Quantum Eigen Sweet

Rohit Prasad

Indian Institute of Science Education and Research, Kolkata

Mentor
Petar Korponaić
CEO, Quantastica

Quantum Open Source Foundation

23rd January, 2021

1 Introduction

The traditional way of finding minimum eigenvalue of a hermitian matrix consists of Semi-Classical method where we use Classical Optimization and variational principles to find the lowest eigenvalue of a matrix.

Here we try to find a new method to find the lowest eigenvalue without optimization. We start with only 2x2 Hermitian Matrix as our target matrix. We use two methods- "Quantum Approach" and "Semi-Quantum Approach" method.

The "Quantum Approach" method, is implemented using Quantastica's Circuit Generator. If we input an initial statevector and the final statevector to the generator, it creates a circuit that takes initial state to the final state. We prepare a set of initial states that encode various matrices and we prepare a corresponding set of final states corresponding to lowest energy eigenstates of each encoded matrix.We assume that, for a sufficient set of trial initial states and final states, the circuit generator is able to generate a circuit that can find lowest energy eigenvector for any hermitian matrix of dimension 2x2.

The second method i.e the "Semi-Quantum" method, we try to find some correlation between the Eigenstate and its matrix. Getting the correlation, fitting it with curves to make some prediction to check our method.

2 Quantum Approach

This approach employs use of a finite set of predetermined matrices encoded into an initial state and their eigenvectors as final state as an input to the circuit generator to find a circuit that is unique to all the individual pairs solution circuit and may be able to predict eigenvector corresponding to lowest eigenvalue of some new matrix outside of the trial set.

2.1 Encoding the Hermitian Matrix

To encode a hermitian matrix into an initial state, we first transform it into a unitary matrix. Exp(iH) has been used as the mapping of hermitian matrix to unitary matrix. The new unitary matrix can simply be applied as a quantum gate upon ground state or superposition state of a qubit. We consider both the cases i.e the ground state and the superposition state.

With the Encoded Matrix state we now prepare the initial state for the generator. The circuit consists of 3 qubit, for which the initial state is defiend as |Encoded State> $\otimes|0\rangle \otimes |0\rangle$, expecting the final state to be of form $|0\rangle \otimes |Eigenstate\rangle \otimes |0\rangle$.

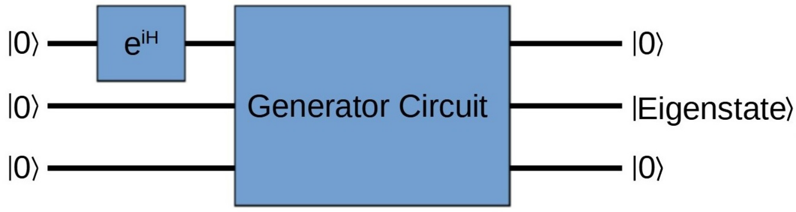

Fig. 1. Quantum Approach Circuit

2.2 Hermitian Matrix Set

The first set of hermitian matrices we input into the generator was set of all hermitian matrices of dimension 2x2. But with random hermitian matrix of dimension 2x2, the generator was unable to find a circuit unique to all pairs of initial and final statevectors. So we try to find a solution for a subset of 2x2 hermitian matrices defined uniquely.

2.2.1 Same Eigen Value

We now input a set of 2x2 hermitian matrices that have same lowest eigenvalues.We choose multiple such sets. One of them is subset of 2x2 matrices with same lowest eigenvalue of "-1".

Few elements of the subset are listed as:

- Pauli X matrix
- -1×Pauli X matrix
- Pauli Z matrix
- -1×Pauli Z matrix
- 2×Identity matrix + Pauli X matrix

and other elements that give us the minimum eigenvalue of -1.

2.2.2 Similar Decomposition

The second subset type contains elements that can be decomposed using similar decomposition into Pauli matrices. We choose the subset to be of form n×(A+B) where n $\in \mathbb{R}$ and A, B $\in$ [Pauli X, Pauli Y, Pauli Z, Identity].

A specific subset is given as, S = {n×(Pauli X+Pauli Z) | n $\in \mathbb{R}$ }.

$$S = \begin{bmatrix} n & n \\ n & -n \end{bmatrix} \text{ such that } n \in \mathbb{R}$$

2.2.3 Linear Shift

This subset works as set of all matrices of decomposition,
H = n×Identity + (n+a)×Pauli X + (n+b)×Pauli Y + (n+c)×Pauli Z, where a, b, c are fixed and belongs to $\mathbb{R}$ where n varies while belonging to $\mathbb{R}$.

$S_{a,b,c}$= {n∗I +(n+a)∗X +(n+b)∗Y +(n+c)∗Z|n ∈$\mathbb{R}$}

$$S_{a,b,c} = \begin{bmatrix} 2*n+c & a+n-(b+n)i \\ a+n+(b+n)i & c \end{bmatrix} \text{ such that } n \in \mathbb{R}$$

2.3 Programming with Quantastica's Circuit Generator

Quantastica Circuit Generator, when given a set of input and output states, tries to find a circuit unique to all inputs and their respective outputs, using reverse engineering circuit method.

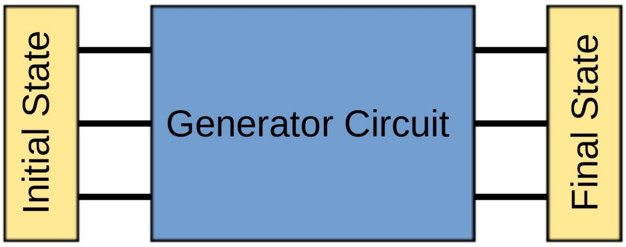

Fig. 2. Generator programming Diagram

2.4 Results

With all sets of initial and final state-vectors tried individually, the generator was able to give at least one circuit that could be implemented to evolve the initial to final statevector.

When we put all the initial and final statevector pairs of a subset together in the circuit generator to potentially find a general circuit that could give the eigenvector corresponding to lowest eigenvalue as the output, the circuit generator was unable to find any unique circuit solution to all the initial and final statevector pair within tolerance of 1.

So, we conclude that with current circuit generator, there isn't any general algorithm for used subset of dimension 2x2 hermitian matrix that could give the eigenvector corresponding to minimum eigenvalue when encoded with the general exp(iH) method.

3 Semi-Quantum Approach

This approach works with a third subset of the previously defined list of subsets of 2x2 hermitian matrices.
H = n×Identity + (n+a)×Pauli X + (n+b)×Pauli Y + (n+c)×Pauli Z, where a, b, c are fixed and belongs to $\mathbb{R}$ where n varies while belonging to $\mathbb{R}$.

$$S_{abc} = \{\text{n}*\text{I} + (\text{n}+\text{a})*\text{X} + (\text{n}+\text{b})*\text{Y} + (\text{n}+\text{c})*\text{Z} | \text{n} \in \mathbb{R}\}$$

3.1 Subgroup $S_{3,5,7}$

We have worked with multiple such subsets, taking the following as a reference, a=3, b=5, c=7.

$$S_{3,5,7} = \{\text{n}*\text{I} + (\text{n}+3)*\text{X} + (\text{n}+5)*\text{Y} + (\text{n}+7)*\text{Z} | \text{n} \in \mathbb{R}\}$$

To make a prediction model for this set we need to predict the 2 complex numbers corresponding to the state $\alpha|0> + \beta|1>$. For the eigenvector, global phase doesn't matter so without any issue we can assume that α is a real number. We can map the $\alpha|0> + \beta|1>$ state to a new 2x1 matrix defined as ,

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ |\beta| e^{i\theta} \end{bmatrix} \Longrightarrow \begin{bmatrix} \alpha \\ \theta \end{bmatrix} = \begin{bmatrix} \alpha \\ \cos^{-1}(\frac{real(\beta)}{abs(\beta)}) \end{bmatrix}$$

This way we only have to deal with 2 variables while predicting. α and θ.
Function of a curved line in 3d can be generalised in a equation as,

$$\text{f(x)} = \text{g(y)} = \text{h(z)}$$

We can make 2 separate equations out of it to make two 2-dimensional graphs which could be fitted more effectively. Making our new generalised equation of the form,

$$\text{f(x)} = \text{g(y)}$$
$$\text{f(x)} = \text{h(z)}$$

We can then fit the points to get a curve in two 2-dimensional plots. For our case the equation stands as,

$$\alpha = f_{a,b,c}(\text{n})$$
$$\theta = g_{a,b,c}(\text{n})$$

We take 100 values of "n" between -100 to 100 and plot in 3d plot taking n along x axis, α along y axis and θ along z axis,

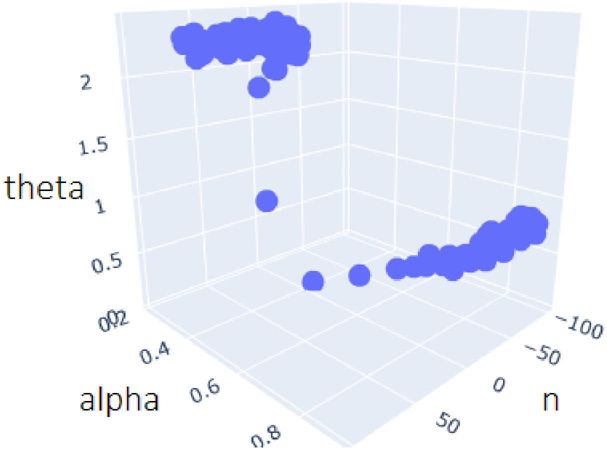

Fig. 3. Graph of n vs α vs θ

After this, we plot the two equations separately and fit it using sum of two Sigmoid functions.

$$\alpha = \frac{24.26}{37.21+7.283\times\exp(-0.62\times n)} - \frac{1.085}{1+0.01759\times\exp(-0.7687\times n)} + 0.8706$$

$$\theta = \frac{-2.661}{1+0.004604\times\exp(-1.033\times n)} + \frac{4.206}{1+0.005635\times\exp(-1.307\times n)} + 0.7446$$

with the graphs,

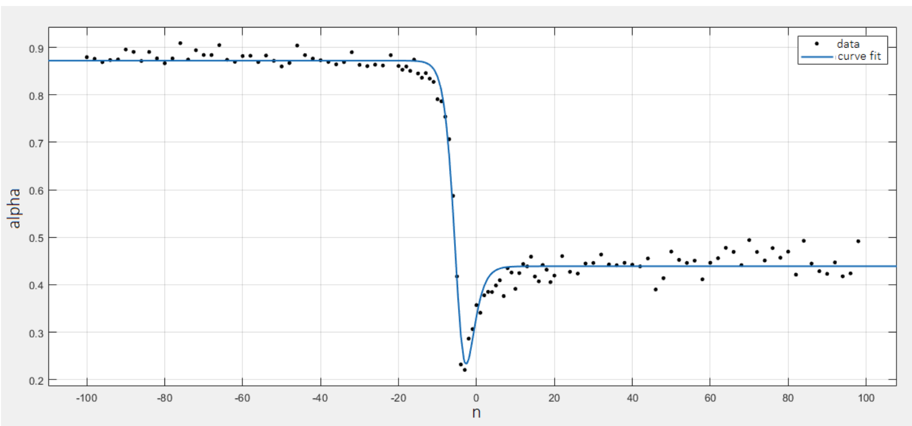

Fig. 4. Graph of n vs α

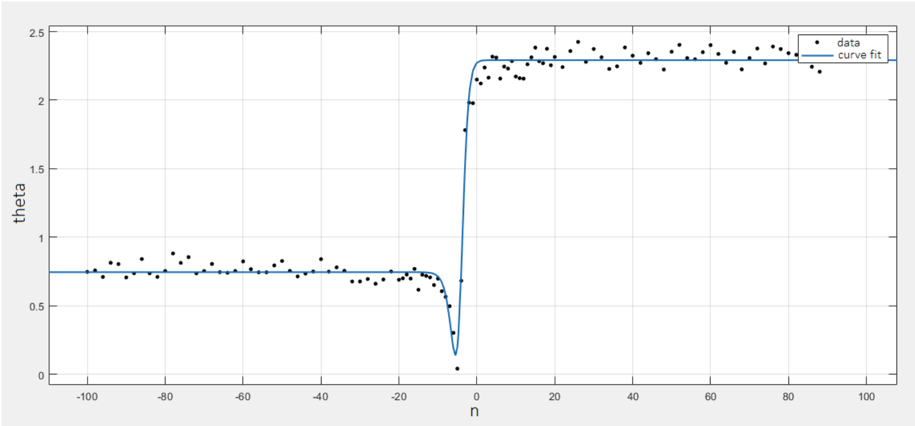

Fig. 5. Graph of n vs θ

3.2 Result

We got a curve that predicts the eigenvector of a matrix in the subgroup $S_{3,5,7}$. When tested it with 500 random "n" between (-500,500) we observed that for 249 of our predictions had errors in eigenvalue under 5$\%$.

4 Conclusion

We see above results passable for the relatively new methods we tried to implement. For the first method, we see a possibility of improvement by trying multiple methods of encoding the hermitian matrices into the the initial state and additionally by encoding the eigenstate via some mapping into the final state, rather than providing the eigenstate as the final state to the input of the generator . Since the generator works on reverse engineering method, it requires a well-defined final statevector thus limiting the use of ancillary qubits. Additionally, if the generator is able to generate a circuit independent of the qubit 1 and 3 i.e. it generates a circuit with respect to only the eigenstate stored in the second qubit , then there is a prospect to obtain a better circuit.

We implemented the second method for a=3, b=5, c=7. This method holds for other subgroups as well with parameters α and θ have functional dependence of the form
$\alpha = f_{a',b',c'}$(n) and $\theta = g_{a',b',c'}$(n), where a',b',c'.
The group of 2x2 Hermitian Matrix (H) can be taken as the union of all such sets as,

$$H_{2x2} = \bigcup_{a,b,c \in \mathbb{R}} S_{a,b,c}$$

If we can find a general formula for the 5 dimensional curve with variables [a,b,c,n,α,θ], connected together as,

$$\alpha = \text{f(n,a,b,c)}$$
$$\theta = \text{g(n,a,b,c)}$$

we will be able to predict the eigenstate of all 2x2 Hermitian Matrix using the curve.

Reference

1. Quantastica's Circuit Generator.

    - https://quantastica.com/#generator
    - https://youtu.be/D4DyquIWlh8

2. Qiskit

    - https://qiskit.org/
    - https://www.ibm.com/quantum-computing/