



Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Física

**Cálculo del estado fundamental con algoritmos cuánticos
variacionales**

Autor: Héctor Melchor Alaiz

Tutores: Juan Carlos García Escartín y Luis Miguel Nieto Calzada

Año: 2022-2023

Dedicado a mis padres y a María.

Resumen

This work deals with the applications of the *variational quantum eigensolver* method for quantum computers to solving the Ising model. The algorithm allows to use the techniques from quantum computing to find out the energy of the ground state of certain Hamiltonians. The Ising model is studied for a one-dimensional spin chain with nearest neighbour interaction under the presence of a trasverse magnetic field. The solution is found for a system with 4 particles analyzing phenomena like the phase transition that occurs when the relative magnetic field amplitude is changed and the effect of noise in the quantum computer on the precision of the results.

Este trabajo trata las aplicaciones del algoritmo cuántico variacional llamado *variational quantum eigensolver* al modelo de Ising. El algoritmo permite usar las técnicas de la computación cuántica para el cálculo de la energía del estado fundamental de ciertos Hamiltonianos. Se estudiará un modelo de Ising unidimensional con varios espines con interacción entre vecinos inmediatos bajo la presencia de un campo magnético transversal. Se presenta la solución para un sistema con 4 espines en la que se pueden apreciar fenómenos como la transición de fase que ocurre al variar el efecto relativo del campo magnético así como la influencia del ruido en el ordenador cuántico sobre los resultados.

Índice general

1. Introducción	4
2. Fundamentos de la computación cuántica	5
2.1. Postulados de la mecánica cuántica	5
2.2. Teorema de descomposición espectral	6
2.2.1. Superposición, entrelazamiento e interferencia	7
2.3. <i>Qubits</i>	8
2.3.1. Modelo matemático	8
2.3.2. Base binaria y representación	9
2.4. Puertas cuánticas	10
2.4.1. Puerta X	11
2.4.2. Puerta Y	11
2.4.3. Puerta Z	11
2.4.4. Puerta H o puerta Hadamard	12
2.4.5. Puerta CNOT	12
2.5. Esfera de Bloch	13
3. Algoritmos cuánticos	15
3.1. Introducción	15
3.2. <i>Variational quantum eigensolver</i> (VQE)	16
4. El modelo de Ising	18
5. Elementos del VQE	20
5.1. Construcción del Hamiltoniano:	20
5.2. Transformación del Hamiltoniano:	20
5.3. Proceso de medida:	20
5.4. Preparación del <i>ansatz</i> :	21
5.4.1. Optimización de los parámetros:	21
5.4.2. Mitigación del error:	22
6. Funcionamiento del algoritmo	23
6.1. Pre-procesamiento	23
6.1.1. Representación del Hamiltoniano	23
6.1.2. Agrupamiento y estrategia de medición	24
6.2. El cuerpo del algoritmo	24
6.2.1. Preparación del <i>ansatz</i>	24
6.2.2. Rotación de la base	24
6.2.3. Medida final	25
6.2.4. Actualización de parámetros	25
6.3. Post-procesamiento	25

6.3.1. Mitigación del error	25
7. Experimentos	26
7.1. Cálculo de valores esperados	26
7.1.1. 1 qubit	26
7.1.2. 2 qubits	27
7.2. Algoritmo VQE para la resolución de un modelo de Ising programado en Python	29
8. Resultados. Análisis de un sistema de 4 espines.	34
8.1. $h=0$	34
8.1.1. $J=-1$	35
8.1.2. $J=1$	35
8.2. $h \gg J$	36
8.3. Magnetización y transición de fase	36
8.3.1. Simulador sin ruido	36
9. Resumen y conclusiones	39

Capítulo 1

Introducción

En este trabajo se explicará cómo utilizar un VQE (*Variational Quantum Eigensolver*) para la solución del modelo de Ising, que permite estudiar los campos de fase con un sistema sencillo pero no trivial.

Se empezará dando una pequeña introducción a la mecánica cuántica junto al teorema de descomposición espectral y la superposición cuántica, el entrelazamiento cuántico y la interferencia en el capítulo 2. Entre otros temas, se darán los fundamentos para entender las distintas puertas cuánticas y la visualización de sus efectos usando la esfera de Bloch (sección 2.5). Además, en el capítulo 3 se comentan los algoritmos que pueden realizarse combinando las puertas cuánticas descritas en el capítulo 2.

En el capítulo 4 se presenta el modelo de Ising que vamos a tratar con el algoritmo cuántico. En este capítulo se describe el sistema unidimensional de espines acoplados bajo estudio y su Hamiltoniano. El objetivo final será la búsqueda de los estados estables de mínima energía que se corresponden al autovector más pequeño del Hamiltoniano.

El capítulo 5 repasa los elementos básicos necesarios para poder usar el algoritmos VQE, incluyendo cómo se construye y transforma el Hamiltoniano, cómo se prepara el *ansatz* y cómo optimizaremos los parámetros del mismo.

El capítulo 6 muestra como se integran esos ingredientes para resolver el problema bajo estudio.

Por último, se presenta el algoritmo elaborado para estudiar la energía mínima y la magnetización del estado fundamental en el modelo de Ising. Primero se comentarán los métodos generales y el código utilizado en el capítulo 7. En el capítulo 8 se usa el código desarrollado para estudiar un sistema de 4 espines analizando el estado fundamental para sistemas ferromagnéticos y paramagnéticos así como la aparición de una transición de fase al ir variando un campo magnético externo perpendicular a la orientación en la que se miden en los espines. Además, se muestra el efecto del ruido en un ordenador cuántico en la precisión de los resultados.

El capítulo 9 resume los resultados de estas simulaciones y comenta las conclusiones que se pueden extraer.

Capítulo 2

Fundamentos de la computación cuántica

Antes de entrar a valorar la idea de los algoritmos cuánticos y más en concreto de nuestro VQE vamos a recordar las ideas clave de la mecánica cuántica y en particular de cómo se aplica esta a la computación.

2.1. Postulados de la mecánica cuántica

La mecánica cuántica se sustenta sobre los siguientes postulados [3]:

1. Todo estado físico en un tiempo t_0 se describe por medio de un ket $|\psi(t_0)\rangle$ perteneciente al espacio de estados ϵ .

2. Las magnitudes físicas se representarán a través de operadores lineales que serán observables, esto es, hermíticos con autovalores reales y autovectores que formen base en el espacio de estados.

3. El único resultado posible de cualquier medición de un observable es uno de los autovalores a_i del observable correspondiente A .

4. **Principio de descomposición espectral:** La probabilidad de obtener el autovalor a_i en una medida será:

$$P(a_i) = |\langle u_i | \psi \rangle|^2. \quad (2.1)$$

Donde $|u_i\rangle$ es el autovector del observable A con autovalor a_i .

Si el autovalor está degenerado:

$$P(a_i) = \sum_{n=1}^{g_i} |\langle u_i^n | \psi \rangle|^2, \quad (2.2)$$

donde g_i es el grado de degeneración del autovector.

5. **Reducción del paquete de ondas:** No se realiza una medida sin perturbar el estado correspondiente.

La medición de un autovalor hará que se proyecte al estado correspondiente:

$$|\psi\rangle \rightarrow |\psi'\rangle = \frac{P_i |\psi\rangle}{\sqrt{\langle\psi|P_i|\psi\rangle}}. \quad (2.3)$$

Donde P_i es el proyector del autovector correspondiente al autovalor concreto:

$$P_i = \sum_{n=1}^{g_i} |u_i^n\rangle\langle u_i^n|. \quad (2.4)$$

6. La evolución de un sistema vendrá dada por la ecuación de Schrödinger:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle. \quad (2.5)$$

Con estos postulados podemos entrar ya a descubrir la relación entre la computación y la mecánica cuántica.

2.2. Teorema de descomposición espectral

Este teorema relacionado con el postulado número 1 nos ofrece información que será útil más adelante.

El álgebra lineal nos ofrece uno de sus resultados más interesantes y es el que dice que toda matriz autoadjunta $A \in M_{n \times n}(\mathbb{C})$ puede diagonalizarse a través de una matriz unitaria y posee solo valores propios reales.

Este teorema aplicado a la mecánica cuántica nos dice que cualquier operador normal sobre un espacio de Hilbert es diagonal respecto a alguna base ortonormal del mismo.

Con esto podemos ver que todo operador normal se podrá siempre descomponer en términos de sus autovectores:

$$\begin{aligned} O &= \sum_i \lambda_i |i\rangle\langle i| = \sum_i \lambda_i P_i, \\ \sum_i P_i &= I, \\ P_i P_j &= \delta_{ij} P_i; \end{aligned}$$

siendo $\{|i\rangle\}$ la base de autovectores con los autovalores $\{\lambda_i\}$ y P_i el proyector sobre el subespacio de autovalor λ_i .

2.2.1. Superposición, entrelazamiento e interferencia

Vamos a explicar tres aspectos fundamentales que nos servirán para trabajar con los *qubits*, que se definirán más adelante.

- **Superposición de estados:** La superposición se produce cuando encontramos un estado que se representa como combinación lineal de estados.

Por ejemplo, en computación cuántica trabajaremos mucho con los siguientes estados:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle; \quad (2.6)$$

donde encontramos un estado superposición del estado $|0\rangle$ y del estado $|1\rangle$.

- **Entrelazamiento cuántico:** Veamos primero un ejemplo con el siguiente estado.

$$|\psi\rangle = \alpha |00\rangle + \beta |10\rangle + \gamma |01\rangle + \delta |11\rangle. \quad (2.7)$$

Este es un estado no correlacionado, también llamado estado entrelazado, conociendo el primer valor del *ket*, siendo un *ket* la representación de los estados, no conocemos el valor del segundo, podría ser cualquiera de los dos restantes. Es decir, si cogemos el primer *ket* y obtenemos que su primer valor es un 0 no podemos asegurar el segundo valor pues podríamos estar cogiendo un estado que fuese $|00\rangle$ o uno que fuese $|01\rangle$.

Para poder dar el paso a comprender el entrelazamiento cuántico debemos ver donde se encuentran los estados con los que trabajamos.

Podemos definir el espacio de Hilbert de un qubit A como ϵ_A y el espacio de Hilbert de un qubit B como ϵ_B . Con esto podemos definir el producto tensorial de estos dos espacios como:

$$\epsilon = \epsilon_A \otimes \epsilon_B. \quad (2.8)$$

Con esto podemos definir un *ket* en el espacio total como producto tensorial de los *kets* de cada espacio:

$$|u_i\rangle_A \otimes |v_j\rangle_B \equiv |u_i, v_j\rangle \in \epsilon. \quad (2.9)$$

El entrelazamiento se produce entre estados correlacionados, esto es, conociendo la medida de un estado podemos conocer la medida de más.

$$|\psi\rangle = \alpha |01\rangle + \beta |10\rangle. \quad (2.10)$$

Este es un estado correlacionado, conociendo el valor del primer *qubit* conocemos al instante el valor del segundo.

En este caso no podemos expresar el estado fundamental a partir del estado individual de cada *qubit*.

- **Interferencia:** En física, interferencia designa lo que ocurre cuando dos ondas se encuentran. Los fenómenos de interferencia ocurren también con la luz. Al trabajar con el *hardware* cuántico pueden producirse fenómenos de interferencia debido a la dualidad entre la onda y la partícula [3].

La interferencia tiene mucha relevancia en la computación cuántica pues nos permite optimizar nuestros algoritmos debido a la probabilidad de obtener valores no deseados.

2.3. Qubits

Los *qubits* son los análogos a los bits en computación clásica. Recordemos que en clásica se trabaja con bits, estos bits son: 0 y 1. Bits que sirven para crear todo el entramado computacional que hoy en día conocemos.

Los *qubits* se definen como *quantum bits*, es decir, bits cuánticos, esto significa que se van a encontrar en dos estados cuánticos.

Así, si un bit puede ser 0 o 1, un *qubit* se encontrará en:

- Estado puro: $|0\rangle$.
- Estado puro: $|1\rangle$.
- Estado puro en superposición: $\alpha |0\rangle + \beta |1\rangle$.

Es este último estado en superposición la gran diferencia con la computación clásica y lo que le proporcionará una gran ventaja frente a los algoritmos clásicos.

¿Y por qué este estado en superposición es una ventaja?

Suponiendo que trabajamos en la base $\{|0\rangle, |1\rangle\}$.

Nuestro estado en superposición se mantendrá en este estado hasta que lo observemos, momento en el que se proyectará hacia uno de los dos estados de la base.

La ventaja de esta posibilidad es el hecho de poder trabajar con estados superpuestos de tal manera que ahorremos pasos al no trabajar con cada uno de los estados individuales que lo componen, trabajamos con la posibilidad de su proyección.

Sería similar a, cuando por ejemplo, haces una mudanza. En vez de cargar cada uno de tus objetos individualmente los introduces en una caja y los transportas de manera simultánea.

¿Cuántos estados posibles tendremos en una base en relación a los *qubits* asignados?

Los estados posibles serán 2^n siendo n el número de *qubits*. Así con 1 *qubit* tendremos 2 estados puros posibles en la base, $\{|0\rangle, |1\rangle\}$ y con 2 *qubits* tendremos 4 estados puros posibles en la base, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

2.3.1. Modelo matemático

Vamos a valorar matemáticamente lo comentado anteriormente.

Nuestros estados $|0\rangle$ y $|1\rangle$ pueden ser representados por los vectores $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ y $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ en \mathbb{C}^2 , que forman una base ortogonal.

Así, un estado genérico es una combinación lineal $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ de los vectores de la base, donde α y β pueden ser números complejos.

Su módulo al cuadrado nos dará la probabilidad de obtener el estado al que acompañan.

Por ejemplo, si en $|\psi\rangle$ medimos la probabilidad de que nuestro estado sea $|0\rangle$ obtendremos $|\alpha|^2$ y si medimos la probabilidad de que nuestro estado sea $|1\rangle$ obtendremos $|\beta|^2$, siendo $|\alpha|^2 + |\beta|^2 = 1$, a estos coeficientes se les llama amplitudes de probabilidad.

Si ahora hablamos de varios *qubits*, (n) , tenemos:

$$|\underbrace{0\dots 0}_n\rangle, |\underbrace{0\dots 1}_n\rangle, \dots, |\underbrace{1\dots 1}_n\rangle, \quad (2.11)$$

donde cada uno de estos estados tiene n *qubits*, así el número de estados será 2^n , con n *qubits* cada uno.

Cada uno de estos estados se podrá expresar en una base de \mathbb{C}^n como un vector.

2.3.2. Base binaria y representación

Comentemos antes cómo funciona la base binaria y cómo se representan en la misma los números naturales.

La base binaria permite representar números en función de los bits que se tengan, así, solo podrás representar, con n bits, hasta el número $2^n - 1$, pudiendo representar en total 2^n números.

Por ejemplo:

- Con $n=1$ se podrá representar hasta $2^1 - 1 = 1$, pudiendo representar $2^1 = 2$ números: el 0 y el 1.
- Con $n=2$ se podrá representar hasta $2^2 - 1 = 3$, pudiendo representar $2^2 = 4$ números: el 0, el 1, el 2 y el 3.

Dicho esto podemos preguntarnos cómo representar los *qubits* 0 y 1.

Como comentamos antes, su representación es la siguiente:

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

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

Vemos que estamos colocando un 1 en la primera posición en el caso del primer estado y un 1 en la segunda posición en caso del segundo estado. ¿Podemos extender esta situación a más *qubits*?

Valoremos el estado $|101\rangle$ que se encuentra en la siguiente base:

$$\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}. \quad (2.12)$$

Este estado, pasándolo de base binaria a base decimal sería: $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$.

Podemos decir que el estado es: $|101\rangle = |5\rangle$.

Representando este estado en la base de \mathbb{C}^8 , sabiendo que tenemos $2^3 = 8$ distintos estados como hemos visto antes, obtenemos:

$$|5\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \quad (2.13)$$

Encontrando el 1 en la sexta posición, siendo el 5 nuestro número en la base decimal. Comprobamos así como va encajando todo.

Por tanto, cada estado corresponde a un vector en la base \mathbb{C}^{2^n} dado por:

$$|a_{2,n}\rangle = \left. \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \right\} 2^n, \quad (2.14)$$

siendo $a_{2,n}$ la representación del número decimal en binario con n el número de *qubits*.

Con esta idea podemos ver cuál es la relación matemática entre *qubits*.

Tomando cómo ejemplo el estado de dos *qubits* $|00\rangle$ podemos ver que sigue las mismas ideas que acabamos de ver:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (2.15)$$

que corresponde al vector $|0\rangle$ en la base \mathbb{C}^4 .

Otro ejemplo sería:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad (2.16)$$

que corresponde al vector $|4\rangle$ en la base \mathbb{C}^4 .

2.4. Puertas cuánticas

Al igual que en la computación clásica, en computación cuántica también existen puertas lógicas que nos permitirán la creación de distintos algoritmos.

Estas puertas tienen una serie de características que difieren de las puertas tradicionalmente clásicas.

Estas características son:

- Se exige que actúen linealmente en los estados en superposición.
- Son puertas unitarias, es decir, reversibles, puedes obtener el estado inicial teniendo el final.

Veamos las puertas más conocidas.

2.4.1. Puerta X

Es el análogo a la puerta NOT clásica y viene dada por la matriz:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (2.17)$$

Una puerta que transforma los estados como sigue:

$$\begin{aligned} |0\rangle &\rightarrow |1\rangle, \\ |1\rangle &\rightarrow |0\rangle. \end{aligned}$$

Es una puerta lineal:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \alpha |1\rangle + \beta |0\rangle. \quad (2.18)$$

Es también una puerta unitaria:

$$XX^\dagger = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = I. \quad (2.19)$$

Como veremos a continuación, esta puerta corresponde a una rotación alrededor del eje x en la esfera de Bloch. Véase sección 2.5 en la sección actual.

2.4.2. Puerta Y

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (2.20)$$

Esta puerta transforma los estados como sigue:

$$\begin{aligned} |0\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = i \begin{pmatrix} 0 \\ 1 \end{pmatrix} = i |1\rangle, \\ |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -i \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -i |0\rangle. \end{aligned}$$

Se puede comprobar que cumple también el ser unitaria y lineal tal y como lo cumplirán el resto de puertas cuánticas que veremos.

Equivale a una rotación alrededor del eje y en la esfera de Bloch.

2.4.3. Puerta Z

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.21)$$

Esta puerta transforma los estados como sigue:

$$\begin{aligned} |0\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle, \\ |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1\rangle. \end{aligned}$$

Esta puerta equivale a una rotación alrededor del eje z en la esfera de Bloch.

Con esta puerta se aplica un desfase de π al estado $|1\rangle$, es decir, si tenemos un estado inicial:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \quad (2.22)$$

se le aplica un desfase extra $e^{i\pi} = -1$ al estado $|1\rangle$.

2.4.4. Puerta H o puerta Hadamard

Esta puerta sirve para generar una superposición de estados equiprobables. Esto es una gran herramienta y será útil en multitud de algoritmos cuánticos.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (2.23)$$

Veamos cómo se transforma un *qubit*:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle), \quad (2.24)$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle). \quad (2.25)$$

Si ahora ponemos 2 *qubits* y aplicamos una puerta Hadamard a cada uno se nos genera una superposición igualmente probable de los 4 estados posibles. Por ejemplo:

$$(H \otimes H)(|0\rangle \otimes |0\rangle) = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle). \quad (2.26)$$

2.4.5. Puerta CNOT

Esta puerta es muy importante para algoritmos como el de Deutsch-Jozsa [6] y viene dada por la matriz:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}. \quad (2.27)$$

Esta puerta actúa sobre 2 *qubits* de la siguiente manera:

$$|xy\rangle \rightarrow |xx \oplus y\rangle. \quad (2.28)$$

Donde \oplus es la operación **XOR**. La puerta **XOR** es una puerta lógica clásica que implementa el *o* exclusivo, es decir, una salida verdadera resulta si una, y solo una de las entradas a la puerta es verdadera. Si ambas entradas son falsas o ambas son verdaderas, resulta en una salida falsa.

La tabla de verdad de la puerta CNOT es:

x	y	$x \oplus y$
1	1	0
1	0	1
0	1	1
0	0	0

El primer *qubit* x se denomina de control y es invariante en la transformación, el segundo *qubit* se denomina objetivo y se niega siempre que el *qubit* de control sea 1.

2.5. Esfera de Bloch

La esfera de Bloch es una representación geométrica del espacio de estados de un sistema de dos niveles [12] [13].

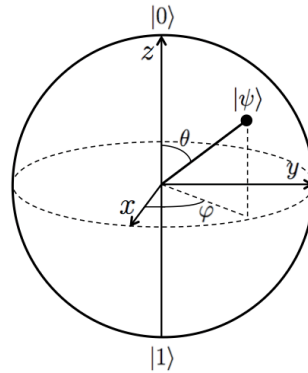


Figura 2.1: Representación de la esfera de Bloch.

Se puede demostrar que cualquier estado dentro de un sistema de dos niveles se puede representar mediante la siguiente expresión:

$$|\psi\rangle = \cos\frac{\theta}{2} |0\rangle + e^{i\varphi} \sin\frac{\theta}{2} |1\rangle, \quad (2.29)$$

donde $0 \leq \theta \leq \pi$ y $0 \leq \varphi \leq 2\pi$.

Podemos ver qué ocurre aplicando las distintas puertas cuánticas vistas en la subsección anterior:

- Puerta X

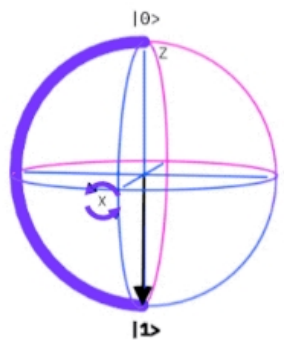


Figura 2.2: Puerta X en la esfera de Bloch.

- Puerta Y

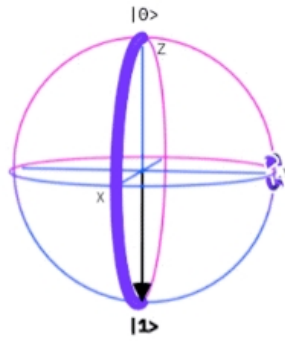


Figura 2.3: Puerta Y en la esfera de Bloch.

- Puerta Z

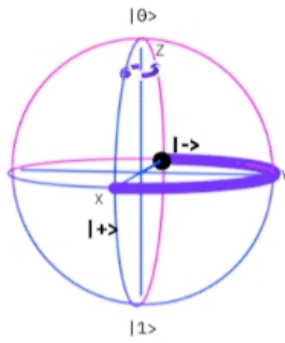


Figura 2.4: Puerta Z en la esfera de Bloch.

- Puerta Hadamard

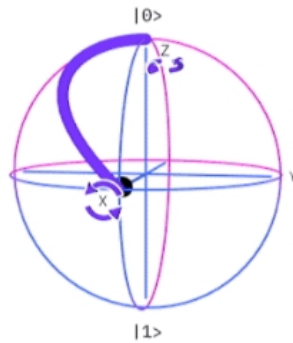


Figura 2.5: Puerta Hadamard en la esfera de Bloch.

Capítulo 3

Algoritmos cuánticos

3.1. Introducción

Después de entender las herramientas de la computación cuántica es el momento de descubrir por qué esta trajo una revolución en algunos aspectos computacionales.

Por ejemplo, el algoritmo de Shor permitió la factorización de los números enteros con una velocidad mucho más alta que los ordenadores tradicionales y el *variational quantum eigensolver* permitió la resolución de problemas de química cuántica o problemas con el modelo de Ising [5] [8] [10].

Es importante recordar que pese a que existen muchos algoritmos cuánticos variacionales es, el que vamos a comentar a continuación, el que nos trae los verdaderos avances en el cálculo de energías base atómicas o moleculares o en el modelo de Ising que trabaja con parejas de espines.

En el caso del modelo de Ising que será el que resolvamos en el capítulo 7 trataremos con estados como el $|0\rangle \equiv |\uparrow\rangle$, que significará que el espín a tratar estará en la dirección positiva del eje Z mientras o el $|1\rangle \equiv |\downarrow\rangle$, que significará que el espín a tratar estará en la dirección negativa del eje Z.

A pesar de que la computación cuántica ha sufrido un enorme avance en los últimos años la realidad sigue siendo que la supremacía cuántica es todavía un reto al alcance de muy pocos. Esta supremacía consiste en la creación de ordenadores cuánticos que permitan resolver algunos problemas a una velocidad superior respecto a sus análogos clásicos y en un tiempo suficientemente razonable.

Por el momento el hardware cuántico, los ordenadores existentes, poseen muy pocos *qubits*, lo que constriñe enormemente las posibilidades que la mecánica cuántica nos ofrece.

Es por esto que los ordenadores denominados NISQ, *Noisy-Intermediate Scale Quantum*, con un número limitado de *qubits* son los ordenadores llamados a superar a los convencionales en los próximos años [2].

Estos ordenadores permiten ejecutar algoritmos donde sean necesarios un pequeño número de *qubits*, y a menudo estos algoritmos son algoritmos híbridos con cierto grado de resistencia al ruido. Estos algoritmos híbridos comúnmente se reparten los pasos entre los ordenadores convencionales y los cuánticos.

El número de operaciones debe ser mínimo para reducir los errores en los estados cuánticos.

Debido a estas restricciones los algoritmos a considerar son reducidos.

3.2. *Variational quantum eigensolver*(VQE)

El *variational quantum eigensolver* es uno de los algoritmos más prometedores para implementar en ordenadores NISQ, su objetivo es calcular una cota superior para la energía del estado base de un Hamiltoniano como primera idea antes de calcular las propiedades energéticas de moléculas, átomos o materiales.

Este algoritmo es, por tanto, muy interesante para estudios relacionados con la química cuántica o la física de la materia condensada.

La pregunta esencial aquí es, **¿qué puede proporcionarnos esta nueva rama que no pueda hacer la química computacional?**

La química computacional proporciona métodos eficientes para aproximar propiedades a estudiar, pero su coste es muy alto si queremos obtener una precisión adecuada.

Una de las razones por las que estos métodos pueden perder precisión es tratar de manera incorrecta las relaciones entre los electrones que constituyen el sistema.

Todas estas interacciones entre electrones producen que el coste computacional crezca exponencialmente al tamaño del sistema haciendo intratable el coste computacional.

Ahora bien, sabemos que un conjunto de *qubits* obedece las leyes de la mecánica cuántica del mismo modo que la función de onda electrónica.

Esto hace que gracias al principio de superposición lo que antes tenía un crecimiento exponencial ahora requerirá un número linealmente creciente de *qubits*. Esto permite con una función de onda habitual, modelar, y trabajar en ella reduciendo el número de operaciones computacionales.

Este algoritmo se ha utilizado también en muchos otros problemas de física nuclear, de estructura nuclear, de física de altas energías, de espectroscopia vibracional, de resolución de ecuaciones de Schrödinger no lineales o incluso para el cálculo de estados cuánticos de un agujero negro de Schwarzschild-de Sitter [9]. En nuestro caso, veremos en la siguiente sección la aplicación en el modelo de Ising. Véase el capítulo 4.

Este algoritmo se basa en el hecho de que el valor esperado del Hamiltoniano a estudiar nos va a otorgar unos valores propios:

$$\langle H \rangle_\psi = \langle \psi | \sum_i \lambda_i | \varphi_i \rangle \langle \varphi_i | \psi \rangle. \quad (3.1)$$

Donde cada medida es una cota: $\lambda_0 \leq \lambda_1 = \dots$, obteniendo así cotas superiores. Donde cualquier estado que probemos va a tener energías superiores a la cota mínima que conozcamos.

El *variational quantum eigensolver* comienza con un conjunto de *qubits*, después se le aplica un circuito cuántico para modelarlo y entrelazar la función de onda. El circuito cuántico está definido por una serie de operaciones cuánticas que se aplicarán a los *qubits*, el número de operaciones a aplicar se le denomina profundidad [2].

Dicho circuito viene dado por dos partes:

- Una estructura de puertas cuánticas ordenadas denominada *ansatz*.
- Un conjunto de parámetros que dirigen algunas de esas puertas.

Después se origina un estado de *qubits* que modelan una función de onda de prueba, el Hamiltoniano del sistema a estudiar se medirá con respecto a esta función de onda para estimar la energía.

El VQE por tanto optimizará variacionalmente los parámetros del *ansatz* para minimizar la energía de prueba con la restricción de ser siempre mayor que la energía exacta del estado base del Hamiltoniano usando el principio variacional.

Como veremos más adelante, para que podamos usar el VQE correctamente necesitaremos que el *ansatz* esté bien optimizado para reducir el número de operaciones cuánticas requeridas.

Sin embargo debe entenderse que un *ansatz* más superficial, es decir, con menos operaciones cuánticas, generalmente cubrirá una extensión menor del espacio de todas las posibles funciones de onda y podría resultar en una menor precisión de la energía del estado base.

Antes de ver todo lo que nos interesa del mismo vamos a hablar de algunos obstáculos importantes que han traído este tipo de algoritmos:

- Una de las problemáticas que existen es que el número de mediciones aumenta polinomialmente al tamaño del sistema y de la descripción del estado, y, pese a que esto es mucho más eficiente que un escalado exponencial clásico, en ocasiones son suficientes mediciones como para hacer intratable el método.

Por ello se siguen desarrollando esquemas de medición más eficientes que permitan compactar el Hamiltoniano o medir conjuntamente observables que conmuten.

- Un enfoque alternativo al problema de la medición podría residir en el enorme potencial de paralelización del VQE.
- Otro de los problemas es el denominado como *problema de la meseta estéril*. Este problema se produce en muchos algoritmos de aprendizaje en ordenadores cuánticos que producen auténticos callejones sin salida en problemas de optimización.

Esta meseta estéril es el resultado de que los gradientes de los parámetros desaparezcan exponencialmente con el número de *qubits* utilizados, impidiendo una optimización manejable de sistemas grandes.

- El último problema es saber hasta que punto este algoritmo es resistente al ruido cuántico. Muchos algoritmos cuánticos han tenido la capacidad de aprender de estos ruidos no obstante, la eliminación o mitigación de dichos problemas a veces requiere un coste mayor al beneficio a obtener.

Por tanto vemos que este algoritmo pese a presentar algunos inconvenientes es un firme candidato a la implementación en ordenadores NISQ.

Y es que, según avance el hardware cuántico lo harán los algoritmos.

En lo que sigue se dará una visión completa pero resumida de todo lo que significa este algoritmo. Si desea trabajar más en detalle sobre el mismo le recomiendo que acceda a la bibliografía de este trabajo donde encontrará otros más extensos.

Capítulo 4

El modelo de Ising

El modelo de Ising permite entender el comportamiento de materiales ferromagnéticos. Se trata un modelo tremendamente útil pues posee soluciones exactas sin necesidad de hacer aproximaciones, por lo menos hasta añadir un campo magnético[4].

El modelo de Ising consiste en N sitios en una red d -dimensional donde encontramos un espín en cada uno de esos sitios. El espín puede estar apuntando hacia arriba o hacia abajo siguiendo los modelos cuánticos conocidos.

Estas posiciones otorgan valores propios a los espines que denotaremos como $+1$ si el espín apunta hacia arriba y -1 si apunta hacia abajo. Se puede asociar a cada posición un observable.

Esta idea es importante porque al estar asentados en campos magnéticos dichos campos otorgan una ventaja energética a los que apunten hacia arriba.

El modelo de Ising es interesante porque trabaja con la interacción entre espines vecinos por lo que trabajaremos con un Hamiltoniano sencillo que será:

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \otimes \sigma_j^z - \sum_i h \sigma_i^x, \quad (4.1)$$

donde hemos usado una notación compacta que mantiene el producto tensorial con la matriz identidad en las posiciones no indicadas, manteniendo así la dimensión correcta del Hamiltoniano, además encontramos que la suma $\langle i,j \rangle$ se produce en todos vecinos adyacentes y donde σ es la matriz de Pauli que nos permite obtener los dos valores propios que necesitamos, h es un factor relacionado con el campo magnético aplicado y J es la constante de interacción que describe la fuerza entre espines [7].

Este Hamiltoniano describe un sistema con un campo magnético en la dirección del eje x y una interacción entre espines en la dirección del eje z .

J es la constante de interacción que describe la fuerza entre espines en la dirección del eje z mientras que h es el término que ajusta el tamaño relativo de la interacción del campo magnético comparada con la fuerza de los pares de espín.

Si $J > 0$ los espines vecinos se colocan en posición paralela y se produce un sistema denominado ferromagnético. Si por el contrario $J < 0$ los espines se colocan en posición antiparalela y se produce un sistema denominado antiferromagnético.

Este modelo se denomina modelo clásico de Ising y el cambio de J afecta a la fuerza entre espines vecinos.

Si $h=0$ encontramos estados fundamentales en el que todos los espines están apuntando hacia una dirección del eje Z , variando el sentido si J es, o no, positiva.

Posteriormente al aplicar el campo magnético externo los estados tienden a alinearse con el mismo.

Esto lo veremos más adelante con el algoritmo programado en la sección 7 y en los resultados experimentales en la sección 9.

Además, trabajando con el modelo de Ising podemos encontrar otro parámetro que dará cuenta de la intensidad del campo magnético y del punto donde se produce una transición de fase. Este parámetro es la magnetización, que es el grado en el que se encuentra un cuerpo una vez está magnetizado. La magnetización es el proceso por el cual un elemento adquiere unas propiedades magnéticas, mediante la aplicación de un campo magnético:

$$M = \frac{1}{N} \sum_i \sigma_i^z, \quad (4.2)$$

que veremos también en el capítulo 7.

Capítulo 5

Elementos del VQE

5.1. Construcción del Hamiltoniano:

El primer paso en la construcción de un buen VQE es la definición del Hamiltoniano sobre el que queremos obtener su estado base. La construcción de este Hamiltoniano es importante pues su complejidad nos proporcionará el coste del resultado final con el correspondiente inconveniente que eso supone cuando hablamos de la optimización en computación cuántica.

La construcción del Hamiltoniano implica encontrar sus operadores y sus pesos en los grados de libertad de las partículas.

En el caso a tratar del modelo de Ising debemos comprender que nuestro Hamiltoniano se compondrá de espines vecinos interactuando con un campo magnético perpendicular externo, esto ocasionará un movimiento en ellos que provocará cambios en su disposición y energía para acoplarse al campo.

5.2. Transformación del Hamiltoniano:

Es importante saber que los ordenadores cuánticos solo miden observables expresados en función de las matrices de Pauli: $\hat{P}_a \in \{I, X, Y, Z\}^{\otimes N}$, con N el número de qubits [5] [10].

X, Y, Z son formas alternativas de escribir los operadores de Pauli σ^x , σ^y y σ^z en la notación de puertas cuánticas vista en el capítulo 2.

Esto es relevante pues en determinadas ocasiones será necesario hacer cambios que permitan estas mediciones.

5.3. Proceso de medida:

Una vez establecida la pauta del Hamiltoniano necesitamos encontrar una estrategia para medir eficientemente y con sentido los valores esperados de la función *ansatz* que introduzcamos. Al tratarse de procesos iterativos parece claro que cuántas menos medidas hagamos para llegar a nuestro resultado esperado mejor optimizado estará nuestro algoritmo [5] [10].

Normalmente, para poder conseguir una precisión de ϵ en el valor esperado de un operador necesitaremos realizar $O(\epsilon^2)$ repeticiones en la ejecución del circuito.

Alguna de las posibilidades para optimizar el número de repeticiones pueden ser el hecho de identificar grupos de operadores que conmuten entre sí y encontrar la base en la que puedan ser medidos todos de manera simultánea.

5.4. Preparación del *ansatz*:

Una vez preparado el Hamiltoniano y entendida la estrategia en la medida nos encontramos con uno de los apartados más importantes cuando tratamos con algoritmos cuánticos variacionales, en nuestro caso, con el VQE [8].

Este es el *ansatz*. El *ansatz* es la estructura de nuestro circuito cuántico parametrizado y lo que aplicaremos directamente a nuestro estado inicial.

El *ansatz* trabaja como un subespacio de nuestro espacio total de Hilbert y se parametriza con ciertos parámetros a elección.

Es por esto que elegir un *ansatz* adecuado es fundamental para encontrar un valor adecuado en nuestra medición sin que tenga un coste computacional elevado.

Estos *ansatz* tienen dos características fundamentales:

- *Expresividad*: Dícese de la capacidad del *ansatz* para abarcar una gran clase de estados en el espacio de Hilbert definiendo la máxima precisión que puede alcanzar su aproximación a los estados relevantes [5] [10].

Hay que encontrar un *ansatz* lo suficientemente expresivo para que alcance una buena aproximación del estado base pero no tan expresivo de tal manera que tenga tantos parámetros que se convierta en intratable.

- *Entrenabilidad*: La entrenabilidad de un *ansatz* explica la capacidad del *ansatz* para ser optimizado usando técnicas trazables en dispositivos cuánticos.

Este *ansatz* es utilizado posteriormente para medir el Hamiltoniano y obtener un determinado valor a optimizar.

5.4.1. Optimización de los parámetros:

Es importante la actualización de los parámetros tras cada vuelta pues deben actualizarse de forma iterativa hasta la convergencia en el estado deseado.

Con esto se quiere decir que la optimización de los parámetros vendrá dado por el valor esperado medido del Hamiltoniano, por ende, se deberá medir varias veces para encontrar la regla que los optimice tras cada vuelta hasta la convergencia [5] [10].

La razón para elegir una optimización adecuada es importante por estos motivos:

- Influye en el número de mediciones necesarias para completar un paso de optimización.
- Afecta al número de iteraciones hasta converger.

5.4.2. Mitigación del error:

Es necesario darse cuenta que el conocido como ruido cuántico es un problema que afecta a la viabilidad del algoritmo cuántico variacional [5] [10].

La idea es mitigar este ruido mediante el procesado de los datos obtenidos al final.

Las técnicas para mitigar este ruido varían en términos de coste por lo que también se requiere encontrar la más eficiente.

En nuestros experimentos trabajaremos con un simulador con ruido y sin ruido para comparar resultados. Véase la sección 7.

Capítulo 6

Funcionamiento del algoritmo

A través del siguiente esquema y comprendido el apartado anterior trataremos de explicar el procedimiento de resolución a través del algoritmo VQE:

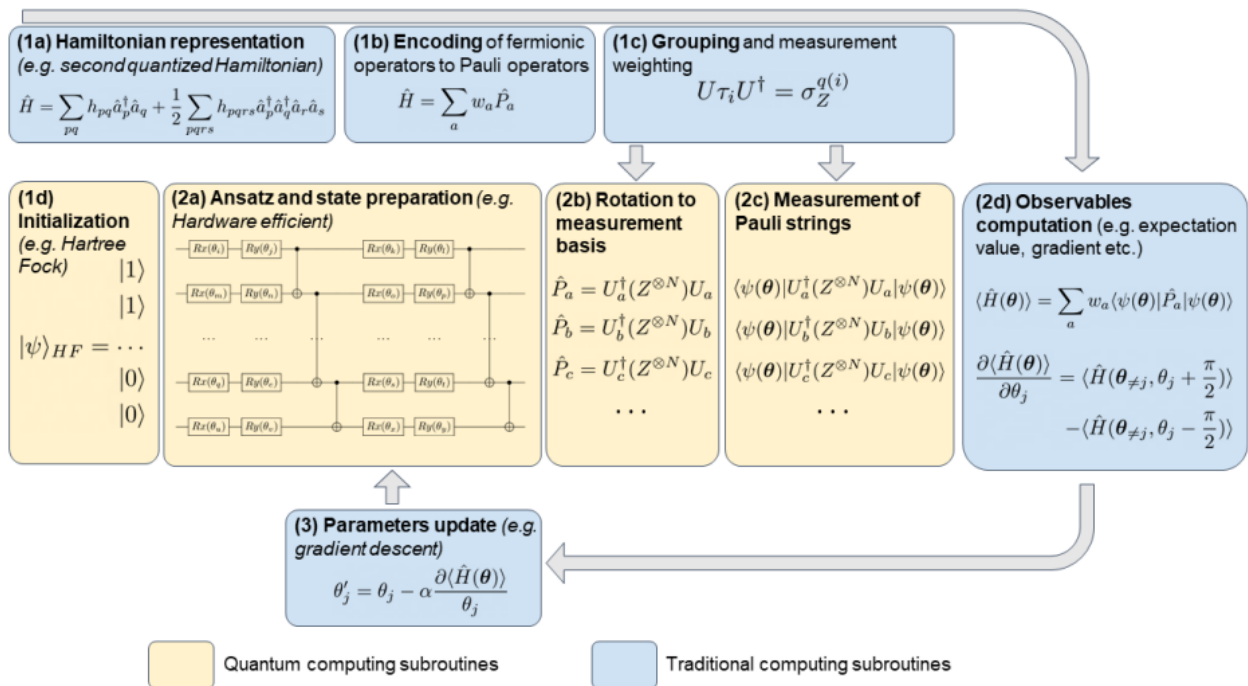


Figura 6.1: Esquema de nuestro algoritmo cuántico variacional[5][10].

6.1. Pre-procesamiento

6.1.1. Representación del Hamiltoniano

Como hemos comentado anteriormente la representación del Hamiltoniano es fundamental.

En nuestro ejemplo del modelo de Ising cabe destacar que trataremos la posición de los espines con los estados $|0\rangle$ y $|1\rangle$.

Así el espín que apunte en el sentido positivo del eje Z será nuestro estado $|0\rangle$ y el espín que apunte en el sentido negativo del eje Z será el estado $|1\rangle$.

Esto nos servirá para tratar más adelante los valores esperados en la base computacional Z, base en la que trabajan todos los ordenadores cuánticos.

6.1.2. Agrupamiento y estrategia de medición

Es interesante ahora vislumbrar la necesidad de agrupar operadores para poder medirlos simultáneamente y optimizar con ello las medidas. Se suele requerir girar la base de medición hasta obtener una donde todos los operadores de un grupo estén diagonalizados.

Es también aquí cuando se decide qué estrategia se seguirá en relación a la ponderación a la hora de medir.

6.2. El cuerpo del algoritmo

6.2.1. Preparación del ansatz

El primer paso de nuestro bucle cuántico es aplicar nuestro *ansatz* al estado inicial. Es importante recordar que nuestro *ansatz* fue creado y todos sus parámetros inicializados anteriormente.

En nuestro caso trabajaremos con el *ansatz* **Efficient SU2** [11]. El circuito **Efficient SU2** está formado por capas de operaciones de qubits simples atravesadas por $SU(2)$ y entrelazamientos. Se trata de un patrón heurístico que puede utilizarse para preparar funciones de onda de prueba para algoritmos cuánticos variacionales o circuitos de clasificación para el aprendizaje automático.

Por ejemplo para un, un circuito con tres *qubits* sería:

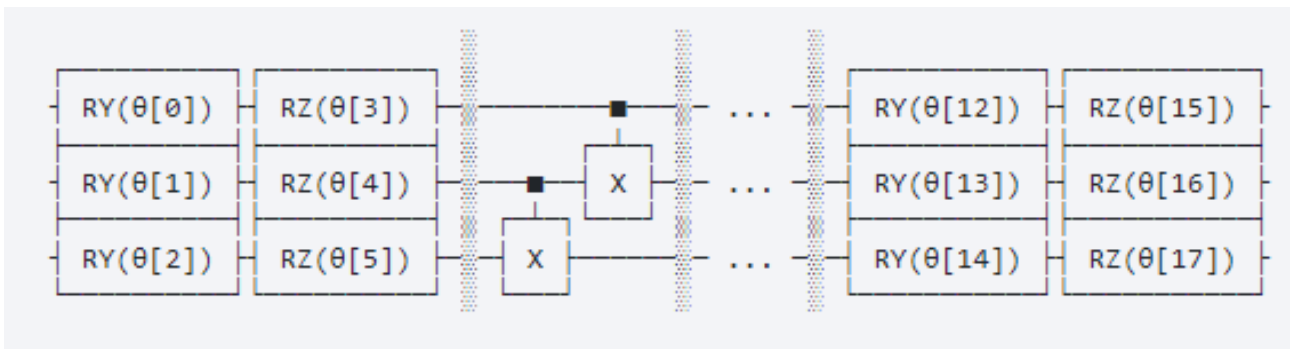


Figura 6.2: **Efficient SU2**[11].

Las puertas RX y RY son rotaciones paramétricas en la esfera de Bloch con ángulos variables ($\theta[0]$ a $\theta[17]$) con respecto a los ejes X y Y , que son versiones con ángulos generales de las puertas X e Y .

Las X controladas se corresponden a las puertas $CNOT$ de la sección (2.4.5).

Con las puertas con parámetros se ajusta el estado. Las puertas $CNOT$ proporcionan entrelazamiento.

6.2.2. Rotación de la base

Una vez hemos obtenido la función de onda de prueba después de haber pasado nuestro estado inicial por el *ansatz* es necesario hacer unos cambios antes de obtener el valor esperado del Hamiltoniano y es rotarla a la base de medida del operador que nos interese o en una base diagonal de un grupo de cadenas de Pauli, igual que tuvimos que transformar nuestro Hamiltoniano, es necesario transformar nuestra función de onda de prueba.

Al tener que trabajar siempre en la base computacional Z en los ordenadores, será necesario cambios de base para poder trabajar con puertas ortogonales a la Z . Véase la sección 7.1 del capítulo 7.

6.2.3. Medida final

Hecho todo lo anterior llega el momento de medir el valor esperado y comprobar, esta estrategia es muy común en cualquier proceso variacional tratado y suelen usarse técnicas de aprendizaje automático.

6.2.4. Actualización de parámetros

Una vez visto el resultado final podemos actualizar y mejorar los parámetros del *ansatz* o dejar programado un aprendizaje automático que permita una optimización continua en base a iteraciones en el algoritmo.

6.3. Post-procesamiento

6.3.1. Mitigación del error

Como vimos anteriormente, estos algoritmos están muy ligados al ruido cuántico y sus problemas.

La idea es aplicar capas computacionales adicionales al terminar la medición con el propósito de reducir este problema.

Capítulo 7

Experimentos

7.1. Cálculo de valores esperados

El punto más importante de estos algoritmos es el hecho de conocer y saber medir los valores esperados de las matrices de Pauli que serán las puertas cuánticas que colocaremos en nuestro circuito. El resto de operadores serán combinaciones de estas.

Vamos a ver cómo podemos calcular estos valores en la plataforma de **Qiskit** [11].

7.1.1. 1 qubit

Vamos a trabajar con un qubit para ir entendiendo cómo podemos programar y visualizar los valores esperados [11].

7.1.1.0.1 Base Z

Vamos a trabajar en la base estándar $\{|0\rangle, |1\rangle\}$.

Calculemos lo siguiente, $\langle Z \rangle$, el valor esperado de Z para un estado qubit cualquiera $|q\rangle$.

$$\langle Z \rangle = \langle q|Z|q\rangle = \langle q|0\rangle\langle 0|q\rangle - \langle q|1\rangle\langle 1|q\rangle = |\langle 0|q\rangle|^2 - |\langle 1|q\rangle|^2. \quad (7.1)$$

Observamos que podemos calcular cualquier valor esperado de Z con una resta de dos probabilidades. Veamos cómo podemos hacer esto en el programa qiskit.

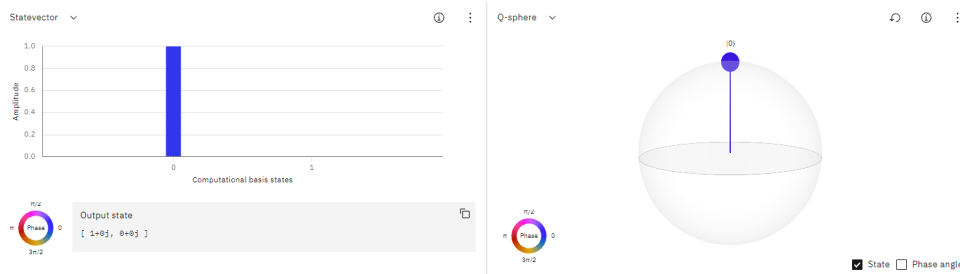


Figura 7.1: Qubit $|0\rangle$

Este estado es muy simple ya que nada más encender el simulador nos encontramos con el estado $|0\rangle$. Dicho esto y aplicando la ecuación anteriormente nombrada obtenemos que el valor esperado de Z en este estado es 1 como nos dice la teoría.

Vamos a aplicar una puerta Hadamard al estado $|0\rangle$ para obtener una superposición equiprobable de ambos estados, $|q\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

En este caso obtenemos las siguientes probabilidades:

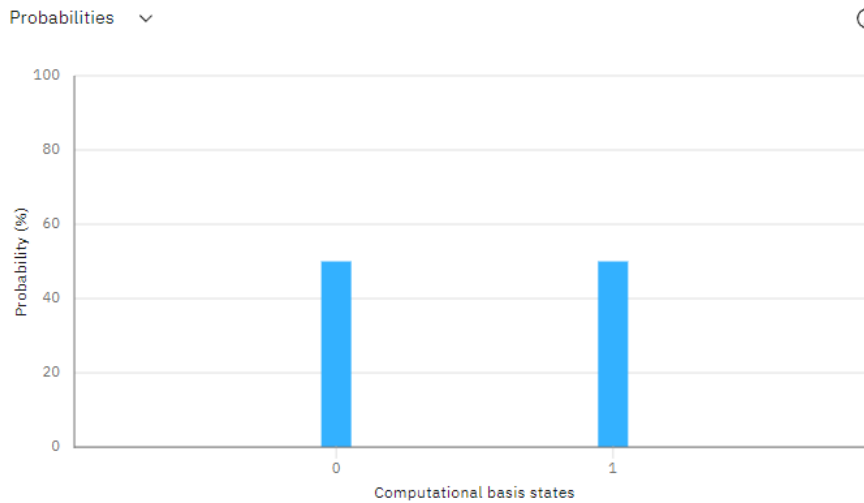


Figura 7.2: Probabilidades del estado $|q\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

Aplicando la fórmula (31), vemos que obtenemos un valor esperado 0.

7.1.1.0.2 Base X

Ya que hemos escrito este último estado aplicando una puerta de Hadamard vamos a hablar de otra base muy usual en computación cuántica. La base en la que la matriz de Pauli X es diagonal y que viene dada por, $\{|+\rangle, |-\rangle\}$.

Estos estados $|+\rangle$ y $|-\rangle$ no son más que puertas Hadamard aplicadas a los estados $|0\rangle$ y $|1\rangle$ respectivamente.

Si quisiésemos calcular ahora los valores esperados de X por ejemplo, empezaríamos haciendo lo mismo que hicimos con Z. Sabiendo que, $X = |0\rangle\langle 1| + |1\rangle\langle 0|$, o en su propia base, $X = |+\rangle\langle +| - |-\rangle\langle -|$.

$$\langle X \rangle = \langle q|X|q\rangle = \langle q|0\rangle\langle 1|q\rangle + \langle q|1\rangle\langle 0|q\rangle = |\langle +|q\rangle|^2 - |\langle -|q\rangle|^2. \quad (7.2)$$

Aquí podemos obtener las amplitudes de nuestros estados que hayamos introducido en qiskit y calcularlo o hacerlo con las probabilidades si trabajásemos en la base $\{|+\rangle, |-\rangle\}$.

Veamos un ejemplo con el anterior estado que teníamos, $|q\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

Este estado es el estado $|+\rangle$. Por ende, calculando el valor esperado observamos que:

$$\langle X \rangle = \langle +|X|+\rangle = \langle +|+\rangle\langle +|+\rangle - \langle +|-\rangle\langle -|+\rangle = 1 - 0 = 1. \quad (7.3)$$

Con Y y su propia base donde es diagonal pasaría exactamente lo mismo.

7.1.2. 2 qubits

Con 2 qubits la idea es igual solo que los estados se complican.

Veamos primero como calcular el valor esperado $\langle ZZ \rangle$:

$$\langle ZZ \rangle = \langle q | ZZ | q \rangle = \langle q | (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes (|0\rangle\langle 0| - |1\rangle\langle 1|) | q \rangle = |\langle 00 | q \rangle|^2 - |\langle 01 | q \rangle|^2 - |\langle 10 | q \rangle|^2 + |\langle 11 | q \rangle|^2. \quad (7.4)$$

Con esto podemos calcular valores esperados de estados como por ejemplo:

$$Tri1 = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle). \quad (7.5)$$

Aplicándolo en nuestro simulador sabiendo que tenemos que usar una puerta Hadamard y un C-NOT para conseguir el estado deseado.

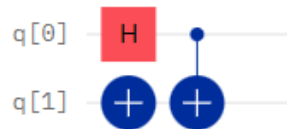


Figura 7.3: Estado $Tri1 = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle)$

Obteniendo las siguientes probabilidades para aplicarlas en la fórmula:

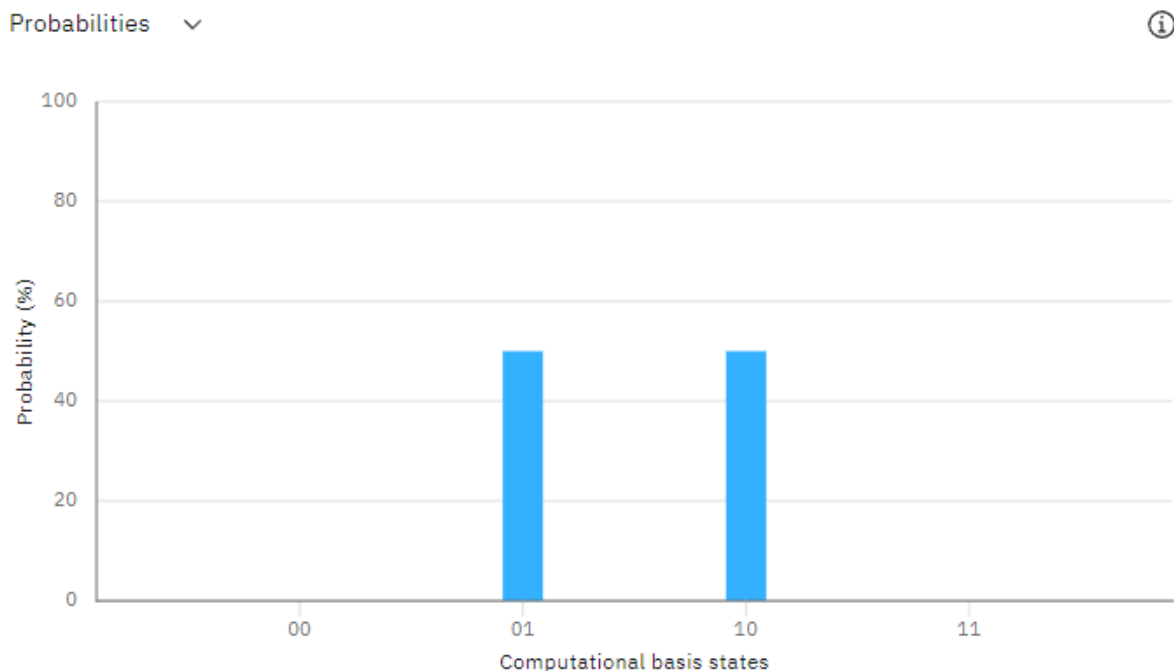


Figura 7.4: Probabilidades $Tri1 = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle)$

Obteniendo que el valor esperado es igual a -1 como podemos ver usando el álgebra de la mecánica cuántica.

$$\begin{aligned} \langle Tri1|ZZ|Tri1\rangle &= \frac{1}{\sqrt{2}}(0, 1, 1, 0) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}(0, 1, 1, 0) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{2}(0 \quad -1 \quad -1 \quad 0) \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = -1. \end{aligned}$$

Llegado el momento vamos a realizar un algoritmo con **Qiskit** [11] que nos permita resolver el modelo de Ising.

Para ello lo que haremos será crear dicho algoritmo variacional que nos permita obtener la menor energía para los distintos estados que introduzcamos partiendo de un Hamiltoniano y unas constantes dadas.

7.2. Algoritmo VQE para la resolución de un modelo de Ising programado en Python

Vamos a trabajar en el *Quantum Lab* del IBM usando *Qiskit* que utiliza el lenguaje de programación *Python*.

Para nuestro ejemplo sacaremos la energía mínima y la magnetización de un modelo de Ising de 4 qubits cuyos espines situados de manera natural en la dirección de eje Z serán expuestos a un campo magnético perpendicular en la dirección del eje X.

Esto generará distintas configuraciones energéticas para distintos valores de acoplamiento y de campo externo.

El Hamiltoniano que vamos a tratar es el siguiente:

$$H_4 = -J\sigma_1^z\sigma_2^zI_3I_4 - JI_1\sigma_2^z\sigma_3^zI_4 - JI_1I_2\sigma_3^z\sigma_4^z - h(\sigma_1^xI_2I_3I_4 + I_1\sigma_2^xI_3I_4 + I_1I_2\sigma_3^xI_4 + I_1I_2I_3\sigma_4^x). \quad (7.6)$$

El primer paso en nuestro código es importar la librería *numpy* e importar las librerías estándar de **Qiskit** de las que podremos obtener determinadas funciones.

```
1 import numpy as np
2
3 from qiskit import QuantumCircuit, transpile, Aer, IBMQ, execute
4 from qiskit.tools.jupyter import *
5 from qiskit.visualization import *
6 from ibm_quantum_widgets import *
7 from qiskit.providers.aer import QasmSimulator
8 from qiskit.circuit.library import EfficientSU2
```

A continuación definiremos una función llamada *signo* que, introduciendo los bits que componen el estado cuántico y el operador correspondiente, nos devuelve el signo que acompaña al estado.

```
1 def signo(bits, operador):
2     signo=1
3     n=len(operador)
4     for j in range(n):
```

```

5         if operador[n-1-j]!='I':
6             if bits[j]=='1':
7                 signo=signo*(-1)
8
9         return signo

```

Siguiendo con el algoritmo procedemos a definir una función que nos devuelva el valor esperado introduciendo un circuito cuántico concreto y operadores cuánticos como una cadena.

```

1 def getvaloresperadoRES(circ, operador:str):
2
3     nq=len(operador)
4     VALORESPERADO = 0
5     circOperator=QuantumCircuit(n,0)
6     circOperator.compose(circ, inplace=True)
7
8     for l in range(len(operador)):
9         if operador[l]=='X':
10             circOperator.h(l)
11             circOperator.measure_all()
12             qU = transpile(circOperator, simulator)
13
14
15
16         job = execute(qU, simulator, shots=1024)
17
18
19         result = job.result()
20         probs=result.get_counts()
21
22
23         for k in probs:
24
25             VALORESPERADO = VALORESPERADO+probs[k]/1024*signo(k, operador)
26
27     return VALORESPERADO

```

El siguiente paso define una función en la que, introduciendo ya los parámetros, los operadores y las constantes del Hamiltoniano correspondiente, te devuelve la energía del mismo.

Aquí hemos llamado a una librería de **Qiskit**, para introducir uno de los ansatz que podíamos haber introducido. Probar con distintos ansatz aceptables puede otorgar mejores o peores resultados.

```

1 from qiskit.circuit.library import EfficientSU2
2 def get_listaprobabilidades(parametros:[], operadores:[], constantes:[]):
3
4     n=len(operadores[0])
5
6     ansatz = EfficientSU2(n, reps=1)
7     qc = QuantumCircuit(n)
8     qc.compose(ansatz, inplace=True)
9     bind_dict = {}
10
11     k=0
12     for key in ansatz.parameters:
13         bind_dict[key] = parametros[k]
14         k=k+1

```



```

15 qc.assign_parameters(bind_dict, inplace = True)
16
17
18
19 EH=0
20 for k in range(len(constantes)):
21     EH=EH+getvaloresperadoRES(qc, operadores[k])*constantes[k]
22
23
24 return EH

```

Con este paso llegamos a la piedra angular del algoritmo pues vamos a obtener la energía mínima y los parámetros del ansatz que han permitido llegar a la misma.

Para ello debemos importar primero nuevas funciones de **Qiskit**, más adelante definiremos los operadores de nuestro Hamiltoniano de 4 qubits y llamaremos al simulador.

Además debemos añadir que parámetro h , la amplitud del campo magnético, del Hamiltoniano queremos escribir, en nuestro caso vamos a calcular los valores para 4 de ellos y comprobaremos sus resultados. También añadimos el vector J , donde incluimos las amplitudes de campo magnético si las hay, que es la parte de las *constantes* que serán introducidas en la función *getvaloresperadoRES* que definimos anteriormente.

Con todo ejecutaremos la casilla y obtendremos que cada valor de h nos ejecuta cinco veces la variación para obtener el valor energético menor, con el que al final nos quedamos junto con los parámetros del ansatz.

```

1 from qiskit.algorithms.optimizers import GradientDescent
2 from qiskit.algorithms.optimizers import SPSA
3 OP=['ZZII', 'IZZI', 'IIZZ', 'XIII', 'IXII', 'IIXI', 'IIIX']
4 Energias=[]
5 Estados=[]
6 simulator = Aer.get_backend("statevector_simulator")
7
8 for h in [0.04, 0.06, 1.1, 0.8]:
9     J=[-1, -1, -1, -h, -h, -h, -h]
10
11     spsa = SPSA(maxiter=100)
12     n=len(OP[0])
13
14     ansatz = EfficientSU2(n, reps=1)
15     nparams=ansatz.num_parameters
16     Emin=100
17     for k in range(1):
18         InitialParams=(np.random.random([nparams]) - 0.5)*2*np.pi
19         result = spsa.minimize((lambda x: get_listaprobabilidades(x, OP, J)),
InitialParams)
20         if result.fun < Emin:
21             Emin=result.fun
22             Estadofundamental=result.x
23
24     print(result, Emin, Estadofundamental)
25     Energias.append(Emin)
26     Estados.append(Estadofundamental)

```

Hecho esto solo queda calcular la magnetización.

$$M = \frac{1}{N} \sum_i \sigma_i^z. \quad (7.7)$$

Para ello cogemos los parámetros de nuestra energía mínima y los introducimos en nuestra función `getlistaprobabilidades`.

Si se prueba con distintos `ansatz` como: `ansatz=TwoLocal(n, 'ry', 'cx', 'linear', reps=1, insert_barriers=True)` se puede ver bien que existe simetría en z y puede tener dos estados fundamentales.

Es decir, si con $J > 0$ tienes comportamiento ferromagnético los espines se acaban alineando.

En nuestro caso eso vale tanto para el estado $|0000\rangle$ como para $|1111\rangle$ y los dos son soluciones válidas de la misma energía.

Como se empieza con estados al azar se pueden tener saltos entre esas configuraciones. Eso puede explicar cambios de signo en la magnetización media que puedan salir ya que siempre puedes girar sobre el eje z y quedarte con el módulo.

```
1 ParametrosEstadofundamental=[-2.57687621, 0.6703786, -2.49962733, -2.25149458,
  -0.15748387, 0.00814727, 0.01902273, 1.49975986, 1.46341504, -1.55722019,
  -1.51506515, 1.59391282, 3.09565379, 0.04934559, -0.24643992, -0.90424364]
2 get_listaprobabilidades(ParametrosEstadofundamental, ['ZIII', 'IZII', 'IIZI', 'IIIZ',
  ''], [1.0/4.0, 1.0/4.0, 1.0/4.0, 1.0/4.0])
```

Además, a continuación, se añadirá una nueva celda que calcula los parámetros de la mínima energía teórica correspondiente a los parámetros generados en nuestro cálculo variacional y el estado cuántico asociado. que nos servirá en la sección 9 para explicar el hecho de la alineación de espines antes y después de la aplicación del campo magnético.

```
1 def state(parametros, n):
2     simulator = Aer.get_backend("statevector_simulator")
3     ansatz = EfficientSU2(n, reps=1)
4     #ansatz=TwoLocal(n, 'ry', 'cx', 'linear', reps=1,
5     insert_barriers=True
6     qc = QuantumCircuit(n)
7     qc.compose(ansatz, inplace=True)
8     bind_dict = {}
9     k=0
10    for key in ansatz.parameters:
11        bind_dict[key] = parametros[k]
12        k=k+1
13
14    qc.assign_parameters(bind_dict, inplace = True)
15    job = execute(qc, simulator, shots=1)
16    result = job.result()
17    return result.get_statevector()
18
19 print(abs(np.round(state([-2.57687621, 0.6703786, -2.49962733, -2.25149458,
  -0.15748387, 0.00814727, 0.01902273, 1.49975986, 1.46341504, -1.55722019,
  -1.51506515, 1.59391282, 3.09565379, 0.04934559, -0.24643992, -0.90424364],4),
  3))**2)
```

Para completar el algoritmo, introduciré el código para obtener el valor de la energía mínimo para un valor de h dado.

```
1 import numpy as np
2
3 # Importing standard Qiskit libraries
4 from qiskit import QuantumCircuit, transpile, Aer, IBMQ
5 from qiskit.tools.jupyter import *
6 from qiskit.visualization import *
7 from ibm_quantum_widgets import *
8 from qiskit.providers.aer import QasmSimulator
```

```

9
10 # Loading your IBM Quantum account(s)
11 provider = IBMQ.load_account()

```

Después de haber importado de nuevo las librerías vamos a definir las matrices que nos ayudarán a formar el Hamiltoniano.

```

1 Z= np.matrix([[1,0],[0,-1]])
2 X= np.matrix([[0,1],[1,0]])
3 I2=np.identity(2)
4 h=1

```

Con esto podremos definir el Hamiltoniano de nuestro ejemplo:

```

1 H=-(np.kron(np.kron(np.kron(Z,Z),I2),I2))-(np.kron(I2,(np.kron(np.kron(Z,Z),I2))
  ))-(np.kron(I2,np.kron(I2,np.kron(Z,Z))))-h*((np.kron(np.kron(np.kron(X,I2),
  I2),I2))+np.kron(I2,np.kron(np.kron(X,I2),I2)))+(np.kron(I2,np.kron(I2,np.
  kron(I2,X))))

```

Siguiendo el código podemos obtener los autovalores y autovectores:

```

1 eigenvalue , eigenvector =np.linalg.eigh(H,UPL0='L')

```

Visto lo anterior ya podemos obtener el autovalor mínimo y comprobar que es muy cercano a nuestro valor en el algoritmo variacional:

```

1 Minautoval=np.min(eigenvalue)

```

Por tanto se podría introducir cualquier Hamiltoniano, parámetros externos, operadores y el resto de cosas necesarias para obtener los estados mínimos.

Capítulo 8

Resultados. Análisis de un sistema de 4 espines.

Vamos a comentar todo lo que hemos ido obteniendo a partir del código de la sección anterior.

Lo primero que haremos será comentar la relevancia que tienen J y h en los estados cuánticos. Recordemos, de la sección del modelo de Ising, el Hamiltoniano con el que estamos trabajando:

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \otimes \sigma_j^z - \sum_i h \sigma_i^x, \quad (8.1)$$

donde encontramos que la suma $\langle i,j \rangle$ se produce en todos vecinos adyacentes y donde σ es la matriz de Pauli que nos permite obtener los dos valores propios que necesitamos, h es un factor relacionado con el campo magnético aplicado y J es la constante de interacción que describe la fuerza entre espines.

El hamiltoniano, como vimos anteriormente, es:

$$H_4 = -J \sigma_1^z \sigma_2^z I_3 I_4 - J I_1 \sigma_2^z \sigma_3^z I_4 - J I_1 I_2 \sigma_3^z \sigma_4^z - h \left(\sigma_1^x I_2 I_3 I_4 + I_1 \sigma_2^x I_3 I_4 + I_1 I_2 \sigma_3^x I_4 + I_1 I_2 I_3 \sigma_4^x \right). \quad (8.2)$$

Con esta situación vamos a dividir nuestro estudio para $h=0$ y $h>0$:

8.1. $h=0$

Como sabemos que h está relacionado con la intensidad del campo magnético aplicado parece evidente que $h=0$ nos indica una ausencia del mismo. Podemos expresarlo también pensando que $h \ll J$, es decir, que el campo magnético no es suficientemente intenso para competir con el acoplamiento entre espines.

Esto se verá reflejado en la posición de los espines, los cuales estarán apuntando en la dirección del eje Z.

Lo que haremos a continuación será, en base a nuestro código anteriormente escrito, fijar el valor de $h=0$ y cambiar J entre dos valores, 1 y -1. Con ello obtendremos un estado cuántico que nos permitirá ver si lo que hemos comentado anteriormente es correcto.

8.1.1. $J=-1$

El estado resultante es:

$$\begin{array}{l} 1 [0.00000e+00 \ 1.00000e-06 \ 2.60000e-05 \ 0.00000e+00 \ 8.90000e-05 \ 5.24233e-01 \\ 2 \ 3.40000e-04 \ 4.00000e-05 \ 4.00000e-05 \ 2.90000e-04 \ 4.74857e-01 \ 6.50000e-05 \\ 3 \ 0.00000e+00 \ 2.50000e-05 \ 2.00000e-06 \ 0.00000e+00] \end{array}$$

Siendo el estado, de manera más visual, el siguiente:

$$\begin{aligned} |\psi\rangle = & 10^{-6} |0001\rangle + 2,6 \cdot 10^{-5} |0010\rangle + 8,9 \cdot 10^{-5} |0100\rangle + 5,24 \cdot 10^{-1} |0101\rangle \\ & + 3,4 \cdot 10^{-4} |0110\rangle + 4 \cdot 10^{-5} |0111\rangle + 4 \cdot 10^{-5} |1000\rangle + 2,9 \cdot 10^{-4} |1001\rangle \\ & + 4,74 \cdot 10^{-1} |1010\rangle + 6,5 \cdot 10^{-5} |1011\rangle + 2,5 \cdot 10^{-5} |1101\rangle + 2 \cdot 10^{-6} |1110\rangle \end{aligned}$$

Este estado es aproximadamente igual a escribir el siguiente:

$$|\psi\rangle \approx 5,24 \cdot 10^{-1} |0101\rangle + 4,74 \cdot 10^{-1} |1010\rangle \quad (8.3)$$

Aquí observamos que, en ausencia de campo magnético externo, las probabilidades son despreciables excepto en el estado $|0101\rangle$ y el estado $|1010\rangle$, estados contrarios en posiciones adyacentes.

Esto, como comentamos en el capítulo 6, implica que los valores 0 en el estado, son espines en dirección positiva del eje Z , mientras que los valores 1 , son espines en la dirección negativa del eje Z .

$$|\psi\rangle \approx 5,24 \cdot 10^{-1} |\uparrow\downarrow\downarrow\rangle + 4,74 \cdot 10^{-1} |\downarrow\uparrow\uparrow\rangle \quad (8.4)$$

Finalmente, obtenemos una superposición, más o menos uniforme, de los dos estados posibles que son estables, de mínima energía.

8.1.2. $J=1$

El estado resultante es:

$$\begin{array}{l} 1 [0.00000e+00 \ 0.00000e+00 \ 0.00000e+00 \ 9.70000e-04 \ 0.00000e+00 \ 0.00000e+00 \\ 2 \ 0.00000e+00 \ 2.90000e-05 \ 0.00000e+00 \ 1.00000e-05 \ 0.00000e+00 \ 2.50000e-03 \\ 3 \ 0.00000e+00 \ 3.70000e-05 \ 4.00000e-06 \ 9.96914e-01] \end{array}$$

Siendo el estado, de manera más visual, el siguiente:

$$\begin{aligned} |\psi\rangle = & 9,7 \cdot 10^{-4} |0011\rangle + 2,9 \cdot 10^{-5} |0111\rangle + 1 \cdot 10^{-5} |1001\rangle \\ & + 2,5 \cdot 10^{-3} |1011\rangle + 3,7 \cdot 10^{-5} |1101\rangle + 4 \cdot 10^{-6} |1110\rangle + 9,96 \cdot 10^{-1} |1111\rangle \end{aligned}$$

Este estado es, aproximadamente, igual que el siguiente:

$$|\psi\rangle \approx 9,96 \cdot 10^{-1} |1111\rangle \quad (8.5)$$

Este estado vemos que tiene casi toda la probabilidad en el estado $|1111\rangle$ confirmando que los espines con un campo magnético externo nulo se mantienen en la dirección del eje Z .

Siguiendo los comentarios del capítulo 6, este estado indica que todos los espines están en la dirección del eje Z pero en el sentido negativo.

$$|\psi\rangle \approx 9,96 \cdot 10^{-1} |\downarrow\downarrow\downarrow\downarrow\rangle \quad (8.6)$$

Visualizamos además, que la diferencia en la J provoca acoplamientos paralelos o antiparalelos aunque manteniéndose en la dirección del eje Z.

8.2. $h \gg J$

Vamos a comprobar ahora como aumentando el campo magnético los espines tienden a alinearse con él.

En este caso vamos a colocar una $h=100$, la J en este caso no va a servirnos para visualizar nada en concreto pues nos van a quedar probabilidades muy similares en todos los estados, ya que mide la interacción entre espines y en este caso el campo magnético prevalece.

El estado resultante es:

```
1 [0.05629 0.064045 0.0533 0.060029 0.067181 0.076225 0.067604 0.075701
2 0.053069 0.059949 0.049738 0.055901 0.061153 0.069194 0.06152 0.069653]
```

Siendo el estado, de manera más visual, el siguiente:

$$\begin{aligned} |\psi\rangle = & 0,056 |0000\rangle + 0,064 |0001\rangle + 0,053 |0010\rangle + 0,060 |0011\rangle \\ & + 0,067 |0100\rangle + 0,076 |0101\rangle + 0,067 |0110\rangle + 0,075 |0111\rangle \\ & + 0,053 |1000\rangle + 0,059 |1001\rangle + 0,049 |1010\rangle + 0,055 |1011\rangle \\ & + 0,061 |1100\rangle + 0,069 |1101\rangle + 0,061 |1110\rangle + 0,069 |1111\rangle \end{aligned}$$

Viendo que es igualmente probable cada uno de esos estados cuánticos, demostrando que los espines están intentando alinearse en la dirección del campo.

8.3. Magnetización y transición de fase

8.3.1. Simulador sin ruido

Siguiendo con el algoritmo programado vamos a comprobar, viendo una gráfica, los valores de la magnetización.

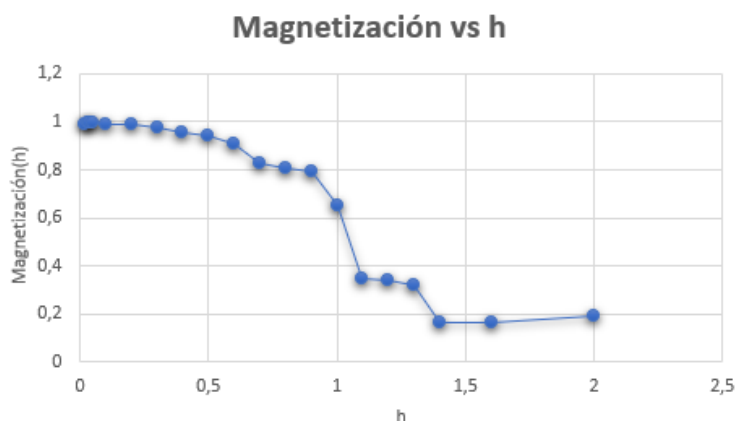


Figura 8.1: Gráfica de la magnetización respecto a los valores de h . Programado en Python.

Se observa como en los valores de $h=1$ se produce un cambio brusco en la magnetización, que producirá un cambio de estado. Según aumentamos el campo magnético se produce una lucha entre los dos términos del Hamiltoniano y el sistema pasa a ser paramagnético.

Para valores pequeños de h la magnetización tiende a 1 prácticamente ya que corresponde al estado ferromagnético. Si $h \approx 0$ recordemos que el modelo cuántico se reduce prácticamente a un modelo de Ising clásico, en este caso, tenemos una transición de fase a temperatura finita desde una fase paramagnética en la que los espines están desordenados por las fluctuaciones térmicas a una fase ferromagnética en la que todos apuntan a la dirección z y, en consecuencia, un estado básico ferromagnético a $T=0$.

Vamos a darnos cuenta que en nuestro cálculo estamos inspeccionando el estado básico del sistema. Dado que $T=0$, las fluctuaciones que producen la transición son puramente cuánticas.

Esto podemos relacionarlo con el aumento de la temperatura en el modelo clásico de Ising, donde son las fluctuaciones térmicas las que producen una transición de fase clásica de un estado ferromagnético a uno paramagnético.

Por esta razón, el estado que observamos a altas intensidades de campo magnético se denomina paramagnético cuántico.

Si lo comparásemos con la solución exacta tras la resolución del Hamiltoniano,

$$H_4 = -J\sigma_1^z\sigma_2^zI_3I_4 - JI_1\sigma_2^z\sigma_3^zI_4 - JI_1I_2\sigma_3^z\sigma_4^z - h(\sigma_1^xI_2I_3I_4 + I_1\sigma_2^xI_3I_4 + I_1I_2\sigma_3^xI_4 + I_1I_2I_3\sigma_4^x), \quad (8.7)$$

podríamos identificar cómo de exacto es nuestro modelado variacional.

Esta solución la vamos a obtener a partir de la programación en **Julia** [1]. En este caso se obtuvo la solución exacta diagonalizando todas las matrices y obteniendo una matriz final de dimensión 16×16 . Esto les permitió obtener los estados y valores propios y realizar su propia gráfica de la magnetización:

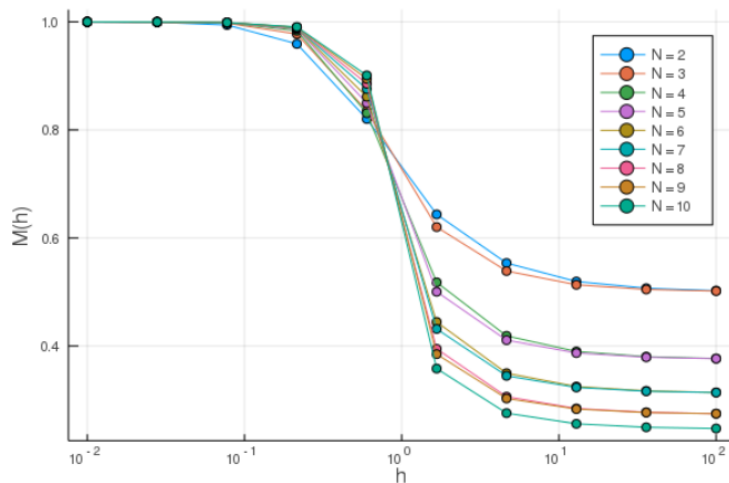


Figura 8.2: Gráfica de la magnetización respecto a los valores de h . Programado en Julia

Como se puede observar, se obtuvo la solución exacta para varios *qubits*. En nuestro caso solo nos interesa $N=4$ pues solamente trabajamos con ese número de ellos.

Los resultados arrojan que en torno al valor $h=1$ se produce la transición de fase que obtuvimos en nuestro experimento.

8.3.1.1. Simulador con ruido

Con todo esto debemos comentar que, aun así, determinadas simulaciones tienen un ruido externo como comentamos anteriormente en teoría, que puede producir algunos cambios en los valores.

Por ello vamos a introducir un simulador con ruido denominado, `simulador = FakeVigoV2()` [11].

El módulo de proveedores falsos contiene clases de proveedores y *backends* falsos. Los *backends* falsos se construyen para imitar los comportamientos de los sistemas **IBM Quantum** utilizando instantáneas del sistema. El **IBM** posee modelos, más o menos exactos, del ruido bajo un número determinado de *qubits* que se ajusta empíricamente sobre el sistema a estudiar.

Con este simulador, introduciendo los mismos parámetros hemos obtenido una gráfica de la magnetización como la siguiente:

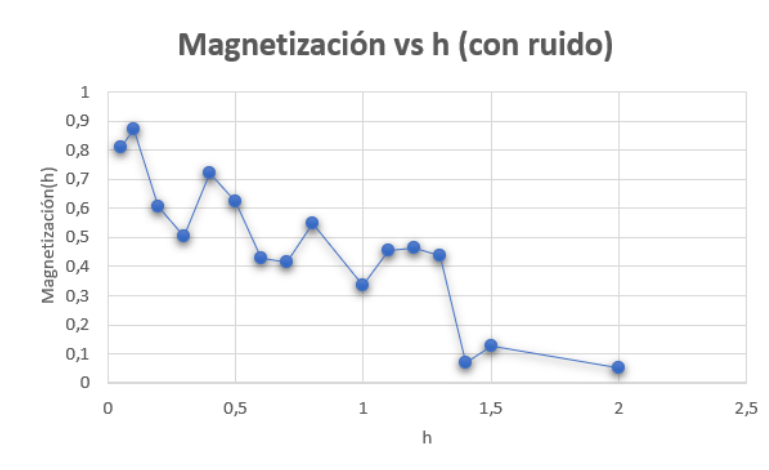


Figura 8.3: Gráfica de la magnetización respecto a los valores de h . Para varios espines. Con ruido.

Como podemos observar los valores están mucho más dispersos dando lugar a medidas erróneas y, la simulación, aunque permite ver la tendencia general, pierde el detalle de la transición de fase.

Capítulo 9

Resumen y conclusiones

En este trabajo hemos visto cuáles son las mejoras de la computación cuántica frente a la computación clásica. La ventaja principal es que podemos trabajar con estados en superposición ahorrando tener que demostrar explícitamente el estado que de otra manera harían de los algoritmos algo intratable.

Si hablamos de los algoritmos cuánticos variacionales encontramos que tan solo necesitamos encontrar una buena expresión del Hamiltoniano y un buen *ansatz* con el que trabajar.

En concreto, se ha presentado *variational quantum eigensolver*, un algoritmo híbrido eficiente a la hora de calcular la energía del estado fundamental del sistema. Este algoritmo tiene un coste computacional relativamente pequeño. Por contra, hemos comprobado que el ruido cuántico puede ser muy relevante.

En este documento, se han cubierto varios objetivos:

- Hemos introducido los postulados de la mecánica cuántica y hemos visto como funcionan los *qubits* y las puertas cuánticas.
- Hemos dado una introducción a las ventajas y desventajas de los algoritmos cuánticos, incluido el VQE.
- Hemos explicado el modelo de Ising y presentado un algoritmo cuántico para su resolución.
- Hemos explicado con profundidad el VQE aplicado a nuestro problema del modelo de Ising.
- Y por último se ha implementado un algoritmo para un Hamiltoniano concreto y poder estudiar así un modelo de Ising concreto.

Al realizar el estudio del sistema de 4 espines hemos podido comprobar como aparecen transiciones de fase cuánticas cuando se aplican campos magnéticos externos comparables a la interacción entre espines vecinos. Aunque los resultados con un *ansatz* sencillo son correctos, al simular un sistema ruidoso se ha podido comprobar que se pierde mucha precisión, sobre todo para encontrar la frontera en la que aparece la transición de fase, lo que limita el alcance de este método para circuitos de cierta profundidad.

Bibliografía

- [1] CARSTEN, B. y KATHARINE, H.
Quantum Ising Phase Transition.
<https://juliaphysics.github.io>
- [2] CEREZO, M. , ARRASMITH, A., BABBUSH, R., ET AL.
Variational quantum algorithms. Nat Rev Phys 3, 625–644, 2021.
- [3] COHEN-TANNOUJDI, C., DIU, B. y LALOE, F.
Quantum mechanics. Wiley.1991
- [4] FAMAF
El modelo de Ising-Notas-Termoll.
<https://www.famaf.unc.edu.ar>
- [5] KOSUKE, M., YUYA, O. N. y WTARU, M..
Theory of analytical energy derivatives for the variational quantum eigensolver. Phys. Rev. Research. Volumen 2, 2020.
- [6] LÓPEZ, M.P., MENESES VIVEROS,A. y MORANTES-LUNA, G..
Algoritmo cuántico de Deutsch y Jozsa en GAMA.
Rev. Mex. Fís. E vol.64 no.2 México jul./dic. 2018
- [7] MARSOLAIS, A.M.
The equiavalence between the Kitaev the transverse, quantum Ising model and the classical Ising model.
<https://etd.ohiolink.edu/apexprod>
- [8] MEITI, O.R., GARD, B.T., BARRON, G.S., ET AL.
Gate-free state preparation for fast variational quantum eigensolver simulations. npj Quantum Inf 7, 155, 2021.
- [9] SOSAPANTA SALAS, L.D.
Efectos cuánticos gravitacionales sobre la estabilidad de acreción hacia un agujero negro de Schwarzschild.
<https://repositorio.unal.edu.co/handle/unal/64741>
- [10] TILLY, J., HONGXIANG, C., SHUXIANG, C., ET AL.
The Variational Quantum Eigensolver: a review of methods and best practices.

- [11] PROGRAMA QISKIT Y TODO LO RELACIONADO CON QISKIT Y EL IBM.
<https://qiskit.org/textbook/ja/ch-applications/vqe-molecules.html>

Todas las imágenes de la esfera de Bloch [12] [13].

- [12] GONZÁLEZ, J.E.
Conceptos matemáticos básicos de computación cuántica
[https://docirs.cl/Puertas Circuitos Cuanticos.asp](https://docirs.cl/Puertas_Circuitos_Cuanticos.asp)
- [13] QUIÑONES, D.
Síntesis de circuitos cuánticos, 2012, month=9
<https://researchportal.bath.ac.uk/en/publications>
doi=10.13140/RG.2.2.10878.38722