2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Demonstration of quantum linear equation solver on the IBM qiskit platform

Wen Ji[1], Xiangdong Meng[2]
1. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, P.R. China
2. Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450002, P.R. China
Email: ts19060065a31@cumt.edu.cn

***Abstract*—HHL algorithm is a basic quantum algorithm, which is mainly used to solve linear equations. It has a wide range of potential applications in science, engineering, finance, economics and other fields. Although the current scale of quantum computers is small, we can still demonstrate basic quantum algorithms. In this paper, taking the HHL quantum algorithm as an example, the quantum circuit of the HHL quantum algorithm corresponding to 4 qubits and 7 qubits is given, and the simulation verification is carried out on the IBM qiskit development platform. The results show that when the input matrix is a second-order matrix, the fidelity is very high, but when the input is a fourth-order matrix, the probability that the result is close to the real value is very high only when the sparsity of the matrix is relatively strong. However, when the sparsity of the input matrix is very weak, the result will be very different from the real value.**

***Index Terms*—HHL algorithm; quantum algorithm; qiskit; linear equations;**

I. Introduction

Quantum computing is a calculation method that follows the law of quantum mechanics, which is essentially the change of the state of quantum system, that is, a series of unit transformations. The design of the quantum algorithm is to construct these unitary operators carefully so that the state of the quantum system called quantum register evolves step by step and finally reaches the desired state. The so-called required state is the ground state corresponding to the correct output, and the probability determined by its probability amplitude should be sufficiently large; in other words, once the system state is measured, it will collapse to the ground state corresponding to the correct output with a sufficiently large probability. In other words, quantum computing is probabilistic rather than deterministic, except in very special cases. This is different from the certainty of classical calculation. In quantum computing, the algorithms of all computing problems must be reflected by a series of unitary operators. Unitary operator plays a fundamental role in quantum computing. Any quantum algorithm consists of a series of unitary operators from beginning to end. The unitary operator is composed of some standard basic units, which are quantum gates. Quantum gates are usually expressed by unitary operators.The logic gate in classical physics acts on one or two bits, and the common quantum gate in quantum physics also acts on one or two qubits. This also means that these quantum gates can be expressed by 2×2 or 4×4 unitary operaters.

Harrow et al put forward the HHL algorithm in 2009 [1]. The proposal of HHL algorithm makes scholars really start the in-depth study of quantum machine learning. In recent years, many machine learning algorithms based on HHL algorithm have emerged. For the realization of quantum acting for solving linear equations, see in [2,3]. In [4,5], the basic principles and some basic knowledge of the HHL algorithm are introduced. Compared with traditional classic algorithms, under certain conditions, the speed of HHL algorithm can reach exponential level. In the case of high-dimensional data or large amounts of data, the time required for the algorithm is greatly reduced. In [6], it introduces how to implement HHL algorithm on quantum computer.

HHL algorithm solves the problem of solving linear equations. It is well known that solving linear systems is the core of most scientific, engineering, financial and economic applications. Linear equations are usually used to solve differential equations and partial differential equations. Linear equations are also commonly used in regression analysis. For example, the quantum algorithm for solving inhomogeneous linear partial differential equations is introduced in [7]. In [8], how to use superconducting quantum processor to solve linear equations is introduced. However, with the passage of time and the continuous development of science, the data set of linear equations is getting larger and larger, so more and more data need to be processed to solve linear equations, and it takes more and more time. The fastest algorithm for solving N linear equations on a classical computer also requires $O(N)$ time complexity. Moreover, in some cases, we need some functions of the system of equations rather than all its solutions. In these cases, the calculation of classical computers will waste a lot of unnecessary time. In contrast, the new quantum algorithm can only cause the calculation time proportional to the logarithm of the variable N in some cases. The result of the HHL algorithm is the expected value of the operator associated with the solution of the system of linear equations, not the solution of the system itself.

In this paper, the HHL algorithm is briefly introduced at first, then the experiments of different input matrices are carried out on the quantum computing platform of IBM, and the quantum circuit diagram of solving two-solution linear equations by using HHL algorithm is given in figure 2. Finally, the fidelity of the results obtained by using qiskit to realize HHL algorithm with different input matrices is summarized.

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00086

II. The HHL Algorithm

The definition of general linear equations satisfies the following conditions:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \cdots + a_{1n} \cdot x_n = b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \cdots + a_{2n} \cdot x_n = b_2 \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \cdots + a_{mn} \cdot x_n = b_m \end{cases} \tag{1}$$

Among them, $a_{11} \ldots a_{mn}, b_1 \ldots b_m$ are constants, $x_1 \ldots x_n$ are unknown numbers that need to be solved, and this kind of equation is called a system of linear equations.

Another representation can be expressed in the form of linear algebra:

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

where $\mathbf{A}$ is $m \times n$ matrix:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$$

$a_{11} \ldots a_{mn}$ are constants. $\vec{b}$ and $\vec{x}$ are column vector with m and n elements, respectively. The aim is to find the $\vec{x}$ that adapts to these conditions, which satisfies $\mathbf{A}\vec{x} = \vec{b}$.

The above mentioned is a system of linear equations in the classical algorithm, which is expressed in quantum computation as follows:

$$\mathbf{A}|x\rangle = |b\rangle \tag{3}$$

where $|x\rangle$ and $|b\rangle$ are quantum states. For example, when $(0\ 0\ 1\ 0)^T$ the quantum state is $|x\rangle = |10\rangle$.

The HHL algorithm is used to solve quantum equations. However, when using HHL algorithm to solve the problem, some specific conditions still need to be met. Matrix $\mathbf{A}$ must be an $n \times n$ square matrix, and $\mathbf{A}$ must be a Hermitian operator. If not, $\mathbf{A}$ needs to be transformed into a Hermitian matrix in some way. The core idea of HHL algorithm includes phase estimation in [9], controlled rotation and inverse phase estimation.

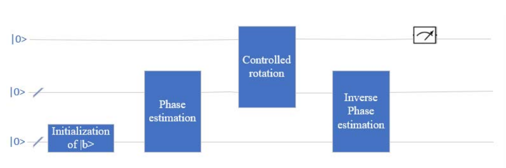

Fig. 1. Schematic diagram of HHL algorithm

The general steps of the HHL algorithm are:

1) First, initialize the quantum state $|b\rangle$,
2) The quantum phase estimation method is used to decompose $|b\rangle$ into the superposition of the linear combination of eigenvectors of $\mathbf{A}$,
3) After inverting the matrix $\mathbf{A}$, we get the quantum state $\mathbf{A^{-1}}|b\rangle = |x\rangle$.
4) Canceling the eigenvalues stored in the register by using the inverse quantum phase estimation method.

III. Qiskit Implement

After a certain understanding of the HHL algorithm, the algorithm can be achieved on the qiskit of IBM Company. Here, the qiskit software will be used to calculate the solution of the system of fourth-order linear equations, and different fourth-order matrices are calculated to analyze the accuracy and practicability of the algorithm.

First of all, before the operation of the fourth-order matrix, let's verify the effect that the operation of the second-order matrix can achieve. Use an example to verify this:

Input matrix

$$\mathbf{A} = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}, b = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$$

The quantum circuit diagram is shown in figure 2 and the result of the operation of the quantum circuit can be seen in figure 3.

We can see from figure 3 that the result is $\begin{bmatrix} \sqrt{0.159\%} \\ \sqrt{0.098\%} \end{bmatrix}$, normalize $\begin{bmatrix} \sqrt{0.159\%} \\ \sqrt{0.098\%} \end{bmatrix}$ to get $x = \begin{bmatrix} 0.7866 \\ 0.6175 \end{bmatrix}$. The result's fidelity is 1.0000. When analyzing the experimental results, quantum fidelity is one of the important indicators, which is defined as:

$$f = |\langle x(t)|x_e(t)\rangle|^2 \tag{4}$$

In which the initial state $x(0)\rangle$ gets the state $x(t)\rangle$ after the evolution of time t, and $|x_e(t)\rangle$ is the perturbed state corresponding to state $|x(0)\rangle$ [10].

In the experiment in this paper, fidelity can be understood as the inner product of two vectors:

$$f = (x_{theory}\ ,\ x_{exp}) \tag{5}$$

where x_{theory} and x_{exp} are normalized values. And $f \in (0, 1)$. It can be seen that the operation of HHL algorithm on qiskit can achieve very good results, and the results obtained have high fidelity for second-order matrices.

Next, we will use a few examples to verify what effect can be achieved in the fourth-order matrix. The main program of HHL algorithm of fourth-order matrix on qiskit platform is:

```
from qiskit.aqua import run_algorithm
from qiskit.aqua.input import
LinearSystemInput
from qiskit.quantum_info import
state_fidelity
from qiskit.aqua.algorithms.classical import
ExactLSsolver
import numpy as np

params = {
        'problem': {
```

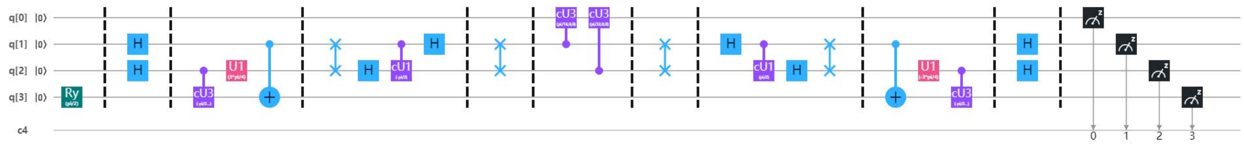

Fig. 2. Quantum circuit diagram of HHL algorithm for second order linear equations

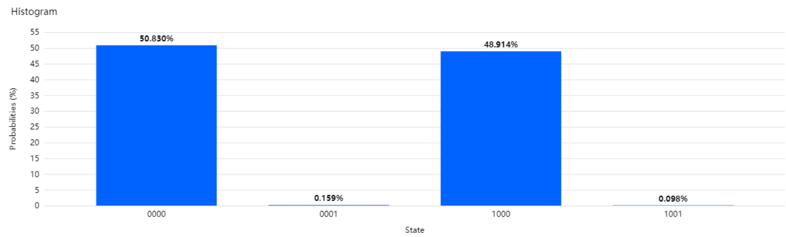

Fig. 3. The result of HHL algorithm for two-solution system of linear equations

```
                'name': 'linear_system'
        },
        'algorithm': {
                'name': 'HHL'
        },
        'eigs': {
                'expansion_mode':
                'suzuki',
                'expansion_order': 2,
                'name': 'EigsQPE',
                'num_ancillae': 3,
                'num_time_slices': 50
        },
        'reciprocal': {
                'name': 'Lookup'
        },
        'backend': {
                'provider':
                'qiskit.BasicAer',
                'name':
                'statevector_simulator'
        }
}

def fidelity(hhl, ref):
solution_hhl_normed =
hhl / np.linalg.norm(hhl)
solution_ref_normed =
ref / np.linalg.norm(ref)
fidelity = state_fidelity
(solution_hhl_normed,
solution_ref_normed)
print("fidelity %f" % fidelity)
```

1) When the input is a diagonal matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

After performing this operation, the result we get that the quantum solution is

$$\begin{bmatrix} 0.4986 + 0.j \\ 1.0417 + 0.j \\ 0.32078 + 0.j \\ 0.4986 + 0.j \end{bmatrix}$$

and the classical solution is

$$\begin{bmatrix} 0.5 \\ 1 \\ 0.33333 \\ 0.5 \end{bmatrix}.$$

The result's fidelity is 0.999353. We can see that the correct rate of the result at this time is very high.

2) When the input is a non-diagonal matrix, but the sparsity is very strong

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

At this point, the result is that the quantum solution is

$$\begin{bmatrix} 0.34947 + 0.j \\ -0.08757 + 0.j \\ 0 + 0.j \\ 0.30551 + 0.j \end{bmatrix}$$

and the classical solution is

$$\begin{bmatrix} 0.33333 \\ -0.33333 \\ 0 \\ 0.33333 \end{bmatrix}.$$

The result's fidelity is 0.823963. The result shows that the fidelity of the result is still very high.

3) Next, enter a real symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 2 & 6 & 4 & 3 \\ 1 & 4 & 2 & 5 \\ 0 & 3 & 5 & 4 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The result is that the quantum solution is

$$\begin{bmatrix} 0.2943 - 0.j \\ -0.0756 + 0.j \\ 0.0361 - 0.j \\ 0.1445 - 0.j \end{bmatrix}$$

and the classical solution is

$$\begin{bmatrix} 0.54818 \\ -0.51852 \\ 0.39259 \\ 0.14815 \end{bmatrix}.$$

The result's fidelity is 0.653278. Looking at the current output, compared with the previous calculation, the result this time does not seem to be very satisfactory.

IV. Conclusion

This paper mainly introduces the basic situation of HHL algorithm, and carries on the related verification on qiskit, mainly in solving the problem of four-dimensional linear equations is discussed in detail. Specific analysis of the HHL algorithm from different types of fourth-order matrices, the experimental results show that the fidelity of the algorithm in solving the system of linear equations composed of high sparse matrix is very good, but when the sparsity of the matrix is very poor, the result of the algorithm is not very ideal. The problems studied in this paper are mainly suitable for solving some simple linear equations, and the problems solved by HHL algorithm for complex linear equations need to be further studied.

Acknowledgment

This work is supported by Henan Key Laboratory of Network Cryptography Technology (No LNCT2019-S06).

References

[1] Harrow Aram W, Hassidim Avinatan, Lloyd Seth. *Quantum algorithm for linear systems of equations*. Phys. Rev. Lett. 15, 103 (2009).

[2] J. Pan, Y. Cao, X. Yao, et al. *Experimental realization of quantum algorithm for solving linear systems of equations*. Phys. Rev. A 89, 022313 (2014).

[3] X. -D. Cai, C. Weedbrook, Z. -E. Su, et al. *Experimental quantum computing to solve systems of linear equations*. Phys. Rev. Lett. 110, 230501 (2013).

[4] Y. Cao, A. Daskin, S. Frankel, et al. *Quantum circuit design for solving linear systems of equations*. Journal of Molecular Physics, 1675-1680, (2012).

[5] Stephan Eidenbenz, Scott Pakin, Gopinath Chennupati, et al. *Quantum Algorithm Implementations for Beginners*. arXiv:1804.03719v1 (2018).

[6] Stefanie Barz, Ivan Kassal, Martin Ringbauer,et al. *Solving systems of linear equations on a quantum computer*. arXiv:1302.1210v1 (2013).

[7] Juan Miguel Arrazola, Timjan Kalajdzievski, Christian Weedbrook, et al. *Quantum algorithm for non-homogeneous linear partial differential equations*. arXiv:1809.02622v1 (2018).

[8] Y. Zheng, C. Song, M. -C. Chen, et al. *Solving Systems of Linear Equations with a Superconducting Quantum Processor*. Phys. Rev. Lett. 118, 210504 (2017).

[9] U. Dorner, R. Demkowicz-Dobrzanski, B. J. Smith, et al. *Optimal Quantum Phase Estimation*. Phys. Rev. Lett. 102, 040403 (2009).

[10] T. Gorin, T. Prosen, T. H. Seligman, et al. *Dynamics of Loschmidt echoes and fidelity decay*. Physics Reports, 435, 33-156 (2006).