



DESENVOLVIMENTO DE UM MÓDULO DE QUÍMICA COMPUTACIONAL QUÂNTICA PARA A PLATAFORMA KET

Maria Eduarda W. M. Vianna¹, José Carlos Libois Neto¹, Amanda M. de Lima², Evandro C. R. Rosa¹, Eduardo I. Duzzioni¹, Erico S. Teixeira³

¹*Universidade Federal de Santa Catarina*

²*Universidade Federal de Pernambuco*

³*Venturus Centro de Inovação Tecnológica*

E-mail: mewmvianna@gmail.com

Palavras-Chave: estrutura eletrônica, segunda quantização, mapeamento férmion-qubit

Introdução

A Química Quântica busca compreender a estrutura eletrônica e as propriedades de átomos e moléculas com base nas leis da mecânica quântica. Entretanto, a obtenção de soluções exatas para a equação de Schrödinger em sistemas complexos representa um desafio computacional. À medida que o número de elétrons e de orbitais aumenta, o número de configurações eletrônicas possíveis cresce significativamente, tornando a dimensão do espaço de Hilbert intratável para computadores clássicos (THOGERSEN; OLSEN, 2004).

A busca por alternativas à simulação de sistemas quânticos impulsionou o desenvolvimento da computação quântica. Em 1982, Richard Feynman propôs o uso de sistemas quânticos controláveis para simular outros sistemas quânticos, destacando as limitações dos computadores clássicos para essa tarefa (FEYNMAN, 1982). Ao explorar propriedades fundamentais como a superposição, o emaranhamento e a interferência, os processadores quânticos permitem o mapeamento direto e a manipulação eficiente de estados quânticos. No contexto da química computacional, essa mudança de paradigma motivou a criação de algoritmos voltados à simulação de sistemas moleculares e à determinação de suas propriedades eletrônicas.

Atualmente, os computadores quânticos encontram-se predominantemente na era denominada NISQ (Noisy Intermediate-Scale Quantum), caracterizada por dispositivos com um número moderado de qubits, mas ainda sujeitos a ruídos e limitações que dificultam a implementação de algoritmos quânticos de grande profundidade (PRESKILL, 2018). Diante dessas limitações, os algoritmos híbridos clássico-quânticos surgem como uma abordagem relevante. Entre eles, destaca-se o Variational Quantum Eigensolver (VQE), que combina circuitos quânticos parametrizados com otimizadores clássicos para estimar propriedades de



sistemas quânticos, como a energia do estado fundamental de moléculas (PERUZZO et al., 2014). Nesse algoritmo, o processador quântico é utilizado para preparar e medir estados quânticos, enquanto um computador clássico ajusta iterativamente os parâmetros do circuito com base nos resultados obtidos na etapa quântica.

A exploração dessas aplicações depende, entretanto, de ferramentas de software capazes de integrar os formalismos da química quântica e da teoria de muitos corpos à programação de computadores quânticos. Atualmente, o ecossistema global é dominado por bibliotecas internacionais, muitas delas impulsionadas por gigantes tecnológicas e consórcios estrangeiros, como Qiskit (IBM), PennyLane (Xanadu) e OpenFermion (Google). Nesse cenário, o desenvolvimento de uma infraestrutura nacional de software torna-se estratégico para a soberania tecnológica brasileira. Dominar essas camadas fundamentais da pilha de computação reduz a dependência externa e assegura a autonomia científica das instituições nacionais. Nesse contexto, plataformas brasileiras de código aberto, como a linguagem Ket (ROSA; SANTIAGO, 2021), desempenham um papel crucial, fornecendo a base de programação necessária para consolidar um ecossistema nativo de pesquisa, desenvolvimento e inovação em tecnologias quânticas.

Diante disso, este trabalho apresenta o desenvolvimento inicial de um módulo de Química Computacional Quântica integrado à plataforma Ket. Essa nova camada fornece ferramentas para a representação e manipulação de operadores fermiônicos, possibilitando sua utilização em aplicações de química computacional e em outros problemas que envolvam sistemas de férmions. Até o momento, foram implementados componentes para a representação de férmions em segunda quantização, utilizando operadores de criação e aniquilação, além de rotinas de transformação responsáveis por mapear a álgebra fermiônica em operadores de qubits, amplamente estabelecidos na literatura de química quântica (CAO et al., 2019; MCARDLE et al., 2020). Entre os mapeamentos contemplados estão os métodos de Jordan-Wigner (JORDAN; WIGNER, 1928), Paridade (SEELEY; RICHARD; LOVE, 2012), Bravyi-Kitaev (BRAVYI; KITAEV, 2002) e Bravyi-Kitaev com redução de simetria (BRAVYI et al., 2017). Paralelamente, desenvolveu-se uma interface de integração com o solver clássico PySCF (SUN et al., 2018) para a extração de dados orbitais e geração do Hamiltoniano eletrônico molecular. Dessa forma, o projeto expande as capacidades do ecossistema Ket, estabelecendo uma ponte funcional entre os métodos tradicionais de estrutura eletrônica e o paradigma da computação quântica.

Material e Métodos

O módulo foi desenvolvido em Python (≥ 3.10) integrado ao Ket, com simulações quânticas executadas pelo simulador KBW (escrito em Rust). Sua implementação é organizada nos seguintes componentes principais:

1. `ket.fermion`: provê a representação formal e álgebra de operadores fermiônicos.

2. `ket.pyscf_interface`: processa a especificação molecular (geometria, base, carga e multiplicidade) via PySCF, extraíndo energia nuclear e integrais eletrônicas.
3. `ket.mapping`: mapeamentos que convertem operadores fermiônicos em combinações lineares de cadeias de Pauli.

O fluxo do pipeline (da molécula ao Hamiltoniano de qubits) é ilustrado na Figura 1, e código está disponível no GitLab: <https://gitlab.com/quantum-ket/ket>.

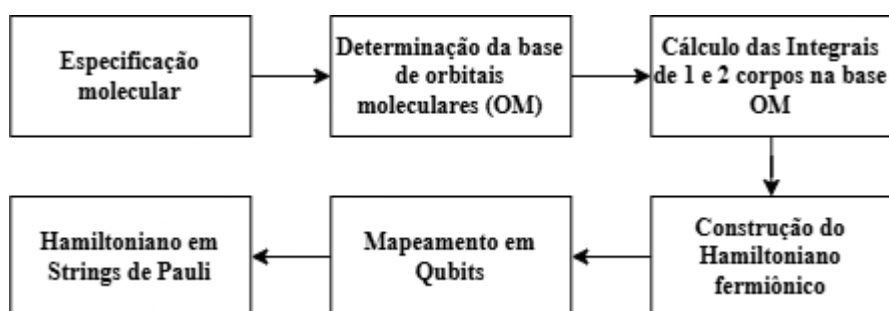


Figura 1: Pipeline do módulo de química quântica do Ket, da especificação molecular ao Hamiltoniano de qubits.

O Ket disponibiliza quatro mapeamentos. O Jordan-Wigner (JORDAN; WIGNER, 1928), em que cada qubit armazena diretamente a ocupação de um orbital fermiônico; a regra de anticomutação é preservada aplicando uma cadeia de operadores Z em todos os orbitais precedentes, enquanto a criação ou aniquilação no orbital de interesse é realizada por operadores X e Y. No mapeamento de Paridade (SEELEY; RICHARD; LOVE, 2012), cada qubit armazena a paridade acumulada da ocupação de todos os orbitais até aquele índice. Em consequência, a criação ou aniquilação no orbital de interesse combina operadores Z para extrair a paridade dos orbitais anteriores com uma cadeia de operadores X aplicada aos qubits subsequentes.

O mapeamento de Bravyi-Kitaev abrange a formulação padrão (BRAVYI; KITAEV, 2002; HAV LÍČEK et al., 2017) e a extensão com conservação de simetria (SCBK) via remoção de qubits (*tapering off*) (BRAVYI et al., 2017; MCARDLE et al., 2020), ambas estruturadas sobre árvores de Fenwick. O BK padrão admite duas abordagens: a convencional, executada por operações bitwise, e outra via árvore adaptativa alocada explicitamente em memória. Embora a via bitwise seja a mais comum na literatura, o Ket adota a árvore adaptativa como padrão tanto para o BK quanto para o SCBK, preservando a versão bitwise como opção legada. No SCBK, a exploração das simetrias $\mathbb{Z}_2$ de paridade de spin possibilita a redução de dois qubits na representação do sistema.

A validação do módulo fermiônico consistiu em testes unitários para verificação de operadores, propriedades de spin e simetrias físicas, acompanhados de checagens algébricas e testes cruzados dos mapeamentos com o OpenFermion. O pipeline completo foi avaliado integrando Hartree-Fock (PySCF) e mapeamento Jordan-Wigner, confirmando concordância



com o OpenFermion para as moléculas de H_2 , LiH , BeH_2 e H_2O , conjunto escolhido por abranger diferentes escalas de tamanho de base, número de elétrons e complexidade de correlação.

Resultados e Discussão

O componente `ket.fermion` fornece a infraestrutura simbólica para representação e manipulação de operadores em segunda quantização. Entre as principais funcionalidades implementadas, destacam-se:

- Composição de produtos e combinações lineares: permite representar produtos de operadores de criação e aniquilação, como $a^+_{0a_1}$, e construir combinações lineares com coeficientes reais ou complexos, como $1,5a^+_{0a_0} + (0,2 + 0,5i)a^+_{0a_1}$.
- Simplificação numérica: remove termos cujos coeficientes apresentam magnitude inferior à tolerância definida. Para $tol = 10^{-8}$, por exemplo, $0,5a^+_{0a_1} + 10^{-10}a^+_{1a_0}$ é reduzido a $0,5a^+_{0a_1}$.
- Ordenação normal: reorganiza os operadores de modo que os de criação precedam os de aniquilação, aplicando as relações de anticomutação fermiônicas necessárias. Por exemplo, $a_{1a}a^+_1$ é transformado em $I - a^+_1a_1$. Em expressões complexas envolvendo múltiplos orbitais, os índices dos orbitais são ordenados de forma decrescente, do maior para o menor, enquanto termos semelhantes são agrupados e coeficientes inferiores à tolerância são automaticamente eliminados, garantindo uma representação canônica única.
- Verificação da conservação do número de partículas: verifica se cada termo possui quantidade igual de operadores de criação e aniquilação. Assim, $a^+_{0a_1}$ é classificado como conservador, enquanto $a^+_{0a_1}$ não conserva o número total de partículas.
- Verificação da conservação de S^z : avalia separadamente os balanços de criação e aniquilação dos operadores associados aos spins α e β , permitindo verificar a conservação da componente z do spin.
- Validação de operadores de até dois corpos: verifica se cada termo contém no máximo quatro operadores fermiônicos, correspondentes à identidade, a termos de um corpo (como $a^+_{0a_1}$) ou de dois corpos (como $a^+_{0a_1}a_{3a_2}$), e se o operador conserva o número de partículas.

Para avaliar o comportamento prático dos mapeamentos implementados, analisou-se a transformação de um mesmo operador fermiônico, correspondente ao termo de acoplamento entre os spin-orbitais não vizinhos 3 e 7, $a^+_{3a_7}$, responsável por aniquilar um elétron no orbital 7 (inicialmente ocupado) e criá-lo no orbital 3 (inicialmente vazio), em um sistema com $N = 8$ spin-orbitais e 4 elétrons, com $n_\alpha = n_\beta = 2$. A aplicação dos quatro mapeamentos evidencia diferenças na estrutura das cadeias de operadores de Pauli e, no caso do SCBK, no número de qubits utilizado. Segue a descrição de cada um dos mapeamentos:

- Jordan-Wigner (JW):
 - 8 qubits necessários



- $JW(a^+_{3a7}) = -0.25j * Y(3) Z(4) Z(5) Z(6) X(7) + (0.25-0j) * Y(3) Z(4) Z(5) Z(6) Y(7) + (0.25+0j) * X(3) Z(4) Z(5) Z(6) X(7) + 0.25j * X(3) Z(4) Z(5) Z(6) Y(7)$
- Neste mapeamento, cada qubit corresponde diretamente a um spin-orbital (exigindo os 8 qubits). Como os orbitais 3 e 7 não são adjacentes, a transformação introduz a cadeia de operadores de fase $Z_4Z_5Z_6$ nos qubits intermediários, fazendo com que cada termo resultante atue sobre 5 qubits simultaneamente.
- Paridade
 - 8 qubits necessários
 - $Parity(a^+_{3a7}) = -0.25j * Z(2) X(3) X(4) X(5) Y(6) + (-0.25+0j) * Z(2) X(3) X(4) X(5) X(6) Z(7) + (-0.25+0j) * Y(3) X(4) X(5) Y(6) + 0.25j * Y(3) X(4) X(5) X(6) Z(7)$
 - A transformação armazena a paridade acumulada da ocupação dos orbitais anteriores. Como ainda representa a totalidade do espaço de estados sem redução de simetria, são necessários os mesmos 8 qubits, gerando operadores que atuam nos qubits de 2 a 7.
- Bravyi-Kitaev
 - 8 qubits necessários
 - $BK(a^+_{3a7}) = -0.25j * Z(1) Z(2) Y(3) Z(5) Z(6) + (-0.25+0j) * Z(1) Z(2) X(3) Z(7) + (0.25+0j) * X(3) Z(5) Z(6) + 0.25j * Y(3) Z(7)$
 - O mapeamento BK também necessita de 8 qubits para representar a totalidade dos spin-orbitais. No entanto, ao distribuir a informação por meio de uma árvore de Fenwick, em vez de uma cadeia linear, reduz-se a necessidade de sequências contínuas de operadores Z nos qubits intermediários, resultando, em geral, em termos que atuam simultaneamente sobre um número menor de qubits.
- Bravyi-Kitaev com simetria
 - 6 qubits necessários
 - $SCBK(a^+_{3a7}) = -0.25j * Z(3) Y(4) Z(5) + (-0.25+0j) * Z(3) X(4) + (0.25+0j) * X(4) Z(5) + 0.25j * Y(4)$
 - O método SCBK explora a conservação do número de partículas em cada setor de spin ($n_\alpha=2$ e $n_\beta=2$). Como as paridades totais de cada setor de spin são constantes e conhecidas a priori, os 2 qubits associados a essas simetrias podem ser eliminados matematicamente (tapering off). Isso reduz o registrador de 8 para 6 qubits e simplifica os operadores resultantes.

Tais mapeamentos foram confrontados com implementações de referência. Para os mapeamentos Jordan-Wigner, Bravyi-Kitaev e SCBK, os resultados foram comparados com as implementações correspondentes da biblioteca OpenFermion, enquanto o mapeamento por Paridade foi validado em comparação com a biblioteca PennyLane. Para o operador a^+_{3a7} , as



representações obtidas apresentaram concordância nos coeficientes e na estrutura algébrica, considerando as respectivas convenções de ordenação dos termos. No caso do SCBK, ambas as implementações resultaram na redução do registrador de oito para seis qubits.

A validação dos coeficientes do Hamiltoniano gerados pelo Ket foi realizada por comparação direta com os resultados da biblioteca OpenFermion para as moléculas de H_2 , LiH , BeH_2 e H_2O na base STO-3G após o mapeamento Jordan-Wigner. Após a transformação fermiônica, cada Hamiltoniano é representado no espaço de qubits como uma combinação linear de operadores de Pauli, $H = \sum c_k P_k$, em que P_k representa uma cadeia de Pauli e c_k seu respectivo coeficiente. A divergência termo a termo avalia a diferença absoluta entre os coeficientes calculados por ambas as bibliotecas, $|c_k^{Ket} - c_k^{Openfermion}|$, enquanto a divergência máxima (Δ_{max}) corresponde ao maior desvio absoluto observado entre todos os termos que compõem o Hamiltoniano: $\Delta_{max} = \max_k |c_k^{Ket} - c_k^{Openfermion}|$.

A Tabela 1 apresenta o número de qubits e a divergência absoluta máxima entre os coeficientes calculados por ambas as bibliotecas. Em todos os casos, a divergência manteve-se na ordem de 10^{-16} a 10^{-14} , valores atribuídos ao erro computacional decorrente da precisão numérica das operações em ponto flutuante.

Molécula	Número de qubits	Divergência Máxima
H_2	4	$1,80 \times 10^{-16}$
LiH	12	$2,00 \times 10^{-14}$
BeH_2	14	$1,60 \times 10^{-14}$
H_2O	14	$3,55 \times 10^{-14}$

Tabela 1: Comparação dos coeficientes dos Hamiltonianos obtidos pelo Ket e pelo OpenFermion

Conclusões

O módulo de química quântica em desenvolvimento para a plataforma Ket estabeleceu com sucesso a infraestrutura fundamental para a simulação de sistemas moleculares em registradores quânticos. As validações realizadas confirmam a consistência das transformações algébricas e a perfeita concordância dos Hamiltonianos de qubits gerados em relação às bibliotecas de referência. Essas entregas consolidam a base para o ecossistema Ket e representam um passo estratégico na construção de uma infraestrutura aberta, acessível e



nacional, fortalecendo a autonomia tecnológica brasileira no desenvolvimento de software para computação quântica.

Como etapas futuras, pretende-se expandir a integração com o PySCF para contemplar o método Unrestricted Hartree-Fock (UHF), permitindo o tratamento de matrizes de coeficientes distintas para os spins α e β ($C_\alpha \neq C_\beta$), e a teoria de perturbação de Møller–Plesset de segunda ordem (MP2), incorporando efeitos de correlação eletrônica pós-Hartree-Fock. Adicionalmente, está planejada a integração do algoritmo Variational Quantum Eigensolver (VQE), cujos resultados com o pipeline completo serão apresentados durante o evento.

Agradecimentos

Os autores agradecem o apoio do INCT em Computação Quântica Aplicada (CNPq nº 408884/2024-0). E.S.T. agradece ao Venturus e seu Centro de Excelência em Computação Quântica.

Referências

- BRAVYI, S. B.; KITAEV, A. Y. Fermionic quantum computation. *Annals of Physics*, v. 298, p. 210–226, 2002.
- BRAVYI, S.; GAMBETTA, J. M.; MEZZACAPO, A.; TEMME, K. Tapering off qubits to simulate fermionic Hamiltonians. arXiv:1701.08213, 2017.
- CAO, Y. et al. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, v. 119, n. 19, p. 10856–10915, 2019. DOI: 10.1021/acs.chemrev.8b00803.
- DA ROSA, Evandro Chagas Ribeiro; DE SANTIAGO, Rafael. Ket Quantum Programming. *ACM Journal on Emerging Technologies in Computing Systems*, v. 18, n. 1, art. 12, p. 1–25, 2021. DOI: 10.1145/3474224.
- FEYNMAN, R. P. Simulating physics with computers. *International Journal of Theoretical Physics*, v. 21, n. 6-7, p. 467–488, 1982.
- HAV LÍČEK, V. et al. Operator locality in the Bravyi-Kitaev transformation. *Physical Review A*, v. 96, n. 6, art. 062332, 2017. DOI: 10.1103/PhysRevA.96.062332.
- JORDAN, P.; WIGNER, E. Über das Paulische Äquivalenzverbot. *Zeitschrift für Physik*, v. 47, p. 631–651, 1928.
- MCARDLE, S. et al. Quantum computational chemistry. *Reviews of Modern Physics*, v. 92, n. 1, art. 015003, 2020. DOI: 10.1103/RevModPhys.92.015003.
- PERUZZO, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, v. 5, art. 4213, 2014.
- PRESKILL, J. Quantum computing in the NISQ era and beyond. *Quantum*, v. 2, p. 79, 2018.
- SEELEY, J. T.; RICHARD, M. J.; LOVE, P. J. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, v. 137, art. 224109, 2012.
- SUN, Q. et al. PySCF: the Python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, v. 8, e1340, 2018. DOI: 10.1002/wcms.1340.
- THOGERSEN, L.; OLSEN, J. A coupled cluster and full configuration interaction study of CN and CN⁻. *Chemical Physics Letters*, v. 393, n. 1-3, p. 36–43, 2004.