HHL Algorithm Implementation

蔡維禎 李宥頡 鄭婷予

B05901004 B06901147 B05502048

**Abstract**

Solving linear systems of equation is an important problem and also crucial for many other useful problems to have it as a subroutine. The Harrow Hassidim Lloyd Algorithm provide us a way to solve this problem in QC. In this paper our goal is to learn how the algorithm work, try to implement the HHL algorithm in Qiskit and tune some of the parameters to observe the effect.

**1 Introduction**

Linear system of equation is described as

, where A is an N by N Hermitian matrix, is an unit vector, and is the solution we intend to find. In the algorithm, is represented by a quantum state , where | is the eigenstates of A, then is apply to by using the technique of quantum phase estimation to obtain eigenvalue of A. After that, rotation gate is used in order to map to . Then, after further uncomputation of the phase estimation, a measurement of |1> on the first qubit let us obtain in |b> register.

**2 The Harrow Hassidim Lloyd Algorithm**

From the linear equation, we can derive the answer by computing inverse of A.

In the HHL algorithm, quantum state is decomposed by the eigenstates of matrix A as Since A can be diagonalized, we can re-written A to be . It is then easy to show that the inverse of A is . Thus, can be expressed as .

The following three sections shown in figure 1 will show how the quantum circuit of HHL can lead to the answer .

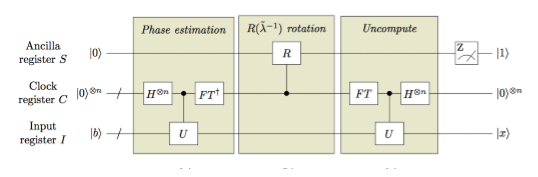


FIG. 1. HHL Algorithm Schematic [3]

1. Phase Estimation

We have learnt from class that quantum phase estimation performs where |u> is an eigenvector of a unitary operator U with eigenvalue . The input of HHL algorithm replace |u> by the superposition of eigenvectors that equals to |b>, i.e. . Thus, at the end of the phase estimation, the state becomes

where is the eigenvalue of A accurate to t bits, and is often chosen to be to eliminate the factor of .

In this part, the U gate is . The decomposition of it will be briefly explained later in this report.

1. R) rotation

To obtain the state

in the ancilla register S, mapping of to is needed. By the technique of small angle approximation, the angle in the rotation operation can be decided to be , and it only be activated when its control bit is |1>, which occur only when the state in register C is |.

1. Uncomputation

In the third part, the uncompute process is applied to the register C and register I, leaving the final state to be

A measurement on ancilla bit S is performed in the end to generate state

in register I when the measurement observes |1> in the ancilla. The state is proportional to

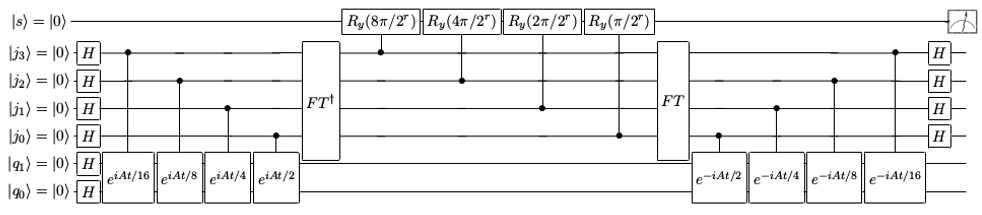


FIG. 2. Quantum circuit for solving a 4×4 system of linear equation . The top qubit is the ancilla qubit. The four qubits in the middle stand for the Clock register C. The two qubits at the bottom form the Input register I and two Hadamard gates are applied on them to initialize the state |b>. [3]

by a factor C, so the solution

is just the coefficient of the final state in register I with some scaling factor.

**3 Example**

Consider to solve with Hermitian matrix

and . The eigenvalues of A are , , , with corresponding eigenvectors

]

Thus, |b> can be expressed as .

The quantum circuit for solving this problem by HHL is demonstrated in figure 2. After the quantum phase estimation part, the state in register will evolve to eigenvalues of A as

The next is applying rotation gate to the ancilla bit, and after finishing the uncomputation and measurement, the state produced in register I will be

The normalized state is

which is proportional to the solution

and thus we obtain the answer.

**4 Qiskit Implementation**

The full circuit we use is the same as figure 2.

1. Some special gate:

Rzz(It’s not the qiskit Rzz!):

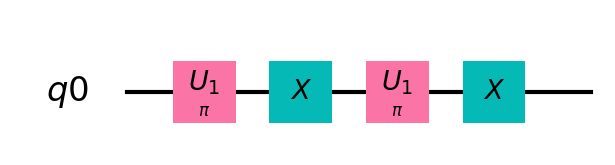


FIG. 3.

1. decomposition:

(1) Group Leader Optimization Algorithm [1]: We can decompose into the form in Fig 4, where the angle of every gate is obtained by the following optimization method: we first set the parameters to the ones given in paper, then we compute the error between the constructed circuit and the ideal matrix. use scipy.optimize.minimize to obtain better parameter and repeat the process.

(2) Suzuki expansion[2]: we can express A as , where are all pauli matrices where is easy to be computed. Notice that when A is more complex, the decomposition result may divert more from our desirable result.

1. Rotation parameter r

Observing the circuit implementation, we’d notice that there’s an important adjustable parameter r to change the angle of Ry. We’ll discuss this in detail in the ‘Experiment Result Session’

**5 Experiment Result**

We’ve simulated our circuit in both qasm simulator and statevector simulator, which doesn’t differ much in terms of the results. The result on an actual machine is not tested, since most of our circuits might be too large to run on those machine.

In comparison of different selection of j, we can see that as j become larger, it indeed became easier to get the right results.

Compare the 2 expansion style we can see that the group leader optimization are more tolerate to the difference of the value r. This may result from the usage of optimization process in

the expansion process. As a result, it would take more time to generate the circuit.

Due to the intolerance of different r selection, the Suzuki expansion style, being the one used in qiskit standard HHL implementation, it use another reciprocal algorithm other than the one with parameter r provided over here.

一張含有 時鐘 的圖片

自動產生的描述

FIG. 4.

|  |  |  |  |
| --- | --- | --- | --- |
| j | Fidelity- r graphs | j | Fidelity- r graphs |
| 2 |  | 3 |  |
| j | Fidelity- r graphs | | |
| 4 |  | | |

Table 1. graphs for A = [[1, -1/3], [-1/3, 1]], b = [1, 0]. Probability(orange) = probability of measuring ‘1’ in the ancilla bit, which could lead to correct solution. Fidelity(blue) = closeness of the exact solution and experiment result.

|  |  |  |  |
| --- | --- | --- | --- |
| j | Suzuki expansion | j | Suzuki expansion |
| 2 |  | 3 |  |
| j | Suzuki expansion | j | Group Leader expansion |
| 4 |  | 4 | 一張含有 文字, 地圖 的圖片  自動產生的描述 |

Table 2. graphs for A = [[15/4, 9/4, 5/4, -3/4], [9/4, 15/4, 3/4, -5/4], [5/4, 3/4, 15/4, -9/4], [-3/4,-5/4, -9/4, 15/4]], b = [1/2, 1/2, 1/2, 1/2. Probability=orange, Fidelity=blue.

References

[1] Anmer Daskin, Sabre Kais, *Group leaders optimization algorithm*, Molecular Physics Vol. 109, No. 5, 10 March 2011, 761–772.

[2] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders, *Eﬃcient quantum algorithms for simulating sparse Hamiltonians*,

arXiv:05081392v2 [quant-ph].

[3] Sanchayan Dutta, Adrien Suau, Sagnik Dutta, Suvadeep Roy, Bikash K. Behera, and Prasanta K. Panigrahi, *Demonstration of a Quantum Circuit Design Methodology for Multiple Regression* , arXiv:1811.01726v2 [quant-ph].