

UNIVERSIDADE FEDERAL DE
ALFENAS

Instituto de Ciências
Exatas

RAPHAEL CÉSAR DE SOUZA PIMENTA

**Estudo da viabilidade do uso da computação
quântica para análise de propriedades de sólidos
periódicos**

Poços de Caldas

2022

Raphael César de Souza Pimenta

**Estudo da viabilidade do uso da computação quântica
para análise de propriedades de sólidos periódicos**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre-Mestre em Física pela Universidade Federal de Alfenas. Área de concentração: Física da Matéria Condensada

Orientador: Anibal Thiago Bezerra

Poços de Caldas/MG

2022

Sistema de Bibliotecas da Universidade Federal de Alfenas
Biblioteca Campus Poços de Caldas

Pimenta, Raphael César de Souza .

Estudo da viabilidade do uso da computação quântica para análise de propriedades de sólidos periódicos / Raphael César de Souza Pimenta. - Poços de Caldas, MG, 2022.

92 f. : il. -

Orientador(a): Anibal Thiago Bezerra.

Dissertação (Mestrado em Física) - Universidade Federal de Alfenas, Poços de Caldas, MG, 2022.

Bibliografia.

1. Computação Quântica. 2. Algoritmos Quânticos. 3. Física da Matéria Condensada. 4. Computação Quântica Aplicada a Materiais. I. Bezerra, Anibal Thiago, orient. II. Título.

“ESTUDO DA VIABILIDADE DO USO DA COMPUTAÇÃO QUÂNTICA PARA ANÁLISE DE PROPRIEDADES DE SÓLIDOS PERIÓDICOS”

A Banca examinadora abaixo-assinada aprova a Dissertação apresentada como parte dos requisitos para a obtenção do título de Mestre em Física pela Universidade Federal de Alfenas. Área de concentração: Física da Matéria Condensada.

Aprovada em: 07 de março de 2022

Prof. Dr. Anibal Thiago Bezerra

Instituição: Universidade Federal de Alfenas

Prof. Dr. Daniel Zini Rossatto

Instituição: Universidade Estadual Paulista

Prof. Dr. Rodrigo Rocha Cuzinatto

Instituição: Universidade Federal de Alfenas



Documento assinado eletronicamente por **Anibal Thiago Bezerra, Professor do Magistério Superior**, em 07/03/2022, às 16:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rodrigo Rocha Cuzinatto, Professor do Magistério Superior**, em 07/03/2022, às 16:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Daniel Zini Rossatto, Usuário Externo**, em 07/03/2022, às 16:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.unifal-mg.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0687782** e o código CRC **4F21C44C**.

AGRADECIMENTOS

Agradeço ao meu professor e orientador Dr. Anibal Thiago Bezerra, que com muitos ensinamentos e incentivos permitiu que eu fosse capaz de chegar até aqui. Me ensinando e motivando em todo momento durante esses dois anos, e também durante toda a graduação. Um grande carinho e meus sinceros agradecimentos.

Agradeço meus companheiros de trabalho Lucas Marques Fagundes da Silva e Leticia Carolaine Silva Faria que estavam sempre disponíveis para discussões e ajudas rotineiras. Assim como todas as pessoas que sabem que contribuíram para este trabalho de alguma forma, em especial à Franciele Cristina Galle.

Agradeço a IBM por disponibilizar tantos materiais e acesso aos seus computadores para simulações e análises, e ao Laboratório de Desenvolvimento em Física Teórica da UNIFAL-MG, assim como a própria instituição da Universidade Federal de Alfenas.

Agradeço também a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por financiar o presente trabalho.

Agradeço a Deus por ter guiado tantas pessoas ao meu caminho que me auxiliaram, assim como aos meus pais por todo o apoio.

Todos àqueles que de alguma forma, direta ou indiretamente, contribuíram para minha formação e caminho tem minha profunda gratidão.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

*“Se você pretende impor
sua vontade ao mundo,
deve exercer controle
sobre aquilo em que acredita.”
([ROTHFUSS, 2007](#), p. 75)*

RESUMO

A física de materiais tem sido desenvolvida ao longo das últimas décadas com grande ajuda de simulações digitais nos supercomputadores que temos disponíveis no século atual. Porém, é certo que há conhecidos limites de até onde podem chegar essas simulações e quais sistemas somos capazes de simular em um tempo de computação eficaz. A computação quântica surge com promessas de romper certas barreiras da computação clássica, aumentando a capacidade computacional disponível para as mais diversas tarefas e, possivelmente, uma delas será a física de materiais. Em nosso trabalho realizamos uma revisão bibliográfica do que é feito atualmente para simulação de materiais periódicos através da computação quântica, assim como também realizamos algumas simulações em computadores quânticos disponíveis para acesso público. Um grande ponto de nosso levantamento foi que, dentre todas as aplicações da computação quântica, a aplicação em física de sólidos periódicos ainda é modesta, sendo a aplicação em sistemas moleculares uma das maiores fontes de trabalhos e referências. Pudemos realizar com o que descobrimos, simulações para a estrutura de bandas de alguns átomos, como o silício e o grafeno em um simulador local. Também pudemos simular em uma máquina quântica *in silico* um modelo generalizado de Hamiltoniano para estudar o seu comportamento a medida que variávamos seus parâmetros. Para o grafeno, fomos capazes de calcular sua estrutura em um simulador quântico local e o resultado foi similar ao que era esperado pela literatura. Para o silício obtivemos uma diferença acentuada no seu estado de energia excitado, que acreditamos ser devido a simplicidade de nossa aproximação.

Palavras-chaves: computação quântica; algoritmos quânticos; computação quântica aplicada a materiais.

ABSTRACT

Materials science has been developed over last few decades with great help from digital simulations on the supercomputers that we have available in the current century. However, there are certainly known limits to how far these simulations can go and which systems we can simulate in an effective computation time. Quantum computing comes with promises to break certain barriers of classical computing, increasing the computing capacity available for the most diverse tasks, and, possibly, one of them will be materials science. In our work, we carried out a bibliographic review of what is currently done for the simulation of periodic materials through quantum computing, as well as we also carried out some simulations on quantum computers available for public access. A major point of our survey was that, among all applications of quantum computing, the application in periodic solid physics is still modest, while the applications on molecules being one of the largest sources of works and references. With that in mind, we were able to perform simulations for the band structure of some atoms, such as silicon and graphene, in a local simulator. We were also able to simulate a generalized Hamiltonian model in a quantum machine *in silico*, to study its behavior as we varied its parameters. For graphene, we were able to compute its structure in a local simulator and the result was similar to that expected in the literature. For silicon, we obtained a marked difference in its excited energy state, which we believe to be due to the simplicity of our approximation.

Keywords: quantum computing; quantum algorithms; material science.

LISTA DE ILUSTRAÇÕES

Figura 1 – Número de publicações por ano com o termo "Quantum Computer" . . .	17
Figura 2 – Número de publicações por ano com os termos Quantum Chemistry (painel superior), Quantum Computational Chemistry (painel intermediário) e Quantum Algorithms for Chemistry (painel inferior)	18
Figura 3 – Número de publicações por ano com o termo <i>Quantum computing for periodic systems</i>	19
Figura 4 – Possíveis estruturas para uma combinação 2D. Em a) temos uma estrutura plana oblíqua, em b) quadrada, em c) retangular, em d) uma retangular centrada em em e) uma hexagonal onde $a=b$ e $\gamma=60^\circ$	21
Figura 5 – As 14 formações tridimensionais possíveis	22
Figura 6 – Diagrama de bandas para diferentes materiais. A extensão vertical indica as regiões com energias permitidas, as áreas em vermelho indicam ocupação de elétrons. No metal, temos a banda de condução vazia enquanto que no semiconductor essa banda está apenas parcialmente preenchida dada a excitação de elétrons da banda de valência para a banda de condução	24
Figura 7 – Relação dos tempos de relaxamento e defasagem. Podemos interpretar o tempo de relaxamento como perturbações orientados perpendicularmente ao plano XY. Quanto ao tempo de defasagem podemos interpreta-lo como perturbações ortogonais ao eixo Z	32
Figura 8 – Esferas de Bloch representando os estados antes e após a transformada quântica de Fourier. As esferas acima representam os estados iniciais e logo abaixo suas respectivas transformadas	41
Figura 9 – Circuito quântico para implementação de uma transformada quântica de Fourier	44
Figura 10 – Circuito simples para representação dos dois registros de qubits, de modo a realizar o "phase kick-back"	46
Figura 11 – Diagrama de circuito para implementação do "phase kick-back", onde sem alterarmos o estado de interesse podemos obter informações através dos qubits ancilla	48
Figura 12 – Diagrama do circuito completo para o algoritmo de estimação de fase, onde após o "phase kick-back" empregamos uma transformada de Fourier inversa para recuperarmos o estado original pelo qual estamos interessados	48

Figura 13 – Diagrama de operação do algoritmo VQE. Inicializamos o ansatz na máquina quântica e realizamos a medida do seu valor esperado, passamos o resultado para um otimizador operado na máquina clássica que busca os parâmetros para o menor auto-valor daquele estado e repetimos a medida através da máquina quântica, com os novos parâmetros retornados pelo otimizador	50
Figura 14 – SSH para uma dimensão	54
Figura 15 – Cadeia hexagonal para o modelo de Haldane onde teremos mais uma possibilidade de salto. Além de dentro da célula unitária e entre vizinhos diretos também são considerados saltos entre vizinhos próximos	56
Figura 16 – Circuito aplicado à E_k para obtenção dos auto-valores, também chamado de ansatz	63
Figura 17 – Circuito quântico de 3 qubits implementado para acelerar o processo de medida, onde medimos os coeficientes de todas as matrizes de Pauli numa mesma execução	63
Figura 18 – Comportamento de diferentes átomos executadas no simulador QASM. (a) representa os dados para o silício, (b) para o germânio e (c) para o α -Sn	64
Figura 19 – Estrutura de bandas calculados por computação clássica	65
Figura 20 – Comportamento do Hamiltoniano a medida que variamos t_k , à esquerda os resultados para o simulador e à direita os resultados para a máquina quântica da IBM	67
Figura 21 – Modelo de Haldane para o grafeno simulado na máquina local para $t_1 > t_2$ à esquerda. À direita temos o valor esperado na literatura para o modelo	69

LISTA DE TABELAS

Tabela 1 – As 7 possibilidades de combinações entre vetores de base e ângulos, formando os 7 sistemas cristalinos distintos da cristalografia	21
--	----

LISTA DE ABREVIATURAS E SIGLAS

NISQ	Noisy Intermediate-Scale Quantum
SSH	Su-Schrieffer-Heeger
DFT	Density Functional Theory
JW	Jordan-Wigner
BK	Bravy-Kitaev
VQE	Variational Quantum Eigensolver
VQD	Variational Quantum Deflation
QPE	Quantum Phase Estimation
IPE	Iterative Phase Estimation
QFT	Quantum Fourier Transform
Qiskit	Quantum Information Software Kit

SUMÁRIO

1	INTRODUÇÃO	13
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Metodologia de Revisão	16
2.2	Sólidos cristalinos periódicos	20
2.2.1	O avanço computacional e o desenvolvimento de materiais	24
2.2.1.1	A teoria do funcional de densidade	25
2.2.1.2	Modelo Tight-Binding	27
2.2.1.3	Modelo Tight-Binding aplicado à computação quântica	27
2.3	Computação Quântica	29
2.4	Computação quântica de sistema moleculares	33
2.5	Algoritmos para computadores quânticos	37
2.6	Computação quântica aplicada ao estudo da matéria condensada	38
2.7	Algoritmos quântico-computacionais aplicados a matéria condensada	38
2.8	Evolução temporal com computação quântica, o método de Trotterização	39
2.9	Algoritmos de estimação de fase	40
2.9.1	Transformada Quântica de Fourier	41
2.9.2	O algoritmo de estimação de fase	46
2.10	Algoritmos quânticos variacionais	49
2.11	Aplicação dos algoritmos na simulação de sistemas sólidos	51
3	SIMULAÇÃO DE SISTEMAS PERIÓDICOS USANDO COMPUTADORES QUÂNTICOS	53
3.1	Modelo Su-Schrieffer-Heeger (SSH)	54
3.2	Modelo de Haldane	56
4	RESULTADOS	58
4.1	Simulação para o Silício	58
4.1.1	Estruturas de bandas	63
4.1.2	Fases topológicas	66
4.2	Simulações para o grafeno	68
	REFERÊNCIAS	71
	APÊNDICE A Apêndice A - Circuitos de 1 e 3 qubits para implementação do SSH	79

1 INTRODUÇÃO

A área de computação quântica tem ganhado notório destaque e desenvolvimento na última década, crescendo amplamente em interesse, investimentos e publicações (GEORGESCU; ASHHAB; NORI, 2014; LUND; BREMNER; RALPH, 2017; MULLER; BLUME-KOHOUT, 2015; PRESKILL, 2018; PRESKILL, 2021). Diferentemente das máquinas clássicas, as quânticas usam qubits ao invés de bits tradicionais e são esses qubits que trazem o potencial para uma revolução computacional. Com sua capacidade de efetuar emaranhamento quântico e superposição de estados, novas possibilidades se abrem para a simulação de materiais e moléculas além de diversas outras áreas.

Por mais que ainda não haja muitas confirmações da real superioridade computacional dessa tecnologia, recentemente dois grandes grupos de pesquisa já trouxeram resultados para defender a chamada “Supremacia Quântica”¹. A demonstração dessa supremacia se trata de executar uma tarefa que é possível de ser feita em um dispositivo quântico, mas que não é possível de ser realizada em um computador clássico dentro de um tempo praticável e eficaz. Um dos grupos empenhado nessa tarefa se trata de pesquisadores da *Google* que demonstraram que suas máquinas seriam capazes de operar um circuito quântico que gerava cadeias de bits consideradas muito difíceis de um computador clássico simular (ARUTE et al., 2019). Já um grupo chinês, em junho de 2021, disponibilizou na plataforma *arXiv* sua pré-publicação onde executaram em sua máquina quântica uma tarefa de amostragem com um tempo de execução de 72 minutos que, segundo os autores, levaria 8 anos em um super computador clássico (WU et al., 2021).

Voltando aos interesses do presente trabalho de pesquisa, tratar sistemas de muitos corpos em um computador clássico é uma tarefa complexa atualmente, principalmente se tratando de problemas de mecânica quântica. A complexidade da função de onda aumenta fatorialmente com o número de partículas que compõem o sistema físico, enquanto que a capacidade computacional clássica aumenta linearmente com o número de bits. Isso torna impraticável simular sistemas quânticos muito grandes dados os limites de hardware da computação clássica.

Uma solução para esse obstáculo surgiu pela contribuição Feynmann, que trouxe uma proposta de usar computação quântica para resolver problemas e simular fenômenos da natureza, pois a natureza em si é quântica (FEYNMAN, 1982). Nesta linha de pensamento, o presente trabalho teve como objetivo verificar se analisar propriedades

¹ Até mesmo o termo “Supremacia Quântica” movimenta a comunidade acadêmica. Há um movimento tentando cunhar um novo termo - “Vantagem Quântica” (*Quantum Advantage*), tentando evitar alusões à termos pejorativos como “Supremacia Branca” (PALACIOS-BERRAQUERO; MUECK; PERSAUD, 2019).

de sólidos periódicos usando computadores quânticos é uma boa alternativa, dado o status da tecnologia atual.

Para representarmos matematicamente um sólido periódico precisamos de uma função de onda que represente todas as partes do sistema de interesse, tal que esta função aumenta de complexidade ao aumentarmos o número de partículas no sistema. Esta complexidade se reflete no tempo e na capacidade computacional necessária para a obtenção das propriedades físicas do sistema. Um sólido periódico, que possui um número extravagante de partículas, mesmo com aproximações utilizadas pela computação clássica como a teoria do funcional de densidade, é algo problemático de ser simulado. Mas grandes problemas vêm com grandes recompensas e entender propriedades fundamentais óticas e de transporte para esses materiais periódicos pode levar ao desenvolvimento de novos materiais, com propriedades ajustadas às demandas atuais em diferentes áreas de aplicação como nas indústrias, fármacos ou eletrônica. De fato, a simulação com computadores quânticos parece ter um enorme potencial de aplicações que vão desde a física da matéria-condensada, cosmologia e até mesmo o caos quântico ([GEORGESCU; ASHHAB; NORI, 2014](#)).

Com isso em mente, decidimos buscar na literatura qual o estado da arte para aplicação de computação quântica no estudo de propriedades físicas de materiais cristalinos periódicos. Iniciamos uma revisão bibliográfica do que é feito para a simulação de materiais quânticos com a tecnologia disponibilizada atualmente, como fazê-las e as limitações a serem superadas. A tecnologia das máquinas quânticas atuais é apelidada de “NISQ”, *Noisy Intermediate-Scale Quantum* ([PRESKILL, 2018](#)) e, como é evidenciado pelo nome, são máquinas de média escala com algumas dezenas de qubits suscetíveis a ruídos e erros em suas execuções.

Além disso, também buscamos simular propriedades de sólidos periódicos nas máquinas quânticas utilizando os métodos encontrados na revisão bibliográfica. Acreditamos que averiguar a viabilidade do uso da computação quântica, assim como dos algoritmos criados atualmente dentro dos limites tecnológicos, é um importante passo no desenvolvimento de novas pesquisas na física de materiais. Nossa pretensão foi de utilizar as máquinas disponibilizadas pela empresa IBM, que conta com um acesso gratuito para execução de algoritmos quânticos em máquinas de até 5 qubits.

O número de trabalhos publicados que tratam da computação quântica aplicada à sólidos cristalinos ainda é relativamente baixo, com a maioria dos trabalhos concentrados em pesquisas com moléculas - na chamada Química Quântica Computacional, principalmente devido a baixa quantidade de átomos destes sistemas. Isso favorece a escrita do Hamiltoniano cuja diagonalização requer baixo custo computacional para as máquinas NISQ. Para sólidos, uma das estratégias se baseia em utilizar um mapeamento do Hamiltoniano com o modelo *Tight Binding*. Este modelo nos permite escrever o Hamiltoniano

como uma combinação linear de matrizes de Pauli, que se adequam perfeitamente à base computacional de spin $1/2$ utilizada pelos computadores quânticos nos quais executamos as simulações.

Para as simulações, escolhemos dois modelos para descrever dois sólidos periódicos diferentes. Primeiro utilizamos o modelo Su-Schrieffer-Heeger (SSH) ([BATRA; SHEET, 2020](#)) para obter as propriedades eletrônicas para o silício. Efetuamos simulações em máquinas quânticas reais para obtermos a estrutura de bandas do material. O segundo modelo empregado foi o de Haldane ([HALDANE, 1988](#)), para o qual utilizamos o grafeno como objeto de estudo e obtivemos resultados para a estrutura eletrônica muito similares aos esperados na literatura para este modelo com o material empregado. Adicionalmente analisamos, usando o modelo SSH, propriedades topológicas em cadeias atômicas unidimensionais, revelando a possibilidade do uso de computadores quântico também nesses sistemas.

Este trabalho foi dividido em três partes. Começamos apresentando como fizemos a busca bibliográfica, o que encontramos na revisão bibliográfica e como funciona a simulação com computação quântica. Apresentamos alguns algoritmos amplamente utilizados para a simulação de sistemas moleculares, e que tem diversas aplicações para o estudo da matéria condensada, assim como alguns trabalhos que implementam tais algoritmos. Seguimos discutindo alguns modelos para modelagem atômica que utilizamos para simular interações atômicas numa rede cristalina. Finalizamos então mostrando quais resultados obtivemos, como eles se comportam comparados ao esperado na literatura por simulações feitas com computação clássica e as principais limitações do método.

2 REVISÃO BIBLIOGRÁFICA

Neste trabalho temos dois focos principais: uma revisão bibliográfica acerca dos métodos utilizados na computação quântica para física de materiais e também a aplicação de técnicas computacionais encontradas nessa revisão na simulação de materiais periódicos usando computadores quânticos.

De modo específico, buscamos entender como a computação quântica pode ser aplicada para estudos na física de materiais e como essa nova tecnologia pode impactar os estudos que já são realizados atualmente com técnicas convencionais como “DFT”, que comentaremos posteriormente no texto. Nos métodos clássicos, diversas aproximações são consideradas e estas não podem ser aplicadas para todos os materiais de forma geral, devido à fatores como demanda de alta precisão, complexidade das funções de onda, ou alto nível de correlação (BASSMAN et al., 2021).

A computação quântica pode trazer novas possibilidades para a obtenção de propriedades de sistemas cristalinos periódicos ao utilizar qubits ao invés de bits. Isso se deve ao fato dos qubits serem capazes de trabalhar com superposição e emaranhamento de estados, o que faz a capacidade de processamento dos computadores quânticos crescer exponencialmente com o número de qubits implementados. Porém, os processadores atuais não tem mais do que algumas dezenas de qubits suscetíveis a erros e com baixo tempo de coerência - chamados de NISQ (PRESKILL, 2018). Esse tipo de operação limita muito os tipos de tarefas que podem ser executadas de modo preciso e reproduzível. Sabemos que, devido a natureza quântica dos qubits, podemos os configurar para armazenar efetivamente uma função de onda que descreve o sistema físico. Nesse sentido, algoritmos para simular um Hamiltoniano, assim como sua evolução temporal, podem ser executados com os processadores atuais. Pensando nisso, é notável que aplicações para a física de materiais podem ser uma das mais promissoras para o desenvolvimento e pesquisa com computação quântica (BASSMAN et al., 2021).

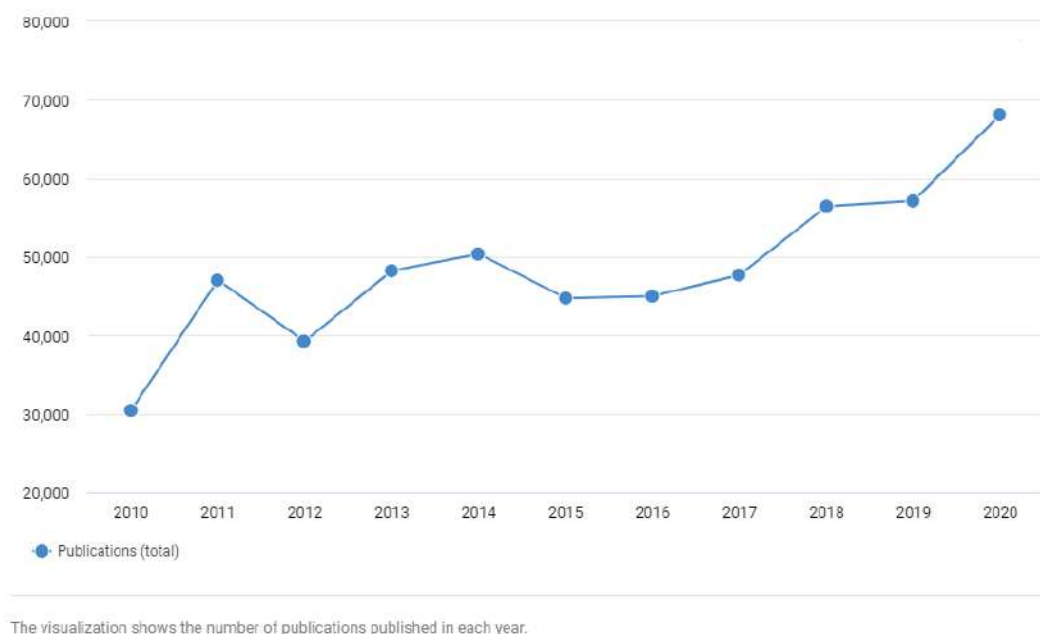
Dessa forma, nesse capítulo, discutiremos a metodologia utilizada na revisão bibliográfica e os resultados dessa revisão sobre o uso da computação quântica em materiais.

2.1 Metodologia de Revisão

Realizamos buscas em repositórios de revistas científicas, como o “Web of Sciece” (WEB OF SCIENCE, 1997), “Google Scholar” (GOOGLE SCHOLAR, 2004) e “ScienceDirect” (SCIENCEDIRECT, 1997), com a expressão-chave genérica “Quantum Computer”. Para termos um panorama de pesquisas relacionadas a computação quântica,

utilizamos uma ferramenta online chamada *Dimensions* fornecida pela *Digital Science & Research Solutions, Inc.* ([DIMENSIONS, 2018](#)). Tal ferramenta se trata de uma plataforma de pesquisa acadêmica em diferentes bases de dados que permite visualizações analíticas de produções científicas. Na figura 1, podemos verificar o crescente número de publicações que citam computadores quânticos na última década.

Figura 1 – Número de publicações por ano com o termo "Quantum Computer"

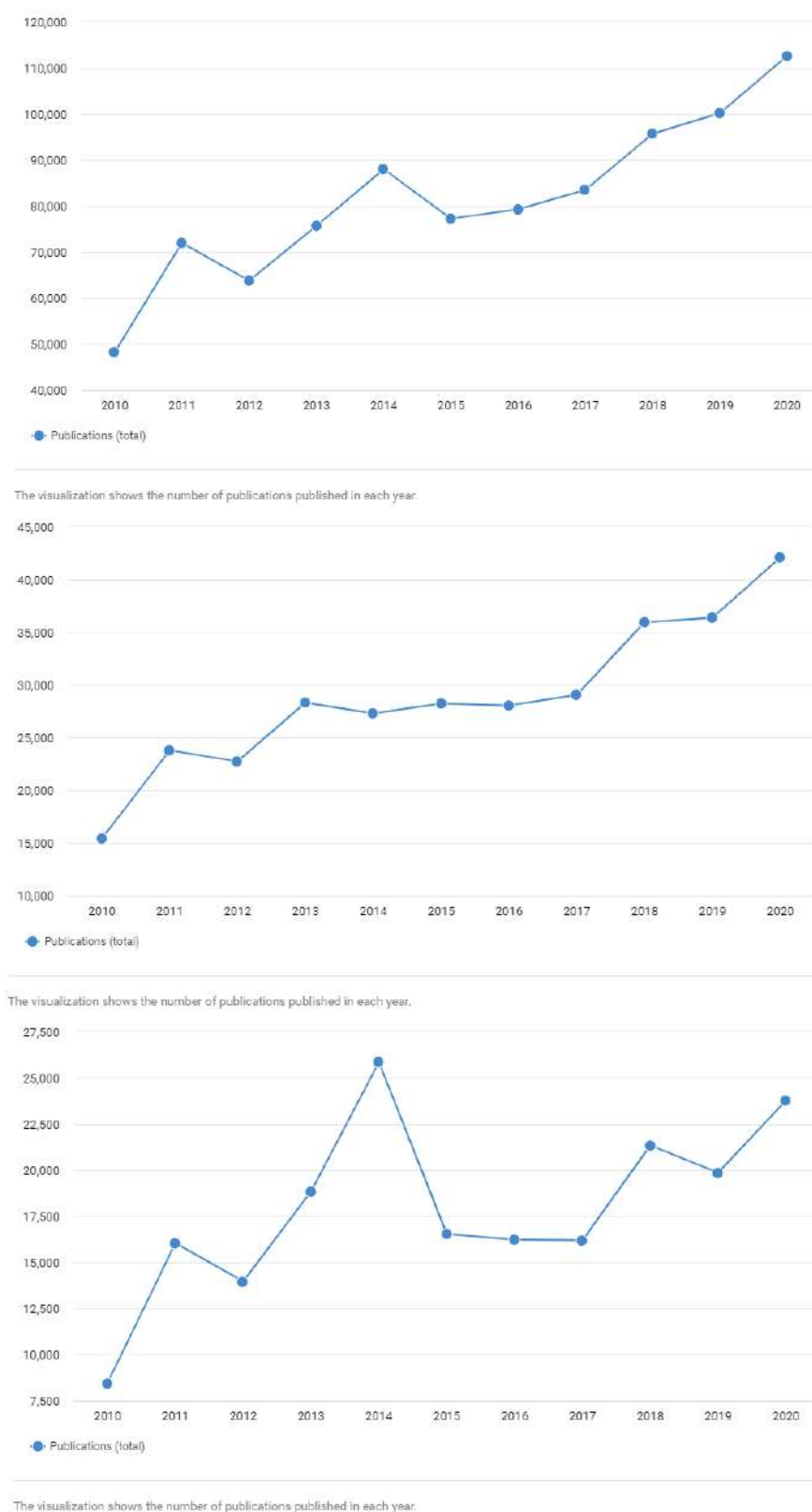


Fonte: ([DIMENSIONS, 2018](#)).

O número de resultados retornado foi relativamente grande, da ordem de 70000 apenas em 2020. Após uma leitura inicial dos artigos de maior impacto encontrados, decidimos refinar nossas buscas mudando a expressão-chave a fim de direcionar a pesquisa para o uso de computação quântica em materiais. Os trabalhos de maior impacto que encontramos e que relacionavam a computação quântica a materiais, tratavam de uma especificidade em Química chamada de Química Quântica Computacional. O termo “Quantum Computational Chemistry” era recorrente em grande parte dos trabalhos e decidimos refinar nossa busca a partir dele. Outros termos que tinham grande recorrência também foram refinados, como “Quantum Chemistry”, “Quantum Computational Chemistry” e “Quantum Algorithms for Chemistry”. A incidência dessas expressões-chave pode ser visualizada na figura 2, onde podemos notar um crescente aumento no número de publicações na última década.

O crescimento na área de química quântica computacional é tão grande que supera o número de publicações encontradas na busca de um termo mais geral, como “Quantum Computer” na figura 1 que, no ano de 2020, retorna 68162 trabalhos, enquanto o termo “Quantum Chemistry” no mesmo ano retorna 112476 trabalhos (conforme mostra o painel superior da figura 2). No entanto, para sistemas periódicos o número de publicações

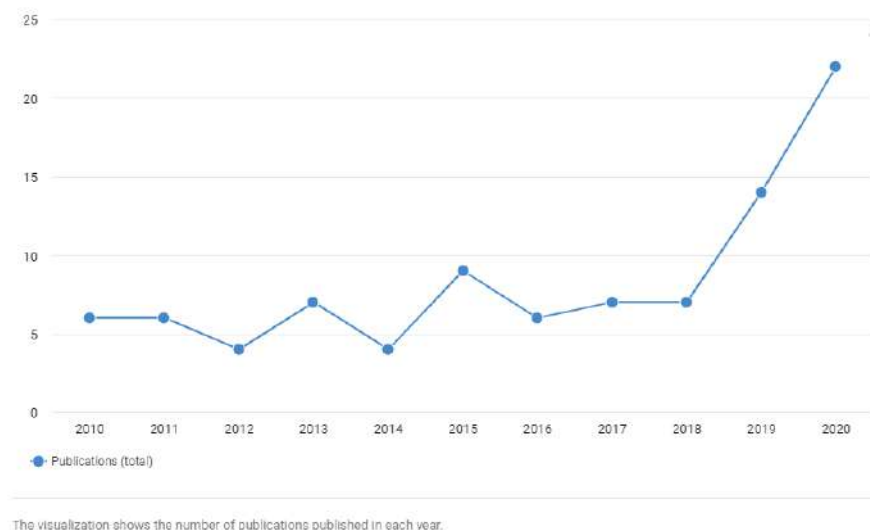
Figura 2 – Número de publicações por ano com os termos Quantum Chemistry (painel superior), Quantum Computational Chemistry (painel intermediário) e Quantum Algorithms for Chemistry (painel inferior)



Fonte: (DIMENSIONS, 2018).

é drasticamente menor, conforme figura 3. No entanto, as aparições do termo “Quantum computing for periodic systems” seguem também um comportamento claramente crescente na última década.

Figura 3 – Número de publicações por ano com o termo *Quantum computing for periodic systems*



Fonte: (DIMENSIONS, 2018).

Podemos notar que no máximo duas dúzias de trabalhos tratavam de computação quântica aplicada a materiais periódicos em 2020. Neste gráfico filtramos para que apenas trabalhos que relacionavam a expressão-chave em seu título ou resumo fossem retornados. Sem esse filtro, os trabalhos que encontrávamos tratavam do termo em algum local em seu texto, mas apenas mencionavam essa aplicação como potencial da computação quântica ou como uma área a ser explorada. Dada a escassez de trabalhos acerca da computação quântica aplicada a materiais periódicos, nossa decisão inicial foi visitar a literatura acerca do que é feito utilizando computação quântica aplicada a sistemas moleculares, para buscar entender os métodos empregados e de que forma estes métodos tem sido utilizados também para sistemas periódicos.

Pudemos observar que a simulação de materiais pode ser dividida em dois tipos de problemas, estáticos e dinâmicos. Os trabalhos direcionados para moléculas utilizam em sua grande maioria uma simulação estática para obtenção das propriedades do material (BASSMAN et al., 2021) pois estão interessados principalmente na estrutura eletrônica da molécula (MCARDLE et al., 2020), focando seus esforços na determinação de propriedades diretas do sistema, como o estado de menor energia e estados excitados. Já as simulações dinâmicas trabalham com a evolução temporal da função de onda e a mudança de propriedades que são dependentes do tempo, como, para a química quântica por exemplo, se dois ou mais sistemas irão reagir um com o outro ao receberem determinada energia (MCARDLE et al., 2020).

Quando chegamos nos trabalhos referentes à física de materiais periódicos percebemos essa mesma divisão entre simulações estáticas e dinâmicas, com trabalhos que apresentam propostas para ambos os tipos (CERASOLI et al., 2020; KIVLICHAN et al., 2018; MA; GOVONI; GALLI, 2020; MEI et al., 2020; MOTTA et al., 2021). A grande promessa de vantagem ao se usar a computação quântica reside nas simulações dinâmicas, pois tratam exatamente de propriedades dependentes do tempo e que, para a computação clássica, são muito custosas de se realizar.

Infelizmente, o que temos percebido em nossa revisão da literatura é que ainda é cedo para uma máquina quântica operar algoritmos profundos que demandam um alto número de qubits para efetuar sua simulação com precisão. As máquinas NISQ disponíveis atualmente não tem mais que algumas dezenas de qubits (PRESKILL, 2018). No entanto, simulações estáticas podem ser muito proveitosas para a ciência de materiais, que está sempre interessada na estrutura de bandas e outras propriedades como as fases topológicas de diversos sistemas periódicos (BATRA; SHEET, 2020; CERASOLI et al., 2020; MURTA; CATARINA; FERNÁNDEZ-ROSSIER, 2020). Estas simulações já são cabíveis de serem efetuadas nas máquinas NISQ, seus algoritmos são mais rasos que aqueles para a função de onda dependente do tempo e necessitam de um tempo de execução menor.

Nas seções a seguir introduziremos conceitos básicos quanto a computação e algoritmos quânticos, que trazem a base para um entendimento tanto da literatura especializada em computação quântica e das limitações da máquinas NISQ apontadas, quanto dos nossos resultados para a revisão e simulações de materiais.

2.2 Sólidos cristalinos periódicos

Em um sólido cristalino, os átomos estão organizados de tal forma que a mínima energia só é obtida quando se formam “blocos de construção”, ou seja, organizações periódicas simétricas que se repetem para gerar o sólido como o conhecemos. Essa periodicidade de repetição também dá ao material propriedades típicas dessa formação, além de facilitar o entendimento teórico de sua composição e permitir artifícios matemáticos para estudarmos essas propriedades (IBACH; LÜTH, 2010).

Os blocos que se repetem periodicamente em tais estruturas cristalinas são as chamadas células unitárias. As células unitárias são composições que pertencem a grupos de simetria definidos por dois ou mais vetores unitários que partem da mesma origem e se repetem para gerar a estrutura. As características dessa célula unitária dependerão dos vetores $\mathbf{a}$ e $\mathbf{b}$ (e $\mathbf{c}$, para o caso de sólidos tridimensionais) assim como dos ângulos γ_i formados entre eles. Para o caso bidimensional, teremos 5 possibilidades de combinações para as estruturas, como pode ser visto na figura 4, já para o caso 3D teremos 7 possibilidades de combinações cristalinas, que podemos ver na tabela 1 que levarão a

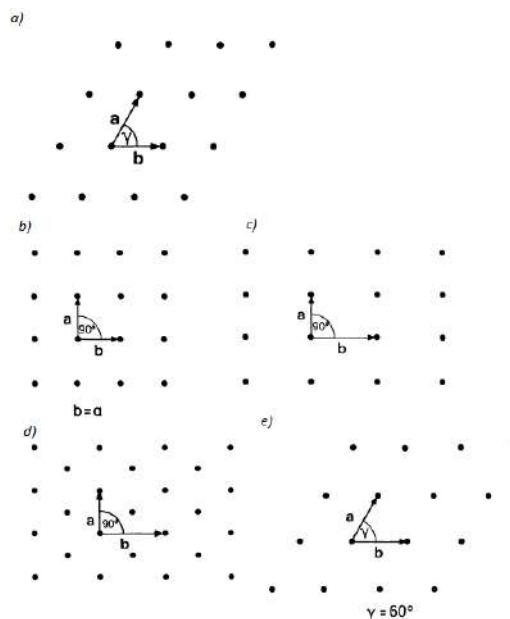
Tabela 1 – As 7 possibilidades de combinações entre vetores de base e ângulos, formando os 7 sistemas cristalinos distintos da cristalografia

Vetores da Base	Ângulos	Tipo de cristal
$a \neq b \neq c$	$\alpha \neq \beta \neq \gamma \neq 90^\circ$	Triclínico
$a \neq b \neq c$	$\alpha = \beta = 90^\circ, \gamma \neq 90^\circ$	Monoclínico
$a \neq b \neq c$	$\alpha = \beta = \gamma = 90^\circ$	Ortorrômbico
$a = b \neq c$	$\alpha = \beta = \gamma = 90^\circ$	Tetragonal
$a = b \neq c$	$\alpha = \beta = 90^\circ, \gamma = 120^\circ$	Hexagonal
$a = b = c$	$\alpha = \beta = \gamma \neq 90^\circ$	Romboédrico
$a = b = c$	$\alpha = \beta = \gamma = 90^\circ$	Cúbico

Fonte: (IBACH; LÜTH, 2010, p. 23).

14 tipos de estruturas tridimensionais como na figura 5, conhecidas como estruturas de Bravais (KITTEL; MCEUEN; MCEUEN, 1996).

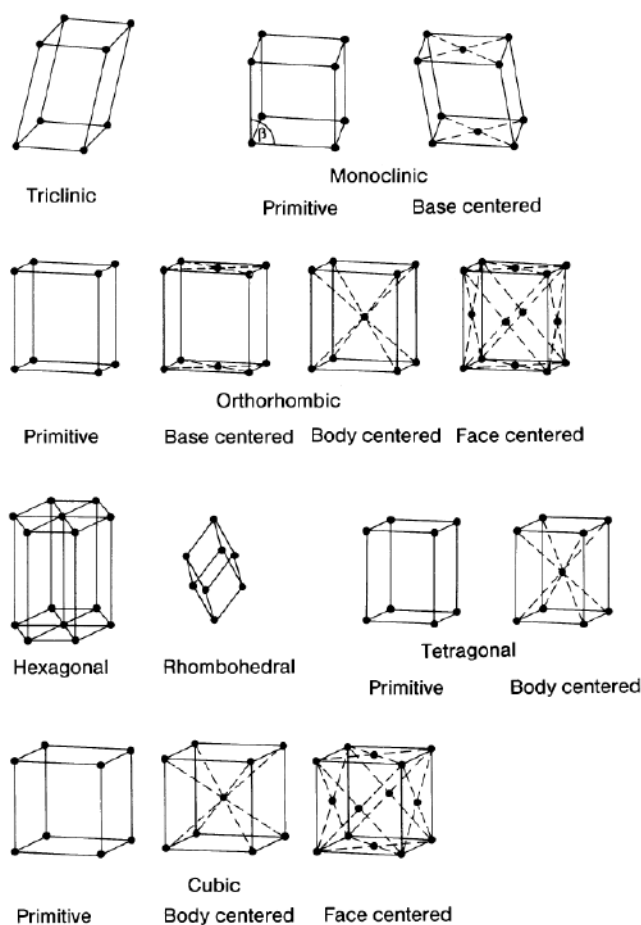
Figura 4 – Possíveis estruturas para uma combinação 2D. Em a) temos uma estrutura plana oblíqua, em b) quadrada, em c) retangular, em d) uma retangular centrada em e) uma hexagonal onde $a=b$ e $\gamma=60^\circ$



Fonte: (IBACH; LÜTH, 2010, p. 23).

Diferentemente dos líquidos, o estado sólido de um material é uma configuração rígida de seus átomos, onde estes no máximo vibram ao redor do seu ponto de equilíbrio na rede. São exemplos de materiais comumente usados em nosso cotidiano e que fazem parte de materiais cristalinos periódicos, metais, semicondutores, uma certa variedade de cerâmicas e alguns tipos de polímeros. Suas aplicações variam em larga escala, como na indústria farmacêutica (GOHANE et al., 2015; MAZUREK; SZELESZCZUK; PISKLAK, 2020), células solares (EL CHAAR; LAMONT; EL ZEIN, 2011; MICHELLE et al., 2001) e dispositivos semicondutores (CASADY; JOHNSON, 1998; KRAUSS; RUE, 1999; WANG

Figura 5 – As 14 formações tridimensionais possíveis



Fonte: (IBACH; LÜTH, 2010, p. 24).

et al., 2018), biomedicina (DENRY; HOLLOWAY, 2010; DOROZHKIN, 2018), entre outros (PISANI, 2012).

Uma estrutura cristalina tem uma energia menor do que um arranjo de átomos livres, porém suas propriedades como sólido ainda dependem do comportamento desses átomos na estrutura. As propriedades podem ser divididas naquelas que dependem da dinâmica atômica ao redor do seu ponto de equilíbrio e naquelas que dependem do grau de liberdade dos elétrons na cadeia cristalina. Porém, ambas acabam sendo complementares pois quando os átomos se movem de sua posição de equilíbrio os elétrons também adotam uma nova distribuição, com nível de energia maior do que o estado de equilíbrio. Quando o átomo retorna a posição de origem a energia gasta anteriormente é recuperada através do nível elevado em que os elétrons se situavam, de tal forma que a energia total é conservada. Sendo a energia função das coordenadas dos átomos, também será o potencial eletrônico na cadeia cristalina, e portanto das propriedades do determinado material. Além disso, o comportamento eletrônico também depende da estrutura na qual o elétron se encontra dentre àquelas disponíveis, como na figura 5. Mas, para que muitas dessas propriedades façam sentido, devemos notar que os níveis de energia que essas partículas ocupam não

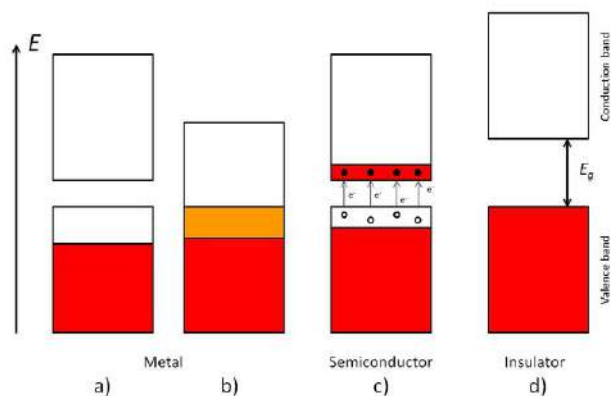
podem ter infinitas possibilidades, e sim níveis discretos e bem definidos chamados de bandas de energia.

As bandas surgem da interação entre as funções de onda dos elétrons na estrutura. No caso de um sistema simples, formado por dois átomos que estão muito próximos, as densidades de probabilidade eletrônica sofrem interferência e as funções, que eram degeneradas, perdem a degenerescência dando origem a níveis energéticos próximos. Pelo princípio de Pauli, a ocupação desses níveis deve se dar em uma configuração que leve à minimização na energia, um elétron por estado, formando um estado de partículas indistinguíveis estável. Esses átomos que tiveram suas funções de onda eletrônicas superpostas são também chamados de molécula diatômica e vão dividir suas energias criando novos estados possíveis para um mesmo orbital (KITTEL; MCEUEN; MCEUEN, 1996).

Quando temos muitos átomos em um sólido periódico, haverá muitas sobreposições entre as densidades dos orbitais eletrônicos, gerando uma banda de estados. A ocupação desses estados é o que definirá o nível de condutividade do sólido, isto é, se o classificamos como metal, isolante ou semicondutor. Um metal, por exemplo, possui seus orbitais mais energéticos apenas parcialmente preenchidos de modo que os elétrons podem transitar pela rede cristalina com baixa resistência (figura 6(a-b)). Dizemos que sua banda de condução é vaga e que a mobilidade dos elétrons nessa banda define uma condutividade elétrica alta para o material. Já em um isolante, por exemplo, suas bandas mais energéticas são completamente preenchidas e existe uma separação em energia entre elas e a próxima banda vazia (figura 6(d)). A banda totalmente preenchida recebe o nome de banda de valência enquanto que a banda vazia recebe o nome de banda de condução. Caso a separação entre a banda de valência e a banda de condução seja muito grande, o movimento de elétrons dentro do material é bastante comprometido, ou seja, a mobilidade eletrônica é pequena. A condutividade eletrônica é proporcional à mobilidade, portanto, os isolantes não conduzem bem eletricidade - a menos que fatores externos permitam a excitação de elétrons da banda de valência para a banda de condução. Nos semicondutores, o preenchimento das bandas é similar aos isolantes, porém a separação de energia entre as bandas não é tão grande como em um isolante (figura 6(c)) (KITTEL; MCEUEN; MCEUEN, 1996). Nesse caso, fatores externos como a temperatura são suficientes para excitar elétrons para a banda de condução, aumentando a condutividade elétrica do material.

Para que um elétron transite de uma banda para outra, é necessário fornecer uma quantidade de energia que seja superior à separação de energia entre as bandas. Isso trouxe uma ferramenta experimental importante para a determinação das estruturas de bandas, a espectroscopia. A espectroscopia Raman, por exemplo, estuda o espalhamento inelástico da luz ao excitar o átomo para certos níveis de interesse através da conservação

Figura 6 – Diagrama de bandas para diferentes materiais. A extensão vertical indica as regiões com energias permitidas, as áreas em vermelho indicam ocupação de elétrons. No metal, temos a banda de condução vazia enquanto que no semiconductor essa banda está apenas parcialmente preenchida dada a excitação de elétrons da banda de valência para a banda de condução



Fonte: (NYLANDER, 2015).

de energia. Isso ocorre através da polarizabilidade de elétrons de valência (IBACH; LÜTH, 2010). Os estudos da função dielétrica pela espectroscopia ótica são muito úteis para a determinação da estrutura de banda completa do cristal (KITTEL; MCEUEN; MCEUEN, 1996), além de ser um estudo fundamental para áreas de pesquisa como a de energia solar (MICHELLE et al., 2001). Porém, um entendimento teórico aprofundado da estrutura eletrônica de um sólido cristalino exige solucionar a equação de Schrödinger para todos os elétrons e núcleos da rede cristalina, o que exige um ferramental teórico sofisticado, conforme discutiremos na seção a seguir.

2.2.1 O avanço computacional e o desenvolvimento de materiais

Sabemos que em um sólido o número de átomos e elétrons é exorbitante, que escala com a constante de Avogadro. A constante de Avogadro forma o vínculo entre a escala atômica e a escala microscópica, representando a igualdade em número de entidades por molécula de qualquer substância (BECKER et al., 2003), sendo este igual a $6.022 \times 10^{23} \text{ mol}^{-1}$. Com a grande dificuldade em se resolver a equação de Schrödinger para todos os elétrons e núcleos no sólido cristalino, há uma grande demanda de métodos que acelerem essas resoluções principalmente na indústria de materiais, onde há uma grande busca por materiais com propriedades específicas, mais baratos ou que supram qualquer necessidade própria, seja de empresas ou novas tecnologias que estão sendo desenvolvidas (D'ALESSANDRO; SMIT; LONG, 2010; HENTALL et al., 1999).

Nos últimos anos, métodos computacionais de primeiros princípios ou *ab initio* se tornaram a ferramenta mais utilizada em ciências de materiais (HAUTIER; JAIN; ONG, 2012), sendo o mais popular entre eles a teoria do funcional de densidade (DFT, do inglês,

Density Functional Theory). O método DFT conta com uma ampla implementação nas áreas teóricas de física, química e também biologia, devido ao fato de ser mais econômico, aplicar uma modelagem computacional para analisar e classificar um determinado material, antes de sua implementação real em um dispositivo ou uma experimentação em laboratório. Teoricamente, podemos prever informações úteis e entender melhor as características fundamentais de um determinado material e seu possível uso em dispositivos com mais facilidade.

2.2.1.1 A teoria do funcional de densidade

Uma ferramenta muito versátil para pesquisas em mecânica quântica e que nos possibilita investigar átomos, moléculas e materiais cristalinos é a teoria do funcional de densidade (DFT) (DOBSON; VIGNALE; DAS, 1996). Essa versatilidade se dá principalmente pela redução de complexidade do problema ao utilizarmos a DFT, fazendo diversas aproximações que diminuem o custo computacional da simulação. Porém, essas aproximações também significam limitações de aplicabilidade e precisão dos resultados, principalmente para sistemas com alto nível de correlação ou com excessivo tamanho (número de partículas necessárias para um cálculo preciso).

As aproximações básicas mais utilizadas se tratam da aproximação de Born-Oppenheimer (BORN; OPPENHEIMER, 1927) e Hartree-Fock (SLATER, 1951). Na aproximação de Born-Oppenheimer buscamos separar a função de onda que descreve o átomo como um todo em duas funções separadas, uma função de onda eletrônica e uma função de onda para os núcleos atômicos. A primeira aproximação que se é feita é tratar os núcleos como estáticos, para isso admitimos que a massa nucleica é duas mil vezes maior que a massa da nuvem eletrônica. Além disso, o núcleo e os elétrons sofrem uma força de atração mútua devido a oposição de suas cargas, o que significa que ambos estão sendo acelerados e essa aceleração, como sabemos, é inversamente proporcional a massa da partícula. Sendo assim, aceleração do elétron é muito maior do que a aceleração do núcleo fazendo com que ele se movimente e responda à ação de forças muito mais rapidamente do que o núcleo, de modo que o movimento do núcleo pode ser considerado desprezível e sua função de onda passa a depender apenas de sua posição.

No método de Hartree-Fock o que buscamos é otimizar a forma espacial dos orbitais de spin, de forma a minimizar a energia da função de onda (MCARDLE et al., 2020). Tipicamente, devemos considerar um número “m” de orbitais de spin maior do que um número “n” de elétrons presentes no sistema. Também devemos assumir que “n” dos orbitais de spin estão ocupados tal que (m-n) são orbitais livres. Devemos então fazer duas aproximações iniciais, a primeira delas desprezando a força de repulsão entre os elétrons reduzindo a complexidade do problema para um conjunto de orbitais atômicos de elétrons-únicos. A segunda se trata de considerarmos o movimento dos elétrons somente

na distribuição de carga média dos demais elétrons, de modo que possamos admitir um potencial efetivo e aplicar uma solução iterativa para os “ n ” elétrons acoplados. Primeiro calcula-se a posição de cada elétron e então o potencial dada a configuração obtida, o processo então é repetido até que os orbitais convirjam. Esse método é bastante eficaz mas demonstra certa fraqueza quando trata de sistemas fortemente correlacionados, pois a força de Coulomb não pode ser desprezada.

A ideia principal da DFT se dá ao assumirmos que a energia do estado fundamental é função exata da densidade de probabilidade eletrônica, de modo que não precisamos utilizar necessariamente a função de onda para descrever e adquirir informações de um determinado sistema, trazendo uma grande simplificação para a simulação. Dada a antissimetria das funções de onda de sistemas fermiônicos, a função de onda eletrônica total é dada pelo determinante de Slater e escala com $N!$, onde N é o número de elétrons no sistema, tornando a solução proibitiva para sistemas com mais que uma dezena de elétrons. O uso da densidade eletrônica acaba com esse problema, pois trata de um potencial médio e o problema de muitos corpos é reduzido ao problema de elétrons não interagentes sujeitos a um potencial médio definido pela densidade eletrônica. A densidade de probabilidade que descreve o sistema é obtida pela minimização do funcional da energia em um processo iterativo que determina a energia do estado fundamental do sistema, segundo o Teorema de Khom-Sham (PARR, 1980). Assim, sendo determinado o estado fundamental função da probabilidade de densidade, diversas outras propriedades poderão ser inferidas através da mesma. Ao trabalharmos com densidade de probabilidade, podemos obter a energia do estado fundamental através do processo iterativo, como na aproximação de Hartree-Fock, em que cada passo da iteração há uma minimização de energia.

Como podemos notar, a obtenção das propriedades eletrônicas via DFT requer um ferramental teórico e matemático sofisticado, cujo principal objetivo é reduzir a complexidade da solução do problema de muitos corpos. Seu sucesso está associado, principalmente, ao fato de necessitar de uma quantidade pequena de informações *a priori* sobre o sistema sob análise, tais quais a organização cristalina. Praticamente nenhuma parametrização experimental é necessária. Contudo, é um modelo que somente se desenvolveu nas últimas duas décadas, dado o aumento maciço no poder computacional. Buscando um custo computacional menor, outros modelos se desenvolveram. Tais modelos envolvem o uso de parâmetros experimentais, como é o caso do modelo Tight-Binding. A “simplicidade” de modelos como o Tight-Binding pode ser usada como ponto de partida para a implementação de soluções usando computadores quânticos, enquanto o mapeamento do Hamiltoniano completo nos qubits ainda é impraticável.

2.2.1.2 Modelo Tight-Binding

Paralelamente ao DFT, uma aproximação de grande utilidade para a simulação de materiais é o modelo Tight-Binding. Nele, buscamos descrever a estrutura de bandas em termos de uma superposição linear de auto-funções atômicas, considerando os elétrons fortemente ligados aos núcleos (IBACH; LÜTH, 2010). Como discutimos anteriormente, quando átomos estão muito próximos suas funções de onda começam a se sobrepor e com a quebra de degenerescência dos estados surgem as bandas de energia. Como em uma estrutura cristalina temos uma quantidade enorme de átomos, essas sobreposições permitem que o elétron se desloque na estrutura sendo compartilhados entre os átomos. Essa troca de elétrons entre os átomos é dada por um fator de *hooping*, determinado por constantes no Hamiltoniano que descreve o sistema, entre estados em diferentes sítios no cristal. Nesse sentido, temos um modelo em que os elétrons não interagem entre si, somente com os núcleos que os contém e o acoplamento entre as funções de onda eletrônicas é traduzido por *hoopings* entre átomos vizinhos. As constantes de *hooping* podem ser obtidas pelo ajuste da estrutura eletrônica aos resultados experimentais.

2.2.1.3 Modelo Tight-Binding aplicado à computação quântica

Como buscamos usar computadores quânticos para determinar as propriedades de sistemas cristalinos periódicos, o Hamiltoniano simplificado de estrutura pode ser mapeado usando o método Tight-Binding. Em uma aproximação inicial, esse tipo de Hamiltoniano se mostrou capaz de elucidar os componentes básicos da estrutura eletrônica de sistemas cristalinos. Primeiro, escrevemos o Hamiltoniano Tight-Binding geral em segunda quantização como (LE et al., 2020)

$$\hat{H} = \sum_{i,n} \epsilon_n \hat{c}_{i,n}^\dagger \hat{c}_{i,n} - \sum_{\langle i,j \rangle, n, m} t_{in,jm} \hat{c}_{i,n}^\dagger \hat{c}_{j,m}, \quad (1)$$

onde o primeiro somatório, associado à energia potencial do sistema, se dá em termos da interação na i -ésima célula unitária do material para o orbital n . O operador $\hat{c}_{i,n}$ aniquila um elétron no orbital n -ésimo da célula i -ésima, e o operador $\hat{c}_{j,m}^\dagger$ cria um elétron no orbital m -ésimo da célula j -ésima. O produto $\hat{c}_{i,n}^\dagger \hat{c}_{i,n}$ é o operador número que determina a ocupação do orbital. O segundo somatório, associado à energia cinética do sistema, contabiliza o salto (*hooping*) entre as células unitárias e orbitais adjacentes. Na aproximação que estamos utilizando, esse termo será avaliado apenas entre os vizinhos próximos $\langle i, j \rangle$. ϵ_n é a energia do orbital n -ésimo, e $t_{in,jm}$ é a probabilidade de salto entre as camadas e orbitais.

Como sabemos, o operador de criação atuando sobre o estado de vácuo cria um elétron nesse estado

$$c_i^\dagger |v_{ac}\rangle = |i\rangle, \quad (2)$$

onde os símbolos de chapéus sobre os operadores foram omitidos para encurtar a notação. O i -ésimo operador de criação atua no estado de vácuo $|v_{ac}\rangle$ e cria um elétron na camada i -ésima. Seja $\mathbf{c}^\dagger$ o vetor criação formado por todos os operadores de criação organizados preservando a ordem das células unitárias na estrutura,

$$\mathbf{c}^\dagger = \begin{bmatrix} c_1^\dagger \\ c_2^\dagger \\ c_3^\dagger \\ \vdots \\ c_n^\dagger \end{bmatrix}, \quad (3)$$

o vetor de aniquilação será seu adjunto $\mathbf{c} = (\mathbf{c}^\dagger)^\dagger$, logo

$$\mathbf{c} = [c_1 \ c_2 \ c_3 \ \dots \ c_n]. \quad (4)$$

O Hamiltoniano Tight-Binding pode ser escrito matricialmente como

$$\mathbf{H} = \mathbf{c}^\dagger \underline{\mathbf{T}} \mathbf{c}, \quad (5)$$

onde $\underline{\mathbf{T}}$ é a matriz que descreve a energia das camadas e a probabilidade de salto entre elas. Então

$$\underline{\mathbf{T}} = \begin{bmatrix} \varepsilon_1 & t_{12} & t_{13} & \dots & t_{1n} \\ t_{21} & \varepsilon_2 & t_{23} & & \\ t_{31} & t_{32} & \varepsilon_3 & & \\ \vdots & & & \ddots & \\ t_{n1} & & & & \varepsilon_n \end{bmatrix}. \quad (6)$$

Manipulando a equação 5, podemos recuperar corretamente o somatório com o Hamiltoniano, onde ε_j é o auto-valor j -ésimo da camada, e t_{ij} o salto entre as camadas i e j .

Para diagonalizar o Hamiltoniano é possível atuarmos uma transformação fermiônica que preserva o anti-comutador entre os operadores de aniquilação e criação

$$\mathbf{c}^\dagger = \mathbf{f}^\dagger \underline{\mathbf{U}}^\dagger, \quad (7)$$

onde $\underline{\mathbf{U}}$ é a matriz unitária cuja transposta descreve a transformação entre os sub-espacos de Fock. $\mathbf{f}^\dagger$ também é um vetor coluna formado pelos operadores de criação $f^\dagger$. Então, reescrevendo o Hamiltoniano, temos

$$\mathbf{H} = \mathbf{f}^\dagger [\underline{\mathbf{U}}^\dagger \underline{\mathbf{T}} \underline{\mathbf{U}}] \mathbf{f}. \quad (8)$$

Supondo que $\underline{U}$ é tal que $\underline{D} = \underline{U}^\dagger \underline{T} \underline{U}$ é uma matriz diagonal com elementos $[\underline{D}]_{k,k'} = \delta_{k,k'} \varepsilon_k$, então o Hamiltoniano fica

$$\mathbf{H} = \sum_k \varepsilon_k \mathbf{f}_k^\dagger \mathbf{f}_k, \quad (9)$$

onde $\mathbf{f}_k^\dagger \mathbf{f}_k$ é o operador número que é conservado pela transformação canônica. Para sistemas periódicos, a transformação canônica que pode ser utilizada para a diagonalização do Hamiltoniano é a própria transformada de Fourier, o que demonstraremos mais adiante.

2.3 Computação Quântica

Diferentemente das máquinas clássicas, as quânticas usam qubits ao invés de bits tradicionais e são esses qubits que trazem o potencial para uma revolução computacional. Segundo Preskill ([PRESKILL, 2018](#)) os computadores quânticos tolerantes a erros estão longe de serem construídos e ele mesmo introduz um termo para falar das máquinas atuais, *NISQ - Noisy Intermediate Scale Quantum*. Essas máquinas NISQ disponíveis atualmente já são um grande avanço para a computação quântica, pois estão chegando a marca de 50 qubits ou mais. Com esse número de qubits em seus processadores, já ultrapassam o que é possível de ser simulado utilizando supercomputadores disponíveis atualmente ([LUND; BREMNER; RALPH, 2017](#); [PRESKILL, 2018](#)). Mas como o nome diz, são máquinas com muito ruído e de média escala o que relativamente limita o tipo e tamanho de tarefas a serem realizadas nelas.

Um qubit pode ser construído com diversos sistemas quânticos, um átomo, um elétron, um fóton ou mesmo um circuito elétrico. Falaremos de alguns deles aqui. A representação de um qubit se dá no espaço complexo de Hilbert, com uma base mutuamente ortogonal onde comumente seus estados possíveis são representados por $|0\rangle$ e $|1\rangle$,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (10)$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (11)$$

Essa representação é similar àquela da computação clássica onde os bits podem ser apenas 0 ou 1, e a representação $|0\rangle$ e $|1\rangle$ também é chamada de base computacional. Porém, é possível encontrar os qubits em estados que não sejam $|0\rangle$ e $|1\rangle$ e sim uma combinação linear de estados, chamada superposição ([NIELSEN; CHUANG, 2010](#))

$$\begin{aligned}
|\psi\rangle &= \alpha |0\rangle + \beta |1\rangle, \\
\alpha, \beta &\in \mathbb{C}, \\
|\alpha|^2 + |\beta|^2 &= 1.
\end{aligned} \tag{12}$$

Como podemos aferir através dessa definição, ao buscarmos a informação de um qubit podemos encontrar $|0\rangle$ com probabilidade $|\alpha|^2$ ou $|1\rangle$ com probabilidade $|\beta|^2$. Por mais que o estado do qubit seja uma superposição de estados enquanto algum algoritmo é executado, os postulados da mecânica quântica nos permitem apenas medi-lo na base computacional para sistemas computacionais empregados no mundo real, e por isso sempre o encontraremos em um desses estados. Com sua capacidade de efetuar emaranhamento quântico e superposição de estados, novas possibilidades se abrem para a simulação de materiais e moléculas além de diversas outras áreas.

Nas máquinas quânticas “NISQ”, *Noisy Intermediate-Scale Quantum* ([PRES-KILL, 2018](#)), os qubits são suscetíveis a ruídos e erros em suas execuções. Esses erros de forma geral ocorrem por interação dos qubits com o ambiente e por não sermos capazes de controlá-los de forma perfeita ([MCARDLE et al., 2020](#)). Também ocorrem erros em algoritmos muito extensos, com uma quantidade relativamente grande de portões lógicos, de modo que o tempo de execução do algoritmo supera o tempo de coerência do qubit, isto é, o tempo pelo qual ele é capaz de manter seu estado quântico sem colapsar. Assim como no início da computação clássica, onde os computadores eram institucionais, os computadores quânticos ainda estão longe de serem de uso pessoal e talvez nunca o sejam, tendo em vista as dificuldades que existem para operar e manter as máquinas em funcionamento. Estas dificuldades vão desde dificuldades impostas pelos fundamentos da mecânica quântica, até quanto problemas de engenharia para construção dessas máquinas.

Para adquirirmos uma informação de um sistema quântico durante a medida geramos uma perturbação que colapsa a função de onda, como é imposto pelos postulados da mecânica quântica. Para realizar uma medição devemos interagir com o sistema de interesse e interações geram perturbações no sistema e, se queremos que o sistema opere em perfeitas condições, devemos mantê-lo isolado do ambiente externo devido a alta sensibilidade dos qubits à flutuações térmicas e magnéticas. Além disso, também queremos poder controlar a máquina pelo lado de fora enquanto os qubits interagem fortemente uns com os outros, sem que percam a coerência de estado, o que por si só já é um grande problema. Conectar controles externos e manter uma baixa temperatura para os qubits apresenta muitas restrições em termos de área ocupada e consumo de energia ([GECK et al., 2020](#)), tanto para manter o isolamento dos qubits como para os circuitos de controle operarem corretamente, já havendo algumas pesquisas sobre este tema ([PATRA et al., 2018](#); [VAN DIJK et al., 2019](#)). Com certeza se trata de uma tarefa extremamente

desafiadora(PRESKILL, 2018).

Para se construir um processador quântico são necessárias tecnologias para gerar bits quânticos, qubits. Também é importante salientar o que é chamado de “O Critério DiVincenz”, o qual estabelece as condições necessárias para que um qubit seja considerado otimizado para uso (DIVINCENZO, 2000). São elas:

- Deve ser possível inicializar o qubit em um estado conhecido.
- Deve ser possível aplicar operadores com fidelidade para operações de um ou 2-qubits.
- Deve ser possível ler o estado do qubit, tipicamente de forma individual.
- O tempo de coerência dos qubits deve ser longo o bastante para que as operações com 1 e 2-qubits possam ser realizadas em grande número.
- Deve haver interfaces para interação qubit-fóton, garantindo transferências por longa distância de informações quânticas e emaranhamentos.

O tempo de coerência geralmente é dividido em duas categorias, o tempo de defasagem e o tempo de relaxamento. O relaxamento de fase ocorre por interações de flutuação térmica e magnética com o ambiente, que perturba o circuito quântico (aqui, o circuito físico) e, após certo tempo, a fase de polarização pode se tornar randômica e o estado controlado dos qubits, perdido. Já o tempo de relaxamento está relacionado com a amplitude e a polarização do sistema, que também podem ser alteradas pelas mesmas perturbações anteriores ou por flutuações de carga dentro do circuito (HU; DE SOUZA; SARMA, 2002). Podemos entender melhor do que se trata o relaxamento e a defasagem, olhando para o estado quântico na forma vetorial conforme a figura 7.

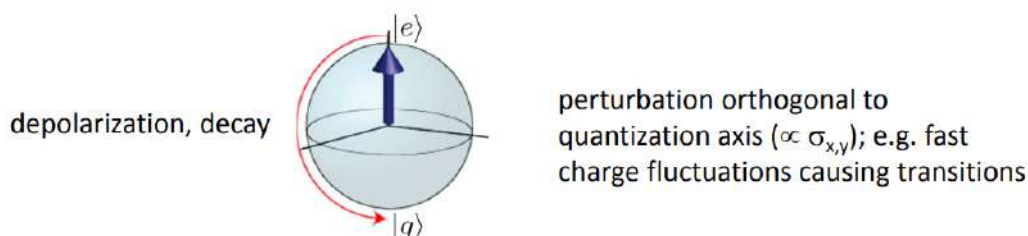
Na literatura encontraremos rotineiramente os dois tempos chamados de T1 e T2 assim como na figura 7 para encurtar e facilitar as terminologias (HOUCK et al., 2008; HU; DE SOUZA; SARMA, 2002), sendo T1 e T2 os tempos de relaxamento e defasagem, respectivamente.

Para se construir um qubit existem diferentes arquiteturas. Dentre as tecnologias conhecidas atualmente, as mais empregadas são os qubits supercondutores (CLARKE; WILHELM, 2008; WENDIN, 2017), qubits *trapped-ion* (SCHÄFER et al., 2018), qubits fotônicos (O'BRIEN, 2007), spins em *quantum-dots* (LOSS; DIVINCENZO; DIVINCENZO, 1997) e também em átomos de silício (KANE, 1998).

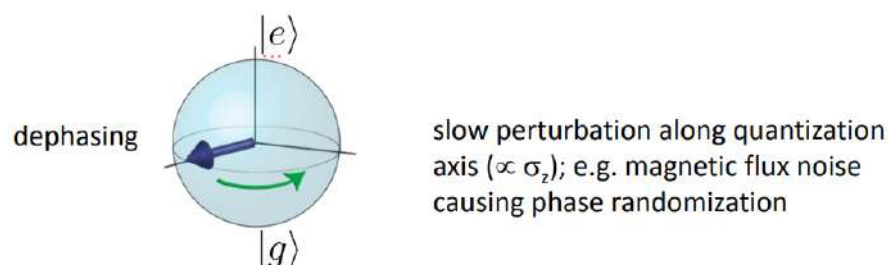
Os qubits supercondutores são utilizados principalmente pelas empresas IBM e Rigetti, que fornecem acesso via nuvem para suas máquinas ao público interessado. Estes qubits são fabricados com circuitos supercondutores em estado-sólido e mostram

Figura 7 – Relação dos tempos de relaxamento e defasagem. Podemos interpretar o tempo de relaxamento como perturbações orientados perpendicularmente ao plano XY. Quanto ao tempo de defasagem podemos interpreta-lo como perturbações ortogonais ao eixo Z

- T_1 : energy relaxation time



- T_2 : dephasing time



Fonte: (BIANCHETTI, 2010).

uma maior facilidade para serem controlados externamente por outros dispositivos eletrônicos (BASSMAN et al., 2021). Seus circuitos podem ser produzidos com capacitores, indutores e junções Josephson (SCHOELKOPF; GIRVIN, 2008). Essas junções trabalham com dois supercondutores dispostos entre um material não-supercondutor de modo que ocorra o efeito Josephson, nomeado em homenagem a Brian Josephson que previu, em 1962, que este arranjo permitiria elétrons tunelarem através do material não-supercondutor (JOSEPHSON, 1962). Diferentes valores de capacitância e indutância constroem os diferentes tipos de circuitos e qubits, como os qubits de fase (STEFFEN et al., 2006), fluxo (MOOIJ et al., 1999) e carga (NAKAMURA; PASHKIN; TSAI, 1999), qubits transmons (KOCH et al., 2007), qubits Xmons (BARENDTS et al., 2013) e gatemons (CASPARIS et al., 2016). As máquinas quânticas que usaremos pertencem à IBM e utilizam os qubits transmons supercondutores em sua arquitetura. Esse tipo de qubit apresenta grandes vantagens no que diz respeito ao controle e à leitura de dados, pois sua interação com os circuitos é forte, o que permite uma maior separação entre os qubits e os fios que conectam os sistemas, o que melhora a refrigeração do qubit (HOUCK et al., 2008).

Qubits *trapped-ion* se tratam de íons presos em um poço de potencial eletromagnético, cujo espectro de níveis eletrônicos determinam os estados para o qubit. As grandes empresas que fornecem acesso via nuvem à esses tipos de máquinas são a IonQ

e a Honeywell. Essa arquitetura de processador tem mostrado um tempo de coerência mais elevado do que as demais, sendo alcançados até mesmo 10 minutos (WANG et al., 2017). Porém, há uma significativa desvantagem quando se trata de velocidade de atuação dos operadores e de processamento por esses qubits, sendo de até duas ordens inferior à velocidade de processamento de um qubit supercondutor (BASSMAN et al., 2021). Isso devido ao fato de que as interações em qubits baseados em estado-sólido são muito mais fortes (HOUCK et al., 2008; SCHÄFER et al., 2018).

Já os qubits fotônicos são capazes de codificar a informação quântica em propriedades do fóton como sua polarização, sua direção de propagação, sua distribuição espacial (chamada de codificação de momento angular orbital) e também no tempo (SLUSARENKO; PRYDE, 2019). Essa arquitetura é geralmente empregada quando se precisa transferir informação quântica através de longas distâncias (BASSMAN et al., 2021), utilizando-se de deslocadores de fase, divisores de feixe e detectores de fótons.

2.4 Computação quântica de sistema moleculares

Neste capítulo é impossível não falarmos sobre computação quântica de sistemas moleculares, que se trata de um ramo de pesquisa com grande aplicabilidade para a computação quântica e que abriu nosso caminho para o estudo na física de materiais periódicos. Apresentaremos em seguida os resultados de revisão na área.

É fato que a computação quântica carrega grandes expectativas e promessas de superar os limites clássicos da computação nas mais diversificadas áreas e aplicações (MCARDLE et al., 2020; STENTA; DAL PERARO, 2011; SUGISAKI et al., 2021; XIA; KAIS, 2020). A computação quântica de sistema moleculares é sem dúvidas uma das aplicações atuais mais promissoras dessa computação, com vários métodos e algoritmos já sendo amplamente implementados por um grande número de pesquisadores (CAO et al., 2019). Mesmo com a limitação de hardware atual, onde os experimentos ficam limitados à pequenos sistemas que são possíveis de se simular classicamente, é especulado que, com o avanço das simulações quânticas, um dia elas nos deem resultados mais precisos do que os próprios experimentos reais (MCARDLE et al., 2020).

A computação quântica de sistemas moleculares se mostra superior à clássica em alguns aspectos, e aqui nomearemos dois dos principais: a representação de estados quânticos e a evolução temporal do estado. Este aprimoramento também é esperado para aplicações em pesquisas de física de sistemas em fase sólida periódica, pois é fato que os métodos utilizados para a computação de moléculas são primordiais para as pesquisas e aplicações para a simulação de sistemas sólidos. Isto é principalmente devido à maioria dos algoritmos utilizados hoje em sistemas sólidos periódicos terem surgido inicialmente para pesquisas para a área de moléculas (CAO et al., 2019).

Dentro da categoria de simulação de sistemas moleculares há uma diversidade de métodos, tanto para mapear o sistema de interesse de estudo em qubits, quanto para realizar o experimento desejado. Mapear o sistema que queremos simular se trata de representar o comportamento dos elétrons através do comportamento dos qubits e portões quânticos. Em geral, principalmente na computação quântica com qubits supercondutores, se trata de transformarmos a descrição fermiônica em uma descrição baseada em spins (MA; GOVONI; GALLI, 2020), que é a base utilizada por qubits reais em que serão aplicados os portões. Além do mais, esses portões representam operadores quânticos que atuam no estado de interesse e fazem parte do algoritmo específico à operação que se deseja realizar.

Pudemos verificar que os principais métodos para mapeamento são divididos em duas grandes classes: algoritmos que empregam primeira quantização e algoritmos que empregam segunda quantização (MCARDLE et al., 2020). Ambos algoritmos servem para mapear o Hamiltoniano que descreve o sistema nos qubits do computador quântico. Devido ao princípio de exclusão de Pauli, a função de onda deverá ser antissimétrica ao estudarmos os níveis de energia dos elétrons. Essa anti-simetria poderá ser contabilizada na própria função de onda (primeira quantização), ou nos operadores que atuam na função de onda (segunda quantização). Alguns dos principais métodos que encontramos foram:

- Primeira Quantização:
 - Métodos *grid-based* (LIDAR; WANG, 1999),
 - Métodos de conjuntos de base (ABRAMS; LLOYD, 1997).
- Segunda Quantização (também derivados de conjuntos de base):
 - Codificação de Jordan-Wigner (JORDAN; WIGNER, 1993),
 - Codificação de Bravyi-Kitaev (BRAVYI; KITAEV, 2002).

Nos métodos *grid-based* temos a função de onda representada na base das posições no espaço real, de modo que uma função de N-partículas pode ser expressa de forma discretizada em uma rede com M pontos por eixo, por

$$|\Psi\rangle = \sum_{x_1, \dots, x_N} \psi(x_1, \dots, x_N) A(|x_1, \dots, x_N\rangle), \quad (13)$$

onde temos $|x_i\rangle = |r_i\rangle |\sigma_i\rangle$ possuindo as coordenadas espaciais e de spin do i-ésimo elétron, com $\sigma \in \{0, 1\}$, $|r_i\rangle = (|x_i\rangle |y_i\rangle |z_i\rangle)$, $x_i, y_i, z_i \in \{0, 1, \dots, M-1\}$, $\forall i \in \{1, 2, \dots, N\}$. "A" representa uma função para contabilizar a antissimetria da função de onda. Este método se mostra bastante útil quando tratamos de sistema que não necessitam, ou não são apropriados de se tratar, com uma aproximação de Born-Oppenheimer (MCARDLE

et al., 2020). Contudo, não é um método apropriado para máquinas NISQ, pois teremos $2^{(3p+1)N}$ amplitudes complexas para a função de onda, onde p é um número arbitrário que determina a precisão da simulação, de modo que, assim como para a computação clássica, fica impossível armazenar e processar a quantidade de informações necessária para sistemas reais em um computador quântico atual.

Para o método de conjuntos de base, devemos projetar o Hamiltoniano em uma base de m funções de onda, $\{\phi_p(x_i)\}$, onde novamente x_i representa as coordenadas espacial e de spin do i -ésimo elétron, $x_i = (r_i, \sigma_i)$. A simetria de troca entre os elétrons deve ser contabilizada pela função de onda, tal que a função de onda para n elétrons pode ser representada como um produto anti-simétrico das funções de elétron único que compõe a base. De modo geral, podemos escrever tudo isso através do determinante de Slater

$$\Psi(x_0, \dots, x_{n-1}) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \phi_0(x_0) & \phi_0(x_1) & \dots & \phi_0(x_{m-1}) \\ \phi_1(x_0) & \phi_1(x_1) & \dots & \phi_1(x_{m-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n-1}(x_0) & \phi_{n-1}(x_1) & \dots & \phi_{n-1}(x_{m-1}) \end{vmatrix}, \quad (14)$$

de modo que alterar posições entre dois elétrons corresponde a trocar duas linhas da matriz, o que muda o sinal da função de onda como necessário pela simetria. O número m de orbitais considerados geralmente é maior do que o número n de elétrons, de modo que apenas n orbitais estarão efetivamente ocupados. O método *grid-based* discutido anteriormente armazena a função de onda de forma direta, sem se aproveitar de informações já conhecidas sobre suas propriedades. Já o método de conjuntos de base, ao projetar o Hamiltoniano nos spins orbitais $\{\phi_p(x_i)\}$, reduz os recursos necessários para armazenar a função de onda (MCARDLE et al., 2020).

A codificação de Jordan-Wigner (JW) trabalha mapeando os estados dos elétrons do espaço de Fock nos estados de qubits no espaço de Hilbert, da base fermionica para a base de spins. Nela, armazenamos o número de ocupação de um orbital ϕ_i nos estados $|0_i\rangle$ ou $|1_i\rangle$ dos qubits. Se considerarmos um número m de ocupações teremos:

$$|f_{m-1}, f_{m-2}, \dots, f_0\rangle \rightarrow |q_{m-1}, q_{m-2}, \dots, q_0\rangle, \quad q_i = f_i \in 0, 1. \quad (15)$$

Os operadores de criação e aniquilação da base fermiônica podem ser descritos na base dos qubits por

$$\begin{aligned} a_i &= b_i \otimes Z_{i-1} \otimes \cdots \otimes Z_0, \\ a_i^\dagger &= b_i^\dagger \otimes Z_{i-1} \otimes \cdots \otimes Z_0, \end{aligned} \quad (16)$$

com

$$\begin{aligned} b_i &= \frac{X_i + iY_i}{2}, \\ b_i^\dagger &= \frac{X_i - iY_i}{2}, \end{aligned} \quad (17)$$

onde X , Y e Z são operadores de Pauli, o operador $\mathbf{Z}$ atua recuperando a paridade do estado com um fator de fase $(-1)^{\sum_{i=0}^{i-1} f_i}$. Dessa forma, podemos representar um Hamiltoniano fermiônico através de uma combinação linear de um produto de matrizes de Pauli,

$$H = \sum_j h_j P_j = \sum_j h_j \bigotimes_i \sigma_i^j, \quad (18)$$

onde h_j é um coeficiente real, P_j é um produto de matrizes de Pauli (também chamado de Pauli *string*), σ_i são os operadores I, X, Y ou Z , i determina qual qubit estamos operando e j o termo no Hamiltoniano.

Na codificação de Bravyi-Kitaev (BK) as informações de ocupação de orbitais e paridade são armazenadas em forma parcialmente local (MCARDLE et al., 2020). Teremos então a definição

$$\begin{aligned} |f_{m-1}, f_{m-2}, \dots, f_0\rangle &\rightarrow |q_{m-1}, q_{m-2}, \dots, q_0\rangle, \\ q_i &= \left[\sum_{q=0}^i \beta_{iq} f_q \right], \end{aligned} \quad (19)$$

onde β_{iq} são elementos de uma matriz $\mathbf{BK}$ tal que

$$\begin{aligned} \beta_1 &= [1], \\ \beta_{2^x+1} &= \begin{pmatrix} \beta_{2^x} & \mathbf{0} \\ \mathbf{A} & \beta_{2^x} \end{pmatrix} \end{aligned} \quad (20)$$

onde $\mathbf{0}$ representa uma matriz $(2^x \times 2^x)$ de zeros e $\mathbf{A}$ também uma matriz $(2^x \times 2^x)$ porém com a última linha completa com 1's. Por exemplo, para $m = 2$ e $x = 1$ teríamos

$$\beta_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad (21)$$

e para $m = 4$ e $x = 1$ teríamos

$$\beta_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \quad (22)$$

se o número de orbitais, e por consequência o número de qubits, não for uma potência de 2 essa codificação se dá criando uma matriz BK para a maior potência de 2 seguinte ao número m , utilizando-se depois apenas as primeiras m linhas. Este mapeamento BK é, em geral, tão eficiente quanto o mapeamento JW com alguns casos onde o mapeamento BK torna os cálculos consideravelmente mais eficientes (TRANTER et al., 2018).

Também existem técnicas focadas em reduzir os custos de hardware para implementação das simulações levando em consideração a estrutura do sistema estudado (BRAVYI et al., 2017; MCCLEAN et al., 2020; MOTTA et al., 2021). Utilizando algum destes métodos de mapeamento podemos categorizar o sistema molecular na representação dos qubits. Para obtermos informações de um sistema de interesse precisamos então de algoritmos quânticos. Podemos dividi-los em dois tipos majoritariamente utilizados: *algoritmos de estimação (estimativa¹) de fase* e *algoritmos variacionais*.

2.5 Algoritmos para computadores quânticos

Devemos notar que há uma mudança de paradigma para a criação de algoritmos em relação a computação binária, certas coisas que são simples para a computação clássica são problemas de relativa complexidade para um computador quântico, como realizar operações aritméticas ou como copiar o estado de um qubit. Uma das maiores dificuldades já se dá no primeiro passo da simulação em um computador quântico, inicializar o estado interesse através dos qubits (GEORGESCU; ASHHAB; NORI, 2014). Após a inicialização do circuito quântico, é preciso o desenvolvimento de sequências de operações, dadas pela atuação de operadores quânticos, para preparar o estado numa configuração cuja medida possa retornar algum parâmetro de interesse. Essa sequência de operações recebe o nome de algoritmo, os quais são especialmente desenvolvidos para a solução de problemas envolvendo computadores quânticos. Esse é o caso, por exemplo, da troca de informação entre qubits que é feita através de protocolos de teletransporte, que envolvem estados de Bell (PIMENTA; BEZERRA, 2021).

Os computadores quânticos da atualidade são naturalmente suscetíveis a erros, seja por imprecisão do processo ou por interações indesejadas com o ambiente externo, o que traz limitações quanto a profundidade dos algoritmos implementados (CAO et al., 2019). Nesse sentido, desenvolver um algoritmo quântico requer uma certa atenção e otimização de etapas, assim como a implementação de um código para correção de erros em algoritmos de relativa larga-escala (FOWLER et al., 2012). Essa correção pode ser implementada através de um algoritmo secundário que trata os retornos da máquina quântica, ou através de hardware com qubits extras no processador que estabilizam um conjunto mais preciso de qubits implementados na execução do algoritmo (GOTTESMAN,

¹ Não temos uma tradução definida para o termo, neste trabalho usaremos “estimação” de fase.

2010).

2.6 Computação quântica aplicada ao estudo da matéria condensada

Após sumarizar os tópicos fundamentais de aplicação da computação quântica, principalmente voltada para sistemas moleculares, procedemos em filtrar melhor os resultados de busca afim de direcionarmos a pesquisa ao uso específico da computação quântica para sólidos periódicos. Como já mencionamos, as referências são mais escassas do que aquelas relacionadas à simulação de sistemas moleculares, porém os trabalhos de modo geral fazem uso dos mesmos tipos de algoritmos adaptados aos sistemas periódicos. É importante salientarmos que na química quântica é incomum ser feita uma simulação direta da dinâmica da função de onda de um determinado sistema, em vez disso, se busca encontrar o *ground-state* de uma sistema de interesse estimando seus auto-valores.

2.7 Algoritmos quântico-computacionais aplicados a matéria condensada

Utilizando do poderio computacional das máquinas NISQ, buscamos algoritmos que melhor se adéquem as limitações que essas máquinas possuem. Tal limitação, torna fundamental a utilização de algoritmos de baixa profundidade e tempo de operação, isto é, utilizar operadores que operam suficientemente mais rápido do que o tempo de coerências dos qubits envolvidos no circuito. Um dos maiores desafios para pesquisadores que se utilizam dos computadores quânticos, e algo que também já passou a ser uma área de pesquisa relevante, se trata de adequar algoritmos ideais às limitações de hardware atuais, além também de construir máquinas capazes de operar algoritmos mais caros. Como mencionamos anteriormente, vamos tratar aqui dos dois métodos mais encontrados na literatura e aplicados à simulação de moléculas e sistemas periódicos em computadores quânticos, os algoritmos de estimação de fase e os algoritmos variacionais.

O algoritmo de estimação de fase é o que atualmente mais atende as necessidades do estudo computacional de moléculas tendo em vista a grande quantidade de trabalhos que o citam ou utilizam (BAUER et al., 2020; CAO et al., 2019; MCARDLE et al., 2020; PAESANI et al., 2017; PARKER; JOSEPH, 2020; SUGISAKI et al., 2019, 2021). Contudo, ainda é inviável de ser utilizado para sistemas com algumas dezenas ou centenas de átomos, como sistemas cristalinos e grandes moléculas, visto o custo de hardware necessário para sua operação (BASSMAN et al., 2021). Já os algoritmos variacionais são menos custosos de serem implementados nos processadores quânticos. Trazem soluções para problemas

de otimização com uma espécie de algoritmo híbrido fazendo uso tanto da computação clássica quanto da quântica em sua execução. Com eles, podemos realizar a minimização de energia para um determinado Hamiltoniano e, por conseguinte, determinar com uma boa aproximação seu *ground-state* (MCCLEAN et al., 2016). Estes dois tipos de algoritmos bastante utilizados para a simulação de moléculas também serão os que detalharemos, mais profundamente, na próxima seção sobre a aplicação da computação quântica para a física de materiais.

2.8 Evolução temporal com computação quântica, o método de Trotterização

Resolver sistemas dependentes do tempo implica em resolver a equação de Schrödinger dependente do tempo, também conhecida como

$$i\hbar \frac{\delta}{\delta t} |\psi(t)\rangle = H |\psi(t)\rangle. \quad (23)$$

Para trabalharmos com a função de onda $\psi(t)$ podemos usar o operador de evolução temporal

$$U(t) = \exp\left(\frac{-i}{\hbar} \int_0^T H(t) dt\right), \quad (24)$$

que, no caso de um hamiltoniano independentes do tempo, se resume à $e^{-iHt/\hbar}$, mas que pode ser complexo de ser computado para o caso dependente do tempo. Então, se o Hamiltoniano é difícil de ser diagonalizado também há uma dificuldade na solução do operador de evolução temporal (BASSMAN et al., 2021). Para tanto, podemos superar essa dificuldade discretizando a evolução de $U(t)$ em intervalos adiabáticos Δt , de modo que o Hamiltoniano seja aproximadamente constante nesses intervalos. Seja o Hamiltoniano expresso pelo conjunto de operadores $\{H_j : j = 1, \dots, m\}$ de modo que

$$H = \sum_{j=1}^m H_j. \quad (25)$$

Podemos escrever U como

$$U(t, \mu) \approx e^{-iH_j N} e^{-iH_{jN-1}} \dots e^{-iH_{j1}} \\ U(t, \mu) \approx \prod_{i=1}^N e^{-iH_{j_i(t_i)} \Delta t_i} \quad (26)$$

onde μ é um número real que determina o tamanho do intervalo (SUZUKI, 1991). Podemos então pensar em duas situações comumente encontradas, uma delas onde queremos minimizar a quantidade de erro nessa aproximação tornando Δt o menor possível ou, por outro lado, minimizarmos a quantidade de operadores envolvidos no Hamiltoniano

aumentando o intervalo Δt , diminuindo assim o número de exponenciais envolvidas. Por exemplo, se conseguirmos dividir nosso Hamiltoniano em dois termos tal que $H = \hat{\sigma}_x + \hat{\sigma}_y$ teremos para o primeiro caso

$$e^{-i\Delta t(\hat{\sigma}_x + \hat{\sigma}_y)} = e^{-i\Delta t\hat{\sigma}_x} e^{-i\Delta t\hat{\sigma}_y} + \mathcal{O}(\Delta t^2) \quad (27)$$

que nos dá uma boa aproximação quando Δt é relativamente pequeno. Para o segundo caso onde Δt se torna grande, temos de fazer

$$e^{-i\Delta t(\hat{\sigma}_x + \hat{\sigma}_y)} = (e^{-i\Delta t\hat{\sigma}_x/N} e^{-i\Delta t\hat{\sigma}_y/N})^N + \mathcal{O}(\Delta t/N)^2. \quad (28)$$

Pode-se pensar que a relação 27 é trivial, porém é importante lembrar que para o caso onde $\hat{\sigma}_x + \hat{\sigma}_y$ não comutam, a relação $e^{A+B} = e^A e^B$ não é válida e para resolvermos a evolução do operador devemos utilizar a equação 28, também conhecida como fórmula Lie-Trotter ou Trotterização. O termo $\mathcal{O}$ representa o erro da aproximação e é controlado pela magnitude da discretização em Δt . Para obtermos um valor arbitrário ϵ de erro, o valor de N deve aumentar com $\mathcal{O}(\Delta t/\epsilon)$, sendo N proporcional ao número de exponenciais (WIEBE et al., 2010).

2.9 Algoritmos de estimação de fase

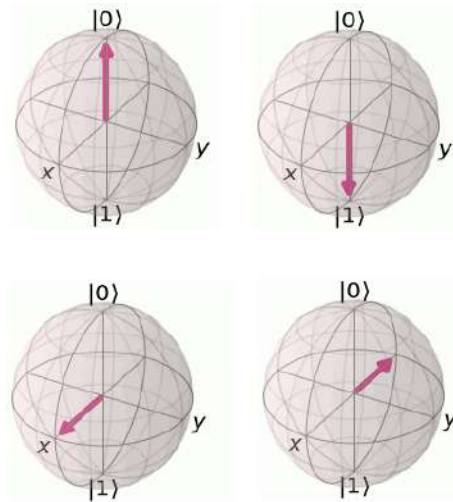
O algoritmo de estimação de fase foi proposto por Abrams e Lloyd em 1999 (ABRAMS; LLOYD, 1999), que buscavam um algoritmo quântico com aplicações diretas para a física e a química. Eles foram inspirados por Kitaev (KITAEV, 1995) e Cleve et al (CLEVE et al., 1998) que propuseram o que hoje é chamado de *Iterative Phase Estimation*, um algoritmo semelhante mas aplicado em um contexto e em aproximações diferentes do buscado por Abrams e Lloyd. Enquanto Kitaev buscava solucionar problemas de otimização e fatoração de números e Cleve e colaboradores queriam otimizar a implementação de portões quânticos, Abrams e Lloyd implementaram essas ideias para encontrar auto-valores e auto-vetores de um Hamiltoniano local.

Dado um operador $e^{i\Phi}$ e um auto-estado desse operador que seja representado por um conjunto de qubits, o algoritmo de estimação de fase pode retornar uma representação binária que corresponda a fase Φ do operador. Se utilizarmos um operador que seja e^{iHt} , com esse algoritmo podemos extrair informações quanto ao espectro de energias do Hamiltoniano de interesse (CAO et al., 2019), como no caso da Trotterização. Para entender melhor a estimação de fase, precisamos primeiro entender um artifício matemático na computação quântica chamado de transformada quântica de Fourier.

2.9.1 Transformada Quântica de Fourier

Uma das bases para a estimação de fase está na transformada quântica de Fourier (QFT, do inglês *Quantum Fourier Transform*), é ela que nos permite ler a fase do sistema que codifica os auto-valores de energia (no caso do operador de evolução temporal e^{iHt} , ou a fase de um operador $e^{i\Phi}$ qualquer) (MCARDLE et al., 2020). De modo geral, a QFT se trata de uma mudança de base da base computacional (Z) $\{|0\rangle, |1\rangle\}$ para a base de Fourier $\{|+\rangle, |-\rangle\}$, ou base (X) (IBM QUANTUM, 2020). Pensando na esfera de Bloch, a transformada muda os estados polares $|0\rangle$ e $|1\rangle$ para estados equatoriais $|\tilde{0}\rangle$ e $|\tilde{1}\rangle$ (IBM QUANTUM, 2020), como podemos ver na figura 8. Essa mudança de base polar para equatorial nos fornece uma redução polinomial de termos para o Hamiltoniano e também uma leitura rápida da fase, como demonstraremos a seguir.

Figura 8 – Esferas de Bloch representando os estados antes e após a transformada quântica de Fourier. As esferas acima representam os estados iniciais e logo abaixo suas respectivas transformadas



Fonte: produção do autor, gerada com o pacote Qiskit (ABRAHAM; ADUOFFEI; AL., 2019).

Podemos definir a transformada da seguinte forma: dado um estado $|x\rangle$ qualquer que, na base computacional, pode ser representado pelos estados $|0\rangle$ ou $|1\rangle$, sua representação na base de Fourier será dada por

$$|\tilde{x}\rangle = QFT |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle, \quad (29)$$

que é análogo à transformada de Fourier clássica inversa. N representa os 2^n estados possíveis para um grupo de n qubits. QFT representa a transformada quântica de Fourier como um operador que atua sobre o estado $|x\rangle$, gerando o estado $|\tilde{x}\rangle$. Podemos entender melhor olhando para o caso de apenas 1 qubit e depois generalizar para n qubits. Para 1 qubit temos $N = 2^1 = 2$:

$$\begin{aligned}
|\tilde{x}\rangle &= QFT |x\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{2^1-1} e^{\frac{2\pi i x y}{2^1}} |y\rangle \\
QFT |x\rangle &= \frac{1}{\sqrt{2}} \left[e^{\frac{2\pi i x * 0}{2}} |0\rangle + e^{\frac{2\pi i x * 1}{2}} |1\rangle \right] \\
QFT |x\rangle &= \frac{1}{\sqrt{2}} \left[|0\rangle + e^{\pi i x} |1\rangle \right], \tag{30}
\end{aligned}$$

quando $|x\rangle = |0\rangle$ ou $|x\rangle = |1\rangle$ teremos:

$$QFT |0\rangle = \frac{1}{\sqrt{2}} \left[|0\rangle + e^{\pi i * 0} |1\rangle \right] = \frac{1}{\sqrt{2}} [|0\rangle + |1\rangle] \tag{31}$$

$$QFT |1\rangle = \frac{1}{\sqrt{2}} \left[|0\rangle + e^{\pi i * 1} |1\rangle \right] = \frac{1}{\sqrt{2}} [|0\rangle - |1\rangle], \tag{32}$$

então para n qubits com N estados possíveis teremos que $|x\rangle$ pode ser representado nessa base como

$$|x\rangle = |x_1 x_2 \dots x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle, \tag{33}$$

onde $|x_j\rangle$ representa o qubit j no estado x_j , onde x_j vale 0 ou 1, o que nos permite escrever [29](#) com a expansão binária

$$|\tilde{x}\rangle = \sum_{y=0}^{N-1} \prod_{k=1}^n e^{2\pi i x \frac{y_k}{2^k}} |y_1 y_2 \dots y_n\rangle. \tag{34}$$

Se pensarmos que

$$\sum_{y=0}^{N-1} = \sum_{y_1=0}^1 \sum_{y_2=0}^1 \dots \sum_{y_n=0}^1, \tag{35}$$

a equação [30](#) se torna

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N}} \left((|0\rangle + e^{\frac{\pi i x}{2^1}} |1\rangle) \otimes (|0\rangle + e^{\frac{\pi i x}{2^2}} |1\rangle) \otimes \dots (|0\rangle + e^{\frac{\pi i x}{2^n}} |1\rangle) \right), \tag{36}$$

que é a representação generalizada de um estado $|x\rangle$ na base de Fourier. Quanto à implementação do algoritmo devemos lembrar que os operadores que atuam nos qubits devem sempre ser unitários para preservar a norma ([BARENCO et al., 1995](#)). Pensando na figura [8](#), isso significa que precisamos de um operador U que faça uma rotação unitária.

No caso para 1 qubit mostrado em 30, o operador unitário *Hadamard* exerce essa função facilmente, de modo que

$$\begin{aligned} H |x_k\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x_k}{2}} |1\rangle) \\ H |0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H |1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \end{aligned} \quad (37)$$

onde H representa o operador Hadamard. Este operador atua no estado de um qubit de formas diferentes, à depender de qual estado ele encontra ativamente no qubit. Sua representação matricial é dada por

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (38)$$

e sua atuação é como descrita pela transformada de Fourier descrita anteriormente, ela nos permite sair dos polos na esfera de Bloch e ir para o equador, criando uma superposição de estados no qubit entre $|0\rangle$ e $|1\rangle$.

Então, para um estado genérico $|x_j\rangle$, precisamos pensar em um operador unitário que exerça uma rotação $e^{\frac{2\pi i}{2^k} x_j}$, isto é

$$U_k |x_j\rangle = e^{\frac{2\pi i}{2^k} x_j} |x_j\rangle. \quad (39)$$

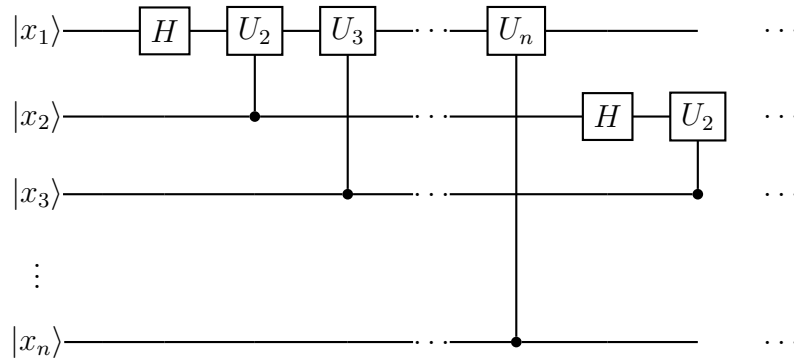
Podemos realizar esta operação através do Qiskit (ABRAHAM; ADUOFFEI; AL., 2019). Lançado em 2017 pela IBM Research, o Qiskit foi criado para facilitar a programação e execução de experimentos em processadores quânticos bem como em simuladores locais de máquinas quânticas. O Qiskit possui em sua estrutura quatro bibliotecas principais que operam em conjunto para seu funcionamento; “Terra”, “Aqua”, “Ignis” e “Aer”. A biblioteca Terra é aquela que compõe a base estrutural do Qiskit e seus demais componentes, é ela quem permite construir os códigos quânticos e manipulá-los como circuitos quânticos através da linguagem OpenQASM e também pulsos com a linguagem OpenPulse. Além disso, ela fornece otimização automática do código para o hardware onde o software será executado, sem que o usuário precise se preocupar ou aprender sobre isso. Já a biblioteca Aqua se trata de uma série de ferramentas criadas para buscar aplicações da computação quântica no mundo real, ela é muito utilizada por especialistas nas áreas de química, finanças e de inteligência artificial. A biblioteca permite que o usuário execute tarefas sem que ele precise fazer uma programação quântica específica, deixando que o próprio Qiskit Aqua aplique o algoritmo quântico já de forma

adequada. O Qiskit Ignis, por sua vez, é destinado a correções de erros e ruídos em resultados. Também fornece ferramentas para quem quer se dedicar a criação de códigos de correção de erros e de otimização de circuitos e códigos quânticos. Por fim, no Qiskit Aer, estão incluídos os simuladores e emuladores utilizados para prever o comportamento dos circuitos e algoritmos quânticos antes de enviá-los ao hardware quântico. As máquinas quânticas disponíveis com acesso gratuito pela IBM são 6 máquinas de 5 qubits e 1 máquina de 1 qubits (IBM QUANTUM, 2016). No total, no acervo da IBM, são 22 máquinas de 1 a 65 qubits porém estas outras são de acesso apenas com parceria com a empresa através de programas da IBM Quantum Network (IBM QUANTUM, 2016). Com ele, podemos aplicar uma rotação através do portão “CPGate” ou “Controlled-CP gate”, que se trata exatamente de uma rotação unitária simétrica que induz uma fase no qubit alvo, dependendo do estado do qubit de controle. Na sua forma matricial teremos

$$U_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}. \quad (40)$$

Então, para efetuar a transformada em n qubits, buscamos um algoritmo que nos forneça um circuito como o da figura 9.

Figura 9 – Circuito quântico para implementação de uma transformada quântica de Fourier



Fonte: produção do autor.

Como podemos notar na figura 9, os qubits são inicializados nos estados $|x_i\rangle$, formando um estado de n qubits $|x\rangle$. Em seguida, uma sequência de operadores é aplicada nos qubits, representada pelas caixas com os nomes dos operadores. Temos também os operadores controlados, em que o resultado de suas operações sobre um qubit depende do estado do qubit que o controla, representados pelas caixas ligadas ao circuitos de outro qubit².

² O circuito quântico se lê da esquerda para direita, varrendo vertical e sequencialmente todos os portões (operadores) atuando em cada qubit individual.

O circuito ao ser inicializado se encontra no estado $|x\rangle$ que sempre irá representar o estado atual do sistema. Aplica-se então o primeiro portão Hadamard no primeiro qubit, que nos leva ao estado

$$|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \frac{x_1}{2^1}} |1\rangle) \otimes |x_2 x_3 \dots x_n\rangle, \quad (41)$$

Na sequencia temos a primeira rotação com o portão controlado, que nos dá:

$$|x'\rangle = (|0\rangle + e^{\frac{2\pi i}{2^2} x_2} e^{\frac{2\pi i}{2^1} x_1} |1\rangle) \otimes |x_2 x_3 \dots x_n\rangle. \quad (42)$$

O próximo portão controlado terá a forma:

$$|x''\rangle = (|0\rangle + e^{\frac{2\pi i}{2^3} x_3} e^{\frac{2\pi i}{2^2} x_2} e^{\frac{2\pi i}{2^1} x_1} |1\rangle) \otimes |x_2 x_3 \dots x_n\rangle, \quad (43)$$

e assim teremos suscetivamente até o qubit n , onde a fase final pode ser representada por

$$e^{2\pi i \left(\frac{x_n}{2^n} + \frac{x_{n-1}}{2^{n-1}} + \dots + \frac{x_1}{2^1} \right)}. \quad (44)$$

Porém, se lembrarmos de (33) podemos mudar a representação de x de binária para decimal, de modo que

$$x = x_1 * 2^{n-1} + x_2 * 2^{n-2} + \dots + x_n * 2^0, \quad (45)$$

o que faz de (44)

$$e^{2\pi i \left(\frac{x}{2^n} \right)}, \quad (46)$$

e então

$$|x\rangle = (|0\rangle + e^{\frac{2\pi i}{2^n} x} |1\rangle) \otimes |x_2 x_3 \dots x_n\rangle, \quad (47)$$

o coeficiente de normalização $\frac{1}{\sqrt{N}}$ foi omitido. Este então é o estado $|\tilde{x}\rangle$ que buscávamos, porém com a ordem reversa da exponencial na fase. Essa inversão pode ser resolvida facilmente com portões “SWAP” entre os qubits que, de maneira simples, funcionam como a operação de "swap" na computação clássica $|a, b\rangle \rightarrow |b, a\rangle$. Para maior aprofundamento sobre o tema recomenda-se visitar (ASFAW et al., 2020).

2.9.2 O algoritmo de estimação de fase

Sabemos que um operador unitário tem auto-valores da forma

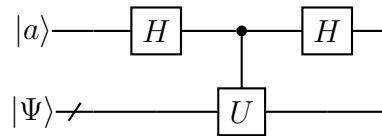
$$U |\Psi\rangle = e^{i\theta_\psi} |\Psi\rangle, \quad (48)$$

mas para obtermos o valor de θ_ψ precisamos de algoritmo capaz de extrair essa informação sem perturbar o estado $|\Psi\rangle$, que é o objetivo do algoritmo proposto.

Para trabalharmos com a estimação de fase precisaremos de dois registros de qubits, dois conjuntos de f qubits que funcionarão com diferentes propósitos. O primeiro se trata de m qubits ancilla ³ $|a\rangle$ inicializados em $|0\rangle$ que usaremos para extrair a fase θ_ψ em uma representação binária. O segundo registro se trata da representação do estado $|\Psi\rangle$ e possui quantos qubits forem necessários para tal. O algoritmo começa aplicando portões *Hadamard* nos qubits ancilla para deixá-los em superposição na base computacional e logo após, aplicamos rotações unitárias no segundo registro de qubit para que aconteça o chamado “phase kick-back” (CLEVE et al., 1998) que nos fornece a fase θ_ψ pelos qubits ancilla sem colapsar o estado $|\Psi\rangle$, como será demonstrado a seguir. Vamos trabalhar com o caso de 1 qubit ancilla

$$|x\rangle = |a\rangle |\Psi\rangle = |0\rangle |\Psi\rangle. \quad (49)$$

Figura 10 – Circuito simples para representação dos dois registros de qubits, de modo a realizar o “phase kick-back”



Fonte: produção do autor.

Inicialmente teremos algo como a figura 10. Ao aplicarmos o portão Hadamard no qubit ancilla nos leva ao estado

$$|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle |\Psi\rangle + |1\rangle |\Psi\rangle). \quad (50)$$

³ qubits ancilla são qubits auxiliares que podem não precisar ser computados ao efetuar uma medida, servem para uma tarefa ou operação específica ser realizada, como emaranhar um qubit para atuar um operador ou armazenar um estado específico

Aplicamos o operador unitário controlado pelo qubit ancilla, que nos leva a

$$|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle |\Psi\rangle + |1\rangle e^{i\theta_\Psi} |\Psi\rangle). \quad (51)$$

O último portão *Hadamard* nos dará a projeção binária da fase e o estado fica

$$|x\rangle = \frac{1}{2}[(|0\rangle + |1\rangle) |\Psi\rangle + e^{i\theta_\Psi} (|0\rangle - |1\rangle) |\Psi\rangle], \quad (52)$$

que podemos escrever como

$$|x\rangle = \frac{1}{2}[(|0\rangle (1 + e^{i\theta_\Psi}) + |1\rangle (1 - e^{i\theta_\Psi})) |\Psi\rangle]. \quad (53)$$

Percebemos que a fase foi “embutida” no qubit ancilla enquanto o estado $|\Psi\rangle$ permaneceu inalterado. Porém, com apenas 1 qubit precisaremos efetuar muitas medidas com o algoritmo para chegarmos a um resultado relevante e satisfatório, dada a característica probabilística do processo de medida (NIELSEN; CHUANG, 2010). Para resolver esse empecilho, deve-se escolher o número m de qubits pensando em duas coisas: quantas casas decimais de precisão buscamos para o valor estimado da fase, e com qual probabilidade de sucesso precisamos trabalhar. De fato, é possível verificarmos que para uma probabilidade de sucesso de δ com n bits de precisão são necessários

$$m = n + \left\lceil \log_2 \left(2 + \frac{1}{2\delta} \right) \right\rceil \quad (54)$$

qubits (NIELSEN; CHUANG, 2010). Assim, usando múltiplos qubits para o algoritmo pode-se estimar a fase com apenas uma medida, que nos leva a um circuito na forma da figura 11.

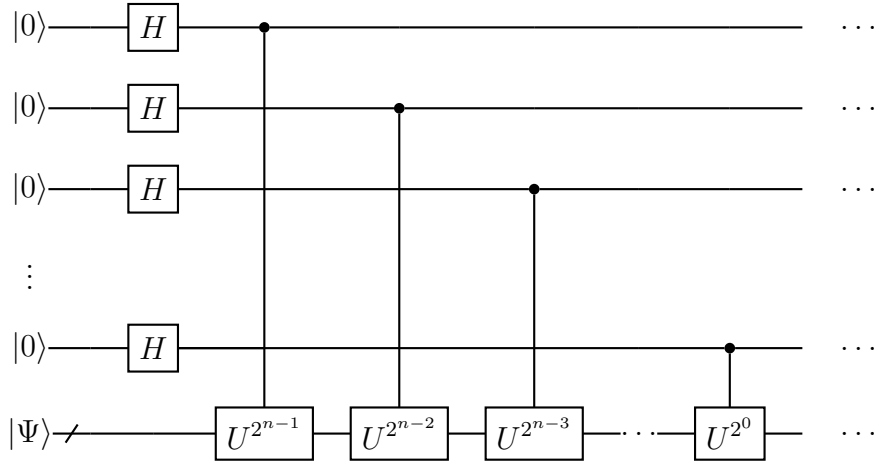
Após a aplicação dos portões *Hadamard* necessários, temos o estado

$$|x\rangle = \frac{1}{(\sqrt{2})^n} (|0\rangle + |1\rangle)^{\otimes n} |\Psi\rangle, \quad (55)$$

e então os portões controlados operam a cada qubit de modo que o estado final após a última atuação será

$$|x\rangle = \frac{1}{(\sqrt{2})^n} [(|0\rangle + e^{i\theta_\psi^{2^{n-1}}} |1\rangle) \otimes (|0\rangle + e^{i\theta_\psi^{2^{n-2}}} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{i\theta_\psi^{2^0}} |1\rangle)] \quad (56)$$

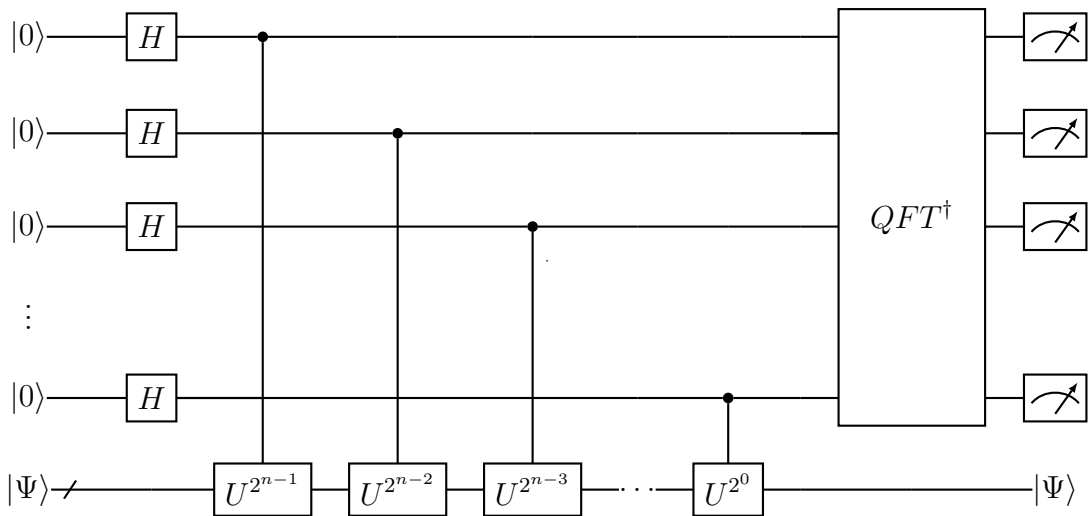
Figura 11 – Diagrama de circuito para implementação do “phase kick-back”, onde sem alterarmos o estado de interesse podemos obter informações através dos qubits ancilla



Fonte: produção do autor.

Podemos notar a incrível semelhança da equação (56) com a (43), com exceção da constante de normalização e também o coeficiente da fase, que para o QFT é $\theta_\psi \frac{2\pi}{2^n}$. Então, o que fizemos foi gerar uma transformada nos qubits ancilla. Para extrairmos a informação da fase basta aplicar uma transformada inversa de Fourier quântica, de modo que o circuito final para o QPE se dá conforme mostra a figura 12.

Figura 12 – Diagrama do circuito completo para o algoritmo de estimação de fase, onde após o “phase kick-back” empregamos uma transformada de Fourier inversa para recuperarmos o estado original pelo qual estamos interessados



Fonte: produção do autor.

Dessa forma, ao executarmos a medida podemos aferir qual fase é obtida em sua representação binária. Se escolhermos U tal qual o Hamiltoniano do sistema de interesse,

poderemos trabalhar sua evolução temporal de forma unitária, onde os auto-valores correspondem aos níveis de energia do sistema (BASSMAN et al., 2021).

É notável também que esse algoritmo não é completo por si só, também é necessária uma maneira de criar o estado $|\Psi\rangle$ de interesse que também é chamado de “*Ansatz*”, e a estimação de fase irá operar nele. Podemos interpretar a estimação de fase como uma sub-rotina de um algoritmo maior (CAO et al., 2019). Infelizmente para sistemas relativamente complexos precisaremos de um circuito de grande profundidade para extrairmos a fase do operador, o que necessita de muitos qubits implementados, os quais as máquinas NISQ não possuem. Além do que, manter a coerência entre as interações dos qubits durante toda a execução do algoritmo só se torna mais difícil quanto mais qubits estão sendo utilizados (PRESKILL, 2018). Porém, é fato que o algoritmo de estimação de fase possui muito potencial. Se utilizarmos um Hamiltoniano que descreva um sistema físico, podemos obter a evolução temporal desse sistema com apenas uma operação do circuito quântico. Com isso, podemos obter informações importantes como o resultado de ação de operadores, relações de dispersão e propriedades topológicas do material, com apenas uma execução, pois o uso de operadores de evolução temporal sequenciais não são necessários para esse algoritmo (CAO et al., 2019).

2.10 Algoritmos quânticos variacionais

A estimação de fase é o algoritmo de maior eficácia que conhecemos atualmente (BAUER et al., 2020), porém, dadas as atuais máquinas quânticas disponíveis para simulações, é inviável operar um algoritmo tão profundo e que requer tantos qubits para sistemas muito complexos. Uma das soluções para esse empecilho e efetuar as simulações de materiais é utilizar algoritmos variacionais, mais especificamente o chamado “*Variational Quantum Eigensolver*” (VQE).

O VQE se trata de um algoritmo híbrido que, até a presente data, foi aplicado somente em segunda quantização (MCARDLE et al., 2020). Por ser um algoritmo híbrido, utiliza tanto da máquina quântica como da clássica para encontrar o menor auto-valor do Hamiltoniano que descreve uma determinada configuração. Sua melhor aplicabilidade se dá ao fato de substituir longos procedimentos que requerem altos tempos de coerência do algoritmo de estimação de fase, por pequenas evoluções com baixo tempo de coerência (PERUZZO et al., 2014).

O papel da máquina quântica é preparar o estado de interesse e realizar sua medida, obtendo o valor esperado do Hamiltoniano nesse estado, enquanto na máquina clássica processamos essa medida e fazemos uma atualização no estado da máquina quântica de acordo com nosso interesse. De modo específico, buscamos encontrar a configuração com o menor autovalor do Hamiltoniano, representando o estado de menor energia do

sistema descrito por esse Hamiltoniano.

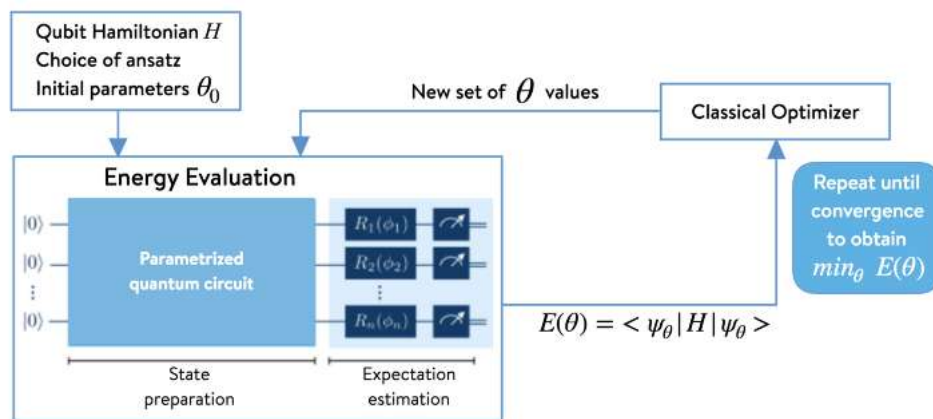
O primeiro passo para efetuarmos a simulação é mapear o Hamiltoniano em uma soma linear das matrizes de Pauli, que torna possível trabalhá-lo de forma direta na base dos qubits. Precisamos então de um circuito parametrizado $V(\theta)$ com o qual podemos preparar o *ansatz*

$$|\Phi(\theta)\rangle = V(\theta) |\Psi_0\rangle, \quad (57)$$

onde $|\Psi_0\rangle$ representa a base computacional. A escolha de $V(\theta)$ deve levar em consideração a limitação de hardware com o qual será trabalhado e também o sistema específico a ser avaliado, isto é, o Hamiltoniano que será trabalhado. Em segunda quantização podemos mapear um Hamiltoniano hermitiano em termos de uma combinação linear de matrizes de Pauli, como já mencionamos anteriormente. Para preparar-se o *ansatz* devemos escolher um conjunto de portões que melhor represente um auto-estado do Hamiltoniano e consequentemente, a performance do algoritmo depende também da estrutura do *ansatz* gerado (CAO et al., 2019).

Então, mede-se o valor esperado de H e esse valor é passado para a máquina clássica, que irá variar θ e repetir o procedimento até que o valor de $\langle \Psi(\theta) | H | \Psi(\theta) \rangle$ seja minimizado, onde H representa o Hamiltoniano do sistema em termos das matrizes de Pauli. Para uma escolha adequada de *ansatz*, tal que todas as possíveis configurações do sistema estejam representadas, a minimização paramétrica do Hamiltoniano irá representar o estado fundamental do sistema (MCCLEAN et al., 2016).

Figura 13 – Diagrama de operação do algoritmo VQE. Inicializamos o *ansatz* na máquina quântica e realizamos a medida do seu valor esperado, passamos o resultado para um otimizador operado na máquina clássica que busca os parâmetros para o menor auto-valor daquele estado e repetimos a medida através da máquina quântica, com os novos parâmetros retornados pelo otimizador



Fonte: (1QBIT, 2019).

O VQE é um algoritmo flexível, no sentido que podemos escolher os portões

parametrizados que irão representar o *ansatz* e buscarmos uma configuração mais otimizada para execução da otimização (CAO et al., 2019). Também, a quantidade de configurações possíveis de serem representadas com n -qubits escala com 2^n , podendo assim representar sistemas complexos com um número reduzido de qubits. Ele é seguro à certos erros computacionais (MCCLEAN et al., 2016) e já se mostrou capaz de encontrar os *ground-states* de pequenos sistemas utilizando circuitos de baixa profundidade (KANDALA et al., 2017; MCARDLE et al., 2020). Podemos visualizar sua lógica de operação através da figura 13, onde as etapas de cada máquina durante a execução do algoritmo híbrido são esquematizadas.

Porém, devemos ter em mente que se trata de um algoritmo híbrido e também heurístico, onde a otimização dos parâmetros pode não convergir de forma adequada dependendo principalmente do *ansatz* utilizado (MCCLEAN et al., 2016). Além do que, o número de medidas e execuções para essa convergência pode ser extremamente alto com este mesmo fator.

2.11 Aplicação dos algoritmos na simulação de sistemas sólidos

Em nossas buscas do uso de computação quântica para a física de materiais encontramos alguns trabalhos com resultados interessantes e que nos ajudaram a progredir com o nosso. Aqui, citaremos alguns deles.

Em (KIVLICHAN et al., 2018) os autores propõem uma otimização para simulação de um Hamiltoniano de estrutura eletrônica, em termos do número de portões quânticos e qubits necessários para simular um “*Trotter step*”. Para um número N de orbitais de spin, eles conseguiram executar um algoritmo baseado na Trotterização com $N^2/2$ portões de 2-qubits emaranhados e preparar um determinante de Slater arbitrário com $N/2$ de profundidade. Com sua aproximação conseguiram aumentar as perspectivas de algoritmos que são praticáveis nas máquinas NISQ e que podem ser úteis para sistemas com um tamanho não-trivial.

Em (MURTA; CATARINA; FERNÁNDEZ-ROSSIER, 2020) os autores estimam a fase de Berry para um átomo e propõem o uso de seu algoritmo para sistemas interagentes de tamanho não-trivial de forma analítica, permitindo a classificação topológica de um modelo. Para isso, utilizaram o algoritmo de estimação de fase de modo iterativo, além é claro da computação quântica baseada em portões quânticos para simular evoluções adiabáticas cíclicas dos operadores.

Em (IBE et al., 2021) temos uma extensão do algoritmo VQE chamada de VQD (Variational Quantum Deflation), além de outras extensões do algoritmo. Aqui, eles propõem uma aproximação para calcular as amplitudes de transição através dos

algoritmos, para diversos sistemas reais. Os autores foram capazes de realizar as simulações em computadores quânticos reais, o que demonstrou que seu algoritmo é praticável nas máquinas NISQ e abrem novos caminhos para a aproximação de estados excitados utilizando máquinas quânticas.

Um trabalho de particular importância para nossa pesquisa é o presente na referência (CERASOLI et al., 2020), onde os autores utilizam o método VQE atrelado ao modelo Tight-Binding para simular um modelo de duas bandas para o silício. Os autores usam um ansatz simples de 1 qubit com um Hamiltoniano 2×2 , e também propõem um modelo mais complexo com um Hamiltoniano 8×8 juntamente com um ansatz de 3 qubits para calcular mais bandas e modelar a estrutura eletrônica de forma mais completa. Este trabalho serviu como inspiração para nossas simulações.

Em (MEI et al., 2020) os autores trabalham um algoritmo de baixa profundidade, viável para máquinas NISQ com o intuito de simular acoplamentos spin-órbita e propriedades topológicas calculadas de modo analítico. Eles demonstraram ser possível estudar propriedades topológicas para materiais utilizando um circuito com 8 qubits, sendo possível até mesmo o estudo de sistemas interagentes e simetrias topológicas protegidas que, segundo eles, são de grande interesse para a ciência de materiais mas uma tarefa além da capacidade da computação clássica.

Na referência (KANNO; TADA, 2021) é apresentado um uso do VQE para resolver o problema de muitos corpos para materiais periódicos. Os autores ressaltam a escassez de trabalhos no campo da física de materiais, e propõem um algoritmo variacional para construções de modelos de primeiros princípios para sistemas periódicos. Utilizando um método chamado RBM-based VQE (restricted Boltzmann machine-based variational quantum eigensolver) foi possível calcular os auto-valores do modelo Hubbard com aproximação de campo médio, para a estrutura de bandas do grafeno.

3 SIMULAÇÃO DE SISTEMAS PERIÓDICOS USANDO COMPUTADORES QUÂNTICOS

No mesmo passo da revisão bibliográfica também encaminhamos um trabalho de simulação de materiais, utilizando-se dos métodos apreendidos nesta revisão. A descrição dessa etapa do trabalho de pesquisa se encontra a seguir.

Existem diversas dificuldades em se manter e operar uma máquina quântica, que vão desde problemas inerentes aos fundamentos da mecânica quântica quanto problemas de engenharia para construção dessas máquinas. Para adquirimos uma informação de um sistema quântico geramos uma perturbação que destrói sua função de onda e, se queremos que o sistema opere em perfeitas condições, devemos mantê-lo isolado do ambiente externo.

Nosso trabalho de simulação foi feito utilizando máquinas fornecidas pela IBM acessadas via nuvem, com o pacote de ferramentas fornecido pela própria empresa chamado Qiskit ([ABRAHAM; ADUOFFEI; AL., 2019](#)). Para realizar simulações foi necessário apenas um computador local com a biblioteca Qiskit instalada bem como um ambiente Python e o ambiente Jupyter como IDE (Integrated Development Environment) para se trabalhar os algoritmos. Para realizar experimentos em computadores quânticos reais, foi preciso efetuar um cadastro na plataforma “IBM Q”, da IBM, e a partir daí foi possível executar os algoritmos nos computadores fornecidos, através de um *token* agregado ao Qiskit.

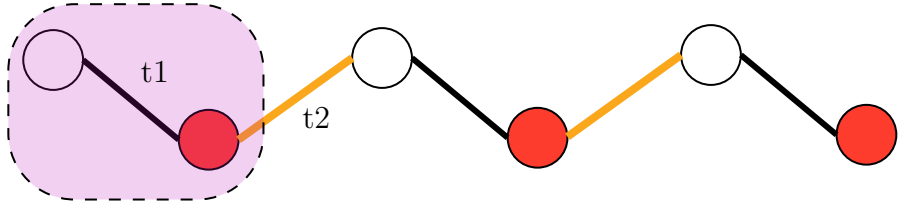
Fazendo o uso destas ferramentas, planejamos, em um primeiro momento, utilizar de algoritmos variacionais em segunda quantização para um estudo inicial do uso das máquinas quânticas na física de materiais cristalinos. Escolhemos o silício e o grafeno inicialmente como objeto de estudo. Para tanto, utilizamos dois modelos para os Hamiltonianos de materiais periódicos, que são os modelos de Su-Schrieffer-Heeger ([BATRA; SHEET, 2020](#)) e de Haldane ([HALDANE, 1988](#)). Ambos são modelos onde empregamos um Hamiltoniano do tipo Tight-Binding em segunda quantização e que, para sistemas como o silício e o grafeno, podem ser descritos em termos de poucos qubits.

A seguir, apresentaremos com mais detalhes os algoritmos utilizados no presente trabalho.

3.1 Modelo Su-Schrieffer-Heeger (SSH)

Uma vez descrito o tipo de Hamiltoniano a ser utilizado, devemos determinar a quantidade de orbitais na célula unitária e a forma com a qual esses orbitais interagem entre si e com os orbitais das células unitárias vizinhas. O modelo SSH consiste em um Tight-Binding para um sistema unidimensional periódico com dois orbitais nas células unitárias (BATRA; SHEET, 2020). Teremos então duas possibilidades de salto, entre os elétrons da mesma célula unitária e entre os elétrons de células vizinhas, como mostra a figura 14.

Figura 14 – SSH para uma dimensão



Fonte: produção do autor.

Na figura 14, t_1 é a probabilidade de salto dentro da própria célula, e t_2 a interação entre células unitárias vizinhas. Nesse caso, teremos um Tight-Binding da forma

$$\mathbf{H} = \sum_{i=1}^N [\varepsilon_a c_{i,a}^\dagger c_{i,a} + \varepsilon_b c_{i,b}^\dagger c_{i,b}] + \sum_{i=1}^N t_1 [c_{i,a}^\dagger c_{i,b} + h.c] + \sum_{i=1}^{N-1} t_2 [c_{i,b}^\dagger c_{i+1,a} + h.c], \quad (58)$$

onde a e b denotam os orbitais com energias ε_a e ε_b respectivamente, e $h.c$ se trata do conjugado que mantém o Hamiltoniano hermitiano. O primeiro somatório contabiliza a energia potencial e os dois seguintes a energia cinética dos saltos entre os elétrons da própria célula i -ésimo e os elétrons de células vizinhas i -ésimo e $(i+1)^{th}$, respectivamente. Vamos considerar aqui $t_1 = t_2 = t$, uma vez que os orbitais em que os saltos ocorrem são ambos do tipo p .

Para o silício, o orbital mais energético é o orbital $3p$ que é preenchido com dois elétrons, que se cristaliza numa estrutura do tipo liga de zinco, com ligações covalentes entre seus orbitais de valência (IBACH; LÜTH, 2010). Os dois elétrons são compartilhados entre átomos adjacentes, podendo ocupar dois estados nos orbital p . Nesse sentido, o modelo SSH pode ser utilizado como uma aproximação para descrever os orbitais de valência do silício, o que motiva sua utilização nesse trabalho.

Aplicando a transformação canônica de Fourier podemos escrever os operadores

de criação e aniquilação como

$$c_j^\dagger = \frac{1}{\sqrt{N}} \sum_k e^{ikx_j a_0} c_k^\dagger \quad (59)$$

$$c_j = \frac{1}{\sqrt{N}} \sum_k e^{-ikx_j a_0} c_k, \quad (60)$$

onde N representa o número de células, $c_k^\dagger$ e c_k são os operadores de criação e aniquilação na base dos momentos, a_0 define o tamanho da célula unitária e tem unidade de $1/m$ e x_j , a posição do sítio j -ésimo. Transpondo o Hamiltoniano na segunda quantização temos:

$$\mathbf{H} = \frac{1}{N} \sum_{j,k,k'} \left[\varepsilon e^{-i(k-k')x_j a_0} c_k^\dagger c_{k'} + t \left(e^{-i(k-k')x_j a_0} e^{-ika_0} + h.c. \right) c_k^\dagger c_{k'} \right]. \quad (61)$$

Devido a ortogonalidade, podemos escrever $\mathbf{H}$ na forma diagonalizada

$$\mathbf{H} = \sum_k [\varepsilon + t(e^{ika_0} + e^{-ika_0})] c_k^\dagger c_k, \quad (62)$$

onde o somatório j -ésimo sobre o exponencial pôde ser escrito como a função $\delta_{k,k'}$. Podemos então escrever $\mathbf{H}$ na forma compacta

$$\mathbf{H} = \sum_k \varepsilon_k c_k^\dagger c_k, \quad (63)$$

onde $\varepsilon_k = \varepsilon + 2t \cos(ka)$, uma relação de dispersão periódica em $[0 < k \leq 2\pi/a]$.

Para o caso unidimensional do modelo SHH com dois orbitais por célula unitária, podemos escrever o Hamiltoniano de forma similar a eq (5)

$$\mathbf{H} = \sum_x \left[\mathbf{c}_x^\dagger \underline{\mathbf{M}} \mathbf{c}_x + \mathbf{c}_x^\dagger \underline{\mathbf{P}} \mathbf{c}_{x+1} \right], \quad (64)$$

onde a matriz $\underline{\mathbf{T}}$ foi decomposta em duas matrizes $\underline{\mathbf{M}}$ e $\underline{\mathbf{P}}$. $\underline{\mathbf{M}}$ é uma matriz 2×2 representando o salto dentro da própria célula, tal que

$$\underline{\mathbf{M}} = \begin{bmatrix} \mu & t_1 \\ t_1 & \mu \end{bmatrix}, \quad (65)$$

onde μ se trata to potencial químico e t_1 a probabilidade de salto dentro da própria célula unitária. Já a matriz $\underline{\mathbf{P}}$ computa o tunelamento entre as células vizinhas, de modo que

$$\underline{\mathbf{P}} = \begin{bmatrix} 0 & t_2 \\ t_2 & 0 \end{bmatrix}. \quad (66)$$

Logo, a relação de dispersão total pode ser representada por

$$\underline{E}_{k_x} = \begin{bmatrix} \mu & t_1 \\ t_1 & \mu \end{bmatrix} + \begin{bmatrix} 0 & 2t_2 \cos(k_x a_0) \\ t_2 \cos(k_x a_0) & 0 \end{bmatrix}. \quad (67)$$

Se escolhermos $\psi = [\psi_1 \ \psi_2]^T$ tal que

$$\underline{E}_{k_x} \psi = \lambda \psi. \quad (68)$$

onde podemos escrever o sistema de equações

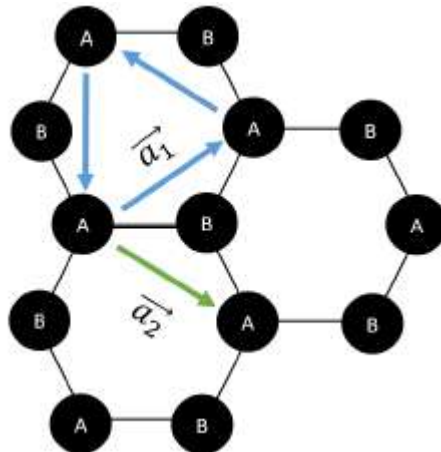
$$\begin{bmatrix} \mu & \alpha \\ \alpha & \mu \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = (\mu \pm \alpha) \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}, \quad (69)$$

com $\alpha = t_1 + 2t_2 \cos(k_x a_0)$, tal que $\lambda = \mu \pm \alpha$ representa as autonenergias do sistema em função do número de onda k .

3.2 Modelo de Haldane

Outro modelo interessante de se utilizar é o modelo de Haldane ([HALDANE, 1988](#); [LE et al., 2020](#)). A implementação do modelo de Haldane segue os mesmos princípios descritos para o modelo SSH, porém, esse novo modelo também levará em consideração a interação de vizinhos próximos na cadeia do material abrindo um novo grau de liberdade. Isso nos permite modelar outros materiais e obter uma imagem 3D de sua topografia. Nas primeiras simulações utilizamos a estrutura cristalina do grafeno para experimentação, como mostra a figura 15.

Figura 15 – Cadeia hexagonal para o modelo de Haldane onde teremos mais uma possibilidade de salto. Além de dentro da célula unitária e entre vizinhos diretos também são considerados saltos entre vizinhos próximos



Fonte: ([LE et al., 2020](#)).

Com essa interação entre segundos vizinhos podemos descrever um Hamiltoniano cuja relação de dispersão reflete diretamente a relação de dispersão do grafeno, na forma

$$H_{\mathbf{k}} = \begin{bmatrix} g(k) & f(k) \\ f^*(k) & -g(k) \end{bmatrix}, \quad (70)$$

onde a função $f(k)$ corresponde aos saltos entre os primeiros vizinhos

$$f(k) = t_1 \left[1 + \left(e^{-ika_1} + e^{-ika_2} \right) \right] \quad (71)$$

e o termo $g(k)$ as demais interações tal que

$$g(k) = \frac{\Delta_0}{2} + 2t_2 \left(\text{sen}(ka_1) - \text{sen}(ka_2) + \text{sen}(k(\Delta a)) \right), \quad (72)$$

onde Δ_0 é um termo ligado a massa atômica considerada.

Temos então dois Hamiltonianos que descrevem dois sistemas físicos diferentes de forma matemática. Nas próximas seções iremos descrever como mapear esses hamiltonianos nos qubits, de modo a manipulá-los com operadores quânticos através das máquinas e simuladores.

4 RESULTADOS

Com a plataforma Qiskit em mãos e os princípios do algoritmo VQE, decidimos utilizar ambos em conjunto para efetuar simulações de materiais cristalinos. O primeiro passo foi definir qual tipo de Hamiltoniano seria utilizado. Optamos pelo Tight-Binding e, quanto aos modelos para descrever um material cristalino, escolhemos primeiramente o modelo Su-Schrieffer-Heeger (SSH). Este modelo se trata de um modelo Tight-Binding para um sistema unidimensional com dois orbitais entre as células unitárias, desconsiderando interações de spin, que é útil para descrever sistemas periódicos como dimmers (IBACH; LÜTH, 2010). Após as simulações com o SSH buscamos um modelo mais avançado, o modelo de Haldane que é um Tight-Binding onde além das probabilidades entre os primeiros vizinhos como no SSH, também são consideradas interações entre vizinhos adjacentes.

4.1 Simulação para o Silício

Em nossa simulação trabalhamos com a estrutura de bandas do Silício, que possui uma rede cúbica do tipo diamante. Consideramos o caso simples de apenas um orbital por átomo onde a célula unitária possui 2 graus de liberdade, e também que a ligação entre os átomos de silício é a mesma em toda a estrutura, de modo que a probabilidade de salto na célula unitária seja t .

Seja a energia do orbital representada por μ com uma célula unitária em uma posição $\mathbf{R}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ da rede. Então teremos

$$\underline{\mathbf{M}} = \begin{bmatrix} \mu & t \\ t & \mu \end{bmatrix}. \quad (73)$$

O salto entre os eixos x, y, z nos fornece uma matriz semelhante em todas as direções

$$\underline{\mathbf{T}} = \begin{bmatrix} 0 & t \\ t & 0 \end{bmatrix}, \quad (74)$$

de modo que a relação de dispersão total é dada por

$$\underline{\mathbf{E}}_{\mathbf{k}} = \begin{bmatrix} \mu & t \\ t & \mu \end{bmatrix} + 2t \left(\begin{bmatrix} 0 & \cos k_x a_0 \\ \cos k_x a_0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \cos k_y a_0 \\ \cos k_y a_0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \cos k_z a_0 \\ \cos k_z a_0 & 0 \end{bmatrix} \right), \quad (75)$$

onde $\mathbf{k}$ é um vetor de onda específico, e então

$$\underline{E}_{\mathbf{k}} = \begin{bmatrix} \mu & t_k \\ t_k & \mu \end{bmatrix}, \quad (76)$$

tal que

$$t_k = t \left[1 + 2 \sum_{i=x,y,z} \cos(k_i a_0) \right]. \quad (77)$$

De modo que a matriz diagonalizada para a relação de dispersão ficará na forma

$$\underline{E}_{\mathbf{k}}^D = \begin{bmatrix} \mu + t_k & 0 \\ 0 & \mu - t_k \end{bmatrix}, \quad (78)$$

que é a que iremos trabalhar para a simulação. Como podemos notar, o Hamiltoniano representando a relação de dispersão para o Silício, dada a periodicidade de rede cristalina, é dado por uma matriz que pode ser diagonalizada analiticamente sem maiores dificuldades. O ponto em questão é a verificação da possibilidade do uso de computadores quânticos para mapear tal Hamiltoniano e obter seus autoestados. O fato de termos acesso à valores analíticos permite uma validação direta dos métodos empregados, possibilitando uma análise precisa dos componentes empregados na simulação quântica.

Como mencionado na seção 2.10, o primeiro passo para implementação do VQE se trata de mapear o Hamiltoniano na base dos qubits. Isso é feito utilizando matrizes de Pauli $\hat{\sigma}_i$ que vão descrever a matriz na base de spin meio, que é a própria base computacional dos qubits. As matrizes de Pauli são:

$$\begin{aligned} \sigma_x &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \sigma_y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ \sigma_z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ \sigma_{\mathbb{I}} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

Teremos que a relação de dispersão será escrita como

$$\underline{E}_{\mathbf{k}}^D = \sum_{i=x,y,z,\mathbb{I}} c_{i,\mathbf{k}} \hat{\sigma}_i, \quad (79)$$

onde o coeficiente $c_{i,\mathbf{k}}$ é da forma

$$c_{i,\mathbf{k}} = \frac{1}{2} \text{Tr}(\hat{\sigma}_i \underline{\mathbf{E}}_{\mathbf{k}}). \quad (80)$$

O coeficiente para cada matriz de Pauli será então

$$c_{z,\mathbf{k}} = t_k \quad (81)$$

$$c_{1,\mathbf{k}} = \mu \quad (82)$$

$$c_{x,\mathbf{k}} = 0 \quad (83)$$

$$c_{y,\mathbf{k}} = 0, \quad (84)$$

de modo que, na base dos qubits, a relação de dispersão diagonalizada é escrita

$$\underline{\mathbf{E}}_{\mathbf{k}}^D = \mu \sigma_{\mathbb{1}} + t_k \sigma_z. \quad (85)$$

Assim, o valor esperado para o Hamiltoniano será composto do valor esperado das quatro matrizes de Pauli, onde o menor auto-valor corresponde a

$$\epsilon_0(k) = \mu \langle \sigma_{\mathbb{1}} \rangle + t_k \langle \sigma_z \rangle \quad (86)$$

para o qual podemos escrever explicitamente

$$\langle \sigma_{\mathbb{1}} \rangle = \langle \psi_0 | \sigma_{\mathbb{1}} | \psi_0 \rangle \quad (87)$$

$$\langle \sigma_z \rangle = \langle \psi_0 | \sigma_z | \psi_0 \rangle. \quad (88)$$

Aqui, ψ_0 se trata da função de onda associada ao estado de menor energia do sistema, que é definida através do processo variacional.

Sabemos que as medidas na verdade são feitas na base computacional projetando-as nos autoestados $|0\rangle$ e $|1\rangle$. A matriz $\sigma_{\mathbb{1}}$ sempre tem auto-valores iguais a 1, independentemente do sistema ou circuito utilizado, podemos atribuí-la ao valor esperado do Hamiltoniano. Já a matriz σ_z possui auto-valores +1 ou -1 ao projetarmos na base computacional, associados aos estados $|0\rangle$ e $|1\rangle$, respectivamente. Então, para garantirmos a confiança na medida efetuada devemos repeti-la o maior número de vezes possível. Seja p a probabilidade de medirmos +1 em vez de -1, o valor esperado de σ_z será

$$\langle \sigma_z \rangle = (+1)p + (-1)(1 - p) = 2p - 1. \quad (89)$$

Já σ_x e σ_y não comutam com σ_z , não compartilhando seus autovetores, e, portanto, não podendo ser medidos diretamente na base computacional. Então, devemos

aplicar operadores unitários antes de realizarmos a medida para representá-los nesta base. Fazemos

$$\sigma_x^{comp} = \hat{H}^\dagger \sigma_x \hat{H}, \quad (90)$$

$$\sigma_y^{comp} = \hat{H}^\dagger \hat{S}^\dagger \sigma_y \hat{S} \hat{H}, \quad (91)$$

onde $\hat{H}$ é o operador Hadamard, e $\hat{S}$ se trata do operador de mudança de fase. Após ambos operarem no qubit o procedimento para obter o valor esperado é semelhante ao de σ_z

$$\langle \sigma_x^{comp} \rangle = (+1)p_x + (-1)(1 - p_x) = 2p_x - 1, \quad (92)$$

$$\langle \sigma_y^{comp} \rangle = (+1)p_y + (-1)(1 - p_y) = 2p_y - 1, \quad (93)$$

onde $p_{x(y)}$ é a probabilidade de medir-se $|0\rangle$ em vez de $|1\rangle$ para os operadores $\sigma_{x(y)}^{comp}$.

Ao realizarmos essas etapas, podemos com sucesso obter os auto-valores correspondentes ao estado de menor energia do sistema. Para obtermos o valor esperado para estados excitados devemos fazer uma decomposição do espectro do Hamiltoniano. Vamos escrever o Hamiltoniano como projetores ponderados pelo auto-valor ϵ_j

$$\mathbf{H} = \epsilon_0 |\psi_0\rangle \langle \psi_0| + \sum_{j=1} \epsilon_j |\psi_j\rangle \langle \psi_j|, \quad (94)$$

onde o primeiro termo condiz com o estado de menor auto-valor. Seja $\mathbf{H}'$ o segundo somatório, podemos escrever

$$\mathbf{H}' = \mathbf{H} - \epsilon_0 |\psi_0\rangle \langle \psi_0|, \quad (95)$$

mas podemos escrever um operador matriz de densidade para

$$\rho_0 = |\psi_0\rangle \langle \psi_0|. \quad (96)$$

Logo podemos chamar o termo para o menor auto-valor como

$$\mathbf{H}_0 = \epsilon_0 \rho_0, \quad (97)$$

que permite escrevermos a soma de matrizes de Pauli

$$\mathbf{H}_0 = \epsilon_0 \sum_k d_k \sigma_k, \quad (98)$$

e por conseguinte

$$d_k = \frac{1}{2} \text{Tr} \left\{ \frac{\mathbf{H}_0^\dagger \sigma_k}{\epsilon_0} \right\} = \frac{1}{2} \text{Tr} \{ \rho_0^\dagger \sigma_k \}. \quad (99)$$

Utilizando a propriedade

$$\text{Tr} \{ (\rho \mathbf{A}) \} = \langle \mathbf{A} \rangle, \quad (100)$$

temos

$$d_k = \frac{1}{2} \langle \sigma_k \rangle. \quad (101)$$

O Hamiltoniano geral ficará na forma

$$\mathbf{H}' = \mathbf{H} - \sum_k \frac{\epsilon_0}{2} \langle \sigma_k \rangle \sigma_k, \quad (102)$$

que podemos simplificar como

$$\mathbf{H}' = \sum_k \left[c_i - \frac{\epsilon_0}{2} \langle \sigma_k \rangle \right] \sigma_k. \quad (103)$$

Como subtraímos o estado de menor auto-valor, o que obtemos então é o primeiro estado excitado do Hamiltoniano ([HIGGOTT; WANG; BRIERLEY, 2019](#))

$$\epsilon_1 = \langle \mathbf{H}' \rangle = \sum_k \left[c_i - \frac{\epsilon_0}{2} \langle \sigma_k \rangle \right] \langle \sigma_k \rangle. \quad (104)$$

Repetindo o processo, podemos obter então os estados excitados seguintes.

Com tudo isso em mãos, nos resta apenas determinar o circuito no qual aplicaremos $\underline{E}_{\mathbf{k}}$ para obtenção dos auto-valores. Podemos representá-lo conforme a figura 16 ([CERASOLI et al., 2020](#)), onde os dois operadores iniciais $R_y(\theta)$ e $R_z(\phi)$ levam o qubit que inicia no estado $|0\rangle$ para o estado $|\psi(\theta, \phi)\rangle$, isto é

$$|\psi(\theta, \phi)\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle. \quad (105)$$

Para medirmos então o valor esperado do Hamiltoniano basta fazermos a projeção no estado $|\psi(\theta, \phi)\rangle$

$$\langle \underline{E}_{\mathbf{k}} \rangle = \langle \psi(\theta, \phi) | \underline{E}_{\mathbf{k}} | \psi(\theta, \phi) \rangle. \quad (106)$$

Para a implementação do algoritmo na máquina quântica fizemos uma alteração no circuito anterior da figura 16. Fazíamos a medida de todas as matrizes de Pauli no mesmo

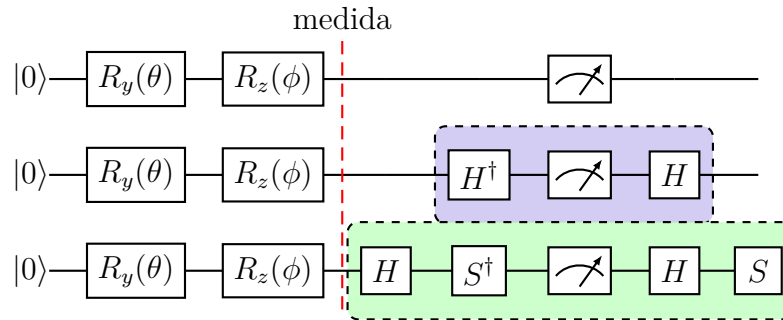
Figura 16 – Circuito aplicado à E_k para obtenção dos auto-valores, também chamado de ansatz

$$|0\rangle \xrightarrow{R_y(\theta)} \xrightarrow{R_z(\phi)} |\psi\rangle \xrightarrow{E_k}$$

Fonte: (CERASOLI et al., 2020)

qubit, o que gerava a necessidade de 3 execuções para uma só medida e, conseqüentemente, aumentava o tempo total de execução. Para a implementação do modelo SSH nas máquinas quânticas e de Haldane no simulador, compomos um circuito de 3 qubits, que pode ser visto na figura 17, levando a somente uma execução para uma medida completa. Como em nosso circuito não há necessidade de operadores com 2-qubits, essa decomposição do *bitstring* é possível de forma simples pois os qubits operam de forma independente. Cada qubit realizará a medida para um dos coeficientes das matrizes da Pauli, nas bases Z, X e Y respectivamente de cima para baixo no circuito. Um detalhe importante é que, como não há máquinas de 3 qubits para acesso na IBM, executamos o algoritmo em máquinas de 5 ou mais qubits. Porém, os qubits que não são necessários no circuito não são inicializados na máquina quântica, de modo que não impactam no resultado final e nem na interpretação do bitstring de retorno. O resultado das simulações pode ser visto na seção 4.1.1. O código fonte contendo o algoritmo utilizado pode ser encontrado no apêndice.

Figura 17 – Circuito quântico de 3 qubits implementado para acelerar o processo de medida, onde medimos os coeficientes de todas as matrizes de Pauli numa mesma execução



Fonte: produção do autor.

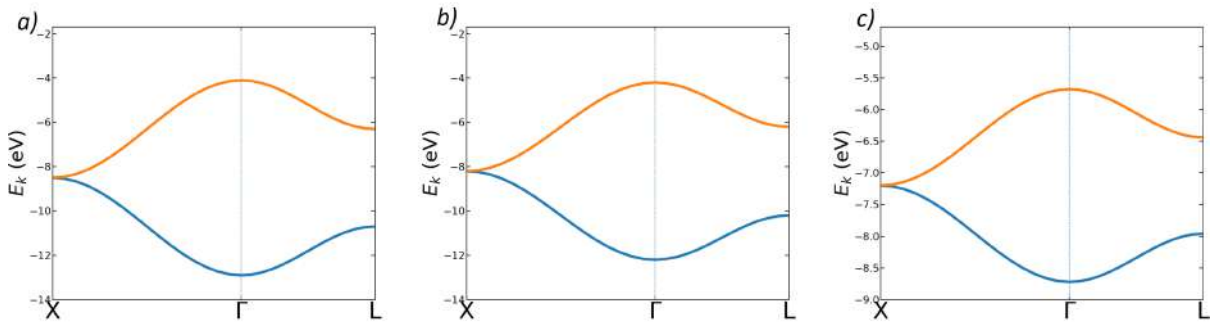
4.1.1 Estruturas de bandas

O algoritmo foi executado tanto no simulador local quanto na máquina real da IBM. O Hamiltoniano que descreve o comportamento do silício para o modelo SSH utilizado para o algoritmo VQE foi o seguinte

$$H_{SSH} = \begin{pmatrix} \mu & t_1 + t_2(\cos(\vec{k}) + i\sin(\vec{k})) \\ t_1 + t_2(\cos(\vec{k}) - i\sin(\vec{k})) & \mu \end{pmatrix}, \quad (107)$$

onde $\vec{k}$ varia em um caminho entre os pontos de alta simetria na primeira zona de Brillouin para a estrutura cristalina do Silício. t_1 e t_2 foram considerados iguais e seu valor foi ajustado para obter a estrutura de bandas que mais se assemelhava aos resultados esperados da literatura. Como podemos notar na figura 18(a), o caminho escolhido no espaço dos momentos foi entre os pontos de alta simetria $X - \Gamma - L$ e foram obtidos os autovalores em função do vetor de onda $\vec{k}$. Para o silício o valor de $t_1 = t_2$ foi de 0.60, e o valor de μ utilizado para as simulações foi de -8.5. Este último valor determina a energia potencial química e opera como um shift global, como podemos ver pela equação 85, onde a energia é dada pelo valor esperado da matriz identidade vezes μ mais o produto de σ_z por t_k .

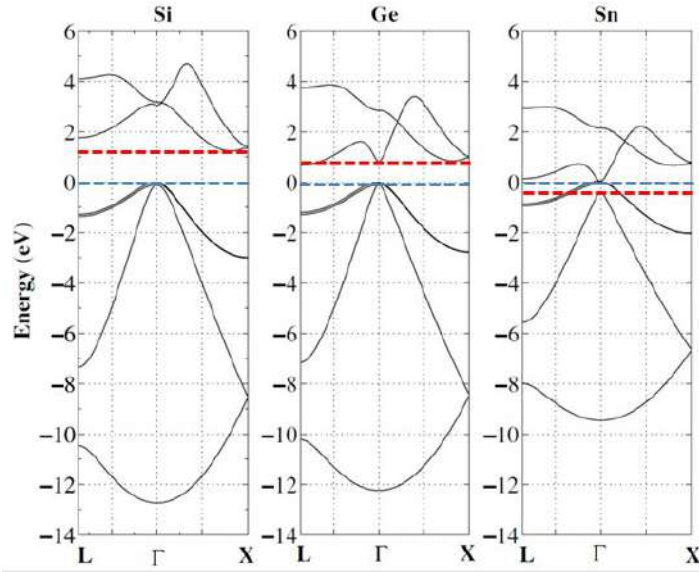
Figura 18 – Comportamento de diferentes átomos executadas no simulador QASM. (a) representa os dados para o silício, (b) para o germânio e (c) para o α -Sn



Fonte: produção do autor.

Pertencente à mesma família do silício, temos o germânio que apresenta uma configuração eletrônica $[\text{Ar}]3d^{10}4s^24p^2$ e o valor para t_k na simulação foi de 0.45 e μ o valor de -8.2. Tendo em vista a similaridade nos orbitais de valência, é de se esperar que o mesmo Hamiltoniano representando os orbitais de valência para o silício possa representar os orbitais de valência para o germânio. Nesse sentido, a principal diferença entre os dois sistemas periódicos se encontra nos parâmetros t e μ no Hamiltoniano, mas o comportamento geral dos autovalores em função de $\vec{k}$ deve ser preservado. Os resultados da simulação da relação de dispersão para o germânio se encontram na figura 18(b). Finalmente, temos o estanho ($[\text{Kr}]4d^{10}5s^25p^2$), que também pertence à família do Silício. No entanto apenas a fase α do estanho possui um arranjo cristalino também do tipo blenda de zinco (diamante), refletindo uma estrutura eletrônica semelhante ao silício para os orbitais de valência. Seu valor para t_k foi de 0.19 e μ de -7.2. Diferentemente do silício e do germânio que são semicondutores, a fase α do estanho é metálica. Os resultados da simulação da relação de dispersão para fase α do estanho se encontram na figura 18(c).

Figura 19 – Estrutura de bandas calculados por computação clássica



Fonte: (MOONTRAGOON; IKONIĆ; HARRISON, 2007).

A título de comparação, apresentamos na figura 19 a estrutura eletrônica para o silício (painel da esquerda), germânio (painel central) e fase α do estanho (painel da direita) calculadas usando DFT (MOONTRAGOON; IKONIĆ; HARRISON, 2007). Podemos notar que o nível de mais baixa energia das duas simulações, usando a computação quântica e o DFT, é muito semelhante para todos os materiais. Tendo em vista que o Hamiltoniano empregado para a obtenção da relação de dispersão usando VQE é uma matriz de ordem dois, só esperamos obter dois valores para as autoenergias desse Hamiltoniano. A comparação entre as energias em função do número de onda para o segundo estado (curva laranja) possui uma diferença significativa em seu perfil. Isso se dá principalmente pela simulação usando o VQE ser bastante simplificada. Nela, não levamos em consideração a interação entre bandas a qual é responsável pela mudança no perfil do segundo estado para todas as estruturas na figura 19. O próximo passo na simulação da estrutura de bandas usando o VQE, objetivando a obtenção de resultados mais próximos aos esperados, seria a utilização de Hamiltonianos de ordem superior, que levem em consideração termos de acoplamento entre as bandas. Tais Hamiltonianos poderiam ser conseguidos, aumentando os termos de Hooping no Hamiltoniano Tight-Binding, por exemplo. Em seu trabalho (CERASOLI et al., 2020), Cerasoli e colaboradores demonstraram que para se obter bandas mais altas do silício era necessário diagonalizar um Hamiltoniano 8×8 em um circuito de 3 qubits relativamente profundo e que necessitava de um ansatz mais complexo. Sua simulação demandava mais poder computacional e também métodos para análise e mitigação de erro, mas se mostrou possível de ser realizado.

De modo geral, conseguimos obter os resultados quanto a estrutura de bandas dos três materiais utilizando um computador quântico por meio da implementação do

modelo SSH e sua diagonalização usando o algoritmo VQE. É importante notarmos que os valores assumidos pelos parâmetros t estão diretamente ligados às propriedades do material. Como o método Tight-Binding é semi-empírico, os parâmetros utilizados devem ser ajustados conforme o resultado experimental. Porém, ainda assim o método fornece informações importantes sobre a dinâmica eletrônica e podemos estender seus resultados para obter-se outras propriedades como, por exemplo, a fase de Berry do material e por conseguinte suas transições de fase topológicas. Mesmo que os resultados obtidos não sejam tão próximos dos esperados e obtidos pela computação clássica, é possível trilhar-se um grande caminho para este tipo de técnica. Podemos usar outros métodos para obtenção de Hamiltonianos mais completos e que descrevam interações relevantes para a dinâmica atômica onde se pode descrever a estrutura de bandas de forma mais geral, como é o caso do método KP (BASTARD, 1990).

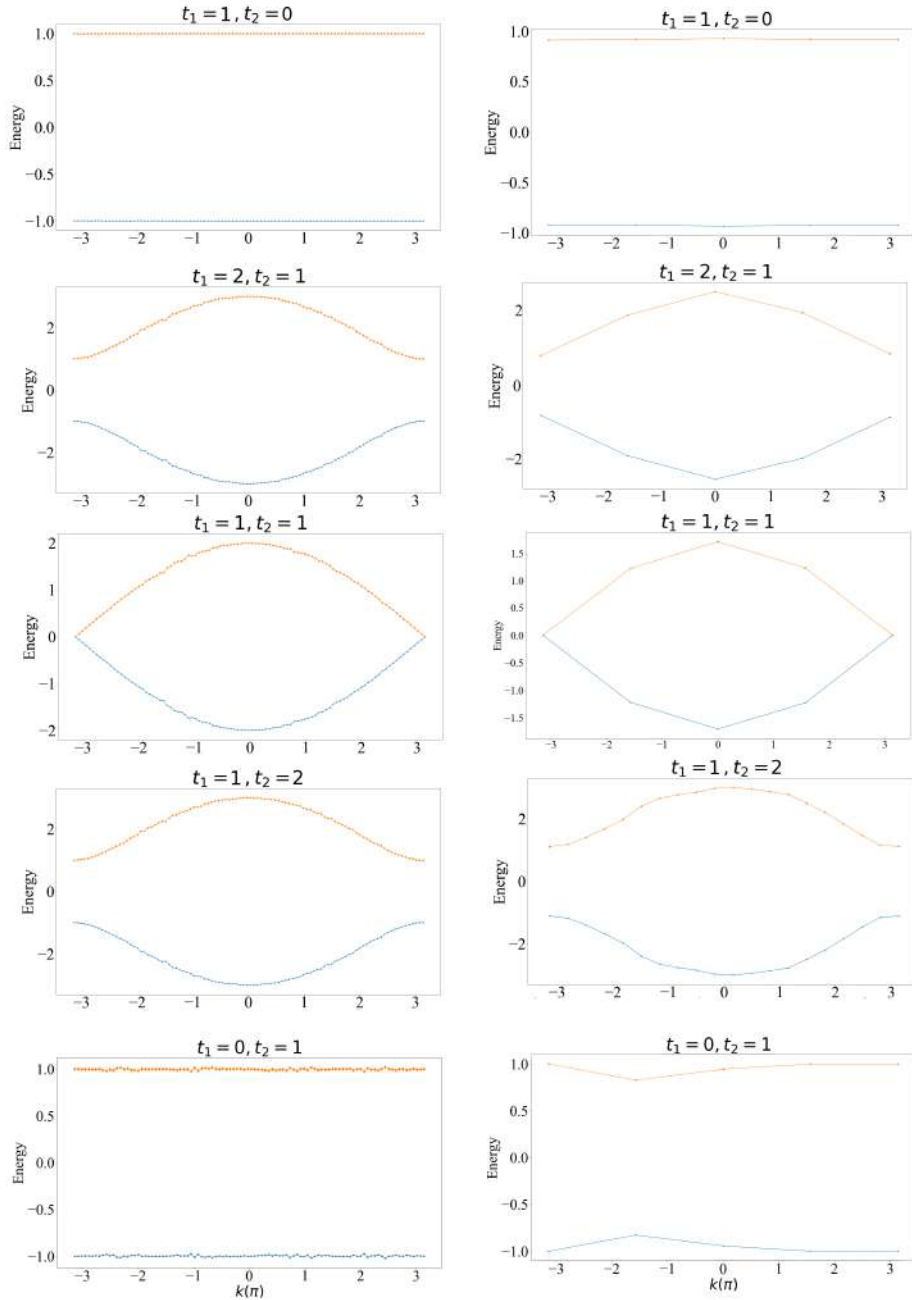
Ainda assim, mesmo os resultados não gerando retornos de forma exata ao esperado na literatura, continuamos o utilizando para outros Hamiltonianos, continuando a análise da viabilidade do método VQE. Visto que há uma grande esperança para solução de problemas bem mais complexos do que os possíveis de serem simulados com computadores clássicos, dado o escalonamento de estados possíveis com 2^n à medida que adicionamos qubits, esse tipo de solução ainda se mostra bastante promissora. Porém, por se tratar de um algoritmo híbrido, é fato que essa técnica carrega muito da base de métodos já utilizados na computação clássica. No entanto, vislumbramos um desprendimento dessas bases clássicas e um lado inovador para o uso de computadores quânticos na análise de propriedades de sistemas cristalinos periódicos como, por exemplo, na obtenção de estados excitados e estimativas de taxa de transição diretamente pela atuação de operadores no estado quântico, antes da minimização (IBE et al., 2021; MA; GOVONI; GALLI, 2020).

4.1.2 Fases topológicas

Utilizando o mesmo modelo VQE podemos generalizar a matriz (107) para obter comportamento de fases topológicas em cadeias unidimensionais, o resultado está na figura 20. Através do modelo SSH podemos obter indícios de uma transição de fase topológica, através de um vetor de onda k que varia linearmente de $-\pi/a$ até π/a ao variarmos os índices de hopping t_1 e t_2 . Esses valores correspondem ao número de junções topologicamente protegidas (MURTA; CATARINA; FERNÁNDEZ-ROSSIER, 2020). A medida que variamos os índices podemos ver mudanças no gap de energia entre o estado fundamental e o primeiro estado excitado.

Podemos notar que para o simulador (painéis da esquerda) as figuras possuem muitos pontos, enquanto que para a máquina quântica (painéis da direita) poucos pontos são representados. Isso se dá ao fato das máquinas reais serem ainda de difícil acesso sem ter-se algum vínculo com a IBM para acessar todas as máquinas que a empresa possui. Há

Figura 20 – Comportamento do Hamiltoniano a medida que variamos t_k , à esquerda os resultados para o simulador e à direita os resultados para a máquina quântica da IBM



Fonte: produção do autor.

filas para que os trabalhos sejam executados nos processadores, a média de tempo para o retorno de 1 ponto no gráfico era de 4 horas de execução, tendo atenção para enviar o algoritmo para uma máquina com pouca fila no dia. Para o simulador realizamos as medidas com um circuito inicial de 1 qubit, o que levava a 3 execuções diferentes para medir todas as matrizes de Pauli. Este circuito foi modificado para que tivesse 3 qubits e, em uma única execução, todas as medidas fossem realizadas, o que resultou em uma

redução considerável do tempo de computação para utilizarmos as máquinas quânticas da IBM, mais especificamente no processador *ibmq_belem* de 5 qubits. A figura 18 possui dados apenas do simulador, devido a dificuldade de acesso as máquinas reais da IBM.

Para obtermos os diversos comportamentos de uma cadeia *dimmer*, moléculas formadas por dois átomos diferentes na sua estrutura, basta variarmos os parâmetros de hopping t_1 e t_2 entre 0 e 2, como podemos ver na figura 20. Esta é a relação de dispersão para duas bandas, temos a energia dessas bandas em função do número de onda limitada aos valores do vetor de onda $\vec{k}$. As bandas variam de acordo com os parâmetros t_k que determinam o salto entre as células unitárias e, dependendo da relação entre as intensidades dos parâmetros de salto, dentro da célula e entre as células, o perfil de bandas é bastante alterado. Para $t_2 = 0$ temos duas dispersões lineares de acordo com a magnitude do termo t_1 com o material tendo um comportamento de isolante, conforme podemos observar com o gap entre as bandas. Para $t_2 > t_1$ e $t_2 < t_1$ o material também se comporta como um isolante, conforme demonstrado pelo gap que, ainda que tenha diminuído, continua aberto entre as bandas. Quando temos $t_1 = t_2$ observamos que nos valores extremos de k o gap se fecha, temos uma transição de isolante para metal. Portanto, modulando a relação entre estes parâmetros pudemos observar indícios dessa transição de fase entre metal e isolante, para o Hamiltoniano generalizado. Esse tipo de comportamento mostra uma possível transição de fase topológica.

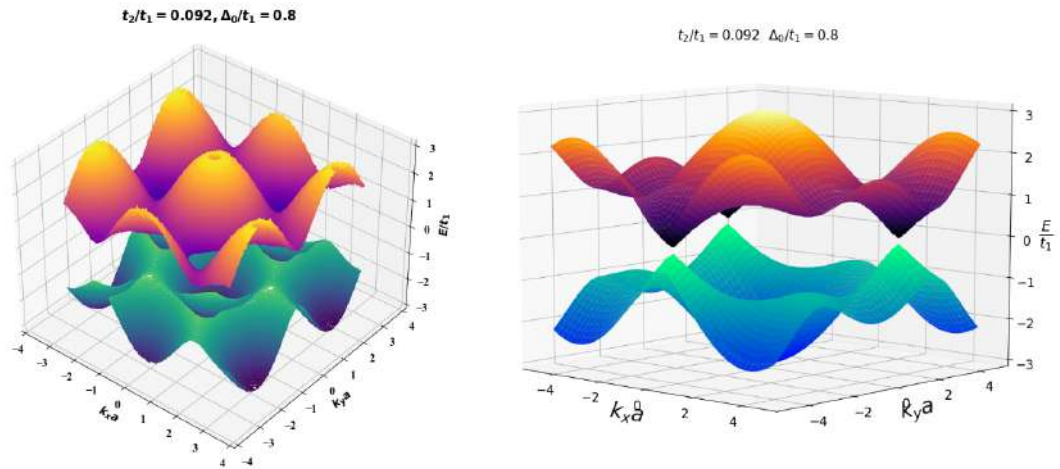
Para atestarmos corretamente se realmente ocorre uma mudança de fase topológica do material, teríamos de calcular a fase de Berry com o modelo SSH para uma classificação topológica eficaz. Essa classificação leva em consideração um *winding number*, que é uma invariante topológica diretamente ligada a fase de Berry. Para afirmarmos uma transição na fase topológica devemos avaliar alterações nesse parâmetro, o que não é diretamente possível apenas com a simulação do modelo SSH usando o VQE. Para esse modelo já foi demonstrado ser possível calcular a fase de Berry utilizando computação quântica por meio do IPE (Iterative Phase Estimation) (MURTA; CATARINA; FERNÁNDEZ-ROSSIER, 2020), que muda de π para zero a medida que a razão t_1/t_2 varia de 0 até 2.

4.2 Simulações para o grafeno

Para o grafeno utilizamos o mesmo algoritmo VQE. Mudamos o Hamiltoniano utilizado para o silício pelo Hamiltoniano encontrado em (70).

Para o modelo de Haldane utilizamos o simulador local, que nos gerou um resultado muito semelhante com o esperado na literatura. Na figura 21 comparamos o nosso resultado (painel da esquerda) com o esperado para o modelo que encontramos na literatura (painel da direita) (HAMILTONIAN, 2019).

Figura 21 – Modelo de Haldane para o grafeno simulado na máquina local para $t_1 > t_2$ à esquerda. À direita temos o valor esperado na literatura para o modelo



Fonte: produção do autor e ([HAMILTONIAN, 2019](#)).

Temos então uma relação de dispersão para uma cadeia bidimensional. Podemos observar a formação dos pontos de Dirac nesse material e, com bastante sucesso, reproduzimos o resultado esperado na literatura utilizando da computação quântica. Segundo ([HAMILTONIAN, 2019](#)) com este modelo temos duas fases isolantes topologicamente diferentes, onde podemos associar as bordas dessas fases com a variação dos parâmetros no modelo de SSH ([HAMILTONIAN, 2019](#); [HAO et al., 2008](#)).

Não encontramos este tipo de resultado feito com computação quântica em nenhuma referência até o presente momento. Acreditamos que esta seja a primeira demonstração para um modelo simples da estrutura eletrônica do grafeno utilizando-se de máquinas quânticas.

Neste modelo, também é possível demonstrar transições topológicas variando a relação entre os parâmetros, então também poderia-se obter essas propriedades topológicas através das máquinas quânticas.

A revisão bibliográfica quanto ao uso de computação quântica no estudo de materiais nos revelou diversos trabalhos de grupos de pesquisa em todas as partes do mundo, demonstrando ser uma área de ciência de fronteira que está ganhando mais espaço e resultados a cada dia. As referências encontradas foram de extrema ajuda e o bom resultado de nossas simulações comprova a veracidade das informações publicadas.

A pesquisa de materiais pode estar prestes a receber grandes ferramentas, a computação quântica, mesmo que em fase embrionária, tem mostrado um grande potencial para a simulação de moléculas e materiais. Os algoritmos e resultados atualmente disponíveis, mesmo que limitados pela quantidade de qubits e ruídos nos retornos, provaram ter certa aplicabilidade e grande potencial de melhoria.

Também mostramos ser possível executar simulações com computadores quânticos de qualquer lugar, através de acesso via nuvem de máquinas disponíveis por empresas privadas, que no nosso caso se tratou da IBM.

Em nossos resultados obtivemos e comparamos com sucesso a estrutura eletrônica para materiais silício e o grafeno. Com certa diferença com os resultados esperados, as simulações para o silício ainda se mostraram promissoras, sendo possível melhorar a exatidão com um modelo mais complexo que leve em consideração interações com estados superiores de energia. Para o grafeno obtivemos um resultado próximo ao esperado na literatura, utilizando o modelo de Haldane e um simulador de máquina quântica, o resultado é bastante promissor. O uso de computadores quânticos se mostrou possível para trabalhar modelos aproximados dos materiais. O uso de mais qubits para efetuar o algoritmo demonstrou grande otimização no tempo de execução, como se era esperado, mesmo para algoritmos simples como o que utilizamos.

Utilizar a computação quântica para simulações de materiais com um grande número de átomos e interações ainda é inviável, pela limitação que possuem as máquinas NISQ. Porém, para sistemas mais simples que demandem poucos qubits, tanto as nossas simulações como as que encontramos e citamos, demonstraram bons resultados e boa aplicabilidade para a computação quântica ao se estudar propriedades de sólidos periódicos.

Acreditamos ser possível melhorar ainda mais os resultados e também aprofundar um pouco mais o que se é obtido utilizando a computação quântica, como estimar-se a fase de Berry com o modelo de Haldane através do VQE, bem como para diversos materiais com o modelo SSH.

REFERÊNCIAS

- 1QBIT. **1QBit Open QEMIST documentation**. 2019. Disponível em: <<http://openqemist.1qbit.com/docs/>>. Acesso em: 23 out. 2021.
- ABRAHAM, Héctor; ADUOFFEI; AL., Rochisha Agarwal et. **Qiskit: An Open-source Framework for Quantum Computing**. 2019. Disponível em: <<https://github.com/Qiskit/qiskit>>. Acesso em: 2 abr. 2020.
- ABRAMS, Daniel S.; LLOYD, Seth. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. **Physical Review Letters**, v. 83, n. 24, p. 5162–5165, 1999. ISSN 10797114.
- ABRAMS, Daniel S; LLOYD, Seth. Simulation of many-body fermi systems on a universal quantum computer. **Physical Review Letters**, v. 79, n. 13, p. 2586–2589, 1997. ISSN 10797114.
- ARUTE, Frank et al. Quantum supremacy using a programmable superconducting processor. **Nature**, Springer US, v. 574, n. 7779, p. 505–510, 2019. ISSN 14764687.
- ASFAW, Abraham et al. **Learn Quantum Computation Using Qiskit**. 2020. Disponível em: <<http://community.qiskit.org/textbook>>. Acesso em: 4 dez. 2020.
- BARENCO, Adriano et al. Elementary gates for quantum information. **Phys. Rev. A**, v. 52, n. 5, p. 3457–3467, 1995.
- BARENDT, R. et al. Coherent josephson qubit suitable for scalable quantum integrated circuits. **Physical Review Letters**, v. 111, n. 8, p. 1–5, 2013. ISSN 00319007.
- BASSMAN, Lindsay et al. Simulating Quantum Materials with Digital Quantum Computers. **arXiv**, v. 2101.08836, 2021.
- BASTARD, Gerald. **Wave mechanics applied to semiconductor heterostructures**. Hoboken, Nova Jersey: John Wiley e Sons Inc., 1990. v. 1.
- BATRA, Navketan; SHEET, Goutam. Physics with Coffee and Doughnuts: Understanding the Physics Behind Topological Insulators Through Su-Schrieffer-Heeger Model. **Resonance**, v. 25, n. 6, p. 765–786, 2020. ISSN 0973712X.
- BAUER, Bela et al. Quantum Algorithms for Quantum Chemistry and Quantum Materials Science. **Chemical Reviews**, v. 120, n. 22, p. 12685–12717, 2020. ISSN 15206890.
- BECKER, P. et al. Determination of the Avogadro constant via the silicon route. **Metrologia**, v. 40, n. 5, p. 271–287, 2003. ISSN 00261394.

- BIANCHETTI, R. **Control and readout of a superconducting artificial atom**. 2010. Tese (Doutorado) – Quantum Device Lab, Swiss Federal Institute of Technology, Zürich, Alemanha, 2010.
- BORN, Max; OPPENHEIMER, Robert. Zur quantentheorie der molekeln. **Annalen der physik**, Wiley Online Library, v. 389, n. 20, p. 457–484, 1927.
- BRAVYI, Sergey et al. Tapering off qubits to simulate fermionic Hamiltonians. **arXiv**, v. 1701.08213, n. 1, p. 1–15, 2017.
- BRAVYI, Sergey B.; KITAEV, Alexei Yu. Fermionic quantum computation. **Annals of Physics**, v. 298, n. 1, p. 210–226, 2002. ISSN 00034916.
- CAO, Yudong et al. Quantum chemistry in the age of quantum computing. **Chemical Reviews**, ACS Publications, v. 119, n. 19, p. 10856–10915, 2019.
- CASADY, J. B.; JOHNSON, R. W. Status of silicon carbide (SiC) as a wide-bandgap semiconductor for high-temperature applications: A review. **High-Temperature Electronics**, v. 39, n. 96, p. 511–524, 1998.
- CASPARIS, L. et al. Gatemon benchmarking and two-qubit operations. **Physical Review Letters**, v. 116, n. 15, p. 1–5, 2016. ISSN 10797114.
- CERASOLI, Frank T. et al. Quantum computation of silicon electronic band structure. **Physical Chemistry Chemical Physics**, v. 22, n. 38, p. 21816–21822, 2020. ISSN 14639076.
- CLARKE, John; WILHELM, Frank K. Superconducting quantum bits. **Nature**, v. 453, n. 7198, p. 1031–1042, 2008. ISSN 14764687.
- CLEVE, R. et al. Quantum algorithms revisited. **Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 454, n. 1969, p. 339–354, 1998. ISSN 13645021.
- D’ALESSANDRO, DeannaM.; SMIT, Berend; LONG, JeffreyR. Carbon Dioxide Capture: Prospects for New Materials. **Angewandte Chemie International Edition**, v. 49, n. 35, p. 6058–6082, 2010.
- DENRY, Isabelle; HOLLOWAY, Julie A. Ceramics for dental applications: A review. **Materials**, v. 3, n. 1, p. 351–368, 2010. ISSN 19961944.
- DIMENSIONS. **Digital Science Research Solutions**. 2018. Disponível em: <<https://app.dimensions.ai>>. Acesso em: 12 set. 2021.
- DIVINCENZO, David P. The physical implementation of quantum computation. **Fortschritte der Physik**, v. 48, n. 9-11, p. 771–783, 2000. ISSN 00158208.
- DOBSON, John F; VIGNALE, Giovanni; DAS, Mukunda P. **Electronic density functional theory: recent progress and new directions**. Alemanha: Springer Science & Business Media, 1996. v. 1.

- DOROZHKIN, S. V. Current state of bioceramics. **Journal of Ceramic Science and Technology**, v. 9, n. 4, p. 353–370, 2018. ISSN 21909385.
- EL CHAAR, L.; LAMONT, L. A.; EL ZEIN, N. Review of photovoltaic technologies. **Renewable and Sustainable Energy Reviews**, Elsevier Ltd, v. 15, n. 5, p. 2165–2175, 2011. ISSN 13640321.
- FEYNMAN, Richard P. Simulating physics with computers. **International Journal of Theoretical Physics**, v. 21, n. 6-7, p. 467–488, 1982. ISSN 00207748.
- FOWLER, Austin G. et al. Surface codes: Towards practical large-scale quantum computation. **Physical Review A - Atomic, Molecular, and Optical Physics**, v. 86, n. 3, 032324(1–48), 2012. ISSN 10502947.
- GECK, Lotte et al. Control electronics for semiconductor spin qubits. **Quantum Science and Technology**, v. 5, n. 1, p. 1–15, 2020. ISSN 20589565.
- GEORGESCU, I. M.; ASHHAB, S.; NORI, Franco. Quantum simulation. **Reviews of Modern Physics**, v. 86, n. 1, p. 153–185, 2014. ISSN 15390756.
- GOHANE, S. A. et al. Solid Dispersion: A Review. **International Journal of Institutional Pharmacy and Life Sciences**, v. 5, n. 1, p. 248–266, 2015.
- GOOGLE SCHOLAR. Google. 2004. Disponível em: <<https://scholar.google.com>>. Acesso em: 12 set. 2021.
- GOTTESMAN, Daniel. An introduction to quantum error correction and fault-tolerant quantum computation. **arXiv**, v. 0904.2557, 2010.
- HALDANE, F. D.M. Model for a quantum hall effect without landau levels: Condensed-matter realization of the "parity anomaly". **Physical Review Letters**, v. 61, n. 18, p. 2015–2018, 1988. ISSN 00319007.
- HAMILTONIAN, Bloch. **Topological phases in 1D , 2D and tight-binding models**, in proceedings, n. 2, p. 2–6, 2019.
- HAO, Ningning et al. Topological edge states and quantum Hall effect in the Haldane model. **Physical Review B - Condensed Matter and Materials Physics**, v. 78, n. 7, p. 1–6, 2008. ISSN 10980121.
- HAUTIER, Geoffroy; JAIN, Anubhav; ONG, Shyue Ping. From the computer to the laboratory: Materials discovery and design using first-principles calculations. **Journal of Materials Science**, v. 47, n. 21, p. 7317–7340, 2012. ISSN 00222461.
- HENTALL, Philip L et al. New materials for polymer electrolyte membrane fuel cell current collectors. **Journal of Power Sources**, v. 80, n. 1, p. 235–241, 1999. ISSN 0378-7753.
- HIGGOTT, Oscar; WANG, Daochen; BRIERLEY, Stephen. Variational quantum computation of excited states. **Quantum**, v. 3, p. 1–11, 2019. ISSN 2521327X.

- HOUCK, A. A. et al. Controlling the spontaneous emission of a superconducting transmon qubit. **Physical Review Letters**, v. 101, n. 8, p. 1–4, 2008. ISSN 00319007.
- HU, Xuedong; DE SOUZA, Rogerio; SARMA, S. DAS. Decoherence and Dephasing in Spin-Based Solid State Quantum Computers. **arXiv**, cond-mat/0108339, p. 3–11, 2002.
- IBACH, Harald; LÜTH, Hans. **Solid-state physics: An introduction to principles of materials science**. Alemanha: Springer Science & Business Media, 2010. v. 35. ISBN 9783540938033.
- IBE, Yohei et al. Calculating transition amplitudes by variational quantum deflation. **arXiv**, v. 2002.11724v2, p. 1–12, 2021.
- IBM QUANTUM, Team. **IBM Quantum**. 2016. Disponível em: <<https://quantum-computing.ibm.com/>>. Acesso em: 31 abr. 2021.
- IBM QUANTUM, Team. **Introduction to Quantum Computing and Quantum Hardware**. 2020. Disponível em: <<http://qiskit.org/learn/intro-qc-qh>>. Acesso em: 12 set. 2021.
- JORDAN, Pascual; WIGNER, Eugene Paul. uber das paulische äquivalenzverbot. In: THE Collected Works of Eugene Paul Wigner. Switzerland: Springer, 1993. P. 109–129.
- JOSEPHSON, B.D. Possible new effects in superconductive tunnelling. **Physics Letters**, v. 1, n. 7, p. 251–253, 1962. ISSN 0031-9163.
- KANDALA, Abhinav et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. **Nature**, Nature Publishing Group, v. 549, n. 7671, p. 242–246, 2017.
- KANE, B. E. A silicon-based nuclear spin quantum computer. **Nature**, v. 393, n. 6681, p. 133–137, 1998. ISSN 00280836.
- KANNO, Shu; TADA, Tomofumi. Many-body calculations for periodic materials via restricted Boltzmann machine-based VQE. **Quantum Science and Technology**, IOP Publishing, v. 6, n. 2, 025015(1–13), 2021.
- KITAEV, A. Yu. Quantum measurements and the Abelian Stabilizer Problem. **arXiv**, quant-ph/9511026, p. 1–22, 1995.
- KITTEL, Charles; MCEUEN, Paul; MCEUEN, Paul. **Introduction to solid state physics**. New York: Wiley New York, 1996. v. 8.
- KIVLICHAN, Ian D. et al. Quantum Simulation of Electronic Structure with Linear Depth and Connectivity. **Physical Review Letters**, American Physical Society, v. 120, n. 11, 110501(1)–110501(6), 2018. ISSN 10797114.
- KOCH, Jens et al. Charge-insensitive qubit design derived from the Cooper pair box. **Physical Review A - Atomic, Molecular, and Optical Physics**, v. 76, n. 4, p. 1–19, 2007. ISSN 10502947.

- KRAUSS, Thomas F; RUE, Richard M De La. Photonic crystals in the optical regime. **Progress in Quantum Electronics**, v. 23, p. 51–96, 1999.
- LE, Nguyen H. et al. Topological phases of a dimerized Fermi–Hubbard model for semiconductor nano-lattices. **npj Quantum Information**, Springer US, v. 6, n. 1, p. 1–10, 2020. ISSN 20566387.
- LIDAR, Daniel A; WANG, Haobin. Calculating the thermal rate constant with exponential speedup on a quantum computer. **Physical Review E**, APS, v. 59, n. 2, p. 2429–2438, 1999.
- LOSS, Daniel; DIVINCENZO, David P; DIVINCENZO, P. Quantum computation with quantum dots. **Phys. Rev. A**, v. 57, n. 1, p. 120–126, 1997.
- LUND, Austin P; BREMNER, Michael J; RALPH, Timothy C. Quantum sampling problems, BosonSampling and quantum supremacy. **npj Quantum Information**, Nature Publishing Group, v. 3, n. 1, p. 1–8, 2017.
- MA, He; GOVONI, Marco; GALLI, Giulia. Quantum simulations of materials on near-term quantum computers. **npj Computational Materials**, v. 6, n. 1, p. 1–8, 2020. ISSN 20573960.
- MAZUREK, Anna Helena; SZELESZCZUK, Łukasz; PISKLAK, Dariusz Maciej. Periodic DFT calculations—Review of applications in the pharmaceutical sciences. **Pharmaceutics**, v. 12, n. 5, p. 415, 2020. ISSN 19994923.
- MCARDLE, Sam et al. Quantum computational chemistry. **Reviews of Modern Physics**, American Physical Society, v. 92, n. 1, 1500(1)–15003(51), 2020. ISSN 15390756.
- MCCLEAN, Jarrod R. et al. Discontinuous Galerkin discretization for quantum simulation of chemistry. **New Journal of Physics**, v. 22, n. 9, p. 1–21, 2020. ISSN 13672630.
- MCCLEAN, Jarrod R. et al. The theory of variational hybrid quantum-classical algorithms. **New Journal of Physics**, IOP Publishing, v. 18, n. 2, 023023(1–22), 2016. ISSN 13672630.
- MEI, Feng et al. Digital Simulation of Topological Matter on Programmable Quantum Processors. **Physical Review Letters**, v. 125, n. 16, p. 1–8, 2020. ISSN 10797114.
- MICHELLE, J. Mc Cann et al. A review of thin-film crystalline silicon for solar cell applications. Part 1: Native substrates. **Solar Energy Materials and Solar Cells**, v. 68, n. 2, p. 135–171, 2001. ISSN 09270248.
- MOOIJ, J. E. et al. Josephson persistent-current qubit. **Science**, v. 285, n. 5430, p. 1036–1039, 1999. ISSN 00368075.

- MOONTRAGOON, Pairot; IKONIĆ, Zoran; HARRISON, Paul. Band structure calculations of Si-Ge-Sn alloys: Achieving direct band gap materials. **Semiconductor Science and Technology**, v. 22, n. 7, p. 742–748, 2007. ISSN 02681242.
- MOTTA, Mario et al. Low rank representations for quantum simulation of electronic structure. **npj Quantum Information**, v. 7, n. 1, p. 1–8, 2021. ISSN 20566387.
- MULLER, Richard P.; BLUME-KOHOUT, Robin. The Promise of Quantum Simulation. **ACS Nano**, v. 9, n. 8, p. 7738–7741, 2015. ISSN 1936086X.
- MURTA, Bruno; CATARINA, G.; FERNÁNDEZ-ROSSIER, J. Berry phase estimation in gate-based adiabatic quantum simulation. **Phys. Rev. A**, American Physical Society, v. 101, 020302(1–5), 2 2020.
- NAKAMURA, Y.; PASHKIN, Yu A.; TSAI, J. S. Coherent control of macroscopic quantum states in a single-Cooper-pair box. **Nature**, v. 398, n. 6730, p. 786–788, 1999. ISSN 00280836.
- NIELSEN, Michael A; CHUANG, Isaac. **Quantum computation and quantum information**. Maryland, EUA: American Association of Physics Teachers, 2010. 10th Anniversary edition published 2010.
- NYLANDER, Andreas. **Fabrication and reliability study of thermoelectric modules, Accelerated aging of SAC solder enhanced with BiTe nano particles**. 2015. Diss. (Mestrado) – Chalmers University of Technology, Gotemburgo, Suécia, 2015.
- O'BRIEN, Jeremy L. Optical quantum computing. **Science**, v. 318, n. 5856, p. 1567–1570, 2007. ISSN 00368075.
- PAESANI, Stefano et al. Experimental Bayesian quantum phase estimation on a silicon photonic chip. **Physical review letters**, APS, v. 118, n. 10, 100503(1–6), 2017.
- PALACIOS-BERRAQUERO, Carmen; MUECK, Leonie; PERSAUD, Divya M. Instead of 'supremacy' use 'quantum advantage'. **Nature**, Nature Publishing Group, v. 576, n. 7786, p. 213–214, 2019.
- PARKER, Jeffrey B.; JOSEPH, Ilon. Quantum phase estimation for a class of generalized eigenvalue problems. **Physical Review A**, v. 102, n. 2, 022422(1–5), 2020. ISSN 24699934.
- PARR, Robert G. Density functional theory of atoms and molecules. In: HORIZONS of quantum chemistry. Switzerland: Springer, 1980. P. 5–15.
- PATRA, Bishnu et al. Cryo-CMOS Circuits and Systems for Quantum Computing Applications. **IEEE Journal of Solid-State Circuits**, IEEE, v. 53, n. 1, p. 309–321, 2018. ISSN 00189200.
- PERUZZO, Alberto et al. A variational eigenvalue solver on a photonic quantum processor. **Nature Communications**, v. 5, May, p. 1–7, 2014. ISSN 20411723.

- PIMENTA, Raphael César de Souza; BEZERRA, Anibal Thiago. Leveraging Qiskit to teach basic quantum computation algorithms. **Parana Journal of Science and Education**, v. 7, n. 6, p. 1–5, 2021.
- PISANI, Cesare. **Quantum-mechanical ab-initio calculation of the properties of crystalline materials**. Alemanha: Springer Science & Business Media, 2012. v. 67.
- PRESKILL, John. Quantum computing 40 years later. **arXiv**, v. 2106.10522, p. 1–49, 2021.
- PRESKILL, John. Quantum computing in the NISQ era and beyond. **Quantum**, v. 2, July, p. 1–20, 2018. ISSN 2521327X.
- ROTHFUSS, Patrick. **O Nome do Vento**. Guarulhos - SP: Arqueiro, 2007. v. 1.
- SCHÄFER, V. M. et al. Fast quantum logic gates with trapped-ion qubits. **Nature**, Nature Publishing Group, v. 555, n. 7694, p. 75–78, 2018. ISSN 14764687.
- SCHOELKOPF, R J; GIRVIN, S M. Wiring up quantum systems. **Nature**, v. 451, February, p. 664–669, 2008.
- SCIENCEDIRECT. **Elsevier**. 1997. Disponível em: <<https://www.sciencedirect.com>>. Acesso em: 12 set. 2021.
- SLATER, John C. A simplification of the Hartree-Fock method. **Physical review**, APS, v. 81, n. 3, p. 385, 1951.
- SLUSSARENKO, Sergei; PRYDE, Geoff J. Photonic quantum information processing: A concise review. **Applied Physics Reviews**, v. 6, n. 4, 041303(1–20), 2019. ISSN 19319401.
- STEFFEN, Matthias et al. State tomography of capacitively shunted phase qubits with high fidelity. **Physical Review Letters**, v. 97, n. 5, p. 4–7, 2006. ISSN 00319007.
- STENTA, Marco; DAL PERARO, Matteo. An introduction to quantum chemical methods applied to drug design. **Front. Biosci.(Elite Ed.)**, v. 3, p. 1061–1078, 2011.
- SUGISAKI, Kenji et al. Quantum Algorithm for the Direct Calculations of Vertical Ionization Energies. **The Journal of Physical Chemistry Letters**, ACS Publications, v. 12, n. 11, p. 2880–2885, 2021.
- SUGISAKI, Kenji et al. Quantum chemistry on quantum computers: quantum simulations of the time evolution of wave functions under the S^2 operator and determination of the spin quantum number S . **Physical Chemistry Chemical Physics**, Royal Society of Chemistry, v. 21, n. 28, p. 15356–15361, 2019.
- SUZUKI, Masuo. General theory of fractal path integrals with applications to many-body theories and statistical physics. **Journal of Mathematical Physics**, v. 32, n. 2, p. 400–407, 1991. ISSN 00222488.

- TRANter, Andrew et al. A Comparison of the Bravyi-Kitaev and Jordan-Wigner Transformations for the Quantum Simulation of Quantum Chemistry. **Journal of Chemical Theory and Computation**, v. 14, n. 11, p. 5617–5630, 2018. ISSN 15499626.
- VAN DIJK, J. P.G. et al. Impact of Classical Control Electronics on Qubit Fidelity. **Physical Review Applied**, American Physical Society, v. 12, n. 4, 044054(1–20), 2019. ISSN 23317019.
- WANG, Chengliang et al. Organic semiconductor crystals. **Chemical Society Reviews**, Royal Society of Chemistry, v. 47, n. 2, p. 422–500, 2018. ISSN 14604744.
- WANG, Ye et al. Single-qubit quantum memory exceeding ten-minute coherence time. **Nature Photonics**, Springer US, v. 11, n. 10, p. 646–650, 2017. ISSN 17494893.
- WEB OF SCIENCE. **Clarivate**. 1997. Disponível em: <<https://www.webofknowledge.com>>. Acesso em: 12 set. 2021.
- WENDIN, G. Quantum information processing with superconducting circuits: A review. **Reports on Progress in Physics**, IOP Publishing, v. 80, n. 10, p. 1–50, 2017. ISSN 00344885.
- WIEBE, Nathan et al. Higher order decompositions of ordered operator exponentials. **Journal of Physics A: Mathematical and Theoretical**, v. 43, n. 6, p. 1–16, 2010. ISSN 17518113.
- WU, Yulin et al. Strong quantum computational advantage using a superconducting quantum processor. **arXiv**, v. 2106.14734, 2021.
- XIA, Rongxin; KAIS, Sabre. Hybrid quantum-classical neural network for calculating ground state energies of molecules. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 22, n. 8, 828(1–12), 2020.

APÊNDICE A – Apêndice A - Circuitos de 1 e 3 qubits para implementação do SSH

Código-fonte 1 – Circuito de 1 qubits para implementação do SSH

```

1  #!/usr/bin/env python
   # coding: utf-8
3
   # In[1]:
5
7  #get_ipython().run_line_magic('matplotlib', 'inline')
   from qiskit import *
9  import scipy.optimize as scp
   from sympy import *
11 import numpy as np
   import matplotlib.pyplot as plt
13 from numpy import linalg as LA
   IBMQ.load_account()
15
17 # In[2]:
19
   def H_decompose(H):
21     '''Performs the Hermitian H 2x2 matrix in terms of the Pauli
       matrices plus the identity
       Inputs:
23         -> H - Hermitian matrix
       Returns:
25         -> ci - four entry list with the decomposition coefficients - ci
           = Tr(sigma_i H) / 2
           ,,,
27     sx = np.array([[0, 1], [ 1, 0]], dtype=np.complex128)
     sy = np.array([[0, -1j], [ 1j, 0]], dtype=np.complex128)
29     sz = np.array([[1, 0], [ 0, -1]], dtype=np.complex128)
     si = np.array([[1, 0], [ 0, 1]], dtype=np.complex128)
31
     S = [sx,sy,sz,si]
33     ci = []
35
     for s in S:
         ci.append( HS( s,H ) / 2. )

```

```

37     return ci
39
40 def HS(M1,M2):
41     return ( np.dot(M1.conjugate().transpose() , M2) ).trace()
42
43
44 def q_Circuit(ang, backend, shots=8192):
45     '''Determines the structure for a single qubit circuit and executes
46     it to obtain the pauli
47     matrices expectation values.
48     Inputs:
49         -> ang - two entry list with the angles used within the ansatz
50         -> backend - "quantum machine"
51         -> shots - number of quantum cycles (default 1024)
52     Returns:
53         -> sig_mean - four entry list with the expectation value of the
54         Pauli matrices and the identity
55     '''
56
57     qr = QuantumRegister(1,'q') # initializes the quantum register
58     cr = ClassicalRegister(1,'c') # initializes the classical register
59
60     #----- x measurement (computer basis)
61     qc = QuantumCircuit(qr, cr, name='circuit')
62     qc = ansatz(qc, qr, ang)
63
64     qc.h(qr) #hadamard
65     qc.measure(qr,cr)
66
67     sigx_mean = runQC(qc, backend, shots)
68
69     #----- z measurement (computer basis)
70     qc = QuantumCircuit(qr, cr, name='circuit')
71     qc = ansatz(qc, qr, ang)
72
73     qc.measure(qr,cr)
74     sigz_mean = runQC(qc, backend, shots)
75
76     #----- y measurement (computer basis)
77     qc = QuantumCircuit(qr, cr, name='circuit')
78     qc = ansatz(qc, qr, ang)
79     qc.sdg(qr)#phase shift
80     qc.h(qr) #hadamard
81     qc.measure(qr,cr)
82
83     sigy_mean = runQC(qc, backend, shots)

```

```

83     #----- identity measurement (computer basis)
      sigId_mean = 1.
85
      sig_mean = [sigx_mean, sigy_mean, sigz_mean, sigId_mean]
87
      return sig_mean
89
91 def runQC(qc, backend, shots):
    '''Executes the quantum circuit in the "quantum machine" defined by
    the backend
93     Inputs:
        -> qc - quantum circuit instance
95         -> backend - "quantum machine"
        -> shots - number of quantum runs
97     Return:
        -> 2p-1 - expectation value in the computational basis after
    several runs
99     '''
    provider = IBMQ.get_provider('ibm-q')
101    qcomp = provider.get_backend('ibmq_armonk')
    exec = execute(qc, backend = qcomp, shots = shots) # executes the
    quantum circuit
103    results = exec.result() # retrieves the results
    counts = results.get_counts() # gets the counts for all the shots
105
    if counts.get('0'): #certifies that the |0> projection is present
107        p = counts.get('0') / shots
    else: # if not the probability is null
109        p = 0.
111
    return 2 * p - 1
113
114 def ansatz(qc, qr, ang):
115     '''General ansatz for the quantum circuit, formed by specific Y and
    Z rotations
    Inputs:
117         -> qc - quantum circuit instance
        -> qr - quantum register to apply the rotations
119         -> ang - two entry list with the Y-rotation angle, and the Z-
    rotation angle
    Returns:
121         -> qc - after the rotations
    '''
123    qc.ry(ang[0], qr) # Y-rotation

```

```

125     qc.rz(ang[1], qr) # Z-rotation
126     return qc
127
128 def H_sigma(ci, sig_mean):
129     ''' Performs the summation of the decomposed Hamiltonian in terms
130     of Pauli matrices
131     Inputs:
132         -> ci - four entry list with the decomposition weights ci = Tr(
133             sigma_i H) / 2
134         -> sig_mean - four entry list with the expectation value of the
135             Pauli matrices and the identity
136     Returns:
137         Real and Imaginary parts of the decomposed Hamiltonian, already
138         weighted and summed
139     '''
140     H_sig_re = 0
141     H_sig_imag = 0
142
143     for i, si in enumerate(sig_mean):
144         H_sig_re += ci[i].real * si.real
145         H_sig_imag += ci[i].imag * si.imag
146
147     return H_sig_re, H_sig_imag
148
149 def decomposed_Hamilt(ang):
150     ''' Function to be optimized
151     Inputs:
152         -> ang - two entry list with the angles used within the ansatz
153     Returns:
154         -> Decomposed hamiltonian expected value
155     '''
156     global sig_mean
157     global ci
158
159     bk = Aer.get_backend('qasm_simulator') # initiates the backend
160
161     sig_mean = q_Circuit(ang, bk) # calls the quantum circuit and
162     executes it
163     ci = H_decompose(H) # decomposes the Hamiltonian in term od Pauli
164     matrices
165
166     creal=[]
167     for c in ci:
168         creal.append(c.real) # casts only the real part

```

```

H_sig, H_sig_imag = H_sigma(creal, sig_mean) # sums the decomposed
Hamiltonian
165
    return H_sig
167
def excited(e0, ang):
169     bk = Aer.get_backend('qasm_simulator') # initiates the backend
    sig_mean = q_Circuit(ang, bk) # calls the quantum circuit and
    executes it
171     ci = H_decompose(H) # decomposes the Hamiltonian in term od Pauli
    matrices

173     e1 = 0
    c_ex = []
175     for i, c in enumerate(ci):
        c_ex = (c - e0 * sig_mean[i] / 2.)
177
        e1 += c_ex * sig_mean[i]
179     return e1

181 def excited_after(e0, H, ang):
    bk = Aer.get_backend('qasm_simulator') # initiates the backend
183     sig_meanp = q_Circuit(ang, bk) # calls the quantum circuit and
    executes it
    cip = H_decompose(H) # decomposes the Hamiltonian in term od Pauli
    matrices
185
    e1 = 0
187     c_ex = []
    for i, c in enumerate(cip):
189         c_ex = (c - e0 * sig_meanp[i] / 2.)

        e1 += c_ex * sig_meanp[i]
191     return e1
193
195 # In[3]:
197
199 #defining the Hamiltonian
    t1 = 0
201     t2 = 1

203 k = np.linspace(-np.pi, np.pi, num=100)
    e0k = e1k = np.zeros(len(k))
205     e0k_np = np.zeros((2, len(k)), dtype=np.complex128)

```

```

angs = []
207 mu = 0

209 ang0 = [1.,2.]

211 for ik, ke in enumerate(k):
    #H = np.array([[0, 2 * t1 * np.cos(ke)], [2 * t1 * np.cos(ke), 0]],
    dtype=np.complex128) - mu * np.eye(2)

213
    h12 = t1 + t2 * ( np.cos(ke) + 1j * np.sin(ke) )
215 h21 = t1 + t2 * ( np.cos(ke) - 1j * np.sin(ke) )
    H = np.array([[0,h12],[h21,0]], dtype=np.complex128) - mu * np.eye
    (2, dtype=np.complex128)

217
    w, v = LA.eig(H)
219 e0_min = scp.minimize(decomposed_Hamilt, ang0, method='COBYLA', tol
    =1e-18)

221
    ang0 = e0_min.x #updadtes the angles with the previous to speed up
    the search
    angs.append(ang0)
223 print('fun = ',e0_min)

225
    e0k[ik] = e0_min.fun
    print(sig_mean, ci)

227
    e1k[ik] = excited(e0k[ik], angs[ik])

229
    print(e0_min.fun)
231 e0k_np[0,ik] = w[0]
    e0k_np[1,ik] = w[1]

233
    #psii = get_psi(e0_min.x)

235

237 # In[ ]:

239
    for ik, ke in enumerate(k):
241 h12 = t1 + t2 * ( np.cos(ke) + 1j * np.sin(ke) )
    h21 = t1 + t2 * ( np.cos(ke) - 1j * np.sin(ke) )
243 H = np.array([[0,h12],[h21,0]], dtype=np.complex128) - mu * np.eye
    (2, dtype=np.complex128)

245
    e1k[ik] = excited_after(e0k[ik], H, angs[ik])

247

```

```

# In[ ]:
249
251 get_ipython().run_line_magic('matplotlib', 'qt')
    font = {'family' : 'Times New Roman',
253         'size' : 40}
    plt.rc('font', **font)
255 plt.plot(k,e0k,'-*', label='')
    plt.plot(k,-e1k,'-d', label='')
257 plt.title('$t_1 = 1, t_2 = 0$')
    plt.ylabel('Energy')
259 plt.xlabel('$k(\pi)$')
261
# In[ ]:

```

Código-fonte 2 – Circuito de 3 qubit para o SSH

```

#!/usr/bin/env python
2 # coding: utf-8
4 # In[ ]:
6
    get_ipython().run_line_magic('matplotlib', 'inline') #ferramenta para
        plotagem dos gráficos
8 from qiskit import * #pacote qiskit para trabalharmos com os circuitos
        e computadores quânticos
    import scipy.optimize as scp #otimizador clássico para o VQE
10 from sympy import * #ferramentas matemáticas para mapeamento do
        hamiltoniano
    import numpy as np #constantes matemáticas, como pi
12 import matplotlib.pyplot as plt #ferramenta para plotagem de gráficos
    from numpy import linalg as LA #ferramentas matemáticas para mapeamento
        do hamiltoniano
14 IBMQ.load_account() #login na conta da IBM para acessar os computadores
        quânticos
    provider = IBMQ.get_provider('ibm-q') #definindo a IBM como provedor
        dos computadores
16
18 # In[ ]:
20
    def H_decompose(H):
22         '''Performs the Hermitian H 2x2 matrix in terms of the Pauli
            matrices plus the identity
            Inputs:

```

```

24         -> H - Hermitian matrix
Returns:
26         -> ci - four entry list with the decomposition coefficients - ci
= Tr(sigma_i H) / 2
'''
28     sx = np.array([[0, 1], [ 1, 0]], dtype=np.complex128)
sy = np.array([[0, -1j], [ 1j, 0]], dtype=np.complex128)
30     sz = np.array([[1, 0], [ 0, -1]], dtype=np.complex128)
si = np.array([[1, 0], [ 0, 1]], dtype=np.complex128)
32
S = [sx,sy,sz,si]
34     ci = []

36     for s in S:
        ci.append( HS( s,H ) / 2. )
38
return ci
40
def HS(M1,M2):
42     return ( np.dot(M1.conjugate().transpose() , M2) ).trace()
44
def q_Circuit(ang, backend, shots=5120):
46     '''Determines the structure for a single qubit circuit and executes
it to obtain the pauli
matrices expectation values.
48     Inputs:
        -> ang - two entry list with the angles used within the ansatz
50        -> backend - "quantum machine"
        -> shots - number of quantum cycles (default 1024)
52     Returns:
        -> sig_mean - four entry list with the expectation value of the
Pauli matrices and the identity
54     '''

56     qr = QuantumRegister(3,'q') # initializes the quantum register
cr = ClassicalRegister(3,'c') # initializes the classical register
58     qc = QuantumCircuit(qr, cr, name='circuit')
#----- x measurement (computer basis)
60     #qc = QuantumCircuit(qr, cr, name='circuit')
qc = ansatz(qc, qr[0], ang)
62
qc.h(qr[0]) #hadamard
64     qc.measure(qr[0],cr[0])

66     #sigx_mean = runQC(qc, backend, shots)

```

```

68      #----- z measurement (computer basis)
        #qc = QuantumCircuit(qr, cr, name='circuit')
70      qc = ansatz(qc, qr[2], ang)

72      qc.measure(qr[2], cr[2])
        #sigz_mean = runQC(qc, backend, shots)

74      #----- y measurement (computer basis)
76      #qc = QuantumCircuit(qr, cr, name='circuit')
        qc = ansatz(qc, qr[1], ang)
78      qc.sdg(qr[1])#phase shift
        qc.h(qr[1]) #hadamard
80      qc.measure(qr[1], cr[1])

82      sig_mean = runQC(qc, backend, shots)

84      #----- identity measurement (computer basis)
        sigId_mean = 1.

86      #sig_mean = [sigx_mean, sigy_mean, sigz_mean, sigId_mean]
88      sigx_mean = sig_mean[0]
        sigy_mean = sig_mean[1]
90      sigz_mean = sig_mean[2]
        #print(qc)
92      #print (sig_mean)
        #print(qc)
94      return sigx_mean, sigy_mean, sigz_mean, sigId_mean

96      #return sig_mean

98 def runQC(qc, backend, shots):
    global counts
100    '''Executes the quantum circuit in the "quantum machine" defined by
        the backend
        Inputs:
102        -> qc - quantum circuit instance
            -> backend - "quantum machine"
104        -> shots - number of quantum runs
        Return:
106        -> 2p-1 - expectation value in the computational basis after
        several runs
        '''
108    #provider = IBMQ.get_provider('ibm-q')
        #qcomp = provider.get_backend('ibmq_16_melbourne')
110    #count = []

    exec = execute(qc, backend = qcomp, shots = shots) # executs the
        quantum circuit

```

```

112     results = exec.result() # retrieves the results
113     counts = results.get_counts() # gets the counts for all the shots
114
115     px = ['000', '010', '100', '110']
116     py = ['000', '100', '001', '101']
117     pz = ['000', '010', '001', '011']
118
119     ppx = 0
120     for mx in px:
121         if counts.get(mx):
122             ppx += counts.get(mx)/shots
123         else:
124             ppx += 0
125
126     ppy = 0
127     for my in py:
128         if counts.get(my):
129             ppy += counts.get(my)/shots
130         else:
131             ppy += 0
132
133     ppz = 0
134     for mz in pz:
135         if counts.get(mz):
136             ppz += counts.get(mz)/shots
137         else:
138             ppz += 0
139
140     return (2 * ppx - 1), (2 * ppy - 1), (2 * ppz - 1)
141     #return 2 * ppy - 1
142     #return 2 * ppz - 1
143
144 def ansatz(qc, qr, ang):
145     '''General ansatz for the quantum circuit, formed by specific Y and
146     Z rotations
147     Inputs:
148         -> qc - quantum circuit instance
149         -> qr - quantum register to apply the rotations
150         -> ang - two entry list with the Y-rotation angle, and the Z-
151         rotation angle
152     Returns:
153         -> qc - after the rotations
154     '''
155     qc.ry(ang[0], qr) # Y-rotation
156     qc.rz(ang[1], qr) # Z-rotation
157     return qc

```

```

158
160 def H_sigma(ci, sig_mean):
161     ''' Performs the summation of the decomposed Hamiltonian in terms
162     os Pauli matrices
163     Inputs:
164         -> ci - four entry list with the decomposition weights ci = Tr(
165         sigma_i H) / 2
166         -> sig_mean - four entry list with the expectation value of the
167         Pauli matrices and the identity
168     Returns:
169         Real and Imaginary parts of the decomposed Hamiltonian, already
170         weighted and summed
171     '''
172     H_sig_re = 0
173     H_sig_imag = 0
174
175     for i, si in enumerate(sig_mean):
176         H_sig_re += ci[i].real * si.real
177         H_sig_imag += ci[i].imag * si.imag
178
179     return H_sig_re, H_sig_imag
180
181 def decomposed_Hamilt(ang):
182     ''' Function to be optimized
183     Inputs:
184         -> ang - two entry list with the angles used within the ansatz
185     Returns:
186         -> Decomposed hamiltonian expected value
187     '''
188     global sig_mean
189     global ci
190
191     bk = Aer.get_backend('qasm_simulator') # initiates the backend
192
193     sig_mean = q_Circuit(ang, bk) # calls the quantum circuit and
194     executes it
195
196     ci = H_decompose(H) # decomposes the Hamiltonian in term od Pauli
197     matrices
198
199     creal=[]
200     for c in ci:
201         creal.append(c.real) # casts only the real part
202
203     H_sig, H_sig_imag = H_sigma(creal, sig_mean) # sums the decomposed
204     Hamiltonian

```

```

198     return H_sig
200 def excited(e0, ang):
201     bk = Aer.get_backend('qasm_simulator') # initiates the backend
202     sig_mean = q_Circuit(ang, bk) # calls the quantum circuit and
    executes it
203     ci = H_decompose(H) # decomposes the Hamiltonian in term od Pauli
    matrices
204
205     e1 = 0
206     c_ex = []
207     for i, c in enumerate(ci):
208         c_ex = (c - e0 * sig_mean[i] / 2.)
209
210         e1 += c_ex * sig_mean[i]
211     return e1
212
213 def excited_after(e0, H, ang):
214     bk = Aer.get_backend('qasm_simulator') # initiates the backend
215     sig_meanp = q_Circuit(ang, bk) # calls the quantum circuit and
    executes it
216     cip = H_decompose(H) # decomposes the Hamiltonian in term od Pauli
    matrices
217
218     e1 = 0
219     c_ex = []
220     for i, c in enumerate(cip):
221         c_ex = (c - e0 * sig_meanp[i] / 2.)
222
223         e1 += c_ex * sig_meanp[i]
224     return e1
225
226
227
228 # In[ ]:
229
230
231 #defining the Hamiltonian
232 t1 = 1
233 t2 = 0
234
235 k = np.linspace(-np.pi, np.pi, num=5)
236 e0k = e1k = np.zeros(len(k))
237 e0k_np = np.zeros((2, len(k)), dtype=np.complex128)
238 ang = []
239 mu = 0

```

```

240 ang0 = [1.,2.]
242
244 for ik, ke in enumerate(k):
246     h12 = t1 + t2 * ( np.cos(ke) + 1j * np.sin(ke) )
246     h21 = t1 + t2 * ( np.cos(ke) - 1j * np.sin(ke) )
246     H = np.array([[0,h12],[h21,0]], dtype=np.complex128) - mu * np.eye
246     (2, dtype=np.complex128)
248
248     w, v = LA.eig(H)
250     e0_min = scp.minimize(decomposed_Hamilt, ang0, method='COBYLA', tol
250     =1e-18)
252
252     ang0 = e0_min.x #updatdes the angles with the previous to speed up
252     the search
252     ang0.append(ang0)
254     print('fun = ',e0_min)
256
256     e0k[ik] = e0_min.fun
256     print(sig_mean, ci)
258
258     e1k[ik] = excited(e0k[ik], ang0[ik])
260
260     print(e0_min.fun)
262     e0k_np[0,ik] = w[0]
262     e0k_np[1,ik] = w[1]
264
264     #psii = get_psi(e0_min.x)
266
268 # In[ ]:
270
270 for ik, ke in enumerate(k):
272     h12 = t1 + t2 * ( np.cos(ke) + 1j * np.sin(ke) )
272     h21 = t1 + t2 * ( np.cos(ke) - 1j * np.sin(ke) )
274     H = np.array([[0,h12],[h21,0]], dtype=np.complex128) - mu * np.eye
274     (2, dtype=np.complex128)
276
276     e1k[ik] = excited_after(e0k[ik], H, ang0[ik])
278
278 # In[ ]:
280
282 get_ipython().run_line_magic('matplotlib', 'qt')

```

```
font = {'family' : 'Times New Roman',  
284      'size'    : 40}  
plt.rc('font', **font)  
286 plt.plot(k,e0k,'-*', label='')  
plt.plot(k,-e1k,'-d', label='')  
288 plt.title('$t_1 = 1, t_2 = 0$')  
plt.ylabel('Energy')  
290 plt.xlabel('$k(\pi)$')
```