Simulating lithium-ion batteries on quantum computers

Gopal Ramesh Dahale

dahalegopal27@gmail.com,

Abstract. One of the frontiers for the Noisy Intermediate-Scale Quantum (NISQ) era's practical uses of quantum computers is quantum chemistry. By employing Hybrid Quantum Classical Optimization we aim to investigate the potential of quantum computing for simulating lithium-ion batteries. We implement quantum algorithms that efficiently calculates the ground state energy of materials used in batteries. We use a variant of Variational Quantum Eigensolver (VQE) known as Contextual Subspace VQE (CS-VQE) and focus on exploring the power of hardware-efficient ansatzes to improve the convergence of finding the ground state energy of the materials. Furthermore, techniques like ADAPT-VQE and Rotoselect algorithms are employed to solve for the contextual subspace hamiltonians, hence, presenting an extension to the CS-VQE algorithm. We use these techniques to simulate a realistic cathode material: dilithium iron silicate and were able to reduce the number of qubits required in the algorithm by more than half, proving to be suitable for the ISQ era. Our implementation leverages NVIDIA's GPU and CUDA Quantum Platform which enable us to perform faster simulations. The findings from our study demonstrates the potential of quantum computing to effectively perform quantum simulations of lithium-ion batteries. The code can be found here.

Keywords: VQE · CS-VQE · Rotoselect · ADAPT-VQE · HEA · Lithium-ion Battery

1 Introduction

Variational Quantum Eigensolver (VQE) is one of the successful quantum algorithm for calculating the ground state energy of materials and molecules. However, the choice of the ansatz used to approximate the ground state is important in determining a close upper bound to the actual energy. There exists problem inspired ansatzes like UCCSD that utilizes the excitations present within the electronic structure of the molecules. As we know, these problem inspired ansatzes use multi-qubit gates and are high depth circuits. These are not suitable for the quantum hardware and infeasible due to limitations of NISQ devices. Even in ISQ era, where the qubits are less noisy, the high depth of the UCCSD ansatz can be a problem with increasing number of qubits.

With a focus on hardware-efficient ansatzes (HEA), we aim to improve the convergence of the VQE algorithm and make it feasible with current NISQ devices. We employ the methods presented in [KTL21] and extend them with existing variants of VQE to simulate properties of lithium-ion battery. The number of qubits can be varied based on the availability of quantum resources, thus, also making it suitable for the ISQ era. We focus on simulation of cathode materials, specifically, dilithium iron silicate. The results include finding the ground state of the material with varying number of qubits that is performed using CUDA Quantum [The] on NVIDIA A-100 GPU.

The rest of this manuscript is organized as follows. Sec. 1.1 answers the questions for NVIDIA Power-ups. A brief background about lithium-ion batteries is presented in Sec. 2.

Sec. 3 gives a comprehensive description about the quantum algorithms used for simulation. In Sec. 4, we report the results for simulation of Li_2FeSiO_4 followed by conclusion in Sec. 5. Sec. 6 is the outlook for future research directions.

1.1 NVIDIA Power-Ups

- **Aim**

  To simulate properties of lithium-ion battery on quantum computers using algorithms like VQE and CS-VQE. Specifically, we focus on finding the ground state of Li_2FeSiO_4.

- **Why is it interesting?**

  As we know that lithium-ion batteries have transformed portable electronics, enabling them to function independently and safely for extended periods through multiple recharge cycles. However, despite their current success, there's a growing need for even better batteries with longer lifespans, faster charging times, higher capacities, and lower costs.

  Quantum computing offers a fundamentally new way to simulate these systems, potentially overcoming limitations of traditional methods. It has the potential to perform electronic structure calculations with chemical accuracy and the resources scales polynomially with the size of the system.

- **Use of CUDA Quantum and NVIDIA GPUs**

  We use CUDA Quantum and NVIDIA GPUs to simulate VQE algorithm and its variants like CS-VQE, ADAPT-VQE. We also used Rotoselect optimizer in one of our experiments. For traditional VQE, we used the inbuilt function `cudaq.vqe`. Rotoselect and ADAPT-VQE were implemented from scratch but both used `cudaq.vqe` as their sub-routine. We were able to simulate the quantum circuits efficiently with upto 8 qubits. This is especially true with CS-VQE algorithm even with high depth (12 layers) (Fig. 6).

- **Performance results, numerical / scientific analysis**

  Results and Analysis can be found in Sec. 4.

2 Background

[Del+22] gives insights to answer the question of simulation of lithium-ion battery on a quantum computer. The work estimates the quantum resources needed to simulate dilithium iron silicate (Li_2FeSiO_4) with Quantum Phase Estimation (QPE) as the primary algorithm. Taking inspiration from that work, we now present a brief background and motivation on the problem at hand.

A battery as shown is Fig. 1 comprises an anode, cathode, separator, electrolyte, and two current collectors (positive and negative). The anode and cathode serve as storage for lithium. The electrolyte transports positively charged lithium ions between the anode and cathode through the separator. This movement generates free electrons in the anode, resulting in a positive charge at the current collector. The electric current then flows from the positive current collector through the device being powered to the negative current collector. The separator prevents the flow of electrons within the battery.

During discharge, as the battery provides an electric current, the anode releases lithium ions to the cathode, creating a flow of electrons. When the device is plugged in, the reverse occurs: Lithium ions are released by the cathode and received by the anode.

The cathode materials can be lithium cobalt ($LiCoO_2$), lithium manganese oxide ($LiMn_2O_4$) or the polyanion materials, e.g., Li_2FeSiO_4. Improving the performance of

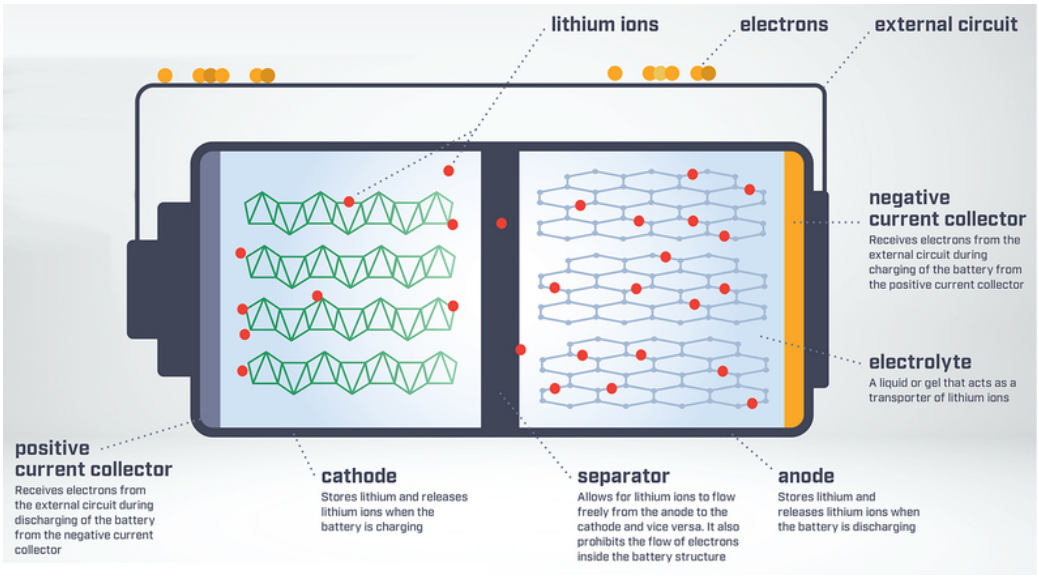

Figure 1: Working of a lithium ion battery. Figure from [All19]

lithium-ion batteries is crucial for creating better energy storage systems in the future. This requires not only finding new materials, but also developing more precise ways to predict how these batteries will behave.

3 Methodology

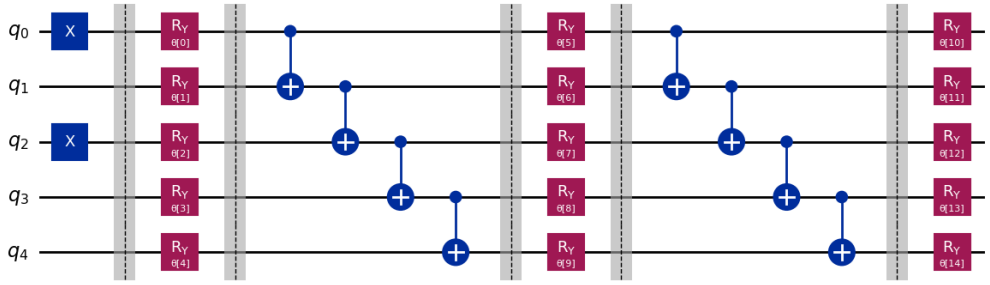

Figure 2: Hardware-efficient ansatz with 2 layers and initial Hartree-Fock state

[Del+22] uses first-quantization to construct the hamiltonian in plane-wave basis and QPE for resource estimation. Contrary, in this work, we use second-quantization for hamiltonian construction and focus on VQE algorithm and its variant called Contextual Subspace VQE (CS-VQE) (Sec. 3.2). The contextual part of the hamiltonian is then tested on the traditional VQE and ADAPT-VQE (Sec. 3.4). We call ADAPT-VQE as CS-ADAPT-VQE as we use the contextual subspace hamiltonians. We will be using these names interchangeably. Sec. 3.3 describes the use of Rotoselect algorithm to enhance CS-VQE. Assuming we have the molecular hamiltonian constructed H, we use a hardware efficient ansatz for L layers for the VQE algorithm as shown in the Fig. 2 above.

3.1 Qubit Tapering

Running calculations on large molecules using quantum computers is challenging because it requires a large number of qubits. This is especially true for algorithms like VQE used in quantum chemistry. To address this, researchers have developed various techniques to reduce the number of qubits needed. We use the symmetry-based qubit tapering approach [Bra+17][Set+20], which leverages the inherent symmetries in molecules to decrease the required qubits for simulations.

To differentiate CS-VQE from qubit tapering, we first simplified the Hamiltonians by exploiting the symmetries used in qubit tapering. Subsequently, we applied CS-VQE on the resulting tapered Hamiltonians to further reduce the required number of qubits.

3.2 Contextual Subspace Variational Quantum Eigensolver

[KTL21] describes a hybrid-quantum classical approach to approximate the ground energy of given hamiltonian by decomposing it into **non-contextual** and **contextual** parts. The noncontextual part is efficiently solved using established classical algorithms. The contextual part, often requiring fewer qubits than traditional VQE, is tackled using VQE on a quantum computer. One can vary the number of qubits for the contextual part to better approximate the ground state energy.

CS-VQE allows to achieve accurate predictions for the ground state energies with significantly fewer qubits compared to the traditional VQE approach. This is achieved by leveraging the inherent symmetries in the system, which reduces the number of terms and measurements needed in the calculations.

3.3 Rotoselect algorithm

Rotoselect [OGB21] is a gradient-free optimization algorithm. The algorithm updates the parameters $\theta = \theta_1, \dots \theta_D$ and rotation gates $R = R_1, \dots R_D$ one at a time according to the closed form for the optimal value of the d^{th} parameter θ_d^* when other parameters and gate choices are fixed:

$$\begin{aligned}\theta_d^* &= \underset{\theta_d}{\operatorname{argmin}} \langle H \rangle_{\theta_d} \\ &= -\frac{\pi}{2} - \arctan\left(\frac{2\langle H\rangle_{\theta_d=0} - \langle H\rangle_{\theta_d=\pi/2} - \langle H\rangle_{\theta_d=-\pi/2}}{\langle H\rangle_{\theta_d=\pi/2} - \langle H\rangle_{\theta_d=-\pi/2}}\right)\end{aligned}$$

where $\langle H \rangle_{\theta_d}$ is the expectation value of the objective function optimized over the parameter θ_d. We consider $R_d \in \{R_x, R_y, R_z\}$. The Rotoselect algorithm works by repeatedly going through each parameter and performing a calculation to improve the circuit. This cycle continues for a set number of times or until the results become stable (converge). This allows it to discover not only the best parameter values but also the most effective gate choices for minimizing a specific cost function.

3.4 ADAPT-VQE

Building a quantum circuit for a material or molecule typically involves choosing a predefined approach like UCCSD [Rom+18], which considers all possible single and double electron excitations. However, including all possible excitations can unnecessarily increase the complexity and cost of simulations without necessarily improving accuracy. Therefore, researchers have developed strategies to approximate the contribution of these excitations and select only the most relevant ones for each specific molecule. This can be achieved

using adaptive methods [Gri+19] that tailor the circuit to the individual problem. While sacrificing some generality, this approach can significantly improve performance.

We perform manual construction taking inspiration from [Jah].

1. Calculate gradients for all double excitations. We use parameter-shift gradient method.
2. In principle, one should identify double excitations with gradients surpassing a predetermined threshold. However we choose the one with maximum absolute gradient value and therefore only one gate is chosen.
3. Execute VQE to derive optimized parameters for the chosen double excitations.
4. Repeat steps 1 and 2 for single excitations.
5. Execute the final VQE for all selected excitations.

3.5 Application: simulation of a cathode material

We focus on simulating Li_2FeSiO_4. This material stands out for its sustainability because both silicon and iron are plentiful on Earth. Additionally, its strong covalent silicon-oxygen bonds make it highly resistant to heat. We obtain the structure from the Materials Project for (ID: mp-759942) from database version v2023.11.1 [Jai+13]. The structure and conventional unit cell is show in Fig. 3

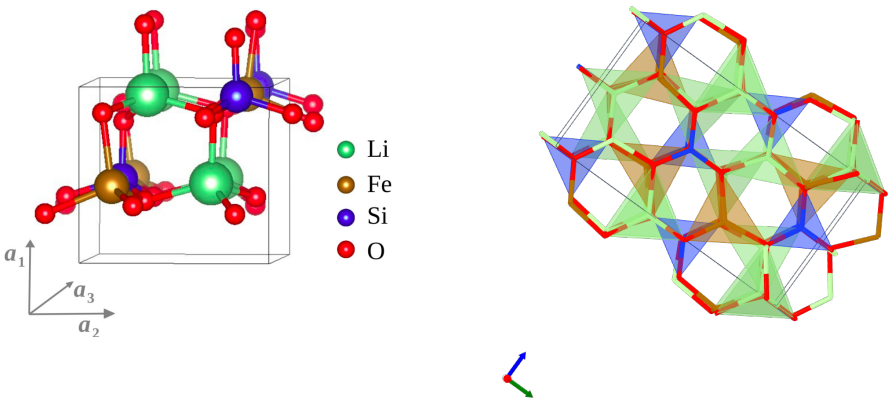

Figure 3: (left) Conventional unit cell. Image from [Del+22]. (right) Crystal Structure (Monoclinic)

4 Results and Discussion

For our analysis, we focus on the ground state energy of Li_2FeSiO_4. All calculations were performed using the STO3G basis. Jordan-Wigner qubit mapping was used to map the fermionic hamiltonian to qubit hamiltonian. We first performed active space reduction by selecting the orbtials from HOMO-3 to LUMO+3 level to reduce the orbitals count and therefore the number of qubits. CCSD was used as a classical solver to compare the quantum results.

4.1 Analysis on Active space reduction

We varied the HUMO-LUMO levels from 1 till 6 and plotted the qubit and measurement requirements as shown in Fig. 4. It is evident that any quantum algorithm for the challenge should not only aim to reduce the number of qubits but also handle the large number of hamiltonian terms efficiently as they scale exponentially with the number of qubits. Considering the time constraints, we continue our experiments with 16 qubit Hamiltonian.

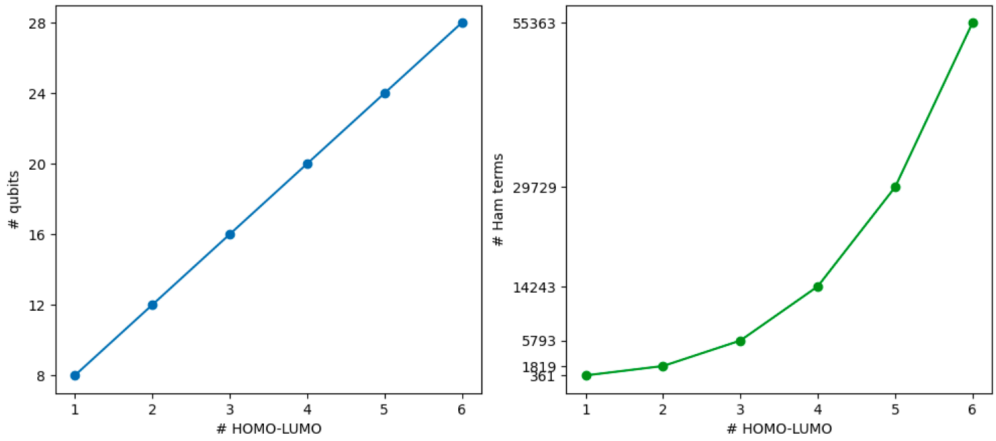

Figure 4: (left) Qubit requirements. (right) Number of hamiltonian terms.

From the table below, we can observe that the number of the circuit width for UCCSD ansatz is more than 43k. As in the ISQ era, longer circuit depths is still a problem, the UCCSD ansatz is not a favourable choice and therefore we continue with the Hardware-Efficient Ansatz.

Table 1: Resources needed by VQE with UCCSD ansatz

Parameter	Value
# Qubits	16
Circuit width	43371
# 2 qubit gates	36480
# Variational parameters	152

4.2 Tapering qubits in S3 formalism

We taper qubits in the Stabilizer Subspace Projection formalism. Unlike CS-VQE, this technique is exact, in the sense that it perfectly preserves the ground state energy of the hamiltonian. The number of qubits is reduced to 14 from 16. However, the number of hamiltonian terms are unaffected which can cause large measurement overhead. Therefore, we use CS-VQE to handle the same.

4.3 CS-VQE Simulation

We test CS-VQE algorithm on the 14 qubit tapered hamiltonian of Li_2FeSiO_4. Fig. 5 shows the scaling of hamiltonian terms with the number of qubits under consideration. For our analysis, we varied the qubits from 2 to 8 (max one can go is 14).

We run the CS-VQE algorithm with 4, 8, and 12 layers of HEA on hamiltonians needing qubits 2 to 8. 20 samples were taken for each experiment. Nelder-Mead optimizer for 1000 iterations was used. Fig. 6 shows the relative error and convergence time taken for each

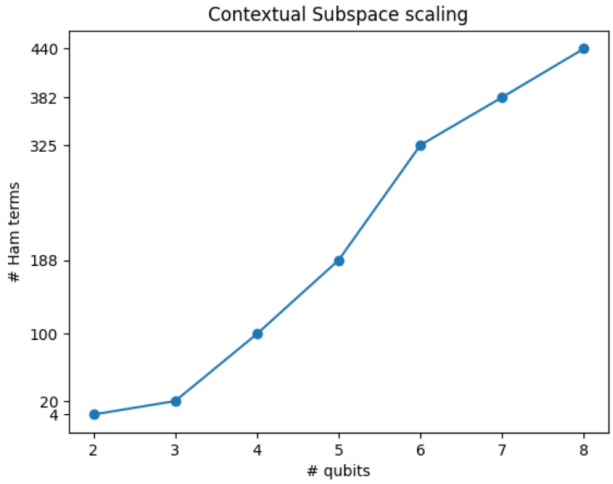

Figure 5: Number of hamiltonian terms vs Number of qubits in the contextual operator.

experiment. We achieved a relative error below 10^{-4} with 6 qubits. However, the relative error increases with the number of layers which is a bit counter-intuitive. As [KTL21] mentions, that the ansatz might suffer from barren plateau problem [McC+18]. Also, in experiments with 7 and 8 qubits, the relative error rises. The reason for this is not clear yet but we anticipate that it might be related to the expressibility of the ansatz [SJA19]. Interestingly, CS-ADAPT-VQE is able to perform better in this regime (Sec. 4.5) .

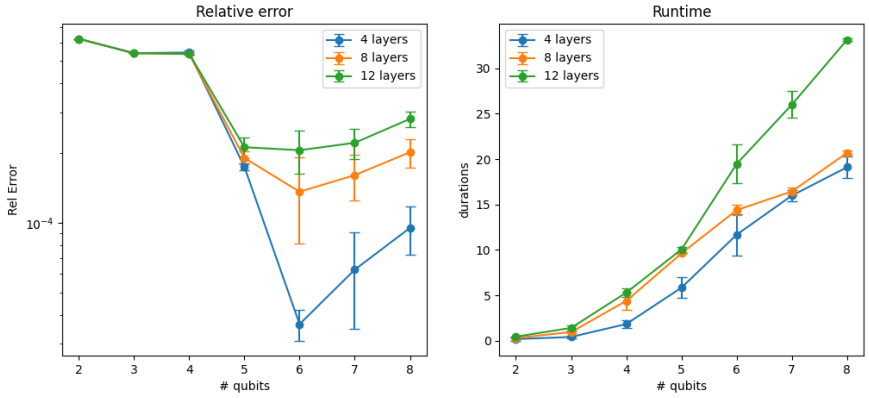

Figure 6: (left) Relative error vs qubits. (right) Convergence time vs qubits. Repeated for 20 sets. Mean and standard deviation are shown.

4.4 CS-VQE Simulation with Rotoselect Optimizer

The contextual hamiltonians are tested with the Rotoselect optimizer. We run the optimizer for 50 steps and 10 samples are taken for each experiment. Fig. 7 shows the initial and final quantum circuit in an experiment with 6 qubits. It can be observed that the final circuit has a combination of R_x, R_y and R_z gates.

Fig. 8 shows the relative error and convergence time taken for each experiment. A similar trend is seen as with the CS-VQE simulations in the previous section. However,

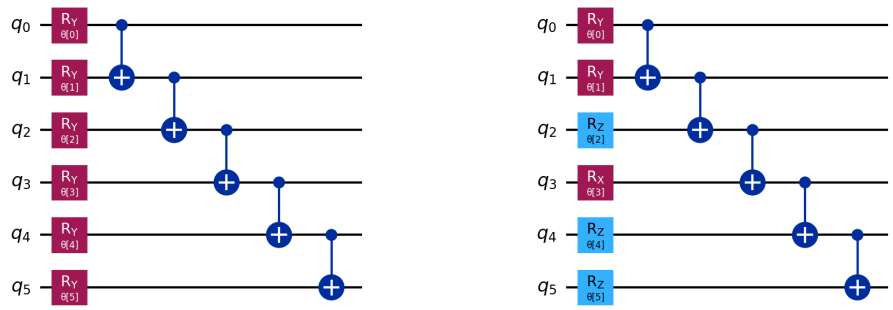

Figure 7: (left) Initial circuit. (right) Circuit after optimization with Rotoselect algorithm. Hartree-fock state is not shown here.

with Rotoselect optimization the depth and the number of trainable parameters is much less.

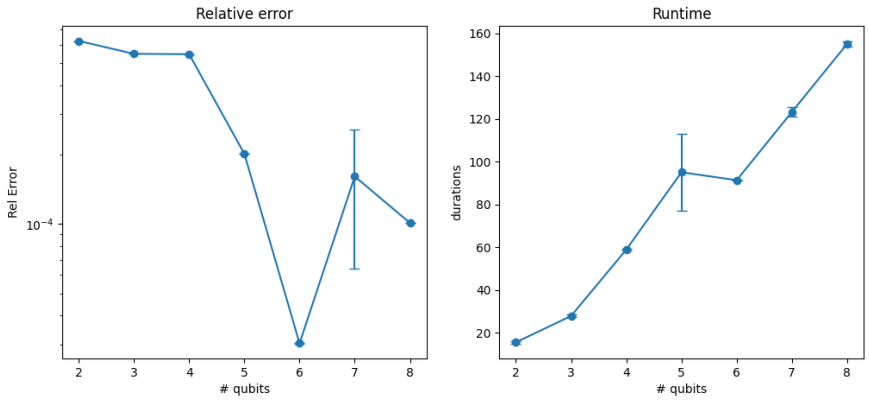

Figure 8: (left) Relative error vs qubits. (right) Convergence time vs qubits. Repeated for 10 sets. Mean and standard deviation are shown.

4.5 CS-ADAPT-VQE

Finally, we run the ADAPT-VQE algorithm on the contextual hamiltonians. 20 samples were taken for each experiment. Nelder-Mead optimizer for 1000 iterations was used. Fig. 9 shows the quantum circuit used for the 8 qubit hamiltonian. The algorithm was tested only with 7 and 8 qubit hamiltonians because the algorithm fails to project the Hartree-Fock state in the contextual subspace with qubits less than 7. Since the ADAPT-VQE algorithm needs the number of electrons to find the excitations, we continue our experiments with 7 and 8 qubits.

Fig. 10 shows the relative error and convergence time taken for each experiment. The relative error is comparable with the 6 qubit case of the previous two sections. This is particularly suprising as we only used only one double and single excitation gate.

Table 2 shows a comparison of the best results obtained by the 3 algorithms. The runtime for Rotoselect is unusually high in comparison to the other two algorithms because of its implementation. Unlike PennyLane [Ber+18]), creating quantum circuits in CUDA Quantum is not performed in a functional manner and therefore we needed to create the

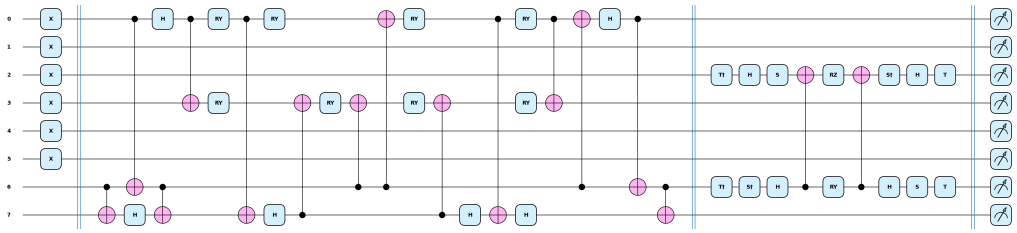

Figure 9: CS-ADAPT-VQE circuit. Hartree-fock state is followed by the Double excication circuit after which a single excitation circuit is applied. Finally the expectation value of the hamiltonian is computed. Circuit drawn using [Ber+18]

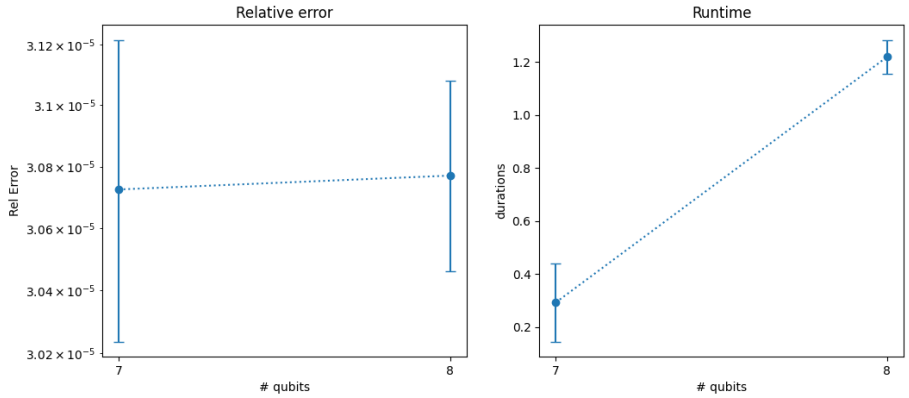

Figure 10: (left) Relative error vs qubits. (right) Convergence time vs qubits. Repeated for 20 sets. Mean and standard deviation are shown.

quantum circuit for every rotoselect cycle whose overhead is reflected in the runtime.

Table 2: Best results obtained and resources needed by various algorithms. We do not include the time required for computing the contextual subspace hamiltonians. Rel. Err is $\times 10^{-5}$

Algorithm	Qubits	Depth	2Q gates	Params	Rel. Err	Runtime (s)
CS-VQE	6	18	28	40	3.64 ± 0.54	11.67 ± 2.28
Rotoselect	6	6	5	6	3.03 ± 0.00031	91.33 ± 0.13
CS-ADAPT-VQE	7	29	16	2	3.07 ± 0.0049	0.29 ± 0.14

5 Conclusions

We investigated application of hybrid-quantum classical algorithm in the context of simulation of lithium-ion battery and attempts to implement the same using high-performance quantum computing libraries. We used a Hardware-efficient ansatz with the CS-VQE algorithm as well as enhanced it with Rotoselect and ADAPT-VQE algorithm. We believe that CS-VQE is a good candidate for the ISQ era. This research demonstrates a crucial link between simulating batteries and harnessing the power of quantum computing.

Active space selection is a technique used in quantum chemistry to decrease the number of qubits needed for calculations.Identifying which orbitals to "freeze" (exclude) presents

a significant challenge. Unfortunately, there's no universal algorithm for making this choice. However, an initial step towards selecting appropriate orbitals for freezing involves examining their occupancies. This is an important area of research in future to reduce the number of qubit requirements.

We employed techniques like active space reduction and qubit tapering to lower the number of qubit requirements and simulated the resulting hamiltonian with various quantum algorithms obtaining proimising results. This project has established a strong foundation for further research in applying quantum computing to battery simulations.

6 Outlook

Quantum algorithms for quantum chemistry is an exciting area of research. We have used VQE in our research, however, as the number of layers increase, the number of trainable parameters and the circuit depth grows rapidly. For this, we tried ADAPT-VQE [Gri+19], a promising candidate which focuses on reducing the total number of variational parameters as well as circuit depth, hence speeding the process. In future, we would like to experiment with ADAPT-QSCI, a recent algorithm in this regime [Nak+23] which aims at reducing the quantum resource requirements and can be an interesting candidate for the simulation of batteries. Since QPE is not a qubit friendly algorithm, its variants like Iterative QPE [Dob+07] and Rejection Filtering Phase Estimation [WG16] also opens a promising direction of research for lithium-ion battery simulation.

7 Acknowledgements

We acknowledge the hardware and software support from CUDA Quantum [The] and NVIDIA. This project would not have been possible with Tangelo [Sen+22] and Symmer [Sym]. PennyLane [Ber+18] and Qiskit [Qis23] were used for visualization purposes.

References

[Dob+07] Miroslav Dobšíček et al. "Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark". *Physical Review A* 76.3 (2007), p. 030306.

[Jai+13] Anubhav Jain et al. "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation". *APL Materials* 1.1 (July 2013), p. 011002. ISSN: 2166-532X. DOI: 10.1063/1.4812323. eprint: https://pubs.aip.org/aip/apm/article-pdf/doi/10.1063/1.4812323/13163869/011002_1_online.pdf. URL: https://doi.org/10.1063/1.4812323.

[WG16] Nathan Wiebe and Chris Granade. "Efficient Bayesian phase estimation". *Physical review letters* 117.1 (2016), p. 010503.

[Bra+17] Sergey Bravyi et al. "Tapering off qubits to simulate fermionic Hamiltonians". *arXiv preprint arXiv:1701.08213* (2017).

[Ber+18] Ville Bergholm et al. "Pennylane: Automatic differentiation of hybrid quantum-classical computations". *arXiv preprint arXiv:1811.04968* (2018).

[McC+18] Jarrod R McClean et al. "Barren plateaus in quantum neural network training landscapes". *Nature communications* 9.1 (2018), p. 4812.

[Rom+18] Jonathan Romero et al. "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz". *Quantum Science and Technology* 4.1 (2018), p. 014008.

[All19] Leslie J. Allen. *A look at how lithium ion batteries work*. 2019. URL: https://www.autonews.com/shift/look-how-lithium-ion-batteries-work.

[Gri+19] Harper R Grimsley et al. "An adaptive variational algorithm for exact molecular simulations on a quantum computer". *Nature communications* 10.1 (2019), p. 3007.

[SJA19] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms". *Advanced Quantum Technologies* 2.12 (2019), p. 1900070.

[Set+20] Kanav Setia et al. "Reducing qubit requirements for quantum simulations using molecular point group symmetries". *Journal of Chemical Theory and Computation* 16.10 (2020), pp. 6091–6097.

[KTL21] William M Kirby, Andrew Tranter, and Peter J Love. "Contextual subspace variational quantum eigensolver". *Quantum* 5 (2021), p. 456.

[OGB21] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. "Structure optimization for parameterized quantum circuits". *Quantum* 5 (2021), p. 391.

[Del+22] Alain Delgado et al. "Simulating key properties of lithium-ion batteries with a fault-tolerant quantum computer". *Physical Review A* 106.3 (2022), p. 032428.

[Sen+22] Valentin Senicourt et al. "Tangelo: An Open-source Python Package for End-to-end Chemistry Workflows on Quantum Computers". arXiv:2206.12424 (2022). DOI: 10.48550/arXiv.2206.12424. eprint: arXiv:2206.12424. URL: https://arxiv.org/abs/2206.12424.

[Nak+23] Yuya O Nakagawa et al. "ADAPT-QSCI: Adaptive Construction of Input State for Quantum-Selected Configuration Interaction". *arXiv preprint arXiv:2311.01105* (2023).

[Qis23] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

[Jah] Soran Jahangiri. *Adaptive circuits for quantum chemistry*. URL: https://pennylane.ai/qml/demos/tutorial_adaptive_circuits/#romero2017.

[Sym] Symmer team. *Symmer*. URL: https://github.com/UCL-CCS/symmer.

[The] The CUDA Quantum development team. *CUDA Quantum*. URL: https://github.com/NVIDIA/cuda-quantum.