Approximate Fermionic Hamiltonian Simulation for H-chains

Louis Marquis

November 27, 2024

1 Introduction

One of the most promising applications of quantum computing is in the field of quantum computational chemistry, which aims to efficiently simulate the behavior (i.e. Hamiltonian evolution) of chemical systems that are too complex for classical computers. The ability to computationally predict behavior of molecular systems would significantly benefit drug discovery efforts. This paper presents a sub-quadratic (in n) quantum algorithm to simulate the Hamiltonian evolution of the hydrogen chain, a benchmark chemical system in this field.

2 Algorithm

Denote n as the number of hydrogen atoms. Since we're only dealing with 1s orbitals, this is also the number of orbitals. The electronic Hamiltonian has a form that can be separated into one-electron and two-electron components.

$$H = H_1 + H_2 = \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{i,j,k,l} a_i^\dagger a_j^\dagger a_k a_l \tag{1}$$

It is always the case that H is symmetric between i and j, between k and l, and between ij and kl. This means that the matrix $h_{i,j,k,l}$ can be eigendecomposed with eigenvalues $\{\lambda_r\}_{r\in[n^2]}$ (in descending order) and orthonormal eigenvectors $\{|r\rangle\}_{r\in[n^2]}$. Denote $L^r = \lambda_r |r\rangle \langle r|$, which is symmetric.

$$H_2 = \frac{1}{2} \sum_{r\in[n^2]} (\sum_{i,j\in[n]} L^r_{ij} a_i^\dagger a_j)^2 \tag{2}$$

In the case of the H-chain, numerical results show that there is a significant drop between the λ_n and λ_{n+1}. That is, eigenvalues $n+1$ and above are significantly smaller than the first n eigenvalues. In addition, for the H-chain, the $L^1, ..., L^r$ are all approximately diagonal.

$$H_2 \approx \frac{1}{2} \sum_{r\in[n]} (\sum_{i\in[n]} L^r_{ii} a_i^\dagger a_i)^2 = \frac{1}{2} \sum_{r\in[n]} (\sum_{i\in[n]} L^r_{ii} n_i)^2 \tag{3}$$

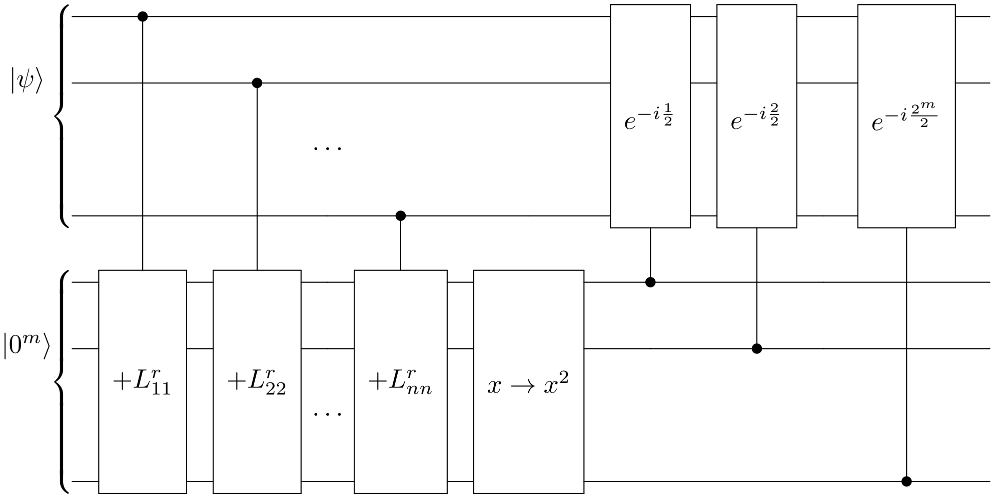
The goal in Hamiltonian simulation is to approximately apply the operation e^{-iHt} to an arbitrary initial state $|\psi\rangle$. If H is a sum of Hermitian operations, then one can Trotterize the Hamiltonian.

$$e^{-iHt} = (e^{-iH_1 \frac{t}{M}} e^{-iH_2 \frac{t}{M}})^M + O(\frac{t^2}{M}) \tag{4}$$

H_1 is made of n^2 terms, which can be grouped into $O(n^2)$ Hermitian terms, so $e^{-iH_1 \frac{t}{M}}$ can be implemented in $O(n^2 \frac{t}{M})$ complexity. This is not the focus of this project, but rather the more complex H_2.

$$e^{-iH_2 \frac{t}{M}} \approx e^{-\frac{it}{2M} \sum_{r\in[n]} (\sum_{i\in[n]} L^r_{ii} n_i)^2} \tag{5}$$

We see that the eigenstates of H_2 are $|x_1, ..., x_n\rangle$ and their eigenvalues are a function $\frac{1}{2}\sum_{r\in[n]}(\sum_{i\in[n]} L^r_{ii}x_i)^2$ of $x_1, ..., x_n$. Therefore, H_2 is approximately the same as rotating each eigenstate at a particular frequency that is constant over time. To achieve this, the frequency can be calculated with quantum arithmetic on a separate register of qubits and used to control the rotation.



For each i, $+L_{ii}$ can be implemented via the QFT-based adder in [2]. Since $L_i i$ is classical, the controls in Fig 2 of [2] are unnecessary and whether each gate instead appears (or doesn't) can be determined classically. Instead, each gate has a control on the ith qubit of $|\psi\rangle$. Therefore, the complexity of such an adder is the same as that in the paper, which is $O(m^2)$. Having n adders results in $O(nm^2)$.

The squaring operation can be implemented via either the QFT-based multiplier in [2] or the method in [1]. The former has $O(m^3)$ complexity. The latter is more appropriate for larger m values, which may exceed the range that this algorithm demands. So we will assume the former for now. Note that to adapt to our use (squaring), one can copy the ancilla to a different ancilla register and apply the multiplier to a third ancilla register. This requires $2m$ additional ancilla bits. The strategy may be improved, but that's not the focus right now.

It is assumed that a controlled phase shift gate can be implemented efficiently. In this case, the total complexity of the phase shifts at the end of the circuit is $O(nm)$, as there is one gate per ancilla qubit and per state qubit.

Such a quantum circuit can be used to apply $e^{-(\sum_{i\in[n]} L^r_{ii}n_i)^2}$ in $O(nm^2+m^3)$, which commutes for all r. Therefore, $e^{-\sum_{r\in[n]}(\sum_{i\in[n]} L^r_{ii}n_i)^2}$ can be implemented in $O(n^2m^2 + nm^3)$.

References

[1] Gregory D. Kahanamoku-Meyer and Norman Y. Yao. *Fast quantum integer multiplication with zero ancillas.* 2024. arXiv: 2403.18006 [quant-ph]. URL: https://arxiv.org/abs/2403.18006.

[2] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. "Quantum arithmetic with the quantum Fourier transform". In: *Quantum Information Processing* 16.6 (Apr. 2017). ISSN: 1573-1332. DOI: 10.1007/s11128-017-1603-1. URL: http://dx.doi.org/10.1007/s11128-017-1603-1.