

f



Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y Tecnología

Máster en Computación Cuántica

Quantum Simulation of Physical Problems: A Focus on Hydrogen Molecule Simulation

Trabajo Fin de Estudios

presentado por: Pablo Serrano Molinero, Rodrigo Vaca Jácome

Dirigido por: Mariano Caruso

Ciudad: Madrid

Fecha: 11 de septiembre de 2024

Índice de Contenidos

Índice de Ilustraciones	III
Índice de Tablas	IV
Resumen	V
Abstract	VI
1. Introducción	1
1.1. Motivación	1
1.2. Planteamiento del Trabajo	2
1.3. Estructura del Trabajo	2
2. Objetivos	4
3. Contexto y Estado del Arte	5
3.1. Modelo de Ising	5
3.1.1. Descripción física.	6
3.1.2. Aplicaciones.	8
3.1.3. Limitaciones	10
3.2. Problemas tipo QUBO.	11
3.2.1. Formalismo	11
3.2.2. Como abordar limitaciones del formalismo	12
3.2.3. Limitaciones de la computación clásica.	16
3.2.4. Ejemplos de problemas tipo QUBO.	18
3.2.5. Conexión entre Ising y QUBO.	19
3.3. <i>Quantum Annealing</i> para problemas Ising-QUBO	20
3.4. Más allá de Ising-QUBO	23
3.4.1. <i>Variational Quantum Eigensolver</i> (VQE)	24
3.5. Computación Cuántica: Una Guía	26
3.5.1. Computación Cuántica y su Relación con las Ciencias	27
3.5.2. Qubits: Principios Fundamentales y Conceptos	28

3.5.3.	Distinciones Entre Qubits en Ordenadores Cuánticos de Annealing y Basados en Puertas	31
3.5.4.	Clasificaciones y Propiedades de los Qubits	32
3.5.5.	Compuertas Lógicas Cuánticas.	34
3.5.6.	Circuitos Cuánticos	38
3.5.7.	Algoritmos Cuánticos	41
3.5.8.	Plataformas de Computación Cuántica.	47
3.5.9.	Aplicaciones.	52
4.	Desarrollo e Implementación	53
4.1.	La molécula de Hidrógeno	53
4.1.1.	Modelización	54
4.1.2.	Propiedades Físicas	58
4.1.3.	Simulación computacional y sus aplicaciones	61
4.2.	<i>Ansatz</i> del Problema	62
4.3.	Implementación	64
4.3.1.	Configuración de Entorno	64
4.3.2.	Caracterización de la Molécula	66
4.3.3.	Geometría Molecular	68
4.3.4.	Configuración Electrónica y Energía	85
4.4.	Resultados	90
5.	Conclusiones y Trabajo Futuro	92
	Bibliografía	94
A.	Anexos Contexto y Estado del Arte	104
A.1.	Computación Cuántica, su relación con las Ciencias	104
A.2.	Eventos Claves en la Computación Cuántica	108
A.3.	Notación de Dirac	112
A.4.	Algebra Lineal Espacio de Hilbert	115
A.5.	Compuertas Cuánticas	118
A.6.	Qubits: Implementaciones Físicas	123
B.	Códigos	129

Índice de Ilustraciones

3.1. Esquema de una red de spins 2D de un modelo de Ising. Esta ilustración muestra los spins $\sigma_i = +1$ como flechas hacia arriba y $\sigma_i = -1$ como flechas hacia abajo. En la imagen se muestra también la interacción entre spins vecinos mediante las aristas que los unen. Fuente: [13].	6
3.2. Estructura de los spins dentro de un material ferromagnético. Fuente: Wikipedia	7
3.3. Estructura de los spins dentro de un material antiferromagnético. Fuente: Wikipedia	8
3.4. Ilustración del problema <i>Max-Cut</i>	18
3.5. Evolución de los parámetros A y B con el paso del tiempo en el algoritmo <i>Quantum Annealing</i> . Fuente: https://docs.dwavesys.com/	23
4.1. Esquema de la estructura de los orbitales moleculares para diferentes aproximaciones. Fuente: Pennylane	58
4.2. Relación de la Geometría Óptima con el nivel de Energía del Sistema. . . .	61
4.3. Circuitos que implementa las rotaciones de Givens necesarias para la definición del <i>ansatz</i>	63
4.4. Energía Media en función de la Longitud de Enlace resultante del proceso de optimización usando formulación Ising-QUBO con annealing simulado y cuántico.	74
4.5. Resultados Proceso de Optimización Geometría Molecular H_2	82
4.6. Energía en función del paso de optimización.	84
4.7. Longitud de Enlace en función del paso de optimización.	84
4.8. Valor del spin del sistema en cada iteración del optimizador	89
4.9. Valor esperado de la energía de la molécula de H_2 en función de las iteraciones	90

Índice de Tablas

3.1. Compuertas Cuánticas Básicas	35
A.2. Resumen de Conceptos de Álgebra Lineal y Espacio de Hilbert en Mecánica Cuántica	115
A.3. Compuertas Cuánticas	118
A.4. Tabla de Tecnologías para Implementación Física de Qubits	125
A.5. Roadmap	128

Resumen

Los algoritmos de optimización cuántica han emergido como herramientas poderosas para abordar problemas de optimización que son intratables computacionalmente mediante métodos clásicos. Este tipo de problemas aparecen en diversos campos como la física, la química y la ciencia de materiales. En este trabajo, se detalla el estado del arte en optimización cuántica y se aplican estos conceptos a la resolución de problemas relacionados con sistemas moleculares, utilizando como caso de estudio la molécula de H_2 . La implementación del algoritmo VQE (por sus siglas en inglés, *Variational Quantum Eigensolver*) ha permitido obtener con alta precisión la energía del estado fundamental, la configuración electrónica y la longitud de enlace, ilustrando así la utilidad de estos métodos para la resolución de problemas de optimización en sistemas moleculares.

Palabras Clave: Simulación cuántica, optimización combinatoria, Ising, QUBO, VQE, *Quantum Annealing*, algoritmos cuánticos.

Abstract

Quantum optimization algorithms have emerged as powerful tools for addressing optimization problems that are computationally intractable by classical methods. These types of problems arise in various fields such as physics, chemistry, and materials science. In this work, we detail the state of the art in quantum optimization and apply these concepts to the resolution of molecular system problems, using the H_2 molecule as a reference. The implementation of the Variational Quantum Eigensolver (VQE) algorithm has yielded highly accurate results for the ground state energy, electronic configuration, and bond length, demonstrating the usefulness of these methods for solving optimization problems related to molecular systems.

Keywords: Quantum simulation, combinatorial optimization, Ising, QUBO, Quantum Annealing, quantum algorithms.

1. Introducción

1.1. Motivación

La simulación de sistemas físicos complejos es un área fundamental en diversas disciplinas científicas, incluyendo la física, la química y la ciencia de materiales. Comprender de forma precisa el comportamiento de estos sistemas a nivel atómico y molecular es crucial para el desarrollo de nuevas tecnologías y materiales. La necesidad de revivir un evento para entenderlo mejor, es inherente al proceso del conocimiento humano, como Richard Feynman daba a entender "*What I cannot create, I do not understand*" [1]. Sin embargo, los métodos clásicos de simulación computacional se enfrentan a limitaciones significativas al abordar sistemas complejos, debido a la complejidad computacional exponencial que aumenta con el tamaño del sistema [2][3].

La computación cuántica surge como una alternativa prometedora para superar las limitaciones de los métodos clásicos, uno de los ejemplos más famosos es el conocido algoritmo de Shor [4]. Los principios de la mecánica cuántica, como la superposición[5] y el entrelazamiento[6], permiten a las computadoras cuánticas realizar cálculos paralelos masivos, lo que las convierte en herramientas ideales para simular sistemas cuánticos complejos. En este contexto, los algoritmos cuánticos de optimización han ganado relevancia como una herramienta versátil para la simulación cuántica de problemas de optimización combinatoria, incluyendo la simulación de sistemas físicos. La capacidad de simular sistemas físicos complejos abre un abanico de posibilidades en diversas áreas, como ciencia de materiales, química, biología e inteligencia artificial.

Los ordenadores cuánticos aprovechan el poder de la superposición y el entrelazamiento cuántico para abordar ciertos problemas, como problemas de optimización formulados según el modelo de Ising y QUBO, con mayor eficacia que los ordenadores clásicos. Algoritmos como el *Quantum Annealing* permiten explorar en paralelo las configuraciones energéticas del sistema, lo que supone una importante aceleración en la resolución de problemas complejos[7].

La formulación de un problema de optimización en base a un Hamiltoniano, nos permitirá sacar provecho de la potencia de la computación cuántica para la resolución de problemas de optimización combinatoria, permitiendo formular y resolver problemas de optimización. Este marco facilita la traducción de parámetros físicos a variables cuánti-

cas y la implementación de algoritmos cuánticos eficientes para simular la estructura y propiedades de sistemas complejos.

1.2. Planteamiento del Trabajo

Este trabajo de fin de máster se centra en el uso de algoritmos cuánticos de optimización para la simulación cuántica de la molécula de hidrógeno. La molécula de hidrógeno, el sistema molecular más simple, representa un caso de estudio fundamental para evaluar la eficacia de este enfoque en la caracterización de sistemas moleculares.

El objetivo principal de este trabajo es mostrar la capacidad de la computación cuántica para caracterizar la molécula de hidrógeno con precisión y eficiencia. En esta caracterización estudiaremos propiedades fundamentales como la geometría molecular, la configuración electrónica y la energía del estado fundamental. Para este propósito, se formulará la simulación de la molécula de hidrógeno como un Hamiltoniano de coste, mapeando las variables físicas a variables cuánticas. Existen diferentes algoritmos cuánticos diseñados para este propósito, como el *Simulated Annealing*, el *Quantum Annealing* (QA), el algoritmo QAOA o el algoritmo VQE, en este trabajo nos enfocaremos en VQE.

Como resultado de este trabajo, se discutirán las ventajas y limitaciones de VQE para la simulación cuántica de sistemas moleculares y se explorarán sus potenciales aplicaciones en la simulación de sistemas físicos más complejos.

1.3. Estructura del Trabajo

Este trabajo de fin de máster se estructura en cinco capítulos principales:

Introducción: Este primer capítulo, del cual esta sección es parte, presenta como preámbulo la motivación y justificación del tema, el planteamiento del trabajo y la estructura del mismo.

Objetivos del Trabajo: En este capítulo se establecerán los objetivos generales y específicos a alcanzar, además de presentar el problema y la metodología de trabajo a seguir.

Contexto y Estado del Arte: En esta sección se presenta el marco teórico, se revisan los conceptos fundamentales relacionados con el marco Ising-QUBO y sus limitaciones, la computación cuántica, los algoritmos cuánticos y las plataformas cuánticas disponibles actualmente.

Desarrollo del trabajo: Este capítulo describe en detalle el desarrollo de los distintos pasos de la investigación, incluyendo la definición matemática del problema de referencia, la formulación de su Hamiltoniano y la implementación de los algoritmos cuánticos necesarios.

En este capítulo, además se presentan y analizan los resultados obtenidos para las diferentes propiedades moleculares estudiadas, discutiendo la precisión, eficiencia y relevancia de la computación cuántica para la simulación de la molécula de hidrógeno.

Conclusiones y Trabajo Futuro: Este capítulo resume los principales hallazgos del trabajo, presenta las conclusiones obtenidas y propone direcciones de investigación futuras para mejorar la simulación cuántica de problemas físicos.

Además de estos capítulos principales, el trabajo incluye referencias, agradecimientos, disponibilidad de datos y el código desarrollado.

2. Objetivos

El objetivo de este trabajo es ilustrar como la computación cuántica es una herramienta de gran utilidad a la hora de resolver problemas de optimización combinatoria. Este tipo de problemas son de gran interés ya que además de aparecer en numerosos ámbitos como logística, economía, química o ciencia de materiales, son problemas *NP-hard* a la hora de abordarlos con computación clásica, lo cual los hace irresolubles de manera exacta cuando el problema crece [8].

En concreto, en este trabajo nos enfocaremos en como abordar problemas relacionados con la caracterización de sistemas moleculares. Este tipo de sistemas cuentan con diferentes características que podemos modelar y simular mediante el uso de algoritmos de optimización cuánticos.

- Optimización de la geometría molecular
- Búsqueda de la configuración electrónica de una molécula
- Obtención de la energía del estado de fundamental de una cierta molécula

Con estos problemas encima de la mesa, detallaremos la física que los define y haremos uso de su Hamiltoniano de spin y del algoritmo VQE para implementar una solución computacional a dichos problemas. Para ello haremos uso de [PennyLane](#), el software para programación de algoritmos cuánticos desarrollado por la empresa [Xanadu](#).

En nuestro trabajo seguiremos la siguiente metodología:

1. Definición detallada del problema
2. Implementación programática del mismo
3. Discusión de resultados.

Una vez claros los objetivos de este trabajo, comenzaremos introduciendo el estado del arte de este tipo de problemas y la tecnología que los sustenta.

3. Contexto y Estado del Arte

En este capítulo nos enfocaremos en aportar el contexto necesario para abordar el tema de la simulación de sistemas moleculares mediante algoritmos cuánticos de optimización. Comenzaremos introduciendo el modelo de Ising [9] y la formulación QUBO (por sus siglas en inglés, *Quadratic Unconstrained Binary Optimization*), que son las bases fundamentales de la formulación de problemas de optimización en computadores cuánticos. Tras esto, analizaremos las limitaciones que puede presentar este marco y como podemos evitarlas. Por último, detallaremos el paradigma de la computación cuántica, en el cual nos basaremos para la implementación de los problemas a resolver. De esta forma, profundizaremos en el estado del arte de este tipo de procedimientos, sentando así las bases de nuestro trabajo.

3.1. Modelo de Ising

El modelo Ising es un modelo matemático desarrollado con el objetivo de caracterizar el ferromagnetismo desde el punto de vista de la mecánica estadística. El modelo de Ising o modelo de Ising-Lenz [9], fue desarrollado por Ernst Ising en 1925 como solución a un problema que le planteó su profesor Wilhelm Lenz.

El problema que propuso Lenz fue el siguiente:

Tenemos una red de partículas de spin $\sigma_i \in \{\pm 1\}$ dispuestas de tal forma que cada partícula únicamente interactúa con sus vecinos, definiendo así un grafo de interacción entre partículas. Los estados de la red en los cuales las partículas vecinas tienen el mismo spin son estados con menor energía, por lo que en el caso aislado, el sistema tenderá a alinear dichos spins para minimizar la energía del sistema. Al introducir al problema por ejemplo, ruido térmico, aparecen nuevas configuraciones del sistema debido a la interacción de los spins con el ruido térmico además de con sus vecinos.

Por tanto, el objetivo de este modelo es caracterizar dicho sistema, y de esa forma, caracterizar también los posibles cambios de fase del sistema cuando existe una fuente de calor que lo perturbe.

Ising en su tesis de 1924 resolvió el problema para el caso de una red 1D, demostrando que, en dicho caso, no existía transición de fase. Este hecho supuso una gran decepción

para Ising y Lenz, ya que no habrían sido capaces de probar que su modelo era capaz de caracterizar cambios de fase, pero el interés por este problema y su modelo no se quedó únicamente ahí. En 1968, el físico Lars Onsager ganó el premio nobel por resolver de forma analítica el problema de una red cuadrada de 2D utilizando el modelo propuesto por Ising [10], demostrando definitivamente, que el modelo era capaz de modelizar cambios de fase. El modelo de Ising de 2D es uno de los modelos más sencillos capaces de caracterizar cambios de fase.

En la actualidad el modelo de Ising sigue siendo un campo de estudio debido a sus aplicaciones en redes neuronales [11] y en resolución de problemas de optimización como el conocido problema *Max-Cut*[12].

En este trabajo nos enfocaremos en la aplicación del modelo de Ising y sus distintos derivados para la resolución de problemas de optimización.

3.1.1. Descripción física.

Una vez introducido el modelo de Ising, en esta sección profundizaremos en la formulación matemática del mismo.

El problema parte de una red de partículas con spin. Estas partículas están conectadas con sus vecinos mediante un grafo de conectividad que forma una red d -dimensional.

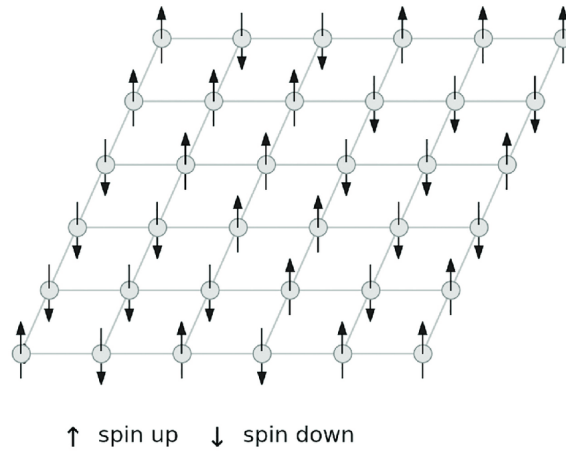


Figura 3.1: Esquema de una red de spins 2D de un modelo de Ising. Esta ilustración muestra los spins $\sigma_i = +1$ como flechas hacia arriba y $\sigma_i = -1$ como flechas hacia abajo. En la imagen se muestra también la interacción entre spins vecinos mediante las aristas que los unen. Fuente: [13].

En una red de spins Δ , denominaremos a cada una de las posiciones en la red $i \in \Delta$

donde cada una de las partículas de spin σ_i puede estar en un estado de spin $\sigma_i \in \{\pm 1\}$. Dado dicho sistema, denominaremos $\sigma = \{\sigma_i\}_{i \in \Delta}$ a su configuración de spin.

Como primera aproximación al problema, podemos considerar el sistema descrito de forma aislada. Para caracterizarlo haremos uso de la formulación hamiltoniana del problema. En este marco debemos definir un operador que caracterice la energía total del sistema, el Hamiltoniano [14]. En este caso el Hamiltoniano del problema de Ising tendría la siguiente expresión:

$$H = - \sum_{i,j \in \Delta} J_{i,j} \sigma_i \sigma_j \quad (3.1)$$

Donde $J_{i,j}$ es el término de interacción entre las partículas vecinas σ_i, σ_j . Este término de interacción caracterizará el comportamiento de nuestro sistema.

Una vez introducido el Hamiltoniano de Ising para el escenario más sencillo, podemos abordar un caso muy habitual del problema, aquel en el que existe un campo magnético sobre el sistema. En este caso definiremos h_i como el campo magnético que actúa sobre la partícula i -ésima de la red, resultando en un Hamiltoniano con la siguiente expresión:

$$H = - \sum_{i,j \in \Delta} J_{i,j} \sigma_i \sigma_j - \mu \sum_{i \in \Delta} h_i \sigma_i \quad (3.2)$$

Donde μ es el momento magnético de las partículas que conforman la red.

Definido el Hamiltoniano que caracteriza nuestro sistema, podemos discutir una serie de casos de interés:

- $J_{i,j} > 0$: En este caso el sistema se comporta como un material ferromagnético. Un material ferromagnético es aquel en el que los spins de sus átomos tienden a estar alineados, y por tanto, es capaz de mostrar magnetización. Los imanes son el ejemplo más convencional de un material ferromagnético.

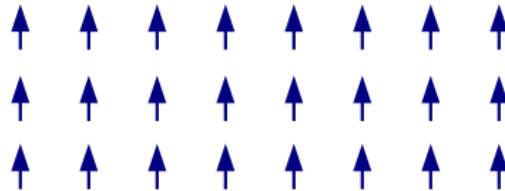


Figura 3.2: Estructura de los spins dentro de un material ferromagnético. Fuente: [Wikipedia](#)

- $J_{i,j} < 0$: Al contrario que en el caso anterior, el sistema se comportará como un material antiferromagnético, en el cual los spins se encuentran completamente despareados los unos de los otros. Estos materiales no presentan magnetización, como por ejemplo, la mayoría de metales de transición.

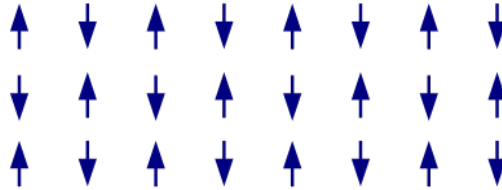


Figura 3.3: Estructura de los spins dentro de un material antiferromagnético. Fuente: [Wikipedia](#)

- $J_{i,j} = 0$: Por último, este es el caso extremo en el cual los spins no interactúan entre sí.

En cuanto al campo magnético h :

- $h_i > 0$: Cuando el campo magnético h_j es positivo, el campo fomenta que el spin σ_j tienda a tomar su valor positivo.
- $h_i < 0$: Cuando el campo magnético h_j es negativo, el campo fomenta que el spin σ_j tienda a tomar su valor negativo.
- $h_i = 0$: No existe una dirección privilegiada del espacio definida por un campo magnético.

Definido de forma general el modelo de Ising, es importante destacar que en base a este modelo, somos capaces de modelizar una gran variedad de sistemas con distintos tipos de propiedades. En el siguiente punto exploraremos las distintas aplicaciones del modelo de Ising, no solo en física, sino también en la resolución de problemas de optimización, como por ejemplo, el problema *Max-Cut*.

3.1.2. Aplicaciones.

En la sección anterior hemos introducido el problema propuesto por Lenz; el cuál motivó a Ising a desarrollar su modelo. Este modelo se ideó con el objetivo de explicar

las transiciones de fase y propiedades físicas como el ferromagnetismo en el contexto de la mecánica estadística. Pero como ya hemos anticipado, este modelo no solo es de gran utilidad para abordar este tipo de problemas, sino que es extrapolable a muchos otros dentro y fuera del ámbito de la física.

Algunas de las aplicaciones ajenas a la física en las cuales el modelo de Ising resulta una herramienta matemática de gran utilidad pueden ser:

- **Sistemas Biológicos:** En biología, el modelo de Ising se puede aplicar en numerosos contextos. Uno de ellos es estudiar cómo emergen patrones espaciales de expresión génica a partir de la interacción de redes génicas individuales mediante un modelo basado en Ising [15]. En este caso se usa un modelo de Ising en 2-D conocido como CPIM (*Contact Process Ising Model*).
- **Economía y Ciencias Sociales:** En torno a los años 90, emergió un nuevo campo de investigación conocido como econofísica, que establecía una conexión entre la física estadística y los modelos económicos y sociales [16]. En este contexto, el modelo de Ising se ha adaptado para estudiar fenómenos sociales y económicos como la difusión de opiniones, el comportamiento de los mercados financieros y la propagación de innovaciones.
- **Redes:** En el campo de la ciencia de redes, el modelo de Ising se usa para analizar la formación y el comportamiento de redes complejas, incluyendo la percolación y la dinámica en redes. Por ejemplo, el modelo de Ising se puede utilizar para analizar redes de co-autoría de publicaciones científicas. En [17] se evaluó si el modelo de Ising era capaz de describir dinámicas de una red de manera análoga a como describe sistemas magnéticos.
- **Ciencia de Materiales:** En materiales, el modelo de Ising ayuda a entender propiedades como la ordenación o el desorden en aleaciones y compuestos, permitiendo predecir cómo diferentes condiciones pueden afectar la estructura y propiedades de los materiales. Por ejemplo el modelo de Ising se puede utilizar para el diseño de compuestos utilizados para construir disipadores de calor [18] de tal forma que sus propiedades sean óptimas para dicho propósito.
- **Teoría de la Información:** El modelo de Ising también se aplica en la teoría de la información y la codificación, especialmente en el estudio de códigos de corrección

de errores y en la comprensión de cómo la información se transmite y se procesa en sistemas ruidosos. Concretamente existen códigos cuánticos de corrección de errores que se basan en modelos de Ising [19].

Podemos ver que el modelo de Ising nos proporciona una serie de herramientas de gran utilidad para la modelización de sistemas complejos. Este tipo de sistemas no están limitados a la física, siendo así de gran utilidad para múltiples sectores dentro de la industria o la ciencia. Si nos fijamos un poco en las distintas aplicaciones que hemos destacado, podemos encontrar ciertos puntos en común dentro de ellas.

- En todos los casos, existen fenómenos que queremos modelizar de tal forma que seamos capaces de optimizar una cierta propiedad del sistema.
- En todos los casos, podemos modelizar el problema en base a variables binarias, las cuales serían análogas a los distintos estados de spin de un modelo de Ising tradicional.

Estas características, hacen que el modelo de Ising sea de gran utilidad en el contexto de la resolución de problemas de optimización utilizando computación cuántica. Para la resolución de este tipo de problemas utilizando computación cuántica, se suele utilizar una formulación conocida como QUBO, la cual, como veremos en la siguiente sección, se encuentra íntimamente relacionada con el modelo de Ising.

3.1.3. Limitaciones

Pese a que el modelo de Ising es de gran utilidad en multitud de sectores y casos de uso, también cuenta con ciertas limitaciones. Una de ellas podría ser que el grado máximo de interacción entre variables es cuadrático, lo que podría suponer un modelo demasiado sencillo en ciertos casos. Más adelante veremos técnicas o alternativas a la hora de abordar este problema.

Otra posible limitación es que a la hora de definir nuestro Hamiltoniano y sus distintos estados de spin, caracterizamos los distintos estados de spin σ tomando una dirección arbitraria del espacio, es decir, en realidad σ se medirá en alguna de las direcciones del espacio $\{x, y, z\}$, por lo que en realidad nuestro Hamiltoniano se puede formular como

$$H = - \sum_{i,j \in \Delta} J_{i,j} \sigma_i^z \sigma_j^z - \mu \sum_{i \in \Delta} h_i \sigma_i^z \quad (3.3)$$

Esto podría presentar una limitación a la hora de modelizar problemas o Hamiltonianos que necesiten de varios operadores de Pauli $\sigma_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}$. Como veremos más adelante, esto se podrá solucionar de manera sencilla con puertas auxiliares.

3.2. Problemas tipo QUBO.

Para poder introducir de manera sencilla la formulación QUBO, vamos a comenzar describiendo el significado de sus siglas [20]:

- **Q**(Quadratic): Los problemas que podremos resolver en este marco son, a lo sumo, cuadráticos.
- **U**(Unconstrained): En los problemas que vamos a resolver no existen restricciones que tengamos que cumplir de manera obligatoria.
- **B**(Binary): Las variables que describen el problema son binarias.
- **O**(Optimization): Los problemas que vamos a resolver buscan encontrar la combinación de variables que minimicen una cierta función de coste.

La formulación QUBO nos permitirá modelizar problemas de optimización en los cuales no existan restricciones, las variables que lo formen sean binarias y las relaciones entre ellas sean a lo sumo cuadráticas.

Aunque puedan parecer un conjunto reducido de problemas, en esta sección definiremos este formalismo y discutiremos una serie de ejemplos que nos ayudarán a entender por qué estos problemas son de gran interés hoy en día.

3.2.1. Formalismo

Introducción al formalismo

De manera general, cualquier problema tipo QUBO [21] se puede expresar con una expresión como esta:

$$f_Q(x) = x^\top Q x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j \quad (3.4)$$

Donde $x \in \mathbb{B}^n$ con $\mathbb{B} = \{0, 1\}$, es un vector de n variables binarias x_i . Por otro lado, $Q \in \mathbb{R}^{n \times n}$ es una matriz triangular superior cuyas entradas Q_{ij} definen los coeficientes

asociados a los pares de índices i, j del vector de variables x . De esta forma, la función $f_Q(x)$ mapea el vector x a un número real que representa el valor de cierta función objetivo.

Podría parecer que la función general que hemos definido solo tiene términos cuadráticos $x_i x_j$, pero al ser variables binarias, si $i = j$ podemos reescribir $x_i x_i$ como x_i , por lo que cubrimos todas las posibilidades con la definición dada.

$$f_Q(x) = \sum_{i=1}^n \sum_{j \neq i}^n Q_{ij} x_i x_j + \sum_{i=1}^n Q_{ii} x_i \quad (3.5)$$

Por otro lado, dado que la formulación QUBO está pensada para resolver problemas de minimización, si buscamos maximizar el valor de una cierta función en lugar de minimizarla, simplemente debemos cambiar el signo de $f_Q(x)$ y la formulación QUBO seguirá siendo válida, ya que

$$\operatorname{argmax}(f_Q(x)) = \operatorname{argmin}(-f_Q(x)) \quad (3.6)$$

3.2.2. Como abordar limitaciones del formalismo

Una vez introducido la base del formalismo QUBO, vamos a describir como podemos añadir cierta complejidad a nuestro problema, con la inclusión de restricciones.

Inclusión de restricciones

Una de las características fundamentales de los problemas tipo QUBO es que el problema que queremos resolver no tiene restricciones explícitas. Esta característica puede parecer algo muy limitante, ya que en la mayoría de problemas de optimización reales, vamos a tener ciertas restricciones como, por ejemplo, la cantidad de recursos disponibles.

Para abordar esta limitación, la formulación QUBO permite introducir restricciones no obligatorias que penalizan el resultado de nuestra función a optimizar si estas no se cumplen [22]. De esta forma, el problema que estamos intentando resolver sigue sin tener restricciones que se deban cumplir de manera obligatoria, pero sí tiene ciertas condiciones que, de no cumplirse, penalizarían el resultado obtenido.

La forma matemática de introducir este tipo de restricciones es a través de multiplicadores de Lagrange. Por ejemplo, si tuviéramos una restricción de la forma $Ax = b$, podríamos introducirla a nuestra función de la siguiente forma:

$$f_Q(x) = \sum_{i=1}^n \sum_{j=i}^n Q_{ij} x_i x_j + \lambda \sum_{i=1}^n (A_{ii} x_i - b)^2 \quad (3.7)$$

Donde $\lambda \in \mathbb{R}^+$ es el coeficiente de Lagrange asociado a la restricción. Vemos que este término añade una cantidad positiva a la función siempre que no se cumpla la condición $Ax = b$, penalizando así la minimización de la función $f_Q(x)$.

Es importante resaltar que el valor del coeficiente de Lagrange es algo que definiremos nosotros, por lo que debemos tener en cuenta que

- Un coeficiente de Lagrange muy pequeño hará que no merezca la pena cumplir la restricción, pudiendo hacer que el resultado no sea válido.
- Un coeficiente de Lagrange muy elevado hará irrelevantes el resto de términos de la función de coste. Esto es debido a que los *Quantum Annealers*, computadores en los que se suelen implementar este tipo de problemas, normalizan el problema para poder implementarlo físicamente, por lo que un coeficiente de Lagrange muy elevado hará que el resto de términos sean despreciables a la hora de buscar la solución.

Por estos motivos, la elección de un valor adecuado es crucial para la obtención de resultados correctos.

Un caso un poco más complejo es el de la implementación de restricciones del tipo $Ax \leq b$, ya que si usáramos la misma formulación que en el caso anterior, estaríamos también penalizando los casos cuando $Ax < b$.

Para este tipo de restricciones necesitamos añadir una variable auxiliar y , que nos ayude a cumplir la igualdad en los casos en los que $Ax < b$.

$$f_Q(x, y) = \sum_{i=1}^n \sum_{j=i}^n Q_{ij} x_i x_j + \lambda \sum_{i=1}^n (A_{ii} x_i + y - b)^2 \quad (3.8)$$

Una vez hemos introducido esta variable auxiliar, debemos escribirla en función de variables binarias para que cumpla con el marco de los problemas QUBO. Este tipo de variables se suelen denominar variables *slack* y su inclusión implica que pasaremos de y a $k = \lceil \log(b) \rceil$ variables *slack* binarias que representan a y .

$$f_Q(x, y) = \sum_{i=1}^n \sum_{j=i}^n Q_{ij} x_i x_j + \lambda \sum_{i=1}^n (A_{ii} x_i + \sum_{j=1}^{k-1} 2^j s_j - b)^2 \quad (3.9)$$

Adaptación de problemas no cuadráticos

Por último, podríamos tener un caso en el que nuestro problema tuviera términos de grado mayor a dos [22][23]. Esto es relativamente sencillo que ocurra, ya que con tener una restricción de grado dos, al elevarla al cuadrado para que siempre sea positiva, ya aparecería este problema.

Para solventarlo podemos introducir cambios de variable. Si por ejemplo tuviéramos la función

$$f(x) = x_1x_2x_3x_4 \quad (3.10)$$

Podríamos hacer el cambio de variable

$$y_1 = x_1x_2 \quad (3.11)$$

$$y_2 = x_3x_4 \quad (3.12)$$

Y obtendríamos como resultado

$$f(y) = y_1y_2 \quad (3.13)$$

Esto parece algo muy sencillo en un primer momento, pero añade más complejidad de la que parece al problema. Al hacer este cambio de variable aparecen nuevas restricciones del tipo $x_1x_2 - y_1 = 0$, que vuelven a generar el problema de que al elevarlas al cuadrado no cumplen la condición de ser cuadráticas.

Una forma de abordar este problema es reescribiendo estas restricciones teniendo en cuenta que las variables con las que estamos trabajando son binarias. De esta forma si se cumple

$$y_1 \leq x_1 \quad (3.14)$$

$$y_1 \leq x_2 \quad (3.15)$$

Tenemos cubiertas todas las posibilidades excepto $x_1 = x_2 = y_1$. Para cubrir este hueco podemos añadir la condición

$$y_1 \geq x_1 + x_2 - 1 \quad (3.16)$$

Con esto, ya hemos completado el cambio de variables necesario para disminuir un grado en un término de nuestra función. Este proceso es algo bastante costoso, ya que para reducir un grado en un término necesitamos de 4 nuevas variables auxiliares, aumentando significativamente la complejidad de nuestro problema. Esto es algo especialmente problemático debido al punto en el que se encuentra la computación cuántica actualmente, ya que, en la era NISQ (*Noisy Intermediate-Scale Quantum*)¹, los recursos de hardware son un gran limitante, por lo que aumentar tanto la complejidad podría llevar a que no fuéramos capaces de implementar nuestro problema en un computador cuántico.

Otro enfoque para reducir el grado de nuestro problema, que en general es más utilizado, es el uso de funciones de penalización [24]. Este tipo de funciones tendrán el siguiente comportamiento

$$P(y_1 = x_1 x_2) = 0 \quad (3.17)$$

$$P(y_1 \neq x_1 x_2) > 0 \quad (3.18)$$

Una forma general de definir una función que tenga el comportamiento ilustrado y que sea cuadrática podría ser la siguiente:

$$P(y_1, x_1, x_2) = d_0 y_1 + d_1 x_1 + d_3 x_2 + d_4 y_1 x_1 + d_5 y_1 x_2 + d_5 x_1 x_2 + d_6 \quad (3.19)$$

Sujeta a ciertas restricciones que harán que su comportamiento sea el que buscamos. Comenzaremos por el comportamiento de la función en los casos en los que $y_1 = x_1 x_2$

- $P(0, 0, 0) = 0$ por lo que $d_6 = 0$
- $P(0, 0, 1) = 0$ por lo que $d_2 = 0$
- $P(0, 1, 0) = 0$ por lo que $d_1 = 0$
- $P(1, 1, 1) = 0$ por lo que $d_0 + d_3 + d_4 + d_5 = 0$

¹**NISQ** se refiere a la era de las computadoras cuánticas de escala intermedia y ruidosa. Esta fase es característica de los dispositivos cuánticos actuales que son suficientemente grandes para realizar cálculos que no pueden ser hechos eficientemente por computadoras clásicas en un tiempo razonable, pero que todavía sufren de un alto nivel de errores y ruido.

Además de estas restricciones cuando la función de penalización debe de tomar el valor cero, también debemos tener en cuenta el resto de casos, donde la función de penalización debe de devolver un valor positivo cuando se cumple $y_1 \neq x_1x_2$.

- $P(1, 0, 0) > 0$ por lo que $d_0 > 0$
- $P(1, 0, 1) > 0$ por lo que $d_0 + d_4 > 0$
- $P(1, 1, 0) > 0$ por lo que $d_0 + d_3 > 0$
- $P(0, 1, 1) > 0$ por lo que $d_1 + d_2 + d_5 > 0$

Cualquier combinación de parámetros d que cumpla las condiciones mencionadas, creará una función de penalización válida para implementar las restricciones necesarias para la reducción del grado de nuestro problema QUBO. Este enfoque, pese a ser más laborioso que el anterior, ya que requiere de calcular previamente la función de penalización, es más eficiente en cuanto a recursos y por ello, más utilizado.

Ante este problema, se han presentado dos posibles métodos para reformular el problema para que siga cumpliendo con la definición de QUBO, pero no es complicado darse cuenta que esto no siempre es viable. Para abordar esto, como veremos más adelante, existen algoritmos que serán capaces de trabajar con problemas que no sean QUBO, siempre y cuando se cumplan una serie de condiciones.

3.2.3. Limitaciones de la computación clásica.

A la hora de introducir las posibles implementaciones computacionales de este tipo de problemas no hemos hablado de su posible implementación en computadores clásicos. Esto se debe a que debido a la naturaleza del problema, este tipo de sistemas cuentan con muchas limitaciones a la hora de resolverlos eficazmente cuando el problema crece en complejidad.

Los algoritmos clásicos, como los métodos exactos (por ejemplo, el método *branch and bound*²) y los métodos heurísticos (por ejemplo, *simulated annealing*³ o algoritmos

²**Branch and Bound** es una técnica para resolver problemas complejos de optimización al dividir el problema en subproblemas más manejables, evaluar estos subproblemas y podar aquellos que no pueden mejorar la solución óptima conocida.

³**Simulated Annealing** es un algoritmo de optimización inspirado en el proceso físico de recocido en metalurgia. Se utiliza para encontrar una solución aproximada a problemas de optimización complejos donde el espacio de búsqueda es grande y no se puede explorar exhaustivamente.

genéticos⁴), enfrentan serias dificultades cuando se trata de problemas a gran escala.

Algunas de las limitaciones que podemos encontrar en las implementaciones clásicas son:

- **Complejidad computacional:** Muchos problemas tipo QUBO o Ising, son problemas *NP-hard*, lo que significa que no se conoce ningún algoritmo eficiente que pueda resolver todos los casos en tiempo polinomial. Los métodos exactos se vuelven impracticables para instancias grandes debido al tiempo exponencial requerido. Por otro lado, los métodos heurísticos y meta-heurísticos, aunque más rápidos, no garantizan encontrar la solución óptima y pueden estancarse en óptimos locales. Ante esto la computación cuántica aparece como una posible alternativa [25][26].
- **Limitaciones de Hardware:** Incluso con los avances en la computación clásica y la disponibilidad de supercomputadores, las limitaciones de hardware imponen restricciones significativas en la resolución de problemas grandes y complejos. La capacidad de almacenamiento y la velocidad de procesamiento siguen siendo barreras importantes para el análisis exhaustivo y la simulación de grandes sistemas.
- **Fenómenos Cuánticos:** Los sistemas cuánticos exhiben fenómenos como la superposición y el entrelazamiento, que no tienen equivalentes clásicos y pueden proporcionar ventajas significativas en la resolución de ciertos problemas. Los algoritmos clásicos no pueden aprovechar estos fenómenos, lo que puede limitar su eficacia en comparación con los algoritmos cuánticos que sí pueden explotarlos. Por otro lado, en simulaciones de sistemas físicos, los modelos clásicos pueden no representar adecuadamente estos efectos cuánticos, lo que limita la precisión y la validez de los resultados.

Eficiencia Energética: La computación clásica requiere una cantidad considerable de energía, especialmente cuando se trata de resolver problemas intensivos. Ante esto, la computación cuántica se postula como una alternativa más eficiente energéticamente [27] debido a la reversibilidad de sus procesos.

En resumen, aunque los métodos clásicos han avanzado significativamente y siguen siendo útiles en muchos contextos, presentan limitaciones importantes que motivan la exploración y el desarrollo de enfoques cuánticos para la resolución de problemas tipo QUBO

⁴Los **algoritmos genéticos** son una técnica de optimización basada en los principios de la evolución biológica y la genética. Se utilizan para encontrar soluciones aproximadas a problemas complejos de optimización y búsqueda

e Ising. La computación cuántica ofrece el potencial para superar algunas de estas limitaciones, proporcionando soluciones más eficientes y efectivas para problemas complejos y de gran escala.

3.2.4. Ejemplos de problemas tipo QUBO.

Una vez definido el marco de la formulación QUBO, vamos a ver ejemplos de problemas tipo QUBO que nos permitan ilustrar su interés [22].

- **Max-Cut:**

El problema *Max-Cut* es uno de los más famosos en optimización. Dado un grafo no dirigido $G(V, E)$ donde V son los vértices del grafo y E sus aristas, buscamos un subconjunto de los vértices $S \subseteq V$ tal que el número de aristas entre S y su complemento $V \setminus S$ sea máximo.

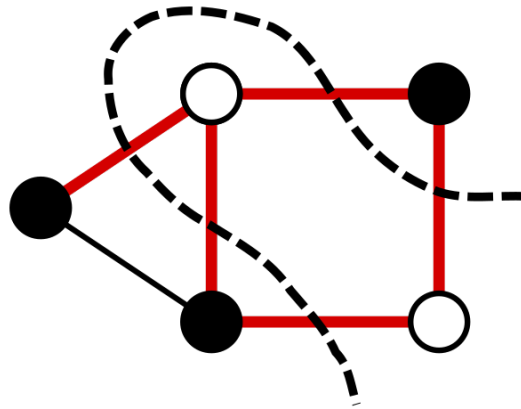


Figura 3.4: Ilustración del problema *Max-Cut*

En este problema las variables binarias que intentaremos optimizar serán los diferentes vértices $v_i \in V$ que tomarán los valores $\{0, 1\}$ según pertenezcan a uno u otro de los subgrafos.

- **El problema de la mochila**

El problema de la mochila consiste en optimizar el espacio disponible en nuestra mochila de tal forma que no nos dejemos nada importante y, además, aprovechemos al máximo su capacidad. En nuestro problema, contamos con la restricción de que la mochila tiene un volumen máximo V . Además, cada objeto que queremos llevar en

la mochila está caracterizado por su volumen v_i y su importancia w_i . Cada objeto es representado en nuestro problema por una variable binaria x_i , que será 0 si no llevamos el objeto en la mochila y 1 si sí lo llevamos.

El objetivo es maximizar la importancia total de los objetos en la mochila, respetando la restricción de volumen.

■ El problema del viajante

El problema del viajante, también conocido como problema TSP por sus siglas en inglés, es un problema más complejo que los anteriores. En este problema tenemos un vendedor que debe recorrer un cierto número de ciudades recorriendo la menor distancia posible. Para hacer esto cuenta con una serie de restricciones:

- Debe de empezar y terminar en la misma ciudad.
- Debe de visitar todas las ciudades exactamente una vez.
- Para poder ir de una ciudad a otra debe de existir una conexión entre ellas, ya que no todas las ciudades tienen porqué estar conectadas entre sí.

Este problema es un problema de optimización muy típico que podemos resolver haciendo uso de la formulación QUBO. En este caso las variables binarias que caracterizan nuestro problema son x_{ij} , que serán uno si la i -ésima ciudad se ha visitado en j -ésimo lugar y cero si no.

Además de los problemas ilustrados, existen infinidad de problemas donde QUBO puede resultar de utilidad. En este trabajo nos enfocaremos en el uso de la formulación QUBO y su estrecha relación con el modelo de Ising para sus aplicaciones en química cuántica. Los problemas que abordaremos buscan poder simular moléculas complejas con el fin conocer características del sistema como su geometría óptima, su configuración electrónica o la energía de su estado fundamental.

3.2.5. Conexión entre Ising y QUBO.

Como hemos introducido anteriormente, el modelo de Ising tiene numerosas aplicaciones más allá de la física. Estas aplicaciones habitualmente son las del uso de la estructura general del modelo de Ising, para modelizar otro tipo de sistemas y fenómenos.

En este contexto, aparecen problemas de optimización, como el *Max-Cut*, en los cuales podemos formular el problema a través del modelo de Ising para más tarde intentar resolverlo.

En este tipo de problemas buscamos elegir la combinación de variables óptima, de forma que una cierta función alcance un extremo. Para abordar la formulación y resolución de este tipo de problemas existen numerosos métodos y marcos de trabajo, pero en este caso nos centraremos en la formulación QUBO y su relación con el modelo de Ising.

Como hemos desarrollado a lo largo de esta sección, la formulación QUBO nos permitirá modelizar problemas de optimización en los cuales no existan restricciones, las variables que lo formen sean binarias y las relaciones entre ellas sean a lo sumo cuadráticas. De esta forma, es relativamente sencillo ver la relación existente entre el modelo de Ising y la formulación QUBO.

- Ambos se basan en variables binarias, en el caso de Ising, los spins son dichas variables.
- Ambos buscan optimizar un sistema o fenómeno.
- En Ising, el grado más alto de interacción es cuadrático, al igual que en QUBO.

Además, para realizar la transformación de un problema QUBO con variables $x_i \in \{0, 1\}$ a uno Ising con spins $s_i \in \{\pm 1\}$, simplemente debemos aplicar el cambio de variable [28]:

$$x_i = \frac{1 + s_i}{2} \quad (3.20)$$

Por estos motivos, es muy habitual usar la formulación QUBO para definir el problema que buscamos resolver, y en los casos que sea necesario, transformar el problema QUBO en un problema Ising o viceversa.

3.3. *Quantum Annealing* para problemas Ising-QUBO

A la hora de abordar problemas de optimización combinatoria que tengan la estructura de un problema Ising-QUBO, uno de los métodos más comunes de abordarlo es a través de *Quantum Annealing*. Este algoritmo cuántico es tan popular a la hora de abordar este tipo de problemas por dos motivos:

- El hardware necesario para implementar este algoritmo son los *Quantum Annealers*. Estos ordenadores pese a no ser de propósito general, son capaces de contar con más qubits que los computadores cuánticos universales basados en puertas, por lo que son capaces de abordar problemas mayores.
- Este hardware tiene mayor disponibilidad que los computadores cuánticos universales basados en puertas, por lo que su uso no es tan restringido.

Es por esto, que siempre que sea posible, buscaremos que nuestro problema siga un formato Ising-QUBO. En esta sección detallaremos el funcionamiento del algoritmo *Quantum Annealing* y su utilidad a la hora de resolver problemas Ising-QUBO.

Quantum Annealing es un algoritmo cuántico que basa su funcionamiento en el teorema adiabático [7]. Este teorema afirma que un sistema físico en un estado propio del sistema, permanece en un estado propio si el sistema se perturba de una forma lo suficientemente lenta y el *gap* entre la energía del auto-estado y el siguiente auto-estado es lo suficientemente grande.

$$\left| \left\langle m(t) \left| \frac{d}{dt} H(t) \right| n(t) \right\rangle \right| \ll \frac{|E_n(t) - E_m(t)|}{\Delta t_{nm}} \quad (3.21)$$

Donde $|n(t)\rangle$ y $|m(t)\rangle$ son estados del sistema y Δt_{nm} el tiempo de la transición entre estados.

Este principio, permitirá que si inicializamos un sistema con un Hamiltoniano y un estado fundamental conocido, podremos evolucionarlo a un Hamiltoniano más complejo con un estado fundamental desconocido si la transición entre ambos Hamiltonianos se hace de forma suficientemente lenta o suave. Este es el fundamento en el que se basa el algoritmo *Quantum Annealing*.

Este algoritmo, implementará físicamente el problema que busquemos resolver en el computador cuántico en el cual se ejecuta. Los pasos del algoritmo se pueden resumir en [29]:

1. Definición del Hamiltoniano inicial y su estado fundamental

El primer paso será generar un Hamiltoniano sencillo H_t el cual tenga un estado fundamental conocido y fácil de implementar en un computador cuántico. Este Ha-

Hamiltoniano se conoce como Hamiltoniano de *tunneling* y se define como

$$H_t = \sum_i^n \sigma_x^i \quad (3.22)$$

Cuyo estado fundamental es el estado $|\psi\rangle = |+\rangle^{\otimes n}$.

Este Hamiltoniano de *tunneling*, además de permitirnos la inicialización sencilla de un auto-estado del sistema, también permitirá al estado escapar de posibles mínimos locales gracias al conocido *tunneling* cuántico.

2. Definición del Hamiltoniano del problema:

Una vez definido nuestro término de *tunneling*, definiremos el Hamiltoniano de Ising que caracteriza nuestro problema.

$$H_i = - \sum_{i,j \in \Delta} J_{i,j} \sigma_z^i \sigma_z^j - \mu \sum_{i \in \Delta} h_i \sigma_z^i \quad (3.23)$$

Este será el Hamiltoniano del cual buscamos encontrar el estado fundamental haciendo uso del teorema adiabático.

3. Definición del Hamiltoniano final

El Hamiltoniano que implementaremos finalmente en nuestro computador cuántico será de la forma

$$H(s) = A(s)H_t + B(s)H_i \quad (3.24)$$

Donde s empieza en 0 y termina en 1 cuando el proceso de *annealing* termina. Los parámetros de *annealing* A, B cumplen que $A(s) \gg B(s)$ cuando $s = 0$ y $A(s) \ll B(s)$ cuando $s = 1$ de tal forma que el Hamiltoniano evolucione lentamente de H_t a H_i (Fig.3.5).

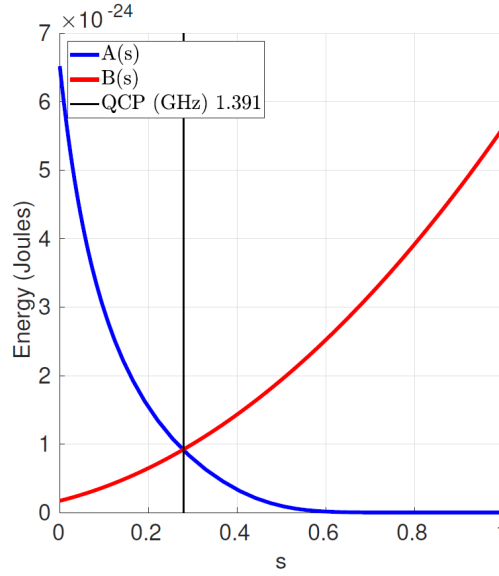


Figura 3.5: Evolución de los parámetros A y B con el paso del tiempo en el algoritmo *Quantum Annealing*. Fuente: <https://docs.dwavesys.com/>

4. Ejecución del proceso de *Annealing*

Cumplir las condiciones del teorema adiabático es algo muy complicado. A la hora de implementar este algoritmo en computadores reales, definiremos un número de ejecuciones k . Dejaremos evolucionar el sistema bajo el algoritmo definido k veces y evaluaremos los resultados obtenidos, pudiendo así mitigar errores debidos al ruido inherente a la ejecución del proceso.

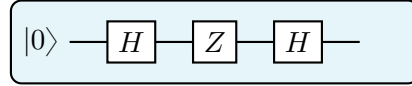
De esta forma, *Quantum Annealing* nos permite conocer el auto-estado del Hamiltoniano de nuestro problema Ising-QUBO, partiendo de un estado conocido y evolucionando dicho estado de forma que nos encontremos bajo las condiciones del teorema adiabático.

3.4. Más allá de Ising-QUBO

Como ya hemos introducido en las secciones anteriores, el marco Ising-QUBO es uno de los más utilizados a la hora de resolver problemas de optimización combinatoria utilizando computadores cuánticos, tanto de propósito general como *annealers*. También hemos podido ver que este modelo, pese a ser de gran utilidad y abordar una gran cantidad de problemas, en ocasiones, puede no ser capaz de modelizar nuestro problema, o que las reformulaciones necesarias sean demasiado costosas.

Para solucionar este problema existen algoritmos cuánticos capaces de resolver hamiltonianos más generales que los presentados por el marco Ising-QUBO. Estos hamiltonianos cuentan con muchas características en común con los descritos anteriormente, y es por esto que el marco Ising-QUBO es tan importante.

Una de las limitaciones del Hamiltoniano de Ising es que está formulado en base al operador de Pauli en una dirección arbitraria del espacio, por ejemplo σ_z . Pero en muchas ocasiones, a la hora de representar nuestro Hamiltoniano en base a puertas cuánticas, pueden aparecer otros operadores de Pauli $\sigma_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}$. Existen rutinas que nos permitirán solucionar esto [5]. Por ejemplo $\sigma_x = H\sigma_zH$, se puede implementar de manera muy sencilla con el siguiente circuito cuántico



Se puede definir una transformación análoga para $\sigma_y = H_y\sigma_zH_y$ donde

$$H_y = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ i & -1 \end{pmatrix} \quad (3.25)$$

De esta forma, podemos definir hamiltonianos de Ising más generales de forma sencilla, únicamente utilizando puertas auxiliares.

Otro punto en el que hemos visto que este tipo de formulación puede no ser la más adecuada, es cuando existen relaciones entre variables que no son cuadráticas. En secciones anteriores, hemos introducido herramientas para conseguir reformular el problema de tal forma que mantenga la condición de cuadrático, pero esto no es siempre algo sencillo.

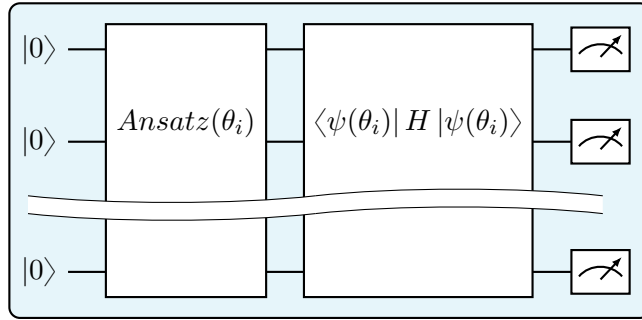
Por este motivo surge la necesidad de poder resolver problemas de optimización basados en variables binarias y operadores que podamos implementar de manera sencilla en computadores cuánticos pero con relaciones más complejas entre las variables. Como más adelante veremos, este es el caso del Hamiltoniano de un sistema molecular.

3.4.1. *Variational Quantum Eigensolver (VQE)*

Ante esto, algoritmos como VQE [30], son una gran opción, ya que pueden trabajar con hamiltonianos que presenten este tipo de relaciones entre variables de manera sencilla. Ya que el único requerimiento de VQE es poder estimar el valor esperado del Hamiltoniano de forma eficiente.

En este punto, debemos introducir el término *ansatz* (del alemán, que significa "suposición" o "enfoque"). VQE se encuentra dentro de la familia de los algoritmos variacionales. Este tipo de algoritmos crea un estado cuántico inicial parametrizado, por ejemplo, mediante rotaciones $\{R_x, R_y, R_z\}$ parametrizadas por ángulos θ_i . Estos parámetros se van ajustando iterativamente con el objetivo de encontrar el estado cuántico que minimice la energía de un cierto Hamiltoniano. El operador cuántico que genera este estado parametrizado, y que será evaluado posteriormente, es conocido como *ansatz*.

De esta forma el objetivo de nuestro algoritmo será optimizar los parámetros del *ansatz* de tal forma que el valor esperado del Hamiltoniano del problema sea mínimo. Para ello utilizamos un circuito cuántico como el siguiente



Donde podremos evaluar el resultado que obtenemos para los valores θ_i . De esta forma podemos optimizar clásicamente dichos parámetros e ir evaluando iterativamente el valor esperado de nuestro Hamiltoniano.

De una forma más compacta, podemos sintetizar el funcionamiento del algoritmo VQE de la siguiente forma:

1. **Inicialización del Ansatz:** Se elige un *ansatz*, es decir, una forma parametrizada del estado cuántico inicial $|\psi(\theta)\rangle$. Este estado es generado por una serie de puertas cuánticas que dependen de un conjunto de parámetros $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$.
2. **Preparación del Estado Cuántico:** Se prepara el estado cuántico $|\psi(\theta)\rangle$ en el circuito cuántico utilizando las puertas cuánticas definidas por el *ansatz* y los valores actuales de los parámetros θ . Estos parámetros se pueden inicializar de manera aleatoria y se irán actualizando según avance el proceso de optimización.
3. **Evaluación del Hamiltoniano:** Se mide el valor esperado del Hamiltoniano H en el estado $|\psi(\theta)\rangle$, es decir, se calcula $\langle\psi(\theta)|H|\psi(\theta)\rangle$. Existen numerosos métodos para obtener este valor, desde algoritmos como EVA [31] hasta la descomposición

del Hamiltoniano en operadores de Pauli para poder evaluarlo de forma sencilla utilizando puertas.

4. **Actualización de Parámetros:** Utilizando un algoritmo clásico de optimización (por ejemplo, descenso por gradiente, COBYLA, Nelder-Mead, etc.), se actualizan los parámetros θ para minimizar la energía $\langle \psi(\theta) | H | \psi(\theta) \rangle$.
5. **Repetición de Pasos 2 a 4:** Los pasos de preparación del estado, evaluación del Hamiltoniano y actualización de parámetros se repiten iterativamente hasta que el proceso de optimización converja, es decir, hasta que la energía medida no disminuya significativamente con nuevas iteraciones.
6. **Obtención del Resultado Final:** Una vez que se alcanza la convergencia, el estado cuántico $|\psi(\theta^*)\rangle$ que minimiza la energía se considera una aproximación al estado fundamental del Hamiltoniano H . La energía asociada a este estado es la aproximación de la energía mínima.

La definición de un *ansatz* que sea capaz de proporcionar buenos resultados no siempre es sencillo. Es habitual crear un *ansatz* que contengan entrelazamiento y puertas de rotación parametrizadas, de forma que el estado que puedan generar sea lo suficientemente general como para obtener soluciones adecuadas.

Como hemos podido ver, el marco Ising-QUBO presenta una base para poder comprender y formular problemas de optimización combinatoria de tal forma que podamos implementarlos de manera sencilla en un computador cuántico, pero cuenta con ciertas limitaciones en su definición. Ante esto, métodos como la definición de otros operadores de Pauli como función del operador de referencia σ_z o el uso de algoritmos que no necesiten que el problema sea cuadrático, surgen como una alternativa muy interesante a la hora de abordar problemas más complejos como el de la simulación de sistemas moleculares.

3.5. Computación Cuántica: Una Guía

La computación cuántica es un paradigma revolucionario que aprovecha los principios de la mecánica cuántica para realizar tareas de cálculo que superan las capacidades de los ordenadores clásicos. Los ordenadores cuánticos todavía están en desarrollo, por tanto el estado del arte en computación cuántica es sobre todo experimental[32].

En esencia, la computación cuántica se basa en *qubits*, los análogos cuánticos de los bits clásicos, que pueden existir en superposiciones de estados[33] y presentar entrelazamiento[34]. Estos fenómenos cuánticos permiten a los ordenadores cuánticos explorar simultáneamente amplios espacios de soluciones y resolver determinados problemas exponencialmente más rápido que los ordenadores clásicos.

En el apéndice A.2 revisamos los momentos relevantes en el desarrollo teórico y experimental que han devenido en lo que hoy conocemos como computación cuántica.

3.5.1. Computación Cuántica y su Relación con las Ciencias

La computación cuántica es un campo interdisciplinar con aportaciones de diversas áreas de la ciencia:

1. Física

- **Mecánica cuántica:** Proporciona los principios fundamentales como la superposición y el entrelazamiento, esenciales para los qubits.
- **Física de la materia condensada:** Contribuye con descubrimientos sobre superconductividad y el efecto Hall cuántico[35], vitales para qubits y circuitos cuánticos.
- **Teoría cuántica de campos:** Ayuda a modelar interacciones en sistemas cuánticos, crucial para la corrección de errores cuánticos.

2. Informática

- **Teoría de la información:** Extiende conceptos clásicos como la entropía a sistemas cuánticos, desarrollando códigos de corrección de errores cuánticos.
- **Algoritmos y teoría de la complejidad:** Desarrolla algoritmos cuánticos eficientes como el de Shor y Grover.
- **Criptografía:** Explora tanto amenazas de los ordenadores cuánticos a protocolos clásicos como el desarrollo de criptografía cuántica segura.

3. Matemáticas

- **Álgebra lineal:** Es fundamental para describir estados y operaciones cuánticas.
- **Probabilidad y estadística:** Es clave para interpretar los resultados de las mediciones cuánticas.

- **Teoría de grupos:** Ayuda a entender simetrías y la estructura matemática de algoritmos cuánticos.

4. Ingeniería Eléctrica y Electrónica

- **Desarrollo de hardware cuántico:** Construcción de qubits y puertas cuánticas, enfrentando desafíos como el ruido y la decoherencia.
- **Teoría de control:** Permite manipular qubits y mantener coherencia mediante técnicas precisas.

5. Química y Ciencia de los Materiales

- **Química cuántica:** Simulación eficiente de sistemas cuánticos, útil para entender estructuras moleculares y desarrollar nuevos materiales y fármacos.
- **Ciencia de los materiales:** Desarrollo de materiales adecuados para qubits y puertas cuánticas.

Cada disciplina contribuye al desarrollo de la tecnología cuántica, ampliando sus posibilidades mediante la integración de conocimientos y técnicas, en el apéndice [A.1](#) revisamos este tópico en más detalle.

3.5.2. Qubits: Principios Fundamentales y Conceptos

Los qubits[36], o bits cuánticos, son sistemas cuánticos que sirven como las unidades fundamentales de la información cuántica, análogas a los bits clásicos de la informática convencional. Sin embargo, a diferencia de los bits clásicos, que sólo pueden existir en uno de dos estados (0 o 1) en un momento dado, los qubits funcionan según los principios de la mecánica cuántica y pueden existir en una superposición de ambos estados simultáneamente. Esta propiedad permite a los qubits codificar y procesar información de un modo fundamentalmente distinto al de los bits clásicos, lo que posibilita una computación y comunicación cuánticas exponencialmente más potentes.

A continuación tenemos una descripción más detallada desde los enfoques de la Física, de las Matemáticas y de la Teoría de la Información:

Estados Cuánticos

- **Física:** Un estado cuántico describe el estado completo de un sistema cuántico, típicamente representado utilizando la notación de Dirac⁵[37] como $|\psi\rangle$. Para un solo qubit, este puede estar en un estado representado como una combinación lineal de estados base $|0\rangle$ y $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.26)$$

Aquí, α y β son números complejos llamados amplitudes de probabilidad.

- **Matemáticas:** Matemáticamente, un estado cuántico es un vector en un espacio de Hilbert⁶[38]. Para un qubit, este espacio es \mathbb{C}^2 , y la condición de normalización $|\alpha|^2 + |\beta|^2 = 1$ asegura que la probabilidad total es 1.
- **Teoría de la información:** En la teoría de la información, un estado cuántico representa la unidad fundamental de la información cuántica[36]. El vector de estado encapsula tanto las probabilidades clásicas como las fases cuánticas, lo que permite codificar más información que en un bit clásico.

Superposición cuántica

- **Física:** La superposición es el principio según el cual un sistema cuántico puede existir en varios estados simultáneamente. Un qubit puede estar en una superposición de $|0\rangle$ y $|1\rangle$.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.27)$$

- **Matemáticas:** Esto se representa como una combinación lineal de estados base, donde los coeficientes α y β son números complejos que determinan las probabilidades de medir el qubit en cualquiera de los estados.
- **Teoría de la información:** La superposición permite a un qubit transportar más información que un bit clásico al codificar un continuo de estados posibles. Este

⁵La notación de Dirac (notación bra-ket), es una forma estandarizada de describir estados cuánticos y operaciones en mecánica cuántica y computación cuántica. Es útil para manejar el álgebra lineal que subyace en la mecánica cuántica. En el apéndice A.3 se describe en más detalle los conceptos principales.

⁶Un espacio de Hilbert complejo es un espacio vectorial sobre los números complejos con un producto interno que cumple ciertas propiedades (linealidad, conjugado hermitiano y positividad). Además, es completo en el sentido de que cualquier secuencia de Cauchy (secuencias donde los elementos se acercan entre sí a medida que avanzan) de vectores tiene un límite dentro del espacio. Estos espacios son fundamentales en matemáticas y física, especialmente en análisis funcional y teoría cuántica. Para más información ver el apéndice A.4

paralelismo permite a los ordenadores cuánticos procesar muchas posibilidades simultáneamente.

Entrelazamiento cuántico

- **Física:** El entrelazamiento es una correlación entre estados cuánticos de múltiples partículas[39]. Por ejemplo, dos qubits pueden estar entrelazados en un estado como:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (3.28)$$

- **Matemáticas:** Matemáticamente, el entrelazamiento significa que el estado de cada qubit no puede describirse de forma independiente; se describen mediante un único vector en un espacio de Hilbert de dimensión superior.
- **Teoría de la información:** El entrelazamiento representa un recurso para la comunicación y la computación cuánticas, permitiendo tareas como el teletransporte y la codificación superdensa, que son imposibles con las correlaciones clásicas.

Medición Cuántica

- **Física:** La medición colapsa un estado cuántico a uno de los estados base[40]. Para un qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.29)$$

midiéndolo en la base $|0\rangle, |1\rangle$ da:

- $|0\rangle$ con probabilidad $|\alpha|^2$
- $|1\rangle$ con probabilidad $|\beta|^2$
- **Matemáticas:** Matemáticamente, la medición proyecta el vector de estado sobre uno de los vectores base, con la probabilidad determinada por el módulo cuadrado de la amplitud correspondiente.
- **Teoría de la información:** La medición extrae información clásica de un estado cuántico, reduciendo la superposición a un único resultado clásico. El proceso de medición cambia fundamentalmente el estado, reduciéndolo al resultado observado.

Interferencia cuántica

- **Física:** La interferencia se produce cuando las amplitudes de probabilidad de diferentes caminos se combinan, afectando a la probabilidad de los resultados[41]. La interferencia constructiva aumenta la probabilidad, mientras que la destructiva la disminuye.
- **Matemáticas:** Si un sistema evoluciona a través de caminos con amplitudes α y β :

$$\Psi_{\text{total}} = \alpha + \beta \quad (3.30)$$

La probabilidad del estado final es

$$P = |\alpha + \beta|^2 \quad (3.31)$$

- **Teoría de la información:** La interferencia se utiliza en algoritmos cuánticos para mejorar los resultados deseados y suprimir los no deseados mediante la manipulación de las fases de las amplitudes de probabilidad.

Realización física

Los qubits pueden construirse utilizando varios sistemas cuánticos, cada uno de los cuales explota diferentes propiedades físicas para la codificación y manipulación de la información cuántica. Algunos ejemplos de implementaciones físicas de qubits podemos revisarlos en el apéndice A.6. Cada tipo de qubit físico tiene sus propias ventajas y desafíos, y los investigadores están explorando activamente diversos enfoques para realizar sistemas de computación cuántica escalables y tolerantes a fallos.

3.5.3. Distinciones Entre Qubits en Ordenadores Cuánticos de Annealing y Basados en Puertas

Qubits de Quantum Annealing

Perspectiva Física

- **Evolución Adiabática:** El quantum annealing se basa en el teorema adiabático, según el cual el sistema se inicializa en el estado básico de un Hamiltoniano simple y

evoluciona lentamente hacia un Hamiltoniano complejo que representa el problema de optimización[29].

Perspectiva Matemática

- **Modelo de Ising:** El problema se representa a menudo como un modelo de Ising, donde los qubits corresponden a spins, y el objetivo es encontrar la configuración que minimiza el Hamiltoniano.

Perspectiva de la Teoría de la Información

- **Optimización:** El quantum annealing es especialmente adecuado para resolver problemas de optimización encontrando estados de baja energía que representan soluciones óptimas.

Qubits de Computación Cuántica Basada en Puertas

Perspectiva Física

- **Operaciones Discretas:** La computación cuántica basada en puertas implica la aplicación de puertas cuánticas discretas en una secuencia específica para manipular los estados de los qubits[33].

Perspectiva Matemática

- **Circuitos Cuánticos:** Los algoritmos se representan como circuitos cuánticos compuestos por una serie de puertas cuánticas.

Perspectiva de la Teoría de la Información

- **Flexibilidad de los Algoritmos:** Los sistemas basados en puertas son versátiles, capaces de implementar una amplia gama de algoritmos cuánticos más allá de la optimización, incluyendo la factorización, la búsqueda y la simulación.

3.5.4. Clasificaciones y Propiedades de los Qubits

La computación cuántica se basa en la manipulación de qubits, las unidades fundamentales de información en los sistemas cuánticos. Existen diversas formas de implementar físicamente los qubits, cada una con sus propias ventajas, desafíos y aplicaciones potenciales.

A continuación presentamos describimos algunas de las implementaciones más prometedoras, en el apéndice [A.6](#) se proporciona información más detallada incluyendo cronología y estado actual de los diferentes proyectos de implementaciones físicas de qubits.

Iones Atrapados

- **Trampas de Iones:** Utilizan campos electromagnéticos para atrapar y manipular iones, que sirven como qubits[42].
- **Precisión:** Conocidos por su alta precisión y sus largos tiempos de coherencia.

Qubits Fotónicos

- **Fotones:** Utilizan la polarización o la trayectoria de los fotones para codificar los estados de los qubits[43].
- **Velocidad:** Aprovechar la velocidad de la luz para el procesamiento rápido de la información y la comunicación.

Qubits de Spin

- **Spins de Electrones:** Utilizar los estados de spin de los electrones en puntos cuánticos o impurezas en sólidos[44].
- **Escalabilidad:** Ofrecen potencial de escalabilidad en dispositivos de estado sólido.

Propiedades Matemáticas

Norma del Vector de Estado

- **Normalización:** El vector de estado de un qubit debe ser normalizado, asegurando que la probabilidad total es 1.

Evolución Unitaria

- **Reversibilidad:** Las puertas cuánticas están representadas por matrices unitarias, asegurando la evolución reversible del estado del qubit.

Medidas de Entrelazamiento

- **Entropía:** Medidas como la entropía de entrelazamiento cuantifican el grado de entrelazamiento entre qubits.

3.5.5. Compuertas Lógicas Cuánticas.

Las puertas lógicas cuánticas son los componentes básicos de los circuitos cuánticos, análogos a las puertas lógicas clásicas de los circuitos digitales convencionales. Las puertas cuánticas manipulan el estado cuántico de los qubits, transformando los estados de entrada en estados de salida según operaciones cuánticas específicas.

Conceptos Generales

Perspectiva Física

- **Manipulación de Qubits:** Las puertas cuánticas manipulan qubits y son análogas a las puertas lógicas clásicas, pero funcionan según los principios de superposición y entrelazamiento.

Perspectiva Matemática

- **Matrices Unitarias:** Las puertas cuánticas se representan mediante matrices unitarias que transforman los vectores de estado.

Perspectiva de la Teoría de la Información

- **Construcción de Circuitos:** Las puertas cuánticas forman los bloques de construcción de los circuitos cuánticos, permitiendo la construcción y ejecución de algoritmos cuánticos. Manipulan y procesan información cuántica, aprovechando la superposición, el entrelazamiento y la interferencia para resolver problemas de forma más eficiente que los algoritmos clásicos.

Tipos de Puertas

Los tipos de puertas cuánticas se pueden clasificar de varias maneras, pero una de las más comunes es por el número de qubits sobre los que actúan, en la tabla a continuación veremos las compuertas básicas y en el apéndice [A.4](#) se puede ver en detalle las puertas de un solo qubit, las de dos qubits y las de múltiples qubits.

Tabla 3.1: Compuertas Cuánticas Básicas

Puerta	Representación Matemática	Función y Descripción
Puerta Pauli-X (Puerta X) (Puerta NOT)	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Invierte el estado de un qubit, análogo a la puerta clásica NOT. $X 0\rangle = 1\rangle$ $X 1\rangle = 0\rangle$
Pauli-Y (Puerta Y)	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Rota el estado alrededor del eje Y e introduce un factor de fase. $Y 0\rangle = i 1\rangle$ $Y 1\rangle = -i 0\rangle$
Puerta Pauli-Z (Puerta Z)	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Rota el estado alrededor del eje Z, introduciendo un factor de fase. $Z 0\rangle = 0\rangle$ $Z 1\rangle = - 1\rangle$
Puerta Hadamard (Puerta H)	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Crea superposición al rotar el estado del qubit. $H 0\rangle = \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$ $H 1\rangle = \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$
Puerta de Fase (Puerta S)	$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	Introduce un desplazamiento de fase de $\pi/2$ (rotación de 90 grados) alrededor del eje Z. $S 0\rangle = 0\rangle$ $S 1\rangle = i 1\rangle$
Puerta $\pi/8$ (Puerta T)	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	Introduce un desplazamiento de fase de $\pi/4$ (rotación de 45 grados) alrededor del eje Z. $T 0\rangle = 0\rangle$ $T 1\rangle = e^{i\pi/4} 1\rangle$

Los circuitos cuánticos se componen de secuencias de estas puertas, lo que permite transformaciones complejas de los estados de los qubits y la ejecución de algoritmos cuánticos. Profundicemos en el tema de las puertas cuánticas, explorando sus definiciones, representaciones matemáticas, propiedades y funciones dentro de los circuitos cuánticos.

Propiedades de las Puertas Cuánticas

Las puertas cuánticas deben satisfacer las siguientes propiedades[5]:

- **Unitaria:** Una puerta cuántica U debe ser unitaria, es decir $UU^\dagger = I$, donde U es la matriz unitaria (transposición conjugada) de U y I es la matriz identidad.
- **Reversibilidad:** Dado que las operaciones cuánticas son unitarias, son intrínsecamente reversibles, a diferencia de las puertas lógicas clásicas.
- **Transformación Lineal:** Las puertas cuánticas realizan transformaciones lineales en los estados de los qubits, preservando las propiedades de superposición y entrelazamiento.

Composición y Descomposición de Puertas Cuánticas

- **Composición:** Las puertas cuánticas pueden componerse para formar operaciones cuánticas más complejas. Por ejemplo, una secuencia de puertas puede aplicarse a un qubit o a un conjunto de qubits.
- **Descomposición:** Las puertas multi-qubit complejas a menudo pueden descomponerse en puertas de un qubit y dos qubits más simples. Por ejemplo, la puerta de Toffoli puede descomponerse en una combinación de puertas CNOT, Hadamard, T y T^\dagger .

Conjuntos de puertas universales

Un conjunto de puertas se considera universal si cualquier operación cuántica puede aproximarse con una precisión arbitraria utilizando sólo esas puertas[5]. Los conjuntos de puertas universales comunes incluyen:

- **H, S, T, CNOT:** La combinación de las puertas Hadamard, Phase, T y CNOT forma un conjunto de puertas universales.

- **H, T, CNOT:** Este conjunto también es universal ya que S puede descomponerse en puertas T.

Implementación de puertas cuánticas

La implementación física de puertas cuánticas implica la manipulación de sistemas cuánticos como iones atrapados, circuitos superconductores o fotones. Cada plataforma tiene sus métodos para realizar puertas:

- **Iones atrapados:** Las puertas se implementan utilizando pulsos láser para controlar los estados energéticos de los iones[42].
- **Circuitos superconductores:** Las puertas se implementan utilizando pulsos de microondas para controlar los estados de los qubits superconductores[45].
- **Sistemas fotónicos:** Las puertas se implementan utilizando elementos ópticos lineales e interacciones no lineales entre fotones[43].

Errores de Puerta y Corrección Cuántica de Errores

- **Errores de puerta:** Las implementaciones prácticas de puertas cuánticas son susceptibles a errores debidos a decoherencia, ruido e imperfecciones en el control. Estos errores pueden modelarse como errores de despolarización, de desfase o de amortiguación de amplitud[5].
- **Corrección de errores cuánticos:** Para mitigar los errores, se emplean códigos de corrección de errores cuánticos (QECC) como el código Shor, el código Steane y el código de superficie[46][47][48]. Estos códigos protegen la información cuántica codificándola en un espacio de Hilbert más amplio y detectando/corrigiendo los errores mediante mediciones de síndrome y operaciones de corrección.

Comprender las puertas cuánticas es crucial para diseñar e implementar circuitos cuánticos. Las puertas de un-qubit y multi-qubit constituyen las operaciones fundamentales, con conjuntos de puertas universales que permiten la construcción de cualquier algoritmo cuántico. A medida que avance la tecnología cuántica, la precisión y fidelidad de las operaciones de puerta seguirán mejorando, convirtiendo la computación cuántica en una poderosa herramienta para resolver problemas complejos.

3.5.6. Circuitos Cuánticos

Ahora profundizaremos en los circuitos cuánticos, explorando su estructura, componentes y aplicaciones[33]. Cubriremos en detalle los fundamentos, los elementos del circuito, los ejemplos clave y las consideraciones prácticas.

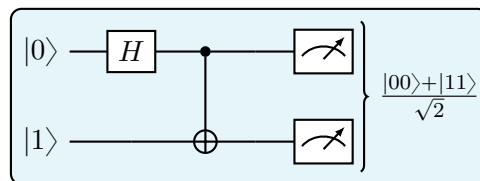
Fundamentos de los Circuitos Cuánticos

Un circuito cuántico es una secuencia de puertas cuánticas aplicadas a un conjunto de qubits. El circuito es una representación visual y matemática de un algoritmo cuántico o de un cálculo cuántico. Consta de:

- **Qubits:** Representados por líneas horizontales o "cables" conectores.
- **Puertas Cuánticas:** Representadas por símbolos colocados en los hilos.
- **Mediciones:** Indicadas por contadores o bits clásicos para representar el resultado de las mediciones de los qubits.

Diagrama de Circuito Cuántico

Un diagrama de circuito cuántico típico consiste en varios qubits inicializados a $|0\rangle$ o $|1\rangle$, seguidos de la aplicación de puertas y medidas cuánticas. He aquí un ejemplo sencillo:



En este ejemplo:

- El primer qubit se inicializa a $|0\rangle$ y luego se somete a una puerta Hadamard (H), creando una superposición.
- A continuación, se aplica una puerta CNOT con el primer qubit como control y el segundo qubit como objetivo.
- Finalmente, se miden ambos qubits.

Elementos de Circuitos Cuánticos

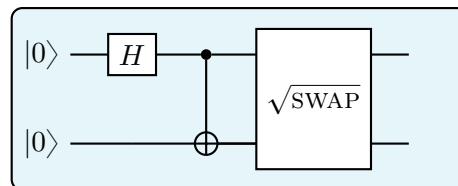
1. Cables Conectores (Qubits)

- **Cables Qubit:** Representan qubits y la evolución de su estado en el tiempo.
- **Cables Clásicos:** Representan bits clásicos utilizados para resultados de medidas y control clásico.

2. Puertas cuánticas

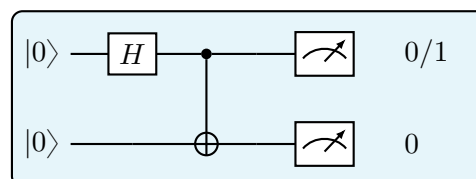
Las puertas cuánticas son operaciones aplicadas a los qubits, como se ha comentado en la sección anterior (ver información detallada para cada compuerta relevante en apéndice A.4). Aquí veremos cómo encajan en los circuitos.

- **Puertas de un solo qubit:** Compuertas como X, Y, Z, H, S y T operan sobre qubits individuales.
- **Compuertas Multi-Qubit:** Compuertas como CNOT, SWAP y Toffoli operan en múltiples qubits.
- **Ejemplo:**



3. Mediciones

Las mediciones colapsan el estado del qubit en un bit clásico, $|0\rangle$ o $|1\rangle$.

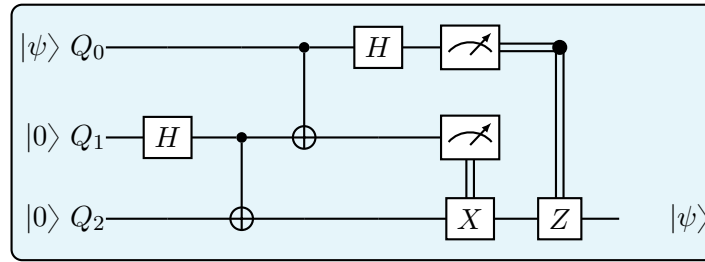


4. Circuitos claves de ejemplo

Teletransporte Cuántico

El teletransporte cuántico es un protocolo para transmitir un estado qubit de un lugar a otro utilizando el entrelazamiento y la comunicación clásica[49].

■ **Circuito:**



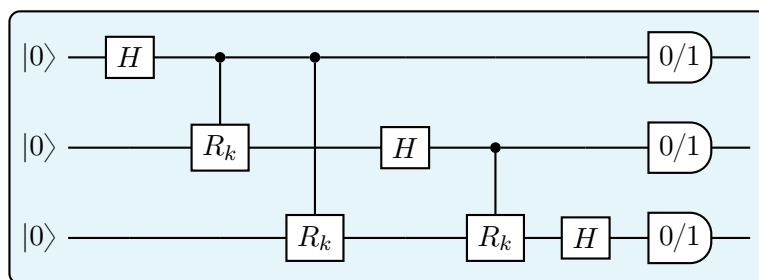
■ **Pasos:**

- Preparar el entrelazamiento:** Crear un par entrelazado entre qubits Q_1 y Q_2 utilizando una puerta CNOT y una puerta Hadamard.
- Medición de Bell⁷:** Aplicar una puerta CNOT y una puerta Hadamard al qubit Q_0 y medir ambos qubits Q_0 y Q_1 .
- Comunicación clásica:** Enviar los resultados de la medida al receptor.
- Recuperación:** Aplicar las puertas X y Z apropiadas a Q_2 basándose en los resultados de la medida para recuperar el estado original $|\psi\rangle$.

Transformada Cuántica de Fourier (QFT)

La QFT es el análogo cuántico de la transformada de Fourier clásica y se utiliza en varios algoritmos cuánticos, incluyendo el algoritmo de Shor para la factorización[4].

■ **Circuito:**



■ **Pasos:**

- Puerta Hadamard:** Aplicar una puerta Hadamard al primer qubit.
- Rotaciones controladas:** Aplicar rotaciones controladas a los qubits restantes.

⁷Una medición de Bell es una prueba experimental para verificar el *entanglement* (enredo cuántico) entre dos qubits. Aprovechando las correlaciones predichas por la mecánica cuántica para partículas entrelazadas, y que son imposibles de explicar con teorías clásicas, se mide el estado de los qubits en diferentes bases para verificar si efectivamente están vinculados. Estas mediciones son cruciales en la computación cuántica para garantizar la calidad del *entanglement* y desarrollar aplicaciones como la comunicación cuántica segura.

- c) **Aplicación recursiva:** Repetir el proceso para los qubits restantes.
- d) **Intercambio:** Intercambiar los qubits para invertir su orden, finalizando la QFT.

3.5.7. Algoritmos Cuánticos

Los algoritmos cuánticos son instrucciones paso a paso para resolver problemas cuánticos. Aprovechan la mecánica cuántica (superposición, entrelazamiento, interferencia cuántica) realizando múltiples cálculos simultáneamente, correlaciones complejas que los bits clásicos no pueden reproducir y utilizan la interferencia para amplificar la probabilidad de respuestas correctas mediante patrones de interferencia constructiva y destructiva resultantes de superposiciones cuánticas. Recordemos que la unidad básica de información es el qubit, que puede estar en dos estados a la vez (0 y 1).

Prometen resolver problemas específicos exponencialmente más rápido que los algoritmos clásicos.

Tienen un gran potencial en áreas como:

- **Criptografía:** En particular el algoritmo de Shor[4], amenazan la seguridad de los sistemas criptográficos clásicos. La criptografía post-cuántica es un campo emergente cuyo objetivo es desarrollar sistemas criptográficos resistentes a los ataques cuánticos.
- **Optimización:** Pueden resolver ciertos problemas de optimización con más eficacia que los algoritmos clásicos. Esto tiene aplicaciones en logística, finanzas y aprendizaje automático, donde es crucial encontrar soluciones óptimas con rapidez.
- **Ciencia de Materiales y Química:** Las simulaciones cuánticas pueden modelizar complejas interacciones moleculares y reacciones químicas, lo que permite avanzar en el descubrimiento de fármacos, el diseño de materiales y la comprensión de procesos químicos y físicos fundamentales.
- **Aprendizaje automático:** Los algoritmos cuánticos de aprendizaje automático pueden acelerar el procesamiento de datos y las tareas de reconocimiento de patrones, mejorando las capacidades de los sistemas de inteligencia artificial.

Comparación con Algoritmos Clásicos

- **Paralelismo:** Los algoritmos clásicos procesan un estado cada vez, mientras que los algoritmos cuánticos pueden evaluar muchos estados simultáneamente gracias a la superposición. Este paralelismo intrínseco puede dar lugar a importantes aumentos de velocidad en determinados tipos de problemas.
- **Complejidad:** A menudo, los algoritmos cuánticos pueden resolver problemas con clases de complejidad que se consideran inviables para los algoritmos clásicos. Por ejemplo, el algoritmo de Shor puede factorizar números enteros grandes en tiempo polinómico, mientras que los algoritmos clásicos más conocidos requieren tiempo superpolinómico.
- **Requisitos de Recursos:** Los algoritmos cuánticos suelen requerir menos pasos computacionales y recursos que sus homólogos clásicos. Sin embargo, dependen de la corrección de errores y de puertas cuánticas de alta fidelidad, lo que dificulta su implementación actual en dispositivos cuánticos ruidosos de escala intermedia (NISQ).

Contexto Histórico e Hitos

- **El Germen de la Idea:** A principios de los años 80, Richard Feynman y Yuri Manin fueron los primeros en darse cuenta del potencial de los ordenadores cuánticos. Propusieron que los sistemas cuánticos, regidos por las leyes de la mecánica cuántica, podían realizar cálculos de forma más eficiente que los ordenadores clásicos[2].
- **Un Desafío a la Criptografía:** En 1994, Peter Shor presentó un algoritmo capaz de descomponer grandes números enteros en tiempo polinómico, una hazaña imposible para los ordenadores clásicos. Su descubrimiento puso en entredicho la seguridad de sistemas criptográficos ampliamente utilizados, como el RSA[4].
- **Investigación Acelerada:** En 1996, Lov Grover desarrolló un algoritmo que permitía buscar en bases de datos no clasificadas el doble de rápido que con un método clásico. Este avance demostró el poder de la computación cuántica para resolver problemas de búsqueda no estructurados de forma más eficiente.
- **Protección de la Información Cuántica:** A mediados de los noventa, Peter Shor,

Andrew Steane y otros investigadores se enfrentaron al reto crucial de la decoherencia, el fenómeno que provoca la pérdida de información cuántica[46][47]. Desarrollaron códigos cuánticos de corrección de errores que preservan la integridad de la información durante los cálculos cuánticos.

- **Hacer Realidad la Cuántica:** Los recientes avances tanto en hardware como en algoritmos han impulsado el campo de la computación cuántica. Esto ha dado lugar a demostraciones reales de supremacía cuántica[51], en las que los ordenadores cuánticos superan a los clásicos en tareas específicas. Además, se han logrado avances significativos en la aplicación de la mecánica cuántica al aprendizaje automático, los problemas de optimización y las simulaciones de sistemas químicos.

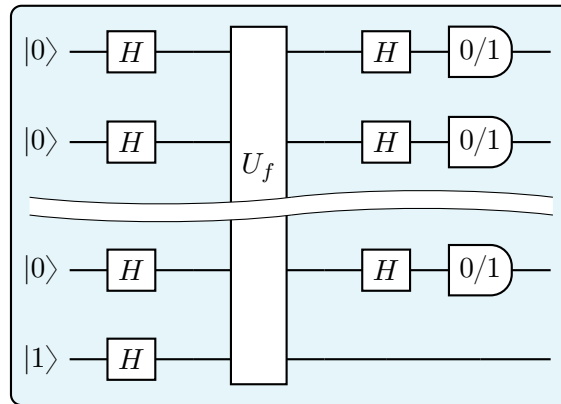
Algoritmos cuánticos fundamentales

1. Algoritmo Deutsch-Jozsa[6]

- **Asunto:** El algoritmo de Deutsch-Jozsa resuelve el problema de determinar si una función booleana dada $f : \{0, 1\}^n \rightarrow \{0, 1\}$ es constante (devuelve el mismo valor para todas las entradas) o balanceado (devuelve 0 para exactamente la mitad de las entradas y 1 para la otra mitad). En la informática clásica, resolver este problema para el peor de los casos requiere evaluar la función $2^{n-1} + 1$ veces. El algoritmo de Deutsch-Jozsa puede resolverlo con una única evaluación utilizando un ordenador cuántico.
- **Explicación del algoritmo:**
 - a) **Inicialización:**
 - Preparar n qubits en el estado $|0\rangle$ y un qubit adicional en el estado $|1\rangle$.
 - Aplicar la puerta de Hadamard a todos los qubits $n + 1$, creando una superposición de todos los estados de entrada posibles.
 - b) **Aplicación de Oráculo:**
 - Aplicar el oráculo cuántico U_f , que codifica la función f . El oráculo mapea $|x\rangle|y\rangle$ a $|x\rangle|y \oplus f(x)\rangle$.
 - c) **Interferencia:**
 - Vuelva a aplicar la puerta Hadamard a los primeros n qubits.
 - Mide los primeros n qubits.

- Si el resultado es completamente cero, la función es constante; de lo contrario está en equilibrio.

■ **Representación del Circuito Cuántico:**



■ **Comparación Clásica y Cuántica:**

- Clásica:** requiere como máximo $2^{n-1} + 1$ evaluaciones.
- Cuántica:** Requiere sólo una evaluación.

2. Algoritmo de Bernstein-Vazirani[52]

- **Asunto:** Dada una cadena de bits ocultos n a , el algoritmo de Bernstein-Vazirani determina a usando una función $f(x) = a \cdot x \text{ mód } 2$, donde \cdot denota el producto interno bit a bit. Clásicamente, encontrar a requiere consultas de n a f .

■ **Explicación del algoritmo:**

a) **Inicialización:**

- Preparar n qubits en el estado $|0\rangle$ y un qubit adicional en el estado $|1\rangle$.
- Aplicar la puerta de Hadamard a todos los qubits $n + 1$.

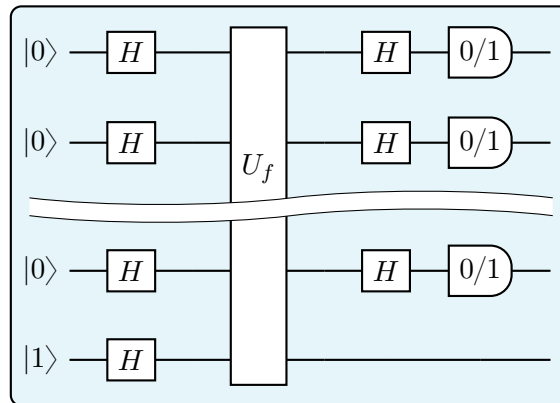
b) **Aplicación de Oráculo:**

- Aplicar el oráculo cuántico U_f , que codifica la función f . El oráculo asigna $|x\rangle|y\rangle$ a $|x\rangle|y \oplus f(x)\rangle$.

c) **Interferencia:**

- Vuelva a aplicar la puerta Hadamard a los primeros n qubits.
- Mide los primeros n qubits y da la cadena a .

d) **Representación del circuito cuántico:**



■ **Comparación Clásica y Cuántica:**

- a) **Clásica:** requiere n evaluaciones.
- b) **Cuántica:** Requiere sólo una evaluación.

3. **Algoritmo de Shor**^[4]

- **Problema de Factorización:** El algoritmo de Shor aborda el problema de la factorización de números enteros, específicamente encontrando los factores primos de un número compuesto N . Esto es crucial para la criptografía porque la seguridad del cifrado RSA depende de la dificultad de factorizar números enteros grandes.
- **Transformada Cuántica de Fourier (QFT):** QFT es un análogo cuántico de la transformada discreta de Fourier y es una parte importante del algoritmo de Shor. Transforma un estado cuántico en una superposición que codifica la información de periodicidad necesaria para la factorización.
- **Pasos del algoritmo:**
 - a) **Acortamiento del período observado:**
 - Reducir el problema de factorización a encontrar el período r de la función $f(x) = a^x \bmod N$, donde a es un número entero elegido al azar.
 - b) **Estimación de fase cuántica:**
 - Utilice QPE para encontrar el punto r .
 - c) **Postprocesamiento clásico**
 - Usa el punto r para encontrar los factores de N usando el algoritmo euclidiano

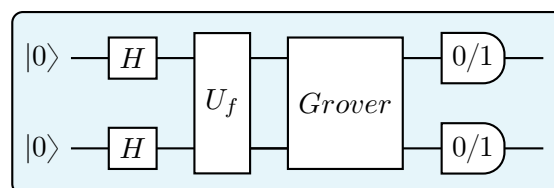
- **Impacto y aplicaciones:**

- **Cifrado RSA:** el algoritmo de Shor puede romper el cifrado RSA factorizando eficientemente números enteros grandes utilizados como claves.
- **Protocolos criptográficos:** La necesidad de algoritmos criptográficos resistentes a los cuánticos surge debido al impacto potencial del algoritmo de Shor.

4. Algoritmo de búsqueda de Grover^[50]

- **Problema de Búsqueda no Estructurada:** El algoritmo de Grover aborda el problema de la búsqueda en bases de datos no estructuradas, donde el objetivo es encontrar un elemento específico en una lista de (N) elementos. Mientras que los algoritmos clásicos requieren $O(N)$ evaluaciones para encontrar el elemento deseado, el algoritmo de Grover puede lograrlo en $O(\sqrt{N})$, lo que representa una mejora significativa. El algoritmo de Grover proporciona una aceleración cuadrática para problemas de búsqueda no estructurados.
- **Amplificación de Amplitud Cuántica:** El núcleo del algoritmo de Grover es la amplificación de amplitud, un proceso que aumenta la probabilidad de encontrar el estado deseado. Esto se hace a través de una serie de iteraciones que implican inversión de fase y rotación hacia el estado objetivo, optimizando el proceso de búsqueda.

- **Circuito:**



- **Pasos del algoritmo:**

- Inicialización:** Aplicar puertas Hadamard a todos los qubits para crear una superposición igual.
- Oráculo⁸ (U_f):** Aplicar la función oráculo para marcar el estado solución.

⁸Un oráculo cuántico es una función "caja negra" dentro de un algoritmo cuántico. Su implementación específica es desconocida para el algoritmo que la utiliza, pero actúa sobre qubits para realizar consultas

- c) **Amplificación de la amplitud:** Aplicar el operador de difusión de Grover para amplificar la amplitud de probabilidad del estado solución.
- d) **Repetición:** Repetir los pasos de oráculo y difusión aproximadamente \sqrt{N} veces, donde N es el número de soluciones posibles.
- e) **Medición:** Medir los qubits para obtener la solución.

■ **Impacto y aplicaciones:**

- **Búsqueda en bases de datos:** Aunque su aplicación a gran escala está limitada por la tecnología actual, el algoritmo de Grover ofrece una mejora significativa con respecto a los métodos clásicos.
- **Encriptación y seguridad:** Aunque Grover no afecta a los protocolos criptográficos como lo hace el algoritmo de Shor, puede reducir a la mitad la seguridad de algoritmos como AES, requiriendo el doble de tamaño de clave para mantener el mismo nivel de seguridad frente a un ataque cuántico.

3.5.8. Plataformas de Computación Cuántica.

Los marcos (*Frameworks* en inglés) de computación cuántica proporcionan herramientas para crear, simular y ejecutar circuitos cuánticos en hardware cuántico. Incluyen bibliotecas, simuladores y entornos de desarrollo adaptados a la computación cuántica.

Cubriremos varios marcos prominentes de computación cuántica, incluyendo Qiskit [53] de IBM, Cirq [54] de Google, Forest de Rigetti, PennyLane [55] de Xanadu, y otros entornos notables.

1. Qiskit (IBM)

Descripción General

Qiskit es un marco de computación cuántica de código abierto desarrollado por IBM. Proporciona herramientas para crear, manipular y simular circuitos cuánticos, así como para ejecutarlos en los ordenadores cuánticos de IBM.

eficientes a la función objetivo del problema. Esencial para algoritmos como el de Shor (factorización) y Grover (búsqueda), los oráculos cuánticos aceleran la resolución de problemas con funciones complejas o búsquedas no estructuradas. Por ejemplo, el oráculo de Shor evalúa la función $f(x, n)$, que determina si x divide a n . El circuito cuántico para este oráculo utiliza puertas CNOT y Toffoli para verificar la divisibilidad.

Principales Características

- **Circuitos cuánticos:** Herramientas para diseñar y visualizar circuitos cuánticos.
- **Simuladores:** Simuladores de alto rendimiento para probar y depurar circuitos cuánticos.
- **Transpilación:** Optimización de circuitos cuánticos para hardware específico.
- **Algoritmos cuánticos:** Librerías para la implementación de algoritmos cuánticos.

Ejemplo de Implementación en Python:

```
from qiskit import QuantumCircuit, Aer, transpile,
assemble, execute

# Crear un circuito cuántico con 2 qubits
qc = QuantumCircuit(2)

# Aplica una puerta Hadamard en el primer qubit
qc.h(0)

# Aplica una puerta CNOT con el primer qubit como control
# y el segundo como objetivo
qc.cx(0, 1)

# Medir ambos qubits
qc.measure_all()

# Simular el circuito
backend = Aer.get_backend('qasm_simulator')
result = execute(qc, backend).result()
counts = result.get_counts(qc)
print(counts)
```

2. Cirq (Google)

Descripción General

Cirq es un marco de computación cuántica de código abierto desarrollado por Google. Está diseñado para crear, simular y ejecutar circuitos cuánticos en los procesadores

cuánticos de Google.

Principales Características

- **API flexible:** Herramientas para construir y manipular circuitos cuánticos.
- **Modelado de ruido:** Capacidades para simular ruido en circuitos cuánticos.
- **Integración:** Funciona bien con otras herramientas y librerías de Google.
- **Algoritmos cuánticos:** Librerías para varios algoritmos y protocolos cuánticos.

Ejemplo de Implementación en Python:

```
import cirq

# Crear qubits
q0, q1 = cirq.LineQubit.range(2)

# Crear un circuito cuántico
circuit = cirq.Circuit(
    cirq.H(q0),
    cirq.CNOT(q0, q1),
    cirq.measure(q0, q1)
)

# Imprimir el circuito
print(circuit)

# Simular el circuito
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=1000)
print(result)
```

3. Forest (Rigetti)

Descripción General

Forest es un marco de computación cuántica desarrollado por Rigetti Computing. Proporciona herramientas para construir y ejecutar circuitos cuánticos, con especial atención a los qubits superconductores.

Principales Características

- **Programas cuánticos:** Herramientas para crear programas cuánticos utilizando Quil (Quantum Instruction Language).
- **Simuladores:** Acceso a máquinas virtuales cuánticas (QVMs) para simulación.
- **Integración con QCS:** Integración con Quantum Cloud Services (QCS) de Rigetti para ejecutar circuitos en hardware cuántico real.
- **Computación Híbrida Cuántica-Clásica:** Herramientas para integrar computación cuántica y clásica.

Ejemplo de Implementación en Python:

```
from pyquil import Program, get_qc
from pyquil.gates import H, CNOT, MEASURE

# Crear un programa cuántico
p = Program()
ro = p.declare('ro', 'BIT', 2)

# Aplica una puerta Hadamard en el primer qubit
p += H(0)

# Aplica una puerta CNOT con el primer qubit como control
# y el segundo como objetivo
p += CNOT(0, 1)

# Medir ambos qubits
p += MEASURE(0, ro[0])
p += MEASURE(1, ro[1])

# Ejecuta el programa en una máquina virtual cuántica
qc = get_qc('2q-qvm')
result = qc.run_and_measure(p, trials=1000)
print(result)
```

4. PennyLane (Xanadu)

Descripción General

PennyLane es un marco de computación cuántica desarrollado por Xanadu. Se centra en cálculos híbridos cuántico-clásicos y está diseñado para trabajar con varios backends de hardware cuántico.

Principales características

- **Aprendizaje automático cuántico:** Herramientas para implementar algoritmos de aprendizaje automático cuántico.
- **Integración:** Funciona con las principales bibliotecas de aprendizaje automático como PyTorch y TensorFlow.
- **Programación Diferenciable Cuántica:** Capacidades para programación diferenciable y optimización.
- **Hardware Agnostic:** Compatible con múltiples plataformas de hardware cuántico.

Ejemplo de Implementación en Python:

```
import pennylane as qml
from pennylane import numpy as np

# Definir un dispositivo cuántico
dev = qml.device("default.qubit", wires=2)

# Define un circuito cuántico como un QNode
@qml.qnode(dev)
def circuit(x):
    qml.Hadamard(wires=0)
    qml.CNOT(wires=[0, 1])
    qml.RX(x, wires=0)
    return qml.expval(qml.PauliZ(0))

# Evalúa el circuito
result = circuit(np.pi / 4)
print(result)
```

3.5.9. Aplicaciones.

Los qubits, como elementos fundamentales de la informática cuántica, incorporan principios y conceptos de la mecánica cuántica, las matemáticas y la teoría de la información. Sus propiedades únicas, como la superposición y el entrelazamiento, permiten potentes capacidades computacionales que van más allá de los sistemas clásicos. Comprender las diferencias entre los qubits utilizados en el quantum annealing y los ordenadores cuánticos basados en compuertas es crucial para optimizar sus aplicaciones en la resolución de problemas diversos. A medida que progrese este campo, la integración de estas perspectivas impulsará los avances de la tecnología cuántica y sus aplicaciones prácticas.

4. Desarrollo e Implementación

En este capítulo desarrollaremos nuestro trabajo en base a los conceptos desarrollados en los capítulos anteriores.

Tomaremos como referencia la molécula de hidrógeno (H_2) e introduciremos cómo la física modela su comportamiento a través de la ecuación de Schrödinger (4.1) [56] y el operador Hamiltoniano del sistema molecular. También discutiremos por qué sus propiedades moleculares pueden resultar interesantes de simular computacionalmente y cuáles son los enfoques actuales para ello.

Las propiedades moleculares sobre las que trabajaremos son las siguientes:

- Geometría molecular
- Configuración electrónica
- Energía del estado fundamental

La idea de usar como referencia una molécula sencilla como es el H_2 es poder introducir estos conceptos sin complicar su desarrollo matemático de manera innecesaria. Asentados estos conceptos, podremos extrapolarlos y aplicarlos a moléculas más complejas.

Tras esta introducción del sistema físico que estamos tratando, implementaremos las rutinas cuánticas que nos permitirán modelar las distintas características moleculares de interés, no solo para el caso del H_2 , sino también para moléculas más complejas.

4.1. La molécula de Hidrógeno

El Hidrógeno no solo es el elemento químico más sencillo y ligero de la tabla periódica, si no que también es el elemento más abundante en el universo. El Hidrógeno cuenta con un número atómico de 1, lo que significa que únicamente está formado por un protón y un electrón, y en condiciones ambientales lo podemos encontrar como un gas di-atómico, conocido como Hidrógeno molecular (H_2).

En este tipo de sistema, de escala microscópica, la física que rige su comportamiento es la mecánica cuántica. La ecuación de Schrödinger (4.1) es una ecuación diferencial que rige el comportamiento de la función de onda de un sistema cuántico.

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle \quad (4.1)$$

Donde t es el tiempo, $|\psi(t)\rangle$ es el vector que representa el estado cuántico y \hat{H} el operador Hamiltoniano del sistema. La modelización de moléculas a través de la ecuación de Schrödinger no es algo sencillo, ya que, para la mayoría de moléculas con más de un electrón, no existe solución analítica para la ecuación de Schrödinger (4.1). Esto, hará que sea necesario el uso de aproximaciones como la de Hartee-Fock [57] para poder obtener el resultado aproximado del estado del sistema.

La molécula de Hidrógeno es la molécula más sencilla que podemos encontrar en la cual existe interacción electrónica, es por eso, que es un muy buen punto de partida para comprender y estudiar la modelización física de moléculas multielectrónicas.

4.1.1. Modelización

Como ya hemos mencionado en la introducción a esta sección, la modelización física de este tipo de sistemas se hace mediante la ecuación de Schrödinger (4.1). Esta ecuación nos permite modelizar la evolución espacio-temporal de un sistema cuántico mediante la aplicación del operador Hamiltoniano que lo caracteriza.

Es sencillo ver que uno de los puntos claves de la modelización física de un sistema molecular es la definición de su Hamiltoniano \hat{H} . Utilizando los conceptos introducidos en secciones anteriores, buscaremos definir el problema en base a un Hamiltoniano electrónico de tal forma que podamos utilizarlo en un computador cuántico para simular dicho sistema [58].

Una de las primeras aproximaciones que utilizaremos en la definición de este Hamiltoniano es asumir que los núcleos de la molécula están fijos y que entonces, el Hamiltoniano dependerá de los parámetros de posición de los distintos núcleos y de sus electrones. Esta aproximación es conocida como la aproximación Born-Oppenheimer [59].

$$\hat{H}(\vec{r}) |\psi(\vec{r})\rangle = E(\vec{r}) |\psi(\vec{r})\rangle \quad (4.2)$$

Donde \vec{r} es el vector que contiene la posición relativa entre los núcleos y los electrones.

Además, para la construcción de dicho Hamiltoniano electrónico, utilizaremos es el formalismo de segunda cuantización [60]. Este formalismo se utiliza para estudiar y describir sistemas cuánticos de muchos cuerpos, donde se se utilizan las ocupaciones de los orbitales moleculares como base del sistema.

Estos orbitales moleculares describirán la función de onda de cada electrón y se pueden

aproximar por una combinación lineal de orbitales atómicos. Para la caracterización de estos orbitales utilizaremos el método de Hartee-Fock [61]. Podemos resumir las asunciones que toma el método de Hartee-Fock en:

- Uso de la aproximación Born-Oppenheimer
- No se tienen en cuenta efectos relativistas.
- Se utiliza una aproximación de los orbitales moleculares que resulta de la combinación lineal de orbitales atómicos. Esta base de orbitales atómicos es ortogonal y aproximadamente completa.
- Cada una de las autofunciones de la energía se puede describir mediante el producto de orbitales de un único electrón. Esto se puede formular también en términos del determinante de Slater [62].
- Asumiremos la aproximación de campo medio, donde los electrones del sistema se tratan como partículas independientes que se mueven libremente por el potencial creado por los núcleos y el resto de electrones que componen la molécula [63].

Una vez introducidas las condiciones bajo las cuales vamos a definir nuestro Hamiltoniano, describiremos en detalle el proceso para definirlo. Como hemos mencionado anteriormente, trabajaremos bajo el formalismo de segunda cuantización, en el cual, la función de onda de la molécula se representa en base a los estados de ocupación de los M orbitales moleculares.

$$|n_0, n_1, \dots, n_{M-1}\rangle \quad (4.3)$$

Donde n_i tomará el valor 0 si el orbital no está ocupado y 1 si sí lo está. En este contexto, podemos definir nuestro Hamiltoniano en base a los operadores de creación y destrucción

$$H = \sum_{p,q} h_{pq} c_p^\dagger c_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} c_p^\dagger c_q^\dagger c_r c_s \quad (4.4)$$

Donde c^\dagger, c son los operadores de creación y destrucción y los coeficientes h_{pqrs} y h_{pq} corresponden a los valores de las integrales de Coulomb para uno y dos electrones.

Estos operadores funcionan de tal forma que el operador de creación aumenta en una unidad el número de partículas del sistema, y el operador de destrucción lo contrario. Es

sencillo ver como estos operadores son de gran utilidad para describir sistemas en los que existen partículas en distintos niveles energéticos, pudiendo transicionar de un nivel a otro.

Es importante destacar que en este tipo de sistemas se debe conservar el número total de partículas. Esto se puede evaluar de manera sencilla haciendo uso del operador número $\hat{N} = c^\dagger c$ donde $\hat{N} |\psi\rangle = n |\psi\rangle$.

Para el cálculo de los valores de las integrales de Coulomb se puede utilizar el método de Hartee-Fock, que nos permitirá aproximar los orbitales moleculares de forma que podamos obtener el valor de las integrales de Coulomb de manera relativamente sencilla [64].

$$h_{pq} = \int dx \phi_p^*(x) \left(-\frac{\nabla^2}{2} - \sum_{i=1}^N \frac{Z_i}{|r_i|} \right) \phi_q(x) \quad (4.5)$$

$$h_{pqrs} = \int dx_1 dx_2 \frac{\phi_p^*(x_1) \phi_q^*(x_2) \phi_r(x_2) \phi_s(x_1)}{|r_1 - r_2|} \quad (4.6)$$

Donde ϕ son los orbitales moleculares aproximados por el método de Hartee-Fock y el término $(-\frac{\nabla^2}{2} - \sum_{i=1}^N \frac{Z_i}{|r_i|})$ hace referencia a la interacción entre los electrones y los núcleos.

En este punto, si queremos simular nuestro sistema molecular en un computador cuántico, debemos encontrar la forma de transformar la información de los estados de ocupación $|n_0, n_1, \dots, n_{M-1}\rangle$ en qubits, así como definir el Hamiltoniano del sistema molecular en base a operadores que puedan ser implementados como puertas cuánticas.

Es sencillo ver que podemos mapear el estado de M orbitales definido en base a los estados de ocupación $|n_i\rangle$ con un estado de M qubits $|q_i\rangle$ donde los valores 0 y 1 sigan la misma lógica.

Cuando hemos introducido en secciones anteriores el Hamiltoniano de Ising, una de las características fundamentales del mismo es que utilizaba los estados de spín del sistema como las variables binarias que se utilizan para definir nuestro problema. Algo muy interesante de esto, es que los distintos estados de spin del sistema se pueden caracterizar con los operadores de Pauli $\sigma_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}$. Estos operadores además son una de las piezas fundamentales de la computación cuántica ya que se pueden implementar en puertas de manera sencilla.

A la hora de definir nuestro Hamiltoniano para su implementación en computadores cuánticos, buscaremos que este esté definido en base a operadores de Pauli $\sigma_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}$. Esto se puede conseguir mediante la transformación de Jordan-Wigner [65]. Esta transformación nos permitirá descomponer en operadores de Pauli los operadores

$\{c^\dagger, c\}$. De esta forma obtendremos un Hamiltoniano expresado como combinación lineal de productos tensoriales de operadores de Pauli $\sigma_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}$.

$$H = \sum_j C_j \otimes_i \sigma_i^{(j)} \quad (4.7)$$

Donde C_j es un coeficiente escalar.

Con esta transformación hemos conseguido mapear un Hamiltoniano fermiónico formulado en base a los operadores de creación y destrucción con un Hamiltoniano de spin caracterizado por operadores de Pauli que podremos implementar de forma sencilla en un computador cuántico de tal forma que seamos capaces de buscar la configuración de spines que minimice la energía de la molécula.

Pese a que los fundamentos de esta formulación se basan en la estructura del Hamiltoniano de Ising y en el uso de operadores de Pauli para la definición del Hamiltoniano, el Hamiltoniano resultante no es exactamente un Hamiltoniano de Ising. Como ya sabemos, esto puede ser un problema para ciertas rutinas cuánticas, como *Quantum Annealing*, pero no lo es para algoritmos como VQE, donde lo que buscamos es tener un Hamiltoniano que podamos implementar de manera relativamente sencilla utilizando las puertas de Pauli.

Por otro lado, pese a que gracias a las aproximaciones que hemos asumido, el problema se puede formular de manera relativamente sencilla, su implementación sigue contando con ciertas dificultades. Pese a que los computadores cuánticos pueden abordar de manera mucho más eficiente que los computadores clásicos este tipo de problemas, con el hardware disponible actualmente, es posible que un computador cuántico tampoco tenga los recursos suficientes para abordar este problema tan complejo.

Para solucionar esto surgen aproximaciones enfocadas en reducir la complejidad del problema que buscamos resolver. Una de las aproximaciones que nos permitirá reducir el tamaño del problema es dividir los orbitales moleculares en tres tipos:

- *Core*: Son orbitales que se encuentran en las capas más profundas de la molécula y que consideraremos que siempre están completamente ocupados.
- *Activos*: Son los orbitales en los que podemos encontrar 0, 1 o 2 electrones. Estos orbitales son los que en la práctica, optimizaremos con nuestro computador cuántico.
- *Exteriores*: Son orbitales muy alejados del núcleo molecular en los cuales nunca encontraremos electrones.

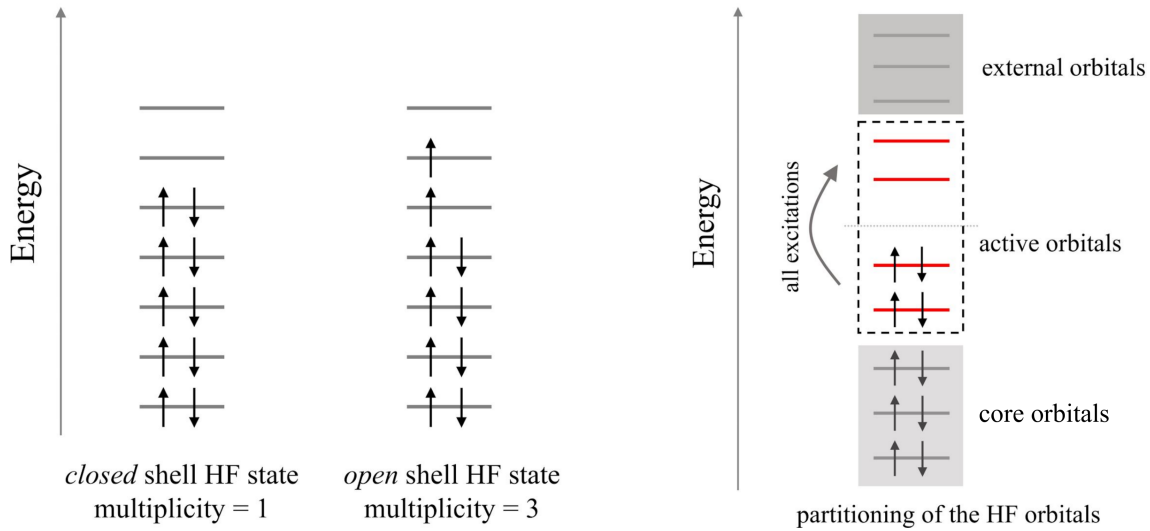


Figura 4.1: Esquema de la estructura de los orbitales moleculares para diferentes aproximaciones. Fuente: [Pennylane](#)

Teniendo en mente que el objetivo de esta caracterización del sistema molecular es su posterior simulación en un computador cuántico, es fácil ver como esta caracterización de orbitales reduce el espacio de búsqueda de soluciones óptimas únicamente a los orbitales activos. Este hecho reduce de forma significativa los recursos necesarios para la simulación.

Hemos introducido brevemente una de las aproximaciones para caracterizar de forma sencilla orbitales moleculares, pero a la hora de definir y caracterizar estos orbitales, existen distintas aproximaciones a la hora de definir nuestro espacio activo (Fig. (4.1)). En este trabajo no vamos a profundizar en cómo definir este espacio activo, ya que es un tema complejo que sigue siendo objeto de estudio. En [66], [67] podemos encontrar una buena base para los lectores que quieran profundizar en ello.

En este punto ya tenemos las herramientas necesarias para modelizar un sistema molecular de tal forma que podamos simularlo en un computador cuántico.

4.1.2. Propiedades Físicas

Una vez caracterizado nuestro sistema molecular, existen ciertas propiedades que podrían ser de interés a la hora de estudiar dichos sistemas. En este trabajo nos enfocaremos en tres de ellas:

- Configuración electrónica

- Energía del estado fundamental
- Geometría molecular

En esta sección detallaremos como estudiar dichas propiedades y como modelizarlas físicamente de tal forma que luego podamos obtener sus valores esperados en un computador cuántico.

Configuración Electrónica

A la hora de estudiar un sistema molecular, conocer su configuración electrónica, por ejemplo, en su estado fundamental resulta de gran interés. La configuración electrónica determinará propiedades físicas y químicas del sistema como por ejemplo su reactividad o propiedades ópticas y magnéticas.

Definiremos la configuración electrónica de un sistema molecular por el estado cuántico $|\psi\rangle$ que define la ocupación electrónica de los distintos orbitales. En el contexto en el que estamos trabajando, el marco de Jordan-Wigner, podemos representar dicho estado de forma que cada qubit represente un hueco dentro del orbital, tomando el valor $|1\rangle$ si dicho hueco está ocupado y $|0\rangle$ si no.

Como estamos trabajando con electrones, partículas fermiónicas, el principio de exclusión de Pauli limita la ocupación de los orbitales de tal forma que como máximo podremos tener dos electrones en un orbital, uno con spin $S = \uparrow$ y otro con spin $S = \downarrow$. Para ilustrar esto, pongamos como ejemplo el siguiente estado:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|1100\rangle + |0011\rangle)$$

En la convención que estamos tomando, representa un estado de superposición de dos electrones que pueden ocupar dos orbitales, donde el primer término representan los dos electrones en el primer orbital, el primero con spin $S = \uparrow$ y el segundo con spin $S = \downarrow$ y el segundo término el caso en el que los electrones estuviesen ambos en el segundo orbital.

Dado un sistema molecular, puede ser interesante no solo conocer su configuración electrónica, si no también el spin total del sistema. Para poder obtener esta propiedad debemos definir primero el operador spin total \hat{S}^2 .

$$\hat{S}^2 = \frac{3}{4}N + \sum_{\alpha,\beta,\gamma,\delta} \langle \alpha, \beta | \hat{s}_1 \cdot \hat{s}_2 | \gamma, \delta \rangle \hat{c}_\alpha^\dagger \hat{c}_\beta^\dagger \hat{c}_\gamma \hat{c}_\delta \quad (4.8)$$

Donde N es el número de electrones y $\langle \alpha, \beta | \hat{s}_1 \cdot \hat{s}_2 | \gamma, \delta \rangle$ representa los diferentes elementos de la matriz del operador de espín de dos electrones $\hat{s}_1 \cdot \hat{s}_2$

Este operador actúa de la siguiente forma sobre los distintos autoestados de spin

$$\langle \psi | \hat{S}^2 | \psi \rangle = s(s + 1) \quad (4.9)$$

Donde s representa el spin del sistema.

Energía del Estado Fundamental

Conocer la energía del estado fundamental es la energía total del sistema en su configuración de mínima energía. Conocer esta energía nos permitirá conocer también la estabilidad de la molécula. Además, puede ser la base para el estudio de otras propiedades termodinámicas de interés como la energía de enlace de la molécula, sus estados excitados.

La energía asociada a un estado $|\psi\rangle$ lo podemos conseguir a través de la obtención del valor esperado del Hamiltoniano del sistema en el estado $|\psi\rangle$

$$\langle \psi | \hat{H} | \psi \rangle = E_\psi$$

Entonces si queremos encontrar la energía del estado fundamental, únicamente debemos calcular el valor esperado del Hamiltoniano en su estado fundamental $|\psi_0\rangle$. El problema de encontrar dicho estado fundamental es lo que buscamos resolver mediante algoritmos cuánticos en este trabajo.

Geometría Molecular

La geometría molecular es la estructura tridimensional que toman los diferentes núcleos que forman una molécula. La optimización de la geometría molecular es una tarea fundamental en química cuántica, crítica para comprender las propiedades físicas y químicas de las moléculas. La configuración geométrica de una molécula, en particular las longitudes de enlace y los ángulos entre átomos, desempeña un papel significativo en la determinación de su estabilidad, reactividad e interacción con otras moléculas. La predicción precisa de estos parámetros geométricos es esencial para las simulaciones que pretenden modelar el comportamiento molecular en diversas condiciones.

Los métodos tradicionales para el estudio de la estructuras e interacciones moleculares, como Hartree-Fock (HF) y la Teoría Funcional de la Densidad (DFT), han permitido

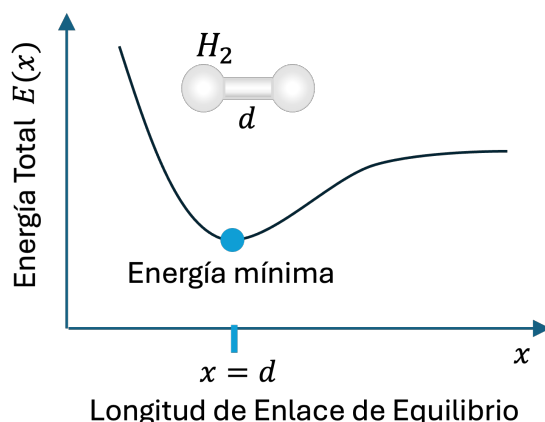


Figura 4.2: Relación de la Geometría Óptima con el nivel de Energía del Sistema.

comprender mejor los sistemas moleculares. Sin embargo, estos métodos a menudo se enfrentan a retos en términos de precisión y coste computacional, especialmente en el caso de moléculas grandes y complejas [68][69].

En el caso de la molécula de hidrógeno (H_2), los núcleos son mucho más pesados que los electrones, lo que nos permite tratarlos como partículas de punto fijo. La energía total de la molécula $E(x)$ depende de las coordenadas nucleares x , que definen la superficie de energía potencial. El objetivo es resolver el problema estacionario $\nabla_x E(x) = 0$, que determina la geometría de equilibrio de la molécula. La siguiente Figura 4.2 ilustra este concepto. En este mínimo, los dos átomos de hidrógeno estarán localizados a una distancia que es la Longitud de Enlace Optimizada d .

La geometría molecular está íntimamente relacionada con la energía del estado fundamental, ya que al crearse los enlaces moleculares, los núcleos se distribuirán en el espacio cumpliendo el principio de mínima energía. Este hecho es muy interesante ya que podemos obtener la geometría molecular de ciertos compuestos optimizando su distancia y ángulos de enlace de tal forma que la energía del sistema sea mínima.

En la siguiente sección discutiremos por qué esto es un tema de gran interés para industrias como la química o la ciencia de materiales.

4.1.3. Simulación computacional y sus aplicaciones

La simulación computacional de sistemas moleculares hace posible el estudio y análisis de ciertos el comportamiento o características de sistemas moleculares sin la necesidad de tener que crearlos y caracterizarlos en un laboratorio.

Debido a la complejidad de los sistemas moleculares, los métodos clásicos de simulación molecular y los métodos experimentales que se pueden implementar en laboratorio, suelen estar limitados a moléculas sencillas con pocos electrones. El motivo es que el número de parámetros necesarios para describir un sistema molecular escala de forma exponencial con el tamaño de la molécula. Como ya hemos introducido en secciones anteriores, la computación cuántica es capaz de atacar eficientemente este tipo de problemas, emergiendo como una herramienta de gran utilidad para tratar este tipo de sistemas complejos [70].

Las aplicaciones de la computación cuántica a este tipo de problemas puede ir desde casos relativamente sencillos como la caracterización de la estructura electrónica, la energía o la geometría de moléculas sencillas, casos que estudiaremos en este trabajo, hasta problemas mucho más complejos como el descubrimiento de nuevos principios activos para fármacos [71] o la creación de disipadores de calor más eficientes [18].

4.2. *Ansatz* del Problema

Una vez definido el problema de optimización que buscamos abordar con computación cuántica, debemos escoger el algoritmo que utilizaremos para ello. Existen numerosas alternativas para esta tarea como *Quantum Annealing*, VQE [30] o QAOA [72]. En este trabajo, dada la definición del Hamiltoniano molecular que hemos visto, utilizaremos el algoritmo VQE (*Variational Quantum Eigensolver*) para resolver los problemas de optimización que estamos tratando.

Como ya hemos mencionado en secciones anteriores, la definición del *ansatz* es algo bastante delicado que debemos de estudiar antes de implementar nuestro algoritmo, pero en nuestro caso esto será más sencillo, ya que los estados electrónicos de un sistema molecular son bien conocidos, por lo que únicamente debemos parametrizar las amplitudes asociadas a cada configuración electrónica, de tal forma que el propio algoritmo aumentará o disminuirá las amplitudes de los distintos estados en función de su energía asociada. En nuestro caso, nuestro *ansatz* será

$$|\Psi(\theta)\rangle = c_{\text{HF}}(\theta)|1100\rangle + c_1(\theta)|0011\rangle + c_2(\theta)|0110\rangle + c_3(\theta)|1001\rangle$$

Siendo $|\psi\rangle(\theta)$ el *ansatz* que representa las distintas configuraciones posibles de un sistema con dos electrones y dos orbitales, donde los diferentes coeficientes c corresponden a las amplitudes de los distintos estados de spin del sistema.

Para generar el *ansatz* lo primero que debemos hacer es generar el estado fundamental de Hartee-Fock. En este estado tendremos ambos electrones en el primer orbital molecular.

$$|\psi_{HF}\rangle = |1100\rangle \quad (4.10)$$

Partiendo de dicho estado crearemos los distintos estados excitados aplicando los operadores de excitación de una y dos partículas. Por ejemplo, el operador de excitación de una partícula creará un nuevo estado en el que una de las partículas ha sido excitado.

Este tipo operadores son conocidos como rotaciones de Givens [73]. Podemos interpretar entonces la acción de este operador como una rotación del tipo

$$G(\theta) |10\rangle = \cos\left(\frac{\theta}{2}\right) |10\rangle - \sin\left(\frac{\theta}{2}\right) |01\rangle$$

Donde el estado cuántico representa un electrón con estado de spin *up* o *down*. Es sencillo ver que además este operador conservará el número de partículas del sistema. El funcionamiento del operador de excitación de dos partículas es análogo pero excita dos partículas en vez de una. El circuito que implementa este tipo de rotaciones tendrá la siguiente forma:

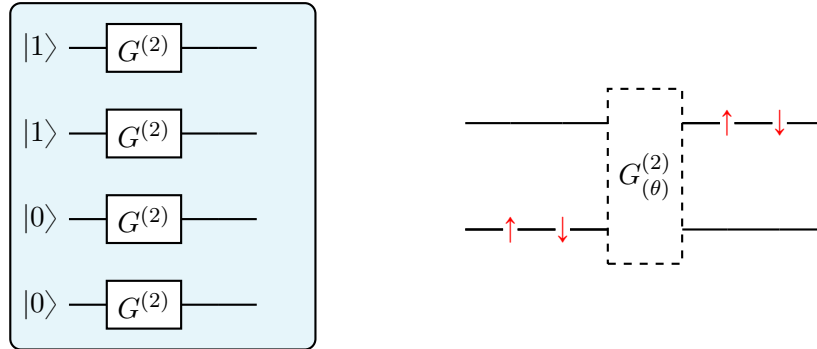


Figura 4.3: Circuitos que implementa las rotaciones de Givens necesarias para la definición del *ansatz*

De esta forma es relativamente sencillo de ver que podemos construir el *ansatz* en base a estos operadores, donde los parámetros variacionales serán los ángulos de estas rotaciones.

La forma definida, es la más general posible para un sistema molecular de dos electrones. Dependiendo del problema que busquemos abordar, este *ansatz* puede variar ligeramente. Por ejemplo, veremos más adelante que en el caso de buscar la geometría óptima

de nuestro sistema molecular, es posible que, por ejemplo, únicamente necesitemos un estado con dobles excitaciones. En ese caso los términos de las excitaciones simples tendrían amplitud cero, resultando en un *ansatz*

$$|\psi(\theta)\rangle = c_{HF}(\theta)|1100\rangle - c_1(\theta)|0011\rangle$$

4.3. Implementación

En las secciones anteriores hemos proporcionado las herramientas necesarias para definir nuestro problema en base a un Hamiltoniano para poder abordarlo como un problema de optimización combinatoria. Hemos presentado también el problema de la molécula de Hidrógeno en base a su caracterización física y como podemos adaptar esa modelización para que podamos implementarlo en un computador cuántico de manera sencilla. Por último se ha detallado el principal algoritmo de optimización que se va a utilizar en este estudio.

En este punto contamos con todas las herramientas teóricas para modelizar y simular una molécula sencilla como la de H_2 en un computador cuántico.

En esta sección mostraremos como podemos abordar este problema en uno de los SDK más populares para computación cuántica, PennyLane. Primero introduciremos como caracterizar la molécula que queremos estudiar, para más adelante caracterizar ciertas propiedades de interés:

- Configuración Electrónica
- Geometría Molecular
- Energía del estado Fundamental

4.3.1. Configuración de Entorno

En esta sección mencionaremos los paquetes de Python que utilizaremos en el desarrollo de los códigos necesarios para la resolución de los problemas propuestos.

NumPy

NumPy es una librería de Python cuyo nombre es la abreviación de *Numeric Python*. Es una librería que permite trabajar con arrays multidimensionales, como vectores o matrices

y operar con ellos de manera sencilla. Las operaciones disponibles van desde ordenaciones y cambios de dimensión a operaciones algebraicas, estadística, etc.

Para instalar NumPy podemos ejecutar `pip install numpy`

PennyLane

PennyLane es una librería de Python que permite el desarrollo de rutinas cuánticas de manera que puedan ser ejecutadas en distintos *backends*. Una de las mayores ventajas de PennyLane es su integración con otras librerías muy comunes en el mundo del *Machine Learning* como TensorFlow, JAX o PyTorch.

Para instalar PennyLane podemos ejecutar `pip install pennylane`

Matplotlib

Matplotlib es una librería de visualización de datos que permite la generación de gráficas tanto estáticas como interactivas de manera sencilla. La utilizaremos para la representación gráfica de los resultados obtenidos.

Se puede instalar a través del comando `pip install matplotlib`

Hemos introducido los distintos paquetes que vamos a necesitar y como instalarlos individualmente. Una manera más sencilla de asegurarnos que tenemos todo lo necesario en nuestro entorno para poder ejecutar los siguientes códigos es a través de un fichero de requerimientos `requirements.txt`. Este fichero contendrá una lista de los paquetes y versiones que necesitamos. En nuestro caso sería el siguiente

```
pennylane==0.32.0
numpy==1.22.4
matplotlib==3.7.2
PennyLane-Lightning==0.36.0
```

Por lo que si ejecutamos el comando `pip install -r requirements.txt` nuestro entorno quedaría configurado. Una vez instalados las distintas librerías únicamente debemos de importarlas en nuestro código según las necesitemos.

```
%matplotlib inline
import matplotlib.pyplot as plt
import pennylane as qml
from pennylane import numpy as np
from pennylane import Hamiltonian, PauliX, PauliY, PauliZ
from pennylane.optimize import AdamOptimizer
```

4.3.2. Caracterización de la Molécula

En esta sección detallaremos como trabajar con el ecosistema de PennyLane para la construcción del Hamiltoniano de una cierta molécula.

En PennyLane se puede generar el Hamiltoniano de una cierta molécula de distintas formas. Veamos como podemos caracterizar la molécula de H_2 :

1. **Coordenadas Nucleares:** En este método, especificaremos de forma explícita las coordenadas nucleares de nuestra molécula en unidades atómicas, así como los átomos que la forman.

```
# Definimos los parámetros de la molécula que buscamos simular  
# Elementos de la molécula  
elementos = ["H", "H"]  
coordenadas = np.array([0.0, 0.0, -0.6614, 0.0, 0.0, 0.6614])  
electrones = 2
```

Una vez definida la molécula, podemos hacer uso de `qml.qchem.Molecule(symbols, coordinates)` para crear el objeto que contendrá la información de nuestra molécula.

En este punto, generaremos el Hamiltoniano y el número de qubits necesario para representarlo haciendo uso de `qml.qchem.molecular_hamiltonian(molecule)`. Esta función será la encargada de construir el Hamiltoniano del sistema haciendo uso del formalismo de segunda cuantización y el método de Jordan-Wigner para convertir el Hamiltoniano fermiónico en un Hamiltoniano de spin basado en puertas de Pauli. En esta función también podemos definir ciertos parámetros de la aproximación, como el espacio activo de orbitales o el método para aproximar el Hamiltoniano, en nuestro caso usaremos el método por defecto que es el de Hartee-Fock.

```
# Creamos el Hamiltoniano de la molécula y el número de qubits  
# necesarios para su simulación  
# Creamos la molécula  
molecula = qml.qchem.Molecule(elementos, coordenadas)  
  
# Obtenemos el Hamiltoniano y el número de qubits  
H, qubits = qml.qchem.molecular_hamiltonian(molecula)  
  
# Imprimimos el número de qubits y el Hamiltoniano  
print("Qubits = ", qubits)  
print("Hamiltoniano: ", H)
```

2. **Fichero de Geometría Molecular:** Existen ficheros que contienen la información de geometría molecular de ciertos sistemas. En nuestro caso, PennyLane da soporte a ficheros del tipo *.xyz*. Este tipo de ficheros se pueden obtener de múltiples repositorios, uno de los cuales ofrece una interfaz intuitiva y muy útil es [Hack-a-Mol](#).

Una vez contamos con nuestro fichero, podemos importarlo haciendo uso de la función `qml.qchem.read_structure()`

```
# Importamos la molécula de un fichero
elementos, coordenadas = qchem.read_structure("h2.xyz")

# Construimos la molécula
molecula = qchem.Molecule(elementos, coordenadas)
```

En este punto podemos construir el Hamiltoniano de forma análoga al caso anterior.

3. **Repositorio PennyLane:** PennyLane cuenta con una base de datos relacionada con computación cuántica llamada [Quantum Data](#). En esta base de datos podemos encontrar multitud de *datasets* como por ejemplo:

- Sistemas Moleculares
- Hamiltonianos
- Estados cuánticos o circuitos para su preparación
- Operadores Unitarios de interés

Dentro de toda esta información podemos encontrar información relacionada con sistemas moleculares, en la cual nosotros estamos interesados. Para hacer uso de esta información debemos cargarla en nuestra sesión haciendo uso de `qml.data.load()` de tal forma que podemos obtener la estructura molecular y el Hamiltoniano de un sistema de la siguiente forma:

```
# Cargamos los datos de la molécula
molecula = qml.data.load('qchem', molname="H2")[0]

# Obtenemos el Hamiltoniano y los qubits necesarios
H, qubits = molecula.hamiltonian, len(dataset.hamiltonian.wires)
```

El Hamiltoniano obtenido es análogo al que hemos obtenido por cualquiera de los métodos anteriores.

Elijamos el método que elijamos, en este punto ya tendríamos nuestra molécula caracterizada por un Hamiltoniano de spin. Este Hamiltoniano lo podremos atacar con computadores cuánticos como un problema de optimización combinatoria basado en variables binarias en el que, por ejemplo, buscamos la configuración de spines que minimice la energía de la molécula.

4.3.3. Geometría Molecular

Una vez caracterizada nuestra molécula por su Hamiltoniano, en esta sección investigaremos la optimización cuántica de la geometría molecular de la molécula de hidrógeno (H_2) utilizando dos enfoques distintos: el marco Ising-QUBO con Annealing Cuántico y el Eigensolver Cuántico Variacional (VQE) y . Estos métodos ofrecen diferentes vías para abordar el problema de la optimización de la geometría molecular, cada una con sus propios puntos fuertes y desafíos.

Enfoque del marco Ising-QUBO

Como vimos en la sección concerniente a Ising 3.1, este es un modelo clásico de mecánica estadística en el que el estado del sistema se describe mediante variables binarias (spines) que interactúan según un Hamiltoniano. También vimos que la formulación QUBO 3.2 representa los problemas de optimización como una función cuadrática de variables binarias. Ambos modelos pueden ser mapeados en annealing cuánticos o simularse utilizando algoritmos cuánticos para encontrar soluciones óptimas [21][74].

Como sabemos, en el problema que buscamos abordar, el Hamiltoniano del sistema no es un Hamiltoniano Ising que podamos implementar de manera directa en este tipo de algoritmos. Es por eso, que como primer acercamiento a la resolución del problema, definiremos una aproximación del mismo que tenga un formato QUBO e implementaremos dicha aproximación en un *annealer* con el objetivo de evaluar el resultado obtenido por este método.

Esta sección detalla la aplicación del enfoque Ising-QUBO para optimizar la geometría molecular de la molécula de hidrógeno (H_2), utilizando las bibliotecas OpenFermion/PySCF y Dwave para su implementación.

1. Paso 1: Definición del Hamiltoniano Molecular

El primer paso en la implementación del enfoque Ising-QUBO es definir el Hamilto-

niano molecular para H_2 . Este Hamiltoniano representa la energía total del sistema, incluyendo la energía cinética de los electrones, la energía potencial debida a la repulsión electrón-electrón, y la atracción electrón-nuclear.

Matemáticamente, el Hamiltoniano para una molécula en la segunda forma cuantizada se expresa como:

$$H_{\text{molecular}} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$$

donde h_{pq} y g_{pqrs} son integrales de uno y dos electrones, respectivamente, y a_p^\dagger , a_q son operadores fermiónicos de creación y aniquilación [68].

Usando las librerías OpenFermion y PySCF, ampliamente utilizadas en el campo de la química computacional, definimos la molécula de Hidrógeno y obtenemos el $H_{\text{molecular}}$.

```
# Definir la molécula en PySCF
mol = gto.Mole()
mol.build(
    atom='H 0 0 0; H 0 0 0.7414',
    basis='sto-3g',
    unit='angstrom'
)

# Calcular el Hartree-Fock con PySCF
mf = scf.RHF(mol)
mf.kernel()

# Almacenar los datos moleculares en un objeto MolecularData
molecular_data = MolecularData(
    geometry=geometry,
    basis='sto-3g',
    multiplicity=1,
    charge=0
)

# Ejecutar PySCF con OpenFermion para obtener los integrales
molecular_data = run_pyscf(molecular_data,
    run_scf=True, run_fci=True)

# Obtener el Hamiltoniano molecular en formato de fermiones
hamiltonian = molecular_data.get_molecular_hamiltonian()
```

Se construye el Hamiltoniano molecular ($H_{\text{molecular}}$), que sirve como punto de partida para el mapeo al modelo de Ising.

2. Paso 2: Mapeo Qubit y Representación del Modelo Ising

Para aplicar técnicas de computación cuántica, el Hamiltoniano molecular debe transformarse de operadores fermiónicos a operadores qubit. Esta transformación se realiza normalmente utilizando métodos como las transformaciones Jordan-Wigner o Bravyi-Kitaev [75]. Estas transformaciones convierten el Hamiltoniano fermiónico en un Hamiltoniano qubit, que es una suma de matrices de Pauli.

El Hamiltoniano qubit puede escribirse como:

$$H_{\text{qubit}} = \sum_i c_i P_i$$

donde P_i son productos tensoriales de matrices Pauli, y c_i son coeficientes derivados de las integrales en el Hamiltoniano molecular.

OpenFermion maneja automáticamente esta transformación:

```
# Transformación Jordan-Wigner para obtener Hamiltoniano qubit  
qubit_hamiltonian = qml.Hamiltonian(H.terms, qubits)
```

En el contexto del modelo de Ising, el Hamiltoniano puede interpretarse como una función de energía donde los operadores Pauli-Z corresponden a variables de spin $s_i = \pm 1$. El Hamiltoniano de Ising se representa como

$$H_{\text{Ising}} = - \sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i$$

Aquí, los coeficientes J_{ij} y h_i corresponden a los términos de interacción y campos externos en el modelo de Ising, que se relacionan con los coeficientes c_i en el Hamiltoniano del qubit [21].

3. Paso 3: Mapeo a QUBO

Una vez que el Hamiltoniano molecular se expresa como un Hamiltoniano qubit, el siguiente paso es convertirlo en una formulación QUBO. El modelo QUBO representa

el problema en términos de variables binarias $x_i \in \{0, 1\}$. El problema QUBO viene definido por la función de energía:

$$Q(x) = \sum_i a_i x_i + \sum_{i < j} b_{ij} x_i x_j$$

donde a_i y b_{ij} son coeficientes que definen los términos lineales y cuadráticos, respectivamente [76].

La correlación entre el Hamiltoniano qubit y la formulación QUBO se establece mapeando las variables de spin en el modelo de Ising a variables binarias:

$$s_i = 2x_i - 1$$

Usando este mapeo, el Hamiltoniano de Ising puede reescribirse como una matriz QUBO:

$$Q(x) = (2x - 1)^T J (2x - 1) + h^T (2x - 1)$$

El siguiente código Python ilustra cómo realizar este mapeo y construir la matriz QUBO:

```
# Función para mapear Hamiltoniano_qubit a Hamiltoniano_qubo
def qubit_hamiltonian_to_qubo(qubit_hamiltonian):
    qubo = {}
    offset = 0.0

    for term, coeff in qubit_hamiltonian.terms.items():
        # Obtener los qubits y operadores de Pauli del término
        qubits = []
        is_qubo = True

        # Iterar sobre operadores de Pauli y asegurarse de que
        solo haya términos de Z o identidad
        for qubit, pauli in term:
            if pauli == 'Z': # Solo los términos Z
                # pueden mapearse directamente a QUBO
                qubits.append(qubit)
            else:
                is_qubo = False
                break
```

```
if not is_qubo:
    # Saltamos los términos con operadores X o Y,
    # ya que no pueden mapearse directamente a QUBO
    continue

    # Asignar el término al diccionario QUBO
if len(qubits) == 0: # Término constante
    offset += coeff.real
elif len(qubits) == 1: # Término lineal (un solo qubit)
    qubit = qubits[0]
    qubo[(qubit, qubit)] = qubo.get((qubit, qubit),
                                     0) + coeff.real
elif len(qubits) == 2: # Término cuadrático
    # (interacción entre dos qubits)
    qubit1, qubit2 = qubits
    qubo[(qubit1, qubit2)] = qubo.get((qubit1, qubit2),
                                       0) + coeff.real

return qubo, offset

# Convertir a QUBO
qubo, offset = qubit_hamiltonian_to_qubo(qubit_hamiltonian)
```

Esta función convierte el Hamiltoniano qubit en una matriz QUBO, preparándolo para la optimización utilizando optimizadores cuánticos o clásicos.

4. Paso 4: Resolver el Problema QUBO

El problema QUBO está ahora listo para ser resuelto. La solución puede obtenerse utilizando el annealing cuántico o algoritmos clásicos como el annealing simulado.

Annealing Clásico El problema QUBO puede resolverse primero utilizando métodos clásicos como el annealing simulado. En este caso, utilizamos la biblioteca dimod::

```
# Creamos un BinaryQuadraticModel a partir de la matriz QUBO
bqm = dimod.BinaryQuadraticModel.from_qubo(qubo)

# Utilizamos un optimizador clásico (ej: annealing simulado)
sampler = dimod.SimulatedAnnealingSampler()
solution = sampler.sample(bqm, num_reads=1000)

# Extraemos la mejor solución
best_solution = solution.first.sample
best_energy = solution.first.energy

print(f"Mejor solución: {best_solution}")
```



```
print(f"Mejor energía: {best_energy}")
```

Annealing Cuántico: Ofrece un enfoque alternativo explotando la tunelización cuántica para encontrar el mínimo global del problema QUBO. Esto puede simularse utilizando las librerías de software Ocean de D-Wave, que proporciona una interfaz para annealing cuántico:

```
import dwave.system
from dwave.system import DWaveSampler, EmbeddingComposite

sampler = EmbeddingComposite(DWaveSampler())
solution = sampler.sample_qubo(qubo_matrix, num_reads=1000)

best_solution = solution.first.sample
best_energy = solution.first.energy

print(f"Mejor solución (quantum): {best_solution}")
print(f"Mejor energía (quantum): {best_energy}")
```

El annealing cuántico es especialmente ventajoso para los problemas a gran escala en los que los métodos clásicos pueden tener dificultades con los mínimos locales [29].

El algoritmo de optimización devuelve la configuración binaria que minimiza la función objetivo QUBO, correspondiente a la energía del estado fundamental de la molécula de hidrógeno.

Para comprobar el algoritmo propuesto iteramos sobre 20 longitudes de enlace alrededor del valor esperado de distancia inter-nuclear de la molécula de H_2 , y aplicamos el algoritmo de optimización para 3 diferentes optimizadores (exacto, annealing simulado de Dwave y annealing cuántico de Dwave). La figura 4.4 muestra la comparación de resultados (energía media vs longitud de enlace) para los tres procesos de optimización indicados.

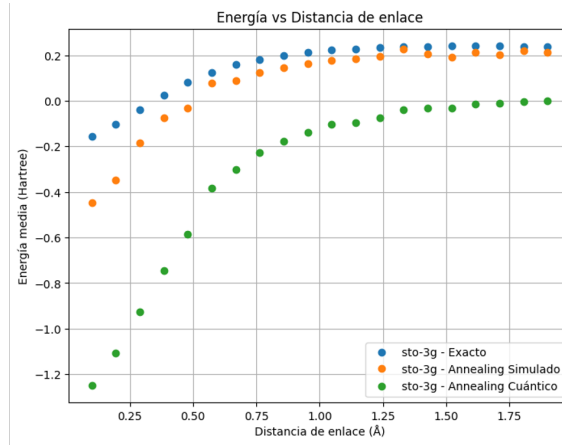


Figura 4.4: Energía Media en función de la Longitud de Enlace resultante del proceso de optimización usando formulación Ising-QUBO con annealing simulado y cuántico.

5. Paso 5: Interpretación y Análisis

El último paso consiste en interpretar la solución del problema QUBO en términos de la geometría molecular del H_2 . La solución binaria representa la configuración óptima de qubits (o spins) que minimiza la función de energía. Esta configuración se correlaciona con la estructura electrónica de la molécula de hidrógeno, proporcionando información sobre su geometría en estado básico.

La energía obtenida puede compararse con valores teóricos y resultados de otros métodos (como VQE) para evaluar la precisión y eficacia de la aproximación Ising-QUBO [77]. En las siguientes secciones abordaremos el uso de VQE para la geometría molecular.

Por el momento comparamos los resultados con optimizadores exactos y de annealing simulado, lo que se muestran en la figura 4.4. Los 3 métodos de optimización muestran conformaciones energéticas semejantes al relacionarlas con las longitudes de enlace lo que nos dice que la optimización de annealing cuántico captura lo esencial del modelo molecular, sin embargo muestra una curva un 20 % menor a la del optimizador exacto, sobre-estimando los niveles de energía de la molécula por lo que estima una longitud de enlace aproximada de 1,3Å, la curva se estabiliza después de la iteración 14 mientras que las curvas del optimizador exacto y de annealing simulado se estabilizan después de la iteración 8 equivalente a una longitud de enlace de 0.75 Å semejante al valor esperado. Para tratar de mejorar los resultados se exploró variando todos

los parámetros pero no se logró modificar la curva, lo que nos lleva a la conclusión de que puede ser una limitación inherente al optimizador de annealing cuántico al ser usado en este tipo de problemas. Para futuros ejercicios se recomienda explorar el uso de optimizadores híbridos (annealing cuántico más optimizadores clásicos).

Correlación con los modelos matemáticos Ising-QUBO

Los pasos anteriores se correlacionan directamente con la formulación matemática de los modelos Ising y QUBO:

- **Definición del Hamiltoniano:** El Hamiltoniano molecular representa la energía total del sistema físico, modelado matemáticamente como una suma de matrices de Pauli después de la transformación.
- **Mapeo Qubit:** El mapeo de operadores fermiónicos a qubits permite la representación del Hamiltoniano en términos de variables de spin, esencial para el modelo de Ising.
- **Mapeo de qubits:** La traslación de variables de spin a variables binarias tiende un puente entre la mecánica cuántica y la optimización combinatoria, permitiendo el uso del annealing cuántico para la resolución del problema.
- **Optimización:** Resolver el problema QUBO equivale a encontrar el estado fundamental del modelo de Ising, correspondiente a la configuración de energía mínima del sistema molecular.

Enfoque del Eigensolver Cuántico Variacional (VQE)

Como habíamos visto previamente, el Eigensolver Cuántico Variacional (VQE) es un algoritmo híbrido cuántico-clásico desarrollado para aproximar la energía del estado fundamental de los sistemas cuánticos. Combina los puntos fuertes de la computación cuántica y clásica para abordar problemas de química cuántica y ciencia de materiales. Basándonos sobre todo en la documentación y demos publicadas por PennyLane para la librería *qchem* [78], esta sub-sección desarrolla el enfoque VQE, cubriendo la definición del sistema molecular, la construcción del Hamiltoniano, la selección del *ansatz*¹, la optimización de parámetros y los criterios de convergencia.

¹Un *ansatz* es una conjetura bien informada sobre la forma de una solución a un problema específico. Sirve de plantilla o molde para empezar a explorar el espacio de soluciones, en lugar de empezar desde cero. El *ansatz* es esencial para la computación cuántica debido a la complejidad de los sistemas cuánticos y a la necesidad de algoritmos eficientes.

Este enfoque nos permite la implementación del Hamiltoniano molecular completo, de tal forma que los resultados del algoritmo puedan ser más precisos ya que podemos modelar nuestro problema de forma más exacta.

Utilizaremos el Hamiltoniano que caracteriza el sistema, ya que este nos proporcionará la energía de un cierto estado definido por el *ansatz*. Este proceso de optimización de la geometría del sistema se puede resumir en:

- Definición del Hamiltoniano $H(x)$ del sistema.
- Diseño del circuito variacional para generar los distintos estados electrónicos $|\psi(\theta)\rangle$.
- Definimos el valor esperado del Hamiltoniano sobre un estado $f(x, \theta) = \langle \psi(\theta) | H(x) | \psi(\theta) \rangle$ como la función de coste de nuestro problema.
- Inicializamos los valores de x y θ para optimizar ambos conjuntos de parámetros de tal forma que no solo optimicemos la función de coste definida, si no también su gradiente con respecto a las coordenadas atómicas $\nabla_x f(x, \theta) = \langle \psi(\theta) | \nabla_x H(x) | \psi(\theta) \rangle$, para encontrar el punto de equilibrio que buscamos.

Definición de la Molécula y Construcción del Hamiltoniano Molecular

La definición de la molécula de hidrógeno (H_2) ya lo abordamos en la sección anterior [1](#). Cabe mencionar que la describimos mediante dos atributos fundamentales: la longitud del enlace y la configuración electrónica. En las simulaciones cuánticas, la longitud del enlace suele variar para encontrar la geometría de equilibrio en la que se minimiza la energía del sistema.

Como ya hemos visto en secciones anteriores, el Hamiltoniano del sistema definido en las bases de Jordan-Wigner, se puede expresar en función de los distintos operadores de Pauli, de forma que sea sencillamente implementable en un computador cuántico.

En PennyLane, el Hamiltoniano para H_2 se puede construir utilizando el módulo ‘qchem’, que traduce automáticamente el Hamiltoniano molecular al Hamiltoniano de spin que necesitamos:

```
# Definimos los símbolos para la molécula de Hidrógeno
elementos = ["H", "H"]

# Establecemos las coordenadas para los dos átomos de hidrógeno
# (x, y, z) en Angstroms
coordenadas = np.array([0.0, 0.0, 0.0, # Coordenadas primer átomo
```

```
0.74, 0.0, 0.0], # Coordenadas segundo átomo
                  # (separación ~0.74 Å)
requires_grad=True)

# Definimos función para crear el Hamiltoniano de spin
def H(x):
    molecule = qml.qchem.Molecule(elementos, coordenadas, charge=0)
    return qml.qchem.molecular_hamiltonian(molecula)[0]
```

Definir el circuito cuántico para preparar el estado electrónico básico (sección del Ansatz)

Aquí describimos el segundo paso del algoritmo cuántico: definir el circuito cuántico para preparar el estado electrónico básico $|\psi(\theta)\rangle$ de la molécula de hidrógeno (H_2).

En este caso, se necesitan cuatro qubits para codificar el número de ocupación de los orbitales de spin moleculares. Para capturar los efectos de las correlaciones electrónicas, necesitamos preparar el sistema de 4 qubits en una superposición del estado Hartree-Fock $|1100\rangle$ con otros estados que difieren por una doble o única excitación. Por ejemplo, el estado $|0011\rangle$ se obtiene excitando dos partículas de los qubits 0, 1 a 2, 3. Del mismo modo, el estado $|1010\rangle$ corresponde a una única excitación desde el qubit 1 al 2. Esto puede lograrse utilizando las puertas de excitación única y doble G y $G^{(2)}$, implementadas en forma de rotaciones Givens² en PennyLane.

Además, utilizamos un algoritmo adaptativo para seleccionar las operaciones de excitación incluidas en el circuito cuántico variacional. El algoritmo procede como sigue:

1. Generar los índices de los qubits envueltos en simple y doble excitación.

Por ejemplo, los índices del estado de excitación simple $|1010\rangle$ vienen dados por la lista $[1, 2]$. Del mismo modo, los índices del estado doblemente excitado $|0011\rangle$ son $[0, 1, 2, 3]$.

2. Construir el circuito utilizando todas las puertas de doble excitación.

Calcular el gradiente de la función de coste $g(\theta, x)$ con respecto a cada puerta de doble excitación y retener sólo aquellas con gradiente distinto de cero.

²En álgebra lineal, una rotación Givens es una rotación en el plano abarcado por dos coordenadas (digamos, las coordenadas ii -ésima y jj -ésima) de un vector. Se utiliza para introducir ceros en matrices (por ejemplo, durante la descomposición QR). En PennyLane, las rotaciones de Givens se utilizan para implementar operaciones de excitación simple y doble en algoritmos cuánticos variacionales, como el Eigensolver Cuántico Variacional (VQE). Estas operaciones son cruciales para preparar estados cuánticos que representen estructuras electrónicas en moléculas.

3. Incluir las puertas de doble excitación seleccionadas y repetir el proceso para las puertas de simple excitación.
4. Construir el circuito cuántico variacional final incluyendo las puertas seleccionadas.

Para la molécula de hidrógeno (H_2) en un conjunto de bases mínimo, tenemos un total de cuatro excitaciones del estado de referencia. Tras aplicar el algoritmo adaptativo, el circuito cuántico final contiene una única operación de doble excitación que actúa sobre los qubits [0, 1, 2, 3]. El circuito se ilustra en la figura 4.3.

Para implementar este circuito cuántico, utilizamos la función `hf_state()` para generar el vector de números de ocupación que representa el estado Hartree-Fock ³.

```
# Definimos número electrones y orbitales para molécula de H_2
electrons = 2 # H_2 tiene 2 electrones
orbitals = 4 # H_2 ocupará 2 orbitales moleculares
              # (1 para cada átomo de hidrógeno)

# Generamos el estado Hartree-Fock para la molécula de H_2
hf = qml.qchem.hf_state(electrons=electrons, orbitals=orbitals)
```

La matriz `hf` es utilizada por la operación `BasisState` para inicializar el registro de qubits. A continuación, se aplican primero las operaciones `DoubleExcitation`, definimos el dispositivo cuántico utilizado para calcular el valor esperado. En este ejemplo, utilizamos el simulador `lightning.qubit`:

```
# Definimos número de qubits (wires) necesarios para H_2
num_wires = 4 # 2 orbitales requieren 4 qubits
              # (2 qubits por orbital)

# Definimos el dispositivo cuántico para calcular valor esperado
dev = qml.device("lightning.qubit", wires=num_wires)

# Definimos circuito cuántico para preparar el estado básico para H_2
@qml.qnode(dev, interface="autograd")
def circuit(params, obs, wires):
    # Inicializamos el circuito con el estado Hartree-Fock
    qml.BasisState(hf, wires=wires)
```

³El estado Hartree-Fock es una aproximación utilizada para describir la estructura electrónica de átomos y moléculas. Se basa en la idea de que los electrones pueden representarse como productos de funciones de onda individuales, denominadas orbitales. El método Hartree-Fock consiste en minimizar la energía total del sistema variando los orbitales. Aunque tiene limitaciones, el estado Hartree-Fock es una aproximación útil y un punto de partida para métodos más avanzados en química cuántica.

```
# Aplicamos una excitación simple para H2  
# (solo una doble excitación es necesaria)  
qml.DoubleExcitation(params[0], wires=[0, 1, 2, 3])  
  
return qml.expval(obs)
```

Este circuito prepara el estado de prueba (*ansatz*). Si sólo se utiliza una excitación doble para la molécula de hidrógeno (H_2), la fórmula se simplifica ya que no es necesario considerar las excitaciones simples. El estado cuántico variacional $|\psi(\theta)\rangle$ sería:

$$|\psi(\theta)\rangle = \cos(\theta)|1100\rangle - \sin(\theta)|0011\rangle$$

En esta expresión:

- $|1100\rangle$ es el estado de referencia Hartree-Fock.
- $|0011\rangle$ representa la doble excitación donde dos electrones son excitados desde los dos primeros qubits (0 y 1) a los dos últimos qubits (2 y 3).

El parámetro θ controla la superposición entre el estado de referencia y el estado de doble excitación, teniendo en cuenta las correlaciones electrónicas dentro de la molécula de hidrógeno.

Definir la función de coste $g(\theta, x)$.

Este paso consiste en definir la función de coste

$$g(\theta, x) = \langle \psi(\theta) | H(x) | \psi(\theta) \rangle$$

Evalúa el valor esperado del Hamiltoniano parametrizado $H(x)$ en el estado de prueba $|\psi(\theta)\rangle$.

A continuación, definimos la función de coste $g(\theta, x)$ que depende tanto del circuito como de los parámetros del Hamiltoniano. En concreto, consideramos los valores esperados del Hamiltoniano.

```
# Definimos el la función de coste  
def cost(params, x):  
    hamiltonian = H(x)  
    return circuit(params, obs=hamiltonian, wires=range(num_wires))
```

La función de coste es minimizada al calcular los gradientes con respecto tanto a los parámetros del circuito θ (cálculo analítico con técnicas de diferenciación automática de PennyLane) como a las coordenadas nucleares x (tomando el valor esperado del Hamiltoniano electrónico).

Utilizamos la función `finite_diff()` para calcular el gradiente del Hamiltoniano utilizando una aproximación por diferencia central. A continuación, evaluamos el valor esperado de los componentes del gradiente $\frac{\partial H(x)}{\partial x_i}$. Esto se implementa mediante la función `grad_x()`:

```
# Definimos la función finite_diff()
def finite_diff(f, x, delta=0.01):
    # Calcular diferencia finita por diferencia central de una función.
    gradient = []

    for i in range(len(x)):
        shift = np.zeros_like(x)
        shift[i] += 0.5 * delta
        res = (f(x + shift) - f(x - shift)) * delta**-1
        gradient.append(res)

    return gradient

# Definimos la función grad_x()
def grad_x(params, x):
    # Calcular el gradiente del valor esperado con respecto
    # a las coordenadas nucleares x.
    grad_h = finite_diff(H, x)
    grad = [circuit(params, obs=obs, wires=range(num_wires))
            for obs in grad_h]
    return np.array(grad)
```

Optimización de la Geometría Molecular

Ahora procedemos a minimizar la función de coste para encontrar la geometría de equilibrio del estado fundamental de la molécula H_2 .

Comenzamos definiendo los optimizadores clásicos, inicializamos los parámetros del circuito θ . El ángulos θ se ponen a cero para que el estado inicial $|\psi(\theta)\rangle$ sea el estado Hartree-Fock.

El conjunto inicial de coordenadas nucleares x , definido al principio, se basa en que la distancia de 0.74 Å es una longitud de enlace típica para la molécula de H_2 . Esta es una elección natural para la geometría de partida que pretendemos mejorar debido a los efectos de correlación electrónica incluidos en el estado de prueba $|\psi(\theta)\rangle$.

Realizamos la optimización sobre un máximo de 100 pasos. Los parámetros del circuito y las coordenadas nucleares se optimizan hasta que el componente máximo del gradiente nuclear $\nabla_x g(\theta, x)$ es menor o igual a 10^{-5} Hartree/Bohr ⁴. Normalmente, este es el criterio de convergencia utilizado para optimizar geometrías moleculares en simulaciones de química cuántica.

```
# Inicialización Parámetro inicial puerta de excitación
theta = np.array([0.0], requires_grad=True)
# Coordenadas iniciales del primer y segundo átomo de H
x = np.array([0.0, 0.0, 0.0,
              0.74, 0.0, 0.0], requires_grad=True)

# Definimos optimizadores clásicos para los parámetros
# y para las coordenadas nucleares
opt_theta = qml.GradientDescentOptimizer(stepsize=0.4)
opt_x = qml.GradientDescentOptimizer(stepsize=0.8)

# Almacenamos los valores la función de coste y
# de la longitud de enlace
energy = []
bond_length = []

# Factor para convertir de Bohrs a Angstroms
bohr_angs = 0.529177210903

# Definimos el Bucle de Optimización
for n in range(100):

    # Optimizamos los parámetros del circuito
    theta.requires_grad = True
    x.requires_grad = False
    theta, _ = opt_theta.step(cost, theta, x)

    # Optimizamos las coordenadas nucleares
    x.requires_grad = True
    theta.requires_grad = False
    _, x = opt_x.step(cost, theta, x, grad_fn=grad_x)

    # Almacenamos los valores de energía y longitud de enlace
    energy.append(cost(theta, x))
    bond_length.append(np.linalg.norm(x[0:3] -
                                       x[3:6]) * bohr_angs)

    # Obtenemos progreso de resultados cada 4 pasos
    if n % 4 == 0:
```

⁴Hartree/Bohr es una unidad de fuerza en unidades atómicas, un Hartree equivale aproximadamente a 27,211 electronvoltios (eV) y un Bohr es aproximadamente igual a 0.52918 Angstroms.

```
print(f"Paso = {n}, E = {energy[-1]:.8f} Ha,  
      longitud de enlace = {bond_length[-1]:.5f} A")  
  
# Comprobamos el componente máximo del gradiente nuclear  
if np.max(grad_x(theta, x)) <= 1e-05:  
    break  
  
# Obtenemos el resultado final  
print("\n" f"Valor final de la energía del estado fundamental =  
      {energy[-1]:.8f} Ha")  
print("\n" "Geometría de equilibrio del estado fundamental")  
print("%s %4s %8s %8s" % ("symbol", "x", "y", "z"))  
symbols = ["H", "H"]  
for i, atom in enumerate(symbols):  
    print(f" {atom}      {x[3 * i]:.4f}   {x[3 * i + 1]:.4f}  
          {x[3 * i + 2]:.4f}")
```

A continuación se muestran los resultados obtenidos por el proceso de Optimización.

RESULTADOS DEL PROCESO DE OPTIMIZACION

```
Paso = 0, E = -0.96759639 Ha, longitud de enlace = 1.36072 A  
Paso = 4, E = -1.12497203 Ha, longitud de enlace = 0.87252 A  
Paso = 8, E = -1.13708572 Ha, longitud de enlace = 0.74090 A  
Paso = 12, E = -1.13729276 Ha, longitud de enlace = 0.73607 A  
Paso = 16, E = -1.13730528 Ha, longitud de enlace = 0.73516 A
```

```
Valor final de la energía del estado fundamental = -1.13730596 Ha
```

Geometría de equilibrio del estado fundamental

symbol	x	y	z
H	-0.3244	0.0000	0.0000
H	1.0644	0.0000	0.0000

Figura 4.5: Resultados Proceso de Optimización Geometría Molecular H_2 .

Convergencia y terminación

La convergencia se evalúa monitorizando el cambio en los valores de energía durante la optimización. Los criterios de convergencia suelen incluir:

- **Umbral de cambio de energía:** El proceso de optimización se considera convergente si el cambio de energía entre iteraciones cae por debajo de un umbral predefinido.
- **Número de iteraciones:** A menudo se fija un número máximo de iteraciones para evitar un cálculo excesivo.

Analizar la convergencia implica trazar los valores de energía a lo largo de las iteraciones y examinar si el algoritmo alcanza una solución estable [72][79].

A continuación, trazamos los valores de la energía de estado fundamental de la molécula y la longitud de enlace en función del paso de optimización.

```
# Valores finales de energía y longitud de enlace para H2
# (debe sustituirse por los valores de referencia)
E_fci = -1.137 # Energía final de FCI para H2 (Hartree)
d_fci = 0.741 # Longitud de enlace de equilibrio para H2 (Å)

# Creamos la gráfica
fig = plt.figure()
fig.set_figheight(5)
fig.set_figwidth(12)

# Agregamos la gráfica de energía a la primera columna
E_vqe = np.array(energy) # Valores de Energía en optimización VQE
ax1 = fig.add_subplot(121)
# Trazamos la diferencia de energía
ax1.plot(range(len(E_vqe)), E_vqe - E_fci, "go", ls="dashed")
# Línea de exactitud química
ax1.plot(range(len(E_vqe)), np.full(len(E_vqe), 0.001),
        color="red")
ax1.set_xlabel("Paso de optimización", fontsize=13)
ax1.set_ylabel("$E_{VQE} - E_{FCI}$ (Hartree)", fontsize=13)
ax1.text(5, 0.0013, r"Exactitud química", fontsize=13)
plt.yscale("log")
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

# Agregamos gráfica de longitud de enlace a la segunda columna
ax2 = fig.add_subplot(122)
# gráfica de longitud de enlace
ax2.plot(range(len(bond_length)), bond_length, "go",
        ls="dashed")
# Línea de longitud de enlace de equilibrio
ax2.plot(range(len(bond_length)), np.full(len(bond_length),
        d_fci), color="red")
ax2.set_ylim([0.7, 0.8]) # Límites y para la longitud de enlace
ax2.set_xlabel("Paso de optimización", fontsize=13)
ax2.set_ylabel("Longitud de enlace ($AA$)", fontsize=13)
ax2.text(10, d_fci + 0.001, r"Longitud de enlace en equilibrio",
        fontsize=13, ha='center')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

plt.subplots_adjust(wspace=0.3)
plt.show()
```

A continuación, se muestra la gráfica obtenida de los valores de la energía de estado fundamental de la molécula en función del paso de optimización.

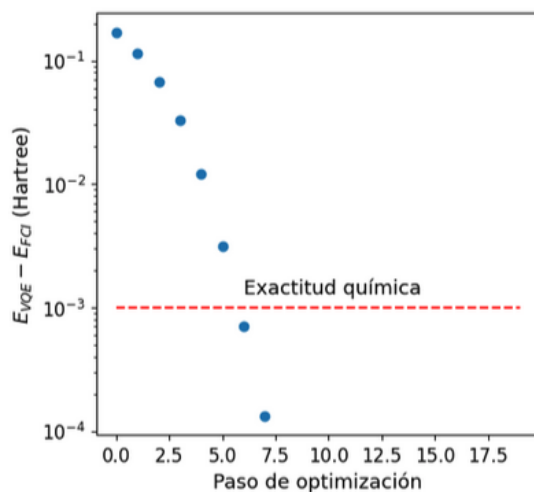


Figura 4.6: Energía en función del paso de optimización.

Ahora se muestra la gráfica de los valores de la longitud de enlace en función del paso de optimización.

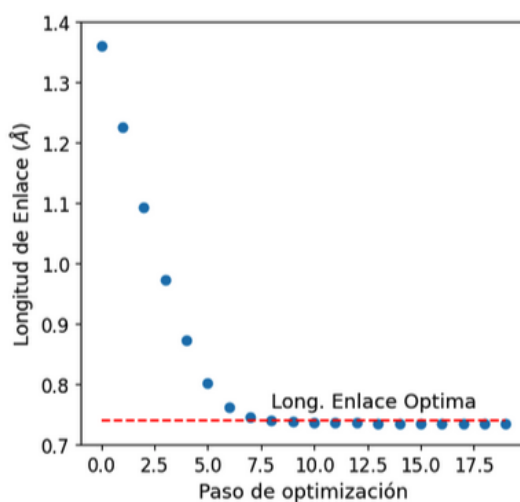


Figura 4.7: Longitud de Enlace en función del paso de optimización.

Se puede apreciar que la energía del estado fundamental converge dentro del límite de exactitud química ($0.0016 Ha$) en la séptima iteración, necesitándose unas pocas más iteraciones para encontrar la longitud de enlace de equilibrio para la molécula H_2 , cuyo valor se aproxima mucho al valor esperado de 0.74 Angstroms .

4.3.4. Configuración Electrónica y Energía

Debido a los resultados obtenidos a la hora de optimizar la geometría molecular del H_2 , en esta sección nos enfocaremos en como podemos optimizar la configuración electrónica de tal forma que la energía del sistema sea mínima utilizando el algoritmo VQE. En este contexto es posible buscar la energía mínima global del sistema, conocida como energía del estado fundamental, o la energía mínima bajo ciertas condiciones, por ejemplo, la energía mínima del estado de spin $S = 1$.

En este trabajo resolveremos la energía y configuración electrónica del estado fundamental del H_2 , es decir, la energía mínima del sistema y la configuración electrónica que da lugar a dicho estado.

Definición del *ansatz*

A la hora de abordar este problema lo primero que debemos hacer es definir el *ansatz* que utilizaremos. En la sección 4.2 hemos introducido el *ansatz* que utilizaremos

$$|\Psi(\theta)\rangle = c_{HF}(\theta)|1100\rangle + c_1(\theta)|0011\rangle + c_2(\theta)|0110\rangle + c_3(\theta)|1001\rangle,$$

En este punto podemos introducir como generar este *ansatz* en PennyLane. Para ello haremos uso de la función `qml.qchem.hf_state()` que nos permitirá generar nuestro estado de Hartree-Fock.

```
# Definimos el estado base de Hartree-Fock
hf = qml.qchem.hf_state(electrones, qubits)

# Imprimimos el estado base de Hartree-Fock
print(hf)
```

Una vez hemos generado nuestro estado base, haremos uso de `qml.qchem.excitations()` para definir las excitaciones que le queremos aplicar. En nuestro caso aplicaremos todas las excitaciones posibles de uno y dos electrones.

```
# Definimos las excitaciones de los electrones
singles, doubles = qml.qchem.excitations(electrones, qubits, delta_sz=0)

# Imprimimos las excitaciones de los electrones
print(singles)
print(doubles)
```

Podemos observar que a la hora de definir las excitaciones de nuestro sistema electróni-

co, debemos definir también si estas excitaciones llevan consigo un cambio en el spin del sistema. En este caso, como estamos buscando el estado fundamental, queremos que el spin se mantenga en cero, por lo que el parámetro `delta.sz` será 0. En el caso que quisiéramos buscar el estado de mínima energía en el que el sistema tiene spin $S = 1$, podemos modificar el parámetro a `delta.sz = 1`.

Debemos ahora crear el circuito que implemente estas excitaciones del sistema en base a rotaciones de Givens. Esto se puede hacer de manera directa con el operador `qml.AllSinglesDoubles()`. Esta función excitará un estado de Hartee-Fock dado según una serie de excitaciones dadas y los ángulos asociados a dichas rotaciones. En nuestro caso, estos ángulos serán un parámetro del circuito que se optimizará para obtener el estado fundamental.

```
# Implementamos un circicuito que implemente
# las excitaciones de los electrones
def circuit(params, wires):
    qml.AllSinglesDoubles(params, wires, hf, singles, doubles)
```

Una vez creado nuestro *ansatz* debemos configurar el proceso de optimización que nos permitirá buscar los parámetros que mejor aproximen el estado fundamental de la molécula.

Configuración del proceso de optimización

En este punto lo primero que debemos crear son los observables que buscamos evaluar en nuestro problema. El primero de ellos, que sería el Hamiltoniano, ya lo hemos construido en el primer paso. El valor esperado del Hamiltoniano actuará como la función de coste de nuestro problema, por lo que implementaremos un circuito que evalúe su valor esperado en función del valor de los parámetros del *ansatz*. Para ello utilizaremos la función `qml.exval()` que devuelve el valor esperado de un operador dado un circuito.

```
# Definimos la función de coste
@qml.qnode(dev, interface="autograd")
def cost_fn(params):
    circuit(params, wires=range(qubits))
    return qml.expval(H)
```

Debemos crear también el operador de spin del sistema S^2 ya que puede ser interesante conocer el estado de spin del estado fundamental.

```
# Definimos el operador spin total S2
S2 = qml.qchem.spin2(electrones, qubits)
```

Indicaremos el número de electrones de nuestro sistema y haremos uso de la función `qml.qchem.spin2()` para definir dicho operador. Una vez hecho esto, calcularemos su valor esperado de forma análoga al caso del Hamiltoniano.

```
# Definimos el circuito para el valor esperado
# de S2
@qml.qnode(dev, interface="autograd")
def S2_exp_value(params):
    circuit(params, wires=range(qubits))
    return qml.expval(S2)
```

En este punto debemos recordar como funciona el valor esperado del operador S^2 (4.9). Por lo que si queremos obtener el valor del spin del sistema este será

$$s = -\frac{1}{2} + \sqrt{\frac{1}{4} \langle S^2 \rangle}$$

Que en Python se implementa de forma sencilla como

```
# Calculamos le valor del Spin total
def spin_total(params):
    return -0.5 + np.sqrt(1 / 4 + S2_exp_value(params))
```

Tenemos ya definida la función de coste y el valor del spin del sistema. El siguiente paso será configurar el optimizador que nos permitirá actualizar los valores de los parámetros θ del circuito de tal forma que seamos capaces de encontrar el estado fundamental del sistema.

Para esto utilizaremos el conocido algoritmo *Gradient Descent* [80]. Este algoritmo actualizará los parámetros θ en la dirección de descenso del gradiente de la función de coste.

El valor inicial de los parámetros θ_0 se tomará de manera aleatoria en base a una distribución normal cuyos valores serán $\theta_0 \in [0, \pi]$.

```
# Definimos el optimizador
opt = qml.GradientDescentOptimizer(stepsize=0.35)
np.random.seed(0)

# Definimos el valor inicial de los parámetros del circuito
theta = np.random.normal(0, np.pi, len(singles) + len(doubles)\
                           ,requires_grad=True)

print(theta)
```

Ejecución y evaluación de resultados

En este punto tenemos todo listo para ejecutar nuestro proceso de optimización. Únicamente debemos ejecutar en bucle el proceso de actualización de parámetros hasta que se cumpla un criterio de parada, como por ejemplo un número de iteraciones o que la mejora por cada nuevo paso sea menor a un umbral.

```
max_iter = 100
stop_criterio = 1e-05
energy_results = []
spin_results = []

# Bucle de optimización
for n in range(max_iter):

    # Calculamos el valor de los parámetros y el gradiente
    # de la energía
    theta, prev_energy = opt.step_and_cost(cost_fn, theta)

    # Calculamos el valor de la energía y el spin para los
    # parámetros actuales
    energy = cost_fn(theta)
    spin = spin_total(theta)

    # Calculamos la convergencia como la diferencia entre
    # la energía actual y la anterior
    conv = np.abs(energy - prev_energy)

    # Imprimimos los resultados cada 4 iteraciones
    if n % 4 == 0:
        print(f"Iteración = {n}, Energía = {energy:.8f} Ha\
              , f"Spin = {spin:.5f}")

    # Comprobamos si se ha alcanzado el criterio de parada
    if conv <= stop_criterio:
        print(f"\n" f"Convergencia alcanzada en la iteración {n}")
        break

    # Guardamos los resultados
    energy_results.append(energy)
    spin_results.append(spin)

# Imprimimos los resultados finales
print("\n" f"Energía del estado fundamental = {energy:.5f} Ha")
print("\n" f"Aproximación al spin del estado fundamental = {spin:.5f}")
print("\n" f"Parámetros óptimos del circuito = {theta}")
```

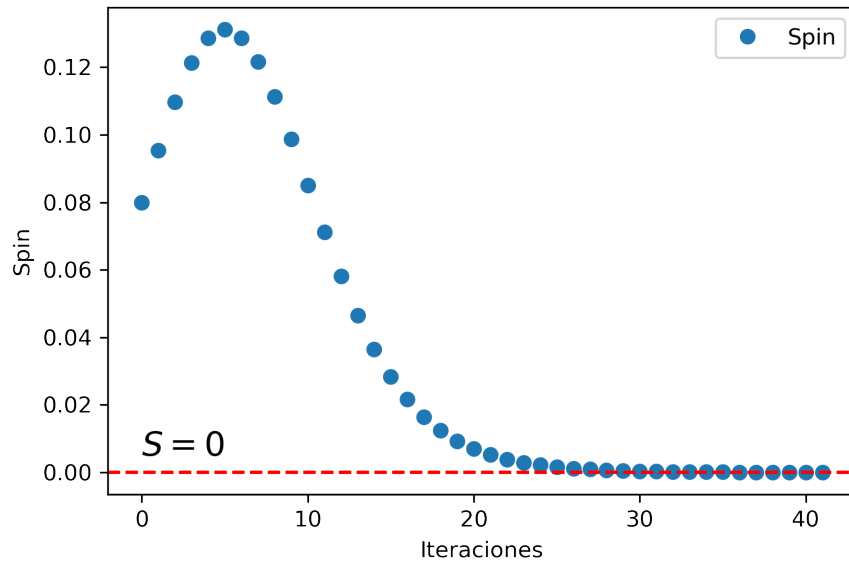



Figura 4.8: Valor del spin del sistema en cada iteración del optimizador

Podemos representar gráficamente los resultados obtenidos en cada iteración para el valor de la energía del estado fundamental y del spin. Dado que estamos trabajando con un caso muy sencillo, la molécula de H_2 podemos anticipar que el estado fundamental será el estado fundamental de Hartee-Fock

$$|\psi\rangle = |1100\rangle$$

donde ambos electrones se encuentran en el orbital de mínima energía. Por esto esperamos que después del proceso de optimización, el sistema vuelva a dicho estado y por lo tanto, tenga $s = 0$.

Podemos ver en la Fig.4.8 que se recupera el valor de $s = 0$ como cabría esperar. En cuanto al estado del sistema, podemos construirlo en base a las rotaciones de Givens y el valor final de los parámetros θ .

```
# Construimos el estado resultado del proceso de optimización
@qml.qnode(dev)
def optimized_state(params, wires):
    qml.AllSinglesDoubles(params, wires, hf, singles, doubles)
    return qml.state()

# Recuperamos el estado óptimo
estado_final = [np.round(x, 3)
                 for x in optimized_state(theta, wires=range(qubits))]
```

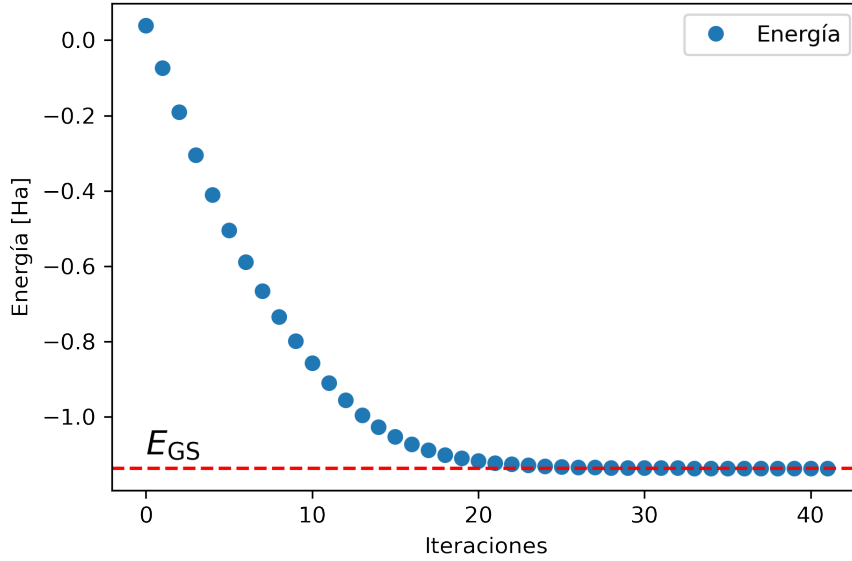


Figura 4.9: Valor esperado de la energía de la molécula de H_2 en función de las iteraciones

```
print(estado_final)
```

Obteniendo así el estado

$$|\psi\rangle = -0,995 |1100\rangle + 0,105 |0011\rangle + 0,001 |0110\rangle$$

Que es una muy buena aproximación al estado fundamental real, ya que nos permite recuperar el estado correcto con un 99 % de probabilidad.

En cuanto al valor de la energía, podemos compararlo con la energía calculada a través de métodos como el FCI (*Full Configuration Interaction*)[81], que nos permite aproximar la energía del estado fundamental a $E_{FCI} \approx -1,13619Ha$.

Podemos ver en la Fig. 4.9 que nuestra aproximación al valor de energía del estado fundamental es muy buena también, con una desviación δ cercana al 0 % ($\delta \approx 0,003\%$)

4.4. Resultados

Pese a que el estudio de la molécula de H_2 es un caso muy sencillo de simulación de un sistema molecular, los resultados obtenidos ilustran la utilidad de este tipo de técnicas para la caracterización y simulación de sistemas moleculares.

La solución proporcionada por el annealing cuántico sobredimensionó el valor esperado

de la longitud de enlace en un 75 %, sesgo que atribuimos a limitaciones del annealing cuántico que parece sobredimensionar el nivel de energía de las moléculas cuando trata problemas de química computacional.

Cuando abordamos el problema de la geometría usando aproximaciones VQE se pudo ver que la energía del estado fundamental converge dentro del límite de exactitud química ($0,0016Ha$) en la séptima iteración, necesitándose unas pocas más iteraciones para encontrar la longitud de enlace de equilibrio para la molécula H_2 , cuyo valor se aproxima mucho al valor esperado de $0,74Angstroms$.

En el caso de la estimación del estado fundamental de la molécula y su energía, hemos obtenido resultados muy buenos usando VQE, con errores δ cercanos a 0 % ($\delta \approx 0,003\%$) para el caso de la energía y un resultado perfecto para el caso del valor del del spin del sistema.

En cuanto al estado fundamental como tal, usando VQE hemos conseguido aproximarlos a un estado $|\psi\rangle$ que devolverá el estado fundamental teórico con un 99 % de probabilidad, lo cual en primera instancia, es una muy buena aproximación.

5. Conclusiones y Trabajo Futuro

En este trabajo hemos introducido los conceptos generales necesarios para abordar problemas de optimización combinatoria con computación cuántica, como el marco Ising-QUBO y los fundamentos en los que se basa la computación cuántica.

Más adelante, hemos aplicado estos conceptos a la modelización de sistemas moleculares, tomando como referencia la molécula de H_2 . De esta forma hemos conseguido utilizar métodos cuánticos, como el algoritmo VQE y también la formulación Ising-QUBO con annealing simulado y cuántico, para aproximar su geometría, su configuración electrónica y energía del estado fundamental, con un nivel de precisión muy alto.

Esta implementación demuestra la aplicación práctica de modelos teóricos en química cuántica y optimización, ilustrando cómo conceptos matemáticos abstractos pueden traducirse en algoritmos computacionales utilizando librerías de software cuántico como PennyLane.

El rendimiento y la convergencia del enfoque QUBO dependen de factores como la calidad de la formulación QUBO y las capacidades del annealing cuántico. La adecuada formulación molecular para conseguir distribuciones realistas de energía y la precisión de la solución con annealer real permiten ver la eficacia de aplicar modelación Ising-QUBO con annealing cuántico y también considerar que sí es relevante su uso en el desarrollo actual de la química computacional cuántica.

El marco Ising-QUBO proporciona una valiosa y novel alternativa a métodos como VQE aprovechando los puntos fuertes de los annealing cuánticos para problemas de optimización binaria a gran escala. Mientras que VQE es más adaptable a varios ansatz y métodos de optimización, el enfoque Ising-QUBO sobresale en problemas que pueden ser directamente mapeados a problemas de optimización binarios [82][21].

La aplicación de modelación Ising-QUBO con annealing cuántico está en su todavía en etapa temprana y ya se han propuesto algoritmos con este enfoque como el Quantum Annealer Eigensolver (QAE). La investigación futura en el marco Ising-QUBO podría centrarse en mejorar las técnicas de formulación QUBO e integrarlas con algoritmos cuánticos avanzados. Además, la exploración de enfoques híbridos que combinen el annealing cuántico con métodos de optimización clásicos podría mejorar el rendimiento y la aplicabilidad del enfoque Ising-QUBO [51][82].

Este ejercicio sienta las bases para introducir la ventaja que esta nueva tecnología

(computación cuántica) puede proporcionar para la resolución eficiente de ciertos problemas que, hoy en día, no se pueden ejecutar de manera eficiente en computadores clásicos. Además, en este trabajo hemos presentado de manera clara las herramientas disponibles para el desarrollo de software cuántico enfocado a problemas de química cuántica.

Este trabajo busca sentar unas bases que permitan al lector introducirse en este ámbito y dotarle de un entendimiento suficiente para que pueda seguir profundizando en él. Dentro de la química cuántica y la simulación de sistemas moleculares, existen muchos casos de uso más complejos que son de gran interés, como la simulación de reacciones químicas, la modelización de nuevas proteínas o materiales y el cálculo de propiedades moleculares de moléculas complejas. Estos problemas se escapan del alcance de este trabajo pero se proponen como nuevas líneas de trabajo futuro que profundicen más aún en la utilidad de la computación cuántica en la resolución de problemas complejos de química cuántica.

Bibliografía

- [1] FEYNMAN, RICHARD P., *The Character of Physical Law*. MIT Press, 1965. URL: <https://mitpress.mit.edu/9780262560030/the-character-of-physical-law/>
- [2] FEYNMAN, RICHARD P., *Simulating physics with computers*, *International Journal of Theoretical Physics*, vol. 21, no. 6-7, pp. 467–488, 1982. Springer. doi:10.1007/BF02650179. <https://doi.org/10.1007/BF02650179>
- [3] KARP, R. M., *Reducibility among combinatorial problems*, in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Springer, 1972, pp. 85–103. doi:10.1007/978-1-4684-2001-2_9. https://doi.org/10.1007/978-1-4684-2001-2_9
- [4] SHOR, PETER W., *Algorithms for quantum computation: Discrete logarithms and factoring*, in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, IEEE, 1994, pp. 124–134. doi:10.1109/SFCS.1994.365700. <https://doi.org/10.1109/SFCS.1994.365700>
- [5] NIELSEN, MICHAEL A. AND CHUANG, ISAAC L., *Quantum Computation and Quantum Information*, 10th ed. Cambridge University Press, 2010, pp. 60. isbn:978-0521635035. <https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE#overview>
- [6] DEUTSCH, D., JOZSA, R., *Rapid Solution of Problems by Quantum Computation*, *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, **439**(1907), 553–558, 1992. doi:10.1098/rspa.1992.0167. <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167>
- [7] FARHI, E., GOLDSTONE, J., & GUTMANN, S. (2000). *Quantum computation by adiabatic evolution*. arXiv:quant-ph/0001106. <https://doi.org/10.48550/arXiv.quant-ph/0001106>
- [8] HENDRICKSON, B., *The Molecule Problem: Exploiting Structure in Global Optimization*, *SIAM Journal on Optimization*, vol. 5, no. 4, pp. 835–857, 1995. doi: <https://doi.org/10.1137/0805040>.

- [9] ISING, ERNST, *Beitrag zur Theorie des Ferromagnetismus*, Zeitschrift für Physik, vol. 31, no. 1, 1925, pp. 253-258, <https://doi.org/10.1007/BF02980577>.
- [10] ONSAGER, LARS, *Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition*, *Physical Review*, vol. 65, no. 3-4, pp. 117-149, 1944. doi:10.1103/PhysRev.65.117. <https://journals.aps.org/pr/abstract/10.1103/PhysRev.65.117>
- [11] HOPFIELD, J. J., *Neural networks and physical systems with emergent collective computational abilities*, *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, 1982, pp. 2554-2558. doi:10.1073/pnas.79.8.2554. <https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554>.
- [12] COMMANDER, C.W., *Maximum cut problem, MAX-cut*, *Encyclopedia of Optimization*, vol. 2, 2009. doi:10.1007/978-0-387-74759-0_358 http://dx.doi.org/10.1007/978-0-387-74759-0_358
- [13] REISINGER, D., ADAM, R., KOGLER, M., FÜLLSACK, M. & JÄGER, G., *Critical transitions in degree mixed networks: A discovery of forbidden tipping regions in networked spin systems*, *PLOS ONE*, vol. 17, pp. e0277347, 2022. DOI: [10.1371/journal.pone.0277347](https://doi.org/10.1371/journal.pone.0277347).
- [14] DIRAC, P. A. M., *Hamiltonian Methods and Quantum Mechanics*, *Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*, vol. 63, pp. 49-59, 1963. <http://www.jstor.org/stable/20488626>.
- [15] SIMPSON, K., L'HOMME, A., KEYMER, J., ET AL., *Spatial biology of Ising-like synthetic genetic networks*, *BMC Biol*, 21, 185 (2023). <https://doi.org/10.1186/s12915-023-01681-4>
- [16] SCHINCKUS, CHRISTOPHE, *Ising model, econophysics and analogies*, *Physica A: Statistical Mechanics and its Applications*, 508, 95-103 (2018). <https://doi.org/10.1016/j.physa.2018.05.063>
- [17] HURTADO-MARÍN, V. A., AGUDELO-GIRALDO, J. D., ROBLEDO, S., ET AL., *Analysis of dynamic networks based on the Ising model for the case of study of co-authorship of scientific articles*, *Sci Rep*, 11, 5721 (2021). <https://doi.org/10.1038/s41598-021-85041-8>

- [18] IMANAKA, Y., ANAZAWA, T., KUMASAKA, F., ET AL., *Optimization of the composition in a composite material for microelectronics application using the Ising model*, Sci Rep, 11, 3057 (2021). <https://doi.org/10.1038/s41598-021-81243-2>
- [19] HONG, YIFAN, YOUNG, JEREMY T., KAUFMAN, ADAM M., & LUCAS, ANDREW, *Quantum error correction in a time-dependent transverse-field Ising model*, Phys. Rev. A, 106(2), 022432 (2022). <https://doi.org/10.1103/PhysRevA.106.022432>
- [20] GLOVER, F. (1989). TABU SEARCH—PART I. *ORSA Journal on Computing*, 1(3), 190–206. <https://doi.org/10.1287/ijoc.1.3.190>
- [21] LUCAS, A. (2014). *Ising formulations of many NP problems*. Frontiers in Physics, 2, 5. <https://doi.org/10.3389/fphy.2014.00005>
- [22] GLOVER, F., KOCHENBERGER, G., & DU, Y. (2019). *A Tutorial on Formulating and Using QUBO Models*. arXiv preprint arXiv:1811.11538. <https://doi.org/10.48550/arXiv.1811.11538>
- [23] BARAHONA, F. (1982). *On the computational complexity of Ising spin glass models*. Journal of Physics A: Mathematical and General, 15 (10), 3241-3251. <https://dx.doi.org/10.1088/0305-4470/15/10/028>
- [24] KOCHENBERGER, G., GLOVER, F., ALIDAEI, B., & REGO, C. (2004). *A unified modeling and solution framework for combinatorial optimization problems*. OR Spectrum, 26(3), 237-250. <https://link.springer.com/article/10.1007/s00291-003-0153-3>
- [25] AU-YEUNG, R., CHANCELLOR, N. & HALFFMANN, P., *NP-hard but no longer hard to solve? Using quantum computing to tackle optimization problems*, Frontiers in Quantum Science and Technology, vol. 2, 2023. <https://www.frontiersin.org/journals/quantum-science-and-technology/articles/10.3389/frqst.2023.1128576>.
- [26] CHATTERJEE, Y., BOURREAU, E. & RANČIĆ, M. J., *Solving various NP-hard problems using exponentially fewer qubits on a quantum computer*, Phys. Rev. A, vol. 109, no. 5, 052441, 2024. <https://link.aps.org/doi/10.1103/PhysRevA.109.052441>.
- [27] JASCHKE, D. & MONTANGERO, S., *Is quantum computing green? An estimate for an energy-efficiency quantum advantage*, Quantum Science and Technology, vol. 8, no. 2, 025001, 2023. <https://dx.doi.org/10.1088/2058-9565/acae3e>.

- [28] NISHIMORI, H. (2001). *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press. <https://academic.oup.com/book/5185>
- [29] KADOWAKI, T., & NISHIMORI, H. (1998). *Quantum annealing of ising spin glass models*. Physical Review E, 58(5), 5355–5363. <https://doi.org/10.1103/PhysRevE.58.5355>
- [30] PERUZZO, A., MCCLEAN, J., SHADBOLT, P., YUNG, M.H., ZHOU, X.Q., LOVE, P.J., ASPURU-GUZI, A. y O'BRIEN, J.L., *A variational eigenvalue solver on a photonic quantum processor*, Nature Communications, vol. 5, p. 4213, 2014. doi: 10.1038/ncomms5213. PMID: 25055053; PMCID: PMC4124861. <http://dx.doi.org/10.1038/ncomms5213>
- [31] ALONSO-LINAJE, G. y ATCHADE-ADELOMOU, P., *EVA: a quantum Exponential Value Approximation algorithm*, arXiv preprint, arXiv:2106.08731, 2021. Disponible en: <https://arxiv.org/abs/2106.08731>.
- [32] SEFFERS, G. I., *DARPA Hopes to Help Settle the Quantum Computing Wild West [Online]*, AFCEA International, 2024. <https://www.afcea.org/signal-media/cyber-edge/darpa-hopes-help-settle-quantum-computing-wild-west>
- [33] DEUTSCH, D., *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 400(1818), 97-117, 1985. <https://royalsocietypublishing.org/doi/10.1098/rspa.1985.0070>
- [34] BENNETT, C. H., & BRASSARD, G., *Quantum cryptography: Public key distribution and coin tossing*, Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, 175-179, 1984. <https://www.sciencedirect.com/science/article/pii/S0304397514004241>
- [35] LAUGHLIN, R. B. (1981). *Quantized Hall conductivity in two dimensions*. Phys. Rev. B, 23(10), 5632–5633. <https://journals.aps.org/prb/abstract/10.1103/PhysRevB.23.5632>
- [36] SCHUMACHER, B. (1995). *Quantum coding*. Physical Review A, 51(4), 2738–2747. <https://doi.org/10.1103/PhysRevA.51.2738>

- [37] DIRAC PAM. *A new notation for quantum mechanics*. Mathematical Proceedings of the Cambridge Philosophical Society. 1939;35(3):416-418. doi:10.1017/S0305004100021162 <https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/abs/new-notation-for-quantum-mechanics/4631DB9213D680D6332BA11799D76AFB>
- [38] VON NEUMANN, J. (1955). *Mathematical Foundations of Quantum Mechanics*. Princeton University Press. (Original work published 1932) <https://press.princeton.edu/books/hardcover/9780691178561/mathematical-foundations-of-quantum-mechanics?srsIid=AfmB0oos0QFcFAn9e7Zzwp30-zV2hJeWTVXxk8S6I20mydNGdDianQXN>
- [39] SCHRÖDINGER, E. (1935). *Die gegenwärtige Situation in der Quantenmechanik*. *Naturwissenschaften, 23*(48), 807-812. DOI: [10.1007/BF01491891] <https://doi.org/10.1007/BF01491891>
- [40] BOHR, N. (1928). *The quantum postulate and the recent development of atomic theory*. *Nature, 121*(3050), 580-590. DOI: [10.1038/121580a0] <https://doi.org/10.1038/121580a0>
- [41] FEYNMAN, R. P. (1965). *The Feynman Lectures on Physics: Volume III – Quantum Mechanics**. Addison-Wesley. https://www.feynmanlectures.caltech.edu/III_01.html#Ch1-S5
- [42] HAFFNER, H., ROOS, C., & BLATT, R. (2008). *Quantum computing with trapped ions*. *Physics Reports, 469*(4), 155–203. <http://dx.doi.org/10.1016/j.physrep.2008.09.003>
- [43] KNILL, E., LAFLAMME, R., & MILBURN, G. J. (2001). *A scheme for efficient quantum computation with linear optics*. *Nature, 409*(6816), 46–52. DOI: [10.1038/35051009] <https://doi.org/10.1038/35051009>
- [44] LOSS, D., & DIVINCENZO, D. P. (1998). *Quantum computation with quantum dots*. *Physical Review A, 57*(1), 120-126. DOI: [10.1103/PhysRevA.57.120] <https://doi.org/10.1103/PhysRevA.57.120>

- [45] KOCH, J., YU, T.-M., GAMBETTA, J. M., HOUCK, A. A., SCHUSTER, D. I., & ET AL. (2007). *Charge-insensitive qubit design derived from the Cooper pair box*. *Physical Review A, 76*(4), 042319. DOI: [10.1103/PhysRevA.76.042319]<https://doi.org/10.1103/PhysRevA.76.042319>
- [46] SHOR, P. W. (1995). *Scheme for reducing decoherence in quantum computer memory*. *Physical Review A, 52*(4), R2493-R2496. DOI: [10.1103/PhysRevA.52.R2493]<https://doi.org/10.1103/PhysRevA.52.R2493>
- [47] STEANE, A. M. (1996). *Error correcting codes in quantum theory*. *Physical Review Letters, 77*(5), 793-797. DOI: [10.1103/PhysRevLett.77.793]<https://doi.org/10.1103/PhysRevLett.77.793>
- [48] KITAEV, A. (2003). *Fault-tolerant quantum computation by anyons*. *Annals of Physics, 303*(1), 2-30. DOI: [10.1016/S0003-4916(02)00018-0][https://doi.org/10.1016/S0003-4916\(02\)00018-0](https://doi.org/10.1016/S0003-4916(02)00018-0)
- [49] BENNETT, C. H., BRASSARD, G., CRÉPEAU, C., JOZSA, R., PERES, A., & WOOTTERS, W. K. (1993). *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*. *Physical Review Letters, 70*(13), 1895-1899. DOI: [10.1103/PhysRevLett.70.1895]<https://doi.org/10.1103/PhysRevLett.70.1895>
- [50] GROVER, L. K. (1996). *A fast quantum mechanical algorithm for database search*. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, 212-219. DOI: [10.1145/237814.237866]<https://doi.org/10.1145/237814.237866>
- [51] ARUTE, F., ARYA, K., BABBUSH, R., BACON, D., BARDIN, J. C., BARENDT, R., ... & MARTINIS, J. M. (2019). *Quantum supremacy using a programmable superconducting processor*. *Nature, 574*(7779), 505-510. DOI: [10.1038/s41586-019-1666-5]<https://doi.org/10.1038/s41586-019-1666-5>
- [52] BERNSTEIN, E., & VAZIRANI, U. (1993). *Quantum complexity theory*. *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC '93)*, 11-20. DOI: [10.1145/167088.167097]<https://doi.org/10.1145/167088.167097>
- [53] JAVADI-ABHARI, A., TREINISH, M., KRSULICH, K., WOOD, C. J., LISHMAN, J., GACON, J., MARTIEL, S., NATION, P. D., BISHOP, L. S., CROSS, A. W., JOHNSON,

- B. R., GAMBETTA, J. M., *Quantum computing with Qiskit*, arXiv, 2024, <https://doi.org/10.48550/arXiv.2405.08810>
- [54] CIRQ DEVELOPERS, *Cirq (v1.4.0)*, Zenodo, 2024. <https://doi.org/10.5281/zenodo.11398048>.
- [55] BERGHOLM, V., IZAAC, J., SCHULD, M., GOGOLIN, C., AHMED, S., AJITH, V., ALAM, M.S., ALONSO-LINAJE, G., AKASHNARAYANAN, B., ASADI, A., ARRAZOLA, J.M., AZAD, U., BANNING, S., BLANK, C., BROMLEY, T.R., CORDIER, B.A., CERONI, J., DELGADO, A., DI MATTEO, O., DUSKO, A., GARG, T., GUALA, D., HAYES, A., HILL, R., IJAZ, A., ISACSSON, T., ITTAH, D., JAHANGIRI, S., JAIN, P., JIANG, E., KHANDELWAL, A., KOTTMANN, K., LANG, R.A., LEE, C., LOKE, T., LOWE, A., MCKIERNAN, K., MEYER, J.J., MONTAÑEZ-BARRERA, J.A., MOYARD, R., NIU, Z., O’RIORDAN, L.J., OUD, S., PANIGRAHI, A., PARK, C.-Y., POLATAJKO, D., QUE-SADA, N., ROBERTS, C., SÁ, N., SCHOCH, I., SHI, B., SHU, S., SIM, S., SINGH, A., STRANDBERG, I., SONI, J., SZÁVA, A., THABET, S., VARGAS-HERNÁNDEZ, R.A., VINCENT, T., VITUCCI, N., WEBER, M., WIERICHS, D., WIERSEMA, R., WILLMANN, M., WONG, V., ZHANG, S., & KILLORAN, N., *PennyLane: Automatic differentiation of hybrid quantum-classical computations*, arXiv preprint arXiv:1811.04968, 2022, <https://arxiv.org/abs/1811.04968>.
- [56] SCHRÖDINGER, E., *An Undulatory Theory of the Mechanics of Atoms and Molecules*, Phys. Rev., 28(6), 1049-1070 (1926). <https://doi.org/10.1103/PhysRev.28.1049>
- [57] BLINDER, S.M., *Mathematical Physics in Theoretical Chemistry*, Developments in Physical & Theoretical Chemistry, Elsevier, 2019, pp. 1-30. <https://doi.org/10.1016/B978-0-12-813651-5.00001-2>.
- [58] CAO, Y., ROMERO, J., *et al.*, *Quantum Chemistry in the Age of Quantum Computing*, Chem. Rev., vol. 119, no. 19, pp. 10856-10915, 2019. <https://doi.org/10.1021/acs.chemrev.8b00803>
- [59] BORN, M. and OPPENHEIMER, R., *Zur Quantentheorie der Molekeln*, Annalen der Physik, vol. 389, no. 20, pp. 457-484, 1927. <https://doi.org/10.1002/andp.19273892002>

- [60] DIRAC, P. A. M., *The quantum theory of the emission and absorption of radiation*, Proc. R. Soc. Lond. A, vol. 114, pp. 243-265, 1927. <https://doi.org/10.1098/rspa.1927.0039>
- [61] SEEGER, R. and POPLE, J.A., *Self-consistent molecular orbital methods. XVIII. Constraints and stability in Hartree-Fock theory*, The Journal of Chemical Physics, vol. 66, no. 7, pp. 3045-3050, Apr. 1977. <https://doi.org/10.1063/1.434318>.
- [62] KAPLAN, T.A. and KLEINER, W.H., *Hartree-Fock Theory: Slater Determinants of Minimum Energy*, Phys. Rev., 156(1):1-11, Apr. 1967, <https://link.aps.org/doi/10.1103/PhysRev.156.1>.
- [63] NEGELE, J.W., *The mean-field theory of nuclear structure and dynamics*, ev. Mod. Phys., 54(4):913-1015, Oct. 1982, <https://link.aps.org/doi/10.1103/RevModPhys.54.913>.
- [64] FERMAN, J.T. and VALEEV, E.F., *Fundamentals of Molecular Integrals Evaluation*, arXiv, 2020. <https://arxiv.org/abs/2007.12057>.
- [65] SEELEY, J.T., RICHARD, M.J., and LOVE, P.J., *The Bravyi-Kitaev transformation for quantum computation of electronic structure*, The Journal of Chemical Physics, vol. 137, no. 22, pp. 224109, Dec. 2012. <https://doi.org/10.1063/1.4768229>.
- [66] VERYAZOV, V., MALMQVIST, P.Å., & ROOS, B.O., *How to select active space for multiconfigurational quantum chemistry?*, International Journal of Quantum Chemistry, 111:3329-3338, 2011, <https://doi.org/10.1002/qua.23068>.
- [67] SAYFUTYAROVA, E.R., SUN, Q., CHAN, G.K.-L., & KNIZIA, G., *Automated Construction of Molecular Active Spaces from Atomic Valence Orbitals*, Journal of Chemical Theory and Computation, 13(9):4063-4078, 2017, <https://doi.org/10.1021/acs.jctc.7b00128>.
- [68] SZABO, A., & OSTLUND, N. S., (1989). *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover Publications.
- [69] JENSEN, F., (2017). *Introduction to Computational Chemistry*, John Wiley & Sons.
- [70] CAO, Y., ROMERO, J., OLSON, J.P., DEGROOTE, M., JOHNSON, P.D., KIEFEROVÁ, M., KIVLICHAN, I.D., MENKE, T., PEROPADRE, B., SAWAYA, N.P.D., SIM,

- S., VEIS, L. & ASPURU-GUZI, A., *Quantum Chemistry in the Age of Quantum Computing*, Chemical Reviews, 119(19):10856–10915, 2019, <https://doi.org/10.1021/acs.chemrev.8b00803>.
- [71] BLUNT, N.S., CAMPS, J., CRAWFORD, O., IZSÁK, R., LEONTICA, S., MIRANI, A., MOYLETT, A.E., SCIVIER, S.A., SÜNDERHAUF, C., SCHOPF, P., TAYLOR, J.M., & HOLZMANN, N., *Perspective on the Current State-of-the-Art of Quantum Computing for Drug Discovery Applications*, Journal of Chemical Theory and Computation, 18(12):7001–7023, 2022, <https://doi.org/10.1021/acs.jctc.2c00574>.
- [72] FARHI, E., GOLDSTONE, J. y GUTMANN, S., *A Quantum Approximate Optimization Algorithm*, arXiv preprint, arXiv:1411.4028, 2014. <https://arxiv.org/abs/1411.4028>.
- [73] ARRAZOLA, J. M., DI MATTEO, O., QUESADA, N., JAHANGIRI, S., DELGADO, A., y KILLORAN, N., *Universal quantum circuits for quantum chemistry*, arXiv:2106.13839, 2021.
- [74] D-WAVE SYSTEMS INC., (2023). *D-Wave Quantum Computers*, Retrieved from <https://www.dwavesys.com/>
- [75] JORDAN, P., & WIGNER, E., (1928). *Über das Paulische Äquivalenzverbot*, Zeitschrift für Physik, 47(9-10), 631-651.
- [76] MCGEOCH, C. C., (2014). *Adiabatic quantum computation and quantum annealing: Theory and practice*, Synthesis Lectures on Quantum Computing, 5(2), 1-93.
- [77] SCHULD, M., & PETRUCCIONE, F., (2018). *Supervised Learning with Quantum Computers*, Springer.
- [78] DELGADO, A. *“optimization of molecular geometries Demo” PennyLane GitHub Repository*, <https://github.com/PennyLaneAI/pennylane/tree/main/demos/qchem>, 2024.
- [79] CEREZO, M., & COLLEAGUES, (2021). *Variational Quantum Algorithms*, Nature Reviews Physics, 3, 620–637.
- [80] RUDER, S., *An overview of gradient descent optimization algorithms*, arXiv:1609.04747, 2017. <https://arxiv.org/abs/1609.04747>.

- [81] KNOWLES, P. J. & HANDY, N. C., *A new determinant-based full configuration interaction method*, Chemical Physics Letters, vol. 111, no. 4, pp. 315–321, 1984. doi: [https://doi.org/10.1016/0009-2614\(84\)85513-X](https://doi.org/10.1016/0009-2614(84)85513-X).
- [82] CHOI, S., & KJAERGAARD, M., (2022). *Superconducting Qubits: A Review*, Annual Review of Condensed Matter Physics, 13, 219-242.

A. Anexos Contexto y Estado del Arte

A.1. Computación Cuántica, su relación con las Ciencias

La computación cuántica es un campo interdisciplinar que se nutre de aportaciones teóricas de diversas áreas de la ciencia. He aquí una visión general de cómo las distintas disciplinas científicas han contribuido al desarrollo de la computación cuántica:

1. Física

- **Mecánica cuántica:** La base de la informática cuántica es la mecánica cuántica. Principios básicos como la superposición, el entrelazamiento y la interferencia cuántica son fundamentales para la computación cuántica. Los bits cuánticos (qubits) pueden existir en varios estados a la vez (superposición), y los qubits pueden estar entrelazados, lo que significa que el estado de un qubit depende del estado de otro, independientemente de la distancia que los separe. Estos principios permiten a los ordenadores cuánticos procesar simultáneamente una gran cantidad de información.
- **Física de la materia condensada:** Estudia las propiedades físicas de las fases condensadas de la materia. Los descubrimientos en este campo, en particular la comprensión de la superconductividad y el efecto Hall cuántico, han sido cruciales para desarrollar tecnologías de qubits y circuitos cuánticos. Los qubits superconductores, por ejemplo, son una aplicación directa de estos conocimientos.
- **Teoría cuántica de campos:** Proporciona un marco teórico para comprender y describir el comportamiento mecánico cuántico de campos y partículas. Las técnicas de la teoría cuántica de campos se utilizan para modelar interacciones en sistemas cuánticos, esenciales para desarrollar métodos de computación cuántica tolerantes a fallos y corrección cuántica de errores.

2. Informática

- **Teoría de la información:** Los conceptos de la teoría clásica de la información, como la entropía y la entropía de la información, se han extendido a los sistemas cuánticos. La teoría cuántica de la información estudia la codificación, la transmisión y el tratamiento de la información en los sistemas cuánticos. Los códigos cuánticos de corrección de errores, esenciales para la informática cuántica práctica, tienen su origen en este campo.
- **Algoritmos y teoría de la complejidad:** Los informáticos han desarrollado algoritmos cuánticos que aprovechan las propiedades únicas de la mecánica cuántica. Algunos ejemplos notables son el algoritmo de Shor para factorizar números enteros grandes, exponencialmente más rápido que los algoritmos clásicos, y el algoritmo de Grover para buscar en bases de datos sin ordenar, cuadráticamente más rápido. La teoría de la complejidad ayuda a clasificar los problemas en función de su dificultad computacional y los recursos necesarios, guiando el desarrollo de algoritmos cuánticos eficientes.
- **Criptografía:** El campo de la criptografía tiene una relación bidireccional con la informática cuántica. Los ordenadores cuánticos suponen una amenaza para los protocolos criptográficos clásicos (por ejemplo, RSA) debido a su capacidad para resolver ciertos problemas mucho más rápidamente. A la inversa, la criptografía cuántica, incluida la distribución cuántica de claves (QKD), aprovecha la mecánica cuántica para crear canales de comunicación teóricamente seguros.

3. Matemáticas

- **Álgebra lineal:** La computación cuántica se basa en gran medida en el álgebra lineal. El estado de un sistema cuántico se describe mediante vectores en un espacio complejo de Hilbert, y las operaciones cuánticas se representan mediante matrices unitarias. Comprender los valores propios, los vectores propios y los productos tensoriales es crucial para describir algoritmos y operaciones cuánticas.
- **Probabilidad y estadística:** La mecánica cuántica es inherentemente probabilística. Los resultados de las mediciones cuánticas se describen mediante distribuciones de probabilidad, y se utilizan métodos estadísticos para interpretar estos resultados. Esta naturaleza probabilística es un aspecto clave del análisis de algoritmos cuánticos y de las técnicas de corrección de errores.

- **Teoría de grupos:** La teoría de grupos y sus aplicaciones son importantes para comprender las simetrías de los sistemas cuánticos y la estructura matemática de los algoritmos cuánticos. Por ejemplo, el algoritmo de Shor para la factorización de números enteros utiliza propiedades de los grupos cíclicos.

4. Ingeniería Eléctrica y Electrónica

- **Desarrollo de hardware cuántico:** Los ingenieros trabajan en la construcción de los sistemas físicos que hacen realidad los qubits y las puertas cuánticas. Esto incluye circuitos superconductores, iones atrapados, puntos cuánticos y qubits topológicos. Comprender y mitigar el ruido, la decoherencia y otros retos físicos son problemas de ingeniería fundamentales para ampliar los ordenadores cuánticos.
- **Teoría de control:** Controlar los sistemas cuánticos con gran precisión es necesario para realizar cálculos cuánticos exactos. Las técnicas de la teoría del control se utilizan para diseñar pulsos y mecanismos de retroalimentación que permitan manipular los qubits y mantener la coherencia a lo largo del tiempo.

5. Química y ciencia de los materiales

- **Química cuántica:** Los ordenadores cuánticos tienen el potencial de simular sistemas cuánticos de forma eficiente, lo que resulta especialmente útil en química cuántica. La comprensión de las estructuras moleculares, las vías de reacción y las propiedades de los materiales a nivel cuántico puede conducir a nuevos materiales y fármacos.
- **Ciencia de los materiales:** El desarrollo de qubits y puertas cuánticas estables requiere materiales con propiedades específicas. Los científicos de materiales trabajan en el descubrimiento y la fabricación de materiales que puedan soportar la coherencia cuántica y sean menos susceptibles al ruido ambiental.

El desarrollo de la computación cuántica es un esfuerzo multidisciplinar. La física aporta los principios fundamentales, la informática desarrolla los algoritmos y modelos computacionales, las matemáticas ofrecen el formalismo y las herramientas necesarias, la ingeniería eléctrica diseña y construye el hardware, y la química y la ciencia de los materiales contribuyen al desarrollo de materiales y aplicaciones po-

tenciales. Cada uno de estos campos se entrelaza para ampliar las posibilidades de la tecnología cuántica.

A.2. Eventos Claves en la Computación Cuántica

Esta cronología ofrece una panorámica completa de los principales acontecimientos y avances en el desarrollo de la computación cuántica, desde principios del siglo XX hasta nuestros días.

1. Surgimiento de la mecánica cuántica

- **1900:** Max Planck introduce el concepto de energía cuantizada en su trabajo sobre la radiación del cuerpo negro.
- **1905:** La explicación de Albert Einstein del efecto fotoeléctrico sugiere la existencia de fotones, sentando las bases de la mecánica cuántica.

2. Dualidad onda-partícula

- **1925:** Werner Heisenberg formula la mecánica matricial, que proporciona un marco matemático para la mecánica cuántica.
- **1926:** Erwin Schrödinger desarrolla la ecuación de onda, que describe el comportamiento de las partículas cuánticas en términos de ondas.

3. Entrelazamiento cuántico y la paradoja EPR

- **1935:** Erwin Schrödinger introduce el concepto de entrelazamiento cuántico en su artículo sobre la paradoja EPR¹, destacando las correlaciones no locales entre partículas.

4. El formalismo de Dirac

- **1926:** Paul Dirac desarrolla el formalismo de la mecánica cuántica introduciendo la notación bra-ket², proporcionando una descripción matemática concisa de los estados y operadores cuánticos.

¹La paradoja EPR (Einstein-Podolsky-Rosen) plantea un experimento mental con partículas entrelazadas que desafía la localidad en la física clásica. Las mediciones correlacionadas instantáneamente violarían la relatividad especial. La mecánica cuántica predice correctamente estas correlaciones, sugiriendo una no-localidad o variables ocultas aún por descubrir. La paradoja EPR impulsa la computación cuántica al destacar el entrelazamiento cuántico, un recurso clave para operaciones simultáneas en múltiples qubits. La no-localidad de la paradoja inspira protocolos de comunicación y computación cuántica distribuida. Además, contribuye a la corrección de errores cuánticos, crucial para la escalabilidad. La teleportación cuántica, inspirada en la paradoja, abre posibilidades de transferencia de estados cuánticos a distancia.

²La notación bra-ket (Dirac) describe vectores y operadores cuánticos. Un ket ($|\psi\rangle$) representa un estado cuántico, mientras que un bra ($\langle\psi|$) es su conjugado hermítico. El producto bra-ket ($\langle\psi|\psi\rangle$) calcula la probabilidad de encontrar el sistema en el estado ket. Lineal, hermítica y útil para estados, operadores, amplitudes de probabilidad y mecánica cuántica relativista, la notación bra-ket ofrece concisión, claridad y versatilidad en el mundo cuántico.

- **1930:** El libro de Dirac «Principles of Quantum Mechanics» consolida su formalismo y se convierte en piedra angular en el estudio de la mecánica cuántica.

5. Era posterior a la Segunda Guerra Mundial: Desarrollo de la Física del Estado Solido

- **1940s:** La mecánica cuántica desempeña un papel crucial en el desarrollo de armas atómicas durante la Segunda Guerra Mundial.
- **1950s:** El auge de la física del estado sólido conduce a la exploración de los fenómenos cuánticos en los materiales, contribuyendo a la comprensión de los semiconductores y al desarrollo de tecnologías como el transistor.

6. Conceptos iniciales y fundamentos teóricos

- **1960s:** Los trabajos teóricos de Paul Benioff y Yuri Manin proponen el concepto de computación cuántica.
- **1968:** El físico Richard Feynman sugiere que la simulación de sistemas cuánticos con ordenadores clásicos es intratable, plantando la semilla de la computación cuántica.
- **1970s:** El físico Stephen Wiesner introduce el concepto de dinero cuántico e información cuántica, estimulando aún más el interés por la computación cuántica.

7. Desarrollo de algoritmos cuánticos clave

- **1980:** Paul Benioff demuestra un modelo teórico de una máquina de Turing mecánica cuántica, mostrando cómo los sistemas cuánticos pueden realizar cálculos.
- **1982:** Richard Feynman propone el concepto de simulación cuántica, sugiriendo que los ordenadores cuánticos podrían simular eficientemente los sistemas cuánticos.
- **1985:** David Deutsch introduce el concepto de ordenador cuántico universal y propone el primer algoritmo cuántico, que puede resolver problemas de forma más eficiente que los algoritmos clásicos en determinados casos.

8. Corrección cuántica de errores

- **1994:** Peter Shor publica su innovador algoritmo para la factorización de números enteros, demostrando el potencial de la computación cuántica para resolver eficazmente problemas complejos que son intratables para los ordenadores clásicos.
- **1995:** Lov Grover desarrolla el algoritmo de Grover, que proporciona un algoritmo cuántico que ofrece una aceleración cuadrática para problemas de búsqueda no estructurados.
- **1996:** Peter Shor y Andrew Steane proponen de forma independiente códigos cuánticos de corrección de errores, esenciales para preservar la información cuántica frente al ruido y la decoherencia, permitiendo así una computación cuántica fiable.
- **1997:** La primera demostración experimental de un algoritmo cuántico (algoritmo Deutsch-Jozsa) se realiza mediante resonancia magnética nuclear (RMN).

9. Primeros bits cuánticos (Qubits)

- **1998:** Investigadores de IBM y de la Universidad de California en Santa Bárbara demuestran los primeros qubits operativos, utilizando uniones Josephson superconductoras y resonancia magnética nuclear (RMN), respectivamente, para manipular y medir estados cuánticos.

10. Realización experimental de la interferencia cuántica y Comercialización:

- **2001:** Investigadores de IBM demuestran el primer ordenador cuántico funcional capaz de factorizar el número 15 mediante técnicas de resonancia magnética nuclear (RMN).
- **2006:** Investigadores de la Universidad de Maryland demuestran la interferencia cuántica con iones atrapados, un fenómeno clave para la computación cuántica, mostrando la capacidad de manipular y controlar qubits para exhibir patrones de interferencia.
- **2007:** D-Wave Systems presenta el primer ordenador comercial de recocido cuántico, aunque surgen debates sobre su naturaleza cuántica.

11. Avances y Supremacía cuántica:

- **2012:** Los investigadores demuestran los primeros ordenadores cuánticos a pequeña escala capaces de realizar cálculos sencillos, incluidos algoritmos como los de Shor y Grover.
- **2015:** IBM hace accesible la computación cuántica a través de la nube con IBM Quantum Experience, permitiendo a los usuarios ejecutar algoritmos en un procesador cuántico.
- **2019:** Google afirma alcanzar la supremacía cuántica con su procesador cuántico de 53 qubits, Sycamore, realizando un cálculo específico significativamente más rápido que los superordenadores clásicos más potentes.

12. Avances continuos y aplicaciones prácticas :

- **2020s:** La investigación en curso se centra en mejorar la coherencia y las tasas de error de los qubits, desarrollar arquitecturas de computación cuántica tolerantes a fallos y explorar aplicaciones prácticas en criptografía, optimización y ciencia de los materiales, impulsadas por los avances en formalismos matemáticos, superposición, entrelazamiento e interferencia de los qubits.
- **2020s:** Desarrollo de algoritmos cuánticos variacionales para resolver problemas de optimización y algoritmos híbridos cuántico-clásicos para aplicaciones prácticas en química y ciencia de materiales.
- **2020s:** Los avances en algoritmos cuánticos de aprendizaje automático abren nuevas posibilidades para la inteligencia artificial y el análisis de datos.
- **2020s:** Los investigadores siguen mejorando las técnicas de corrección de errores cuánticos, allanando el camino para ordenadores cuánticos escalables y tolerantes a fallos.
- **2023:** IBM y otras empresas desarrollan procesadores cuánticos de compuertas universales con más de 1000 qubits, avanzando hacia una ventaja cuántica práctica.

A.3. Notación de Dirac

La notación de Dirac³, también conocida como notación bra-ket, es una forma estandarizada de describir estados cuánticos y operaciones en mecánica cuántica y computación cuántica. Resulta especialmente útil para manejar el álgebra lineal que subyace en la mecánica cuántica. A continuación se presentan en una tabla los detalles de la notación de Dirac y su importancia en la ciencia de la información cuántica.

Concepto	Descripción	Ejemplo
Kets	Un ket $ \psi\rangle$ representa un vector columna en un espacio vectorial complejo (Hilbert). Denota un estado cuántico.	$ \psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
Bras	Un bra $\langle\psi $ Es el conjugado hermitiano (transpuesto conjugado complejo) de un ket, representando un vector fila.	$\langle\psi = (\alpha^* \quad \beta^*)$
Producto Interno	El producto interno de dos estados $ \psi\rangle$ y $ \phi\rangle$ es expresado como $\langle\psi \phi\rangle$. Da el "solapamiento" entre dos estados cuánticos. Es un número complejo.	Para $ \psi\rangle = \alpha 0\rangle + \beta 1\rangle$ y $ \phi\rangle = \gamma 0\rangle + \delta 1\rangle$, $\langle\psi \phi\rangle = \alpha^*\gamma + \beta^*\delta$
Producto Externo	El producto externo de dos estados $ \psi\rangle$ y $ \phi\rangle$ forma un operador (matriz), es expresado como $ \psi\rangle\langle\phi $. Forma un operador que puede actuar sobre otros estados en el espacio de Hilbert.	Para $ \psi\rangle = \alpha 0\rangle + \beta 1\rangle$ y $ \phi\rangle = \gamma 0\rangle + \delta 1\rangle$, $ \psi\rangle\langle\phi = (\alpha 0\rangle + \beta 1\rangle)(\gamma^*\langle 0 + \delta^*\langle 1) = \alpha\gamma^* 0\rangle\langle 0 + \alpha\delta^* 0\rangle\langle 1 + \beta\gamma^* 1\rangle\langle 0 + \beta\delta^* 1\rangle\langle 1 $

³Paul Adrien Maurice Dirac (1902-1984) fue un físico teórico británico pionero que realizó importantes contribuciones a la mecánica cuántica y la electrodinámica cuántica.

Concepto	Descripción	Ejemplo
Operadores	Entidades que actúan sobre kets para producir otros kets. Pueden representar observables físicos o transformaciones.	<p>Operadores unitarios: preservan la norma de los estados. Si U es unitario, entonces $U^\dagger U = U U^\dagger = I$.</p> <p>Operadores Hermitianos: representan cantidades observables. Si A es Hermitiano, entonces $A = A^\dagger$.</p>
Medición y Proyección	Medición asociada con operadores hermitianos. La proyección del estado cuántico a un eigenestado específico con cierta probabilidad.	Para un observable A con eigenestados $ \psi_i\rangle$ y eigenvalores λ_i , la medición de A en estado $ \psi\rangle$ proyecta $ \psi\rangle$ a $ \psi_i\rangle$ con probabilidad $ \langle\psi_i \psi\rangle ^2$
Puertas Cuánticas	Operadores unitarios que transforman estados de qubits.	<p>Pauli-X (NOT) Gate:</p> $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ <p>Hadamard Gate:</p> $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Valores y Vectores Propios	Para un operador \hat{A} , un eigenvector $ \psi\rangle$ con eigenvalor λ satisface:	<p>Sea el operador $\hat{A} = \sigma_z$, los valores propios $\lambda = 1$ y $\lambda = -1$ definen los vectores propios:</p> $\hat{A} \psi\rangle = \lambda \psi\rangle$ <p>.</p> $ \psi_1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ y } \psi_2\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ <p>que representan los estados en los que el operador σ_z actúa simplemente multiplicando por su correspondiente valor propio.</p>

Concepto	Descripción	Ejemplo
Matrices de Densidad	Describen un estado cuántico mixto que generaliza los estados puros y permite mezclas estadísticas de estados.	<p>Para un sistema en estado $\psi_i\rangle$ con probabilidad p_i,</p> $\rho = \sum_i p_i \psi_i\rangle\langle\psi_i $ <p>.</p> <p>Propiedades: ρ es hermitiano, $\text{Tr}(\rho) = 1$, y $\rho \geq 0$.</p>
Notación en Algoritmos Cuánticos	Usada extensamente para formular algoritmos cuánticos como Grover y Shor.	<p>Algoritmo de Grover: Utiliza superposición y amplificación de amplitud.</p> <p>Algoritmo de Shor: Para factorización de enteros, utiliza transformada cuántica de Fourier y exponenciación modular.</p>

La notación de Dirac proporciona un marco conciso y potente para describir estados, operaciones y medidas cuánticas. Al representar los estados cuánticos como vectores (kets) y sus conjugados como bras, y utilizar productos internos y externos para formar operadores y calcular probabilidades, la notación de Dirac encapsula con elegancia el álgebra lineal subyacente a la mecánica cuántica. Esta notación es esencial para manipular y comprender matemáticamente los sistemas cuánticos.

A.4. Álgebra Lineal Espacio de Hilbert

Un espacio de Hilbert complejo es un concepto clave en la teoría cuántica y se puede resumir así:

1. Espacio vectorial complejo: Un conjunto de vectores sobre números complejos \mathbb{C} , donde se pueden realizar operaciones de suma y multiplicación por escalares complejos.

2. Producto interno: Una función $\langle \cdot, \cdot \rangle$ que asigna a cada par de vectores x, y un número complejo, cumpliendo las propiedades de linealidad en el primer argumento, conjugado hermitiano (simetría conjugada) en el segundo argumento $\langle x, y \rangle = \overline{\langle y, x \rangle}$, y positividad $\langle x, x \rangle \geq 0$, con $\langle x, x \rangle = 0$ si y solo si $x = 0$. Es crucial para definir nociones de ángulo y longitud (norma) en el espacio.

3. Norma: La norma (longitud) de un vector se define mediante el producto interno como $\|x\| = \sqrt{\langle x, x \rangle}$, permitiendo medir distancias y evaluar convergencia.

4. Completitud: El espacio es completo, lo que significa que todas las secuencias de Cauchy (secuencias donde los elementos se acercan entre sí a medida que avanzan) convergen a un límite dentro del espacio, garantizando la existencia de límites para todas las secuencias convergentes.

Estos espacios son fundamentales en la formulación matemática de la mecánica cuántica, donde los estados cuánticos se representan como vectores en un espacio de Hilbert complejo y los observables (magnitudes físicas que pueden ser medidas) como operadores lineales en dicho espacio.

Tabla A.2: Resumen de Conceptos de Álgebra Lineal y Espacio de Hilbert en Mecánica Cuántica

Concepto	Descripción	Ejemplos
Espacio de Hilbert	Un espacio vectorial completo con un producto interno, que permite la generalización del espacio euclidiano. Los estados cuánticos se representan como vectores en este espacio.	\mathbb{C}^2 para un qubit, \mathbb{C}^n para un sistema de n niveles
Continúa en la siguiente página		

Concepto	Descripción	Ejemplos
Producto Interno	Para vectores $\psi, \phi \in \mathcal{H}$, el producto interno $\langle \psi, \phi \rangle$ es un número complejo que satisface simetría conjugada, linealidad y positividad definida.	$\langle \psi, \phi \rangle = \sum_i \psi_i^* \phi_i$ para $\psi, \phi \in \mathbb{C}^n$
Ortogonalidad	Dos vectores ψ y ϕ son ortogonales si $\langle \psi, \phi \rangle = 0$.	Los estados $ 0\rangle$ y $ 1\rangle$ de un qubit son ortogonales
Base Ortonormal	Un conjunto de vectores $\{\psi_i\}$ en \mathcal{H} es una base ortonormal si $\langle \psi_i, \psi_j \rangle = \delta_{ij}$, donde δ_{ij} es el delta de Kronecker.	$ 0\rangle, 1\rangle$ forman una base ortonormal en \mathbb{C}^2
Vectores de Estado	Los estados cuánticos se representan por vectores de estado $ \psi\rangle$ en el espacio de Hilbert.	$ \psi\rangle = \alpha 0\rangle + \beta 1\rangle$ para un qubit
Operadores	Los observables físicos y operaciones se representan mediante operadores lineales en el espacio de Hilbert. Un operador \hat{A} actuando sobre \mathcal{H} mapea \mathcal{H} en \mathcal{H} .	Operador Pauli X : $\hat{X} 0\rangle = 1\rangle, \hat{X} 1\rangle = 0\rangle$
Operadores Hermitianos	Un operador \hat{A} es Hermitiano si $\hat{A} = \hat{A}^\dagger$, donde \hat{A}^\dagger es el adjunto de \hat{A} . Los operadores Hermitianos corresponden a cantidades observables en mecánica cuántica.	Operador Pauli Z : $\hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Operadores Unitarios	Un operador \hat{U} es unitario si $\hat{U}^\dagger \hat{U} = \hat{U} \hat{U}^\dagger = \mathbb{I}$, donde \mathbb{I} es el operador identidad. Los operadores unitarios representan evoluciones reversibles de estados cuánticos.	Puerta Hadamard: $\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Espacios Vectoriales	Un espacio vectorial V sobre un campo \mathbb{C} (números complejos) es un conjunto de vectores con dos operaciones: suma de vectores y multiplicación por escalar.	$\mathbb{C}^2, \mathbb{C}^n$
Continúa en la siguiente página		

Concepto	Descripción	Ejemplos
Matrices	Las matrices representan operadores lineales. Por ejemplo, una matriz 2x2 puede representar una puerta de un solo qubit.	Matriz Pauli X : $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Valores y Vectores Propios	Si \hat{A} es un operador lineal, un escalar λ y un vector $ \psi\rangle$ satisfacen $\hat{A} \psi\rangle = \lambda \psi\rangle$, donde λ es un valor propio y $ \psi\rangle$ es el vector propio correspondiente.	Para \hat{Z} , $\hat{Z} 0\rangle = 0\rangle$, $\hat{Z} 1\rangle = - 1\rangle$
Productos Tensoriales	Los productos tensoriales describen sistemas de múltiples qubits. Si $ \psi\rangle \in \mathcal{H}_1$ y $ \phi\rangle \in \mathcal{H}_2$, el producto tensorial $ \psi\rangle \otimes \phi\rangle$ representa el estado combinado en $\mathcal{H}_1 \otimes \mathcal{H}_2$.	$ 00\rangle = 0\rangle \otimes 0\rangle$
Puertas Cuánticas	Las puertas cuánticas son operadores unitarios que actúan sobre qubits. Puertas comunes incluyen: puertas de Pauli (X , Y , Z), puerta de Hadamard (H), puerta de fase (S) y puerta de Control-NOT (CNOT).	Puerta CNOT: $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Circuitos Cuánticos	Secuencias de puertas cuánticas aplicadas a qubits, representadas usando diagramas de circuitos donde cada línea representa un qubit y cada símbolo representa una puerta.	Circuito con una puerta Hadamard seguida de una puerta CNOT

A.5. Puertas Cuánticas

Tabla A.3: Puertas Cuánticas

Puerta	Representación Matemática	Función y Descripción	Representación Matricial	Aplicaciones
Puertas de Un Solo Qubit				
Puerta Pauli-X (Puerta X) (Puerta NOT)	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Invierte el estado de un qubit, análogo a la puerta clásica NOT. $X 0\rangle = 1\rangle$ $X 1\rangle = 0\rangle$	$X \psi\rangle = X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$	Manipulación básica de qubits, creando superposición.
Pauli-Y (Puerta Y)	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Rota el estado alrededor del eje Y e introduce un factor de fase. $Y 0\rangle = i 1\rangle$ $Y 1\rangle = -i 0\rangle$	$Y \psi\rangle = Y \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} -i\beta \\ i\alpha \end{pmatrix}$	Manipulación de qubits, corrección de errores cuánticos.
Puerta Pauli-Z (Puerta Z)	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Rota el estado alrededor del eje Z, introduciendo un factor de fase. $Z 0\rangle = 0\rangle$ $Z 1\rangle = - 1\rangle$	$Z \psi\rangle = Z \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$	Cambios de fase, implementación de algoritmos cuánticos.

Puerta	Representación Matemática	Función y Descripción	Representación Matricial	Aplicaciones
Puerta Hadamard (Puerta H)	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Crea superposición al rotar el estado del qubit. $H 0\rangle = \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$ $H 1\rangle = \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$	$H \psi\rangle = H \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ $H \psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha + \beta \\ \alpha - \beta \end{pmatrix}$	Creación de superposición, esencial para algoritmos cuánticos como el algoritmo de Grover.
Puerta de Fase (Puerta S)	$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	Introduce un desplazamiento de fase de $\pi/2$ (rotación de 90 grados) alrededor del eje Z. $S 0\rangle = 0\rangle$ $S 1\rangle = i 1\rangle$	$S \psi\rangle = S \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ i\beta \end{pmatrix}$	Algoritmos cuánticos, corrección de errores cuánticos.
Puerta $\pi/8$ (Puerta T)	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	Introduce un desplazamiento de fase de $\pi/4$ (rotación de 45 grados) alrededor del eje Z. $T 0\rangle = 0\rangle$ $T 1\rangle = e^{i\pi/4} 1\rangle$	$T \psi\rangle = T \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta e^{i\pi/4} \end{pmatrix}$	Algoritmos cuánticos como el algoritmo de Shor, corrección de errores cuánticos.
Puertas de dos qubits				

Puerta	Representación Matemática	Función y Descripción	Representación Matricial	Aplicaciones
CNOT (Puerta Control-NOT)	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	Realiza una operación NOT en el qubit objetivo si el qubit de control está en el estado $ 1\rangle$.	$\text{CNOT} \psi\rangle = \text{CNOT} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{pmatrix}$	Creación de entrelazamiento, algoritmos cuánticos, corrección de errores cuánticos, teletransportación cuántica.
Puerta SWAP	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	Intercambia los estados de dos qubits.	$\text{SWAP} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \gamma \\ \beta \\ \delta \end{pmatrix}$	Reordenamiento de qubits, implementación de ciertos algoritmos cuánticos.
CZ (Puerta Control-Z)	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$	Introduce un desplazamiento de fase de π si el qubit de control está en el estado $ 1\rangle$.	$\text{CZ} \psi\rangle = \text{CZ} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ -\delta \end{pmatrix}$	Implementación de algoritmos cuánticos, corrección de errores cuánticos.

Puerta	Representación Matemática	Función y Descripción	Representación Matricial	Aplicaciones
Puertas de múltiples qubits				
Toffoli (Puerta CCNOT)	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	Realiza una operación NOT en el qubit objetivo si ambos qubits de control están en el estado $ 1\rangle$.	$\text{Toffoli} \psi\rangle = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \\ \zeta \\ \eta \\ \theta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \\ \zeta \\ \theta \\ -\eta \end{pmatrix}$	Computación reversible, corrección de errores cuánticos.
Fredkin (Puerta CS-WAP)	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	Puerta SWAP condicional, intercambia el segundo y tercer qubit si el primer qubit está en el estado $ 1\rangle$.	$\text{Fredkin} \psi\rangle = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \\ \zeta \\ \eta \\ \theta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \\ \eta \\ \zeta \\ \theta \end{pmatrix}$	Computación reversible, procesamiento de información cuántica.

Puerta	Representación Matemática	Función y Descripción	Representación Matricial	Aplicaciones
QFT (Puerta de Transformada de Fourier Cuántica)	$\frac{1}{\sqrt{N}} \sum_{k,j=0}^{N-1} e^{2\pi i \cdot k j / N} k\rangle \langle j $ <p>donde $N = 2^n$</p>	Realiza la transformada de Fourier cuántica, una operación clave en muchos algoritmos cuánticos.	$\begin{aligned} \text{QFT} \psi\rangle &= \text{QFT}_N \cdot \left(\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_{N-1} \right)^\top \\ &= \frac{1}{\sqrt{N}} \begin{pmatrix} \sum_{j=0}^{N-1} \alpha_j \cdot e^{2\pi i \cdot 0 \cdot j / N} \\ \sum_{j=0}^{N-1} \alpha_j \cdot e^{2\pi i \cdot 1 \cdot j / N} \\ \sum_{j=0}^{N-1} \alpha_j \cdot e^{2\pi i \cdot 2 \cdot j / N} \\ \vdots \\ \sum_{j=0}^{N-1} \alpha_j \cdot e^{2\pi i \cdot (N-1) \cdot j / N} \end{pmatrix} \end{aligned}$	Algoritmos cuánticos para factorización de enteros y problemas de logaritmo discreto.

A.6. Qubits: Implementaciones Físicas

Los qubits pueden construirse utilizando varios sistemas cuánticos, cada uno de los cuales explota diferentes propiedades físicas para la codificación y manipulación de la información cuántica. Algunos ejemplos son:

- **Iones atrapados (Trapped Ion Qubits):**

- Codifican información en los niveles de energía internos de iones atrapados, normalmente iones de elementos como el calcio, el iterbio o el magnesio.
- Son manipulados mediante láseres y campos magnéticos para manipular y controlar los estados de los qubits induciendo transiciones entre distintos niveles de energía. Los largos tiempos de coherencia y las operaciones de alta fidelidad convierten a los qubits de iones atrapados en candidatos prometedores para la computación cuántica.

- **Circuitos superconductores (Superconducting Qubits):**

- Se obtienen mediante uniones Josephson⁴ superconductoras que muestran un comportamiento cuántico debido al fenómeno de la superconductividad.
- Pueden manipularse y medirse con pulsos de microondas y se controlan mediante campos magnéticos externos. Se encuentran entre los principales candidatos para construir ordenadores cuánticos

- **Puntos cuánticos (Quantum Dot Qubits):**

- Utilizan nanoestructuras semiconductoras, conocidas como puntos cuánticos, para confinar electrones individuales. Los estados de spin de estos electrones sirven como estados qubit.
- Pueden manipularse mediante campos eléctricos, magnéticos u ópticos. Ofrecen ventajas como la escalabilidad y la compatibilidad con las tecnologías de fabricación de semiconductores existentes.

- **Qubits topológicos:**

⁴Las uniones Josephson son dispositivos esenciales en la física de la superconductividad y la computación cuántica. Se componen de dos superconductores separados por una barrera delgada, permitiendo el paso coherente de pares de electrones sin pérdida de energía. Esto da lugar al Efecto Josephson, donde una corriente continua puede fluir sin voltaje externo, dependiendo de la diferencia de fase entre los superconductores, esta diferencia de fase representa los estados cuánticos.

- Se basan en estados exóticos de la materia, como los superconductores topológicos o los estados Hall⁵ cuánticos fraccionarios, para codificar y proteger la información cuántica.

- Explotan las propiedades no locales de los sistemas físicos subyacentes, como los anyones no Abelianos⁶, para proporcionar una protección intrínseca contra errores frente a determinados tipos de ruido y decoherencia.

- **Qubits fotónicos:**

- Utilizan los estados de polarización de fotones individuales para codificar información cuántica. Pueden generarse, manipularse y detectarse utilizando elementos ópticos como divisores de haz, desfasadores y detectores.

- Son especialmente adecuados para aplicaciones en comunicación cuántica y criptografía cuántica debido a sus bajas tasas de decoherencia y a su capacidad para propagarse a largas distancias.

- **Qubits de spin:**

- Codifican la información cuántica en los estados de spin de electrones o núcleos individuales en sistemas de estado sólido, como los defectos del diamante (centros de vacantes de nitrógeno) o las impurezas del silicio.

- Se utilizan campos magnéticos o eléctricos para manipular los estados de spin de estos qubits, lo que permite el control y la medición coherentes.

Estos son sólo algunos ejemplos de los tipos de qubits que se están investigando y desarrollando en la actualidad. Cada tipo tiene sus propias ventajas y desafíos, y los investigadores están explorando activamente diversos enfoques para realizar sistemas de computación cuántica escalables y tolerantes a fallos.

⁵Los estados Hall cuánticos son fenómenos observados en sistemas bidimensionales bajo campos magnéticos intensos y bajas temperaturas. En estas condiciones, los electrones se organizan en niveles de energía discretos llamados "landau levels", y la conductividad eléctrica perpendicular al campo magnético se vuelve cuantizada en múltiplos enteros de una constante fundamental. Este efecto es crucial para la metrología de resistencia eléctrica estándar y la investigación en física de la materia condensada. Además, se exploran en la computación cuántica por su potencial en la creación de qubits robustos y libres de errores basados en topología electrónica.

⁶Los anyones no Abelianos son quasipartículas con propiedades estadísticas exóticas que los hacen candidatos prometedores para la computación cuántica topológica. Sus operaciones de trenzado no conmutativas podrían habilitar una computación cuántica robusta y tolerante a fallos, ofreciendo soluciones potenciales a los desafíos actuales en el procesamiento de información cuántica.

Tabla A.4: Tabla de Tecnologías para Implementación Física de Qubits

Tecnología	Funcionamiento	Manipulación de Qubits	Ventajas	Desventajas	Empresas / Instituciones
Superconductores	Basados en circuitos superconductores que mantienen un flujo de corriente persistente, representando el estado "0." "1".	Se manipulan utilizando campos magnéticos y pulsos eléctricos.	- Coherencia prolongada - Alta fidelidad de operaciones - Escalabilidad demostrada	- Requerimiento de temperaturas criogénicas - Complejidad de fabricación - Mayor consumo de energía	IBM, Google, Rigetti Computing, D-Wave Systems, Quantum Circuits Inc., Quantum Brilliance
Iones Atrapados	Utilizan iones atrapados en campos eléctricos y magnéticos, representando el estado "0." "1." en sus niveles de energía electrónicos.	Se manipulan utilizando láseres y campos electromagnéticos.	- Alta precisión y control - Coherencia prolongada - Escalabilidad demostrada	- Requerimiento de infraestructura compleja - Dificultad de fabricación a gran escala - Menor fidelidad de operaciones en comparación con superconductores	IonQ, Honeywell Quantum Solutions, Pasqal, Quantum Labs, Xanadu Quantum
Puntos Cuánticos	Basados en estructuras semiconductoras de tamaño nanométrico que confinan electrones en estados cuánticos, representando el estado "0." "1".	Se manipulan utilizando pulsos eléctricos y campos magnéticos.	- Potencial para escalabilidad y fabricación a gran escala - Integración con tecnología de semiconductores	- Dificultad de lograr coherencia prolongada - Fidelidad de operaciones variable - Menor experiencia en el campo	Microsoft, Intel, QuantumScape, Seeq, Quantum Materials Corporation

Tecnología	Funcionamiento	Manipulación de Qubits	Ventajas	Desventajas	Empresas / Instituciones
Fótones	Utilizan pulsos de luz polarizada, representando el estado "0. ^o "1. ^{en} la polarización de los fotones.	Se manipulan utilizando espejos, filtros y moduladores de luz.	- Capacidad de transmitir información a largas distancias - Alta fidelidad de operaciones	- Dificultad de crear fuentes de fotones individuales - Requerimiento de canales de comunicación de alta calidad - Menor escalabilidad en comparación con otras implementaciones	QuEra, Xanadu Quantum, Psi-Quantum, Light Quantum, Quantum Photonics
Defectos en Materiales	Basados en defectos cristalinos en materiales como el diamante, representando el estado "0. ^o "1. ^{en} los estados electrónicos del defecto.	Se manipulan utilizando campos electromagnéticos y pulsos de microondas.	- Potencial para escalabilidad y fabricación a gran escala - Integración con tecnología de semiconductores	- Dificultad de controlar y manipular defectos - Fidelidad de operaciones variable - Menor experiencia en el campo	AQC, Quantum Diamond Technologies, NVision, Photonic Diamond, Qnami
Átomos Neutros	Utilizan átomos neutros enfriados por láser atrapados en campos electromagnéticos, representando el estado "0. ^o "1. ^{en} sus estados de spin o momento angular orbital.	Se manipulan utilizando láseres y campos magnéticos.	- Potencial para escalabilidad a temperatura ambiente - Alta fidelidad de operaciones	- Requerimiento de infraestructura compleja para enfriamiento láser - Dificultad de manipulación a gran escala	ColdQuanta, IonQ, QuEra, Pasqal, QCTRL

Tecnología	Funcionamiento	Manipulación de Qubits	Ventajas	Desventajas	Empresas / Instituciones
Electron Spins	Aprovecha el spin del electrón en materiales semiconductores para representar el estado "0." "1".	Se manipulan utilizando pulsos eléctricos y campos magnéticos.	- Potencial para integración con tecnología de semiconductores - Operación a temperatura ambiente	- Dificultad de lograr coherencia prolongada - Fidelidad de operaciones variable - Retos en la escalabilidad	Silicon Quantum Computing, Seeq, Quantum Materials Corporation, Quantum Brilliance, Qnami
Nitrogen-Vacancy (NV) Center	Utiliza defectos específicos en diamantes, llamados centros NV, para representar el estado "0." "1." en el spin del electrón atrapado en el defecto.	Se manipulan utilizando luz y campos magnéticos.	- Alta fidelidad de operaciones a temperatura ambiente - Potencial para aplicaciones de sensores cuánticos	- Dificultad de fabricación a gran escala - Desafíos para la manipulación de múltiples qubits	NVision, Photonic Diamond, Qnami, Quantum Diamond Technologies, Aurelius QC
Topológicos	Basados en propiedades topológicas de ciertos materiales que permiten la creación de estados cuánticos robustos a errores, representando el estado "0." "1." en los modos topológicos.	Se manipulan utilizando técnicas basadas en corrientes eléctricas y campos magnéticos.	- Potencial para qubits robustos a errores - Enfoque teórico prometededor	- Etapa inicial de investigación - Dificultad de fabricación y manipulación	Microsoft, Google, IBM, Intel, Quantum Circuits Inc.

Tabla A.5: Roadmap Ampliado de Implementaciones Físicas de Qubits

Organización	Tecnología	Designación del Proyecto	Año Inicio	Qubits Físicos	Fidelidad	Etapas del Proyecto
Alibaba Cloud	Superconducting	Qubit de Transmon	2019	-	-	Investigación [se sospecha]
Alpine Quantum Technologies	Iones atrapados	Qubit de Iones	2020	-	-	Investigación [se sospecha]
Atom Computing	Neutral Atoms	Qubit de Átomo Neutral	2020	-	-	Investigación [se sospecha]
Baidu	Superconducting	Qubit de Transmon	2021	-	-	Investigación [se sospecha]
ColdQuanta	Átomos neutros	Qubit de Átomo Neutral	2021	-	-	Investigación [se sospecha]
ETH Zurich	Superconducting	Qubit de Transmon	2022	50	99.7 %	En desarrollo
Fujitsu	Superconducting	Qubit de Transmon	2021	-	-	Investigación [se sospecha]
Google	Superconducting	Sycamore	2019	53	99.99 %	En funcionamiento
Honeywell	Superconducting	Qubit de Transmon	2023	64	99.5 %	En desarrollo
IBM	Superconducting	Condor	2024	1,121	99.8 %	Planificado
IonQ	Trampas de Iones	Qubit de Iones	2022	12	99.9 %	En funcionamiento
IQM	Superconducting	Qubit de Transmon	2021	-	-	Investigación [se sospecha]
Microsoft	Superconducting	Qubit de Transmon	2023	128	99.4 %	En desarrollo
Pasqal	Trampas de Iones	Qubit de Iones	2022	20	99.8 %	En funcionamiento
Princeton University	Superconducting	Qubit de Transmon	2021	16	99.8 %	En funcionamiento
Quantum Circuits	Superconducting	Qubit de Transmon	2023	32	99.7 %	En desarrollo
Rigetti Computing	Superconducting	Aspen-4	2023	84	99.5 %	En desarrollo
Tencent	Superconducting	Qubit de Transmon	2023	25	99.5 %	En desarrollo
Xanadu	Superconducting	Qubit de Transmon	2023	40	99.6 %	En desarrollo

B. Códigos

Todos los códigos utilizados para la obtención de los resultados presentados se pueden encontrar en el siguiente repositorio [QC-MSc-Thesis](#)