



Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y Tecnología

Máster en Computación Cuántica

Aplicación del algoritmo VQE
al cálculo de la estructura de
bandas del grafeno

Trabajo Fin de Estudios

presentado por: Raúl Medrano Millán

Dirigido por: Rodrigo Gil-Merino y Rubio, y Unai Aseguinolaza Aguirreche

Ciudad: Madrid

Fecha: 23 de Julio de 2025

Índice de contenidos

Resumen	v
Abstract	vi
1. Introducción	1
1.1. Motivación del tema a tratar	2
1.2. Planteamiento del Trabajo	2
1.3. Estructura del Trabajo	3
2. Contexto y estado de la técnica	4
2.1. Introducción a la computación cuántica y a la simulación de materiales . . .	5
2.1.1. Breve descripción de la computación cuántica	5
2.1.2. Potencial de la computación cuántica para la ciencia de materiales .	6
2.1.3. Contexto específico del grafeno	8
2.2. La estructura electrónica y las bandas de energía	9
2.2.1. Formación de las bandas de energía y su relevancia para las propie- dades del grafeno	9
2.2.2. La jerarquía de métodos clásicos utilizados para calcular bandas de energía	14
2.3. Introducción al algoritmo Variational Quantum Eigensolver (VQE)	17
2.3.1. Explicación del VQE como un algoritmo híbrido cuántico-clásico . .	18
2.3.2. Componentes principales del VQE	19
2.3.3. Cálculo de estados excitados (VQD)	22
2.3.4. Ventajas y limitaciones del VQE	23
2.4. Medición de bandas de energía del grafeno con VQE	24
3. Objetivos	25
3.1. Objetivos generales	25
3.2. Objetivos específicos	25
4. Desarrollo del Trabajo	27
4.1. Introducción	27
4.2. Aproximación primeros vecinos	28

4.2.1. Bandas de energía	30
4.2.2. Hamiltoniano en términos de las matrices de Pauli	32
4.3. Aproximación primeros y segundos vecinos	32
4.3.1. Bandas de energía	34
4.3.2. Hamiltoniano en términos de las matrices de Pauli	36
5. Resultados y análisis	37
5.1. Software utilizado	37
5.2. Resultados incluyendo los primeros vecinos	40
5.3. Resultados incluyendo los primeros y segundos vecinos	43
5.4. Comparativa de los optimizadores COBYLA y SPSA	46
5.5. Análisis de los errores obtenidos en las diferentes estimaciones	48
6. Conclusiones	52
7. Trabajo futuro	54
Bibliografía	56
A. Anexos	67
A.1. Programa para el cálculo de las bandas de valencia y de conducción	67

Índice de figuras

2.1. Niveles de energía en función de la distancia de separación entre átomos . .	10
2.2. Posibles estructuras de bandas para un sólido	10
2.3. Energía de un electrón libre en función del número de onda k	11
2.4. Energías permitidas en una red unidimensional de periodicidad a	13
2.5. Orbitales atómicos en un cristal unidimensional con separación a_0 entre átomos	16
4.1. Estructura en panel de abeja del grafeno y su zona de Brillouin	27
4.2. Bandas de energía del grafeno con las interacciones de los vecinos más cercanos	31
4.3. Disposición de los átomos del grafeno	32
4.4. Bandas de energía del grafeno con las interacciones de los dos niveles de vecinos más cercanos	34
4.5. Comparativa de las bandas de energía para un corte en $k_x = 0$	35
5.1. Resultados tomando sólo los primeros vecinos y sin aplicar ruido adicional .	41
5.2. Resultados tomando sólo los primeros vecinos y aplicando ruido adicional .	42
5.3. Resultados tomando los primeros+segundos vecinos, sin aplicar ruido adicional	44
5.4. Resultados tomando los primeros+segundos vecinos, aplicando ruido adicional	45
5.5. Convergencia en el punto Γ para los optimizadores COBYLA y SPSA sin ruido	46
5.6. Convergencia en el punto Γ para los optimizadores COBYLA y SPSA con ruido	47
5.7. Conos de Dirac en el grafeno	51

Índice de tablas

5.1. Errores en el cálculo de los puntos de la red recíproca de la bandas de conducción	48
5.2. Errores en el cálculo de los puntos de la red recíproca de la bandas de valencia	50

Resumen

El grafeno, material de extraordinario interés en la física de la materia condensada por sus singulares propiedades electrónicas, presenta un desafío fundamental: el cálculo preciso de su estructura de bandas. Este Trabajo Fin de Máster aborda este problema proponiendo el uso de uno de los algoritmos más utilizados en computación cuántica.

Para ello, se emplea un modelo de enlace fuerte (tight-binding, por su denominación en inglés) que describe los saltos de los electrones entre los átomos de la red hexagonal monocapa que define al grafeno. Este modelo se extiende para incluir no solo las interacciones de cada átomo de la red con los átomos adyacentes (primeros vecinos), sino también con los siguientes más cercanos (segundos vecinos), lo que permite capturar efectos físicos más sutiles. El hamiltoniano resultante se prepara para su ejecución en los simuladores cuánticos de la plataforma Qiskit de IBM.

Los resultados en condiciones ideales demuestran que el algoritmo reproduce con alta fidelidad las soluciones analíticas, capturando rasgos esenciales como los puntos de Dirac, donde las bandas de valencia y conducción se tocan haciendo que los electrones se comporten como partículas sin masa, y la ruptura de la simetría electrón-hueco, que refleja la diferencia en el comportamiento energético de los electrones y sus ausencias (huecos). Sin embargo, al introducir un modelo de ruido que emula los procesadores cuánticos actuales (dispositivos inherentemente ruidosos y de escala intermedia, la denominada era NISQ por sus siglas en inglés), se observa una desviación sistemática que subraya el impacto limitante del hardware.

El estudio concluye que el algoritmo utilizado es una herramienta conceptualmente viable y potente para la simulación de materiales, pero su aplicación práctica para alcanzar la ventaja cuántica dependerá críticamente del desarrollo de técnicas avanzadas de mitigación de errores.

Palabras clave: computación cuántica, algoritmo VQE, grafeno, estructura de bandas, enlace fuerte, simulación cuántica.

Abstract

Graphene, a material of extraordinary interest in condensed matter physics due to its unique electronic properties, presents a fundamental challenge: the precise calculation of its band structure. This Master's Thesis addresses this problem by proposing the use of one of the most widely used algorithms in quantum computing.

To achieve this, a tight-binding model is used that describes the electron hopping between the atoms of the single-layer hexagonal lattice that defines graphene. This model is extended to include not only the interactions of each atom in the lattice with adjacent atoms (first neighbors), but also with the next closest atoms (second neighbors), allowing for the capture of more subtle physical effects. The resulting hamiltonian is then prepared for execution in the quantum simulators of IBM's Qiskit platform.

The results under ideal conditions demonstrate that the algorithm reproduces analytical solutions with high fidelity, capturing essential features such as Dirac points, where valence and conduction bands intersect, causing electrons to behave as massless particles, and electron-hole symmetry breaking, which reflects the difference in the energetic behavior of electrons and their absences (holes). However, when introducing a noise model that emulates current quantum processors (inherently noisy, intermediate-scale devices, the so-called NISQ era), a systematic deviation is observed that underscores the limiting impact of the hardware.

The study concludes that VQE is a conceptually viable and powerful tool for materials simulation, but its practical application to achieve quantum advantage will critically depend on the development of advanced error mitigation techniques.

Keywords: quantum computing, VQE algorithm, graphene, band structure, tight-binding, quantum simulation.

1. Introducción

El descubrimiento de materiales con propiedades electrónicas novedosas ha sido históricamente un motor de innovación tecnológica (Callister, 2018). En este contexto, el grafeno, una monocapa de átomos de carbono dispuestos en una red hexagonal (Wallace, 1947), ha emergido como uno de los materiales más prometedores del siglo XXI (Geim, 2009). Desde su aislamiento en 2004 (Novoselov, 2004), sus extraordinarias propiedades, como una movilidad de portadores de carga ultraalta (CastroNeto, 2009) y una conductividad excepcional (Morozov, 2008), lo han posicionado como un candidato ideal para revolucionar la electrónica (Schwierz, 2010) y la optoelectrónica (Bonaccorso, 2010).

La clave para comprender y explotar el potencial del grafeno reside en el conocimiento detallado de su estructura de bandas electrónicas (Wang, 2023). Esta describe los niveles de energía permitidos para los electrones y determina fundamentalmente el comportamiento eléctrico y óptico del material (Kittel, 2005). Sin embargo, el cálculo preciso de esta estructura presenta desafíos computacionales significativos (Trevisanutto, 2008). Métodos clásicos consolidados, como la Teoría del Funcional de la Densidad (DFT, por sus siglas en inglés; Jones, 2015), aunque exitosos en muchos ámbitos, se basan en aproximaciones que pueden limitar su precisión, especialmente al tratar con sistemas de muchos cuerpos o con fuertes correlaciones electrónicas (Fan, 2021).

Paralelamente, la computación cuántica se perfila como un nuevo paradigma con el potencial de superar las limitaciones de la simulación clásica (Feynman, 1982). Los algoritmos cuánticos, al operar bajo los mismos principios de la mecánica cuántica que rigen el comportamiento de la materia, ofrecen una vía natural para modelar sistemas moleculares y de materiales con una fidelidad sin precedentes (McArdle, 2020). Dentro del arsenal de herramientas cuánticas (Nielsen y Chuang, 2011), el algoritmo Variational Quantum Eigensolver (VQE, por sus siglas en inglés) destaca por su diseño híbrido cuántico-clásico (Peruzzo, 2014). Esta característica lo hace especialmente adecuado para los ordenadores cuánticos de la era actual, dispositivos inherentemente ruidosos y de escala intermedia (NISQ, por su sigla en inglés; Preskill, 2018), ya que mitiga los requisitos de coherencia y profundidad de los circuitos, convirtiéndolo en una herramienta práctica y de gran interés para la investigación a corto plazo, como demuestran sus implementaciones en hardware real (Kandala, 2017).

1.1. Motivación del tema a tratar

La motivación principal de este Trabajo es explorar la intersección de estos dos campos de vanguardia: la ciencia de materiales y la computación cuántica. Se busca aplicar una de las herramientas más prometedoras de la computación cuántica, el algoritmo VQE y su variante VQD (Higgot, 2019), a un problema de fundamental importancia en la física de la materia condensada como es el cálculo de la estructura de bandas del grafeno.

La justificación de esta investigación se sustenta en la necesidad de desarrollar y validar nuevos métodos computacionales que puedan abordar problemas intratables para los superordenadores clásicos (Bauer, 2020; National Academies of Sciences, 2019). Demostrar la viabilidad del VQE/VQD para un sistema tan relevante como el grafeno no solo valida el potencial del algoritmo, sino que también abre la puerta al estudio de materiales más complejos, catalizadores o moléculas, donde una descripción cuántica precisa es indispensable para el diseño y descubrimiento de nuevas tecnologías.

1.2. Planteamiento del Trabajo

El problema central que aborda este Trabajo Fin de Máster es el cálculo de la estructura de bandas del grafeno utilizando el algoritmo VQE. Para ello, se plantea un enfoque metodológico que abarca los siguientes puntos clave:

- **Modelado del sistema:** se utilizará la aproximación de enlace fuerte (tight-binding, en inglés), un modelo eficaz y ampliamente validado para describir la física electrónica del grafeno. Se considerarán dos niveles de complejidad, uno que incluye únicamente las interacciones con los primeros vecinos y otro que incorpora también a los segundos vecinos.
- **Formulación del hamiltoniano:** para cada punto del espacio recíproco (k-space, en inglés), se construirá el hamiltoniano del sistema en forma de matrices de Pauli, lo que permite su implementación directa en un ordenador cuántico.
- **Aplicación del VQE:** se implementará el algoritmo VQE para encontrar los autovalores más bajos del hamiltoniano. Estos cálculos se realizarán en varios puntos de la zona de Brillouin, el espacio matemático que sirve como mapa de los estados de onda de los electrones en la red periódica del grafeno. Concretamente, se calculará la

energía del estado fundamental (correspondiente a la banda de valencia) y del primer estado excitado (la banda de conducción).

- **Simulación y análisis:** la ejecución se llevará a cabo en simuladores cuánticos de la plataforma Qiskit de IBM, evaluando el rendimiento del algoritmo tanto en condiciones ideales como bajo la influencia de un modelo de ruido realista, que emula el comportamiento de un procesador cuántico actual. Los resultados obtenidos se compararán con las soluciones analíticas derivadas del modelo del enlace fuerte para validar su precisión.

1.3. Estructura del Trabajo

El presente trabajo se desarrolla en diferentes capítulos, que siguen una secuencia lógica: desde los conceptos fundamentales hasta las conclusiones de la investigación.

En el Capítulo 2 (Contexto y estado de la técnica) se presenta el marco teórico necesario, comenzando con una introducción al grafeno y la teoría de bandas. A continuación, se describen los métodos clásicos de cálculo y sus limitaciones, para finalmente introducir la computación cuántica y detallar el funcionamiento, los componentes y las ventajas del algoritmo VQE.

En el Capítulo 3 (Objetivos) se definen de manera clara y concisa los objetivos generales y específicos que guían la investigación.

El Capítulo 4 (Desarrollo del Trabajo) expone la metodología seguida, incluyendo la derivación matemática del hamiltoniano del grafeno bajo el modelo tight-binding para los dos casos de estudio (primeros y primeros+segundos vecinos).

Dentro del Capítulo 5 (Resultados y análisis) se presentan los resultados obtenidos de las simulaciones, mostrando gráficamente las bandas de energía calculadas. Se realiza un análisis comparativo entre los resultados del VQE y los valores teóricos, y se discute el impacto del ruido en la precisión del algoritmo.

Con el Capítulo 6 (Conclusiones) se sintetizan los hallazgos más importantes del Trabajo, valorando el cumplimiento de los objetivos y el rendimiento del VQE para el problema estudiado.

Por último, en el Capítulo 7 (Trabajo futuro) se proponen posibles líneas de investigación y mejoras que podrían dar continuidad al trabajo realizado.

2. Contexto y estado de la técnica

Si bien el grafeno fue descrito por primera vez en Wallace (1947), no adquirió la enorme relevancia que tiene actualmente hasta 2004, cuando se consiguió crear grafeno estable en condiciones normales de temperatura y humedad (Novoselov, 2004). El grafeno consiste en una monocapa de átomos de carbono dispuestos en una red hexagonal bidimensional en forma de panal de abeja (CastroNeto, 2009). Su importancia radica en sus propiedades físicas y químicas únicas, y sus excepcionales propiedades electrónicas (Geim, 2007). El comportamiento excepcional de los electrones en el grafeno es la clave de su potencial. Se describen como fermiones de Dirac sin masa, lo que significa que su movimiento se asemeja al de partículas relativistas como la luz: viajan a una velocidad constante y altísima, como si no tuvieran masa (Novoselov, 2005). Este fenómeno se debe a que exhiben una relación de dispersión lineal, es decir, su energía es directamente proporcional a su momento. Dicha relación ocurre en puntos muy específicos conocidos como los puntos de Dirac, que son los vértices de la zona de Brillouin, el mapa fundamental que describe todos los estados de energía y momento posibles para un electrón en la red periódica del cristal (CastroNeto, 2009). Como resultado de este comportamiento único, los electrones del grafeno tienen una movilidad ultraalta y una conductividad eléctrica excepcional (Bolotin, 2008; Morozov, 2008), propiedades que lo convierten en un candidato ideal para una nueva generación de dispositivos electrónicos y optoelectrónicos (Schwierz, 2010; Bonaccorso, 2010).

En cuanto a la relación del grafeno con la computación cuántica, debido a sus propiedades electrónicas únicas, hacen que se convierta en un material idóneo para el desarrollo de ordenadores cuánticos (Carrasco, 2021; Kroll, 2018).

Para poder comprender y explotar estas propiedades del grafeno (y en general, de un sólido) es preciso conocer su estructura de bandas de energía, porque describen la distribución de los niveles de energía disponibles a los electrones como una función del vector de onda dentro del material (que describe la dirección y el momento de la onda del electrón al propagarse por el material; Wang, 2023).

La teoría del funcional de la densidad (DFT) se ha utilizado en las últimas décadas como una de las herramientas indispensables en el cálculo de la estructura electrónica (Fan, 2021). No obstante, la DFT se basa en aproximaciones, lo que puede limitar su precisión en sistemas complejos, especialmente al describir efectos de estados excitados o de correlación electrónica fuerte (un régimen donde la repulsión mutua entre electrones domina su

comportamiento y sus movimientos dejan de ser independientes; Ohgoe, 2024). Alternativamente, métodos de química cuántica de alta precisión, aunque más rigurosos, presentan un coste computacional que escala exponencialmente con el tamaño del sistema, volviéndose rápidamente intratables incluso para los sistemas de tamaño moderado (Sharkey, 2022).

En este contexto, la computación cuántica emerge como una prometedora solución para abordar problemas de estructura electrónica que superan las capacidades de las actuales computadoras clásicas (Preskill, 2018).

2.1. Introducción a la computación cuántica y a la simulación de materiales

La computación cuántica presenta un paradigma computacional radicalmente diferente al clásico (Nielsen y Chuang, 2011). Su potencial para la ciencia de materiales radica en su capacidad inherente para simular sistemas cuánticos complejos (National Academies of Sciences, 2019), lo que podría mejorar la forma en que entendemos, diseñamos y descubrimos nuevos materiales con propiedades avanzadas (Bauer, 2020). Esta capacidad es aplicable a sistemas como el grafeno, donde la estructura electrónica juega un papel fundamental en sus propiedades (CastroNeto, 2009).

2.1.1. Breve descripción de la computación cuántica

La computación cuántica es un paradigma de computación que se fundamenta en los principios de la mecánica cuántica, la teoría física que describe el comportamiento de la materia a escala atómica y subatómica (Akama, 2015). A diferencia de la computación clásica, que utiliza bits para representar información como 0 o 1, la computación cuántica emplea cubits (bits cuánticos; Hughes, 2021). Un cúbit puede existir en una superposición de estados (principio fundamental que permite a un sistema cuántico existir en una combinación de múltiples estados a la vez, en lugar de en un único estado definido), es decir, puede ser simultáneamente 0 y 1 en diferentes proporciones. Esta capacidad de superposición permite a los ordenadores cuánticos codificar y procesar, para un determinado número de cubits, mucha más información que los ordenadores clásicos con esa misma cantidad de bits (Benenti, 2005).

Otro fenómeno cuántico utilizado por la computación cuántica es el entrelazamiento (fenómeno en el que dos o más sistemas cuánticos quedan tan intrínsecamente conectados

que sus destinos están unidos, sin importar la distancia que los separe; Einstein Podolsky Rosen, 1935). Cuando dos o más cubits se entrelazan, sus estados cuánticos se vuelven dependientes entre sí. Medir el estado de un cúbit entrelazado determina instantáneamente el estado de los otros, sin importar lo distantes que puedan estar entre ellos. (Barnett, 2009).

La computación cuántica se basa en la manipulación de cúbits mediante secuencias de operaciones conocidas como puertas cuánticas (Deutsch, 1985). Estas puertas son transformaciones unitarias. Una transformación unitaria es una operación matemática que describe la evolución de un estado cuántico en el tiempo. Su característica fundamental es que preserva la norma (o “longitud”) del vector de estado cuántico, lo que garantiza que la probabilidad total de todos los posibles resultados del sistema se mantenga siempre en 1, un requisito físico indispensable. Además, toda transformación unitaria es reversible, lo que significa que siempre existe una operación inversa que puede deshacerla, lo que garantiza que la evolución del estado cuántico es reversible, y que las distingue de las puertas clásicas irreversibles como AND u OR (Nielsen y Chuang, 2011). La única puerta reversible, tanto en computación cuántica como clásica, es la puerta NOT, consistente en invertir el estado de entrada, es decir 0 pasa a ser 1 y 1 pasa a ser 0 (Toffoli, 1980). Una vez completada la computación, se extrae un resultado clásico a través de una medición, un proceso intrínsecamente irreversible que colapsa la superposición de estados del cúbit a un valor definido, esto es, la medición obliga al sistema a abandonar su estado de múltiples posibilidades simultáneas y a “elegir” un único resultado clásico y definido (por ejemplo, 0 o 1), destruyendo la superposición original (Von Neumann, 1932). La naturaleza probabilística de este colapso implica que una única medición no es suficiente para determinar el resultado. Por ello, es fundamental y necesario repetir el proceso completo múltiples veces para muestrear la distribución de probabilidad de los resultados y obtener así la estadística cuántica que representa la solución (Stolze, 2004).

2.1.2. Potencial de la computación cuántica para la ciencia de materiales

La ciencia de materiales es un campo multidisciplinario que se centra en el estudio de la relación entre la estructura, constituyentes de los materiales y sus propiedades (González-Viñas, 2002). La computación juega un papel crucial en este proceso, permitiendo simular el comportamiento de los materiales a nivel atómico y molecular para predecir sus propiedades sin necesidad de sintetizarlo en el laboratorio (LeSar, 2013). Sin embargo, la simulación precisa de materiales complejos presenta desafíos significativos para la computación clásica,

especialmente cuando se trata de:

- **Sistemas con gran número de partículas:** la mayoría de las propiedades interesantes de los materiales emergen del comportamiento colectivo de un gran número de electrones e iones, donde muchas de sus propiedades provienen de las interacciones colectivas (Anderson, 1972). La complejidad computacional de estos métodos clásicos (como Hartree-Fock (HF), la teoría del funcional de la densidad (DFT) y métodos post Hartree-Fock como la interacción de configuraciones (CI) o el clúster acoplado (CC)) crece de manera polinómica alta o incluso exponencial con el número de electrones y átomos en el sistema (Szabo, 1996).
- **Sistemas fuertemente correlacionados y limitaciones de las formulaciones clásicas:** muchos materiales, como los superconductores de alta temperatura, los materiales magnéticos y los catalizadores, exhiben fuertes interacciones entre sus electrones que dominan el comportamiento del material. Si bien modelos como DFT han demostrado ser muy exitosos para una amplia gama de materiales, sus aproximaciones al funcional de intercambio-correlación a menudo fallan en estos sistemas fuertemente correlacionados (Kohn, 1999).
- **Simulación de la dinámica cuántica:** los métodos de simulación a menudo se basan en resolver la ecuación de Schrödinger (o aproximaciones de ella) para el sistema de muchos cuerpos. La Ecuación de Schrödinger es la ley fundamental del movimiento en la mecánica cuántica que describe cómo evoluciona el estado de un sistema cuántico (como un electrón en un átomo) a lo largo del tiempo (Schrödinger, 1926). La simulación de la evolución temporal de sistemas cuánticos es inherentemente difícil para los métodos clásicos (Levine, 2013).

Aquí es donde la computación cuántica se presenta como una herramienta alternativa con un gran potencial, sobre todo en aspectos como:

- **Simulación cuántica natural:** los ordenadores cuánticos obedecen las mismas leyes de la mecánica cuántica que los sistemas que se desean simular (Feynman, 1982). Los ordenadores cuánticos, al operar bajo los mismos principios de la mecánica cuántica que rigen el comportamiento de los materiales a nivel atómico, ofrecen una forma más natural y potencialmente más eficiente de simular sistemas cuánticos (McArdle, 2020).

- **Escalabilidad para sistemas grandes:** un sistema de n cubits puede existir en una superposición de 2^n estados base simultáneamente (Nielsen y Chuang, 2011). Para simular N electrones con precisión en un ordenador clásico, la cantidad de memoria necesaria escala exponencialmente con N . Un ordenador cuántico podría, en principio, realizar esta simulación con un número de cubits que escala polinómicamente con N (Benenti, 2005). Aunque la computación cuántica aún se encuentra en sus primeras etapas, se espera que a medida que los ordenadores cuánticos crezcan en tamaño y calidad (número de cúbits y coherencia), puedan abordar problemas de ciencia de materiales que son intratables para los métodos clásicos (Abrams, 1997).
- **Explotación del entrelazamiento:** el entrelazamiento, una correlación cuántica que no tiene análogo clásico, es fundamental en los sistemas de muchos cuerpos como los materiales. Los ordenadores cuánticos pueden generar y manipular estados entrelazados de manera nativa, lo que podría permitir simular con mayor precisión las interacciones electrónicas complejas que son difíciles de capturar con métodos clásicos (Lanyon, 2010).
- **Diseño de nuevos materiales con propiedades a medida:** la capacidad de simular con precisión las propiedades de los materiales a nivel fundamental podría permitir el diseño de nuevos materiales con funcionalidades específicas, como superconductividad a temperatura ambiente, mayor eficiencia en celdas solares o catalizadores más selectivos (IBM, 2025).
- **Comprensión de fenómenos cuánticos en materiales:** la computación cuántica podría proporcionar nuevas herramientas para comprender fenómenos cuánticos fundamentales que ocurren en los materiales, como el transporte cuántico (Lanyon, 2011), el magnetismo cuántico (Bernien, 2017) y la topología electrónica (Kitagawa, 2012).

2.1.3. Contexto específico del grafeno

El grafeno, una lámina de carbono de un átomo de espesor con propiedades electrónicas, mecánicas y térmicas excepcionales, es un material de gran interés en la ciencia de materiales (Novoselov, 2004). Sin embargo, la simulación precisa de sus bandas de energía y otras propiedades, especialmente cuando se consideran efectos de correlación o defectos, puede ser desafiante para los métodos clásicos (Sherbert, 2022). La computación cuántica,

y en particular algoritmos como el VQE, ofrecen la promesa de superar estas limitaciones y proporcionar una comprensión más profunda de la estructura electrónica del grafeno, lo que podría conducir al desarrollo de nuevas aplicaciones tecnológicas (Cerezo, 2021).

2.2. La estructura electrónica y las bandas de energía

La estructura electrónica de un material se refiere a la disposición de los electrones dentro de sus átomos y la forma en que estos electrones se organizan y se comportan en el material sólido. Comprender la estructura electrónica es fundamental porque determina la mayoría de las propiedades físicas y químicas de un material, incluyendo su conductividad eléctrica y térmica, sus propiedades ópticas, magnéticas, su estabilidad química y su capacidad para interactuar con otros materiales (Ashcroft, 1976).

En un átomo aislado, los electrones ocupan niveles de energía discretos, conocidos como orbitales atómicos (Bohr, 1913). Cuando los átomos se unen para formar un sólido, estos orbitales atómicos se solapan e interactúan, dando lugar a la formación de bandas de energía (Kohn, 1999). En lugar de tener niveles de energía discretos, los electrones en un sólido ocupan rangos de energía permitidos, las bandas, separadas por rangos de energía prohibidos, las brechas de banda o band gaps (Wilson, 1931).

2.2.1. Formación de las bandas de energía y su relevancia para las propiedades del grafeno

La teoría cuántica establece que los electrones de un determinado átomo ocupan niveles de energía discretos (Bohr, 1913), y el principio de exclusión de Pauli establece que dos electrones de un mismo átomo no pueden tener la misma función de onda, que se traduce en que cada electrón debe tener un conjunto de números cuánticos diferente (Pauli, 1925). Un sólido está compuesto por una gran número de moléculas (o átomos) que no pueden considerarse aislados, ya que su separación es comparable al tamaño de sus moléculas y por tanto la intensidad de las fuerzas que mantienen unido al sólido es del mismo orden de magnitud que las fuerzas que ligan a los átomos para formar moléculas. (Wigner, 1933). Los sólidos cristalinos presentan una regularidad o periodicidad, de modo que estudiando la unidad básica o celda de la red, podemos comprender la estructura del sólido (Bragg, 1913). Para estudiar la estructura electrónica de un cristal podemos considerar que partimos de átomos individuales suficientemente separados y que los aproximamos entre sí. A medida

que los átomos se acercan, los orbitales atómicos externos comienzan a solaparse. Debido al principio de exclusión de Pauli, los niveles de energía atómicos originales se desdoblan en un gran número de niveles de energía muy cercanos entre sí. (Slater, 1954).

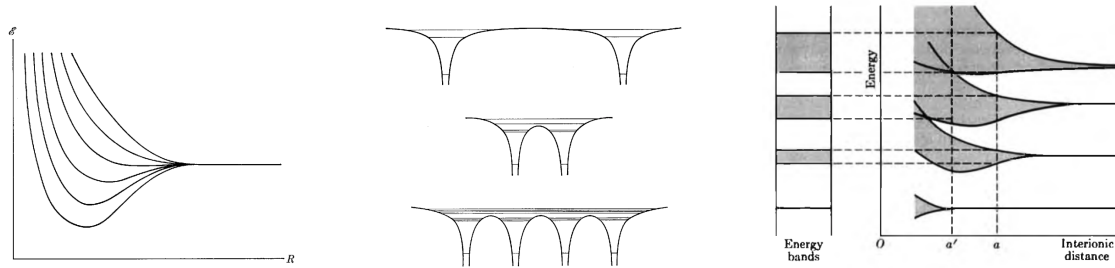


Figura 2.1: Niveles de energía en función de la distancia de separación entre átomos. *Izquierda*: Desdoblamiento de un nivel de energía. *Centro*: Niveles de energía para varios átomos. *Derecha*: Formación de las bandas de energía. Fuente: Eisberg (1979); Alonso Finn (1976)

El número de niveles de energía en una banda es igual al número de orbitales atómicos que contribuyen a esa banda, multiplicado por el número de átomos en el cristal (Slater, 1954). Para un sólido macroscópico, este número es enorme, y los niveles de energía dentro de una banda se consideran prácticamente continuos (Kohn, 1999).

Las bandas, en sus niveles de energía, pueden estar muy separadas, juntarse o incluso solaparse, según el tipo de átomo de que se trate y del tipo de enlace del sólido (Wilson, 1931). Los electrones van ocupando las bandas con menor nivel energético. La banda más alta que contiene electrones se denomina banda de valencia, mientras que la banda más baja en la que hay estados no ocupados se denomina banda de conducción. El espacio o intervalo de energía entre dos bandas permitidas se denomina banda prohibida o gap energético (Iupac, 2019).

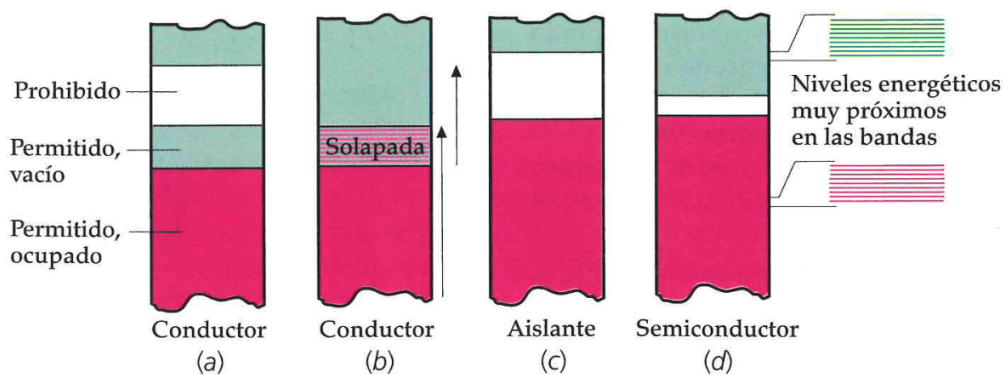


Figura 2.2: Posibles estructuras de bandas para un sólido. Fuente: Tipler (2010)

En la Figura 2.2 se presentan algunas estructuras de bandas para un sólido: (a) muestra la estructura de bandas para un conductor como el cobre donde la banda de valencia no está completamente llena; (b) muestra un conductor como magnesio, donde se solapan dos bandas y generan una banda de valencia incompleta; (c) muestra la estructura de bandas de un aislante típico con su banda de valencia completa y seguida de una banda prohibida ancha; y (d) muestra una estructura de bandas de un semiconductor, donde la banda de valencia está llena, pero la banda prohibida no es muy ancha por lo que una pequeña excitación puede hacer pasar algún electrón a la banda de conducción.

En los sólidos metálicos, en los que la banda de valencia está solapada con la banda de conducción, los electrones poseen gran movilidad ya que cada electrón está sometido al mismo tiempo a la influencia de muchos núcleos y por tanto no está ligado a ninguno en particular (Sommerfeld, 1928). Es por tanto razonable, en primera aproximación, despreciar los efectos individuales de los núcleos y suponer a los electrones sometidos a un potencial promedio generado por caja de paredes rígidas dentro de la cual los electrones se mueven libremente. Se trata del modelo de electrones libres. Si suponemos que la red es suficientemente grande, se pueden ignorar las condiciones de frontera en sus extremos (Alonso Finn, 1976).

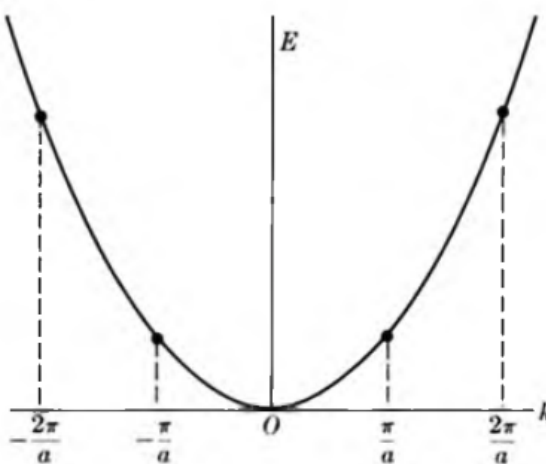


Figura 2.3: Energía de un electrón libre en función del número de onda k . Fuente: Alonso Finn (1976)

La Figura 2.3 presenta la gráfica con los valores permitidos de k uniformemente espaciados para el modelo del electrón libre en una red unidimensional. Para la primera banda del cristal, todos estos valores quedan incluidos en el intervalo $-\pi/a \leq k \leq +\pi/a$, donde a

es a separación entre los iones de la red en la que el electrón se mueve libremente (Eisberg, 1979).

Para ir más allá del modelo de electrones libres y tener en cuenta los efectos debidos a las interacciones de los electrones con la red cristalina, aproximamos el potencial de la estructura de la red a un arreglo de pozos y barreras de potencial con la misma periodicidad de la estructura del sólido (Kroning Penney, 1931). La solución a este problema viene dada por el teorema de Bloch, que establece que las funciones de onda de los electrones en un cristal toman la forma de una onda plana modulada por una función con la misma periodicidad que la red: $\psi(\mathbf{r}) = e^{i\mathbf{k}\mathbf{r}}u(\mathbf{r})$, donde $u(\mathbf{r})$ cumple $u(\mathbf{r} + \mathbf{R}) = u(\mathbf{r})$ para cualquier vector $\mathbf{R}$ de la red de Bravais (Bloch, 1929). Una consecuencia directa de esta periodicidad es que las soluciones solo existen para ciertos rangos de energía (las bandas permitidas), separados por intervalos de energía prohibida (band gaps, en inglés; Kroning Penney, 1931). Esta estructura de bandas determina las propiedades electrónicas fundamentales de un material (Ashcroft, 1976).

En la práctica computacional, existen dos enfoques principales para calcular la estructura de bandas. El primero se basa en la expansión de la función de onda en un conjunto de ondas planas, lo cual es un enfoque natural para electrones muy deslocalizados y funciona especialmente bien para metales simples. El segundo enfoque, conocido como combinación lineal de orbitales atómicos (LCAO, por sus siglas en inglés) o modelo de enlace fuerte (tight-binding), parte de los orbitales atómicos de los átomos que componen el cristal. Este método es más intuitivo para sistemas donde los electrones están más localizados, como en aislantes o moléculas, aunque ambos enfoques se utilizan ampliamente en la práctica (Ashcroft, 1976; Kittel, 2005).

La periodicidad de la red cristalina no solo estructura las energías permitidas, sino también el espacio de momentos (o vectores de onda $\mathbf{k}$) donde se describen. Este espacio, conocido como espacio recíproco, también es periódico (Ewald, 1921). Por conveniencia, toda la información única de la estructura de bandas se puede representar dentro de una celda unitaria de esta red recíproca, conocida como la primera zona de Brillouin (Brillouin, 1939). Esta zona se define como el conjunto de puntos en el espacio recíproco que están más cerca del origen ($\mathbf{k}=0$) que de cualquier otro punto de la red recíproca. En el caso bidimensional del grafeno, la primera zona de Brillouin tiene una forma hexagonal, y sus vértices, los puntos de alta simetría K y K' , son precisamente donde se encuentran los conos de Dirac, lo que subraya la importancia fundamental de este concepto para nuestro estudio

(Wallace, 1947). Los conos de Dirac aparecen en la representación de la energía de los electrones en los diferentes puntos de la red recíproca. En la mayoría de los materiales, esta velocidad es proporcional al cuadrado del momento ($E \propto k^2$, una parábola), sin embargo en los puntos K y K' del grafeno, la energía es lineal ($E \propto |k|$) y de ahí la forma de cono (Geim, 2009).

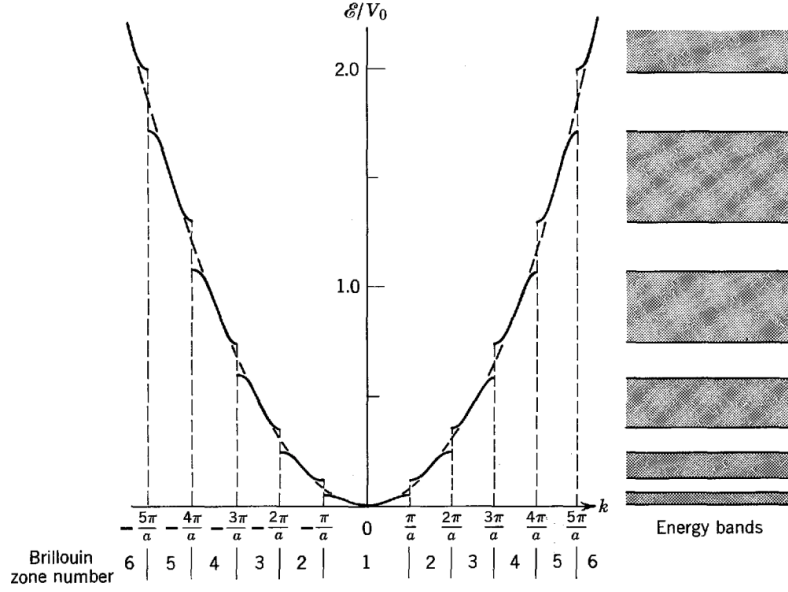


Figura 2.4: Energías permitidas en una red unidimensional de periodicidad a . Fuente: Eisberg (1979)

Para ilustrar la estructura de las zonas y la formación de las bandas, el caso unidimensional es muy clarificador. La Figura 2.4 muestra la relación de dispersión $E(k)$ para una red de periodicidad a . La línea discontinua representa la parábola del electrón libre, mientras que la línea continua muestra cómo el potencial periódico abre 'gaps' de energía en los límites de las zonas de Brillouin ($k = \pm n\pi/a$). La región entre $-\pi/a$ y π/a constituye la primera zona de Brillouin. La segunda zona comprende las regiones $[-\frac{2\pi}{a}, -\frac{\pi}{a}]$ y $[\frac{\pi}{a}, \frac{2\pi}{a}]$, y así sucesivamente. A la derecha de la figura se visualiza cómo esta estructura da lugar a las bandas de energía permitidas (sombreadas) y prohibidas (Eisberg, 1979).

Propiedades del grafeno

En el caso del grafeno, que es un material bidimensional compuesto por una sola capa de átomos de carbono dispuestos en una red hexagonal (Wallace, 1947), la estructura de bandas es particularmente interesante y es la responsable de sus propiedades únicas

(CastroNeto, 2009):

- Conductividad eléctrica: el grafeno tiene una estructura de bandas donde la banda de valencia y la banda de conducción se tocan en puntos específicos llamados puntos de Dirac. En estos puntos, la brecha de banda es cero, lo que permite que los electrones se comporten como partículas sin masa (fermiones de Dirac) y se muevan a velocidades muy altas, dando lugar a su excepcional conductividad eléctrica (Geim, 2007).
- Propiedades ópticas: la estructura de bandas del grafeno también influye en su absorción y transmisión de luz, haciéndolo transparente en un amplio rango del espectro electromagnético a pesar de tener solo un átomo de espesor (Nair, 2008).
- Otras propiedades: las bandas de energía también están relacionadas con las propiedades térmicas (Balandin, 2011), mecánicas (Lee, 2008) y magnéticas (Novoselov, 2005) del grafeno, aunque de manera más indirecta.

2.2.2. La jerarquía de métodos clásicos utilizados para calcular bandas de energía

Para el cálculo de la estructura de bandas de los sólidos, existe en la ciencia de materiales una jerarquía de métodos computacionales utilizados para calcular la estructura electrónica y, por tanto, las bandas de energía de los sólidos, que van desde técnicas de primeros principios (*ab initio*), que no requieren parámetros experimentales, hasta modelos empíricos más sencillos y eficientes (Hohenberg, 1964; Kohn, 1965; Slater, 1954).

Métodos *ab initio*

Por una parte se encuentran los métodos *ab initio*, cuyo objetivo es resolver la ecuación de Schrödinger a partir únicamente de constantes físicas fundamentales. Los dos pilares de esta categoría son el método de Hartree-Fock (HF; Hartree, 1928; Fock, 1930) y, de forma mucho más extendida en la física del estado sólido, la teoría del funcional de la densidad (DFT; Martin, 2020; Jones, 2015). El método de HF es un método que busca aproximar la función de onda electrónica y la energía de un sistema de muchos electrones en estado estacionario. En esta aproximación se considera la interacción entre cada electrón y el núcleo, así como las interacciones entre cada electrón y el resto de electrones mediante un campo promedio creado por todos los demás electrones (la aproximación de campo medio; Martin,

2020). La DFT se basa en la idea de que la energía del estado fundamental de un sistema electrónico está determinada de forma única por su densidad electrónica, tal como indicaba Dirac (1930) y ha demostrado ser una herramienta extraordinariamente exitosa. Sin embargo, ambos enfoques comparten, entre otras, estas importantes limitaciones: su alto coste computacional y sus dificultades inherentes para describir correctamente los sistemas fuertemente correlacionados, donde las interacciones electrón-electrón son dominantes (Jones, 2015).

Métodos empíricos

En el extremo opuesto se encuentran los métodos empíricos, donde se prioriza la eficiencia computacional. Estos modelos utilizan parámetros ajustados para reproducir resultados experimentales conocidos o datos de cálculos *ab initio* más precisos. Un ejemplo es el método del pseudopotencial empírico, donde básicamente se introduce un potencial efectivo, llamado pseudopotencial, que simula el efecto combinado del núcleo y los electrones internos sobre los electrones de valencia en lugar de tratar explícitamente sus interacciones mutuas (Philips, 1959). Si bien son muy rápidos, su poder predictivo está limitado a materiales y condiciones similares a aquellos para los que fueron parametrizados (Ashcroft, 1976).

El modelo tight-binding

Uno de los métodos estándar para resolver los problemas de potencial periódico que se presentan en la teoría de los movimientos electrónicos en sólidos es el método LCAO (combinación lineal de orbitales atómicos), también conocido como método de Bloch o de enlace fuerte (tight-binding, TB; Ashcroft, 1976; Kittel, 2005). La idea fundamental detrás de este modelo es que en lugar de considerar electrones libres moviéndose a través de un potencial periódico débil (como en el modelo de electrones casi libres), los electrones en un sólido están fuertemente ligados a sus átomos individuales, y sus funciones de onda en el sólido pueden aproximarse como una superposición de las funciones de onda de los átomos aislados que componen el sólido. La interacción entre electrones en átomos vecinos se considera como una perturbación que permite a los electrones “saltar” o “tunelar” entre los átomos (Slater, 1954). Este modelo representa un punto intermedio entre los métodos *ab initio* y los métodos empíricos. Por una parte, mantiene una base física clara (los orbitales atómicos), y por otra, la magnitud de estas interacciones o “saltos” (los parámetros del

modelo) a menudo se trata como un parámetro ajustable. Estos pueden ser derivados de un cálculo ab initio o ajustados a datos experimentales, lo que convierte al TB en un método semi-empírico muy flexible (Goringe, 1997). Es por ello que es un método ideal para estudiar la estructura de bandas de muchos materiales y para ser implementado en nuevas plataformas computacionales como los ordenadores cuánticos.

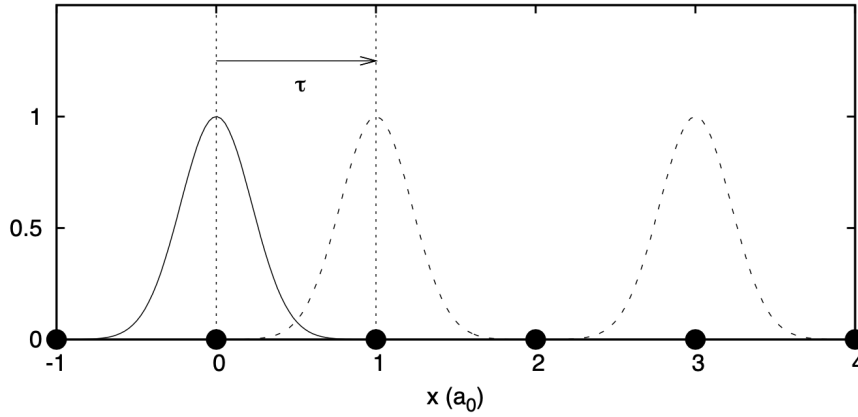


Figura 2.5: Orbitales atómicos en un cristal unidimensional con separación a_0 entre átomos.

Fuente: Roy (2015)

La construcción del modelo seguiría los siguientes pasos (Singleton, 2001):

- Se elige un conjunto de orbitales atómicos relevantes para los átomos que componen el sólido (por ejemplo, los orbitales de valencia).
- Se construyen funciones de Bloch, que respetan la periodicidad de la red cristalina, como combinaciones lineales de estos orbitales atómicos centrados en cada posición de un átomo de la red. La forma general de una función de Bloch es:

$$\psi_{\mathbf{k}}(\mathbf{r}) = \frac{1}{\sqrt{N}} \sum_{\mathbf{R}} e^{i\mathbf{k}\mathbf{R}} \phi(\mathbf{r} - \mathbf{R})$$

donde $\mathbf{k}$ es el vector de onda, $\mathbf{R}$ son los vectores de la red, N es el número de celdas unitarias, y $\phi(\mathbf{r} - \mathbf{R})$ es un orbital atómico centrado en la posición $\mathbf{R}$.

- Se considera el hamiltoniano del cristal, y se calculan los elementos de la matriz del hamiltoniano entre las funciones de Bloch construidas. Estos elementos involucran integrales que representan:

- la energía del orbital atómico en el átomo aislado,

- el solapamiento entre orbitales atómicos en átomos vecinos,
 - y la energía de “salto” o “hopping” de un electrón de un átomo a otro vecino debido al potencial de los átomos circundantes.
- La estructura de bandas $E(k)$ se obtiene diagonalizando la matriz del hamiltoniano para cada valor del vector de onda k en la zona de Brillouin. Los autovalores de esta matriz representan las energías de los electrones en las diferentes bandas.

Este método proporciona soluciones que muestran todas las propiedades de simetría correctas de las bandas de energía (Reich, 2002), y es bastante fácil obtener soluciones para bandas de energía en un punto arbitrario de la zona de Brillouin, mientras que la mayoría de los demás métodos aproximados resultan demasiado difíciles de implementar, excepto en ciertos puntos de simetría de la zona de Brillouin (Slater, 1954). Se sitúa entre los métodos *ab initio*, muy precisos y costosos, y los métodos empíricos, rápidos pero limitados. En comparación con los primeros, TB suele ser de dos a tres órdenes de magnitud más rápido, pero presenta una reducción en la transferibilidad debido a las aproximaciones realizadas (Goringe, 1997); en comparación con los segundos, es de dos a tres órdenes de magnitud más lento, pero conserva la naturaleza mecanocuántica del enlace, lo que garantiza que la naturaleza angular del enlace se describa correctamente lejos de las estructuras de equilibrio (Payne, 1992). Por lo tanto, tight-binding es útil en la gran cantidad de situaciones en las que los efectos mecanocuánticos son significativos, pero el tamaño del sistema hace que los cálculos *ab initio* sean impracticables (Goringe, 1997).

2.3. Introducción al algoritmo Variational Quantum Eigensolver (VQE)

El algoritmo VQE, desarrollado inicialmente por Peruzzo (2014), ha recibido gran atención de la comunidad científica en los últimos años. Utiliza el principio variacional para calcular la energía del estado fundamental de un hamiltoniano (el menor autovalor), un problema fundamental para la química cuántica y la física de la materia condensada (Tilly, 2022).

2.3.1. Explicación del VQE como un algoritmo híbrido cuántico-clásico

En adelante, vamos a utilizar la notación *bra-ket* que definió Dirac (1930), que denomina *ket* a los vectores que se asocian a un estado del sistema en mecánica cuántica y se representa con el símbolo especial $|\rangle$. Al vector complejo conjugado, se le denomina *bra* y su símbolo es $\langle|$. De este modo, el producto escalar entre dos estados, digamos ψ y ϕ , se representa como $\langle\psi|\phi\rangle$.

Partimos del hamiltoniano H y el objetivo consiste en encontrar el menor autovalor E_0 (la menor energía) de H correspondiente al estado fundamental (el estado de menor energía) $|\psi_0\rangle$. La ecuación característica del hamiltoniano H es:

$$H |\psi_i\rangle = E_i |\psi_i\rangle \quad (2.1)$$

donde $|\psi_i\rangle$ es el autovector asociado al autovalor E_i .

El cálculo de la energía E_i correspondiente al autovector $|\psi_i\rangle$ se obtiene mediante el cálculo del valor esperado de H (Sakurai, 1994):

$$\langle\psi_i| H |\psi_i\rangle = \langle\psi_i| E_i |\psi_i\rangle = E_i \langle\psi_i|\psi_i\rangle = E_i \quad (2.2)$$

Por el teorema espectral, podemos escribir el hamiltoniano como (Nielsen y Chuang, 2011):

$$H = \sum_i E_i |\psi_i\rangle\langle\psi_i| \quad (2.3)$$

Y sustituyendo la Ecuación 2.3 en la expresión del valor esperado, obtenemos:

$$\begin{aligned} \langle\psi| H |\psi\rangle &= \langle\psi| \left(\sum_i E_i |\psi_i\rangle\langle\psi_i| \right) |\psi\rangle \\ &= \sum_i E_i \langle\psi|\psi_i\rangle\langle\psi_i|\psi\rangle = \sum_i E_i |\langle\psi, \psi_i\rangle|^2 \end{aligned} \quad (2.4)$$

La Ecuación 2.4 muestra que el valor esperado de H sobre un estado $|\psi\rangle$ puede ser expresado como una combinación lineal de los autovalores de H con todos sus coeficientes iguales o mayores que cero, ya que $|\langle\psi, \psi_i\rangle|^2 \geq 0$, $\forall i$. Por tanto, podemos obtener:

$$\langle\psi| H |\psi\rangle \geq E_0 \quad (2.5)$$

donde E_0 es el menor autovalor de H .

El objetivo del algoritmo VQE (Variational Quantum Eigensolver) consiste en encontrar el estado fundamental de un hamiltoniano dado, que describe, por ejemplo, la energía de un determinado proceso físico o químico. Su estructura general consiste en los siguientes pasos (Peruzzo, 2014):

- preparar un estado cuántico parametrizado,
- medir dicho estado,
- estimar su energía, y
- modificar los parámetros para minimizar dicha estructura;
- entonces, repetir todos estos procesos varias veces hasta alcanzar un criterio de parada.

El algoritmo VQE está incluido dentro de los algoritmos denominados híbridos, ya que parte del mismo se desarrolla en un computador cuántico y otra parte en un ordenador clásico. Un circuito parametrizado o *ansatz* incluye puertas cuánticas (típicamente rotaciones) dependientes de unos parámetros que permiten modificar dichas puertas conforme se varían sus parámetros. El papel del circuito parametrizado consiste en generar el estado candidato $|\psi\rangle$ (Cerezo, 2021). La parte variacional del algoritmo consiste en las mejoras iterativas del estado candidato mediante el ajuste de los parámetros del circuito. Esto es algo que puede realizarse como parte clásica del protocolo híbrido cuántico-clásico (McClean, 2016). La parte cuántica del algoritmo consiste en ejecutar el circuito parametrizado y luego medir H en el estado construido final para obtener el valor esperado de H (Jacquier, 2022).

2.3.2. Componentes principales del VQE

Los componentes principales del VQE son: la función de coste (hamiltoniano), cálculo del valor estimado del hamiltoniano, circuito variacional parametrizado (*ansatz*) y optimizador clásico.

Función de coste (hamiltoniano) y cálculo del valor esperado.

VQE tiene como objetivo determinar la energía del estado fundamental E_0 de un hamiltoniano H . La función de coste se define como $C(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle$, es decir, se busca minimizar el valor esperado de H en un estado de prueba $|\psi(\theta)\rangle = U(\theta) |\psi_s\rangle$ para un circuito parametrizado (*ansatz*) $U(\theta)$ y un estado inicial $|\psi_s\rangle$. Si tenemos en cuenta la Ecuación

2.5, entonces $C(\theta) \geq E_0$, cumpliéndose la igualdad si $|\psi(\theta)\rangle$ es el estado fundamental $|\psi_0\rangle$ de H (Cerezo, 2021).

Cualquier hamiltoniano puede ser escrito como (Jacquier, 2022):

$$H = \sum_{i\alpha} h_{\alpha}^i \sigma_{\alpha}^i + \sum_{ij\alpha\beta} h_{\alpha\beta}^{ij} \sigma_{\alpha}^i \sigma_{\beta}^j + \dots$$

con h real, donde $i, j, \dots$ indica el subespacio en el que actúa el operador, y $\alpha, \beta, \dots$ indica el operador de Pauli. Por ejemplo, si $i = 1$ y $\alpha = x$, entonces $\sigma_x^1 = X$ (puerta NOT) actuando en el cúbit 1. Teniendo en cuenta la linealidad de los observables cuánticos, podemos calcular el valor esperado del hamiltoniano como suma de los valores esperados de los términos individuales (Dirac, 1930):

$$\langle H \rangle = \sum_{i\alpha} h_{\alpha}^i \langle \sigma_{\alpha}^i \rangle + \sum_{ij\alpha\beta} h_{\alpha\beta}^{ij} \langle \sigma_{\alpha}^i \sigma_{\beta}^j \rangle + \dots \quad (2.6)$$

Consideramos hamiltonianos que pueden escribirse como un número polinómico de términos, con respecto al tamaño del sistema. Esta clase de hamiltonianos abarca una amplia gama de sistemas físicos, incluyendo el hamiltoniano de estructura electrónica de la química cuántica, el modelo cuántico de Ising, el modelo de Heisenberg, matrices que se aproximan bien como una suma de n productos tensoriales y, de forma más general, cualquier hamiltoniano k -disperso (se denomina k -escaso si hay como máximo k elementos distintos de cero en cada fila y columna de la matriz), sin una estructura de producto tensorial evidente. Por lo tanto, la evaluación de $\langle H \rangle$ se reduce a la suma de un número polinómico de valores esperados de operadores de Pauli simples para un estado cuántico $|\psi\rangle$, multiplicado por algunas constantes reales (Peruzzo, 2014).

Como resultado, la función de coste $C(\theta)$ se obtiene a partir de una combinación lineal de valores esperados de σ_{α} , y por tanto puede estimarse eficientemente en ordenadores cuánticos con un coste computacional que suele crecer, como máximo, de forma polinómica con el tamaño del sistema (Cerezo, 2021).

Circuito variacional parametrizado (ansatz).

El objetivo del circuito parametrizado o *ansatz* es la generación de estados candidatos que cubran la totalidad del espacio de Hilbert de forma uniforme y sin que se concentren en ninguna región en particular (Jacquier, 2022).

La estructura específica del ansatz puede depender del problema concreto a resolver ('problem-inspired-ansatz', en inglés) o tener una estructura genérica válida para cualquier problema a resolver ('problem-agnostic', en inglés; Cerezo, 2021). Entre las primeras podemos encontrar los métodos de crecimiento iterativo (ADAPT-VQE; Grimsley, 2019) o clúster de cubit acoplado iterativo (iQCC, por sus siglas en inglés; Ryabinkin, 2019). Entre la últimas, podemos citar de hardware eficiente (HEA; Kandala, 2017) o de agrupamiento unitario acoplado (UC; Peruzzo, 2014).

Un ansatz básico para un único cúbit puede construirse mediante una puerta $R_Y(\theta_1)$ seguida de una puerta $R_Z(\theta_2)$. Si partimos del estado inicial $|0\rangle$, podemos alcanzar cualquier otro estado de la esfera de Bloch mediante una rotación sobre el eje Y , seguida de una rotación del eje Z , es decir. mediante los dos parámetros θ_1 y θ_2 (Jacquier, 2022).

Optimizador clásico.

VQE es un algoritmo híbrido, lo que significa que utiliza lo más eficiente de la computación cuántica y la computación clásica. Una de las partes más importantes del algoritmo es la optimización, que por eficiencia y pragmatismo, se lleva a cabo mediante un computador clásico, el cual define gran parte de la eficiencia del algoritmo en su conjunto. Uno de los enfoques más comunes para la optimización consiste en realizar pasos iterativos en las direcciones indicadas por el gradiente de la función coste. Una primera gran división de los optimizadores puede establecerse según que utilice o no alguna versión del descenso de gradiente (Peruzzo, 2014; McClean, 2016; Kandala, 2017; Cerezo, 2021; Tilly, 2022).

El método del gradiente descendente es un algoritmo que permite alcanzar el mínimo (o el máximo) de una función diferenciable siguiendo la dirección de mayor (menor) valor del vector gradiente $\nabla f(x) = \left(\left. \frac{\partial f}{\partial x_1} \right|_x, \dots, \left. \frac{\partial f}{\partial x_n} \right|_x \right)$ de la función (Cauchy, 1847; Robbins, 1951). Sería análogo a dar pequeños pasos en la dirección descendente (ascendente) más empinada de una montaña (Combarro, 2023).

Entre los algoritmos que utilizan el método del gradiente descendente podemos citar la aproximación estocástica de diferencias finitas (FDSA; Spall, 1991), la aproximación estocástica de perturbación simultánea (SPSA; Spall, 1992), el estimador adaptativo del momento (Adam; Kingma, 2015), o el algoritmo Broyden-Fletcher-Goldfarb-Shanno (BFGS; Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). Entre los algoritmos que no utilizan el método del gradiente descendente podemos citar el algoritmo Nelder-Mead (Nelder-Mead, 1965), o la optimización restringida por aproximación lineal (COBYLA; Powell,

1994).

2.3.3. Cálculo de estados excitados (VQD)

El algoritmo VQE se utiliza para obtener el estado fundamental de un hamiltoniano. Con pequeñas modificaciones del algoritmo, es posible obtener los estados excitados (de mayor energía) de dicho hamiltoniano (McArdle, 2020). El algoritmo de deflación cuántica variacional (VQD), desarrollado por Higgot (2019), extiende VQE para encontrar sistemáticamente estados excitados prácticamente sin coste adicional. Como las matrices hermíticas admiten un conjunto completo de vectores propios ortogonales (Gilbert, 2010) y como VQE conserva los parámetros clásicos de los estados ansatz que permiten su reconstrucción (Peruzzo, 2014), añadiendo términos de superposición a la función de optimización, podremos calcular estos estados excitados utilizando los circuitos cuánticos de baja profundidad (Higgot, 2019).

El algoritmo VQD se basa en el siguiente desarrollo (Higgot, 2019; Combarro, 2023). Sea $\lambda_0 = E_0$ la energía del estado fundamental correspondiente a un hamiltoniano H dado, calculada mediante VQE. Consideremos el hamiltoniano modificado

$$H' = H + C |\psi_0\rangle\langle\psi_0| \quad (2.7)$$

donde C es un número real positivo y H' es hermítica ya que es la suma de dos matrices hermíticas.

Para un estado genérico $|\psi\rangle$, el valor esperado de H' será

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C \langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle = \langle\psi|H'|\psi\rangle + C |\langle\psi_0|\psi\rangle|^2 \quad (2.8)$$

donde $C |\langle\psi_0|\psi\rangle|^2$ es un valor no negativo que cuantifica el solapamiento entre ψ y ψ_0 . Si $|\psi\rangle = |\psi_0\rangle$ este término tendrá valor C , pero si $|\psi\rangle$ es ortogonal a $|\psi_0\rangle$, este término será 0.

Sean $\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_n$ los autovalores de H correspondientes a la base de autovectores $\{|\lambda_j\rangle\}$ (puede haber autovalores repetidos). Por hipótesis $|\lambda_0\rangle = |\psi_0\rangle$. Si $j \neq 0$:

$$H' |\lambda_j\rangle = H |\lambda_j\rangle + C |\psi_0\rangle\langle\psi_0|\lambda_j\rangle = H |\lambda_j\rangle + C |\lambda_0\rangle\langle\lambda_0|\lambda_j\rangle = H |\lambda_j\rangle = \lambda_j |\lambda_j\rangle \quad (2.9)$$

ya que $|\lambda_0\rangle$ y $|\lambda_j\rangle$ son ortogonales. Por otra parte:

$$H' |\lambda_0\rangle = H |\lambda_0\rangle + C |\psi_0\rangle\langle\psi_0|\lambda_0\rangle = H |\lambda_0\rangle + C |\lambda_0\rangle\langle\lambda_0|\lambda_0\rangle = H |\lambda_0\rangle + C |\lambda_0\rangle = (\lambda_0 + C) |\lambda_0\rangle \quad (2.10)$$

Como $\langle \lambda_j | H' | \lambda_j \rangle = \lambda_j$ cuando $j \neq 0$ y $\langle \lambda_0 | H' | \lambda_0 \rangle = C + \lambda_0$, si $C > \lambda_1 - \lambda_0$, entonces $|\psi_0\rangle = |\lambda_0\rangle$ no puede ser el estado fundamental de H' , ya que la energía de $|\lambda_1\rangle$ sería menor que la de $|\psi_0\rangle$. Aplicando VQE al hamiltoniano H' obtendremos su estado fundamental, por lo que $|\lambda_1\rangle$ debe ser el estado fundamental de H' . Una vez obtenido λ_1 podemos considerar $H'' = H' + C' |\lambda_1\rangle\langle\lambda_1|$ y utilizar VQE para obtener $|\lambda_2\rangle$, y así, de forma iterativa, obtener los siguientes valores de la energía de los estado excitados de H .

2.3.4. Ventajas y limitaciones del VQE

El enfoque variacional nos permite resolver problemas de optimización difíciles codificados en el hamiltoniano con el modelo de puertas de la computación cuántica, una alternativa a la computación cuántica adiabática, ya que no todos los problemas de optimización se pueden formular de manera eficiente en la formulación QUBO (Jacquier, 2022).

La VQE permite la medición de funciones de onda aproximadas (Peruzzo, 2014), que no pueden muestrearse ni calcularse eficientemente (por ejemplo, en tiempo polinomial) en dispositivos convencionales a medida que aumenta el tamaño del sistema (Feynman, 1982). Esto implica que los hamiltonianos estudiados pueden escribirse como una suma polinomial creciente de observables independientes (McArdle, 2020), como se encuentra comúnmente en diversos campos como la química cuántica (Kandala, 2017), la física de la materia condensada (Georgescu, 2014) o la física nuclear (Dumitrescu, 2018). Cuando estas formas de función de onda se pueden calcular con suficiente precisión en su aproximación al estado fundamental, entonces VQE se convierte en un método prácticamente útil para calcular las propiedades de los sistemas cuánticos (Google, 2020).

Con respecto a otros algoritmos como el algoritmo de estimación cuántica de fase (QPE; Abrams, 1999), éste último ofrece una aceleración exponencial sobre los métodos clásicos y requiere un número de operaciones cuánticas $\mathcal{O}(1/p)$ para obtener una estimación con precisión p , lo que implica circuitos cuánticos de gran profundidad, es decir, compuestos por una larga secuencia de operaciones o puertas cuánticas. En los dispositivos actuales, cada una de estas puertas tiene una pequeña probabilidad de error y los qubits solo pueden mantener su estado cuántico durante un tiempo muy limitado (su ‘tiempo de coherencia’). Por tanto, un circuito con muchas puertas se vuelve impracticable, ya que los errores se acumulan y la información se pierde por decoherencia antes de que la computación termine. En contraste, el algoritmo VQE puede ser implementado con un número de puertas mucho menor, lo que lo hace suficientemente robusto para ser aplicable en los computadores

cuánticos actuales de la era NISQ. (Sherbert, 2022). Es posible encontrar demostraciones experimentales sobre esta limitación y sobre la relativa robustez del VQE en los computadores NISQ disponibles en la actualidad (O'Malley, 2016).

La VQE implica resolver un problema de optimización, por lo que el coste real y el escalamiento dependen del optimizador y del entorno de optimización del problema específico estudiado. Por tanto, encontrar la solución óptima al problema dependerá de encontrar un optimizador que converja en tiempo razonable. La pregunta es si la VQE puede optimizarse heurísticamente en un número polinomial de iteraciones y converger a una solución aproximada pero suficientemente precisa (Tilly, 2022).

2.4. Medición de bandas de energía del grafeno con VQE

Aunque se encuentran trabajos dedicados al estudio de la estructura de bandas del grafeno, unos se dedican al estudio de la bicapa de grafeno (Wang, 2022; Korschuh, 2012), otros utilizan diferentes métodos que el algoritmo VQE para el cálculo de la estructura de bandas (Wang, 2023), y por último, los hay que están focalizados en el algoritmo VQE y el grafeno aparece como un ejemplo más de aplicación (Sherbert, 2022).

El objetivo es la medida de la estructura de bandas del grafeno, que describe los niveles de energía (autovalores), para este sistema periódico infinito (CastroNeto, 2009). Sin embargo, los algoritmos cuánticos actuales calculan sistemas finitos, es decir, un número limitado de qubits pueden representar un limitado número de niveles de energía.

El enfoque más directo aplicable al problema, consiste en simular un cluster finito de grafeno (unos pocos hexágonos) y construir su hamiltoniano mediante TB. Utilizando el algoritmo VQE se puede calcular el estado fundamental para cualquier punto k de la zona de Brillouin. Este proceso se puede repetir para diferentes puntos k .

El cálculo de las bandas superiores se puede realizar mediante algoritmos como VQD (Higgot, 2019) o VQE con búsqueda de subespacios (SSVQE; Nakanishi, 2019), entre otros. En este Trabajo se va a utilizar el primero de ellos (VQD).

3. Objetivos

Para guiar el desarrollo de este trabajo y asegurar que se aborden de manera sistemática los problemas planteados en la introducción, se han definido un objetivo general y una serie de objetivos específicos.

3.1. Objetivos generales

El objetivo general de este Trabajo Fin de Máster es evaluar la viabilidad y el rendimiento del algoritmo Variational Quantum Eigensolver (VQE) como herramienta para el cálculo de la estructura de bandas electrónicas del grafeno, utilizando simuladores cuánticos para determinar su precisión y potencial frente a los métodos teóricos establecidos.

3.2. Objetivos específicos

Para la consecución del objetivo general, se plantean los siguientes objetivos específicos, que estructuran las fases de la investigación:

1. Desarrollar el modelo teórico del hamiltoniano del grafeno utilizando la aproximación de enlace fuerte (tight-binding, en inglés), considerando tanto las interacciones con los primeros vecinos como la inclusión de los segundos vecinos, y formularlo en términos de las matrices de Pauli para su implementación en un circuito cuántico.
2. Implementar un programa en Python, utilizando la librería Qiskit de IBM, que aplique el algoritmo VQE para calcular los autovalores del hamiltoniano en distintos puntos de la zona de Brillouin, configurando para ello el ansatz variacional y el optimizador clásico.
3. Calcular la banda de valencia (estado fundamental) y la banda de conducción (primer estado excitado) del grafeno a lo largo de las trayectorias de alta simetría de la zona de Brillouin (Γ -K-M- Γ), reconstruyendo así la estructura de bandas del material.
4. Analizar y comparar los resultados obtenidos mediante VQE con las soluciones analíticas del modelo TB para validar la precisión del algoritmo en un entorno de simulación ideal.

5. Evaluar el impacto del ruido en el rendimiento del algoritmo, ejecutando simulaciones que emulen el comportamiento de un procesador cuántico real y cuantificando la desviación de los resultados respecto a los valores esperados.

4. Desarrollo del Trabajo

4.1. Introducción

La estructura electrónica de un átomo de carbono aislado es $(1s)^2(2s)^2(2p)^2$ (Ruiz, 2017). En su estado sólido, los electrones $1s$ permanecen más o menos inertes, mientras que los electrones $2s$ y $2p$ se hibridan (CastroNeto, 2009). Una posible configuración electrónica del carbono es la formación de cuatro orbitales híbridos sp^3 , que tienden a organizarse en un patrón de enlaces tetraédrico, ocupando todos los electrones de valencia. Esto es justamente lo que ocurre en el diamante, la forma sólida más conocida del carbono, que actúa como un excelente aislante con una banda prohibida de aproximadamente 5 eV (Geis, 1992). Otra alternativa es la hibridación sp^2 , en la que se forman tres orbitales híbridos y queda un orbital p sin hibridar. En este caso, los orbitales sp^2 se sitúan en un plano formando ángulos de 120° , y la estructura resultante es la red en forma de panel de abeja del grafeno (Geim, 2007).

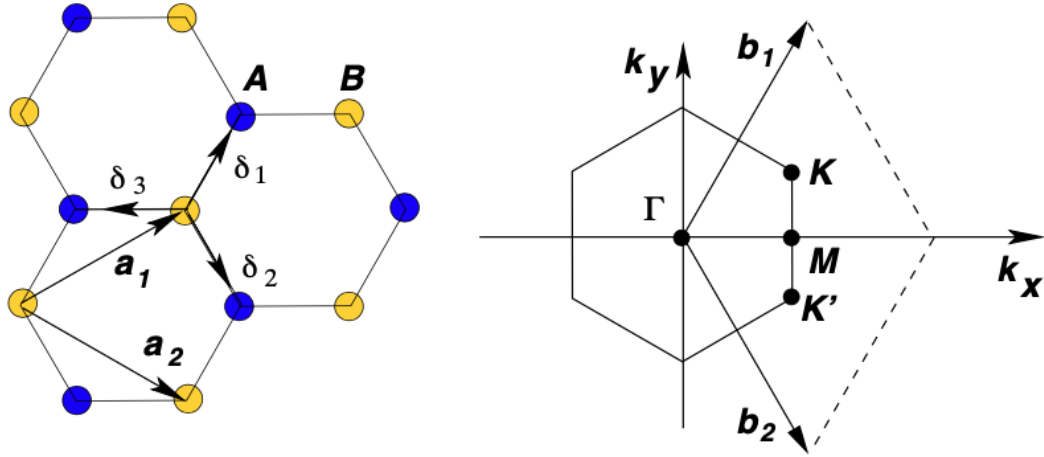


Figura 4.1: Estructura en panel de abeja del grafeno y su zona de Brillouin. *Izquierda:* Estructura reticular del grafeno, compuesta por dos redes triangulares interpenetrantes ($\mathbf{a}_1$ y $\mathbf{a}_2$ son los vectores unitarios de la red, y δ_i , $i = 1, 2, 3$ son los vectores vecinos más cercanos). *Derecha:* Zona de Brillouin correspondiente. Los conos de Dirac se ubican en los puntos K y K' . Fuente: CastroNeto (2009)

La Figura 4.1 muestra dos subredes no equivalentes (A y B) donde los entornos de los átomos correspondientes son imágenes especulares entre sí. Los vectores primitivos de la

estructura anterior se definen como:

$$\mathbf{a}_1 = \frac{a}{2}(3, \sqrt{3}), \quad \mathbf{a}_2 = \frac{a}{2}(3, -\sqrt{3}) \quad (4.1)$$

donde $a \approx 1,42 \text{ \AA}$ es la distancia entre los dos átomos de carbono más próximos.

Para un átomo en la subred B, los vectores de los tres vecinos más cercanos en el espacio real están dados por:

$$\boldsymbol{\delta}_1 = \frac{a}{2}(1, \sqrt{3}), \quad \boldsymbol{\delta}_2 = \frac{a}{2}(1, -\sqrt{3}), \quad \boldsymbol{\delta}_3 = -a(1, 0) \quad (4.2)$$

mientras que los de la subred A son los negativos de estos.

Los siguientes seis vecinos más cercanos están localizados en:

$$\boldsymbol{\delta}'_1 = \pm \mathbf{a}_1, \quad \boldsymbol{\delta}'_2 = \pm \mathbf{a}_2, \quad \boldsymbol{\delta}'_3 = \pm(\mathbf{a}_2 - \mathbf{a}_1) \quad (4.3)$$

Definimos la primera zona de Brillouin (FBZ, por sus siglas en inglés) de la red recíproca como la zona delimitada por los planos que bisecan los vectores hasta los puntos recíprocos más cercanos. Esto da como resultado una FBZ con la misma forma que los hexágonos originales de la red de panal de abeja, pero rotada $\pi/2$. Los vectores de esta estructura recíproca vienen dados por:

$$\mathbf{b}_1 = \frac{2\pi}{3a}(1, \sqrt{3}), \quad \mathbf{b}_2 = \frac{2\pi}{3a}(1, -\sqrt{3}) \quad (4.4)$$

Por último, los puntos K, K', Γ, M de la estructura recíproca se encuentran en:

$$\mathbf{K} = \frac{2\pi}{3\sqrt{3}a}(\sqrt{3}, 1), \quad \mathbf{K}' = \frac{2\pi}{3\sqrt{3}a}(\sqrt{3}, -1), \quad \mathbf{\Gamma} = (0, 0), \quad \mathbf{M} = \frac{2\pi}{3\sqrt{3}a}(\sqrt{3}, 0) \quad (4.5)$$

4.2. Aproximación primeros vecinos

El modelo de enlace fuerte (tight-binding, en inglés) se define por una serie de parámetros que cuantifican las interacciones electrónicas (Slater, 1954). Los más importantes son las energías en el sitio (on-site energies, en inglés), que representan la energía de un electrón en un átomo aislado, y los parámetros de salto (hopping integrals, en inglés), que describen la probabilidad de que un electrón “salte” a un átomo vecino (salto entre diferentes subredes; Leggett, 2010). En el modelo más simple para el grafeno, se asume que la energía en el sitio es la misma para todos los átomos de carbono (y se puede fijar a cero como referencia de energía para establecer la escala general de la banda de energía derivada de π) y solo se considera el parámetro de salto t entre los vecinos más cercanos (Geim, 2007).

Para obtener el valor de este parámetro t existen principalmente dos enfoques, que reflejan la posición del modelo TB como un método entre la teoría y el experimento:

- ajuste a datos experimentales (método empírico). El parámetro t se trata como un valor ajustable. Se calcula una propiedad observable a partir del modelo (como la velocidad de Fermi de los electrones, que es directamente proporcional a t) y se ajusta su valor hasta que la predicción del modelo coincida con los resultados medidos experimentalmente, por ejemplo, mediante técnicas como la espectroscopía de fotoemisión resuelta en ángulo (ARPES, por sus siglas en inglés). Este enfoque da un valor comúnmente aceptado de $t \approx 2,8\text{eV}$ (Bostwick, 2007).
- derivación desde primeros principios (método *ab initio*). Se realiza un cálculo completo de la estructura de bandas del grafeno utilizando un método más fundamental y sin parámetros, como la DFT. Luego, se ajusta el modelo de enlace fuerte simple a las bandas obtenidas por DFT, encontrando el valor de t que mejor reproduce la dispersión de energía cerca de los puntos de Dirac. Este proceso permite crear un modelo simple y efectivo a partir de uno complejo (Goringe, 1997).

Si consideramos únicamente la interacción con los tres vecinos más cercanos, el hamiltoniano del enlace fuerte para el grafeno es (CastroNeto, 2009):

$$\hat{H} = -t \sum_{\langle ij \rangle} (\hat{a}_i^\dagger \hat{b}_j + \hat{b}_j^\dagger \hat{a}_i) \quad (4.6)$$

donde $i(j)$ se refiere a la subred A(B), el operador fermiónico $\hat{a}_i^\dagger \hat{a}_i$ crea (aniquila) un electrón en la subred A cuya posición es $\mathbf{r}_i$, y de forma similar para $\hat{b}_j^\dagger \hat{b}_j$. Podemos reescribir la suma sobre los vecinos más cercanos como:

$$\sum_{\langle ij \rangle} (\hat{a}_i^\dagger \hat{b}_j + \hat{b}_j^\dagger \hat{a}_i) = \sum_{i \in A} \sum_{\boldsymbol{\delta}} (\hat{a}_i^\dagger \hat{b}_{i+\boldsymbol{\delta}} + \hat{b}_{i+\boldsymbol{\delta}}^\dagger \hat{a}_i) \quad (4.7)$$

donde la suma sobre $\boldsymbol{\delta}$ se produce sobre los vectores vecinos más cercanos $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$ y $\boldsymbol{\delta}_3$, y el operador $\hat{b}_{i+\boldsymbol{\delta}}^\dagger$ aniquila un fermión de la subred B en la posición $\mathbf{r}_i + \boldsymbol{\delta}$. Si tenemos en cuenta que

$$\hat{a}_i^\dagger = \frac{1}{\sqrt{N/2}} \sum_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{r}_i} \hat{a}_{\mathbf{k}}^\dagger \quad (4.8)$$

donde $N/2$ es el número de elemento de A, y haciendo lo misma para $\hat{b}_{i+\boldsymbol{\delta}}^\dagger$, podemos

reescribir la Ecuación 4.6 como:

$$\begin{aligned}\hat{H} &= -\frac{t}{N/2} \sum_{i \in A} \sum_{\delta, \mathbf{k}, \mathbf{k}'} [e^{i(\mathbf{k}-\mathbf{k}') \cdot \mathbf{r}_i} e^{-i\mathbf{k}' \cdot \delta} \hat{a}_{\mathbf{k}}^\dagger \hat{b}_{\mathbf{k}'} + e^{-i(\mathbf{k}-\mathbf{k}') \cdot \mathbf{r}_i} e^{i\mathbf{k}' \cdot \delta} \hat{b}_{\mathbf{k}}^\dagger \hat{a}_{\mathbf{k}'}] \\ &= -t \sum_{\delta, \mathbf{k}} (e^{-i\mathbf{k} \cdot \delta} \hat{a}_{\mathbf{k}}^\dagger \hat{b}_{\mathbf{k}} + e^{i\mathbf{k} \cdot \delta} \hat{b}_{\mathbf{k}}^\dagger \hat{a}_{\mathbf{k}})\end{aligned}\quad (4.9)$$

donde se ha aplicado que:

$$\sum_{i \in A} e^{i(\mathbf{k}-\mathbf{k}') \cdot \mathbf{r}_i} = \frac{N}{2} \delta_{\mathbf{k}\mathbf{k}'} \quad (4.10)$$

Se puede expresar el hamiltoniano de una forma más compacta:

$$\hat{H} = \sum_{\mathbf{k}} \Psi^\dagger \mathbf{h}(\mathbf{k}) \Psi \quad (4.11)$$

donde

$$\Psi \equiv \begin{pmatrix} \hat{a}_{\mathbf{k}} \\ \hat{b}_{\mathbf{k}} \end{pmatrix}, \quad \Psi^\dagger \equiv \begin{pmatrix} \hat{a}_{\mathbf{k}}^\dagger & \hat{b}_{\mathbf{k}}^\dagger \end{pmatrix} \quad (4.12)$$

y la representación matricial del hamiltoniano es

$$\mathbf{h}(\mathbf{k}) \equiv -t \begin{pmatrix} 0 & \Delta_{\mathbf{k}} \\ \Delta_{\mathbf{k}}^* & 0 \end{pmatrix}, \quad \Delta_{\mathbf{k}} \equiv \sum_{\delta} e^{i\mathbf{k} \cdot \delta} \quad (4.13)$$

4.2.1. Bandas de energía

Las bandas de energía se corresponderán con los autovalores de la matriz del hamiltoniano según la Ecuación 4.13 (Bloch, 1929). Estos autovalores serán $E_{\pm} = \pm \sqrt{\Delta_{\mathbf{k}}^* \Delta_{\mathbf{k}}}$. Si desarrollamos $\Delta_{\mathbf{k}}$:

$$\begin{aligned}\Delta_{\mathbf{k}} &= e^{i\mathbf{k} \cdot \delta_1} + e^{i\mathbf{k} \cdot \delta_2} + e^{i\mathbf{k} \cdot \delta_3} \\ &= e^{i\mathbf{k} \cdot \delta_3} [1 + e^{i\mathbf{k} \cdot (\delta_1 - \delta_3)} + e^{i\mathbf{k} \cdot (\delta_2 - \delta_3)}] \\ &= e^{-ik_x a} [1 + e^{i3k_x a/2} e^{i\sqrt{3}k_y a/2} + e^{i3k_x a/2} e^{-i\sqrt{3}k_y a/2}] \\ &= e^{-ik_x a} [1 + e^{i3k_x a/2} (e^{i\sqrt{3}k_y a/2} + e^{-i\sqrt{3}k_y a/2})] \\ &= e^{-ik_x a} [1 + 2e^{i3k_x a/2} \cos(\sqrt{3}k_y a/2)]\end{aligned}\quad (4.14)$$

De este modo, las bandas de energía se pueden expresar como:

$$\begin{aligned}E_{\pm}(\mathbf{k}) &= \pm t \sqrt{1 + 4 \cos\left(\frac{3}{2}k_x a\right) \cos\left(\frac{\sqrt{3}}{2}k_y a\right) + 4 \cos^2\left(\frac{\sqrt{3}}{2}k_y a\right)} \\ &= \pm t \sqrt{3 + f(\mathbf{k})},\end{aligned}\quad (4.15)$$

$$f(\mathbf{k}) = 4 \cos\left(\frac{3}{2}k_x a\right) \cos\left(\frac{\sqrt{3}}{2}k_y a\right) + 2 \cos(\sqrt{3}k_y a) \quad (4.16)$$

La Figura 4.2 presenta dos imágenes de las gráficas correspondientes a las bandas de energía E_+ (azul) y E_- (naranja).

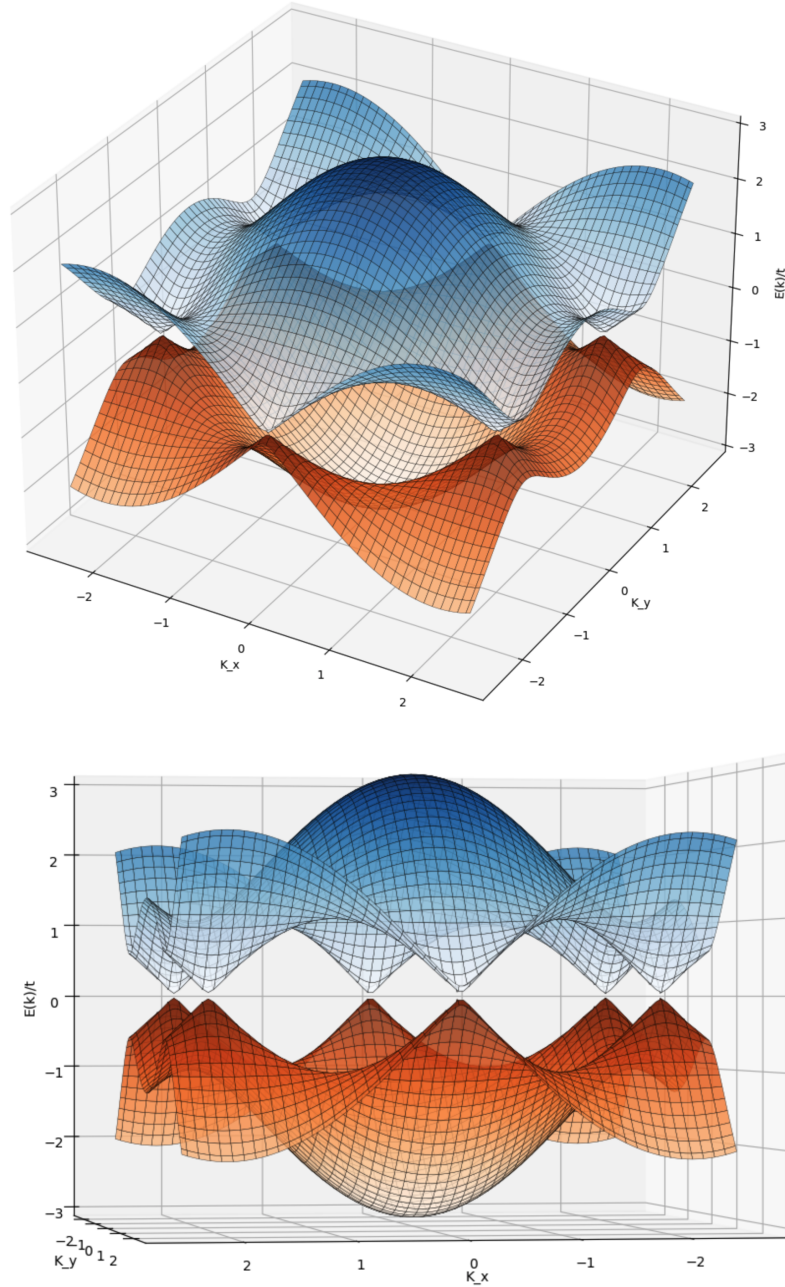


Figura 4.2: Bandas de energía del grafeno con las interacciones de los vecinos más cercanos.
Fuente: elaboración propia

Si en la Ecuación 4.15 utilizamos los valores de K y K' definidos en la Ecuación 4.5, obtenemos que la energía en esos puntos es nula. Estos puntos se denominan *puntos de Dirac*.

4.2.2. Hamiltoniano en términos de las matrices de Pauli

De la Ecuación 4.13 podemos desarrollar $\Delta_{\mathbf{k}}$:

$$\Delta_{\mathbf{k}} = \sum_{\delta} e^{i\mathbf{k} \cdot \delta} = \sum_{\delta} [\cos(\mathbf{k} \cdot \delta) + i \sin(\mathbf{k} \cdot \delta)] \quad (4.17)$$

de este modo el hamiltoniano $\mathbf{h}(\mathbf{k})$ puede escribirse como:

$$\mathbf{h}(\mathbf{k}) = -t \sum_{\delta} \begin{pmatrix} 0 & \cos(\mathbf{k} \cdot \delta) + i \sin(\mathbf{k} \cdot \delta) \\ \cos(\mathbf{k} \cdot \delta) - i \sin(\mathbf{k} \cdot \delta) & 0 \end{pmatrix} \quad (4.18)$$

En términos de las matrices de Pauli, el hamiltoniano es:

$$\mathbf{h}(\mathbf{k}) = -t \sum_{\delta} [\cos(\mathbf{k} \cdot \delta) \sigma_x - \sin(\mathbf{k} \cdot \delta) \sigma_y] \quad (4.19)$$

4.3. Aproximación primeros y segundos vecinos

Una vez analizado el efecto de incorporar únicamente los primeros vecinos (Sección 4.2), vamos a incorporar el efecto de los siguientes vecinos más cercanos.

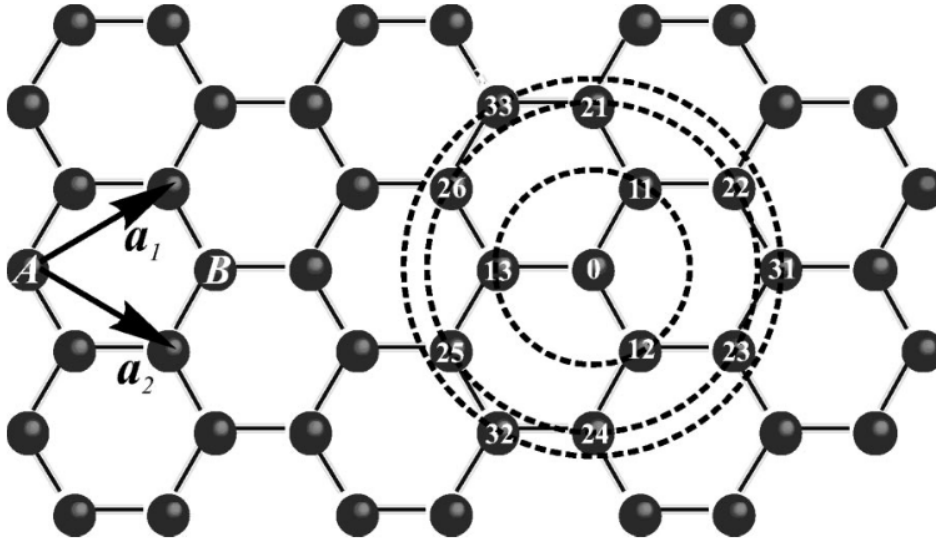


Figura 4.3: Disposición de los átomos del grafeno. Cada celda unitaria contiene dos átomos de carbono, A y B, pertenecientes a las dos subredes. Un átomo de A tiene tres vecinos más cercanos de B, seis vecinos inmediatamente posteriores pertenecientes a A, y tres segundos vecinos más cercanos que pertenecen a B. Fuente: Reich (2002)

Si consideramos la interacción tanto de los tres vecinos más cercanos como de los

siguientes vecinos más próximos (véase Figura 4.3), el hamiltoniano para el grafeno es:

$$\begin{aligned}\hat{H} = & -t \sum_{\langle ij \rangle} (\hat{a}_i^\dagger \hat{b}_j + \hat{b}_j^\dagger \hat{a}_i) \\ & - t' \sum_{\langle\langle ij \rangle\rangle} (\hat{a}_i^\dagger \hat{a}_j + \hat{b}_i^\dagger \hat{b}_j + \hat{a}_j^\dagger \hat{a}_i + \hat{b}_j^\dagger \hat{b}_i)\end{aligned}\quad (4.20)$$

donde $i(j)$ se refiere a la subred A(B), el operador fermiónico $\hat{a}_i^\dagger \hat{a}_i$ crea (aniquila) un electrón en la subred A cuya posición es $\mathbf{r}_i$, y de forma similar para $\hat{b}_j^\dagger \hat{b}_j$. Por otra parte, t es la energía del salto al vecino más cercano (salto entre las dos subredes), que tiene un valor aproximado de 2.8 eV, mientras que t' es la energía del salto a los siguientes vecinos más cercanos (salto en la misma subred), que tiene un valor aproximado calculado empíricamente de 0.1 eV¹.

Operando de manera análoga a como se hizo en la Sección 4.2, Podemos reescribir la suma sobre los segundos vecinos más cercanos como:

$$\begin{aligned}\hat{H} = & -t \sum_{\delta, \mathbf{k}} [e^{-i\mathbf{k} \cdot \delta} \hat{a}_{\mathbf{k}}^\dagger \hat{b}_{\mathbf{k}} + e^{i\mathbf{k} \cdot \delta} \hat{b}_{\mathbf{k}}^\dagger \hat{a}_{\mathbf{k}}] \\ & - Ct \sum_{\delta', \mathbf{k}} [e^{-i\mathbf{k} \cdot \delta'} \hat{a}_{\mathbf{k}}^\dagger \hat{a}_{\mathbf{k}} + e^{i\mathbf{k} \cdot \delta'} \hat{a}_{\mathbf{k}}^\dagger \hat{a}_{\mathbf{k}} + e^{-i\mathbf{k} \cdot \delta'} \hat{b}_{\mathbf{k}}^\dagger \hat{b}_{\mathbf{k}} + e^{i\mathbf{k} \cdot \delta'} \hat{b}_{\mathbf{k}}^\dagger \hat{b}_{\mathbf{k}}]\end{aligned}\quad (4.21)$$

donde se ha aplicado que:

$$t \approx 2,8eV, \quad t' \approx 0,1eV \Rightarrow t' = Ct, \text{ con } C \approx \frac{1}{28} \quad (4.22)$$

Igualmente se puede expresar el hamiltoniano de una forma más compacta:

$$\hat{H} = \sum_{\mathbf{k}} \Psi^\dagger \mathbf{h}(\mathbf{k}) \Psi \quad (4.23)$$

donde

$$\Psi \equiv \begin{pmatrix} \hat{a}_{\mathbf{k}} \\ \hat{b}_{\mathbf{k}} \end{pmatrix}, \quad \Psi^\dagger \equiv \begin{pmatrix} \hat{a}_{\mathbf{k}}^\dagger & \hat{b}_{\mathbf{k}}^\dagger \end{pmatrix} \quad (4.24)$$

y la representación matricial del hamiltoniano es

$$\mathbf{h}(\mathbf{k}) \equiv -t \begin{pmatrix} C \sum_{\delta'} (e^{i\mathbf{k} \cdot \delta'} + e^{-i\mathbf{k} \cdot \delta'}) & \Delta_{\mathbf{k}} \\ \Delta_{\mathbf{k}}^* & C \sum_{\delta'} (e^{i\mathbf{k} \cdot \delta'} + e^{-i\mathbf{k} \cdot \delta'}) \end{pmatrix}, \quad \Delta_{\mathbf{k}} \equiv \sum_{\delta} e^{i\mathbf{k} \cdot \delta} \quad (4.25)$$

¹El valor de t' no es bien conocido, pero según los cálculos de Reich (2002), su valor es de $0,02t \lesssim t' \lesssim 0,2t$. Deacon (2007) encontraron el valor $t' \approx 0,1eV$.

4.3.1. Bandas de energía

Las bandas de energía se corresponderán con los autovalores de la matriz del hamiltoniano según la Ecuación 4.25 (Bloch, 1929).

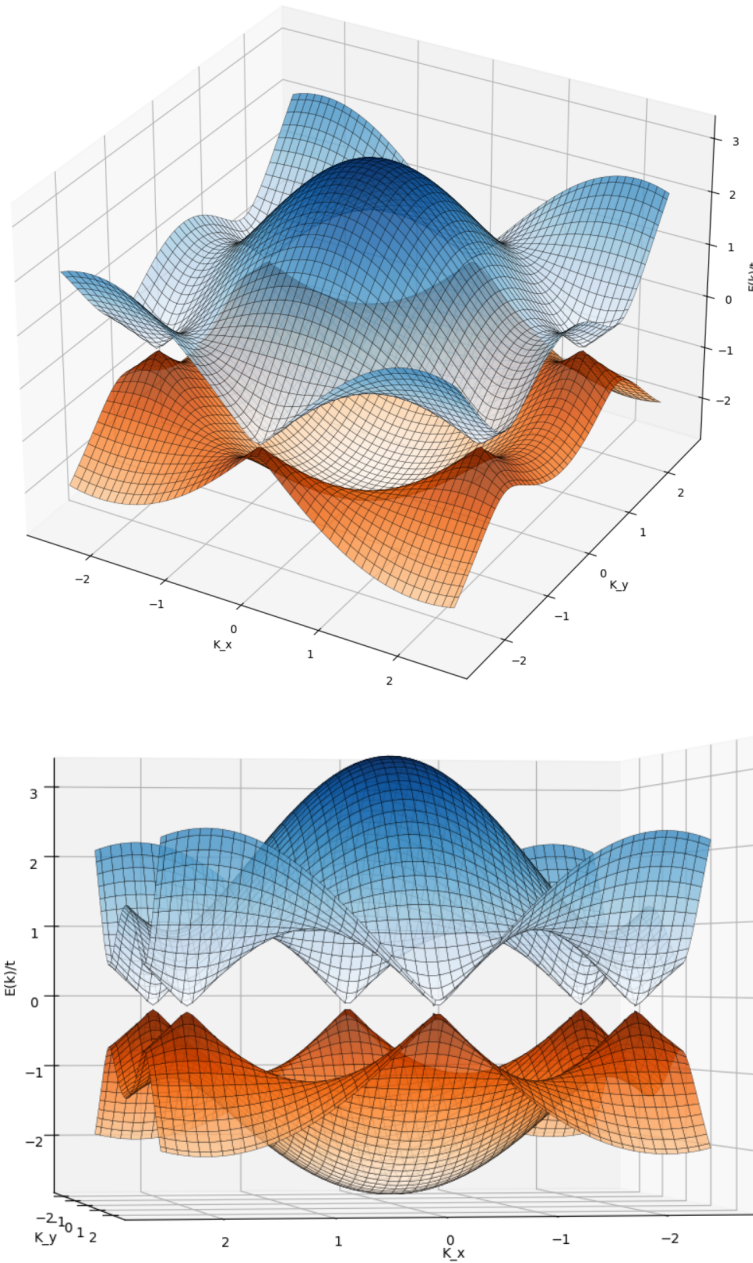


Figura 4.4: Bandas de energía del grafeno con las interacciones de los dos niveles de vecinos más cercanos. Fuente: elaboración propia

Estos autovalores serán $E_{\pm} = C \sum_{\delta'} (e^{i\mathbf{k} \cdot \delta'} + e^{-i\mathbf{k} \cdot \delta'}) \pm \sqrt{\Delta_{\mathbf{k}}^* \Delta_{\mathbf{k}}} = 2C \sum_{\delta'} \cos(\mathbf{k} \cdot \delta') \pm \sqrt{\Delta_{\mathbf{k}}^* \Delta_{\mathbf{k}}}$. Tomando el desarrollo de $\Delta_{\mathbf{k}}$ de la Sección 4.2.1 (Ecuación 4.14), las bandas de

energía se pueden expresar como:

$$E_{\pm}(\mathbf{k}) = t2C \left(2 \cos\left(\frac{3}{2}k_x a\right) \cos\left(\frac{\sqrt{3}}{2}k_y a\right) + \cos(\sqrt{3}k_y a) \right) \pm t \sqrt{1 + 4 \cos\left(\frac{3}{2}k_x a\right) \cos\left(\frac{\sqrt{3}}{2}k_y a\right) + 4 \cos^2\left(\frac{\sqrt{3}}{2}k_y a\right)}$$

$$= tCf(\mathbf{k}) \pm t\sqrt{3 + f(\mathbf{k})}, \quad (4.26)$$

$$f(\mathbf{k}) = 4 \cos\left(\frac{3}{2}k_x a\right) \cos\left(\frac{\sqrt{3}}{2}k_y a\right) + 2 \cos(\sqrt{3}k_y a) \quad (4.27)$$

La Figura 4.4 presenta dos imágenes de las gráficas correspondientes a las bandas de energía E_+ (azul) y E_- (naranja).

Si en la Ecuación 4.26 utilizamos los valores de K y K' definidos en la Ecuación 4.5, obtenemos que la energía en esos puntos (*puntos de Dirac*) ahora ya no es nula, sino que tiene un valor $-3Ct = -3t'$. Por otra parte, comparando las Figuras 4.2 y 4.4, se aprecia que la banda de valencia se ha reducido mientras que la banda de conducción es algo mayor (valores de mínimo a máximo).

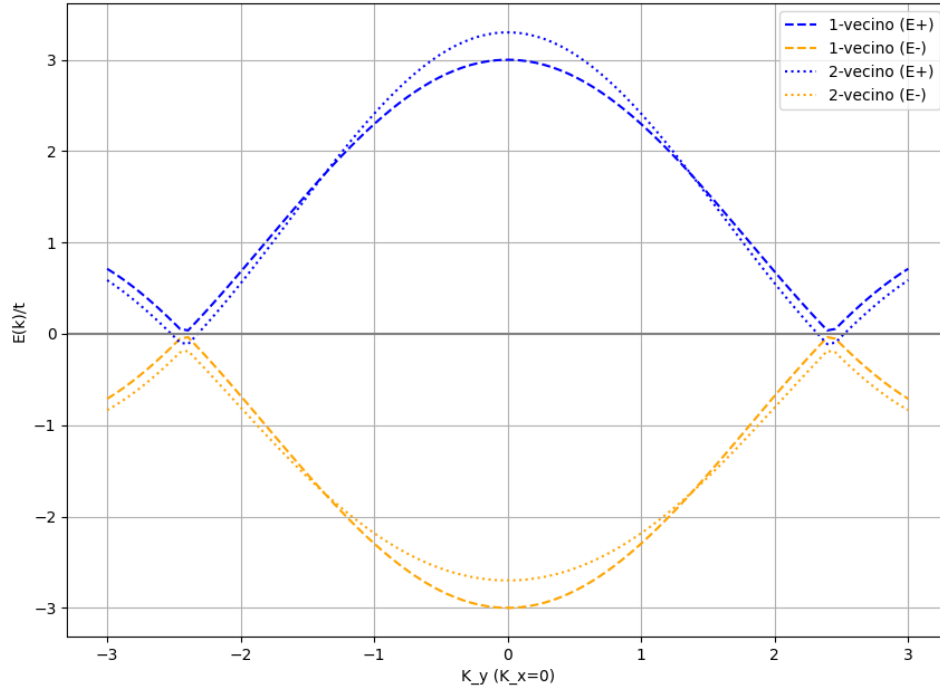


Figura 4.5: Comparativa de las bandas de energía para un corte en $k_x = 0$. Fuente: elaboración propia

La Figura 4.5 muestra una comparativa entre un corte en $k_x = 0$ de las gráficas correspondientes a los vecinos más próximos (Figura 4.2) y de las gráficas correspondientes a los primeros y segundos vecinos más próximos (Figura 4.4). En ella se puede apreciar, tanto el cambio en el valor de los puntos de Dirac, como en las energías de la banda de conducción y la banda de valencia.

4.3.2. Hamiltoniano en términos de las matrices de Pauli

De la Ecuación 4.25 podemos desarrollar $\Delta_{\mathbf{k}}$:

$$\Delta_{\mathbf{k}} = \sum_{\boldsymbol{\delta}} e^{i\mathbf{k} \cdot \boldsymbol{\delta}} = \sum_{\boldsymbol{\delta}} [\cos(\mathbf{k} \cdot \boldsymbol{\delta}) + i \sin(\mathbf{k} \cdot \boldsymbol{\delta})] \quad (4.28)$$

de este modo el hamiltoniano $\mathbf{h}(\mathbf{k})$ puede escribirse como:

$$\begin{aligned} \mathbf{h}(\mathbf{k}) = -t \sum_{\boldsymbol{\delta}} \begin{pmatrix} 0 & \cos(\mathbf{k} \cdot \boldsymbol{\delta}) + i \sin(\mathbf{k} \cdot \boldsymbol{\delta}) \\ \cos(\mathbf{k} \cdot \boldsymbol{\delta}) - i \sin(\mathbf{k} \cdot \boldsymbol{\delta}) & 0 \end{pmatrix} \\ - 2tC \sum_{\boldsymbol{\delta}'} \begin{pmatrix} \cos(\mathbf{k} \cdot \boldsymbol{\delta}') & 0 \\ 0 & \cos(\mathbf{k} \cdot \boldsymbol{\delta}') \end{pmatrix} \end{aligned} \quad (4.29)$$

En términos de las matrices de Pauli, el hamiltoniano es:

$$\mathbf{h}(\mathbf{k}) = -t \sum_{\boldsymbol{\delta}} [\cos(\mathbf{k} \cdot \boldsymbol{\delta}) \sigma_x - \sin(\mathbf{k} \cdot \boldsymbol{\delta}) \sigma_y] - 2tC \sum_{\boldsymbol{\delta}'} \cos(\mathbf{k} \cdot \boldsymbol{\delta}') I \quad (4.30)$$

5. Resultados y análisis

Para la obtención de los resultados derivados de la implementación de la metodología descrita en el Capítulo 4, se ha desarrollado un programa específico, en lenguaje Python, capaz de ejecutarse tanto en un simulación como en hardware cuántico real, si se tiene acceso al mismo. Este software incorpora un conjunto de parámetros que permiten su ejecución con diferentes configuraciones.

La estructura del presente Capítulo presenta, en primer término, una descripción completa del software implementado, detallando sus diferentes opciones y los parámetros disponibles para el usuario. Posteriormente, se procede a la presentación comparativa de los resultados obtenidos mediante la aplicación del algoritmo a un mismo conjunto de puntos de referencia, evaluado tanto teniendo en cuenta únicamente los primeros vecinos, como incluyendo la influencia de los segundos vecinos. En cada uno de estos casos, se lleva a cabo un análisis crítico de los datos obtenidos mediante su comparación con los datos obtenidos del cálculo de los autovalores del correspondiente hamiltoniano.

5.1. Software utilizado

En el Anexo A.1 se incluye el código fuente del programa escrito en Python (Python, 2024) que realiza los cálculos tanto en un simulador como, en el caso de disponer de acceso, en un procesador cuántico real de la plataforma IBM Quantum (IBM, 2023).

Los cálculos se van a realizar para el conjunto de puntos de la red recíproca Γ , K , M y Γ , además de n puntos adicionales, definidos por el usuario, entre cada dos de esos puntos de la red recíproca.

Para ello se ha definido una clase, denominada `BandasEnergiaGrafeno` que incluye diferentes métodos y miembros, que van a dar flexibilidad y variedad de opciones en el momento de realizar los cálculos. Su definición completa es:

```
BandasEnergiaGrafeno(t=1.0, t_prima=1.0/28, a=1.42, vecinos=2),  
donde
```

- **t** es el valor de la energía del salto al vecino más cercano. Su valor por defecto es 1.0 para establecer una escala de referencia en las energías que se obtienen (Sección 4.2).
- **t_prima** es el valor de la energía del salto a los siguientes vecinos más cercanos. Su valor por defecto es 1.0/28 (Sección 4.3 y su pie de página).

- **a** es la distancia entre los dos átomos de carbono más próximos en la estructura del grafeno (Sección 4.1).

Una vez creada una instancia de esta clase, es posible aplicar uno de los dos principales métodos definidos: `calcular_bandas_energia` y `graficos`. El primero de ellos realiza el cálculo de la estructura de bandas y el segundo genera un gráfico con los resultados obtenidos.

La definición completa del método `calcular_bandas_energia` es la siguiente:

```
calcular_bandas_energia(num_points=4, optimizador='COBYLA',  
ansatz_str='EfficientSU2', ruido=None,  
fichero_entrada=None, fichero_salida=None, token=None),
```

donde

- **num_points** es el número de puntos a calcular entre los puntos de la red recíproca Γ , K , M y Γ . Por defecto su valor es 4 (Sección 4.1).
- **optimizador** es el optimizador a utilizar en el algoritmo VQE. Los valores posibles son 'COBYLA' y 'SPSA'. Por defecto, el valor es 'COBYLA' (Sección 2.3.2).
- **ansatz_str** es el ansatz a utilizar como circuito parametrizado. Sólo es posible el valor por defecto 'EfficientSU2' (Sección 2.3.2).
- **ruido** permite la inclusión en el simulado de ruido adicional. Es posible aplicar o no ruido modelo 'FakeTorino' de la librería de IBM qiskit. El valor por defecto es 'None' (no se aplica ningún modelo de ruido adicional).
- **fichero_entrada** permite continuar un cálculo comenzado anteriormente y no concluido que ha sido almacenado en un fichero externo mediante la opción que se describe a continuación. El valor por defecto es None (no se define ningún fichero de entrada).
- **fichero_salida** permite almacenar los resultados de todas las iteraciones que se realicen en un fichero externo. El valor por defecto es 'None' (no se define ningún fichero de salida).
- **token** es el token que habilita a un usuario para utilizar el hardware cuántico real de IBM en lugar de una simulación. Si se proporciona este valor, los cálculos se van a realizar en el QPU menos ocupado disponible en ese momento en la plataforma de

IBM. El valor por defecto es 'None' (no se define ningún token y por tanto el cálculo se realiza en el simulador de IBM Qiskit).

La definición completa del método `calcular_bandas_energia` es la siguiente:

```
graficos(fichero_entrada=None,simulacion=True,ingles=False),
```

donde

- **fichero_entrada** permite cargar los datos desde un fichero externo para presentar un gráfico con esos resultados. El valor por defecto es 'None' (no se define ningún fichero de entrada).
- **simulacion** indica si los datos se han calculado con el simulador de IBM Qiskit ('True') o si se han realizado mediante hardware real ('False'). El valor por defecto es 'False'.
- **ingles** indica que los textos del gráfico se presentan en inglés ('True') o en español ('False'). Por defecto el valor es 'False'.

La teoría desarrollada en el Capítulo 4 está reflejada principalmente en los siguientes bloques de software.

- El método `calcular_delta` calcula los vectores de los tres vecinos más cercanos en el espacio real δ_1 , δ_2 y δ_3 .
- El método `calcular_delta2` calcula los vectores de los siguientes seis vecinos más cercanos δ'_1 , δ'_2 y δ'_3 (cada vector tiene dos valores, uno positivo y otro negativo).
- El método `cost_func` define la función de coste a minimizar, es decir calcula el valor esperado del hamiltoniano para un valor de prueba dado por el circuito parametrizado (ansatz) (Sección 2.3.2). También se aplica al hamiltoniano modificado H' para la obtención del estado excitado (Sección 2.3.3).

El método `calcular_bandas_energia` incluye todos los pasos necesarios para el cálculo de la estructura de bandas a partir de los parámetros iniciales indicados por el usuario. En este sentido cabe destacar:

- Preparación de los puntos de la estructura recíproca y de los puntos intermedios indicados por el usuario.

- Preparación del hamiltoniano correspondiente a los primeros vecinos o a los primeros+segundos vecinos, según se haya definido.
- Preparación del simulador a utilizar o acceso al hardware real, en su caso.
- Selección del ansatz y del optimizador clásico.
- Bucle para las iteraciones correspondientes hasta que el número de iteraciones alcance un máximo o el resultado converja.
- Almacenamiento en memoria (y en fichero externo si así se ha indicado) de los resultados de cada una de las iteraciones.

Mediante el software descrito se han conseguido los resultados que se presentan en las secciones siguientes.

5.2. Resultados incluyendo los primeros vecinos

Mediante el uso del software descrito en la Sección 5.1 se obtienen resultados para las diferentes configuraciones.

En las gráficas que se van a mostrar en esta Sección y en la siguiente, se presentan líneas y puntos. Las líneas unen los puntos correspondientes a los autovalores calculados a partir de la matriz del hamiltoniano correspondiente. Por tanto se trata de los valores reales que queremos alcanzar con el algoritmo cuántico VQE que estamos analizando. Los puntos corresponden a los resultados de aplicar este algoritmo para los puntos de la red recíproca y los puntos intermedios indicados por el usuario (en todos los casos se tomará el valor por defecto, que corresponde a cuatro puntos entre cada uno de los cuatro puntos de la red recíproca). Los valores en color naranja se refieren al estado fundamental (banda de valencia), mientras que los valores en color azul se corresponden con el primer estado excitado (banda de conducción).

Los primeros resultados obtenidos se corresponden con cálculo de la estructura de bandas del grafeno cuando incluimos únicamente el efecto de los primeros tres vecinos, utilizando la función “Estimator” de la librería Qiskit de IBM, sin aplicar ruido adicional alguno y para los dos optimizadores clásicos empleados: COBYLA (Powell, 1994) y SPSA (Spall, 1992). Esta función actúa como un simulador ideal de circuitos cuánticos en un ordenador clásico.

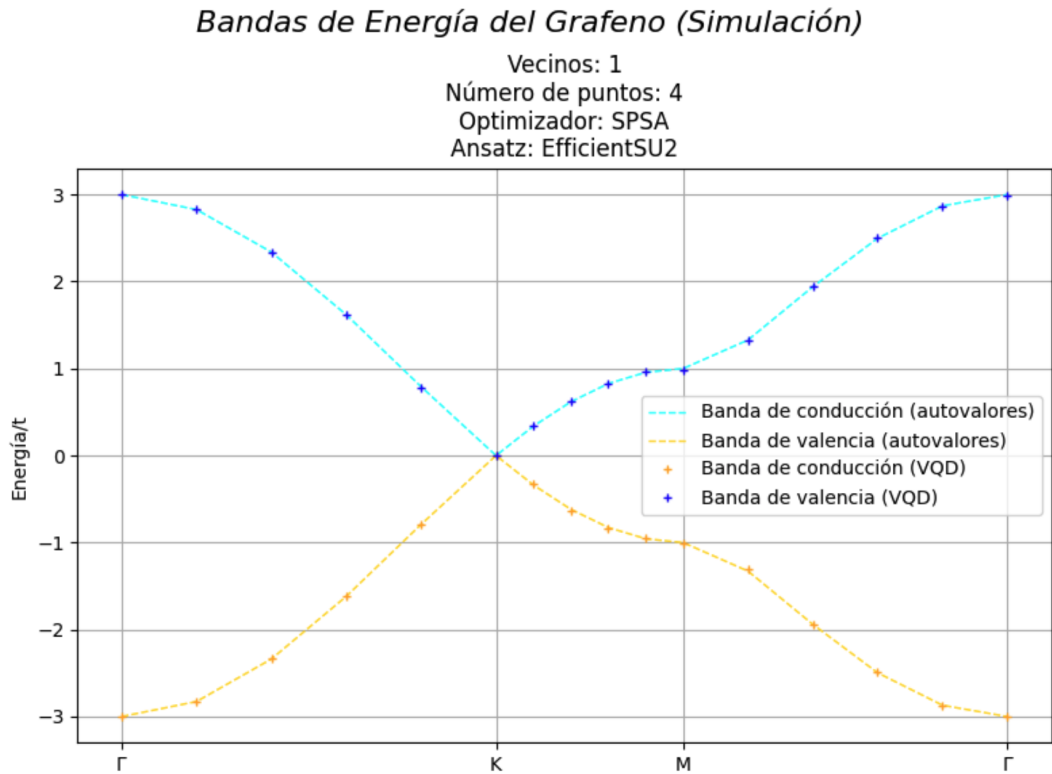
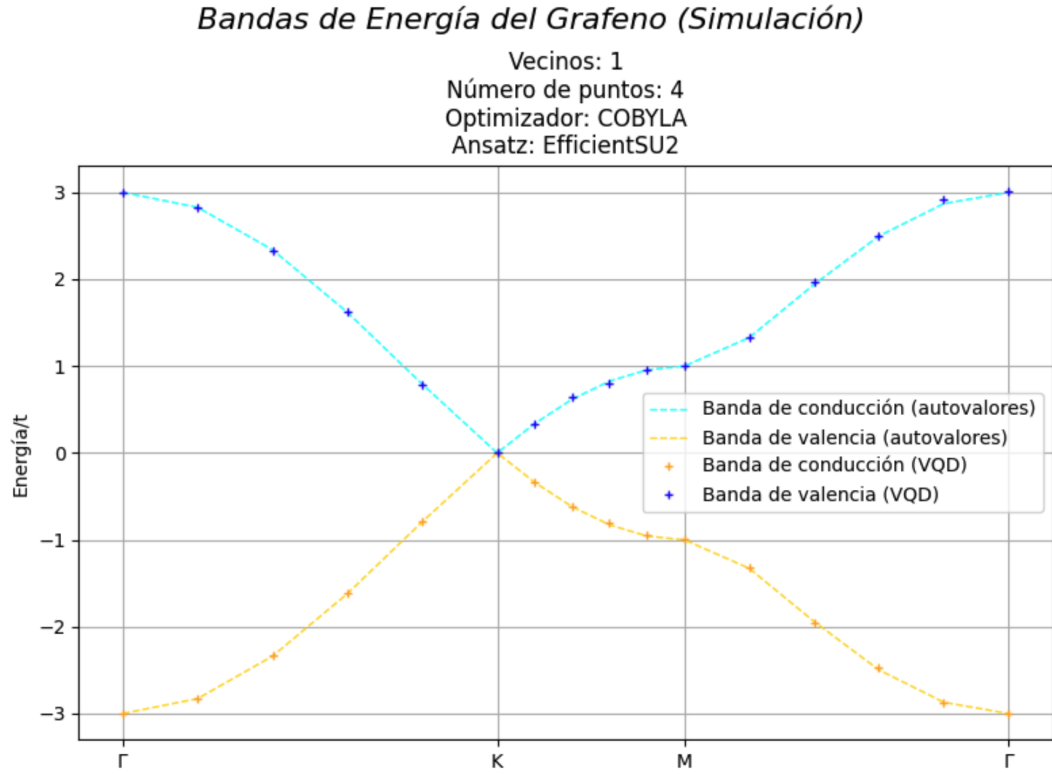
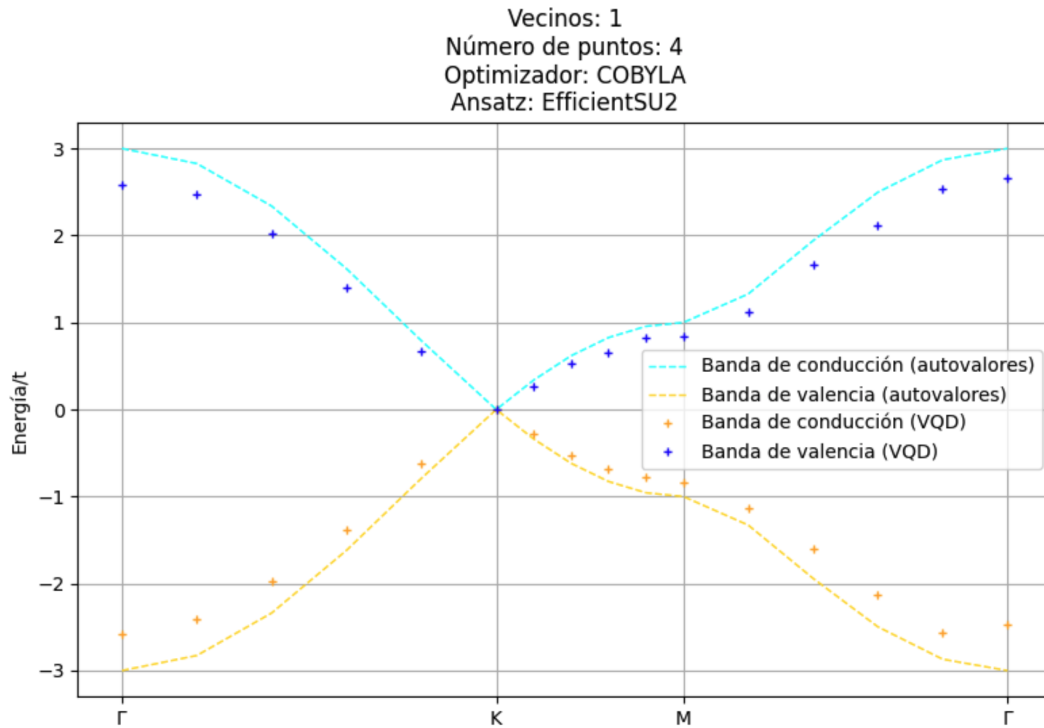


Figura 5.1: Resultados tomando sólo los primeros vecinos y sin aplicar ruido adicional *Arriba*: utilizando el optimizador COBYLA *Abajo*: utilizando el optimizador SPSA. Fuente: elaboración propia

Bandas de Energía del Grafeno (Simulación)



Bandas de Energía del Grafeno (Simulación)

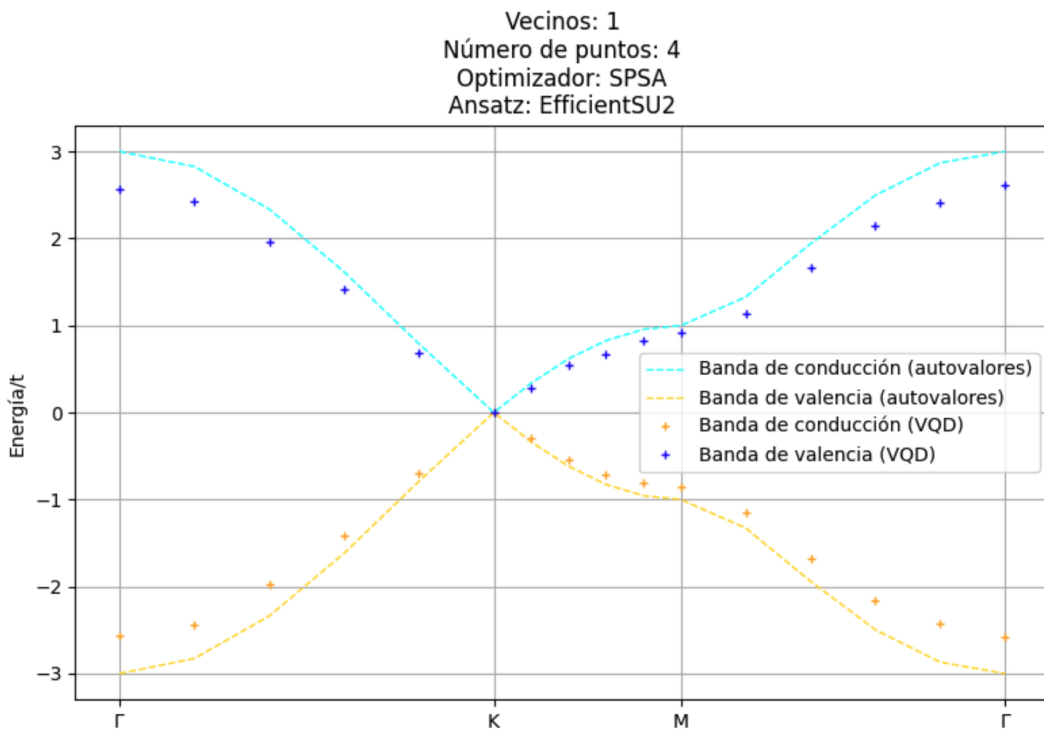


Figura 5.2: Resultados tomando sólo los primeros vecinos y aplicando ruido adicional *Arriba*: utilizando el optimizador COBYLA. *Abajo*: utilizando el optimizador SPSA. Fuente: elaboración propia

Como se puede observar en la Figura 5.1, los resultados son bastante precisos, como se esperaba al emplear un simulador de QPU ideal.

La siguiente etapa consiste en incluir ruido para tratar de simular un procesador real. Este segundo conjunto de cálculos VQE, además de utilizar la función “Estimator”, se incluye el modelo de ruido de un dispositivo real, ‘FakeTorino’, modelo de ruido adoptado que simula el hardware real cuántico “ibm_torino”, un procesador cuántico Heron de 133 cúbits desarrollado por IBM. Al incorporar el modelo de ruido de un dispositivo real, es posible probar la fiabilidad de los algoritmos empleados en presencia de ruido, mientras se ejecutan en un dispositivo clásico. Los resultados obtenidos pueden verse en la Figura 5.2.

En general, podemos apreciar que la banda de valencia y la banda de conducción, son simétricas respecto al eje X que representa energía cero. Mientras que en simulador sin ruido obtenemos, como se ha mencionado, valores muy similares a los reales, cuando aplicamos ruido, estos valores presentan una diferencia apreciable respecto a los valores reales. El valor absoluto del error es proporcional al valor real, lo que implica un error relativo prácticamente igual en todos los puntos calculados.

5.3. Resultados incluyendo los primeros y segundos vecinos

Tal como se indica en la Sección 4.3, es posible incluir, además de la influencia de los tres primeros vecinos, la influencia de los seis vecinos siguientes más próximos. En este caso el hamiltoniano correspondiente al caso que incluye únicamente los primeros vecinos, se ve modificado por una cantidad (dependiente del punto de la red recíproca a calcular) en la diagonal principal (Ecuación 4.30).

Los resultados obtenidos utilizando la función “Estimator” para esta nueva configuración, se presentan en la Figura 5.3.

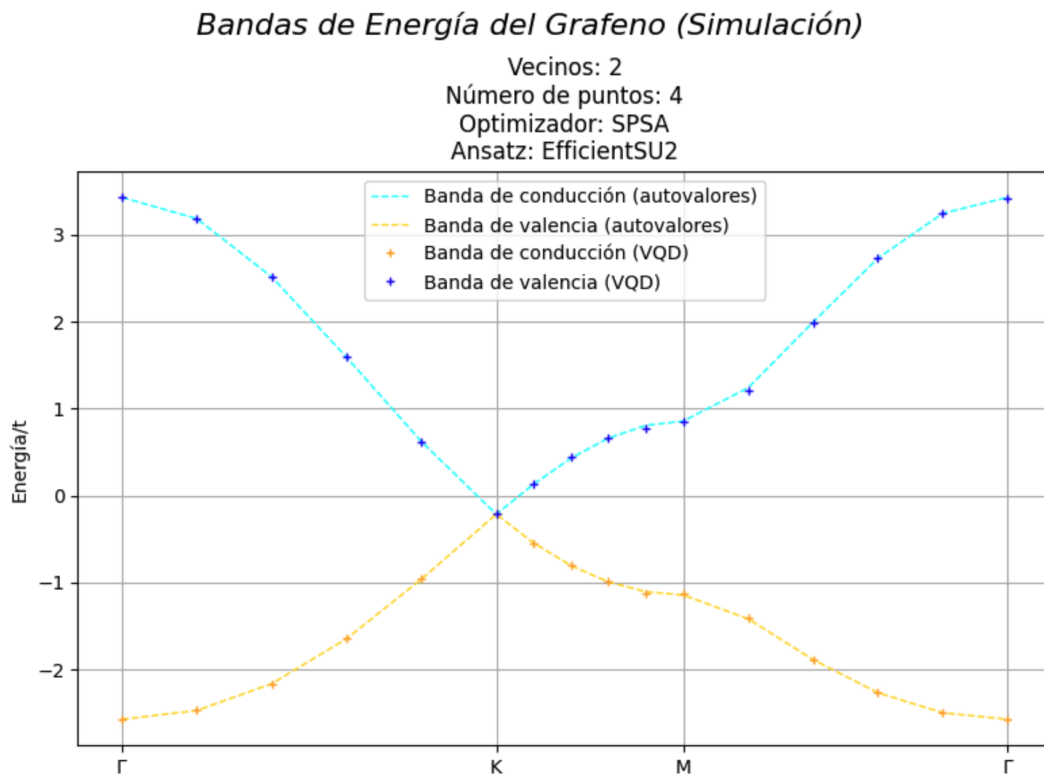
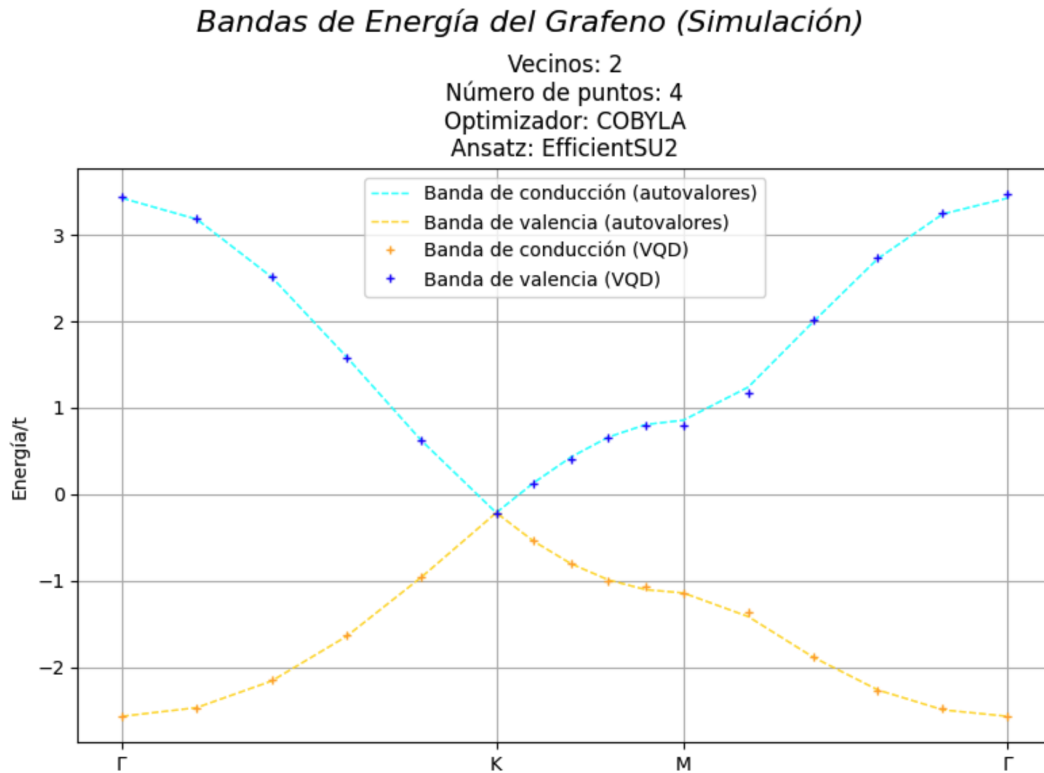
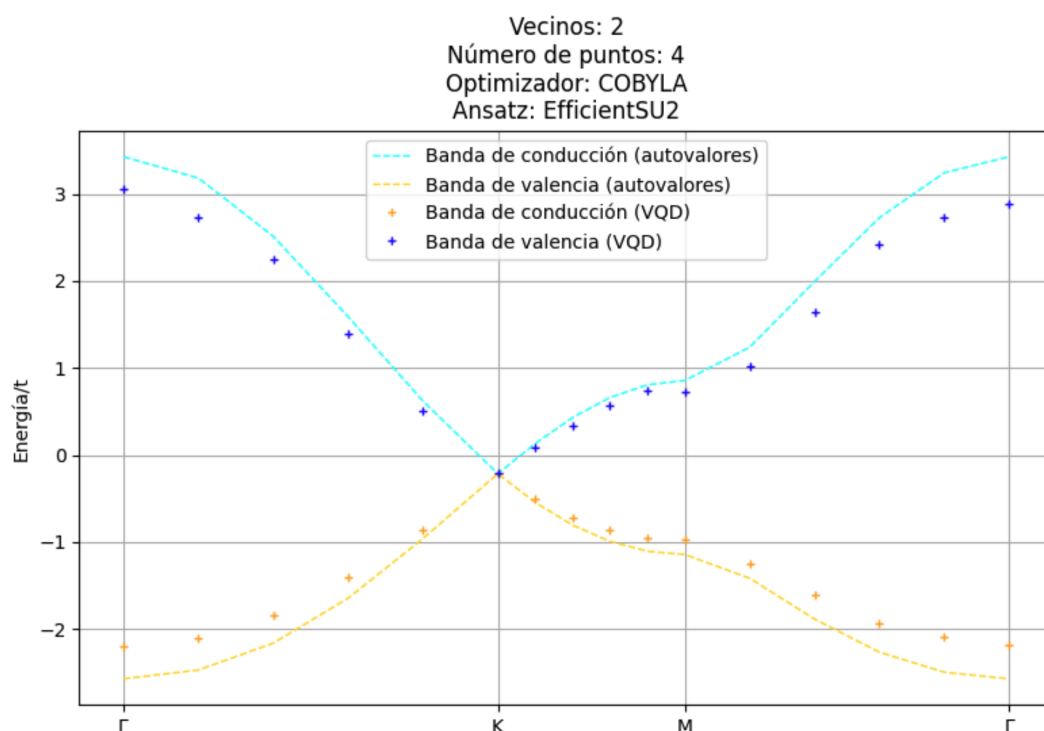


Figura 5.3: Resultados tomando los primeros+segundos vecinos, sin aplicar ruido adicional. *Arriba:* utilizando el optimizador COBYLA. *Abajo:* utilizando el optimizador SPSA. Fuente: elaboración propia

Bandas de Energía del Grafeno (Simulación)



Bandas de Energía del Grafeno (Simulación)

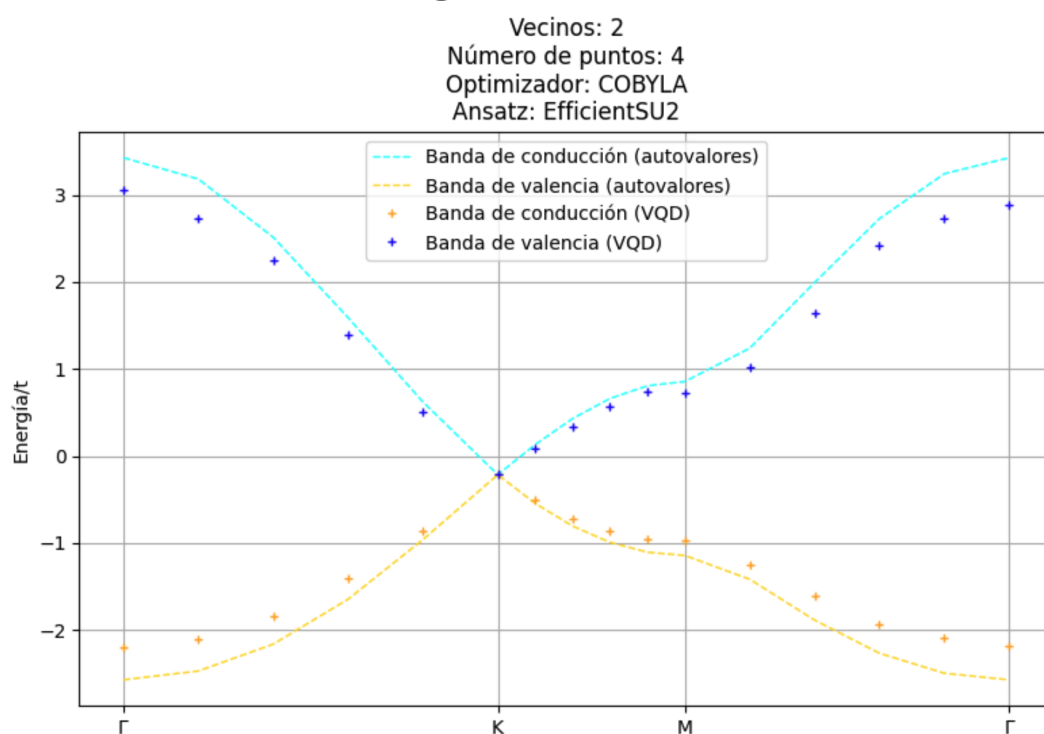


Figura 5.4: Resultados tomando los primeros+segundos vecinos, aplicando ruido adicional. *Arriba:* utilizando el optimizador COBYLA. *Abajo:* utilizando el optimizador SPSA. Fuente: elaboración propia

De nuevo, podemos aplicar ruido simulados para obtener resultados similares a los que se podrían obtener con hardware real (Figura 5.4).

Mientras que en el caso del cálculo con únicamente los primeros vecinos los resultados aparecen simétricos respecto al eje X, el efecto de los segundos vecinos “rompe” esta simetría de modo que el punto K (punto de Dirac) ya no tiene valor cero, sino que adquiere un valor negativo. El resto de puntos, “sufrir” un desplazamiento vertical hacia arriba, de modo que cada valor es algo mayor que en el caso de utilizar sólo los primeros vecinos.

5.4. Comparativa de los optimizadores COBYLA y SPSA

Con el objetivo de analizar la convergencia de los dos optimizadores utilizados (COBYLA y SPSA) vamos a presentar gráficos con los resultados obtenidos en el cálculo del punto Γ para la simulación con sólo los primeros vecinos, sin y con ruido añadido.

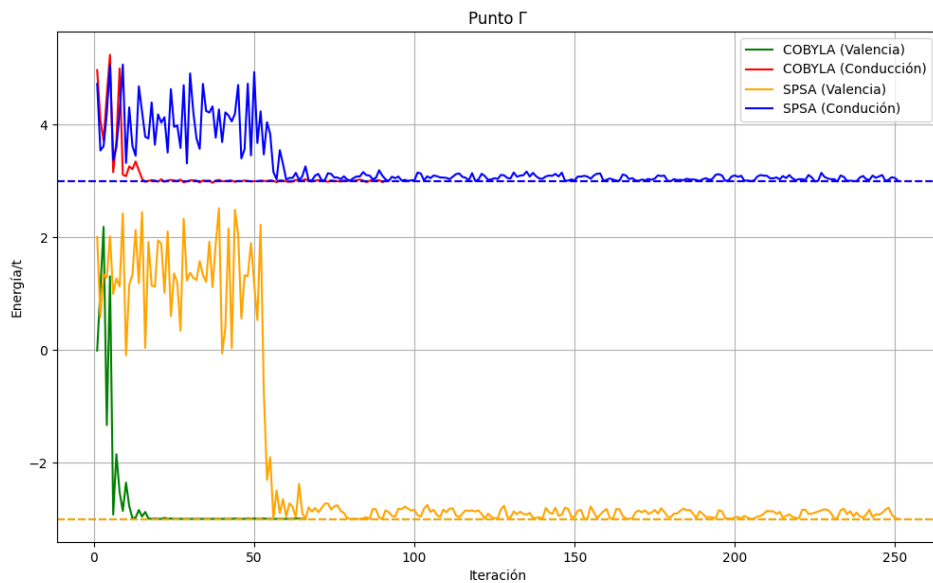


Figura 5.5: Convergencia en el punto Γ para los optimizadores COBYLA y SPSA (sin ruido). Fuente: elaboración propia.

En la Figura 5.5 se representan los resultados obtenidos en la simulación sin ruido, es decir, en un entorno ideal. Ambos optimizadores, COBYLA y SPSA, logran que la energía converja exactamente a los valores teóricos, representados por las líneas discontinuas. Esto demuestra que, en teoría, el algoritmo (VQE) y los optimizadores son capaces de resolver

el problema correctamente. El algoritmo COBYLA converge en muy pocas iteraciones y una vez que encuentra el valor mínimo, presenta oscilaciones muy pequeñas alrededor del valor real. Por otra parte, SPSA tarda más en alcanzar el valor correcto y, una vez alcanzado, muestra oscilaciones importantes alrededor de ese valor correcto. La explicación a las oscilaciones alrededor del valor real vendría de la propia naturaleza de ambos algoritmos. Mientras que COBYLA es determinista, SPSA es aleatorio (estocástico) lo que hace que siga “explorando” en el entorno de la solución real.

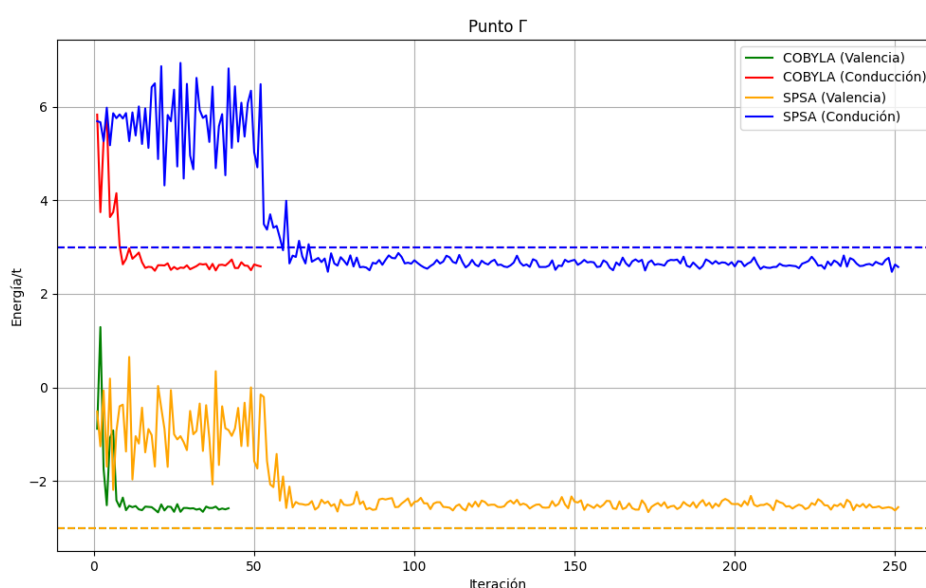


Figura 5.6: Convergencia en el punto Γ para los optimizadores COBYLA y SPSA (con ruido). Fuente: elaboración propia.

En la Figura 5.6 se representan los valores obtenidos en el caso de utilizar un simulador que incluye ruido del tipo ‘FakeTorino’. El primer dato apreciable es que ninguno de los dos algoritmos logra alcanzar el valor teórico. En ambos se produce el mismo desplazamiento del valor real, lo que apunta a un problema sistemático. En cuanto a la velocidad de convergencia (número de iteraciones hasta alcanzar las cercanías del valor final), se aprecia el mismo comportamiento que en el caso sin ruido, es decir, COBYLA converge más rápidamente que SPSA. Por último, las oscilaciones alrededor del valor final, aún siendo menores las obtenidas con COBYLA que con SPSA, sin embargo, con ruido, estas oscilaciones aumentan considerablemente en COBYLA, mientras que las oscilaciones que vemos con SPSA son aproximadamente del mismo orden.

Podemos establecer las siguientes conclusiones:

- ambos algoritmos son capaces de alcanzar el valor real después de un número suficiente de iteraciones, siendo más rápido COBYLA (alcanza el objetivo con un menor número de iteraciones) que SPSA,
- el ruido sistemático afecta a ambos algoritmos por igual, provocando un sesgo en los resultados,
- la naturaleza del propio algoritmo provoca mayores o menores oscilaciones alrededor del valor final alcanzado.

5.5. Análisis de los errores obtenidos en las diferentes estimaciones

En cada uno de los cálculos realizados para los diferentes puntos de la red recíproca se han obtenido unos resultados que difieren en mayor o menor medida de los resultados reales. En esta Sección vamos a presentar algunos de estos errores.

		Sólo primeros vecinos				Primeros+segundos vecinos			
		Simulación sin ruido		Simulación con ruido		Simulación sin ruido		Simulación con ruido	
		COBYLA	SPSA	COBYLA	SPSA	COBYLA	SPSA	COBYLA	SPSA
Γ	Valor real	-3.0	-3.0	-3.0	-3.0	-2.571429	-2.571429	-2.571429	-2.571429
	Valor estimado	-3.000000	-3.000000	-2.583984	-2.560547	-2.571429	-2.571429	-2.196429	-2.161272
	Error	0.000000	0.000000	0.416016	0.439453	0.000000	0.000000	0.375000	0.410157
	Error (%)	0.0000	0.0000	13.8672	14.6484	0.0000	0.0000	14.5833	15.9505
K	Valor real	0.0	0.0	0.0	0.0	-0.214286	-0.214286	-0.214286	-0.214286
	Valor estimado	0.000000	0.000000	0.000000	0.000000	-0.214286	-0.214286	-0.214286	-0.214286
	Error	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Error (%)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
M	Valor real	-1.0	-1.0	-1.0	-1.0	-1.142857	-1.142857	-1.142857	-1.142857
	Valor estimado	-0.997240	-1.007511	-0.843963	-0.853167	-1.146901	-1.119136	-0.976110	-0.985231
	Error	0.002760	0.007511	0.156037	0.146833	0.025956	0.033721	0.166747	0.157626
	Error (%)	0.2760	0.7511	15.6037	14.6833	2.2712	2.9501	14.5904	13.7923

Tabla 5.1: Errores en el cálculo de los puntos de la red recíproca de la bandas de conducción.

Fuente: elaboración propia.

En la Tabla 5.1 se presentan los valores reales, los valores estimados, el error absoluto y el error relativo para los principales puntos de la red recíproca (Γ , K y M) de los estados fundamentales (banda de conducción) para los diferentes escenarios combinación de las diferentes opciones: primeros vecinos, primeros + segundo vecinos, simulación con y sin ruido, y optimizadores COBYLA o SPSA.

Se pueden observar los comportamientos que se puntualizan a continuación.

En todas las simulaciones sin ruido, los errores porcentuales son muy bajos, yendo desde 0 % hasta un máximo de 3 % (en el caso más complejo). Sin embargo, en cuanto se introduce el modelo de ruido, los errores se disparan drásticamente a un rango del 13 % al 16 %, independientemente del optimizador o la complejidad del modelo (vecinos incluidos).

Si comparamos las simulaciones sin ruido, vemos que el modelo de “sólo primeros vecinos” es más fácil de resolver que el de “primeros+segundos vecinos”. Por ejemplo, para el punto M con COBYLA, el error pasa de 0.2760 % (primeros vecinos) a 2.2712 % (primeros+segundos vecinos).

Si no se incluye ruido, en los casos donde hay un error, COBYLA consistentemente obtiene un error menor que SPSA. Con ruido los resultados son mixtos. A veces COBYLA tiene un error ligeramente menor y otras veces es SPSA quien lo presenta mayor. La diferencia entre ellos es pequeña en comparación con el error total inducido por el ruido.

En las simulaciones sin ruido, el algoritmo encuentra la energía exacta (error 0.0 %) para los puntos de alta simetría Γ y K. Sin embargo, para el punto M, siempre hay un error residual, incluso en el caso ideal.

A partir de los comentarios anteriores, se pueden extraer las siguientes conclusiones:

- el ruido del hardware es la fuente de error más dominante;
- la complejidad del modelo físico (número de vecinos que se incluyen en el modelo) aumenta el error;
- en ausencia de ruido, COBYLA es ligeramente más preciso que SPSA mientras que en presencia de ruido, se comportan de manera similar;
- la precisión del algoritmo depende, de forma importante, del punto de la red recíproca específico que se quiere calcular (Γ , K, M).

En la Tabla 5.2 se presentan los resultados para el primer estado excitado (banda de valencia) en los mismos supuestos de la Tabla 5.1.

		Sólo primeros vecinos				Primeros+segundos vecinos			
		Simulación sin ruido		Simulación con ruido		Simulación sin ruido		Simulación con ruido	
		COBYLA	SPSA	COBYLA	SPSA	COBYLA	SPSA	COBYLA	SPSA
Γ	Valor real	3.0	3.0	3.0	3.0	3.428571	3.428571	3.428571	3.428571
	Valor estimado	3.001535	3.000651	2.586314	2.573284	3.446471	3.429462	3.056540	2.984394
	Error	0.001535	0.000651	0.416016	0.439453	0.017900	0.000891	0.372031	0.444177
	Error (%)	0.0512	0.0217	13.8672	14.6484	0.5221	0.02599	10.8501	12.9552
K	Valor real	0.0	0.0	0.0	0.0	-0.214286	-0.214286	-0.214286	-0.214286
	Valor estimado	0.000000	0.000000	0.000000	0.000000	-0.214286	-0.214233	-0.214286	-0.214233
	Error	0.000000	0.000000	0.000000	0.000000	0.000000	0.000053	0.000000	0.000053
	Error (%)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0247	0.0000	0.0247
M	Valor real	1.0	1.0	1.0	1.0	0.857143	0.857143	0.857143	0.857143
	Valor estimado	1.002996	0.980396	0.841205	0.911561	0.795489	0.849626	0.731100	0.749374
	Error	0.002996	0.019604	0.158795	0.088439	0.061654	0.007517	0.126043	0.107769
	Error (%)	0.2966	1.9606	15.8795	8.8439	7.1030	0.8770	14.7050	12.5730

Tabla 5.2: Errores en el cálculo de los puntos de la red recíproca de la bandas de valencia.
Fuente: elaboración propia.

Procediendo de manera análoga a como se hizo con la Tabla 5.1, podemos concretar los comportamientos que se indican en los puntos que se presentan a continuación.

Quizá la conclusión más importante y evidente es que los errores en las simulaciones sin ruido son ahora significativamente mayores que en la Tabla 5.1.

Los errores en las simulaciones con ruido siguen siendo muy altos (entre 8 % y 16 %). Sin embargo, la brecha entre la simulación con y sin ruido ahora es diferente. En algunos casos, el error “sin ruido” ya es tan grande (ej. 7.10 %) que el ruido de hardware “sólo” lo aumenta hasta el 14.70 %.

Si bien en el caso sin ruido COBYLA seguía siendo a menudo superior, en los escenarios más complejos y ruidosos, SPSA muestra un rendimiento muy competitivo e incluso superior. Por ejemplo, en el caso más difícil (“primeros+segundos vecinos”, con ruido, punto M), SPSA obtiene un error del 12.57 % frente al 14.71 % de COBYLA.

Comparando ambas tablas, es inequívoco que obtener una estimación precisa para el primer estado excitado es significativamente más difícil que para el estado fundamental, en todas las condiciones.

Las conclusiones a las que se pueden llegar son las siguientes:

- el error se acumula, ya que el VQD es inherentemente sensible a la precisión de los estados previamente calculados, y si el estado anteriormente calculado ya incluye error, éste se acumula en el cálculo del estado excitado siguiente;
- en presencia de ruido, el error total es una combinación de ruido de hardware y error de propagación;

- la robustez de SPSA en escenarios de coste complejos se hace más evidente;
- el cálculo de estados de mayor energía es fundamentalmente más difícil.

Por tanto, mientras que el VQE para el estado fundamental se enfrenta principalmente al ruido del hardware y a los límites del optimizador, el VQD para el primer estado excitado se enfrenta a todos esos mismos problemas más el error propagado del cálculo anterior.

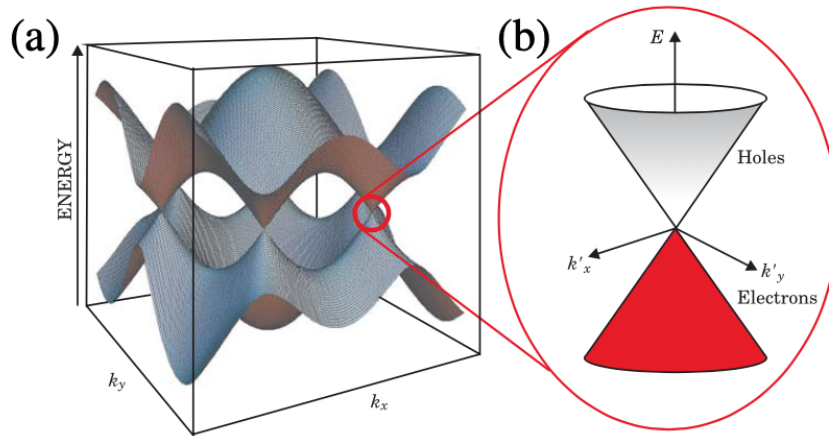


Figura 5.7: Conos de Dirac. (a) Estructura de bandas del grafeno. (b) Ampliación de la estructura de bandas cerca de los puntos K y K' que muestra los conos de Dirac. Fuente: Sarma (2011)

Por otra parte, vemos que el punto K (punto de Dirac) presenta unas características especiales ya que es el punto donde menor error (incluso nulo) se obtiene en los diferentes escenarios. Esto es debido a que se trata de los puntos en los que se encuentran los conos de Dirac (Figura 5.7). En estos puntos la función de energía presenta un máximo (mínimo) muy pronunciado, lo que favorece que los optimizadores lleguen más rápidamente y con mayor precisión a este punto.

6. Conclusiones

Una vez finalizado el desarrollo experimental y el análisis de los resultados, se presentan las principales conclusiones de este Trabajo Fin de Máster, que ha tenido como objetivo central el cálculo de la estructura de bandas del grafeno mediante el uso del algoritmo VQE.

Las conclusiones se pueden resumir en los siguientes puntos clave:

1. El algoritmo VQE es una herramienta viable y teóricamente precisa para el cálculo de la estructura de bandas. La conclusión fundamental de este estudio es que el algoritmo VQE, complementado con el método VQD para los estados excitados, es capaz de resolver el problema del cálculo de los autovalores del hamiltoniano del grafeno. En las simulaciones realizadas en un entorno ideal (sin ruido), los resultados obtenidos mostraron una concordancia excelente con las soluciones analíticas derivadas del modelo del enlace fuerte. Esto valida que el enfoque metodológico, desde la formulación del hamiltoniano en matrices de Pauli hasta la optimización variacional, es correcto y robusto.
2. El VQE es capaz de capturar la física subyacente de modelos con diferente complejidad. Se ha demostrado que el algoritmo no solo funciona para un modelo simplificado, sino que también es sensible a las características físicas más sutiles.
 - Para el modelo de primeros vecinos, el VQE reprodujo con éxito la simetría entre la banda de valencia y la de conducción, así como la degeneración energética en los puntos de Dirac, característica distintiva del grafeno.
 - Al incluir las interacciones con los segundos vecinos, el algoritmo fue capaz de modelar correctamente la ruptura de esta simetría, manifestada en un desplazamiento vertical de las bandas y en la modificación de la energía en los puntos K. Este hallazgo es de especial importancia, ya que demuestra el potencial del VQE para abordar hamiltonianos más complejos y realistas, que son cruciales para describir materiales con mayor precisión.
3. El ruido es el principal factor limitante para la aplicación práctica en hardware cuántico actual. Las simulaciones que incorporaron un modelo de ruido realista, emulando un procesador de la era NISQ, revelaron una desviación sistemática y apreciable de

los resultados con respecto a los valores teóricos. Se observó que el error absoluto es proporcional a la magnitud de la energía, lo que implica que la precisión del algoritmo se ve comprometida significativamente por la decoherencia y los errores de las puertas cuánticas. Este resultado subraya que, aunque el VQE es teóricamente potente, su aplicación práctica exitosa en el hardware actual dependerá críticamente del desarrollo y la implementación de técnicas avanzadas de mitigación de errores.

4. La implementación del software ha sido exitosa y cumple con los objetivos propuestos. Se ha desarrollado una herramienta en Python, utilizando Qiskit, que permite de manera flexible y parametrizable calcular la estructura de bandas del grafeno. El programa es capaz de gestionar diferentes modelos de hamiltoniano, optimizadores, y condiciones de simulación (ideal vs. ruidosa), constituyendo una base sólida para futuras investigaciones.

En síntesis, este Trabajo concluye que la aplicación del VQE para el estudio de materiales como el grafeno es una vía prometedora y conceptualmente válida. Se ha logrado una prueba de concepto exitosa en simuladores, demostrando la capacidad del algoritmo para resolver un problema relevante en la física de la materia condensada. Sin embargo, también se ha puesto de manifiesto que la brecha entre la simulación ideal y la ejecución en hardware real sigue siendo considerable, consolidando el ruido como el principal obstáculo a superar para alcanzar la denominada “ventaja cuántica” en este campo.

7. Trabajo futuro

El presente Trabajo ha establecido una base sólida para la aplicación del algoritmo VQE al cálculo de la estructura de bandas del grafeno, validando su viabilidad teórica y explorando sus limitaciones en entornos ruidosos. A partir de las conclusiones obtenidas, se proponen las siguientes líneas de investigación y desarrollo futuro para expandir y mejorar los resultados aquí presentados:

1. **Implementación de técnicas de mitigación de errores.** Dado que el ruido se ha identificado como el principal obstáculo, el paso más inmediato y relevante es incorporar técnicas de mitigación de errores en el flujo de trabajo. Se podrían explorar métodos como:

- ejecutar el algoritmo con diferentes niveles de ruido (escalando artificialmente el ruido añadido al simulador) para extrapolar el resultado al límite de cero ruido.
- aplicar técnicas de corrección de errores.

La implementación de estas técnicas permitiría cuantificar cuánto se puede mejorar la precisión en el hardware NISQ actual y acercar los resultados de la simulación ruidosa a los valores teóricos.

2. **Ampliación del modelo Físico y del hamiltoniano.** Para aumentar el realismo y la precisión de la simulación, se podría extender el modelo físico:

- **inclusión de interacciones de largo alcance.** Añadir términos de hopping a terceros vecinos o más allá para refinar la descripción de la estructura de bandas, especialmente lejos de los puntos de alta simetría.
- **estudio de sistemas con interacciones.** Avanzar más allá del modelo de partículas independientes e incluir términos de interacción electrón-electrón (por ejemplo, mediante un modelo de Hubbard ¹), abordando así el régimen de sistemas fuertemente correlacionados, donde los métodos cuánticos prometen una mayor ventaja.

¹El modelo de Hubbard es un modelo fundamental en física de la materia condensada que describe la competencia entre la energía cinética de los electrones, que les permite “saltar” entre los sitios de una red (parámetro t), y la fuerte repulsión coulombica que sienten cuando dos de ellos ocupan el mismo sitio (Hubbard, 1963).

3. **Exploración de materiales 2D alternativos.** El marco metodológico desarrollado en este Trabajo se puede aplicar directamente al estudio de otros materiales bidimensionales de gran interés:
 - **bicapa de grafeno.** Estudiar la estructura de bandas de la bicapa de grafeno, que presenta una física más rica, incluyendo la posibilidad de abrir un gap energético mediante la aplicación de un campo eléctrico externo (McCann, 2013).
4. **Ejecución en hardware cuántico real y optimización.** El siguiente paso lógico sería pasar de la simulación a la ejecución en un procesador cuántico real de IBM. Esto implicaría:
 - ejecutar los cálculos en un QPU físico y comparar los resultados con los obtenidos en las simulaciones (ideal y ruidosa).
 - optimizar el circuito para un hardware específico, utilizando técnicas de transpilación avanzadas para minimizar el número de puertas CNOT y adaptar el circuito a la topología de conectividad de los cúbits del procesador.
5. **Investigación de algoritmos y ansätze alternativos.** Finalmente, se podría explorar la mejora del propio algoritmo variacional:
 - **subspace-search variational quantum eigensolver (SSVQE).** Implementar SSVQE para calcular el estado fundamental y los estados excitados simultáneamente dentro de una única optimización, lo cual podría ser más eficiente que el enfoque secuencial de VQD utilizado.
 - **ansätze adaptativos (ADAPT-VQE).** Utilizar ansätze que se construyen de forma iterativa y adaptativa según el gradiente del sistema, lo que puede resultar en circuitos más compactos y precisos que los ansätze de estructura fija como EfficientSU2.

Estas líneas de investigación no solo permitirían profundizar en la aplicación de la computación cuántica a la ciencia de materiales, sino que también contribuirían a abordar algunos de los desafíos más importantes en el camino hacia la ventaja cuántica en problemas de relevancia científica y tecnológica.

Bibliografía

- Abrams D.S., Lloyd S. (1997) *Simulation of Many-Body Fermi Systems on a Universal Quantum Computer*. Physical Review Letters, 83(26), 5162.
- Abrams, D. S., Lloyd, S. (1999). *Quantum Algorithm for Obtaining the Eigenvalues and Eigenvectors of a Hamiltonian*. Physical Review Letters, 83(24), 5162–5165.
- Akama S. (2015) *Elements of Quantum Computing* Springer. New York.
- Alonso M., Finn E.J. (1976) *Física Volumen III: Fundamentos cuánticos y estadísticos*. Fondo Educativo Interamericano. México.
- Anderson P. W. (1972) *More is Different*. Science, New Series, Vol. 177, No. 4047 (Aug. 4, 1972), pp. 393-396
- Ashcroft N.W., Mermin N.D. (1976) *Solid State Physics*. Harcourt College Publishers. Orlando.
- Balandin, A. A. (2011). *Thermal properties of graphene and nanostructured carbon materials*. Nature Materials, 10(8), 569–581.
- Barnett S.M. (2009) *Quantum Information*. Oxford University Press. Londres.
- Bauer B., Bravyi S., Motta M., Chan G.K. (2020) *Quantum algorithms for quantum chemistry and quantum materials science*. arXiv:2001.03685v2 [quant-ph]
- Benenti G., Casati G., Strini G. (2005) *Principles of Quantum Computation and Information*. World Scientific Publishing.
- Bernien H., Schwartz S., Keesling A., Levine H., Omran A., Pichler H., Choi S., Zibrov A.S., Endres M. Greiner M., Vuletic V., Lukin M.D. (2017) *Probing many-body dynamics on a 51-atom quantum simulator*. arXiv:1707.04344v2 [quant-ph]
- Bloch, F. (1929). *Über die Quantenmechanik der Elektronen in Kristallgittern (On the Quantum Mechanics of Electrons in Crystal Lattices)*. Zeitschrift für Physik, 52(7–8), 555–600.
- Bohr, N. (1913). *On the Constitution of Atoms and Molecules*. Philosophical Magazine, 26(151), 1–25.

- Bolotin, K. I., Sikes, K. J., Jiang, Z., Klima, M., Fudenberg, G., Hone, J., Kim, P., Stormer, H. L. (2008). *Ultrahigh electron mobility in suspended graphene*. Solid State Communications, 146(9-10), 351–355.
- Bonaccorso, F., Sun, Z., Hasan, T., Ferrari, A. C. (2010). *Graphene photonics and optoelectronics*. Nature Photonics, 4(9), 611–622.
- Bostwick, A., Ohta, T., Seyller, T., Horn, K., Rotenberg, E. (2007) *Quasiparticle dynamics in graphene*. Nature Physics, 3(1), 36–40.
- Bragg, W. H., Bragg, W. L. (1913). *The Structure of the Diamond*. Proceedings of the Royal Society of London. Series A, 89(610), 277–291.
- Brillouin, L. (1930). *Les électrons libres dans les métaux et le rôle des réflexions de Bragg*. (Los electrones libres en los metales y el papel de las reflexiones de Bragg). Journal de Physique et le Radium, 1(11), 377-400.
- Broyden C.G. (1970). *The convergence of a class of double-rank minimization algorithms*. Journal of the Institute of Mathematics and its Applications, vol. 6, no. 1, pp. 76-90.
- Callister, W. D., Rethwisch, D. G. (2018). *Materials Science and Engineering: An Introduction*. Wiley. Hoboken, NJ.
- Carrasco Ramírez J.G. (2021) *The Role of Graphene in Advancing Quantum Computing Technologies*, Annu. Rep.-Aust. Inst. Criminol, 4(1), pp. 62-77.
- Castro Neto A.H., Guinea F., Peres N.M.R., Novoselov K.S. y Geim A.K. (2008) *The electronic properties of graphene*, Rev. Mod. Phys., Volume 81, Number. 1.
- Cauchy A.L. (1847). *Méthode générale pour la résolution des systèmes d'équations simultanées* (Método general para la resolución de sistemas de ecuaciones simultáneas). Comptes Rendus Hebd. Seances Acad. Sci. 25, 536–538. O C I, 10 (383), 399–402.
- Cerezo M., Arrasmith A., Babbush R., Benjamin S.C., Endo S., Fujii K., McClean J.R., Mitarai K., Yuan X., Cincio L. Coles P.J. (2021) *Variational quantum algorithms*. arXiv:2012.09265v2 [quant-ph]
- Cohen, M. L., Bergstresser, T. K. (1966) *Band Structures and Pseudopotential Form Factors for Fourteen Semiconductors of the Diamond and Zinc-Blende Structures*. Physical Review, 141(2), 789–796.

- Combarro E.F., González-Castillo S. (2023) *Practical Guide to Quantum Machine Learning and Quantum Optimization*. Packt Publishing. UK.
- Deacon R.S., Chuang K.-C., Nicholas R.J., Novoselov K.S., Geim A.K. (2007) *Cyclotron Resonance study of the electron and hole velocity in graphene monolayer* Phys. Rev. B 76, 081406(R).
- Deutsch, D. (1985). *Quantum theory, the Church-Turing principle and the universal quantum computer*. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400(1818), 97–117.
- Dirac P.A.M. (1930) *Note on exchange phenomena in the Thomas atom*. Mth. Proc. Cambridge Philos. Soc. 26, 376-385
- Dirac, P. A. M. (1930). *The Principles of Quantum Mechanics*. Clarendon Press. Oxford.
- Dumitrescu, E. F., et al. (2018). *Cloud Quantum Computing of an Atomic Nucleus*. Physical Review Letters, 120(21), 210501.
- Einstein, A., Podolsky, B., Rosen, N. (1935). *Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?*. Physical Review, 47(10), 777–780.
- Eisberg R., Resnick R. (1979) *Física cuántica*. Ed. Limusa. México.
- Ewald, P. P. (1921). *Das 'reziproke Gitter' in der Strukturtheorie. (La 'red recíproca' en la teoría de la estructura)*. Zeitschrift für Kristallographie, 56, 129-156.
- Fan Y., Liu J., Li Z., Yang J. (2021) *A Quantum Algorithm to Calculate Band Structure at the EOM Level of Theory*. 10.48550/arXiv.2109.01318.
- Feynman R.P. (1982) *Simulating Physics with Computers*. International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982
- Fletcher R. (1970). *A new approach to variable metric algorithms*. The Computer Journal, vol. 13, no. 3, pp. 317-322.
- Fock, V. (1930). *Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems (Método de aproximación para la solución del problema cuántico de muchos cuerpos)*. Zeitschrift für Physik, 61(1-2), 126–148.

- Geim, A., Novoselov, K.S. (2007). *The Rise of Graphene. Nature materials*, Nature Materials, Volume 6. pp. 183-91.
- Geim, A. K. (2009). *Graphene: Status and Prospects*. Science, 324(5934), 1530–1534.
- Geis M.W., Angus J.C. (1992). *Diamond film semiconductors*. Scientific American, 267(4), 84-89
- Georgescu, I. M., Ashhab, S., Nori, F. (2014). *Quantum simulation*. Reviews of Modern Physics, 86(1), 153.
- Goldfarb D. (1970). *A family of variable-metric methods derived by variational means*. Mathematics of Computation, vol. 24, no. 109, pp. 23-26.
- González-Viñas W., Mancini H.L. (2002) *Ciencia de los Materiales*. Universidad de Navarra. <http://fisica.unav.es/>. Visitado 16/07/2025.
- Goringe C.M., Bowler D.R., Hernández E. (1997) *Tight-binding modelling of materials*. Reports on Progress in Physics, Volume 60, Number 12.
- Google AI Quantum and Collaborators. (2020). *Hartree-Fock on a superconducting qubit quantum computer*. Science, 369(6507), 1084-1089.
- Grimsley, H.R., Economou S.E., Barnes E., Mayhall N.J. (2019). *An adaptive variational algorithm for finding the ground state of a molecular Hamiltonian*. Nature Communications, vol. 10, artículo 3007.
- Hartree, D. R. (1928). *The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods*. Proceedings of the Cambridge Philosophical Society, 24(1), 89–110.
- Higgott O., Wang D., Brierley S. (2019) *Variational quantum computation of excited states*. Quantum 2019;3:156. doi:10.22331/q-2019-07-01-156.
- Hohenberg P., Kohn W. (1964) *Inhomogeneous Electron Gas*. Physical Review, Volume 136, Number 3B.
- Hubbard, J. (1963). *Electron Correlations in Narrow Energy Bands*. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 276(1365), 238-257.

Hughes C., Isaacson J., Oerry A., Sun R. F., Turner J. (2021) *Quantum Computing for the Quantum Curious*. Springer. Switzerland.

(2023). *IBM Quantum Platform*. <https://quantum.ibm.com/> Visitado 16/07/2025

IBM Research (2025) *Materials Discovery* <https://research.ibm.com/topics/materials-discovery>. Visitado 16/07/2025.

Imada M., Fujimori A., Tokura Y. (1988) *Metal-insulator transitions*. Reviews of Modern Physics. Vol. 70, Issue 1039

IUPAC. (2019). *Compendium of Chemical Terminology (the "Gold Book")*. (2.3.3 ed.). International Union of Pure and Applied Chemistry. <https://goldbook.iupac.org/> Visitado 16/07/2025.

Jacquier A., Kondratyev O. (2022) *Quantum Machine Learning and Optimisation in Finance*. Packt Publishing. UK.

Jones R.O. (2015) *Density functional theory: its origins, rise to prominence, and future*. Reviews of Modern Physics, Volume 87, Lujy.

Kandala, A., et al. (2017). *Hardware-efficient variational quantum eigensolver for small molecules*. Nature, 549(7671), 242–246.

Kaur M., Tripathi P.K. (2018) *The basic properties of graphene and its applications* IJRAR Volume 5, Issue 4 December 2018

Kingma D.P., Ba J. L. (2015). *Adam: A Method for Stochastic Optimization*. Conferencia Internacional sobre Representaciones del Aprendizaje (ICLR) de 2015.

Kitagawa T., Broome M., Fedrizzi A., Rudner M., Berg E., Kassal I., Aspuru-Guzik A., Demler E., White A. (2012). *Observation of Topologically Protected Bound States in Photonic Quantum Walks*. Nature communications. 3. 882. 10.1038/ncomms1872.

Kittel, C. (2005) *Introduction to Solid State Physics (8th ed.)*. John Wiley Sons. Hoboken.

Kroll, J.G., Uilhoorn, W., van der Enden, K.L. et al. (2018) *Magnetic field compatible circuit quantum electrodynamics with graphene Josephson junctions*. Nat Commun 9, 4615. <https://doi.org/10.1038/s41467-018-07124-x>

- Kohn W., Sham L.J. (1965) *Self-Consistent Equations Including Exchange and Correlation Effects*. Physical Review, Volume 140, Number 4B
- Kohn W. (1999) *Nobel Lecture: Electronic structure of matter-wave functions and density functionals*. Reviews of Modern Physics, Vol. 71, No. 5 (199), pp. 1253-1266.
- Konschuh S., Gmitra M., Kochan D., Fabian J. (2012) *Theory of spin-orbit coupling in bilayer graphene*. arXiv:1111.7223v2 [cond-mat.mes-hall]
- Kronig, R. de L., Penney, W. G. (1931). *Quantum Mechanics of Electrons in Crystal Lattices*. Proceedings of the Royal Society of London. Series A, 130(814), 499–513.
- Lanyon B.P., Whitfield J.D., Guillet G.G., Goggin M.E., Almeida M.P., Kassal I., Biamonte J.D., Mohseni M., Powell B.J., Barbieri M., Aspuru-Guzik A., White A.G. (2010) *Towards Quantum Chemistry on a Quantum Computer* Nature Chemistry, 2(2), 106-111.
- Lanyon B.P., Hempel C., Nigg D., Müller M., Gerritsma R., Zähringer F., Schindler P., Berreiro J.T., Rambach M., Kirchmair G., Hennrich M., Zoller P., Blatt R. Roos C.F. (2011) *Universal digital quantum simulation with trapped ions* arXiv:1109.1512v2 [quant-ph]
- Lee, C., Wei, X., Kysar, J. W., Hone, J. (2008). *Measurement of the Elastic Properties and Intrinsic Strength of Monolayer Graphene*. Science, 321(5887), 385–388.
- Leggett A.J. (2010) *Lecture 5: Graphene: Electronic band structure and Dirac fermions*. <http://web.physics.ucsb.edu/phys123B/w2015/leggett-lecture.pdf>. Visitado 16/07/2025.
- LeSar R. (2013) *Introduction to Computational Materials Science: Fundamentals to Applications*. Cambridge University Press. UK.
- Levine I.N. (2013) *Quantum Chemistry, 7th ed.* Pearson Prentice Hall.
- Martin R.M. (2020) *Electronic Structure. Basic Theory and Practical Methods*. Cambridge University Press. UK.
- McCann, E., Koshino, M. (2013). *The electronic properties of bilayer graphene*. Reports on Progress in Physics, 76(5), 056503.

- McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C., Yuan, X. (2020). *Quantum computational chemistry*. Reviews of Modern Physics, 92(1), 015003.
- McClean, J. R., Romero, J., Babbush, R., Aspuru-Guzik, A. (2016). *The theory of variational hybrid quantum-classical algorithms*. New Journal of Physics, 18(2), 023023.
- Morozov, S. V., Novoselov, K. S., Katsnelson, M. I., Schedin, F., Elias, D. C., Jaszczak, J. A., Geim, A. K. (2008). *Giant intrinsic carrier mobilities in graphene and its bilayer*. Physical Review Letters, 100(1), 016602.
- Nair, R. R., Blake, P., Grigorenko, A. N., Novoselov, K. S., Booth, T. J., Stauber, T., Peres, N. M. R., Geim, A. K. (2008). *Fine Structure Constant Defines Visual Transparency of Graphene*. Science, 320(5881), 1308.
- Nakanishi K.M., Mitarai K., Fujii K. (2019) *Subspace-search variational quantum eigen solver for excited states*. Physical Review Research 1, 033062 (2019).
- National Academies of Sciences, Engineering, and Medicine. (2019). *Quantum Computing: Progress and Prospects*. The National Academies Press. Washington, DC. <https://doi.org/10.17226/25196>.
- Nelder J.A., Mead R. (1965). *A simplex method for function minimization*. The Computer Journal, vol. 7, no. 4, pp. 308-313.
- Nielsen M.A., Chuang I.L. (2011) *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. Cambridge.
- Novoselov K.S., Geim A.K. et al. (2004) *Electric Field in Atomically Thin Carbon Films*, Science Oct 22, 2004.
- Novoselov, K. S., Geim, A. K., Morozov, S. V., Jiang, D., Katsnelson, M. I., Grigorieva, I. V., Dubonos, S. V., Firsov, A. A. (2005). *Two-dimensional gas of massless Dirac fermions in graphene*. Nature, 438(7065), 197–200.
- Ohgoe T., Iwakiri H., Kohda M., Ichikawa K., Nakagawa Y.O., Valencia H.O., Koh S. (2024) *Demonstrating quantum computation for quasiparticle band structures*. Physical Review Research Volume 6, L022022 (2024)
- O'Malley P.J., Babbush R., Kivlichan I.D. et al. (2016) *Scalable Quantum simulation of molecular energies*. Physical Review X, Volume 6, Number 3, p. 031 007.

- Pauli, W. (1925). *Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren* (On the Connexion of the Completion of Electron Groups in an Atom with the Complex Structure of Spectra). *Zeitschrift für Physik*, 31(1), 765–783.
- Payne, M. C., Teter, M. P., Allan, D. C., Arias, T. A., Joannopoulos, J. D. (1992). *Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients*. *Reviews of Modern Physics*, 64(4), 1045–1097.
- Peruzzo A., McClean J., Shadbolt P., Yung M., Zhou X., Love P.J., Aspuru-Guzik A., O'Brien J.L. (2014) *A variational eigenvalue solver on a photonic quantum processor*. *Nature Communications* 2014;5(1)
- Phillips, J. C., Kleinman, L. (1959). *New Method for Calculating Wave Functions in Crystals and Molecules*. *Physical Review*, 116(2), 287–294.
- M. J. D. Powell M.J.D. (1994). *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Dentro del libro *Advances in Optimization and Numerical Analysis* (editado por S. Gomez y J. P. Hennart), pp. 51-67, Springer. México.
- Preskill J. (2018). *Quantum Computing in the NISQ era and beyond*. *Quantum* 2. 2018-08-06, volume 2, page 79. arXiv:1801.00862v3
- Python Software Foundation. (2024) *The Python Programming Language* <https://www.python.org> Visitado 16/07/2025
- Reich S., Maultzsc J., Thomsen C., Ordejón P. (2002) *Tight-binding description of graphene* *Phys. Rev. B* 66, 035412.
- Robbins H., Monro S. (1951). *A Stochastic Approximation Method*. *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400-407.
- Roy M. (2015) *Notas: The Tight Binding Method*, Universidad de Rutgers. <http://www.physics.rutgers.edu/~eandrei/chengdu/reading/tight-binding.pdf>. Visitado 16/07/2025.
- Ruiz M-B, Tröger, R. (2017). *Configuration Interaction Study of the ^3P Ground State of the Carbon Atom*. <https://arxiv.org/abs/1707.09738>

- Ryabinkin I.G., Lang R.A., Genin S.N., Izmaylov A.F. (2019). *Iterative Qubit Coupled Cluster approach for quantum chemistry*. Journal of Chemical Theory and Computation, vol. 15, no. 1, pp. 249–255.
- Sakurai J.J. (1994) *Modern Quantum Mechanics* Addison-Wesley. UK.
- Sarma S., Adam S., Hwang E., Rossi, E. (2011). *Electronic transport in two-dimensional graphene*, Reviews of Modern Physics, Volume 83, Number. 2, pp. 407–470.
- Schrödinger, E. (1926). *Quantisierung als Eigenwertproblem* (Quantization as an Eigenvalue Problem). Annalen der Physik, 384(4), 361–376.
- Schwierz, F. (2010). *Graphene transistors*. Nature Nanotechnology, 5(7), 487–496.
- Shanno D.F. (1970). *Conditioning of quasi-Newton methods for function minimization*. Mathematics of Computation, vol. 24, no. 111, pp. 647-656.
- Shannon C.E. (1948) *A mathematical theory of communication*. The Bell System Technical Journal, 27(3), 379-423.
- Sharkey K.L., Chancé A. (2022) *Quantum Chemistry and Computing for the Curious*. Packt Publishing. UK.
- Sherbert K., Jayaraj A., Buongiorno Nardelli M. (2022) *Quantum algorithm for electronic band structures with local tight-binding orbitals*. Nature. Scientific Reports (2022)12:9876
- Sheridan E., Mineh L. Santos R.A., Cubitt T. (2024) *Enhancing density functional theory using the variational quantum eigensolver*. arXiv:2402.18534v1 [quant-ph]
- Singleton J. (2001) *Band Theory and Electronic Properties of Solids*. Oxford University Press. UK.
- Slater J.C., Koster G.F. (1954) *Simplified LCAO Method for the Periodic Potential Problem*. Physical Review, Volume 94, Number 6.
- Sommerfeld, A. (1928). *Zur Elektronentheorie der Metalle auf Grundlage der Fermischen Statistik* ("On the Electron Theory of Metals on the Basis of Fermi Statistics"). Zeitschrift für Physik, 47(1–2), 1–32.

- Spall J.C. (1991). *A Stochastic Approximation Algorithm Using a Random-Direction-Based Difference* Proceedings of the 30th IEEE Conference on Decision and Control, pp. 1554-1558.
- Spall J.C. (1992). *Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation*. IEEE Transactions on Automatic Control, vol. 37, no. 3, pp. 332-341.
- Stolze J., Suter D. (2004) *Quantum Computing: A short Course from Theory to Experiment*. Wiley-Vch GmbH. Weinheim.
- Gilbert S. (2010). *Lecture 21: Eigenvalues and Eigenvectors*. MIT 18.06 Linear Algebra, Spring 2010. Visitado: 16/07/2025
- Szabo A., Ostjund N.S. (1996) *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications. UK.
- Teixeira R.G. (2014) *The Hartree-Fock Method*. Instituto de Física de São Paulo, Universidade de São Paulo.
- Tilly J., Chen H., Cao S., Picozzi D., Setia K. Li Y., Grant E., Wossnig L., Rungger I., Booth G.H., Tennyson J. (2022) *The Variational Quantum Eigensolver: a review of methods and best practices*. preprint arXiv:2111.05176v3 [quant-ph]
- Tipler P.A., Mosca G. (2010) *Física para la ciencia y la tecnología. Física Moderna. 6ª Ed.* Editorial Reverté. Barcelona.
- Toffoli, T. (1980). *Reversible computing*. Technical Memo MIT/LCS/TM-151, MIT Lab for Computer Science.
- Trevisanutto, P. E., Giorgetti, C., Reining, L., Ladisa, M., Olevano, V. (2008). *Ab initio GW quasi-particle band structure of graphene*. Physical Review Letters, 101(22), 226405.
- Von Neumann, J. (1932). *Mathematische Grundlagen der Quantenmechanik*. (Traducción al inglés: Mathematical Foundations of Quantum Mechanics, Princeton University Press, 1955).
- Wallace, P. R. (1947) *The Band Theory of Graphite*, Physical Review, Volume 71, Number 9.

Wang B., Wen J., Wu J., Xie H., Yang F., Wei S., Long G. (2023). *A powered full quantum eigensolver for energy band structures*. 10.48550/arXiv.2308.03134.

Wang X., Xu L. (2022) *Band structure and Quantum Transport of Bent Bilayer Graphene*. Materials 2022, 15, 8664. <https://doi.org/10.3390/ma15238664>

Wigner, E., Seitz, F. (1933). *On the Constitution of Metallic Sodium*. Physical Review, 43(10), 804–810.

Wilson, A. H. (1931). *The Theory of Electronic Semi-Conductors. I*. Proceedings of the Royal Society of London. Series A, 133(822), 458–491.

A. Anexos

A.1. Programa para el cálculo de las bandas de valencia y de conducción

En este anexo se incluye el programa, desarrollados en lenguaje Python, que se ha utilizado para obtener los diferentes resultados de este Trabajo.

```
1 """
2 BandasEnergiaGrafeno.py
3
4 """
5
6 # Instalación de las librerías
7 # (Descomentar si no se han instalado).
8 #!pip install qiskit==1.4.3 qiskit-aer qiskit-algorithms qiskit-ibm-
   runtime qiskit-nature qiskit-optimization
9 #!pip install pylatexenc
10
11 """
12 ***** BandasEnergiaGrafeno *****
13
14 TFM para el Máster en Computación Cuántica
15         UNIR
16         Raúl Medrano Millán
17         Junio 2025
18
19 *****
20 """
21
22 import numpy as np
23 import matplotlib.pyplot as plt
24 import os
25 import csv
26 import re
27 from qiskit.circuit.library import EfficientSU2
28 from qiskit.primitives import Sampler
29 from qiskit.quantum_info import SparsePauliOp
30 from qiskit.transpiler.preset_passmanagers import import
   generate_preset_pass_manager
31 from qiskit_aer.noise import NoiseModel
32 from qiskit_aer.primitives import Estimator
33 from qiskit_algorithms.optimizers import COBYLA, SPSA
34 from qiskit_algorithms.state_fidelities import ComputeUncompute
35 from qiskit_ibm_runtime import QiskitRuntimeService, Session, EstimatorV2
   as IBMEstimator, EstimatorOptions
36 from qiskit_ibm_runtime.fake_provider import FakeTorino
37
38 class BandasEnergiaGrafeno:
39     """
40     Constructor de la clase.
41     Parámetros:
42         t (float): Parámetro de hopping (usamos unidades normalizadas por
   simplicidad).
43         t_prima (float): Parámetro de hopping de los segundos vecinos (
   usamos t/28 por defecto).
44         a (float): Distancia entre átomos de carbono (usamos 1.43 armstrong
   por defecto).
45         vecinos (int): Número de vecinos próximos (usamos 2 por defecto).
   (1: incluye los tres vecinos próximos, 2: incluye también
46         los 6 siguientes).
```

```
47 """
48 def __init__(self, t=1.0, t_prima=1.0/28, a=1.42, vecinos=2):
49     # --- Se inicializan y almacenan los valores iniciales ---
50     if vecinos not in [1,2]:
51         raise Exception("El número de vecinos debe ser 1 para los tres
52 más próximos, o 2 para incluir los seis siguientes")
53     else:
54         self.vecinos = vecinos
55         self.t = t
56         self.t_prima = t_prima
57         self.a = a
58         self.delta = self.calcular_delta(a)
59         self.delta2 = self.calcular_delta2(a)
60         self.energias = np.array([])
61         self.autovalores = np.array([])
62         self.num_points = 0
63         self.evaluacion = 0
64         self.ruido = None
65         self.optimizador = None
66         self.ansatz_str = None
67
68 # --- Cálculo de los vectores del espacio real para los primeros
69 vecinos ---
70 def calcular_delta(self,a):
71     """
72     Cálculo de los vectores del espacio real para los primeros vecinos.
73     Parámetros:
74     a (float): Distancia entre átomos de carbono.
75     Devuelve:
76     delta (ndarray): Vectores con las posiciones del espacio real.
77     """
78     delta = np.array([
79         [a/2, np.sqrt(3)*a/2],
80         [a/2, -np.sqrt(3)*a/2],
81         [-a, 0]
82     ])
83     return delta
84
85 # --- Cálculo de los vectores del espacio real para los segundos
86 vecinos ---
87 def calcular_delta2(self,a):
88     """
89     Cálculo de los vectores del espacio real para los segundos vecinos.
90     Parámetros:
91     a (float): Distancia entre átomos de carbono.
92     Devuelve:
93     delta (ndarray): Vectores con las posiciones del espacio real.
94     """
95     delta2 = np.array([
96         [a*3/2, np.sqrt(3)*a/2],
97         [-a*3/2, -np.sqrt(3)*a/2],
98         [a*3/2, -np.sqrt(3)*a/2],
99         [-a*3/2, np.sqrt(3)*a/2],
100         [0, -a*np.sqrt(3)],
101         [0, a*np.sqrt(3)]
102     ])
103     return delta2
104
105 # --- Función de coste (energía) a optimizar en el algoritmo VQE ---
106 def cost_func(self, i, beta, params, ansatz, operator, estimator,
107 fidelity, wavefuncs, eid, simulacion, fichero_salida):
108     """
109     Función de coste (energía) a optimizar en el algoritmo VQE
110     Parámetros:
111     i (ndarray): Punto del espacio recíproco.
112     beta (float): Hiperparámetro para el cálculo del siguiente
113 estado excitado.
114     params (ndarray): Array de parámetros del ansatz.
115     ansatz (QuantumCircuit): Circuito parametrizado (ansatz).
```

```

111         estimator (EstimatorV2): Instancia de la primitiva del
112         estimator.
113         fidelity (BaseStateFidelity): Instancia de la primitiva del
114         estimador de fidelidad.
115         wavefuncs (ndarray): Lista de estados excitados.
116         eidx (integer): Índice del estado excitado.
117         simulacion (boolean): Si se está simulando o en un ordenador
118         real.
119         fichero_salida (string): Fichero de salida o None.
120     Returns:
121         float: Energía estimada
122     """
123     # --- Ejecución del estimador en función de si es o no una simulación ---
124     if simulacion:
125         result = estimator.run([ansatz], [operator], params).result().
126         values[0]+ \
127         sum([beta*fidelity.run(ansatz, ansatz, params, wavefuncs[
128         idx]).result().fidelities[0] for idx in range(eidx)])
129     else:
130         pub = (ansatz, [operator], [params])
131         result = estimator.run(pubs=[pub]).result()[0].data.evs[0] + \
132         sum([beta*fidelity.run(ansatz, ansatz, params, wavefuncs[
133         idx]).result().fidelities[0] for idx in range(eidx)])
134         self.evaluacion += 1
135
136     # --- Almaceno los datos si se ha proporcionado un fichero de
137     salida ---
138     if fichero_salida is not None:
139         with open(fichero_salida, "a") as f:
140             valores = [i,[eidx,self.evaluacion,result],params,
141             wavefuncs]
142             f.write(";".join(str(x) for sublist in valores for x in
143             sublist) + "\n")
144
145     return result
146
147     # --- Función para la lectura de un fichero externo generado
148     anteriormente por BandasEnergiaGrafeno --- #
149     def leer_fichero_entrada(self,fichero_entrada):
150         """
151         Función para la lectura de un fichero externo generado
152         anteriormente por BandasEnergiaGrafeno.
153         Parámetros:
154             fichero_entrada (string): Fichero de entrada.
155         Devuelve:
156             current_i (ndarray): Punto del espacio recíproco.
157             current_energies (ndarray): Lista de energías.
158             current_eigenvals (ndarray): Lista de autovalores.
159             current_params (ndarray): Lista de parámetros.
160             current_wavefuncs (ndarray): Lista de estados excitados.
161             current_idx (integer): Índice del estado excitado.
162             last_energie (ndarray): Energía del último estado excitado.
163         """
164         current_i = None
165         current_energies = []
166         current_eigenvals = []
167         current_params = []
168         current_wavefuncs = []
169         current_idx = 0
170         last_energie = []
171         # --- Lectura del fichero de entrada ---
172         if fichero_entrada is not None:
173             with open(fichero_entrada, mode='r', newline='') as csvfile:
174                 lector = csv.reader(csvfile, delimiter=';')
175                 # Parámetros generales de la clase
176                 fila = next(lector)
177                 self.t = float(fila[0])

```

```
168         self.t_prima = float(filas[1])
169         self.a = float(filas[2])
170         self.vecinos = int(filas[3])
171         # Parámetros específicos del cálculo
172         filas = next(lector)
173         self.num_points = int(filas[0])
174         self.optimizador = filas[1]
175         self.ansatz_str = filas[2]
176         ruido = filas[3]
177         if ruido == 'None':
178             ruido = None
179         self.ruido = ruido
180         # Lectura de la última fila
181         ultima_fila = None
182         for f in lector:
183             l = len(f)
184             if l == 0:
185                 continue
186             if l == 2:
187                 # Se trata de los autovalores
188                 s = []
189                 s.append(float(f[0]))
190                 s.append(float(f[1]))
191                 current_eigenvals.append(s)
192                 last_energie.append(float(ultima_fila[4]))
193                 current_energies.append(last_energie)
194                 current_params = []
195                 current_wavefuncs = []
196                 self.evaluacion = 0
197                 current_idx = 0
198                 last_energie = []
199                 continue
200             if 1 == int(f[2]) and len(last_energie) == 0:
201                 last_energie.append(float(ultima_fila[4]))
202             ultima_fila = f
203             current_idx = int(ultima_fila[2])
204             current_i = [float(ultima_fila[0]), float(ultima_fila[1])]
205             self.evaluacion = int(ultima_fila[3])
206             current_result = float(ultima_fila[4])
207             for i in range(len(ultima_fila)):
208                 if i > 4:
209                     x = ultima_fila[i]
210                     x = x.strip()
211                     if x.startswith('(') and x.endswith(')'):
212                         numeros = re.findall(r'-(\d+\.\d+)', x)
213                         array_numeros = [float(num) for num in numeros]
214                         current_wavefuncs.extend(array_numeros)
215                     else:
216                         current_params.append(float(x))
217
218             print(f"current_params {current_params}")
219             print(f"current_wavefuncs {current_wavefuncs}")
220             print(f"current_energies {current_energies}")
221             print(f"current_eigenvals {current_eigenvals}")
222             print(f"current_i {current_i}")
223             print(f"current_idx {current_idx}")
224             print(f"evaluacion {self.evaluacion}")
225             print()
226
227         return current_i, current_energies, current_eigenvals,
228             current_params, current_wavefuncs, current_idx, last_energie
229
230     # --- Función para el cálculo efectivo de las bandas de energía del
231     # grafeno con diferentes parámetros --- #
232     def calcular_bandas_energia(self, num_points=4, optimizador='COBYLA',
233                                ansatz_str='EfficientSU2', ruido=None, fichero_entrada=None,
234                                fichero_salida=None, token=None):
235         """
```

```

232     Función para el cálculo efectivo de las bandas de energía del
233     grafeno con diferentes parámetros.
234     Parámetros:
235         num_points (int): Número de puntos en el espacio recíproco.
236         optimizador (string): Optimizador a utilizar.
237         ansatz_str (string): Ansatz a utilizar.
238         ruido (string): Ruido a utilizar.
239         fichero_entrada (string): Fichero de entrada o None.
240         fichero_salida (string): Fichero de salida o None.
241         token (string): Token de IBM Quantum o None.
242     Devuelve:
243         energias (ndarray): Lista de energías.
244         autovalores (ndarray): Lista de autovalores.
245     """
246     self.num_points = num_points
247     self.optimizador = optimizador
248     self.ansatz_str = ansatz_str
249     self.ruido = ruido
250
251     # --- Lectura del fichero de entrada ---
252     current_i, current_energies, current_eigenvals, current_params,
253     current_wavefuncs, current_idx, last_energie = self.
254     leer_fichero_entrada(fichero_entrada)
255     if fichero_entrada is not None:
256         fichero_salida = fichero_entrada
257
258     # --- Comprobación de los parámetros de entrada
259     if num_points < 0:
260         raise Exception("El número de puntos debe ser mayor o igual que
261         0")
262     if optimizador not in ['COBYLA', 'SPSA']:
263         raise Exception("Optimizador no reconocido")
264     if ansatz_str not in ['EfficientSU2']:
265         raise Exception("Ansatz no reconocido")
266     if ruido not in [None, 'FakeTorino']:
267         raise Exception("Ruido no reconocido")
268
269     # Optimizador
270     if (optimizador == 'SPSA'):
271         optimizer = SPSA()
272     elif (optimizador == 'COBYLA'):
273         optimizer = COBYLA(maxiter=100)
274
275     # Ansatz
276     if (ansatz_str == 'EfficientSU2'):
277         ansatz = EfficientSU2(num_qubits=1, reps=1)
278
279     # --- Gráfica de la convergencia de los puntos de la red ---
280     def draw_convergencia(punto, counts, values, steps):
281         mitad = steps.count(1)
282         plt.figure(figsize=(12, 4))
283         plt.plot(counts[mitad:], values[mitad:], label='Estado excitado',
284         color='blue')
285         plt.plot(counts[:mitad], values[:mitad], label='Estado
286         fundamental', color='orange')
287         plt.xlabel('Nº evaluaciones')
288         plt.ylabel('Energía/t')
289         plt.title(f"Convergencia en el punto {punto}")
290         plt.grid(True)
291         plt.legend()
292         plt.show()
293
294     # --- Almacenar los parámetros en el fichero de salida ---
295     if fichero_salida is not None and fichero_entrada != fichero_salida
296     :
297         if os.path.exists(fichero_salida):
298             os.remove(fichero_salida)
299         with open(fichero_salida, "a") as f:

```

```
294         # Almaceno los parámetros generales
295         valores = [self.t, self.t_prima, self.a, self.vecinos]
296         f.write(":".join(str(x) for x in valores) + "\n")
297         # Almaceno los parámetros de este cálculo
298         valores = [self.num_points, self.optimizador, self.ansatz_str
, self.ruido]
299         f.write(":".join(str(x) for x in valores) + "\n")
300
301     # --- Proceder al cálculo ---
302     if token is None:
303         # Noise backend
304         if (self.ruido == None):
305             noise_backend = None
306             estimator = Estimator()
307         elif (self.ruido == 'FakeTorino'):
308             noise_backend = FakeTorino()
309             noise_model = NoiseModel.from_backend(noise_backend)
310             estimator = Estimator(backend_options={"noise_model":
noise_model})
311         # --- VQE ---
312         sampler = Sampler()
313         fidelity = ComputeUncompute(sampler)
314     else:
315         # --- Procesador real ---
316         service = QiskitRuntimeService(channel="ibm_quantum", token=
token)
317         backend = service.least_busy(operational=True, simulator=False)
318         pm = generate_preset_pass_manager(target=backend.target,
optimization_level=3)
319         ansatz_isa = pm.run(ansatz)
320
321     # --- Camino en el espacio recíproco ---
322     posiciones = []
323     G = np.array([0, 0])
324     K = (2*np.pi/(3*self.a)) * np.array([1, 1/np.sqrt(3)])
325     M = (2*np.pi/(3*self.a)) * np.array([1, 0])
326     points = [G, K, M, G]
327     k_vals = []
328     k_vals.append(G)
329     posiciones.append(0)
330     for i in range(len(points) - 1):
331         for alpha in np.linspace(0, 1, num_points+2):
332             k = (1 - alpha) * points[i] + alpha * points[i + 1]
333             if alpha > 0:
334                 k_vals.append(k)
335             posiciones.append(len(k_vals))
336
337     # --- Cálculo de energías ---
338     energies = []
339     energies2 = []
340     for n,i in enumerate(k_vals):
341         # --- Filtrado por los cálculos ya hechos ---
342         if current_i is not None:
343             if (i == current_i).all() and n==len(current_energies):
344                 # Almaceno los valores leídos hasta ahora
345                 energies = current_energies
346                 current_i = None
347                 energies2 = current_eigenvals
348                 print(i,current_idx)
349             else:
350                 continue
351         print(f"Calculando en el punto {n} = {i}")
352         # --- Productos escalares ---
353         a = 0
354         b = 0
355         for k in self.delta:
356             r = np.dot(i, k)
357             a += np.cos(r)
358             b += np.sin(r)
```

```

359         c = 0
360         if (self.vecinos == 2):
361             for k in self.delta2:
362                 r = np.dot(i, k)
363                 c += np.cos(r)
364                 c *= 2*self.t_prima
365         # --- Hamiltoniano ---
366         H = np.array([
367             [c, -(a+1j*b)],
368             [-(a-1j*b), c]
369         ])
370         # --- Hamiltoniano en matrices de Pauli ---
371         if (self.vecinos == 1):
372             operator = SparsePauliOp(["X", "Y"], coeffs = [-a, b])
373         else:
374             operator = SparsePauliOp(["X", "Y", "I"], coeffs = [-a, b,
c])
375
376         # --- Ejecución de VQE ---
377         kk = 2
378         beta = 10
379         x0=2 * np.pi * np.random.random(ansatz.num_parameters)
380         if len(current_params) > 0:
381             x0 = current_params
382             current_params = []
383         results = []
384         wavefuncs = []
385         if len(current_wavefuncs)>0:
386             wavefuncs = [current_wavefuncs]
387             current_wavefuncs = []
388
389         if token is None:
390             # --- En simulación ---
391             for eidx in range(kk):
392                 if current_idx > eidx:
393                     current_idx = 0
394                     results = last_energie
395                     continue
396                 else:
397                     cost = lambda params: self.cost_func(i, beta,
params, ansatz, operator, estimator, fidelity, wavefuncs, eidx, True,
fichero_salida)
398                     result = optimizer.minimize(cost, x0)
399                     wavefuncs.append(result.x)
400                     results.append(estimator.run(ansatz, operator,
parameter_values=result.x).result().values[0])
401                     self.evaluacion = 0
402
403             else:
404                 # --- En computador real ---
405                 operador_isa = operator.apply_layout(layout=ansatz_isa.
layout)
406                 with Session(backend=backend) as session:
407                     options_es = EstimatorOptions()
408                     options_es.default_shots = 1000
409                     options_es.resilience_level = 1
410                     estimator = IBMEstimator(mode=session,options=
options_es)
411                     sampler = Sampler()
412                     fidelity = ComputeUncompute(sampler)
413                     for eidx in range(kk):
414                         if current_idx > eidx:
415                             current_idx = 0
416                             results = last_energie
417                             continue
418                         else:
419                             cost = lambda params: self.cost_func(i, beta,
params, ansatz_isa, operador_isa, estimator, fidelity, wavefuncs, eidx,
False, fichero_salida)

```

```
420         result = optimizer.minimize(cost, x0)
421         wavefuncs.append(result.x)
422         results.append(estimator.run(ansatz_isa,
operador_isa, parameter_values=result.x).result().values[0])
423         self.evaluacion = 0
424
425     # --- Almacenar los resultados obtenidos ---
426     energies.append(sorted(results))
427
428     # --- Autovalores ---
429     eigvals = np.linalg.eigvalsh(H)
430     evalores = sorted(eigvals)
431     energies2.append(evalores)
432
433     # --- Almaceno los datos ---
434     if fichero_salida is not None:
435         with open(fichero_salida, "a") as f:
436             f.write(";\n".join(str(v) for v in evalores) + "\n")
437
438     self.energies = np.array(energies)
439     if len(self.energies) == 0:
440         self.energies = np.array(current_energies)
441     self.autovalores = np.array(energies2)
442     if len(self.autovalores) == 0:
443         self.autovalores = np.array(current_eigenvals)
444
445     return self.energies, self.autovalores
446
447 # --- Función para generar gráficos con los resultados --- #
448 def graficos(self, fichero_entrada=None, simulacion=True, ingles=False):
449     """
450     Función para generar gráficos con los resultados.
451     Parámetros:
452         fichero_entrada (string): Fichero de entrada o None.
453         simulacion (boolean): Si se está simulando o en un ordenador
454         real.
455         ingles (boolean): Si se quieren los gráficos en inglés o no.
456     Devuelve:
457         Nada
458     """
459     if fichero_entrada is not None:
460         # Los valores de entrada se sobrescriben con los valores
461         # almacenados en el fichero.
462         current_i, self.energies, self.autovalores, current_params,
463         current_wavefuncs, current_idx, last_energie = self.
464         leer_fichero_entrada(fichero_entrada)
465         self.energies = np.array(self.energies)
466         self.autovalores = np.array(self.autovalores)
467         if len(self.energies)==0 or len(self.autovalores)==0 or len(self.
468         energias)!=len(self.autovalores):
469             raise Exception("No se han calculado correctamente las bandas
470             de energía")
471         energies = self.energies
472         kk = len(energies[0])
473         energies2 = self.autovalores
474         # Valores del eje X
475         G = np.array([0, 0])
476         K = (2*np.pi/(3*self.a)) * np.array([1, 1/np.sqrt(3)])
477         M = (2*np.pi/(3*self.a)) * np.array([1, 0])
478         points = [G, K, M, G]
479         k_vals = []
480         k_vals.append(G)
481         posiciones = []
482         posiciones.append(0)
483         for i in range(1, len(points)):
484             dist = np.linalg.norm(points[i] - points[i-1])
485             posiciones.append(posiciones[-1] + dist)
486         x_vals = []
```

```

482         for i in range(3):
483             xi = np.linspace(posiciones[i], posiciones[i+1], self.
num_points+1, endpoint=False)
484             x_vals.extend(xi)
485             x_vals.append(posiciones[-1]) # Añadir el último punto  $\Gamma$  final
486             x_vals = np.array(x_vals)
487             # Graficar
488             plt.figure(figsize=(8, 6))
489             if ingles:
490                 label = "Conduction band (eigenvalues)"
491             else:
492                 label = "Banda de conducción (autovalores)"
493             plt.plot(x_vals, energies2[:, 1], linestyle='--', linewidth=1,
color='cyan', label=label)
494             if ingles:
495                 label = "Valence band (eigenvalues)"
496             else:
497                 label = "Banda de valencia (autovalores)"
498             plt.plot(x_vals, energies2[:, 0], linestyle='--', linewidth=1,
color='gold', label=label)
499             for i in range(kk):
500                 color = 'blue'
501                 if kk == 2:
502                     if i == 1:
503                         if ingles:
504                             label = "Valence band (VQD)"
505                         else:
506                             label = "Banda de valencia (VQD)"
507                     else:
508                         color = 'orange'
509                         if ingles:
510                             label = "Conduction band (VQD)"
511                         else:
512                             label = "Banda de conducción (VQD)"
513                     else:
514                         label = ""
515             plt.plot(x_vals, energies[:, i], '+', markersize=5, color=color
, label=label)
516             plt.xticks(posiciones, [" $\Gamma$ ", "K", "M", " $\Gamma$ "])
517             if ingles:
518                 label = "Energie/t"
519             else:
520                 label = "Energía/t"
521             plt.ylabel(label)
522             if simulacion:
523                 if ingles:
524                     titulo = "Energy Bands of Graphene (Simulation)"
525                 else:
526                     titulo = "Bandas de Energía del Grafeno (Simulación)"
527             else:
528                 if ingles:
529                     titulo = "Energy Bands of Graphene (Real calculation)"
530                 else:
531                     titulo = "Bandas de Energía del Grafeno (Cálculo real)"
532             plt.suptitle(titulo, y=0.97, fontsize=16, style='italic')
533             if ingles:
534                 texto = f"Neighbors: {self.vecinos}\nNumber of points: {self.
num_points}\nOptimizer: {self.optimizador}\nAnsatz: {self.ansatz_str}"
535             else:
536                 texto = f"Vecinos: {self.vecinos}\nNúmero de puntos: {self.
num_points}\nOptimizador: {self.optimizador}\nAnsatz: {self.ansatz_str}"
537             plt.title(texto)
538             plt.grid(True)
539             plt.legend()
540             plt.tight_layout()
541             plt.show()
542
543

```

```
544 # *****
545 # Ejemplos de uso
546 # *****
547
548
549 # Ejecución para un único vecino, en emulación, con el optimizador COBYLA,
550 # con 4 puntos entre cada punto del espacio recíproco
551 # y con presentación de resultados en un gráfico.
552 grafeno = BandasEnergiaGrafeno(vecinos=1)
553 e,e2 = grafeno.calcular_bandas_energia(optimizador='SPSA')
554 grafeno.graficos()
555
556 # Ejecución para dos vecinos, en procesador real, con el optimizador COBYLA
557 # con 4 puntos entre cada punto del espacio recíproco
558 # y con los resultados almacenados en un fichero externo (resultado.txt).
559 grafeno = BandasEnergiaGrafeno()
560 e, e2 =grafeno.calcular_bandas_energia(fichero_salida="resultados.txt",
561 token("<IBM user token>"))
562
563 # Ejecución para dos vecinos, en simulación con ruido, con el optimizador
564 SPSA,
565 # con 6 puntos entre cada punto del espacio recíproco
566 # y presentación de los resultados numéricos.
567 grafeno = BandasEnergiaGrafeno()
568 e, e2 =grafeno.calcular_bandas_energia(num_points=6,ruido='FakeTorino',
569 optimizador='SPSA')
570 print(e)
571 print(e2)
572
573 # Lectura del fichero externo creado anteriormente
574 # y presentación de los datos en un gráfico con los textos en inglés.
575 grafeno = BandasEnergiaGrafeno()
576 grafeno.graficos(fichero_entrada='resultados.txt', ingles=True)
```

Listing A.1: Programa BandasEnergiaGrafeno