Solving system of linear equations using Quantum Computers

Rohan Bhatia
Applied Physics Department, Delhi Technology University, New Delhi, India

July 4, 2021

Abstract

Quantum Computing is the forerunner of all data handling disciplines and opens new possibilities for the future of big data and analytics. Machine Learning algorithms are limited by the computational power of classical computers of today. In this paper a Quantum Computing algorithm has been explained and applied to a sample problem which can be extended to solve Machine Learning problems further. Quantum Computing based HHL algorithm can be used to solve system of linear equations. The algorithm encodes the equation parameters into vector states and uses quantum mechanical tools such as Quantum Phase Estimation to find the inverse of the constant matrix of the given system. HHL algorithm provides an exponential speed-up over the fastest classical algorithms. In the project we explain the algorithm and then solve a system of linear equations using both HHL and classical algorithms like Gauss-elimination method using Python.

Keywords: HHL, Quantum Computing, Linear Regression, Machine Learning

1 Introduction

Linear equations can be found in almost all fields. As the size of data which defines the equation set is increasing at an exponential rate, classical computers are getting overwhelmed. For N linear equations with N unknown variables the best classical algorithms(conjugate gradient method) have a complexity of the form $O(Nk)$ where k is the condition number of the matrix(it shows how far the image is from being inverted).
The Quantum Algorithm that we discuss will be able to solve the given equations in $poly(logN, k)$, which is an exponential speed up over the classical algorithms.
The problem of N linear equations can be expressed by using matrix representation,

$$\begin{bmatrix} A_{11} & A_{12} & ... & ... \\ A_{21} & ... & & \\ ... & & & \\ A_{N1} & A_{N2} & ... & A_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ .. \\ \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ .. \\ \\ b_N \end{bmatrix} \tag{1}$$

or

$$A\vec{x} = \vec{b} \tag{2}$$

The above equation can be solved by multiplying both the sides with A^{-1} which provides us with the solution vector which is given by $\vec{x}$.

$$\vec{x} = A^{-1}\vec{b} \tag{3}$$

we aim to use this in order to solve our system of linear equations.

2 Prerequisites

The HHL algorithm uses a few quantum computing tools such as QPE(quantum phase estimation) , we will be going over them once.

2.1 Quantum Fourier Transform(QFT)

Recalling the frequently used Discrete Fourier Transform (DFT) from classical physics, it is required to decompose a signal (i.e. a function) into its frequency domain and reveal periodicity in input data. The discrete Fourier transform of a real sequence of numbers will be a sequence of complex numbers of the same length. The DFT can be considered as a square invertible matrix D of dimensions N, where

$$D_{jk} = \frac{1}{\sqrt{N}}\omega^{jk},$$

and ω^{jk} is one of the N roots of unity, i.e. $e^{\iota 2\pi jk/N}$.
The columns of this matrix form a **Fourier Basis** as they follow the property of othogonality. A **Basis**, in general sense, is a set of N linearly independent vectors in an N dimensional space and the vectors forming the basis are called **Basis States**. In quantum computing vectors are represented in different basis, the main two basis are the Z basis states :$|0\rangle$ and $|1\rangle$ and the X basis states : $|+\rangle$ and$|-\rangle$. The Z basis is generally referred to as the computational basis and the X basis is the Fourier basis.
A better way to visualize X basis and Z basis is through a bloch sphere. All the quantum states in the xy plane are in the X basis and all the quantum states in the xz plane are in the Z basis. **The main aim of QFT is to transform encoded quantum states from computational basis to Fourier basis.**
The encoded state $|x\rangle$ can be converted to $|\tilde{x}>$ where We define $QFT|x\rangle$ as

$$|\tilde{x}\rangle = QFT|x\rangle \tag{4}$$

Now $|x\rangle$which is defined over n qubits can be expressed as

$$|x\rangle = |x_1 x_2 ... x_n\rangle \tag{5}$$

the QFT of the above quantum state is defiened as

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \frac{x.y}{N}} |y\rangle \tag{6}$$

where $N = 2^n$, n is the number of qubits used, y is the index for summation and y and x in the power of e are the decimal conversions of the vectors $|x>$ and $|y>$.
Another way to write the above equation is

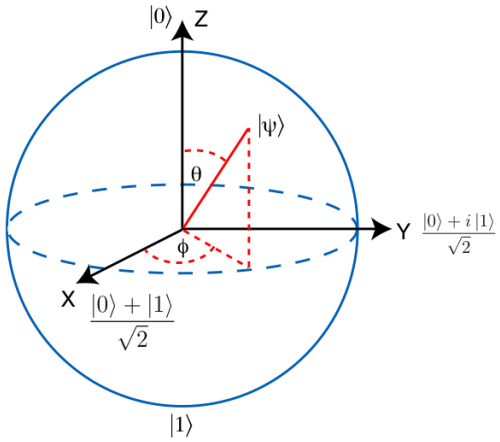

Figure 1: Bloch Sphere

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N}}(|0\rangle + e^{\frac{2\pi ix}{2}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi ix}{2^2}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi ix}{2^3}}|1\rangle) \otimes ...(|0\rangle + e^{\frac{2\pi ix}{2^n}}|1\rangle) \tag{7}$$

Now, if you look closely at all the terms in the above equation, each term is in the basis $|+\rangle$ and $|-\rangle$.

2.1.1 Quantum Circuit for implementation of QFT

The circuit that implements QFT uses two gates. First one is the Hadamard gate,

$$H|x_j >= \frac{1}{\sqrt{2}}(|0> + e^{\frac{2\pi i}{2}x_j}|1>) \tag{8}$$

the second gate is controlled UROT gate, which referes to unitary rotation

$$UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\frac{2\pi i}{2^k}} \end{bmatrix} \tag{9}$$

and as it is a controlled rotation the state of the target qubits depends upon the state of the controlled rotation

$$CROT_k|0x_j >= |0x_j > \tag{10}$$

$$CROT_k|1x_j >= e^{\frac{2\pi i}{2^k}}|1x_j > \tag{11}$$

The circuit given below produces QFT of the input quantum state. We will be breaking down the circuit after the application of each unitary transformation

at (1), the quantum state is given

$$\psi_1 = H|x_1 > \otimes |x_2.....x_n >= \frac{1}{\sqrt{2}}(|0> + e^{\frac{2\pi i}{2}x_1}|1>) \otimes |x_2.....x_n > \tag{12}$$

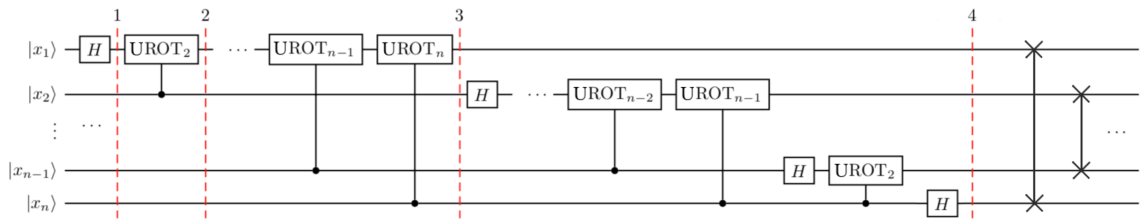

Figure 2: Circuit for Implemetation of QFT

at (2),

$$\psi_2 = \frac{1}{\sqrt{2}}(|0> + e^{\frac{2\pi i}{2}x_1 + \frac{2\pi i}{2^2}x_2}|1>) \otimes |x_2.....x_n> \tag{13}$$

after n UROT gates the state at (3),

$$\psi_3 = \frac{1}{\sqrt{2}}(|0> + e^{\frac{2\pi i}{2}x_1 + \frac{2\pi i}{2^2}x_2 + ... + \frac{2\pi i}{2^n}x_n}|1>) \otimes |x_2.....x_n> \tag{14}$$

by taking out $\frac{1}{2^n}$ and converting to the decimal representation of the x we get

$$\psi_3 = \frac{1}{\sqrt{2}}(|0> + e^{\frac{2\pi i}{2^n}x}|1>) \otimes |x_2.....x_n> \tag{15}$$

Similarly after applying the other gates we get

$$|\tilde{x}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x}{2^n}}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x}{2^{n-1}}}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x}{2^{n-2}}}|1\rangle) \otimes ... \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x}{2}}|1\rangle) \tag{16}$$

In the end the swap gate is used in order to arrange the terms in order.

2.2 Quantum Phase Estimation(QPE)

Quantum Phase Estimation is one of the most important quantum algorithms. It is used to find the eigen value of unitary transformation when it is of the form

$$U|\psi> = e^{2\pi i\theta}|\psi> \tag{17}$$

When the input is $|\psi>$ and unitary transformation U acts on it we will be able to find the value of θ .

2.3 Circuit for implementation of QPE

We will be using similar gates for QPE as we did for QFT. We will also be using n work qubits for this.
The initail qunatum state is given by

$$|\psi_0> = |0>^{\otimes n} |\psi> \tag{18}$$

We start by creating a unifrom superpostion state at (1)

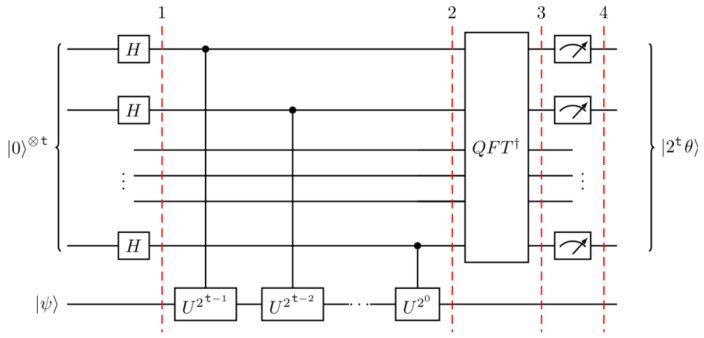

Figure 3: Circuit for Implemetation of QPE

$$|\psi_1>=\frac{1}{\sqrt{2^n}}(|0>+|1>)^{\otimes n}|\psi> \tag{19}$$

Then controlled rotations are applied which change the state of the ancillia qubits.The applications of t-1 U gives us

$$U^{t-1}|\psi>=e^{2\pi i 2^{t-1}\theta}|\psi> \tag{20}$$

similarly applying all the other Unitary operators we get (2)

$$|\psi_2\rangle=\frac{1}{\sqrt{2^t}}(|0\rangle+e^{2\pi i\theta 2^{t-1}}+|1\rangle)\otimes(|0\rangle+e^{2\pi i\theta 2^{t-2}}|1\rangle)\otimes(|0\rangle+e^{2\pi i\theta 2^{t-3}}|1\rangle)\otimes...(|0\rangle+e^{2\pi i\theta 2^{0}}|1\rangle)\otimes|\psi> \tag{21}$$

the value of $|\psi_2>$ is symmetrical to QFT of $|\psi>$ so now we apply $QFT^\dagger$ to the ancillia qubits and get $|2^t\theta>$.
We will be directly using QPE in our process of calculating the solution to linear equations.

3 HHL Algorithm

We start by scaling the system such that $\vec{b}$ and $\vec{x}$ are converted to normalized vectors $|b>$ and $|x>$. So our problem statement changes to

$$A|x>=|b> \tag{22}$$

As all the Quantum Gates are hermitian operators, A is also a Hermitian operator($A=A^\dagger$) and by using the **spectral decomposition** we can represent A as

$$A=\sum_{j=0}^{N-1}\lambda_j|u_j><u_j| \tag{23}$$

where λ_j and $|u_j>$ are the eigen vector and eigen value of the A. Now, we want to calculate the value of A^{-1} .

$$A^{-1}=\sum_{j=0}^{N-1}\frac{1}{\lambda_j}|u_j><u_j| \tag{24}$$

this can be proved very easily as $(|u_j >< u_j|)^{-1} = |u_j >< u_j|$.
Now, as $|u_j>$ acts as a complete basis state we can express $|b>$ in terms of $\sum |u_j>$.

$$|b>=\sum_{j=0}^{N-1} b_j|u_j> \tag{25}$$

using equation (24) and (25) we will get the value of

$$|x>=\sum_{j=0}^{N-1} \frac{b_j}{\lambda_j}|u_j> \tag{26}$$

This is our solution vector and our goal is to develop a circuit which outputs $|x>$.

3.1 Circuit for HHL algorithm

One of the methods to implement quantum computing algorithm is through gate based quantum computing so lets take a look at the circuit for HHL algorithm.
The qubits have been divided into three parts: n_b which is for loading $|b>$, n_l to store the eigen values and the last quantum register is used as an ancillia qubit.
The quantum state at each interval is given by

$$\boxed{|\psi_0>=|0>|0>^{\otimes n_l}|0>^{\otimes n_b}} \tag{27}$$

Steps involved:
1.Load $|b>$ - We start by loading $|b>$ into the n_b. The gates required for this depends upon the value of b.

$$\boxed{|\psi_1>=|0>|0>^{\otimes n_l}|b>} \tag{28}$$

2. QPE - The U is chosen to be

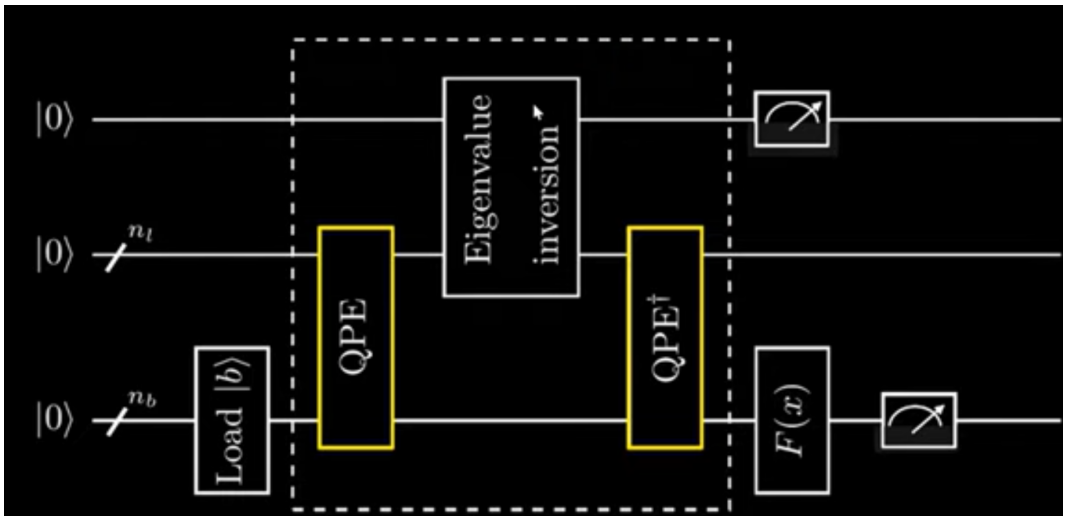

Figure 4: Quantum Circuit

$$U = e^{iAt} \tag{29}$$

which can be expanded as

$$U = \sum_{j=0}^{N-1} |u_j >< u_j| \tag{30}$$

The total quantum phase estimation can be depicted as

$$U = \Sigma_{\tau}^{N-1} |\tau\rangle\langle\tau|^M \otimes e^{iAt_0\tau/N}$$

This converts the quantum state to

$$\boxed{|\psi_2 >= \sum_{j=0}^{N-1} b_j |0> |\lambda_j >^{\otimes n_l} |u_j >} \tag{31}$$

3.Eigen Value Inversion - A controlled rotation of the form $R_y(\lambda^{-1})$ is done on the Ancillary qubit controlled by Matrix register. The controlled register later produces the required λ_j^{-1} values. The R_Y gate is a basic gate that is represented by

$$e^{-i\frac{\theta}{2}\sigma_y} = \begin{bmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{bmatrix}$$

where $\theta = cos^{-1}\frac{C}{\lambda_j}$ and C is the scaling factor. The operation of R_y gives us the final state -

$$\boxed{|\psi_3 >= \Sigma b_j |u_j\rangle^{n_b} |\lambda_j\rangle^{n_l} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}}|0\rangle + \frac{C}{\lambda_j}|1\rangle\right)} \tag{32}$$

4. Quantum Phase Estimation dagger - $QPE^{\dagger}$ is implemented in order to make the n_l register 0.

$$\boxed{|\psi_4 >= \Sigma b_j |u_j\rangle^{n_b} |0\rangle^{n_l} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}}|0\rangle + \frac{C}{\lambda_j}|1\rangle\right)} \tag{33}$$

5. Measuring the auxilary qubit - We measure the auxiliary qubit if the outcome is $|0>$ we discard it.
If the outcome is $|1\rangle$ we get the outcome state as

$$\boxed{|\psi_4 >= C\Sigma \frac{b_j}{\lambda_j} |u_j\rangle^{n_b} |0\rangle^{n_l}} \tag{34}$$

where C is the normalization constant and can be removed as global phase does not have any effect on the output of a qubit.
That is the state we were looking to achieve.

3.2 Implementation using IBM Quantum

Lets look at a specific case.
For convinience the input hermitian matrix A is chosen as;

$$A = \frac{1}{4}\begin{bmatrix} 15 & 9 & 5 & -3 \\ 9 & 15 & 3 & -5 \\ 5 & 3 & 15 & -9 \\ -3 & -5 & -9 & 15 \end{bmatrix}$$

as its Eigenvalues calculated from classical methods are known to be 1, 2, 4, 8, i.e. they are of the form 2^j.
b is chosen in such a fashion that it can be easily encoded,

$$b = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

The **number of qubits is taken to be 7**. q_0 is the ancillia qubit, q_1 - q_4 are used as n_l and $q_5 - q_7$ is used as n_b.

Next, Quantum Phase Estimation subroutine is applied on the Matrix register. First a superposition is created by applying Hadamard Gate on q_1-q_6. The Unitary gate in our system is of the form e^{iAt} and the controlled gates applied on will be $e^{iAt/16}$, $e^{iAt/8}$, $e^{iAt/4}$ and $e^{iAt/2}$. These operators have to be constructed from the basic operators provided by Qiskit.

3.2.1 Gates Used

1.cx- The controlled x gate flips the target qubit if the control qubit is in state one and does not do anything if the state is not one.

$$CX\ q_0, q_1 = I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

2.Hadamard- Is used to create a uniform superposition of quantum state.

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

3.u3- It is the genreal quantum gate and can be used to represent any unitary by controlling the values of input. It is given by the unitary given below

$$U3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta) & -e^{i\lambda}\sin(\theta) \\ e^{i\phi}\sin(\theta) & e^{i(\phi+\lambda)}\cos(\theta) \end{pmatrix}$$

4.ccx- Known as the toffoli gate and the only thing different from cx is that we have added another control.

$$CCXq_0, q_1, q_2 = I \otimes I \otimes |0\rangle\langle 0| + CX \otimes |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

5.cu3- When a u3 gate is controled by another we call it cu3.

$$CU3(\theta, \phi, \lambda)\ q_0, q_1 = I \otimes |0\rangle\langle 0| + U3(\theta, \phi, \lambda) \otimes |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & -e^{i\lambda}\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & e^{i\phi}\sin(\theta) & 0 & e^{i(\phi+\lambda)}\cos(\theta) \end{pmatrix}$$

3.2.2 Steps Followed

1. Initial State - We start with the same initial state.

$$\psi_1 >= |0> |0>^{\otimes n_l} |0>$$

2.Load b- b can be represented as

$$|b\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

where each vector denotes an eigenvector of A.

3.QPE- Quantum Phase Estimation performs the mapping and gives a final state as -

$$\sum \beta_j |u_j\rangle^I |\frac{\tilde{\lambda_j} t_0}{2\pi}\rangle$$

where $\tilde{\lambda_j} s$ are the approximate eigenvalues of A, and t_0 is set equal to 2π.
The Quantum Phase Estimation produces the state

$$\frac{1}{2}|0001\rangle|u_1\rangle + \frac{1}{2}|0010\rangle|u_2\rangle + \frac{1}{2}|0100\rangle|u_3\rangle + \frac{1}{2}|1000\rangle|u_4\rangle$$

where $\frac{1}{2}|u_j\rangle$ represent the eigenvectors of A

$$|u_1\rangle = -|00\rangle + |01\rangle + |10\rangle + |11\rangle$$

$$|u_2\rangle = +|00\rangle - |01\rangle + |10\rangle + |11\rangle$$

$$|u_3\rangle = +|00\rangle + |01\rangle - |10\rangle + |11\rangle$$

$$|u_4\rangle = +|00\rangle + |01\rangle + |10\rangle - |11\rangle$$

4.Controlled Rotation- A controlled rotation of the form $R_y(\lambda^{-1})$ is done on the Ancillary qubit controlled by Matrix register. The controlled register later produces the required λ_j^{-1} values.

The R_Y gate is a basic gate that is represented by

$$e^{-i\frac{\theta}{2}\sigma_y} = \begin{bmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{bmatrix}$$

where $\theta = cos^{-1}\frac{C}{\tilde{\lambda_j}}$ and C is the scaling factor.

The operation of R_y gives us the final state -

$$\Sigma\beta_j|u_j\rangle^I|\lambda_j\rangle^M \left(\sqrt{1-\frac{C^2}{\tilde{\lambda_j^2}}}|0\rangle + \frac{C}{\tilde{\lambda_j}}|1\rangle\right)^A$$

Here the prior knowledge of Eigenvalues of A is utilized to design optimal R_Y gates. C is set as $8\pi/2^r$.

5. Inverse QPE and measurement - Now the Inverse Quantum Phase Estimation is done to to set the matrix register back to $(|0\rangle^{\otimes n})^M$ and the rest of the state is

$$\sum_{j=1}^{N}\beta_j|u_j\rangle^I \left(\sqrt{1-\frac{C^2}{\tilde{\lambda_j^2}}}|0\rangle + \frac{C}{\tilde{\lambda_j}}|1\rangle\right)^A$$

Next the coeffecient of state $|1\rangle$ is selected in order to obtain λ_j^{-1} in the form of

$$C\sum_{j=1}^{N}\frac{\beta_j}{\tilde{\lambda_j}}|u_j\rangle^I$$

.

The desired inverse of the matrix A can be written as $\sum_j \frac{1}{\lambda_j}|u_j\rangle\langle u_j|$. Thus, A^{-1} matches

$$\sum_{j=1}^{N}\frac{\beta_j}{\tilde{\lambda_j}}|u_j\rangle^I$$

Here since $C = 8\pi/2^r$ and λ_j are exactly calculated by QPE as 2^j, the final state of Ancilla register is

$$\frac{8\pi}{2^r}\sum_{j=1}^{4}\frac{1/2}{2^{j-1}}|u_j\rangle$$

3.2.3 Quantum Circuit

The quantum circuit for our case is given on page 11

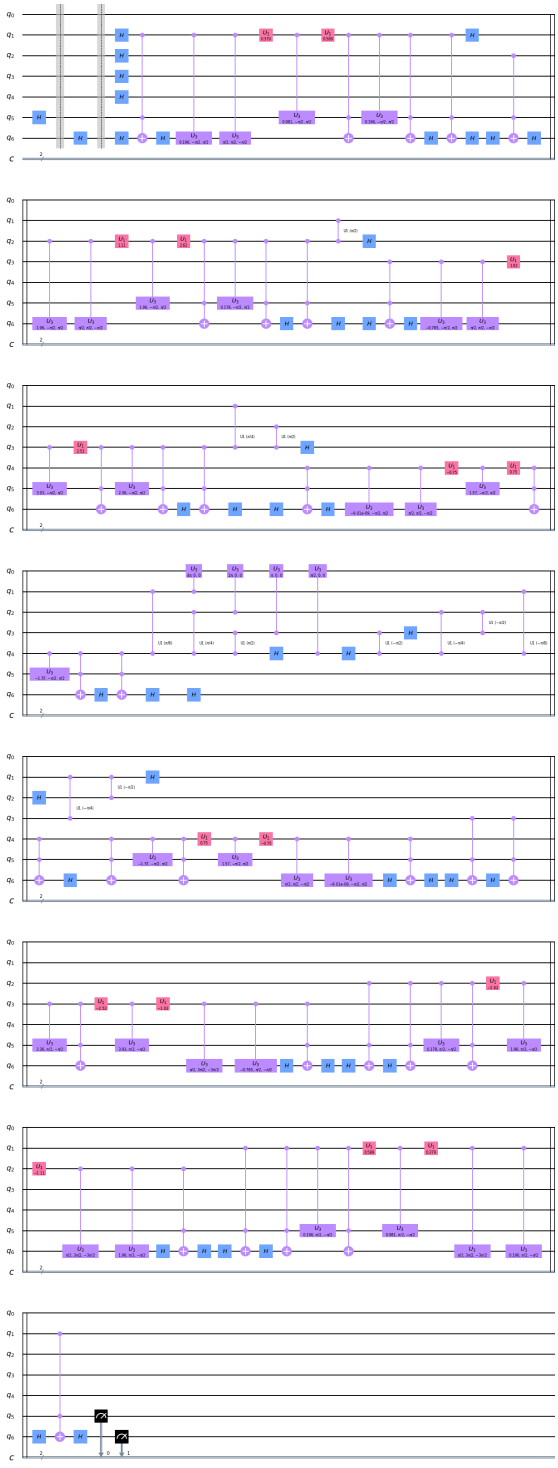

Figure 5: Implementation of our case

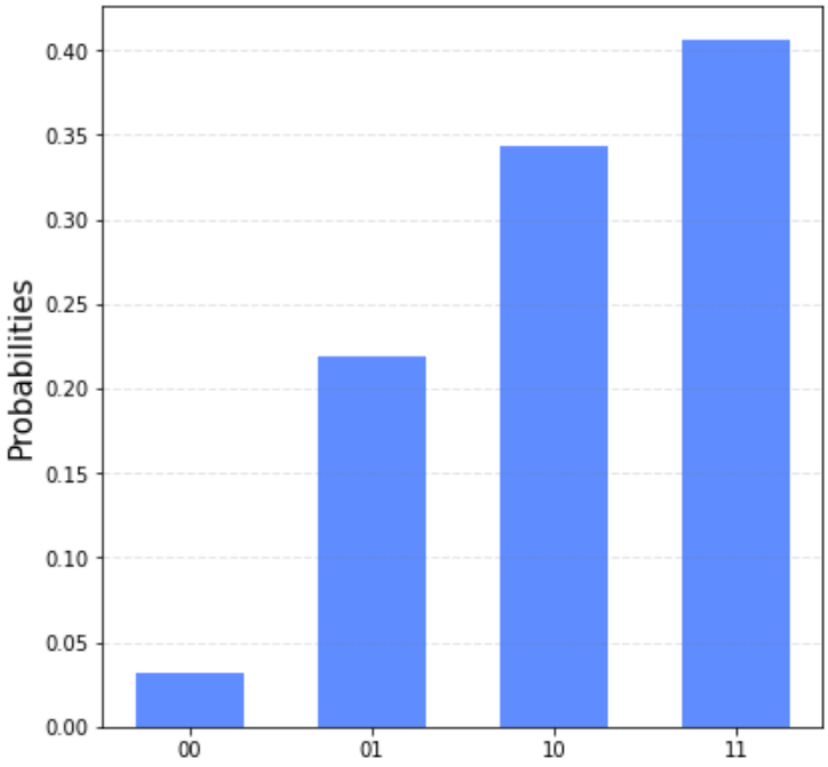

Figure 6: Histogram Plot

3.2.4 Results and discussions

Mathematically the solution of the equation is

$$x = \frac{1}{32}\begin{bmatrix} -1 \\ 7 \\ 11 \\ 13 \end{bmatrix}$$

Measuring the $|1\rangle$ state of Input register, i.e. qubit q_6 gives a solution of the form

$$x = \frac{8\pi}{2^r}\sum_{j=1}^{4}\frac{1/2}{2^{j-1}}|u_j\rangle$$

which should be proportional to the expected solution.

The histogram gives the probability of each state

$$Pr(|00\rangle) = 0.03$$
$$Pr(|01\rangle) = 0.21$$
$$Pr(|10\rangle) = 0.34$$
$$Pr(|11\rangle) = 0.40$$

Clearly, $\Sigma Pr = 1$

The square roots of these Probabilities give the Amplitude, which are values proportional to the vector elements of the required solution.

$$0.0312 \times \begin{bmatrix} 1 \\ 7 \\ 11 \\ 13 \end{bmatrix}$$

3.3 Limitations

- For large systems, the first step, i.e. Hamiltonian Simulation of matrix A is difficult and requires high computational complexity. Only s-sparse Hamiltonians can be simulated easily at present.
- Pre-preparation is required on the input state $|b\rangle$
- For larger matrices, the HHL algorithm does not compute x exactly, it only gives us an approximation of how a unitary matrix would act upon x. To read-out the output, we need to calculate the expectation value, i.e. $\langle x|M|x\rangle$.
- Thus the major limitation of HHL algorithm is the probabilistic treatment of solution and post-preparation on the measured states, required to extract the exact solution.
- For larger systems, a larger number of qubits are required, i.e. access to more and more advanced quantum computers is required.

4 Conclusion

1. Quantum computers can be leveraged to solve systems of linear equations. Which can be found in a lot real life scenarios.
2. This method can also be used to solve differential equations which has been reduced to the matrix form using finite difference method.
3. It is the base of Quantum Machine learning and has shown promising results in Linear Regression.
4. The complexity is considerably less and using this we will be able to take on bigger problems which seem impossible for classical computers.
5. As we get better quantum computers we will be able to solve equations more efficiently.

References

1. Harrow, A.W., Hassidim, A., Lloyd, S.: 'Quantum Algorithm for Linear Systems of Equations' Phys. Rev. Lett., 2009, 103, (15).
2. Cao, Y., Daskin, A., Frankel, S., Kais, S.: 'Quantum circuit design for solving linear systems of equations' Mol. Phys., 2012, 110, (15-16), pp. 1675–1680.
3. Qiskit Textbook Chapter 4.1.1