Investigation of Variational Quantum Eigensolver techniques in Qiskit

PHYS 471 report

Yu Jun Shen

1. Introduction

Quantum computing is an emerging technology that uses the principle of quantum superposition to perform calculations. They may be able to outperform classical, electronic computers at certain calculations in the future. However, current quantum hardware is still too noisy, thus limited in terms of quantum circuit complexity and measurement accuracy. Due to these limits, scientists have turned to hybrid classical-quantum algorithms to maximize the near-term potential of quantum computing. One significant research direction is using quantum computers to conduct eigenvalue estimation, a common problem in many domains of science. My research this semester focused on implementing the Variational Quantum Eigensolver and two derivatives of it, to obtain the lowest energy states and levels of various Hamiltonian systems.

The VQE programs were implemented in IBM Qiskit, an open-source software development platform written in Python. Qiskit is developed by IBM to be compatible with current quantum computers. My research was carried out in the Abhishek lab, where eigenvalue estimation arises as a resource-intensive computational step in human decision-making models.

2. Overview of quantum information system

In quantum computing, a quantum bit, or qubit, is a two level system that can be in superposition between 0 and 1 state (eq. 1). The coefficients α and β are subject to a normalization constraint.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

A Hamiltonian is a measure of the system's energy, constructed by taking its kinetic and potential energy (eq. 2). The Hamiltonian matrix represents various physical systems such as a potential well, so it must be Hermitian (equal to its conjugate transpose).

$$H = \frac{P^2}{2} + V(x) \tag{2}$$

The fundamental principle enabling the VQE is the variational principle of quantum mechanics: for any Hamiltonian H, an estimate of its measurement is at least as large as its theoretical minimum eigenvalue (eq. 3). This allows us to construct "test" quantum circuits (ansatz) and measure its the resulting waveform. In eq. 3, $\psi(\boldsymbol{\theta})$ is the ansatz state generated using a parameterized ansatz circuit $U(\boldsymbol{\theta})$. $\boldsymbol{\theta}$ refers to the parameters and the measurement is obtained by passing in the initial wave function $|\psi_0\rangle$ (eq. 4).

$$\lambda_{min} \leq \lambda_\theta \equiv \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle \tag{3}$$

$$U(\boldsymbol{\theta})|\psi_0\rangle = |\psi(\boldsymbol{\theta})\rangle \tag{4}$$

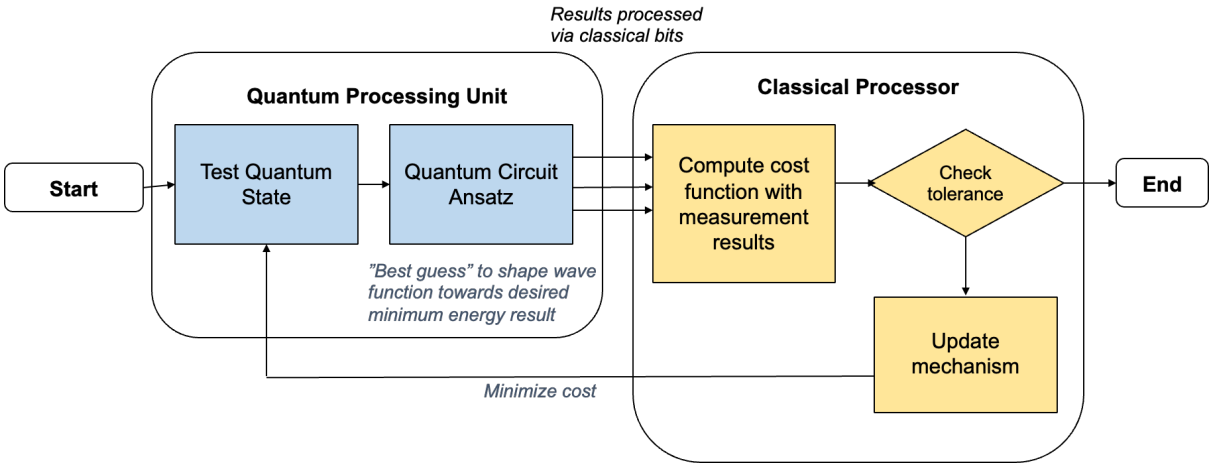


Fig. 1. Overview of Variational Quantum Eigensolver process to get lowest energy state (blue for quantum steps and yellow for classical steps).

A flowchart of the VQE algorithm is presented in Fig. 1. The objective is to find the lowest eigenstate of a Hamiltonian matrix by choosing appropriate circuit parameters. As a hybrid algorithm, a quantum test circuit is first generated and measured. Its result is passed on to the classical part which implements a minimization algorithm. The feedback loop updates the quantum circuit parameters to finally generate the lowest eigenvalue. By the variational principle stated above, this measurement can only be as low as the minimum ground state.

Sample one and two qubit ansatz circuits are presented in Fig. 2a, b for illustration purposes. Qiskit circuits are initialized from the left, with every qubit wire starting as $|0\rangle$. Subsequent (unitary) gates are applied sequentially while measurement induces quantum state collapse. Rx and Ry are standard rotation gates, and t1 and t2 are the parameters for the one-qubit ansatz. The classical optimizer updates t1 and t2 to minimize the measurement result. Similarly, the two-qubit case is parameterized with more angles (using U3 gates). However, at two and more qubits entanglement must be possible hence three CNOT gates are added. The design of appropriate quantum ansatz circuits that cover the design space but are within reach of optimization is an important consideration for the VQE algorithm.

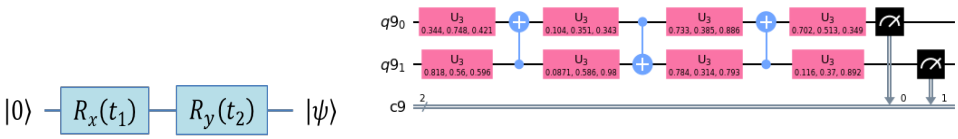


Fig. 2a. Single qubit ansatz example. **2b.** Two qubit ansatz example

3. Research background and specific contribution

My research this semester started with a functional basic VQE implementing the momentum & position space approach proposed by McGuigan and Miceli (2018). Since the potential well Hamiltonian consists only of a kinetic and a potential energy term, it suffices to try separate ansatzes in the position and momentum basis then add their results. The output

bitstrings $|0000\rangle, |0001\rangle, |0010\rangle \ldots |1111\rangle$ are allocated weights based on their distance from the center, “trotterizing” the well.

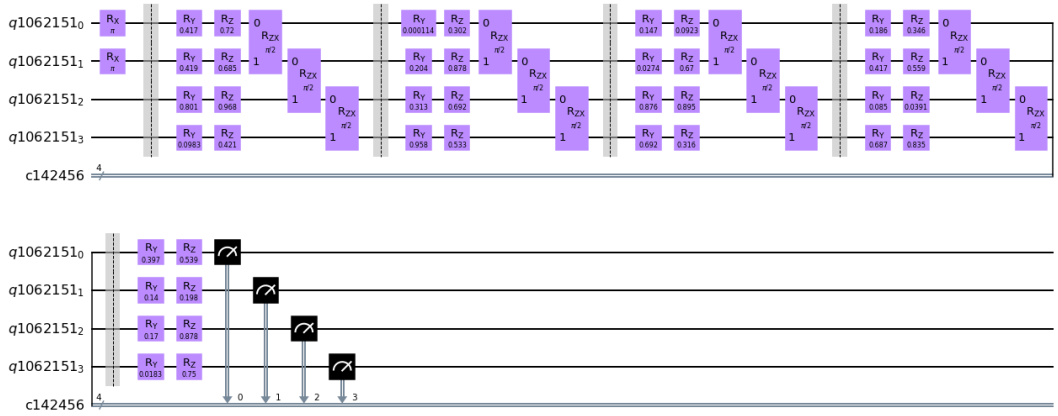


Fig. 3. Four qubit, four layer hardware optimized ansatz example

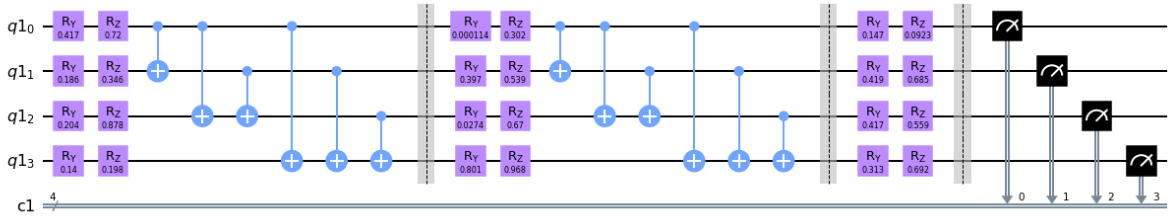


Fig. 4. Four qubit, two layer Ry-style ansatz example

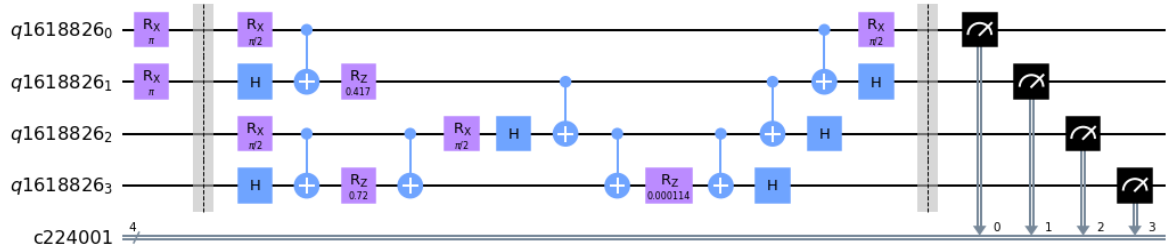


Fig. 5. Four qubit UCC ansatz example

Fig. 3 to 5 show the different ansatzes studied for this research. The hardware optimized ansatz (Fig. 3) has fewer connections across different qubit wires per layer than the Ry style (Fig. 4), though both achieve entanglement of qubits. The code was modularized to be able to include more layers as desired. The UCC ansatz is based on a different encoding incompatible with the initial potential well setup and was not further considered.

My specific contributions this semester was to improve on the capabilities of the VQE to reach multiple energy states beyond the ground state and to solve for arbitrary Hamiltonians beyond the infinite potential well. As such, two modifications were made to the basic VQE process described in Fig. 1: a new algorithm to search for higher energy states, and a different approach to the Hamiltonian instead of only the “position” and “momentum” terms.

4. Subspace Search Variational Eigensolver

Instead of using a single $|0000\rangle$ input, the Subspace Search Variational Eigensolver (SSQVE) prepares multiple, orthogonal wave-states input and calculates a weighted sum as the final cost function (Nakanishi et al. 2019). Multiple circuits are generated and summed (eq. 6). $\mathcal{L}_w(\theta)$ is the total cost for every iteration, while w_j is the cost per energy level. The larger the energy level, the lower the weight given. This maps which energy level corresponds to which input (e.g. out of $|0000\rangle$, $|0001\rangle, |0010\rangle \ldots$)

$$\mathcal{L}_w(\theta) = \sum_{j=0}^{k} w_j \langle \varphi_j | U^\dagger(\boldsymbol{\theta}) \mathcal{H} U(\boldsymbol{\theta}) | \varphi_j \rangle \tag{6}$$

The SSVQE algorithm is sketched out in Fig. 6. First, the Hamiltonian of interest is generated. For every standard ansatz style, X-gates are prepended to get the orthogonal input before the position and momentum space measurements (same as VQE). The position and momentum space ansatz differ by a Fourier block. Both measurements are added and weighed for every energy state. Each energy state's cost is tallied and used for classical optimization of the circuit parameters. The loop continues until the cost is low enough for convergence, returning a set of tuned quantum ansatz parameters.

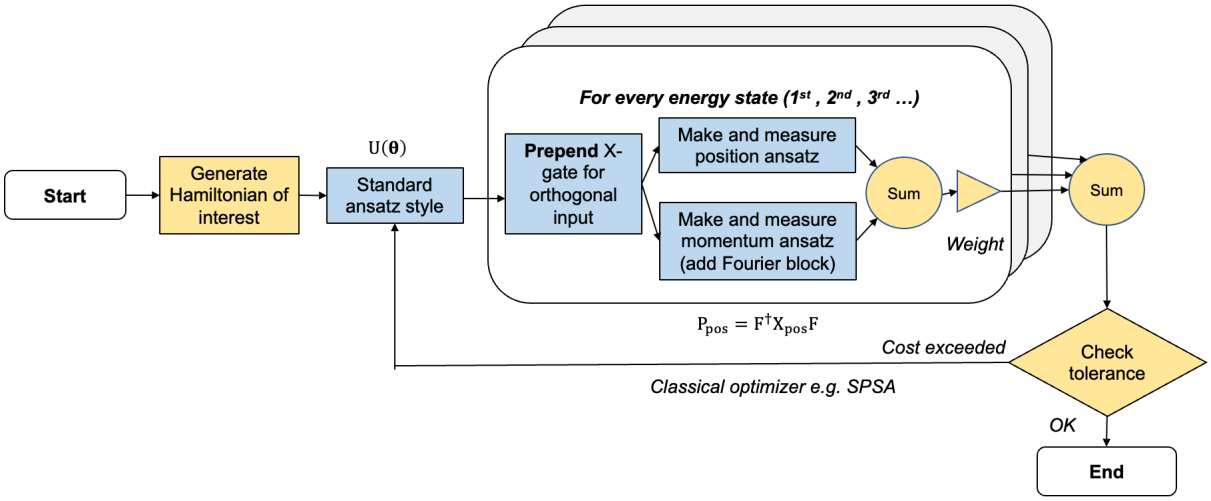


Fig. 6. SSVQE flowchart

5. Hamiltonian Decomposition

The other modification is the decomposition of the Hamiltonian to a linear combination of tensor products (eq. 7). For a two-qubit system, the Hamiltonian is a 4x4 matrix; for four qubits, it is 16x16. The formula to generate the tensor product coefficient is shown in eq. 8.

$$H = \sum_{i,j=1,x,y,z} a_{i,j} (\sigma_i \otimes \sigma_j) \tag{7}$$

$$a_{i,j} = \frac{1}{4} tr[(\sigma_i \otimes \sigma_j) H] \tag{8}$$

The ¼ coefficient in eq. 8 allows for normalization. Pauli matrices are listed in eq. 9.1 to 9.4 for reference.

$$\sigma_I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{9.1}$$

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{9.2}$$

$$\sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{9.3}$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{9.4}$$

Eq. 10 shows an example of the tensor product between σ_I and σ_X.

$$\sigma_I \otimes \sigma_X = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{10}$$

This process was automated in Python together with the ansatz generation process (Fig. 7). For an identity or Z rotation on a qubit line, no end gate is needed. If sigma X is present, the Hadamard gate is appended at the end of the circuit (immediately before the measurement). Lastly, a sigma Y term corresponds to appending a U2 gate at the end.

```
for qubit_line in range(QUBIT_COUNT):
    if operator_list[qubit_line] == "I" or operator_list[qubit_line] == "sigma_z":
        pass
    elif operator_list[qubit_line] == "sigma_x":
        qc.h(qubit_line)    # Hadamard gate from David Khach github tutorial
    elif operator_list[qubit_line] == "sigma_y":
        qc.u2(0, np.pi/2, qr[qubit_line])    # use U2 Gate
```

Fig. 7. Hamiltonian decomposition code

A flowchart of the Hamiltonian decomposition version of VQE is presented in Fig. 8, extending the range of computation to arbitrary Hamiltonians. The Hamiltonian of interest is first decomposed into a linear combination of tensor products. Then, for each term in the combination (implemented as a Python dictionary object), the relevant rotation gate is appended at the end of the corresponding standard ansatz. All ansatzes are measured and the results summed as the cost value for classical optimization. Note the Hamiltonian must be Hermitian; invalid Hamiltonians could result in optimization to a lower “eigenvalue” than allowed by the theoretical minimum.

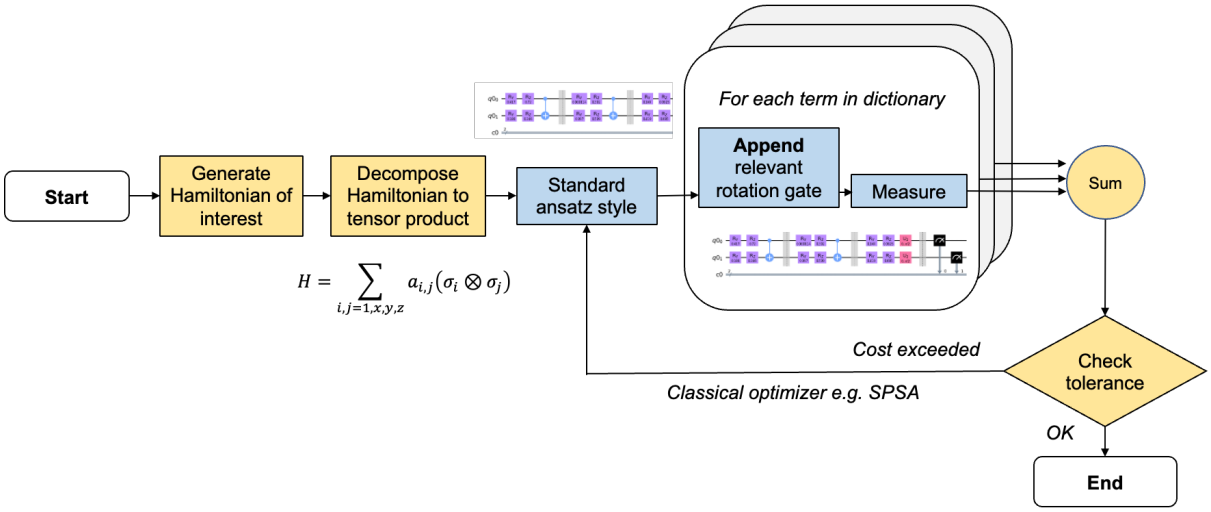
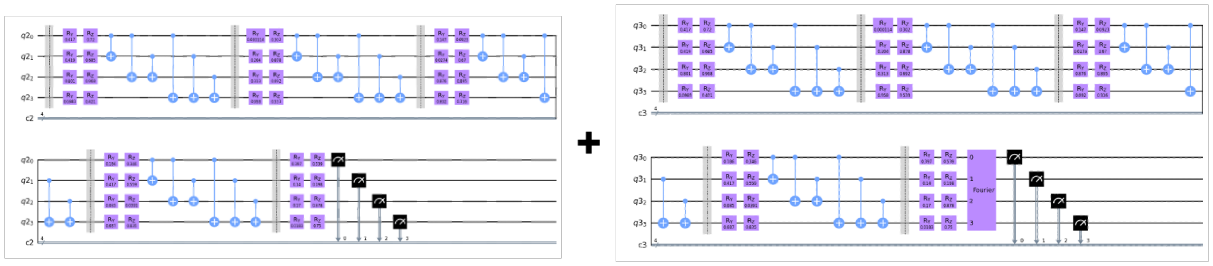


Fig. 8. Hamiltonian decomposition flowchart

The three VQE versions are contrasted with examples below. In the standard VQE of Fig. 9a, the position and momentum measurements are used (with only $|0000\rangle$ input). For the SSVQE (Fig. 9b), the process needs to be repeated for different energy states and X gates are added at the start of the circuit to generate orthogonal inputs. Furthermore, weights are ranked with lower energy states using higher weights. Lastly, for the Hamiltonian decomposition (Fig. 9c) rotation gates are added at the end of the circuit. Measurements are also multiplied by tensor product coefficient from eq. 8.



Position space
(penalize positions away from well center)

Momentum space
(get kinetic energy)

Fig. 9a. Standard VQE illustration

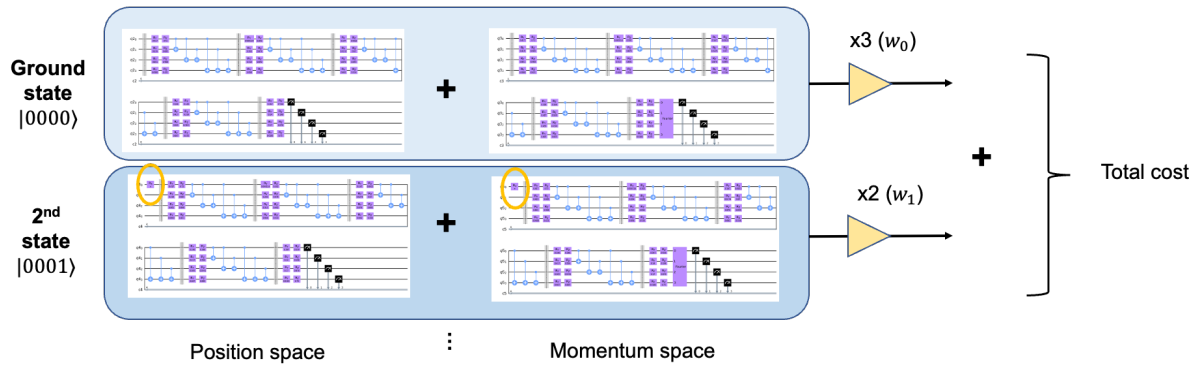


Fig. 9b. Subspace Search VQE illustration

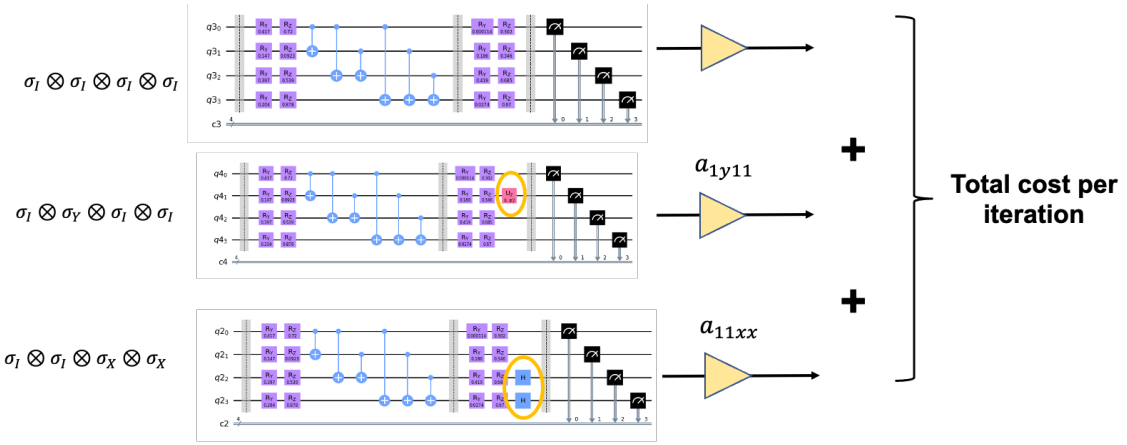


Fig. 9c. Hamiltonian decomposition illustration

The full codes for VQE, SSVQE and Hamiltonian decomposition are available online at my public GitHub repository[1].

6. Results and discussion

Via the Subspace Search method, the three lowest energy states can be obtained accurately at the same time. These are computed to 1x, 4x and 9x the theoretical minimum result as expected (Fig. 10). This computation used the momentum plus position space ansatzes per eq. 2 and Fig. 9a.

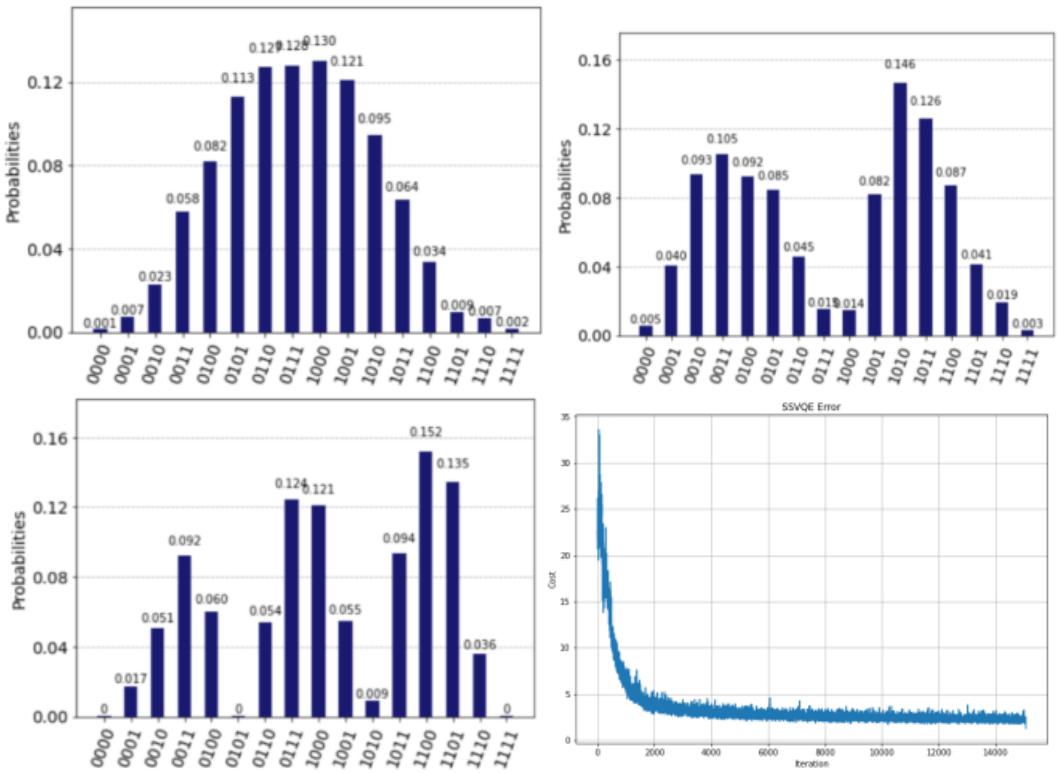


Fig. 10. SSVQE results example. Clockwise from top left: ground state measurement, second state measurement, cost function, third state measurement plots

[1] https://github.com/GryphonTorch/Quantum

The results of the Hamiltonian decomposition code is presented next in Fig. 11, for the cost function trajectory and measured final waveform. The code correctly deconstructs a Hamiltonian into a linear combination of tensor products (Eq. 11); the output dictionary of {I-I, I-sigma X, sigma x-sigma x and sigma y-sigma y} can be manually checked. By measuring once for every component at very step of the optimization, the lowest eigenvalue was accurately estimated at 1.73169 (ideally 1.71885).

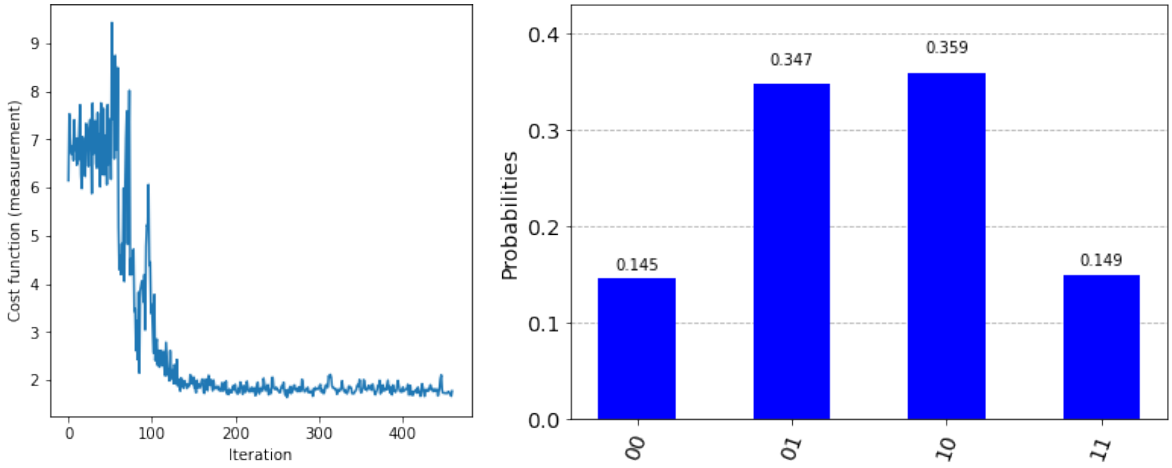


Fig. 11. Hamiltonian decomposition results

$$\begin{pmatrix} 9.0 & -4.5 & 0.0 & 0.0 \\ -4.5 & 9.0 & -4.5 & 0.0 \\ 0.0 & -4.5 & 9.0 & -4.5 \\ 0.0 & 0.0 & -4.5 & 9.0 \end{pmatrix} = 9 \cdot \sigma_I \otimes \sigma_I - 4.5 \cdot \sigma_I \otimes \sigma_X - 2.25 \cdot \sigma_X \otimes \sigma_X - 2.25 \cdot \sigma_Y \otimes \sigma_Y \tag{11}$$

Different classical optimizers such as linearized minimizers and other Qiskit built-in modules were tested as well in the research. Among the options, the Simultaneous Perturbation Stochastic Approximation (SPSA) solver worked best to optimize the ansatz even at high problem dimensions. SPSA estimates the gradient at any point using stochastic guesses (Eq 12.1 to 12.4).

$$a_k = \frac{a}{(A+k+1)^{\alpha}} \tag{12.1}$$

$$c_k = \frac{c}{(k+1)^{\gamma}} \tag{12.2}$$

$$\theta_{k+1} = \theta_k - a_k * g_k(\theta_k) \tag{12.3}$$

$$g_k(\theta_k) = \frac{[y(\theta + c_k * \delta) - y(\theta - c_k * \delta)]}{2c_k} * (\delta^{-1}) \tag{12.4}$$

The optimizers all converge (Fig. 12) but SPSA reaches close to the theoretical minimum in the lowest number of iterations within a multi-layer, multi-qubit ansatz.

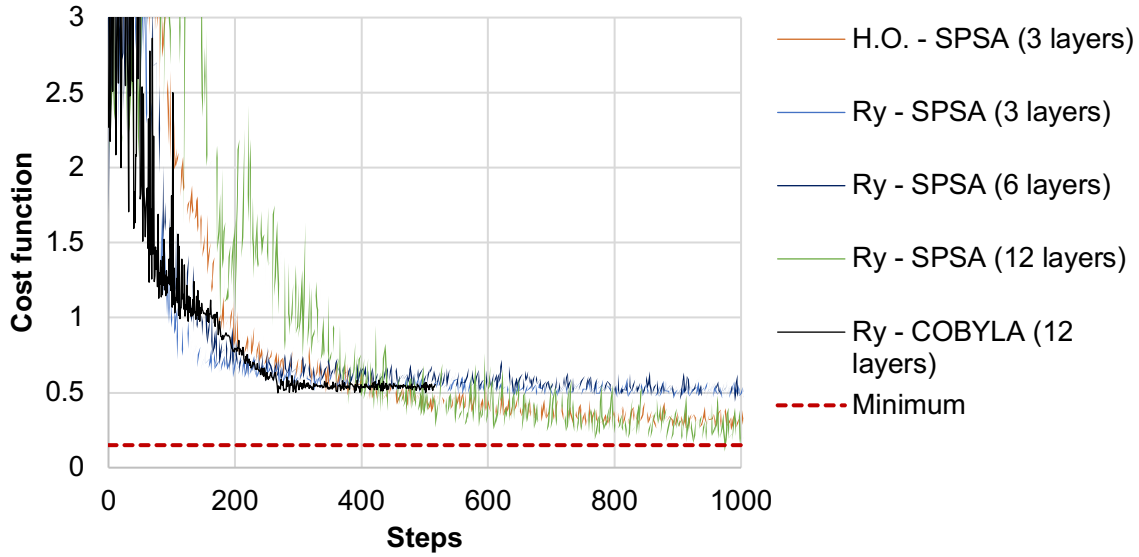


Fig. 12. Comparison of different optimizers

The SPSA parameters such as α and γ in Eq. 12 could not be tuned too aggressively. Otherwise, while the optimizer may work faster for simpler ansatz (e.g. one qubit), it turns unstable at higher problem complexity. The optimizer choice must balance between the ansatz flexibility and accessibility to the solver, to actually reach the lowest state. Similar to machine learning, a more capable circuit tend to be harder to train.

7. Conclusion

In conclusion, we have presented findings on the enhancement of the Variational Quantum Eigensolver to find multiple lowest energy states via the Subspace Search method. In addition, by changing the way Hamiltonians are calculated, the code can now work with arbitrary Hamiltonians beyond just the potential well model. This will be helpful to apply the quantum algorithm to other scientific problems of finding eigenvalues. We have also investigated algorithmic parameters to improve convergence of the program.

8. Acknowledgements

I would like to thank Professor Abhishek Bhattacharjee and Dr Raghavendra Pothukuchi for guiding me in their lab, Professor Steven Girvin for serving as my faculty advisor, and the Yale STARS II Program for supporting my research.

9. References

Kandala A. et al. (2017)“Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets.” *Nature.* doi:10.1038/nature23879

Miceli, R., McGuigan, M. (2018) Quantum computation and visualization of Hamiltonians using discrete quantum mechanics and IBM Qiskit. *Proceedings 2018 New York Scientific Data Summit*.

Nakanishi, K. M., Mitarai, K., Fujiii K. (2019) Subspace-search variational quantum Eigensolver for excited states. Physical Review Research 1. 033062.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M. H., Zhou, X. Q., Love, P. J., Aspuru-Guzik, A., O'Brien, J. L. (2014) A variational eigenvalue solver on a quantum processor. *Nature Communications*, 5:4213.

Shende, V. V., Markov, I. L., Bullock, S. S. (2003) Minimal universal two-qubit quantum circuits. *APS Physical Review A.*

Qiskit textbook and documentation. https://qiskit.org/