperado del hamiltoniano, no es trivial. La dificultad del problema es que el espacio de Hilbert crece exponencialmente con el número de partículas y, por ende, es inviable resolverlo con ordenadores clásicos.

Un ordenador cuántico, en cambio, parte con la ventaja de que no necesita construir un espacio de Hilbert, sino que constituye un espacio de Hilbert en sí mismo. Por lo tanto, es de esperar que un circuito cuántico que presente la suficiente complejidad sea capaz de representar el espacio de Hilbert en el que estamos interesados de manera más eficiente. Esto, a su vez, debería facilitar la obtención del estado fundamental.

No obstante, la realidad actual se encuentra muy alejada de dicha situación. Los ordenadores cuánticos más avanzados a los que tenemos acceso se conocen como ordenadores cuánticos de escala intermedia ruidosa (NISQs)<sup>1</sup>. Los circuitos cuánticos que son capaces de realizar son muy limitados y, por tanto, necesitamos algoritmos cuánticos que se adapten a este tipo de tecnología.

En este contexto, surgen los algoritmos cuánticos variacionales (VQAs), que están basados en el uso de redes neuronales cuánticas (QNNs). Los VQAs presentan una línea de investigación con gran potencial a corto plazo, pues se basan en usar de forma híbrida la tecnología cuántica con la clásica. Gracias al uso de optimizadores clásicos, los VQAs requieren de circuitos cuánticos menos<sup>2</sup> complejos que pueden ser implementados en ordenadores NISQs. No obstante, entre las mayores desventajas de los algoritmos variacionales se encuentran su precisión, eficiencia y entrenabilidad [2].

Cuando los algoritmos variacionales cuánticos se aplican a problemas clásicos de optimización adoptan el nombre de “quantum approximate optimization algorithm” (QAOA) [3]. Por su parte, el VQA diseñado precisamente para determinar la energía del estado fundamental de un sistema cuántico se denomina “variational quantum eigensolver” (VQE) [8].

En última instancia, los dispositivos cuánticos actuales aún no presentan una mejora frente a los métodos clásicos. En palabras de John Preskill [11]:

*“Will NISQ technology running QAOA or VQE be able to outperform classical algorithms that find approximate solutions to the same problems? Nobody knows, but we’re going to try it and see how well we can do.”*<sup>3</sup>

---

<sup>1</sup>John Preskill acuñó el término NISQs en 2018 [11]. La escala intermedia hace referencia a que el número de qubits que son capaces de soportar oscila entre 50 y 100. Los denominamos ruidosos porque no tenemos un control perfecto de los qubits ya que su tiempo de coherencia es muy limitado y las puertas cuánticas implementadas presentan un error significativo.

<sup>2</sup>Previo a los algoritmos variacionales, los algoritmos cuánticos para determinar autoestados de un sistema se basaban en el “quantum phase estimation” (QPE). Para aplicaciones prácticas, dicho algoritmo requiere de un circuito cuántico con un tiempo de coherencia del orden de millones, o incluso miles de millones, de puertas cuánticas mientras que los NISQs únicamente son capaces de soportar del orden de cientos de puertas [8].

<sup>3</sup>¿Podrá la tecnología NISQ ejecutando el QAOA o VQE superar a los algoritmos clásicos que encuentran soluciones aproximadas a los mismos problemas? Nadie lo sabe, pero vamos a intentarlo y ver qué tan bien podemos hacerlo.

## 2 Objetivos y Estructura de la Memoria

El principal objetivo del Trabajo Fin de Grado es demostrar que la red neuronal con una capa asociada al *ansatz* de estados generalizados coherentes (GCS) es simulable clásicamente, diseñando así un algoritmo *quantum-inspired*.

A raíz de esto, la segunda finalidad del trabajo es la comparación del comportamiento de la red neuronal cuántica con una capa frente a su análogo clásico en la resolución del problema de muchos cuerpos cuántico con interacciones de largo alcance. Asimismo, con el fin de disponer de un método que no sea emulable clásicamente, vamos a incluir en el estudio la red neuronal cuántica con dos capas.

La estructura de la memoria se ha llevado a cabo con los objetivos en mente, de forma que se realiza una introducción teórica de todos los conceptos necesarios para entender los objetivos. Por tanto, el autor ha procurado realizar un trabajo que sea accesible al lector experimentado pero que no sea necesariamente experto en la materia.

En primer lugar, la Sección 3 presenta de forma teórica los algoritmos cuánticos variacionales y, en concreto, el VQE. Así, entenderemos las distintas etapas del algoritmo y las condiciones que debe verificar el hamiltoniano del sistema para que pueda ser tratado con el VQE.

A continuación, la Sección 4 recoge una introducción de los sistemas con interacciones de largo alcance (“modelos long-range”). Asimismo, nos centraremos en el modelo *long-range* de Ising cuántico ya que es un sistema idóneo para verificar la precisión tanto de las redes neuronales cuánticas como del algoritmo *quantum-inspired*, pues se trata de un modelo con una fenomenología muy rica.

En tercer lugar, en la Sección 5, presentamos los estados coherentes generalizados (GCS) que no son sino el *ansatz* que vamos a emplear en este trabajo. Además, vamos a explicar su implementación en un circuito cuántico que pueda ser introducido en el lenguaje de programación *Python*, a través del paquete *qiskit*.

A estas alturas, ya hemos introducido todos los conceptos necesarios para satisfacer los objetivos principales de la memoria. El resultado más relevante del trabajo se encuentra en la Sección 6, donde demostramos que el circuito cuántico con una capa se puede simular con un ordenador clásico. Nótese que la demostración viene complementada por el Anexo A, en el que recogemos parte del desarrollo matemático.

Posteriormente, en la Sección 7, discutimos la eficacia de los tres métodos. Primero, se realiza un estudio previo con un sistema trivial para verificar que todos los métodos funcionan correctamente. Finalmente, llevamos a cabo la resolución del modelo *long-range* de Ising cuántico y concluimos escogiendo el método que presenta mejores prestaciones.

En último lugar, la Sección 8 recapitula el desarrollo realizado en el trabajo e indica una manera de mejorar la implementación del algoritmo *quantum-inspired* presentada en este Trabajo Fin de Grado.

En definitiva, al finalizar la lectura del documento, el lector adquirirá una cierta familiaridad con los métodos variacionales cuánticos y modelos cuánticos de largo alcance.

Como nota final, destacamos que el código realizado en este trabajo y las prestaciones del ordenador en el que se han ejecutado todas las optimizaciones se recogen en los Anexos B y C, respectivamente.

### 3 Algoritmos Cuánticos Variacionales (VQAs)

Los algoritmos cuánticos variacionales (VQAs) son un emergente candidato para obtener supremacía cuántica<sup>4</sup> en un futuro no tan lejano con los dispositivos que se disponen a día de hoy; los ordenadores cuánticos de escala intermedia ruidosa (NISQs) [2].

Los VQAs son algoritmos híbridos que se fundamentan en calcular el valor de una función coste (“cost function”) cuánticamente, mediante un circuito cuántico parametrizado, que es posteriormente optimizado clásicamente. De esta forma, los parámetros del circuito cuántico se actualizan con el resultado de la optimización clásica, finalizando así una iteración del algoritmo. Los VQAs se componen, por tanto, de una función coste que puede ser expresada, sin pérdida de generalidad, como:

$$C(\boldsymbol{\theta}) = f(\{\rho_k\}, \{O_k\}, U(\boldsymbol{\theta})) \quad (1)$$

donde  $f$  es una función que depende de un conjunto de observables  $\{O_k\}$  y de estados iniciales  $\{\rho_k\}$ , así como de un operador unitario parametrizado  $U(\boldsymbol{\theta})$ , que corresponde al conjunto de puertas que conforman el circuito cuántico y se denomina *ansatz* [2]. En la Figura 1 se recoge el método operativo de un VQA genérico.

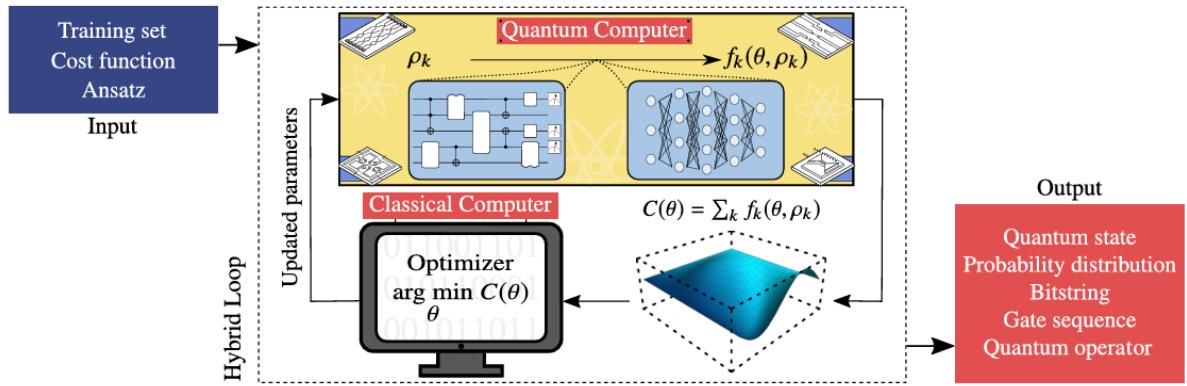


Figura 1: Flujo de trabajo de los VQAs [2]

Cabe destacar que las redes neuronales cuánticas corresponden a circuitos cuánticos parametrizados que sean aptos para ser optimizados, es decir, que el espacio de fases de los parámetros sea lo suficientemente suave como para que pueda efectuarse su gradiente. Debido a esto, en la Figura 1 se muestra la imagen tanto de un circuito cuántico como de una red neuronal. [2]

#### 3.1 Variational Quantum Eigensolver (VQE)

A continuación, vamos a estudiar en mayor detalle el “variational quantum eigensolver” (VQE), que no es sino el primer VQA llevado a cabo experimentalmente [8]. El desarrollo del algoritmo presentado en esta Sección se ha extraído del curso de IBM sobre algoritmos variacionales [5].

El objetivo del algoritmo es encontrar la energía del estado fundamental de un sistema cuántico. Consecuentemente, la función coste del VQE es:

$$C(\boldsymbol{\theta}) \equiv \langle \psi(\boldsymbol{\theta}) | \mathcal{H} | \psi(\boldsymbol{\theta}) \rangle \quad (2)$$

donde  $\mathcal{H}$  es el hamiltoniano del sistema de interés y  $|\psi(\boldsymbol{\theta})\rangle$  es un estado normalizado que depende de los parámetros a minimizar.

<sup>4</sup>El término fue acuñado por John Preskill en 2012 [10] y hace alusión a la habilidad que tienen los ordenadores cuánticos de resolver ciertos problemas computacionales con una mejora en tiempo superpolinomial frente a los dispositivos clásicos.

El VQE se basa en el principio variacional de la mecánica cuántica, que establece que la aproximación óptima del estado fundamental de un sistema cuántico es aquella que minimiza el valor esperado del hamiltoniano:

$$\langle \psi(\boldsymbol{\theta}) | \mathcal{H} | \psi(\boldsymbol{\theta}) \rangle \geq E_0 \quad (3)$$

siendo  $E_0$  la energía fundamental del sistema.

La demostración de dicho enunciado se basa en la descomposición espectral del hamiltoniano del sistema. Para un estado normalizado cualquiera, se verifica que:

$$\langle \psi | \mathcal{H} | \psi \rangle = \langle \psi | \left( \sum_k E_k |\phi_k\rangle \langle \phi_k| \right) | \psi \rangle = \sum_k E_k |\langle \psi | \phi_k \rangle|^2 \geq E_0 \quad (4)$$

donde  $\{|\phi_k\rangle\}_k$  es una base ortonormal compuesta por autoestados del hamiltoniano con autovalor  $E_k$ . Nótese que en la primera igualdad hemos empleado que  $\mathcal{H} = \sum_k E_k |\phi_k\rangle \langle \phi_k|$ . En segundo lugar, hemos introducido la desigualdad ya que la energía fundamental del sistema es menor o igual que el resto de energías, es decir,  $E_k \geq E_0, \forall k$ . Además, en la desigualdad se ha tenido en cuenta que  $\sum_k |\langle \psi | \phi_k \rangle|^2 = 1$ .

La ventaja de expresar el teorema variacional en función de la energía del sistema es que la energía fundamental debe verificar que  $E_0 > -\infty$ , es decir, tiene una cota inferior. Además, en general, la energía del sistema no dispone de una cota superior.

Una vez introducido el teorema variacional, vamos a estudiar con mayor profundidad los conceptos de *estado de referencia* y *ansatz*.

El estado de referencia es el estado inicial fijo con el que va a trabajar el algoritmo,  $|\rho\rangle$ . Es fundamental destacar que dicho estado no se encuentra parametrizado pues es simplemente el estado de partida del algoritmo y es constante. Luego, el primer paso del VQE es inicializar el estado de referencia mediante un operador unitario  $U_R$  tal que  $|\rho\rangle = U_R |0\rangle$ . Nótese que es práctica común emplear como estado de referencia el propio estado predeterminado  $|0\rangle$ , de forma que en estos casos se obtiene que  $U_R = I$ .

A partir del estado de referencia, se introduce la forma variacional,  $U_V(\boldsymbol{\theta})$ , tal que  $|\psi(\boldsymbol{\theta})\rangle = U_V(\boldsymbol{\theta})|\rho\rangle$ , donde recordamos que  $|\psi(\boldsymbol{\theta})\rangle$  es el estado parametrizado con el que vamos a evaluar el valor esperado del hamiltoniano del sistema. Ver Ecuación (2). Por tanto, la forma variacional define el conjunto de estados parametrizados que el algoritmo va a poder explorar. Debido a esto, es imprescindible que los estados parametrizados dependan de los parámetros de forma continua y derivable, para poder así acceder a todos los estados de forma eficiente. Así, el *ansatz* se define como la combinación entre la parte del circuito cuántico sin parametrizar, correspondiente a  $U_R$ , y la parametrizada, asociada a  $U_V(\boldsymbol{\theta})$ . Ver Ecuación (5).

$$U_A(\boldsymbol{\theta}) \equiv U_V(\boldsymbol{\theta}) U_R \quad (5)$$

Cabe destacar que el *ansatz* empleado puede depender del hamiltoniano en cuestión, de forma que se denomina *problem-specific ansatz*, o puede ser completamente independiente del problema, en cuyo caso se conoce como *heuristic ansatz*. En nuestro caso, vamos a trabajar con un *heuristic ansatz*, con el que vamos a derivar analíticamente  $\langle \sigma_n^\alpha \rangle$  y  $\langle \sigma_n^z \sigma_m^z \rangle$ .

Por otro lado, vamos a discutir las condiciones que debe verificar el hamiltoniano del sistema para que sea computacionalmente eficiente llevar a cabo su valor esperado. En concreto, vamos a estudiar cómo *qiskit*<sup>5</sup> [6] calcula dicho valor esperado, pues la implementación del VQE en este trabajo se ha realizado usando dicho SDK.

---

<sup>5</sup> *Qiskit* es un kit de desarrollo de software (SDK) diseñado por IBM que permite trabajar con ordenadores cuánticos.

*Qiskit* dispone de una primitiva<sup>6</sup> llamada “*Estimator*” que, dado un estado cuántico, calcula el valor esperado del observable en cuestión; en nuestro caso, el hamiltoniano del sistema usando los estados  $|\phi(\theta)\rangle$ . Para llevar esto a cabo, descompone el valor esperado de la siguiente manera:

$$\langle \mathcal{H} \rangle = \sum_{\lambda} p_{\lambda} \lambda \quad (6)$$

donde  $p_{\lambda} = |\langle \psi | \lambda \rangle|^2$ , siendo  $|\lambda\rangle$  autoestados de  $\mathcal{H}$  con autovalor  $\lambda$ , representa la probabilidad de medir  $\lambda$ .

Por consiguiente, es necesario disponer de una base de autoestados para poder evaluar la Ecuación (6). Debido a que, en general, no es trivial obtener dicha base, se expresa el hamiltoniano del sistema como una combinación de matrices de Pauli, pues conocemos tanto sus autoestados como sus autovalores. En concreto, una matriz<sup>7</sup>  $2^n \times 2^n$  se puede expresar como una combinación lineal del producto tensorial de  $n$  matrices de Pauli y la identidad:

$$\mathcal{H} = \sum_{k_{n-1}=0}^3 \cdots \sum_{k_0=0}^3 w_{k_{n-1} \dots k_0} \sigma_{k_{n-1}} \otimes \cdots \otimes \sigma_{k_0} \equiv \sum_{k=0}^{4^n-1} w_k P_k \quad (7)$$

donde usamos que  $(\sigma_0, \sigma_1, \sigma_2, \sigma_3) \equiv (I, \sigma^x, \sigma^y, \sigma^z)$  y  $P_k \equiv \sigma_{k_{n-1}} \otimes \cdots \otimes \sigma_{k_0}$  con  $k \equiv k_{n-1} \cdots k_0 \in \{0, 1, 2, 3\}^n$ . Nótese que, debido a que en cada producto tensorial puede haber una matriz de Pauli o la identidad, tenemos  $4^n$  términos en la Ecuación (7).

Por su parte, las matrices de Pauli son hermíticas, luego su producto tensorial también lo es. Por consiguiente,  $P_k$  es diagonalizable unitariamente, es decir, existe una transformación unitaria  $V_k \equiv V_{k_{n-1}} \otimes \cdots \otimes V_0$  tal que  $P_k = V_k^\dagger \Lambda_k V_k$ , donde  $\Lambda_k \equiv \Lambda_{k_{n-1}} \otimes \cdots \otimes \Lambda_0$  es una matriz diagonal compuesta por los autovalores de  $P_k$ . Es fundamental destacar que la virtud de descomponer el hamiltoniano en matrices de Pauli es que, en la base computacional, la forma de las matrices  $V_k$  es conocida<sup>8</sup>. Así, el valor esperado de  $P_k$  se puede expresar como:

$$\begin{aligned} \langle \psi | P_k | \psi \rangle &= \langle \psi | V_k^\dagger \Lambda_k V_k | \psi \rangle \\ &= \langle \psi | V_k^\dagger \left( \sum_{i=0}^{2^n-1} |i\rangle\langle i| \right) \Lambda_k \left( \sum_{j=0}^{2^n-1} |j\rangle\langle j| \right) V_k | \psi \rangle \\ &= \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} \langle \psi | V_k^\dagger |i\rangle \langle i| \Lambda_k |j\rangle \langle j| V_k | \psi \rangle \\ &= \sum_{i=0}^{2^n-1} \langle \psi | V_k^\dagger |i\rangle \langle i| \Lambda_k |i\rangle \langle i| V_k | \psi \rangle \\ &= \sum_{i=0}^{2^n-1} |\langle i | V_k | \psi \rangle|^2 \lambda_{ki} \end{aligned} \quad (8)$$

donde  $|i\rangle$  son los autoestados de  $\Lambda_k$  tales que  $\langle i | \Lambda_k | j \rangle = \delta_{ij} \lambda_{ki}$ , pues  $\Lambda_k$  es diagonal. Además, se ha empleado la relación de clausura  $\sum_{i=0}^{2^n-1} |i\rangle\langle i| = I$ .

Por tanto, el valor esperado del hamiltoniano del sistema es:

$$\langle \psi | \mathcal{H} | \psi \rangle = \sum_{k=0}^{4^n-1} w_k \sum_{i=0}^{2^n-1} |\langle i | V_k | \psi \rangle|^2 \lambda_{ki} \quad (9)$$

---

<sup>6</sup>Las primitivas son los bloques fundamentales a partir de los cuales se pueden realizar algoritmos cuánticos más complejos [6].

<sup>7</sup>Nótese que la dimensión de cada qubit es 2, luego la dimensión de un sistema con  $n$  qubits es  $2^n$ .

<sup>8</sup>Dado un qubit arbitrario, tenemos que:  $V_0 = I$ ,  $V_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ ,  $V_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}$  y  $V_3 = I$ .

Nótese que, gracias a la descomposición del hamiltoniano en matrices de Pauli, conocemos todos los términos de la Ecuación (9) excepto  $|\langle i|V_k|\psi\rangle|^2$ , que no es sino la probabilidad de medir el estado  $|i\rangle$  a partir del estado inicial  $V_k|\psi\rangle$ . Aquí es donde entra el papel del “*Estimator*”, que calcula dicha probabilidad. En concreto, lleva a cabo el circuito cuántico  $V_k|\psi\rangle$ , para poder así diagonalizar  $P_k$  en la base computacional, y lo ejecuta midiendo el estado final un número suficientemente alto de veces para poder determinar  $|\langle i|V_k|\psi\rangle|^2$ .

No obstante, el número de términos en la Ecuación (9) crece exponencialmente y, consecuentemente, no es viable computar  $\langle\mathcal{H}\rangle$ . Luego, necesitamos exigir al hamiltoniano del sistema que pueda ser descompuesto de forma *escasa* en matrices de Pauli (“Sparse Pauli Descomposition”). Rigurosamente, exigimos que los términos  $w_k$  distintos de cero crezcan de forma polinómica con el número de qubits, y no de manera exponencial. Asimismo, las probabilidades  $|\langle i|V_k|\psi\rangle|^2$  también deben escalar de forma polinómica para que sea computacionalmente eficiente realizar el cálculo del valor esperado del hamiltoniano del sistema.

$$\langle\psi|\mathcal{H}|\psi\rangle = \sum_{k=0}^{Pol(n)} w_k \sum_{i=0}^{Pol(n)} |\langle i|V_k|\psi\rangle|^2 \lambda_{ki} \quad (10)$$

Los hamiltonianos que se pueden descomponer de forma *escasa* en matrices de Pauli describen una multitud de sistemas físicos, como la mayoría de modelos de espín [16], entre los que se encuentra el modelo de largo alcance que vamos a simular en este trabajo. Ver Sección 4.

En último lugar, es fundamental destacar que la elección del optimizador clásico juega un papel central en el éxito del VQE. En general, los optimizadores se pueden dividir en dos categorías: “Gradient-Based” y “Gradient-Free”. Los primeros emplean el gradiente de la función de coste para evaluar la dirección donde se minimiza la función, mientras que los segundos minimizan la función de coste a través de otros métodos que no hacen uso de sus derivadas. La mayor diferencia entre los métodos es que los optimizadores “Gradient-Based” suelen ser más eficientes encontrando mínimos locales de la función objetivo, y los “Gradient-Free” son más robustos obteniendo el mínimo global de la función, a cambio de ser más costosos computacionalmente. Además, es práctica común inicializar (“bootstrap”) la optimización con los parámetros resultantes de la optimización previa, consiguiendo así un menor tiempo de convergencia.

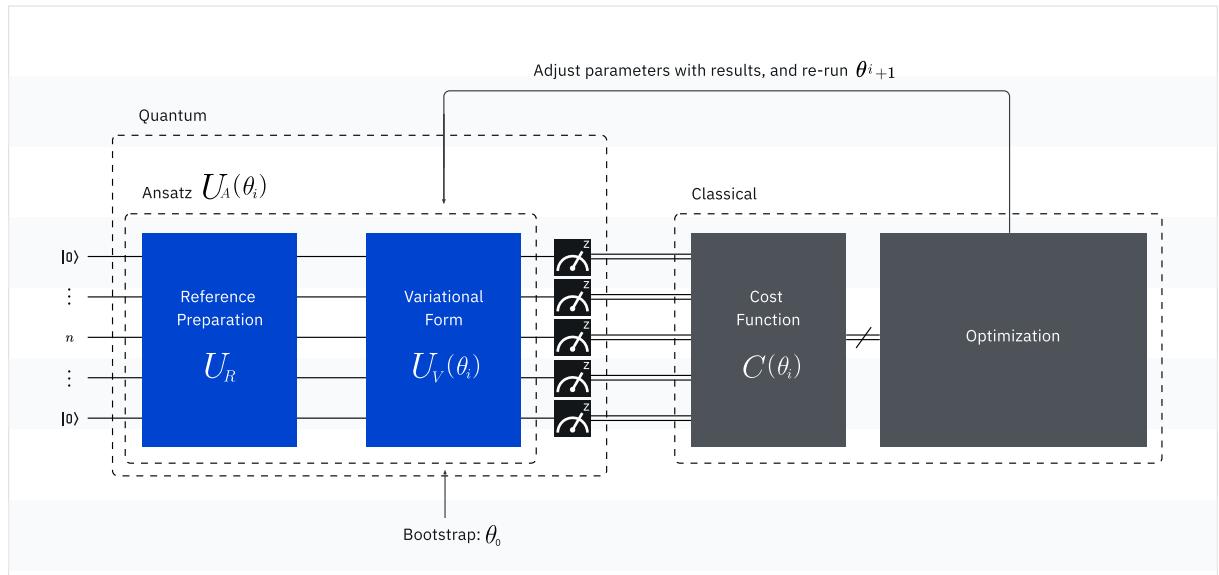


Figura 2: Esquema completo del VQE [5]

## 4 Modelos Long-Range (LRH)

En este trabajo vamos a afrontar el problema de muchos cuerpos cuántico (“quantum many-body problem”), ya que vamos a simular el modelo *long-range* de Ising cuántico. Por tanto, es necesario introducir los sistemas de largo alcance. Las ideas que se van a tratar en esta sección están recogidas en [13; 14].

### 4.1 Introducción de los Modelos Long-Range

En general, los modelos *long-range* cuánticos en una red con dimensión  $d$  y  $N$  sitios, vienen descritos por la siguiente familia de Hamiltonianos:

$$\mathcal{H}_{LRH} = - \sum_{i,j=1}^N J_{ij} \mathcal{C}_i \mathcal{C}_j \quad (11)$$

donde  $\mathcal{C}_i$  es un operador actuando sobre el sitio  $i$ , y las constantes de acoplos  $J_{ij}$  se definen como:

$$J_{ij} = J \frac{\tilde{J}_{ij}}{\tilde{N}} \quad (12)$$

siendo  $J$  una constante que indica la intensidad de las interacciones, mientras que  $\tilde{N}$  es el factor de normalización y  $\tilde{J}_{ij}$  son los coeficientes que decaen acorde a  $1/r^\alpha$ . Ver Ecuación (13).

$$\tilde{J}_{ij}(\mathbf{r}_{ij}) = \begin{cases} b & \text{con } i = j \\ |i - j|^{-\alpha} & \text{con } i \neq j \end{cases} \quad (13)$$

Nótese que  $b$  es un parámetro que desplaza el espectro de  $J$ . A lo largo de este trabajo, las simulaciones se han llevado a cabo con  $b = 1$  y  $d = 1$ .

Finalmente, cabe destacar que la normalización viene dada por:

$$\tilde{N} = \sum_{j=1}^N \tilde{J}_{ij} = 1 + \sum_{j=1}^N \frac{1}{j^\alpha} \quad (14)$$

Es fundamental notar que empleamos condiciones de contorno periódicas (PBC), de forma que la normalización (14) es independiente del sitio  $i$ .

En definitiva, los modelos *long-range* se caracterizan por describir interacciones entre dos cuerpos que, a largas distancias, siguen una ley de  $1/r^\alpha$ . Nótese que este tipo de modelos describen una gran variedad de sistemas, como la interacción de Coulomb o Van-der Waals.

La clasificación de los sistemas de largo alcance gira en torno la constante de decaimiento,  $\alpha$ . Ver Figura 3.

Si  $\alpha < d$ , entonces se dice que el modelo es de largo alcance fuerte (“strong long-range”), ya que las interacciones decaen de manera suficientemente lenta tal que la suma (11) escala superlinealmente con el tamaño del sistema  $N$  y, por tanto, el sistema pierde la extensividad. No obstante, introducimos la normalización, denominada factor de renormalización de Kac, para recuperar dicha extensividad y tener así un límite termodinámico bien definido.

En el caso  $\alpha > d$ , el modelo es extensivo y se distinguen dos regiones diferentes. Si  $\alpha > \alpha_*$ , el sistema presenta los mismos exponentes críticos que el caso  $\alpha \rightarrow \infty$ , que corresponde al sistema con interacciones solo a primeros vecinos. En esta región, el modelo se denomina de corto alcance (“short-range”). En el caso restante, cuando  $\alpha_* > \alpha > d$ , los exponentes críticos son distintos, lo que significa que el sistema presenta interacciones de largo alcance pero manteniendo la extensividad del modelo. Esta región se conoce como el régimen de largo alcance débil (“weak long-range”). Nótese que el valor de  $\alpha_*$  depende de cada modelo en particular.

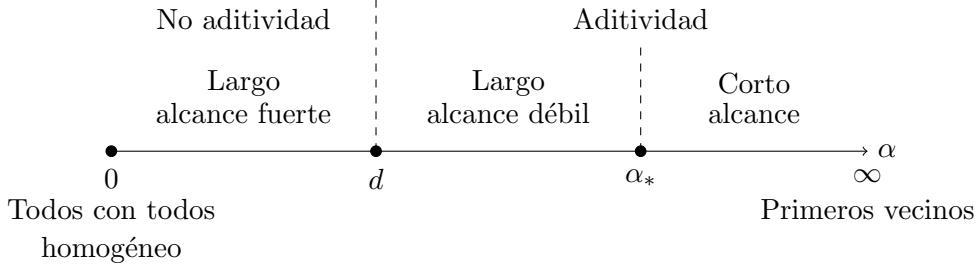


Figura 3: Diagrama de las regiones de los modelos *long-range* en función del valor de  $\alpha$  [14].

## 4.2 Modelo Long-Range de Ising Cuántico

El hamiltoniano que vamos a simular es:

$$\mathcal{H} = \sum_{i,j=1}^N J_{ij} \sigma_i^z \sigma_j^z - h_x \sum_{i=1}^N \sigma_i^x + \epsilon \sum_{i=1}^N \sigma_i^z \quad (15)$$

donde las constantes de acople  $J_{ij}$  se definen en (12), tratándose así de un modelo *long-range*. Destacamos que los términos asociados a las interacciones entre los distintos sitios provocan que los espines se orienten en las dos orientaciones posibles a lo largo de un eje, denominado eje fácil. En nuestro caso, el eje fácil se corresponde al eje  $Z$ . Por otro lado, el término  $h_x$  está asociado a un campo magnético uniforme y transversal, de forma que estamos tratando con el modelo de Ising cuántico<sup>9</sup>. Finalmente, disponemos de un pequeño campo magnético longitudinal,  $\epsilon$ , con el fin de romper la degeneración entre los estados asociados a todos los espines *up* o *down*. Nótese que en las simulaciones realizadas en este trabajo, hemos fijado  $h_x = 1$  tal que el resto de magnitudes de energía las expresamos en unidades de  $h_x$ . Por ejemplo, vamos a trabajar con  $\epsilon/h_x = 0.001$ .

Tal y como hemos comentado en la Sección 4, el comportamiento del sistema (15) varía significativamente en función del valor de  $\alpha$ . En concreto, el sistema presenta interacciones de largo alcance cuando  $\alpha < 1$ , pues se encuentra en el régimen de largo alcance fuerte. Por su parte, en este sistema se verifica que  $\alpha_* = 3$ . Por tanto, el sistema es de largo alcance débil cuando  $1 < \alpha < 3$ , de forma que recupera la aditividad manteniendo interacciones de largo alcance. Finalmente, la región de corto alcance del sistema tiene lugar cuando  $\alpha > 3$ , donde tiene un comportamiento similar al modelo de primeros vecinos.

Por otro lado, el sistema (15) presenta cambios significativos en función de  $J$ . El sistema es ferromagnético si  $J < 0$ , mientras que es antiferromagnético cuando  $J > 0$ . Además, notamos que en el caso  $J = 0$ , el sistema es trivial pues se reduce a un modelo de Ising clásico<sup>10</sup> en el eje  $X$ . Además, existe un punto crítico  $J_c(\alpha)$ , en el que el sistema sufre una transición de fase cuántica (QPT) de segundo orden, desde un estado ordenado a una fase paramagnética.

En último lugar, es fundamental destacar que el modelo (15) verifica las condiciones necesarias para ser tratado con el VQE. El hamiltoniano está expresado en función de matrices de Pauli y, además, su descomposición es *escasa* (“Sparse Pauli Descomposition”).

Por tanto, gracias a la complejidad del modelo, el hamiltoniano *long-range* de Ising cuántico (15) es un sistema idóneo para estudiar el comportamiento de redes neuronales cuánticas.

<sup>9</sup>La introducción del campo magnético transversal provoca que el hamiltoniano no sea diagonal.

<sup>10</sup>Nótese que la contribución del campo longitudinal  $\epsilon$  es despreciable.

## 5 Estados Coherentes Generalizados (GCS)

En esta sección se introduce el *ansatz* de estados atómicos coherentes generalizados (GCS), junto a su implementación en *qiskit*.

### 5.1 Introducción del *ansatz* GCS

En primer lugar, presentamos el *ansatz* de estados atómicos coherentes (CS) [15],  $|\phi(\boldsymbol{\alpha})\rangle$ , que corresponde a realizar una rotación general en cada qubit de forma individual:

$$|\phi(\boldsymbol{\alpha})\rangle \equiv \mathcal{U}(\boldsymbol{\alpha})|0\rangle \quad (16)$$

donde la matriz de rotación está parametrizada de la siguiente manera:

$$\mathcal{U} \equiv \exp \left( -\frac{i}{2} \sum_{j=1}^N \tau_j (\hat{n}_j \cdot \vec{\sigma}_j) \right) = \prod_{j=1}^N \exp \left( -\frac{i}{2} \tau_j (\hat{n}_j \cdot \vec{\sigma}_j) \right) \quad (17)$$

en la que definimos la rotación asociada al qubit  $j$  proporcionando el eje de giro<sup>11</sup>  $\hat{n}_j$  y la intensidad de la rotación  $\tau_j$ . Nótese que los operadores que actúan en distintos qubits commutan, pues no actúan en el mismo espacio vectorial, luego la rotación  $\mathcal{U}$  factoriza como un producto de rotaciones locales. Ver Ecuación (17).

Por tanto, el número total de parámetros de este *ansatz* es  $3N$ , ya que para cada qubit disponemos de tres parámetros:  $\{\tau_j, \theta_j, \varphi_j\}$ . Es decir, el número de parámetros escala de forma polinómica, y no exponencial, con el tamaño del sistema. Así mismo, el número de puertas necesarias para implementar dicho *ansatz* en un circuito cuántico también crece polinómicamente con el número de qubits del sistema. En concreto, la implementación de los estados variacionales en un circuito es inmediata ya que simplemente es necesario una puerta en cada qubit que realice una rotación genérica en la esfera de Bloch, es decir, se requiere de  $N$  puertas en total.

No obstante, la estructura del *ansatz* de estados atómicos coherentes (CS) es muy sencilla y no introduce ningún entrelazamiento entre los qubits, pues estos rotan por separado. Por tanto, no es capaz de describir correctamente el hamiltoniano (15) que tiene acoplos todos con todos.

En este contexto, surge el *ansatz* de estados atómicos coherentes generalizados (GCS) [15],  $|\psi\rangle$ , que no es sino una generalización del *ansatz* de estados atómicos coherentes (CS):

$$|\psi(\boldsymbol{\alpha}, \mathbf{M}, \boldsymbol{\beta})\rangle \equiv \mathcal{U}(\boldsymbol{\beta}) \mathcal{V}(\mathbf{M}) \underbrace{|\phi(\boldsymbol{\alpha})\rangle}_{CS} = \mathcal{U}(\boldsymbol{\beta}) \mathcal{V}(\mathbf{M}) \mathcal{U}(\boldsymbol{\alpha}) |0\rangle , \quad (18)$$

donde la matriz de correlación es:

$$\mathcal{V} \equiv \exp \left( -\frac{i}{4} \sum_{n < m} M_{nm} \sigma_n^z \sigma_m^z \right) \quad (19)$$

siendo  $M$  una matriz simétrica real con la diagonal principal nula. Luego, el número de parámetros que introduce esta matriz es  $N(N - 1)/2$ .

En definitiva, el *ansatz* de estados coherentes generalizados (GCS) está compuesto por una rotación local inicial, seguido por una matriz de entrelazamiento y una rotación local final. Nótese que la última rotación asegura que la elección de la dirección  $Z$  de las matrices de Pauli en la matriz de correlación no presenta una dirección especial (preferente) en la esfera de Bloch. Además, es fundamental destacar que  $\boldsymbol{\alpha}$  y  $\boldsymbol{\beta}$  son el conjunto de parámetros asociados a la primera y segunda rotación local respectivamente y, por tanto, no presentan ninguna relación.

---

<sup>11</sup>Expresado en coordenadas polares:  $\hat{n}_j = (\sin(\theta_j) \cos(\varphi_j), \sin(\theta_j) \sin(\varphi_j), \cos(\theta_j))$ .

La virtud del *ansatz* de estados coherentes generalizados (GCS) es que mantiene las ventajas de los estados atómicos coherentes (CS), ya que el número total de parámetros<sup>12</sup> escala de forma polinómica con el tamaño del sistema y su implementación en un circuito cuántico es eficiente. Además, introduce correlaciones en el sistema gracias a la matriz de entrelazamiento, necesarias para el problema que tratamos de abordar.

## 5.2 Implementación del *ansatz* GCS en un circuito cuántico

En primer lugar, expresamos la matriz de rotación  $\mathcal{U}$  de forma matricial con el fin de expresarla en función de puertas cuánticas. Consideremos ahora la acción de la rotación en un qubit arbitrario  $j$ , pudiendo prescindir así del subíndice  $j$  para aligerar la notación. Notando que  $(\hat{n}\vec{\sigma})$  es una matriz involutiva, es decir que  $(\hat{n}\vec{\sigma}) = I$ , podemos descomponer la rotación local en un seno y coseno.

$$\mathcal{U} = \exp\left(-\frac{i}{2}\tau(\hat{n} \cdot \vec{\sigma})\right) = -i(\hat{n}\vec{\sigma}) \sin\left(\frac{\tau}{2}\right) + \cos\left(\frac{\tau}{2}\right) I \quad (20)$$

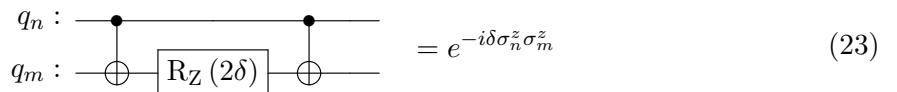
Por tanto, usando la base computacional y expresando  $\hat{n}$  en coordenadas polares, se obtiene que:

$$\mathcal{U}(\tau, \theta, \varphi) = \begin{pmatrix} \cos\left(\frac{\tau}{2}\right) - i \sin\left(\frac{\tau}{2}\right) \cos(\theta) & -i \sin\left(\frac{\tau}{2}\right) \sin(\theta) e^{-i\varphi} \\ -i \sin\left(\frac{\tau}{2}\right) \sin(\theta) e^{i\varphi} & \cos\left(\frac{\tau}{2}\right) + i \sin\left(\frac{\tau}{2}\right) \cos(\theta) \end{pmatrix} \quad (21)$$

Tal y como hemos mencionado en la Sección 3, vamos a usar *qiskit* para llevar a cabo el circuito cuántico. Por ende, cabe desatacar que la puerta nativa correspondiente a una rotación genérica en la esfera de Bloch que ofrece *qiskit* no tiene la misma forma matricial que (21), pues en la documentación de *qiskit* [6] se describe la rotación usando los 3 ángulos de Euler<sup>13</sup>. Por tanto, es necesario definir una puerta personalizada que se comporte como la matriz  $\mathcal{U}$ .

$$q : \boxed{\mathcal{U}(\tau, \theta, \varphi)} \equiv \begin{pmatrix} \cos\left(\frac{\tau}{2}\right) - i \sin\left(\frac{\tau}{2}\right) \cos(\theta) & -i \sin\left(\frac{\tau}{2}\right) \sin(\theta) e^{-i\varphi} \\ -i \sin\left(\frac{\tau}{2}\right) \sin(\theta) e^{i\varphi} & \cos\left(\frac{\tau}{2}\right) + i \sin\left(\frac{\tau}{2}\right) \cos(\theta) \end{pmatrix} \quad (22)$$

En segundo lugar, vamos a expresar la matriz de correlación,  $\mathcal{V}$ , en función de puertas cuánticas. Comenzamos notando que el término  $\exp(-i\delta\sigma_n^z\sigma_m^z)$  es equivalente a la acción de las puertas CNOT- $R_z$ -CNOT<sup>14</sup> [16].



Vamos a demostrar el resultado calculando la forma matricial del circuito cuántico, y verificando que coincide con  $\exp(-i\delta\sigma_n^z\sigma_m^z)$ . Nótese que al disponer de dos qubits, la base computacional no es sino los productos tensoriales:  $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ . Recordamos que, en esta base, las puertas  $R_z$  y CNOT son:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad I \otimes R_z(2\delta) = \begin{pmatrix} e^{-i\delta} & 0 & 0 & 0 \\ 0 & e^{i\delta} & 0 & 0 \\ 0 & 0 & e^{-i\delta} & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{pmatrix} \quad (24)$$

<sup>12</sup>El número total de parámetros del *ansatz* GCS es:  $6N + \frac{N(N-1)}{2}$ .

<sup>13</sup> $U(\tau, \theta, \varphi) = \begin{pmatrix} \cos\left(\frac{\tau}{2}\right) & -\sin\left(\frac{\tau}{2}\right)e^{i\varphi} \\ \sin\left(\frac{\tau}{2}\right)e^{i\theta} & \cos\left(\frac{\tau}{2}\right)e^{i(\theta+\varphi)} \end{pmatrix}$

<sup>14</sup>Sin entrar en detalles, destacamos que las rotaciones en un qubit y la puerta CNOT pueden ser implementadas experimentalmente a través de qubits superconductores con interacciones de resonancia cruzada.

$$\text{donde } R_z(\lambda) = \begin{pmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{pmatrix}.$$

Finalmente, realizando explícitamente el producto CNOT- $R_z$ -CNOT recuperamos la expresión matricial de  $\exp(-i\delta\sigma_n^z\sigma_m^z)$ .

$$\text{CNOT} \cdot R_z(2\delta) \cdot \text{CNOT} = \begin{pmatrix} e^{-i\delta} & 0 & 0 & 0 \\ 0 & e^{i\delta} & 0 & 0 \\ 0 & 0 & e^{i\delta} & 0 \\ 0 & 0 & 0 & e^{-i\delta} \end{pmatrix} = e^{-i\delta\sigma_n^z\sigma_m^z} \quad (25)$$

Una vez estudiada la correspondencia entre  $\exp(-i\delta\sigma_n^z\sigma_m^z)$  y las puertas CNOT- $R_z$ -CNOT, la implementación de la matriz de correlación,  $\mathcal{V}$ , es inmediata. Concretamente, hay que colocar dicho conjunto de puertas para todos los pares de qubits en el circuito, tal y como se recoge en la Figura 4.

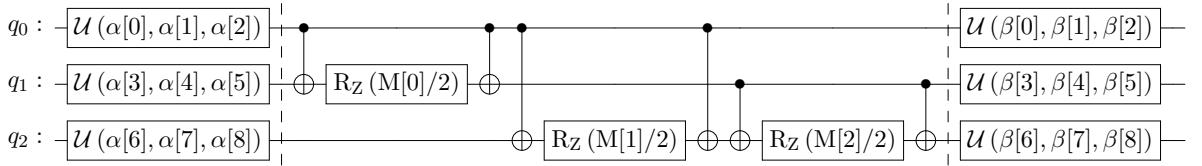


Figura 4: Implementación del *ansatz* GCS con 3 qubits (1 *layer*)

En la Figura 4 se recoge un circuito cuántico con 3 qubits correspondiente al *ansatz* de estados atómicos coherentes generalizados (GCS). Con el fin de poder visualizar las distintas partes del circuito, hemos separado mediante barreras las puertas correspondientes a cada matriz de (18).

Los circuitos cuánticos parametrizados, es decir las redes neuronales cuánticas, se caracterizan por estar compuestas de capas (“layers”) formadas por puertas unitarias que son repetidas a lo largo del circuito. En la Figura 4 se recoge un único bloque de puertas, luego estamos ante la red neuronal correspondiente a una *layer*. Añadir capas al circuito cuántico aumenta el número de parámetros, lo que se traduce en mayor complejidad y expresividad. Sin embargo, puede suponer un detrimiento en la entrenabilidad.

Finalmente, tal y como se ha comentado en la Sección 3, destacamos que estamos trabajando con un *heuristic ansatz*. Por tanto, el *ansatz* GCS es apropiado para resolver cualquier hamiltoniano que pueda ser implementado en un VQE, incluso aquellos que presenten interacciones todos con todos.

## 6 Cálculo analítico de $\langle H \rangle$

En esta sección recogemos el resultado más importante del trabajo: el circuito cuántico asociado a una *layer* del *ansatz* GCS puede ser simulable clásicamente. La demostración del resultado consiste en calcular analíticamente el valor esperado del hamiltoniano recogido en (15) usando el estado GCS definido en (18). Gracias a que el hamiltoniano es una combinación lineal de matrices de Pauli, el valor esperado de este se reduce a calcular  $\langle \sigma_n^\alpha \rangle$  y  $\langle \sigma_n^z \sigma_m^z \rangle$ .

### 6.1 Cálculo de $\langle \sigma_n^\alpha \rangle$

En primer lugar, recogemos la forma que tiene  $\langle \sigma_n^\alpha \rangle$  en la Ecuación (26).

$$\langle \sigma_n^\alpha \rangle = \langle \phi(\boldsymbol{\alpha}) | \mathcal{V}^\dagger \mathcal{U}^\dagger(\boldsymbol{\beta}) \sigma_n^\alpha \mathcal{U}(\boldsymbol{\beta}) \mathcal{V} | \phi(\boldsymbol{\alpha}) \rangle \quad (26)$$

Vamos a realizar los cálculos por partes. Primero, nos centramos en la primera rotación local, cuyo resultado sabemos que es:  $\mathcal{U}^\dagger \sigma_n^\alpha \mathcal{U} = \sum_\beta \lambda_n^{\alpha\beta} \sigma_n^\beta$ . Ver Anexo A. Luego, obtenemos que:

$$\langle \sigma_n^\alpha \rangle = \sum_{\beta=x,y,z} \lambda_n^{\alpha\beta} \langle \phi(\boldsymbol{\alpha}) | \mathcal{V}^\dagger \sigma_n^\beta \mathcal{V} | \phi(\boldsymbol{\alpha}) \rangle \quad (27)$$

Por tanto, necesitamos calcular  $\mathcal{V}^\dagger \sigma_n^\beta \mathcal{V}$ . Nótese que el caso  $\beta = z$  es trivial, pues  $\sigma_n^z$  y  $\mathcal{V}$  commutan y, por ende,  $\mathcal{V}^\dagger \sigma_n^z \mathcal{V} = \sigma_n^z$ . No obstante, para calcular el resto de casos es necesario introducir los operadores escalera asociados a la matrices de Pauli,  $\sigma_n^+$  y  $\sigma_n^-$ .

$$\begin{aligned} \sigma_n^+ &= |0\rangle_n \langle 1|_n = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ \sigma_n^- &= |1\rangle_n \langle 0|_n = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \end{aligned} \quad (28)$$

De forma que se verifica que  $\sigma_n^x = (\sigma_n^+ + \sigma_n^-)$  y  $\sigma_n^y = -i(\sigma_n^+ - \sigma_n^-)$ .

Por consiguiente, al introducir los operadores escalera, hemos conseguido que  $\mathcal{V}^\dagger \sigma_n^{x,y} \mathcal{V}$  se reduzca a  $\mathcal{V}^\dagger \sigma_n^\pm \mathcal{V}$ . Vamos a realizar el cálculo de  $\mathcal{V}^\dagger \sigma_n^+ \mathcal{V}$ :

$$\begin{aligned} \mathcal{V}^\dagger \sigma_n^+ \mathcal{V} &= \exp \left( \frac{i}{4} \sum_{p < j} M_{pj} \sigma_p^z \sigma_j^z \right) (|0\rangle_n \langle 1|_n) \exp \left( -\frac{i}{4} \sum_{p < j} M_{pj} \sigma_p^z \sigma_j^z \right) \\ &= \exp \left( \frac{i}{4} \sum_{p \neq n} M_{np} \sigma_p^z \right) (|0\rangle_n \langle 1|_n) \exp \left( \frac{i}{4} \sum_{p \neq n} M_{np} \sigma_p^z \right) \\ &= \exp \left( \frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z \right) \sigma_n^+ \end{aligned} \quad (29)$$

donde en la segunda igualdad hemos empleado que  $|0\rangle_n$  es un autoestado de  $\sigma_n^z$  con autovalor +1 y, consecuentemente, se verifica que  $\exp(\sigma_n^z \sigma_p^z) |0\rangle_n = \exp(\sigma_p^z) |0\rangle_n$ . Asimismo, el estado  $|1\rangle_n$  es autoestado de  $\sigma_n^z$  con autovalor -1 y cumple que  $\exp(\sigma_n^z \sigma_p^z) |1\rangle_n = \exp(-\sigma_p^z) |1\rangle_n$ .

Nótese que es análogo el desarrollo para obtener  $\mathcal{V}^\dagger \sigma_n^- \mathcal{V}$ .

$$\mathcal{V}^\dagger \sigma_n^- \mathcal{V} = \exp \left( -\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z \right) \sigma_n^- \quad (30)$$

Por tanto, con la introducción de los operadores escalera, el valor esperado de  $\sigma_n^\alpha$  resulta en la Ecuación (31).

$$\begin{aligned}\langle \sigma_n^\alpha \rangle &= \langle \phi(\boldsymbol{\alpha}) | + \lambda_n^{\alpha x}(\boldsymbol{\beta}) \left[ e^{\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^+ + e^{-\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^- \right] \\ &\quad - \lambda_n^{\alpha y}(\boldsymbol{\beta}) i \left[ e^{\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^+ - e^{-\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^- \right] \\ &\quad + \lambda_n^{\alpha z}(\boldsymbol{\beta}) \sigma_n^z |\phi(\boldsymbol{\alpha})\rangle\end{aligned}\quad (31)$$

El siguiente paso para calcular  $\langle \sigma_n^\alpha \rangle$  es hallar  $\langle \phi(\boldsymbol{\alpha}) | \sigma_n^\alpha | \phi(\boldsymbol{\alpha}) \rangle$ .

$$\langle \phi(\boldsymbol{\alpha}) | \sigma_n^\alpha | \phi(\boldsymbol{\alpha}) \rangle = \sum_{\beta=x,y,z} \lambda_n^{\alpha \beta}(\boldsymbol{\alpha}) \langle 0 | \sigma_n^\beta | 0 \rangle = \lambda_n^{\alpha z}(\boldsymbol{\alpha}) \quad (32)$$

A continuación, notando que las matrices de Pauli son involutivas, vamos a descomponer las exponenciales en senos y cosenos para poder así realizar el valor esperado con respecto al estado  $|\phi(\boldsymbol{\alpha})\rangle$ :

$$\begin{aligned}\langle \sigma_n^\alpha \rangle &= + \lambda_n^{\alpha x}(\boldsymbol{\beta}) \left[ + \prod_{p \neq n} \left( i \sin \left( \frac{M_{np}}{2} \right) \lambda_p^{zz}(\boldsymbol{\alpha}) + \cos \left( \frac{M_{np}}{2} \right) \right) \left( \frac{\lambda_n^{xz}(\boldsymbol{\alpha}) + i \lambda_n^{yz}(\boldsymbol{\alpha})}{2} \right) \right. \\ &\quad \left. + \prod_{p \neq n} \left( -i \sin \left( \frac{M_{np}}{2} \right) \lambda_p^{zz}(\boldsymbol{\alpha}) + \cos \left( \frac{M_{np}}{2} \right) \right) \left( \frac{\lambda_n^{xz}(\boldsymbol{\alpha}) - i \lambda_n^{yz}(\boldsymbol{\alpha})}{2} \right) \right] \\ &\quad + \lambda_n^{\alpha y}(\boldsymbol{\beta}) \left[ + \prod_{p \neq n} \left( i \sin \left( \frac{M_{np}}{2} \right) \lambda_p^{zz}(\boldsymbol{\alpha}) + \cos \left( \frac{M_{np}}{2} \right) \right) \left( \frac{-i \lambda_n^{xz}(\boldsymbol{\alpha}) + \lambda_n^{yz}(\boldsymbol{\alpha})}{2} \right) \right. \\ &\quad \left. + \prod_{p \neq n} \left( -i \sin \left( \frac{M_{np}}{2} \right) \lambda_p^{zz}(\boldsymbol{\alpha}) + \cos \left( \frac{M_{np}}{2} \right) \right) \left( \frac{i \lambda_n^{xz}(\boldsymbol{\alpha}) + \lambda_n^{yz}(\boldsymbol{\alpha})}{2} \right) \right] \\ &\quad + \lambda_n^{\alpha z}(\boldsymbol{\beta}) \lambda_n^{zz}(\boldsymbol{\alpha})\end{aligned}\quad (33)$$

donde hemos empleado que  $\sigma_n^+ = \frac{\sigma_n^x + i \sigma_n^y}{2}$  y  $\sigma_n^- = \frac{\sigma_n^x - i \sigma_n^y}{2}$ .

Vamos a simplificar la Ecuación (33) introduciendo el número complejo  $z$ :

$$z \equiv \prod_{p \neq n} \left( -i \sin \left( \frac{M_{np}}{2} \right) \lambda_p^{zz}(\boldsymbol{\alpha}) + \cos \left( \frac{M_{np}}{2} \right) \right) \quad (34)$$

que no es sino  $\langle \phi(\boldsymbol{\alpha}) | e^{-\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} | \phi(\boldsymbol{\alpha}) \rangle$ .

En último lugar, sustituyendo (34) en (33) y sacando factores comunes, se obtiene la expresión simplificada de  $\langle \sigma_n^\alpha \rangle$ .

$$\begin{aligned}\langle \sigma_n^\alpha \rangle &= [\lambda_n^{\alpha x}(\boldsymbol{\beta}) \lambda_n^{xz}(\boldsymbol{\alpha}) + \lambda_n^{\alpha y}(\boldsymbol{\beta}) \lambda_n^{yz}(\boldsymbol{\alpha})] \left( \frac{z + \bar{z}}{2} \right) \\ &\quad + [\lambda_n^{\alpha y}(\boldsymbol{\beta}) \lambda_n^{xz}(\boldsymbol{\alpha}) - \lambda_n^{\alpha x}(\boldsymbol{\beta}) \lambda_n^{yz}(\boldsymbol{\alpha})] i \left( \frac{z - \bar{z}}{2} \right) \\ &\quad + \lambda_n^{\alpha z}(\boldsymbol{\beta}) \lambda_n^{zz}(\boldsymbol{\alpha})\end{aligned}\quad (35)$$

Nótese que la expresión derivada es coherente pues es real.

## 6.2 Cálculo de $\langle \sigma_n^z \sigma_m^z \rangle$

Análogamente al caso anterior, comencemos escribiendo la forma que tiene  $\langle \sigma_n^z \sigma_m^z \rangle$ .

$$\langle \sigma_n^z \sigma_m^z \rangle = \langle \phi(\alpha) | \mathcal{V}^\dagger \mathcal{U}^\dagger(\beta) \sigma_n^z \sigma_m^z \mathcal{U}(\beta) \mathcal{V} | \phi(\alpha) \rangle \quad (36)$$

A continuación, teniendo en cuenta que los operadores  $\mathcal{U}$  y  $\mathcal{V}$  son unitarios, podemos escribir el valor esperado como:

$$\langle \sigma_n^z \sigma_m^z \rangle = \langle \phi(\alpha) | \left( \mathcal{V}^\dagger \mathcal{U}^\dagger(\beta) \sigma_n^z \mathcal{U}(\beta) \mathcal{V} \right) \left( \mathcal{V}^\dagger \mathcal{U}^\dagger(\beta) \sigma_m^z \mathcal{U}(\beta) \mathcal{V} \right) | \phi(\alpha) \rangle \quad (37)$$

Así, podemos seguir el mismo procedimiento que en la Sección 6.1 y obtener que:

$$\begin{aligned} \langle \sigma_n^z \sigma_m^z \rangle &= \langle \phi(\alpha) | \left( \lambda_n^{zx}(\beta) \left[ e^{\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^+ + e^{-\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^- \right] \right. \\ &\quad - \lambda_n^{zy}(\beta) i \left[ e^{\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^+ - e^{-\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^- \right] + \lambda_n^{zz}(\beta) \sigma_n^z \Big) \times \\ &\quad \times \left( \lambda_m^{zx}(\beta) \left[ e^{\frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^+ + e^{-\frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^- \right] \right. \\ &\quad \left. \left. - \lambda_m^{zy}(\beta) i \left[ e^{\frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^+ - e^{-\frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^- \right] + \lambda_m^{zz}(\beta) \sigma_m^z \right) | \phi(\alpha) \rangle \right) \end{aligned} \quad (38)$$

de manera que en  $\langle \sigma_n^z \sigma_m^z \rangle$  vamos a tener dos tipos de productos.

En primer lugar, los términos asociados a  $\lambda^{z\gamma}(\beta)$  con  $\gamma \in \{x, y\}$ , que tienen la siguiente forma:

$$e^{\omega \frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^\omega e^{\nu \frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^\nu = \sigma_n^\omega \sigma_m^\nu e^{\sum_{\tau \neq n, m} (\omega M_{n\tau} + \nu M_{m\tau}) \sigma_\tau^z} \quad (39)$$

donde  $\omega, \nu \in \{+, -\}$ .

Vamos a demostrar la Ecuación (39) desarrollando el caso concreto en el que  $\omega = +$  y  $\nu = +$ , pues de esta forma podemos mostrar explícitamente cómo se simplifican los términos. Nótese que, en este caso, estudiar un caso concreto no presenta una pérdida de generalidad pues para el resto de casos el desarrollo es análogo.

Primero, recordando que operadores que actúan sobre distintos qubits commutan y que  $[\sigma^z, \sigma^z] = 0$ , podemos reorganizar los términos tal y como se indica en la Ecuación (40).

$$e^{\frac{i}{2} \sum_{p \neq n} M_{np} \sigma_p^z} \sigma_n^+ e^{\frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^+ = \left( \sigma_n^+ e^{\frac{i}{2} M_{mn} \sigma_n^z} \right) \left( e^{\frac{i}{2} M_{nm} \sigma_m^z} \sigma_m^+ \right) e^{\sum_{\tau \neq n, m} (M_{n\tau} + M_{m\tau}) \sigma_\tau^z} \quad (40)$$

Finalmente, usando la definición de los operadores escalera, ver Ecuación (28), podemos simplificar las exponenciales que se encuentran en los paréntesis, llegando así a la expresión deseada.

$$(|0\rangle_n \langle 1|_n) \underbrace{e^{-\frac{i}{2} M_{mn}} e^{\frac{i}{2} M_{nm}}}_{=1} (|0\rangle_m \langle 1|_m) e^{\sum_{\tau \neq n, m} (M_{n\tau} + M_{m\tau}) \sigma_\tau^z} = \sigma_n^+ \sigma_m^+ e^{\sum_{\tau \neq n, m} (M_{n\tau} + M_{m\tau}) \sigma_\tau^z} \quad (41)$$

Por otro lado, el segundo tipo de términos que se encuentran en la Ecuación (38) son los asociados a  $\lambda^{zz}(\beta)$ . Dichos términos son triviales de tratar, pues únicamente hay que desarrollar las exponenciales en senos y cosenos, tal y como se indica en la Ecuación (42).

$$\sigma_n^z e^{\frac{i}{2} \sum_{l \neq m} M_{ml} \sigma_l^z} \sigma_m^+ = \left( \cos \left( \frac{M_{mn}}{2} \right) \sigma_n^z + i \sin \left( \frac{M_{mn}}{2} \right) \right) e^{\frac{i}{2} \sum_{\tau \neq n, m} M_{m\tau} \sigma_\tau^z} \sigma_m^+ \quad (42)$$

Donde hemos empleado que las matrices de Pauli son involutivas.

Recapitulando, ya tenemos la forma de los productos que aparecen en la Ecuación (38). Es fundamental destacar que las Ecuaciones (39) y (42) no son sino la base en la que se sustenta la derivación de  $\langle \sigma_n^z \sigma_m^z \rangle$ . Esto es debido a que, a partir de dichas ecuaciones, podemos llevar a

cabo un procedimiento análogo al recogido en la Sección 6.1 e introducir los siguientes números complejos:

$$\begin{cases} s \equiv \prod_{\tau \neq n,m} (\cos(\frac{M_{n\tau}+M_{m\tau}}{2}) + i \sin(\frac{M_{n\tau}+M_{m\tau}}{2}) \lambda_{\tau}^{zz}(\boldsymbol{\alpha})) \\ v \equiv \prod_{\tau \neq n,m} (\cos(\frac{M_{n\tau}-M_{m\tau}}{2}) + i \sin(\frac{M_{n\tau}-M_{m\tau}}{2}) \lambda_{\tau}^{zz}(\boldsymbol{\alpha})) \\ p \equiv (\cos(\frac{M_{nm}}{2}) \lambda_m^{zz}(\boldsymbol{\alpha}) + i \sin(\frac{M_{nm}}{2})) \cdot \prod_{\tau \neq n,m} (\cos(\frac{M_{n\tau}}{2}) + i \sin(\frac{M_{n\tau}}{2}) \lambda_{\tau}^{zz}(\boldsymbol{\alpha})) \\ t \equiv (\cos(\frac{M_{mn}}{2}) \lambda_n^{zz}(\boldsymbol{\alpha}) + i \sin(\frac{M_{mn}}{2})) \cdot \prod_{\tau \neq n,m} (\cos(\frac{M_{m\tau}}{2}) + i \sin(\frac{M_{m\tau}}{2}) \lambda_{\tau}^{zz}(\boldsymbol{\alpha})) \end{cases} \quad (43)$$

con el fin de simplificar  $\langle \sigma_n^z \sigma_m^z \rangle$ . Nótese que los números complejos  $s$  y  $v$  están asociados a los términos con  $\lambda^{z\gamma}(\boldsymbol{\beta})$ , mientras que  $p$  y  $t$  corresponden a los términos con  $\lambda^{zz}(\boldsymbol{\beta})$ .

Con la introducción de los números complejos recogidos en la Ecuación (43) y sacando factores comunes, se obtiene la expresión simplificada de  $\langle \sigma_n^z \sigma_m^z \rangle$ .

$$\begin{aligned} \langle \sigma_n^z \sigma_m^z \rangle = & \left[ A \cdot C + B \cdot D \right] \left( \frac{s + \bar{s}}{4} \right) + i \left[ B \cdot C - A \cdot D \right] \left( \frac{s - \bar{s}}{4} \right) \\ & + \left[ E \cdot G + F \cdot H \right] \left( \frac{v + \bar{v}}{4} \right) + i \left[ F \cdot G - E \cdot H \right] \left( \frac{v - \bar{v}}{4} \right) \\ & + \left[ I \cdot \lambda_n^{xz}(\boldsymbol{\alpha}) + J \cdot \lambda_n^{yz}(\boldsymbol{\alpha}) \right] \left( \frac{p + \bar{p}}{2} \right) + i \left[ I \cdot \lambda_n^{yz}(\boldsymbol{\alpha}) - J \cdot \lambda_n^{xz}(\boldsymbol{\alpha}) \right] \left( \frac{p - \bar{p}}{2} \right) \\ & + \left[ K \cdot \lambda_m^{xz}(\boldsymbol{\alpha}) + L \cdot \lambda_m^{yz}(\boldsymbol{\alpha}) \right] \left( \frac{t + \bar{t}}{2} \right) + i \left[ K \cdot \lambda_m^{yz}(\boldsymbol{\alpha}) - L \cdot \lambda_m^{xz}(\boldsymbol{\alpha}) \right] \left( \frac{t - \bar{t}}{2} \right) \\ & + \left[ \lambda_n^{zz}(\boldsymbol{\beta}) \lambda_m^{zz}(\boldsymbol{\beta}) \lambda_n^{zz}(\boldsymbol{\alpha}) \lambda_m^{zz}(\boldsymbol{\alpha}) \right] \end{aligned} \quad (44)$$

Siendo:

$$\begin{cases} A \equiv \lambda_n^{xz}(\boldsymbol{\alpha}) \lambda_m^{xz}(\boldsymbol{\alpha}) - \lambda_n^{yz}(\boldsymbol{\alpha}) \lambda_m^{yz}(\boldsymbol{\alpha}) & G \equiv \lambda_n^{zx}(\boldsymbol{\beta}) \lambda_m^{zx}(\boldsymbol{\beta}) + \lambda_n^{zy}(\boldsymbol{\beta}) \lambda_m^{zy}(\boldsymbol{\beta}) \\ B \equiv \lambda_n^{xz}(\boldsymbol{\alpha}) \lambda_m^{yz}(\boldsymbol{\alpha}) + \lambda_n^{yz}(\boldsymbol{\alpha}) \lambda_m^{xz}(\boldsymbol{\alpha}) & H \equiv \lambda_n^{zy}(\boldsymbol{\beta}) \lambda_m^{zx}(\boldsymbol{\beta}) - \lambda_n^{zx}(\boldsymbol{\beta}) \lambda_m^{zy}(\boldsymbol{\beta}) \\ C \equiv \lambda_n^{zx}(\boldsymbol{\beta}) \lambda_m^{zx}(\boldsymbol{\beta}) - \lambda_n^{zy}(\boldsymbol{\beta}) \lambda_m^{zy}(\boldsymbol{\beta}) & I \equiv \lambda_n^{zx}(\boldsymbol{\beta}) \lambda_m^{zz}(\boldsymbol{\beta}) \\ D \equiv \lambda_n^{zx}(\boldsymbol{\beta}) \lambda_m^{zy}(\boldsymbol{\beta}) + \lambda_n^{zy}(\boldsymbol{\beta}) \lambda_m^{zx}(\boldsymbol{\beta}) & J \equiv \lambda_n^{zy}(\boldsymbol{\beta}) \lambda_m^{zz}(\boldsymbol{\beta}) \\ E \equiv \lambda_n^{xz}(\boldsymbol{\alpha}) \lambda_m^{xz}(\boldsymbol{\alpha}) + \lambda_n^{yz}(\boldsymbol{\alpha}) \lambda_m^{yz}(\boldsymbol{\alpha}) & K \equiv \lambda_n^{zz}(\boldsymbol{\beta}) \lambda_m^{zx}(\boldsymbol{\beta}) \\ F \equiv \lambda_n^{yz}(\boldsymbol{\alpha}) \lambda_m^{xz}(\boldsymbol{\alpha}) - \lambda_n^{xz}(\boldsymbol{\alpha}) \lambda_m^{yz}(\boldsymbol{\alpha}) & L \equiv \lambda_n^{zz}(\boldsymbol{\beta}) \lambda_m^{zy}(\boldsymbol{\beta}) \end{cases} \quad (45)$$

Nótese que la expresión obtenida<sup>15</sup> es coherente pues es real e invariante bajo el intercambio de los índices  $n$  y  $m$ .

En definitiva, hemos sido capaces de calcular el valor esperado del hamiltoniano (15) usando el *ansatz* GCS (18). Por tanto, podemos introducir la expresión del valor esperado en un ordenador clásico y encontrar su mínimo como si se tratase de un problema de optimización de una función multivariante. En otras palabras, es viable simular con un ordenador clásico el circuito cuántico del *ansatz* GCS correspondiente a una *layer*. Los algoritmos que se fundamentan en el formalismo matemático de la mecánica cuántica pero están diseñados para ser ejecutados en dispositivos clásicos se denominan algoritmos *quantum-inspired* (QI) [4].

Como nota final, destacamos que a partir de  $\langle \sigma_n^{\alpha} \rangle$  y  $\langle \sigma_n^z \sigma_m^z \rangle$ , podemos calcular clásicamente el valor esperado de cualquier hamiltoniano que pueda descomponerse en matrices de Pauli, y presentes términos de interacción de la forma  $\sigma_n^z \sigma_m^z$ . En este trabajo nos vamos a restringir al estudio del modelo *long-range* de Ising cuántico, pero el procedimiento realizado en esta sección es mucho más general. Esto es una consecuencia inmediata de trabajar con un *heuristic ansatz*.

<sup>15</sup>El resultado es inmediato de verificar a partir de la expresión explícita.

## 7 Resultados

Vamos a encontrar el estado fundamental del modelo *long-range* de Ising cuántico (15) usando el *ansatz* GCS (18) a través de dos métodos distintos: el “variational quantum eigensolver” (VQE) y el algoritmo *quantum-inspired* diseñado en la Sección 6. Debido a que el circuito cuántico de una *layer* del VQE es simulable clásicamente, también vamos a trabajar con la red neuronal correspondiente a dos *layers*, para poder disponer así de un método completamente cuántico que no pueda ser emulado en un dispositivo clásico.

El objetivo de la sección es comparar el rendimiento entre el circuito cuántico con una *layer* del VQE y el algoritmo *quantum-inspired*. Además, discutiremos si añadir una capa más al VQE presenta una mejora sustancial frente a una única *layer* o al algoritmo *quantum-inspired*.

Por otro lado, recordamos que una parte central del VQE es la elección del optimizador clásico. Ver Sección 3. Los optimizadores que vamos a usar en este trabajo son: SLSQP [7] y COBYLA [9]. El primero es un optimizador “Gradient-Based” mientras que el segundo es “Gradient-Free”, de forma que el segundo objetivo de la sección es estudiar la diferencia en rendimiento entre ambos tipos de optimizador. Además, dichos optimizadores permiten introducir cotas a las variables. En nuestro caso, el dominio de los ángulos asociados a las rotaciones locales es:

$$\begin{cases} \tau \in [0, \pi] \\ \theta \in [0, \pi] \\ \varphi \in [0, 2\pi] \end{cases} \quad (46)$$

Mientras que los ángulos correspondientes a la matriz de correlación están definidos en el rango:

$$M \in [0, 2\pi] \quad (47)$$

La sección se divide en dos partes. Primero, realizamos un estudio previo en el que comprobamos que las redes neuronales cuánticas y el algoritmo *quantum-inspired* funcionan correctamente con un modelo trivial. En segundo lugar, resolvemos el hamiltoniano *long-range* de Ising cuántico (15). Todas las simulaciones se han realizado para cadenas compuestas por  $N = 8$  qubits.

### 7.1 Estudio Previo

Con el fin de verificar que las redes neuronales cuánticas y el algoritmo *quantum-inspired* se comportan de forma deseada, vamos a resolver el modelo de Ising (15) en el caso de  $J = 0$ . Dicho sistema es trivial pues es el modelo de Ising clásico en el eje  $X$ . Ver Sección 4. Por tanto, es inmediato ver que la energía del estado fundamental es:

$$E_0 \simeq -N = -8 \quad (48)$$

donde la energía no es exactamente  $-N$  debido a la pequeña perturbación introducida por el campo magnético longitudinal,  $\epsilon$ .

A lo largo de las simulaciones, vamos a denotar como “L=1” y “L=2” a la red neuronal con una y dos *layers*, respectivamente. Además, la línea naranja denominada “Ansatz Clásico” está asociada al algoritmo *quantum-inspired*. Finalmente, las líneas azules corresponden a la energía fundamental y del primer excitado del sistema, que son calculadas empleando diagonalización exacta<sup>16</sup> (ED) [12].

En la Figura 5 recogemos la convergencia<sup>17</sup> de la energía<sup>18</sup> del sistema, junto a su error relativo, frente al número de iteraciones realizadas por el optimizador COBYLA tanto para la red neuronal

<sup>16</sup>Nótese que el cálculo de las energías mediante ED es posible porque el número de qubits es limitado.

<sup>17</sup>Los tres métodos parten de una energía inicial aleatoria.

<sup>18</sup>Recordamos que expresamos la energía en unidades de  $h_x$ . Ver Sección 4.

con una y dos *layers*, como para el algoritmo *quantum-inspired*. Notamos que el error relativo se encuentra en escala logarítmica y lo definimos como:

$$\text{Error Relativo} = \frac{E_{\text{sim}} - E_{\text{ED}}}{E_{\text{ED}}} \quad (49)$$

donde  $E_{\text{sim}}$  corresponde a la energía obtenida mediante el método numérico correspondiente y  $E_{\text{ED}}$  es el estado fundamental obtenido por diagonalización exacta.

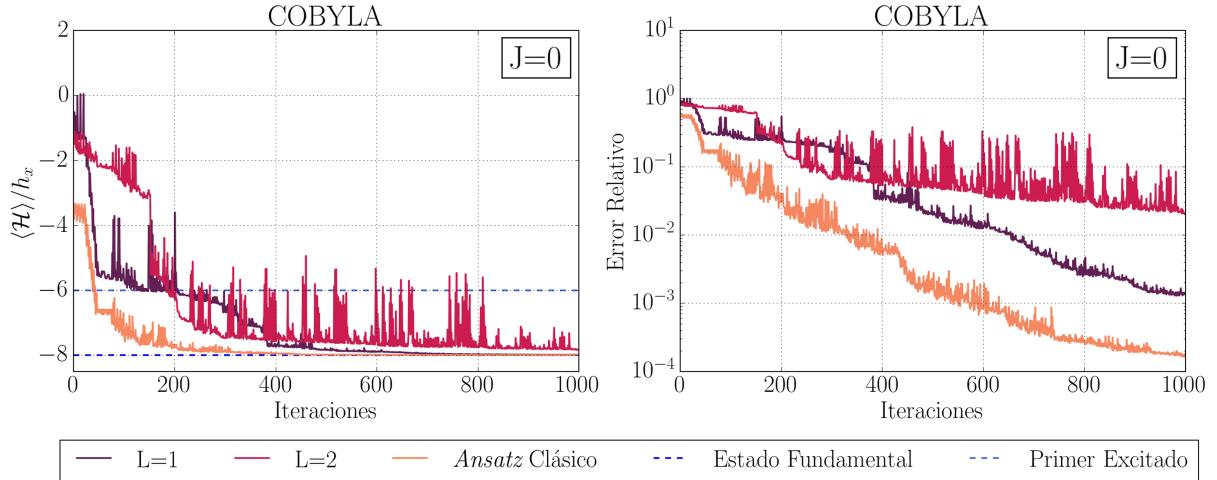


Figura 5: Convergencia de la energía y error relativo del sistema frente al número de iteraciones del optimizador COBYLA.

En primer lugar, vemos que los tres métodos convergen al estado fundamental del sistema, con lo que podemos concluir que están correctamente implementados.

El algoritmo clásico es el primero en converger al estado fundamental, con un error relativo del orden de  $10^{-4}$ . Dicho método es el más preciso de los tres, pues presenta un error que es un orden de magnitud menor que la red neuronal con una *layer*, y es incluso dos órdenes menor que el circuito cuántico con dos *layers*.

La red neuronal con dos *layers* converge antes que con una única *layer* pero, una vez encuentra el mínimo global, no es capaz de reducir el error relativo. De esta forma, el circuito cuántico con una *layer* presenta un mejor rendimiento que con dos *layers*.

Por otro lado, hemos repetido la simulación diez veces con el fin de obtener el tiempo de ejecución promedio junto a su desviación estándar de los tres métodos. Ver Tabla 1.

Tal y como era de esperar, la red neuronal con una *layer* presenta un menor tiempo de ejecución que con dos *layers*, ya que el número de parámetros aumenta al introducir capas en el circuito cuántico.

El tiempo de ejecución asociado al algoritmo *quantum-inspired* destaca por ser del orden de seis veces mayor que el de la red neuronal con una *layer*. Esto es mayoritariamente debido a la rudimentaria implementación del algoritmo clásico en el trabajo, que se ha llevado a cabo con la librería *Scipy* [17] de *Python*. Concretamente, hemos minimizado la energía del sistema con la función *minimize*. Por tanto, es vital notar que el tiempo de ejecución de este algoritmo puede disminuir considerablemente si se implementa de una manera más eficiente. Ver Sección 8.

Por su parte, es interesante verificar numéricamente que la expresión de la energía derivada en la Sección 6 es correcta. Para llevar esto a cabo, hemos introducido los valores de los parámetros

	L=1 (s)	L=2 (s)	<i>Ansatz</i> Clásico (s)
COBYLA	$22,2 \pm 1,8$	$39,2 \pm 2,5$	$122 \pm 3$
SLSQP	$21 \pm 5$	$141 \pm 13$	$81 \pm 18$

Tabla 1: Tiempo de ejecución, promediado a lo largo de 10 simulaciones, de la red neuronal cuántica con una y dos *layers*, y del algoritmo *quantum-inspired* usando los optimizadores clásicos COBYLA y SLSQP.

de la red neuronal con una *layer* en el algoritmo clásico, de forma que la energía calculada clásicamente debe coincidir con la del circuito cuántico.

$$\Delta E = |E_{\text{Clásico}} - E_{\text{VQA}}(\text{L}=1)| \quad (50)$$

Concretamente, hemos calculado la diferencia de energía para el valor de los parámetros en cada una de las iteraciones de la Figura 5 y se muestra en la Figura 6.

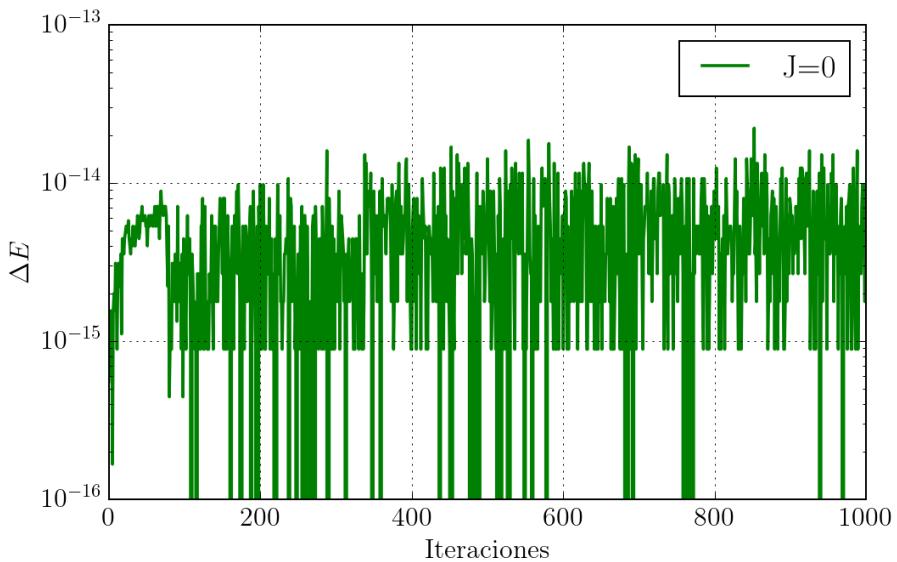


Figura 6: Classical vs 1 Layer

La diferencia de energía, representada en escala logarítmica, no toma un valor constante sino que fluctúa entre valores con órdenes de magnitud entre  $10^{-14}$  y  $10^{-16}$ . Dicho comportamiento es debido a la precisión de la variable *float* de *Python*, de forma que podemos asegurar<sup>19</sup> que los valores calculados clásicamente coinciden con los proporcionados por la red neuronal. Nótese que en el procedimiento para obtener la Figura 6 no hemos usado ningún optimizador, pues simplemente evaluamos la energía clásicamente usando los parámetros que usaba la red neuronal con una *layer* en la Figura 5.

Una vez estudiado el comportamiento de los tres métodos numéricos con el optimizador clásico COBYLA, vamos a repetir el mismo estudio pero usando el optimizador clásico SLSQP. En concreto, resolvemos el modelo de Ising clásico en el eje  $X$ , y recogemos la convergencia de la energía, junto a su error relativo, en función del número de iteraciones. Ver Figura 7.

<sup>19</sup>La diferencia de energía de los puntos que sobrepasan el límite inferior de la gráfica es, exactamente,  $\Delta E = 0.0$ .

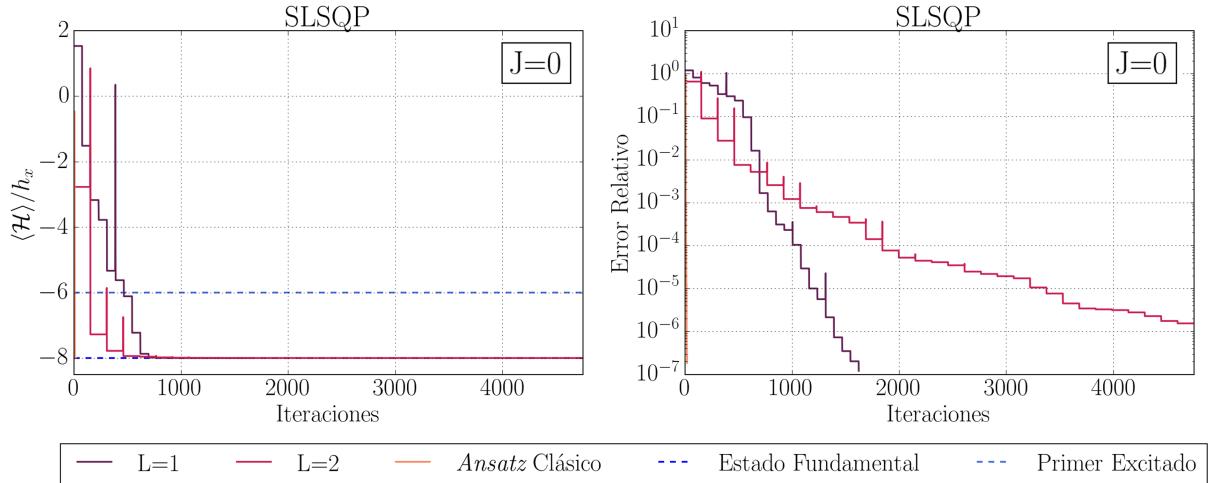


Figura 7: Convergencia de la energía y error relativo del sistema frente al número de iteraciones del optimizador SLSQP.

El error relativo obtenido con el optimizador SLSQP es, en el caso de la red neuronal con una y dos *layers*, hasta cuatro órdenes de magnitud menor que el error obtenido con COBYLA. Asimismo, el algoritmo *quantum-inspired* también presenta una mejoría notable al usar el optimizador SLSQP, pues su error disminuye tres órdenes de magnitud comparado con el error usando COBYLA.

El comportamiento de los tres métodos es similar usando ambos optimizadores: el algoritmo clásico es el primero en converger, y la red neuronal con dos *layers* encuentra el mínimo antes que con una *layer* pero no es capaz de reducir tanto el error relativo. De esta forma, la red neuronal con una capa y el algoritmo clásico presentan la mejor precisión, disponiendo de un error relativo del orden de  $10^{-7}$ , mientras que el circuito cuántico con dos *layers* presenta un error relativo del  $10^{-6}$ . No obstante, a diferencia que en el caso con COBYLA, las iteraciones llevadas a cabo por los tres métodos no son las mismas. Concretamente, el algoritmo clásico solo requiere 22 iteraciones para finalizar la optimización, mientras que la red neuronal con una y dos *layers* necesita 1625 y 4755 iteraciones, respectivamente.

El tiempo de ejecución de los métodos numéricos usando el optimizador SLSQP no sigue la misma tendencia que en el caso de COBYLA, pues la red neuronal con dos *layers* es el método más costoso. Concretamente, el circuito cuántico con dos capas es, aproximadamente, siete veces más lento que con una sola capa. Además, el tiempo de ejecución del algoritmo clásico es, del orden de, cuatro veces mayor que el de la red neuronal con una *layer*. Ver Tabla 1.

Al comparar los tiempos de ejecución entre los distintos optimizadores, vemos que la red neuronal con una *layer* presenta un tiempo similar en ambos casos. En cambio, el tiempo de ejecución de la red neuronal con dos *layers* se ve triplicado al usar el optimizador SLSQP, mientras que el algoritmo clásico sufre una reducción el tiempo de ejecución del 33 % al emplear dicho optimizador.

Asimismo, destacamos que la incertidumbre de los tiempos correspondientes al optimizador SLSQP son más elevados que los asociados a COBYLA. El algoritmo *quantum-inspired* sufre el mayor aumento en incertidumbre, que es hasta seis veces mayor.

En definitiva, los tres métodos presentados son capaces de resolver el modelo de Ising clásico, de forma que podemos abarcar problemas más complejos. Además, destacamos que el método más favorable es la red neuronal cuántica con una *layer* con el optimizador SLSQP, ya que obtiene el mínimo error relativo y presenta el tiempo de ejecución más reducido.

## 7.2 Resolución del modelo long-range de Ising cuántico

Esta sección constituye la parte más relevante del trabajo: la obtención del estado fundamental del modelo *long-range* de Ising cuántico (15) usando la red neuronal cuántica y el algoritmo *quantum-inspired*.

Tal y como hemos comentado en la Sección 4, el sistema (15) no es trivial y su comportamiento depende fuertemente del valor de  $\alpha$ . Por tanto, vamos a resolver el modelo para distintos valores de  $\alpha$ , con el fin de poder estudiar el sistema en sus tres distintas regiones: modelo de corto alcance ( $\alpha > 3$ ), de largo alcance débil ( $3 > \alpha > 1$ ) y de largo alcance fuerte ( $\alpha < 1$ ). En concreto, vamos a considerar los siguientes valores de  $\alpha$ : 0.5, 1.0, 2.5 y 6.0.

A continuación, vamos a exponer el método seguido para llevar a cabo la resolución del sistema. En primer lugar, fijamos el valor de  $\alpha$  del modelo. Segundo, resolvemos el caso en el que  $J = 0$ , ya que sabemos que los tres métodos son capaces de encontrar el estado fundamental, y guardamos el valor de los parámetros optimizados. Posteriormente, aumentamos el valor de  $J$  y, partiendo de los parámetros optimizados, iniciamos la resolución del modelo. Una vez finalizado, aumentamos  $J$  y repetimos el procedimiento sucesivamente. Nótese que la metodología empleada no es sino el *bootstrapping*. Ver Sección 3. En nuestro caso, vamos a simular el modelo desde  $J/h_x = -4$  hasta  $J/h_x = 10$ , en incrementos<sup>20</sup> de  $\Delta J/h_x = 0.01$ .

En la Figura 8 recogemos la energía del sistema para los distintos valores de  $\alpha$  junto a su error relativo en función de  $J/h_x$  calculada por la red neuronal y algoritmo clásico usando el optimizador COBYLA. Además, las líneas verdes corresponden a los valores críticos,  $J_c(\alpha)/h_x$ , donde tiene lugar la transición de fase cuántica de segundo orden. Nótese que dichos valores críticos se han obtenido mediante redes neuronales clásicas [13].

En primer lugar, vemos que la energía del sistema sigue la tendencia indicada por el estado fundamental obtenido con diagonalización exacta, de forma que los tres métodos son capaces de resolver el modelo (15).

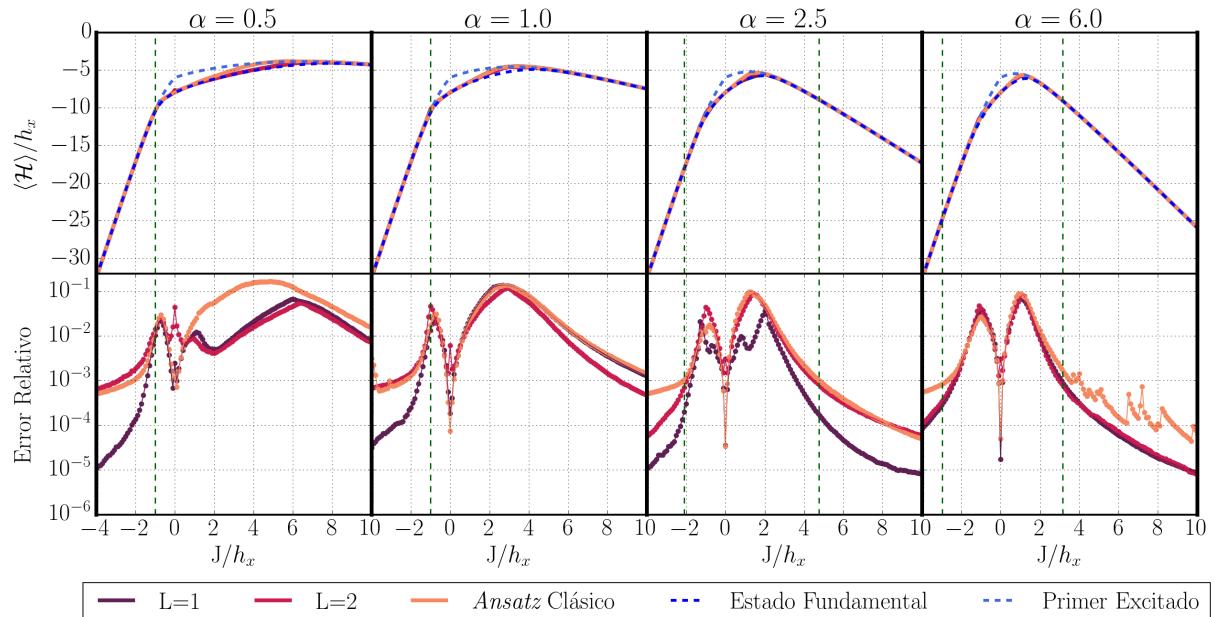


Figura 8: Resolución del estado fundamental del modelo (15), realizado por la red neuronal y algoritmo *quantum-inspired*, para distintos valores de  $\alpha$  usando el optimizador COBYLA.

<sup>20</sup>El número de optimizaciones en el proceso es 141.

A continuación, vamos a llevar a cabo un análisis del comportamiento de los tres métodos más riguroso. En concreto, centramos nuestra atención al error relativo, que está expresado en escala logarítmica. Para todos los valores de  $\alpha$ , el error sigue un comportamiento similar: es mínimo cuando  $J/h_x = 0$  y aumenta para los valores pequeños de  $J/h_x$  hasta que comienza a disminuir cuando el valor del valor absoluto de  $J/h_x$  es elevado. La tendencia del error relativo que siguen los tres métodos coincide con la presentada por redes neuronales clásicas, como la máquina de Boltzmann restringida (RBM) [13]. La explicación de este comportamiento no es trivial.

Alternativamente, puede servir de ayuda para comprender el comportamiento del error relativo ver el problema desde un punto de vista meramente matemático. Esencialmente, el problema en cuestión que estamos resolviendo es diagonalizar una matriz. Cuando  $J/h_x$  es cercano a cero, los términos no diagonales asociados a  $\sigma_i^x$  son mucho mayores que los diagonales correspondientes a la interacción  $\sigma_n^z\sigma_m^z$  tal que la matriz a diagonalizar se puede aproximar a la asociada al modelo de Ising clásico en el eje  $X$ . Sin embargo, cuando  $J/h_x$  es cercano a la unidad, los términos diagonales y no diagonales son comparables entre sí, de forma que la diagonalización del hamiltoniano aumenta en complejidad, explicando así el aumento en error. En último lugar, si el valor absoluto de  $J/h_x$  es elevado, la matriz del hamiltoniano está gobernada por los términos diagonales de manera que los términos no diagonales son despreciables y la diagonalización de la matriz es menos compleja.

Como podemos ver, en los casos donde solo hay transición de fase en el régimen ferromagnético ( $J < 0$ ), el máximo en el error relativo coincide justamente con el valor crítico. Nótese que el orden del error presentado se encuentra entre  $10^{-1}$  y  $10^{-2}$ . No obstante, en los casos restantes, los valores críticos tienen lugar en puntos donde el error no es máximo. Concretamente, el error en estos casos está comprendido entre  $10^{-3}$  y  $10^{-4}$ . Por tanto, concluimos que los tres métodos son capaces de resolver el modelo incluso cuando sufre la transición de fase de segundo orden.

Por su parte, vamos a estudiar por separado el error relativo de cada una de las gráficas reunidas en la Figura 8. En el caso  $\alpha = 0.5$ , el algoritmo clásico presenta el error relativo más elevado, llegando a ser del orden de  $10^{-1}$ , debido a que el método se encuentra en el mínimo local asociado al primer excitado, en vez de describir el estado fundamental. Además, vemos que el comportamiento entre el algoritmo clásico y la red neuronal con dos *layers* es similar para valores de  $J/h_x$  negativos, mientras que el circuito cuántico con una y dos *layers* presentan el mismo comportamiento cuando  $J/h_x$  toma valores positivos.

En la segunda gráfica, correspondiente a  $\alpha = 1.0$ , podemos apreciar que el comportamiento de los tres métodos es similares cuando  $J/h_x > 0$ . En cambio, en la rama  $J/h_x < 0$ , la red neuronal con una *layer* es el método que presenta un menor error relativo, del orden de hasta  $10^{-5}$ , mientras que el algoritmo clásico y la red neuronal con dos *layers* tienen el mismo comportamiento, llegando hasta un error relativo del orden de  $10^{-3}$ .

En la siguiente gráfica, asociada a  $\alpha = 2.5$ , la red neuronal con una capa presenta el error relativo más bajo en ambas ramas, llegando a ser del orden de  $10^{-5}$ . Asimismo, el error relativo del algoritmo clásico es el mismo que el de la red cuántica de dos *layers* para los valores positivos de  $J/h_x$ , que es del orden de  $10^{-4}$ , mientras en la rama de valores negativos de  $J/h_x$  se obtiene que el circuito cuántico con dos capas presenta un menor error que el algoritmo *quantum-inspired*, siendo del orden  $10^{-4}$  y  $10^{-3}$  respectivamente.

En último lugar, en la gráfica correspondiente a  $\alpha = 6.0$ , la red neuronal con una y dos *layers* tiene un comportamiento análogo a lo largo de todos los valores de  $J/h_x$ , tal que presentan un error de  $10^{-4}$  y  $10^{-5}$  en la rama negativa y positiva respectivamente. En cambio, el algoritmo clásico muestra el mismo comportamiento que la red neuronal entre las líneas verdes, pero en el extremo de las rama positiva y negativa tiene un error que es un orden de magnitud mayor que el circuito cuántico.

	$L=1 (\cdot 10^3 \text{ s})$	$L=2 (\cdot 10^3 \text{ s})$	<i>Ansatz Clásico</i> ( $\cdot 10^3 \text{ s}$ )
COBYLA	$7,7 \pm 1,6$	$15 \pm 4$	$37 \pm 8$
SLSQP	$2,9 \pm 0,4$	$42 \pm 4$	$18,9 \pm 2,3$

Tabla 2: Tiempo de ejecución, promediado a lo largo de todas las simulaciones, de la red neuronal cuántica con una y dos *layers*, y del algoritmo *quantum-inspired* usando los optimizadores clásicos COBYLA y SLSQP.

Por otro lado, hemos promediado el tiempo de ejecución, para cada uno de los métodos, de las cuatro optimizaciones llevadas a cabo en la Figura 8. Ver Tabla 2. Tal y como era de esperar en el caso de COBYLA, el algoritmo clásico es el más lento mientras que la red neuronal con una *layer* es el más rápido. Concretamente, el circuito cuántico con una capa es, aproximadamente, siete veces más rápido que el algoritmo *quantum-inspired*, mientras que la red neuronal con dos *layers* es el doble de lenta que con una sola capa. Análogamente a la Sección 7.1, insistimos en que el tiempo de ejecución del método clásico puede verse drásticamente reducido con una mejor implementación. Ver Sección 8.

Una vez finalizado el estudio de los tres métodos con el optimizador COBYLA, procedemos a repetir las mismas optimizaciones usando el optimizador SLSQP. Ver Figura 9.

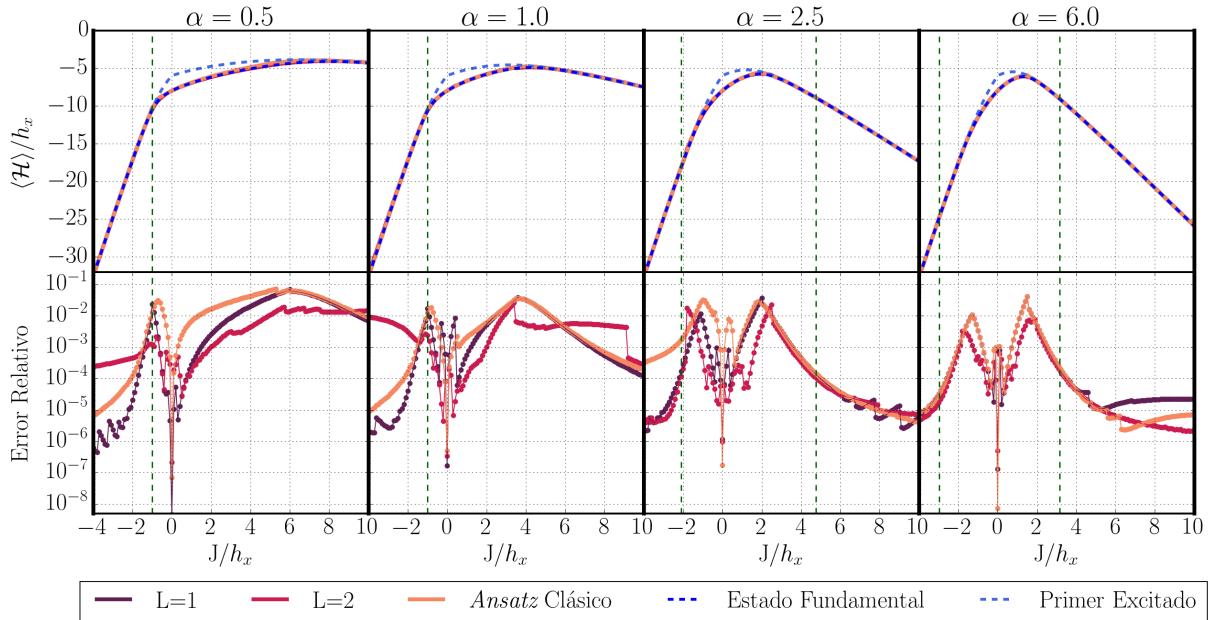


Figura 9: Resolución del estado fundamental del modelo (15), realizado por la red neuronal y algoritmo *quantum-inspired*, para distintos valores de  $\alpha$  usando el optimizador SLSQP.

Como podemos observar, el error relativo sigue la misma tendencia que en el caso con COBYLA. No obstante, la mayor diferencia entre ambos optimizadores es que, al trabajar con SLSQP, los tres métodos son capaces de encontrar el estado fundamental del punto  $J/h_x = 0$  con mayor precisión, que en los casos más favorables presentan un error del orden de  $10^{-8}$ .

Asimismo, para los dos primeros valores de  $\alpha$ , destacamos que el error alrededor del punto crítico oscila entre  $10^{-3}$  y  $10^{-2}$ . En el caso  $\alpha = 2.5$ , el error es de  $10^{-4}$  en el régimen antiferromagnético ( $J > 0$ ), mientras que en el ferromagnético ( $J < 0$ ) oscila entre  $10^{-3}$  y  $10^{-4}$ . Finalmente, cuando  $\alpha = 6.0$ , los tres métodos presentan el mismo error que es del orden de  $10^{-4}$  y  $10^{-5}$  en el régimen antiferromagnético y ferromagnético, respectivamente.

A continuación, vamos a analizar cada una de las gráficas recogidas en la Figura 9. En primer lugar, correspondiente al caso  $\alpha = 0.5$ , vemos que la red neuronal con una *layer* presenta el menor error relativo, que es del orden de  $10^{-5}$ , en la rama de valores negativos de  $J/h_x$ . Mientras que para los valores positivos, es el circuito cuántico con dos capas el método con un menor error, cuyo orden es de  $10^{-2}$ .

En el caso  $\alpha = 1.0$ , la red neuronal con una capa vuelve a ser el método más preciso para los valores negativos de  $J/h_x$ , ya que su error es del orden de  $10^{-6}$ . Además, destaca que el algoritmo clásico presenta un comportamiento similar, con un error del  $10^{-5}$ , pero el circuito cuántico con dos capas presenta un error que es mucho más elevado, siendo del orden de  $10^{-2}$ . Por otro lado, para los valores positivos de  $J/h_x$ , los tres métodos acaban convergiendo hasta tener un error relativo que es del orden de  $10^{-4}$ .

Cuando  $\alpha = 2.5$ , notamos que la red neuronal con una y dos *layers* tienen un comportamiento completamente análogo en el caso  $J/h_x < 0$ , alcanzando así un error entre  $10^{-5}$  y  $10^{-6}$ . Mientras que para  $J/h_x > 0$ , los tres métodos comparten un comportamiento similar, obteniendo así un error de  $10^{-5}$ .

Finalmente, en las gráficas asociadas a  $\alpha = 6.0$ , vemos que los tres métodos siguen la misma tendencia, obteniendo un error hasta del orden de  $10^{-5}$ , en la rama de valores negativos de  $J/h_x$ . Asimismo, para los valores positivos, los tres métodos son similares aunque la red neuronal con dos capas es capaz de converger con un error relativo menor, que se encuentra entre  $10^{-5}$  y  $10^{-6}$ .

Por tanto, debido a que los tres métodos se comportan de forma bastante pareja a lo largo de los cuatro valores de  $\alpha$ , no es inmediato deducir cuál es el mejor de los tres. No obstante, al estudiar los tiempos de ejecución recogidos en la Tabla 2, vemos que la red neuronal con una *layer* es mucho más rápido que el resto de métodos. Concretamente, el tiempo de ejecución es catorce veces menor que el del circuito cuántico con dos capas y hasta seis veces más corto que el del algoritmo clásico. Luego, concluimos que la red neuronal con una capa es el método más apropiado para resolver este problema, pues presenta errores relativos similares al resto de métodos pero su tiempo de ejecución es mucho menor.

Al comparar los tiempos de ejecución entre los optimizadores COBYLA y SLSQP, notamos que se sigue la misma tendencia que en la Sección 7.1: la red neuronal con una *layer* es el método más rápido en ambos casos, mientras que el método más lento es el algoritmo clásico y la red neuronal con dos capas usando COBYLA y SLSQP, respectivamente. El tiempo de ejecución del circuito cuántico con dos *layers* se ve triplicado al emplear SLSQP, de forma que concluimos que el optimizador “Gradient-Based” es más sensible al número de parámetros a optimizar. En cambio, el algoritmo *quantum-inspired* tiene un tiempo de ejecución dos veces mayor al usar COBYLA, y su incertidumbre también aumenta hasta cuatro veces. Esto puede deberse a que dicho optimizador, al ser ‘Gradient-Free’, sea más sensible a la implementación llevada a cabo del algoritmo. Finalmente, cabe destacar que la red neuronal con una capa es entre dos y tres veces más rápida con SLSQP, pues los optimizadores que emplean el gradiente para actualizar los parámetros suelen ser más eficientes computacionalmente. Ver Sección 3.

En definitiva, los tres métodos presentados en el trabajo son capaces de resolver el modelo *long-range* de Ising cuántico (15). Recordamos que esto es debido a que el *ansatz* GCS es capaz de introducir interacciones todos con todos, a través de la matriz de correlación  $\mathcal{V}$ . El método que presenta mejores prestaciones para resolver dicho sistema es la red neuronal cuántica con una *layer* empleando el optimizador SLSQP, ya que presenta el menor error relativo y tiempo de ejecución. Además, destacamos que añadir una capa a la red neuronal no es rentable, ya que no presenta una mejora notable en el error relativo que justifique su aumento considerable en el tiempo de ejecución. En último lugar, el algoritmo *quantum-inspired*, con la implementación actual, tampoco es una mejor alternativa que la red neuronal cuántica.

## 8 Conclusión

Los objetivos de la memoria se han llevado a cabo satisfactoriamente.

En la primera mitad del trabajo, hemos introducido todos los conceptos necesarios para la comprensión del documento. Primero, presentamos el “variational quantum eigensolver” (VQE), entendiendo así las distintas partes que componen el algoritmo. En segundo lugar, se introduce el sistema a simular: modelo *long-range* de Ising cuántico. El modelo destaca por presentar una fenomenología muy rica. Finalmente, presentamos el *ansatz* de estados coherentes generalizados (GCS), que vamos a usar para las redes neuronales cuánticas. La mayor virtud que presentan dichos estados variacionales es que presentan acoplos todos con todos, manteniendo ser eficientes computacionalmente ya que el número de parámetros escala polinómicamente con el tamaño del sistema.

En la segunda mitad, hemos diseñado un algoritmo clásico que simula la red neuronal con una capa usando el *ansatz* GCS. El algoritmo *quantum-inspired* presenta una alternativa a la red neuronal cuántica que puede ser implementada en un dispositivo clásico. Es fundamental destacar que dicho algoritmo es aplicable a cualquier hamiltoniano que pueda ser expresado en matrices de Pauli y presente términos de interacción  $\sigma_n^z \sigma_m^z$ , aunque en este trabajo vamos a particularizar el estudio a sistemas con interacciones de largo alcance.

Posteriormente, se ha discutido el comportamiento de la red neuronal cuántica, con una y dos capas, y del algoritmo *quantum-inspired* resolviendo el modelo *long-range* de Ising cuántico. Los tres métodos han sido capaces de resolver el sistema satisfactoriamente, siendo la red neuronal cuántica con una *layer* usando el optimizador clásico SLSQP la mejor alternativa pues presenta los errores relativos y tiempo de convergencia más reducidos. Cabe destacar que los tres métodos describen el estado fundamental del sistema tanto en el régimen ferromagnético como en el antiferromagnético, tanto con interacciones de largo y corto alcance, e incluso cuando el sistema sufre una transición de fase cuántica de segundo orden.

A estas alturas, es fundamental destacar que la discusión realizada en la Sección 7 se ha llevado a cabo para un número de qubits fijo,  $N = 8$ . El número de qubits es bastante escueto, de forma que lo presentado en el trabajo es un punto de partida (“benchmark”) para futuros trabajos. En concreto, la mejor opción para aumentar sustancialmente el número de qubits es realizar una implementación<sup>21</sup> más eficiente del algoritmo *quantum-inspired*.

Con la ayuda del Dr. Jesús Carrete Montaña, investigador en el Instituto de Nanociencia y Materiales de Aragón (INMA), se ha comenzado a realizar una implementación del algoritmo *quantum-inspired* basada en Google JAX [1].

JAX es una librería que está optimizada para aprovechar las Unidades de Procesamiento Gráfico (GPU) y las Unidades de Procesamiento Tensorial (TPU), permitiendo realizar tareas computacionales de manera más eficiente. Por tanto, JAX presenta una ventaja significativa frente a la biblioteca *SciPy*, que está diseñada para operar en la Unidad Central de Procesamiento (CPU).

---

<sup>21</sup>Recordamos que, en este trabajo, la energía calculada con el algoritmo clásico es minimizada con la función *minimize* de la librería *SciPy*.

## 9 Bibliografía

- [1] James Bradbury, Roy Frostig, Peter Hawkins, et al., *JAX: composable transformations of Python+NumPy programs*, 2018.
- [2] M. Cerezo, Andrew Arrasmith, Ryan Babbush, et al., *Variational quantum algorithms*, Nature Reviews Physics **3** (2021), no. 9, 625–644.
- [3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, *A quantum approximate optimization algorithm*, 2014.
- [4] Roberto Giuntini, Federico Holik, Daniel K. Park, et al., *Quantum-inspired algorithm for direct multi-class classification*, Applied Soft Computing **134** (2023), 109956.
- [5] IBM, *IBM Quantum Learning: Variational Algorithm Design*, <https://learning.quantum.ibm.com/course/variational-algorithm-design/variational-algorithms>, Accedido por última vez a las 19:21 del 09/06/2024.
- [6] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, et al., *Quantum computing with Qiskit*, 2024.
- [7] D. Kraft, *A software package for sequential quadratic programming*, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht, Wiss. Berichtswesen d. DFVLR, 1988.
- [8] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, et al., *A variational eigenvalue solver on a photonic quantum processor*, Nature Communications **5** (2014), no. 1.
- [9] M. J. D. Powell, *Direct search algorithms for optimization calculations*, Acta Numerica **7** (1998), 287–336.
- [10] John Preskill, *Quantum computing and the entanglement frontier*, 2012.
- [11] John Preskill, *Quantum Computing in the NISQ era and beyond*, Quantum **2** (2018), 79.
- [12] Qiskit Community, *Qiskit Algorithms Documentation: NumPyEigensolver*, [https://qiskit-community.github.io/qiskit-algorithms/stubs/qiskit\\_algorithms.NumPyEigensolver.html](https://qiskit-community.github.io/qiskit-algorithms/stubs/qiskit_algorithms.NumPyEigensolver.html), 2024, Accessed on July 6, 2024.
- [13] Sebastián Roca-Jerat, Manuel Gallego, Fernando Luis, et al., *Transformer wave function for quantum long-range models*, 2024.
- [14] Juan Román-Roche, Víctor Herráiz-López, and David Zueco, *Exact solution for quantum strong long-range models via a generalized hubbard-stratonovich transformation*, Physical Review B **108** (2023), no. 16.
- [15] Paul M. Schindler, Tommaso Guaita, Tao Shi, et al., *Variational ansatz for the ground state of the quantum sherrington-kirkpatrick model*, Physical Review Letters **129** (2022), no. 22.
- [16] Francesco Tacchino, Alessandro Chiesa, Stefano Carretta, et al., *Quantum computers as universal quantum simulators: State-of-the-art and perspectives*, Advanced Quantum Technologies **3** (2019), no. 3.
- [17] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al., *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods **17** (2020), 261–272.

## A Cálculo de $\lambda$

Vamos a demostrar que una rotación local sobre una matriz de Pauli, en un qubit arbitrario  $k$ , resulta en una combinación lineal de matrices de Pauli:

$$\mathcal{U}^\dagger \sigma_k^\alpha \mathcal{U} = \sum_{\beta=x,y,z} \lambda_k^{\alpha\beta} \sigma_k^\beta \quad (51)$$

donde  $\alpha \in \{x, y, z\}$ .

Gracias a que consideramos un qubit arbitrario  $k$ , podemos prescindir del subíndice  $k$  para aligerar la notación. Asimismo, usando que  $(\hat{n}\vec{\sigma})^2 = I$ , podemos descomponer la rotación local en un seno y coseno.

$$\mathcal{U} = \exp\left(-\frac{i}{2}\tau(\hat{n} \cdot \vec{\sigma})\right) = -i(\hat{n}\vec{\sigma}) \sin\left(\frac{\tau}{2}\right) + \cos\left(\frac{\tau}{2}\right) I \quad (52)$$

A continuación, con el fin de realizar los cálculos explícitamente, vamos a calcular la acción de la rotación local sobre  $\sigma^z$ .

$$\mathcal{U}^\dagger \sigma^z \mathcal{U} = \left(i(\hat{n}\vec{\sigma}) \sin\left(\frac{\tau}{2}\right) + \cos\left(\frac{\tau}{2}\right) I\right) \sigma^z \left(-i(\hat{n}\vec{\sigma}) \sin\left(\frac{\tau}{2}\right) + \cos\left(\frac{\tau}{2}\right) I\right) \quad (53)$$

Teniendo en cuenta que las matrices de Pauli son involutivas y que el producto de dos matrices de Pauli es otra matriz de Pauli, es inmediato obtener que:

$$\begin{aligned} \mathcal{U}^\dagger \sigma^z \mathcal{U} &= +\sigma^x \underbrace{\left[ \sin^2\left(\frac{\tau}{2}\right) (2n_x n_z) + \sin\left(\frac{\tau}{2}\right) \cos\left(\frac{\tau}{2}\right) (-2n_y) \right]}_{\lambda^{zx}} \\ &\quad + \sigma^y \underbrace{\left[ \sin^2\left(\frac{\tau}{2}\right) (2n_y n_z) + \sin\left(\frac{\tau}{2}\right) \cos\left(\frac{\tau}{2}\right) (2n_x) \right]}_{\lambda^{zy}} \\ &\quad + \sigma^z \underbrace{\left[ \sin^2\left(\frac{\tau}{2}\right) (-n_x^2 - n_y^2 + n_z^2) + \cos^2\left(\frac{\tau}{2}\right) \right]}_{\lambda^{zz}} \end{aligned} \quad (54)$$

donde destacamos que los términos asociados a la identidad se anulan.

Finalmente, introduciendo que  $\hat{n} = (\sin(\theta) \cos(\varphi), \sin(\theta) \sin(\varphi), \cos(\theta))$  en la Ecuación (54), se puede obtener explícitamente los coeficientes  $\lambda$ .

$$\begin{cases} \lambda^{zx} = 2 \sin\left(\frac{\tau}{2}\right) \sin(\theta) (\sin\left(\frac{\tau}{2}\right) \cos(\theta) \cos(\varphi) - \cos\left(\frac{\tau}{2}\right) \sin(\varphi)) \\ \lambda^{zy} = 2 \sin\left(\frac{\tau}{2}\right) \sin(\theta) (\sin\left(\frac{\tau}{2}\right) \cos(\theta) \sin(\varphi) + \cos\left(\frac{\tau}{2}\right) \cos(\varphi)) \\ \lambda^{zz} = \sin^2\left(\frac{\tau}{2}\right) (2 \cos^2(\theta) - 1) + \cos^2\left(\frac{\tau}{2}\right) \end{cases} \quad (55)$$

Análogamente, podemos calcular los coeficientes  $\lambda$  asociados a la matriz de Pauli  $X$ .

$$\begin{cases} \lambda^{xx} = \sin^2\left(\frac{\tau}{2}\right) (2 \sin^2(\theta) \cos^2(\varphi) - 1) + \cos^2\left(\frac{\tau}{2}\right) \\ \lambda^{xy} = 2 \sin\left(\frac{\tau}{2}\right) (\sin\left(\frac{\tau}{2}\right) \sin^2(\theta) \sin(\varphi) \cos(\varphi) - \cos\left(\frac{\tau}{2}\right) \cos(\theta)) \\ \lambda^{xz} = 2 \sin\left(\frac{\tau}{2}\right) \sin(\theta) (\sin\left(\frac{\tau}{2}\right) \cos(\theta) \cos(\varphi) + \cos\left(\frac{\tau}{2}\right) \sin(\varphi)) \end{cases} \quad (56)$$

En último lugar, repetimos el cálculo con la matriz de Pauli  $Y$ , obteniendo así los coeficientes  $\lambda$  correspondientes.

$$\begin{cases} \lambda^{yx} = 2 \sin\left(\frac{\tau}{2}\right) (\sin\left(\frac{\tau}{2}\right) \sin^2(\theta) \sin(\varphi) \cos(\varphi) + \cos\left(\frac{\tau}{2}\right) \cos(\theta)) \\ \lambda^{yy} = \sin^2\left(\frac{\tau}{2}\right) (2 \sin^2(\theta) \sin^2(\varphi) - 1) + \cos^2\left(\frac{\tau}{2}\right) \\ \lambda^{yz} = 2 \sin\left(\frac{\tau}{2}\right) \sin(\theta) (\sin\left(\frac{\tau}{2}\right) \cos(\theta) \sin(\varphi) - \cos\left(\frac{\tau}{2}\right) \cos(\varphi)) \end{cases} \quad (57)$$

## B Código

El código desarrollado para llevar a cabo este Trabajo Fin de Grado está recopilado [aquí](#).

## C Especificaciones del dispositivo

Las características del ordenador en el que se han llevado a cabo todas las simulaciones son:

- **Modelo:** Asus VivoBook S14 S433 (11th Gen).
- **Procesador:** 11th Gen Intel(R) Core(TM) i7-1165G7 a 2.80GHz.
- **Memoria RAM:** 16,0 GB (15,7 GB usable).
- **Tipo de sistema:** Sistema operativo de 64 bits, procesador basado en x64.