Non-Hermitian HHL Algorithm

Muhammad Abdullah Ijaz

1 Abstract

The Harrow-Hassidim-Lloyd (HHL) quantum algorithm solves Linear System Problem with a exponential speedup compared to classical algorithms. The algorithm assumes that the NxN matrix A is hermitian and sparse. But we will see how the algorithm can be used for non hermitian, non sparse matrices at the cost of twice as many quantum bits.

2 Introduction

Systems of Linear equations arise naturally in many fields, as such the applications of the HHL algorithm is limited by the users ability to transform the Linear Problem into a sparse matrix A. Many problems will take a similar form as Poisson equations, which are hermitian as well as easy to simulate.

For the cases when the matrix is not hermitian, HHL paper discussed the method of constructing an arbitrary matrix C which contains the matrix A as a sub-matrix and is hermitian. Allowing the algorithm to solve the problem in poly(log N , k), where k is the sparse factor.

3 HHL Algorithm

$A\hat{x} = \hat{b}$

Where the NxN hermitian matrix is A, and $\hat{b}$ is the input state. Let the singular value decomposition of A be

$$A = \sum_{j=0}^{N} \lambda_j |u_j >< u_j|$$

The non hermitian form of this algorithm can be found as follows. Define

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

where a,b,c,d are real, and b $\neq$ c.

Define H = $\begin{pmatrix} 0 & A \\ A\dagger & 0 \end{pmatrix}$,

In this form the input and output vectors change as $\hat{B} = |b, 0>$, $\hat{X} = |0, x>$,
Hence, $H\hat{X} = \hat{B}$, becomes,

$$\begin{pmatrix} 0 & A \\ A\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

It can be seen that the first equation, is the linear system equation for hermitian matrix.

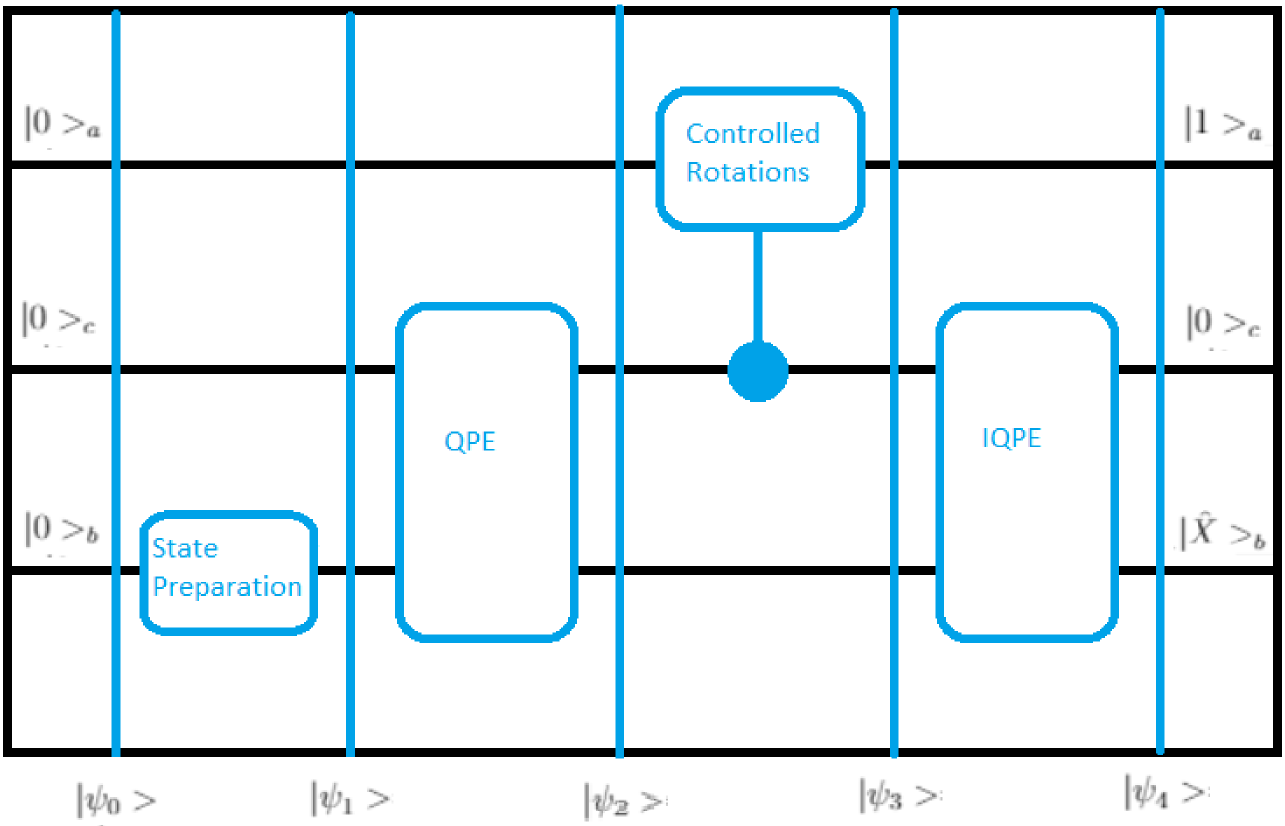

Figure 1: HHL quantum Circuit Schematic

3.1 Definitions

The algorithm requires only 3 quantum registers. Ancilla register (a) with 1 ancilla bit for inverting eigenvalues, input register (b) which scales proportionally with the size of $\hat{x}$ and control register (c) which scales as square of size of $\hat{x}$.

3.2 State Preparation

The initial state to a vacuum state of size, s = a+b+c. As such the arbitrary state $|B>$ can be prepared using rotation gates.

$$|\psi_0>= |0>_b |0>_c |0>_a$$
$$R_\sigma|\psi_0>= |0>_b |0>_c |0>_a$$
$$|\psi_1>= |B>_b |0>_c |0>_a$$

Where $\hat{b}$ can be expressed in the basis of A,

$$\hat{b} = \sum_{j=0}^{N-1} b_j|u_j>$$

3.3 Quantum Phase Estimation (QPE)

Is an eigenvalue estimation algorithm with three components:

1. Superposition of controlled register
2. Hamiltonian Simulation

3. Inverse Quantum Fourier Transform

The circuit of QPE is generic for different problem with the exception of Hamiltonian simulation. In our case we wan to convert the Hamiltonian into a set of unitaries which we can implement.

$$\mathrm{H} = \begin{pmatrix} 0 & 0 & a & b \\ 0 & 0 & c & d \\ a & c & 0 & 0 \\ b & d & 0 & 0 \end{pmatrix},$$

$$\mathrm{H} = \begin{pmatrix} 0 & 0 & a & 0 \\ 0 & 0 & 0 & d \\ a & 0 & 0 & 0 \\ 0 & d & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & b \\ 0 & 0 & c & 0 \\ 0 & c & 0 & 0 \\ b & 0 & 0 & 0 \end{pmatrix}$$

$$\mathrm{H} = \hat{S} \begin{pmatrix} 0 & a & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & d \\ 0 & 0 & d & 0 \end{pmatrix} \hat{S} + \hat{S}\hat{C}\hat{S} \begin{pmatrix} 0 & b & 0 & 0 \\ b & 0 & 0 & 0 \\ 0 & 0 & 0 & c \\ 0 & 0 & c & 0 \end{pmatrix} \hat{S}\hat{C}\hat{S}$$

Where $\hat{S}$ is the Swap gate and $\hat{C}$ is the controlled not gate .Hence, the matrix H can be separated into 2 single sparse matrices and each variable can be present by a unitaries.

$$U_1 = \begin{pmatrix} e^{ia\hat{X}t} & 0 \\ 0 & e^{id\hat{X}t} \end{pmatrix},$$

$$U_2 = \begin{pmatrix} e^{ib\hat{X}t} & 0 \\ 0 & e^{ic\hat{X}t} \end{pmatrix},$$

The application of QPE with these unitaries evolves the state, to $|\psi_4>$, where $\tilde{\lambda_j} = N\lambda_j t/2\pi$.

$$|\psi_2>= \textstyle\sum_{j=0}^{N-1} B_j|U_j>|\tilde{\lambda_j}>|0>_a$$

3.4 Controlled Rotation of Ancillia Qubit

Rotating the ancillia qubit, based on $\tilde{\lambda_j}$ of control register allows the state to evolve such that the coefficient for the term where the ancilla is $|1>$, has a factor of $1/\tilde{\lambda_j}$. A_0 is the normalizing constant.

$$|\psi_3>= A_0 \textstyle\sum_{j=0}^{N-1} C(B_j/\tilde{\lambda_j})|U_j>|\tilde{\lambda_j}>|1>_a$$
$$\text{where, } A_0 = 1/\sqrt{\textstyle\sum_{j=0}^{N-1} |CB_j/\tilde{\lambda_j}|^2}$$

3.5 Uncomputation

To measure $\hat{x}$ from $|\psi_3>$ we need to factorize the control bits but this is not possible as the control and input qubit are entangled. As such we need to perform a reverse QPE to get the $|\psi_3>$.

$$|\psi_4>= A_1|\hat{X}>_b|0>_c|1>_a$$
$$\text{where, } A_1 = 1/\sqrt{\textstyle\sum_{j=0}^{N-1} |CB_j/\lambda_j|^2}$$

4 Example

Let,

$$A = \begin{bmatrix} 1 & -1/3 \\ -1/3 & 1 \end{bmatrix},$$
$$\hat{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

Hence,

$$H = \begin{pmatrix} 0 & 0 & 1 & -1/3 \\ 0 & 0 & -1/3 & 1 \\ 1 & -1/3 & 0 & 0 \\ -1/3 & 1 & 0 & 0 \end{pmatrix},$$
$$\hat{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

Hence from this we can compute $\hat{X}$ using the Unitary,

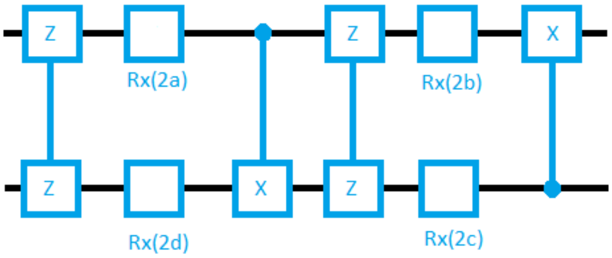

Figure 2: Unitary

4.1 Error

Classical Linear System Solution, for b = 0.

$$|0, x >= C^{-1}|b, 0>$$
$$|x >= 0.75|0> +0.25|1>$$

Quantum Linear System Solution, using probabilities found from "qasm simulator" for 10,000 shots.

$$|1, 0, x >= 0.259^{0.5}|1, 0, 0> +0.031^{0.5}|1, 0, 1>$$
$$|x >= 0.508|0> +0.176|1>$$

Increasing coefficients to compare , $|x >= 0.743|0> +0.257|1>$

$$Error = \sqrt{(0.743 - 0.75)^2 + (0.257 - 0.25)^2)}$$
$$\text{Error} = 0.01$$

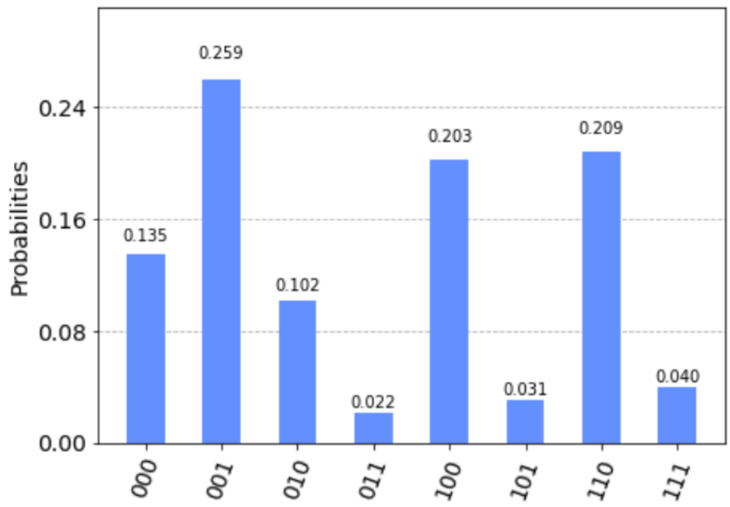

Figure 3: Simulation Results

5 Improvements and Limitations

The unitary in Fig 2, assumes that the matrix A is a real, non-zero, non-Hermitian matrix. It also imposes that the each element "e" in the matrix, satisfies the relation,

$$-1 > e > 1$$

and the Eigen Values, λ_j is sufficiently large. This introduces the advantage used, ignoring of second order terms from Lie-Trotter Expansion, which significantly reduces number of gates in the circuit.

The Hamiltonian simulation method can be extrapolated to include imaginary elements in the Matrix A, and the corresponding unitary has Rotations about the y axis. The number of terms in the Unitary becomes 4, when Lie-Trotter Expansion is applied to separate the unitaries in terms of individual gates, the impact of second order terms are too large to ignored, even if the real and imaginary elements satisfy the inequality for "e". Hence the algorithm discussed above is unable to resolve a system of linear equations for which the Matrix is non-Hermitian and complex.

6 Conclusion

The discussed HHL algorithm can be applied on Linear System which have a non-zero,real arbitrary matrix A, giving a exponential speedup compared to classical computation in the best case. Compared to the original HHL algorithm, for a Linear System of Size 2, the number of qubits used increases from 4 form 7, and the number of gates in the worst case are increased exponentially. The improvement comes form the fact, that the original algorithm had to be reconstructed for each new Linear System, and required prior knowledge of this systems Eigen-Values, which makes the algorithm redundant.

7 References

1. https://arxiv.org/pdf/0811.3171.pdf

2. https://arxiv.org/pdf/2108.09004.pdf

3. $https://github.com/hywong2/HHLExample/blob/main/HHLHectorWong.ipynb$