Quantum Computing

武亦文

2022 年 5 月 19 日

Chapter 1

Introduction

图 1.1: IBM Quantum System One (20qbits)

1.1 What's quantum computing?

1.2 Why quantum computing?

Chapter 2

Quantum Circuit Model

2.1 Quantum bit

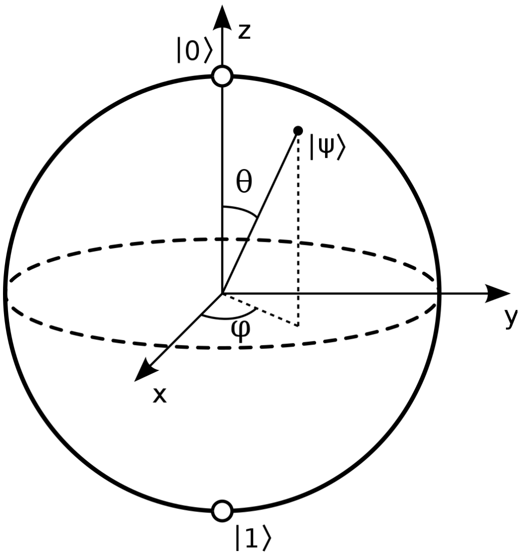

图 2.1: Bloch Sphere

2.2 Quantum gate

2.2.1 Single qubit gate

2.2.2 Multiple qubit gate

2.3 Quantum circuit

2.3.1 Example: Qubit copying circuit?

2.3.2 Example: Bell state (EPR pairs)

2.3.3 Example: Quantum teleportation

2.4 Quantum algorithm

Broadly speaking, there are three classes of quantum algorithms which provide an advantage over known classical algorithms. First, there is **the class of algorithms based upon quantum versions of the Fourier transform**, a tool which is also widely used in classical algorithms. The Deutsch–Jozsa algorithm is an example of this type of algorithm, as are Shor' s algorithms for factoring and discrete logarithm. The second class of algorithms is **quantum search algorithms**. The third class of algorithms is **quantum simulation**, whereby a quantum computer is used to simulate a quantum system.

2.4.1 Quantum Fourier transform

The discrete Fourier transform is defined by

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} x_j \tag{2.1}$$

What is important is that the Hadamard transform used in the Deutsch–Jozsa algorithm is an example of this generalized class of Fourier transforms. Moreover, many of the other important quantum algorithms also involve some type of Fourier transform.

The most important quantum algorithms known, Shor' s fast algorithms for factoring and discrete logarithm, are two examples of algorithms based upon the Fourier transform defined in Equation (2.1).

Classically, the fast Fourier transform takes roughly $Nlog(N) = n2^n$ steps to Fourier transform $N = 2n$ numbers. On a quantum computer, the Fourier transform can be accomplished using about $\log^2(N) = n^2$ steps, **an exponential saving**!

However,the results of such a calculation are not available to us if we go about it in a straightforward manner. More cleverness is required in order to harness the power of quantum computation.

2.4.2 Quantum search algorithm

A completely different class of algorithms is represented by the quantum search algorithm, whose basic principles were discovered by Grover. The quantum search algorithm solves the following problem: Given a search space of size N, and no prior knowledge about the structure of the information in it, we want to find an element of that search space satisfying a known property. How long does it take to find an element satisfying that property? Classically, this problem requires approximately N operations, but the quantum search algorithm allows it to be solved using approximately $\sqrt{N}$ operations.

The quantum search algorithm offers only **a quadratic speedup**, as opposed to the more impressive exponential speedup offered by algorithms based on the quantum Fourier transform. However, the quantum search algorithm is still of great interest, since searching heuristics have a wider range of application than the problems solved using the quantum Fourier transform, and adaptations of the quantum search algorithm may have utility for a very wide range of problems.

2.4.3 Quantum simulation

Simulating naturally occurring quantum mechanical systems is an obvious candidate for a task at which quantum computers may excel, yet which is believed to be difficult on a classical computer.

In general, storing the quantum state of a system with n distinct components takes something like c^n bits of memory on a classical computer, where c is a constant which depends upon details of the system being simulated, and the desired accuracy of the simulation.

By contrast, a quantum computer can perform the simulation using kn qubits, where k is again a constant which depends upon the details of the system being simulated. This allows quantum computers to efficiently perform simulations of quantum mechanical systems that are believed not to be efficiently simulatable on a classical computer.

However, a significant caveat is that, when measured, a kn qubit simulation will collapse into a definite state, giving only kn bits of information; the c^n bits of ‘hidden information’ in the wave function is not entirely accessible.

Thus, a crucial step in making quantum simulations useful is development of systematic means by which desired answers can be efficiently extracted; how to do this is only partially understood.

Chapter 3

Applications in Nuclear Physics

Nuclear physics is the study of systems of interacting protons and neutrons, both of which are spin half fermions. The joint experimental and theoretical understanding of these properties is the field of *nuclear structure* (as opposed to nuclear reactions), and is what concerns us here.The starting point of theoretical studies of nuclear structure is *the many-body Schrodinger equation*, using the nuclear interaction as the potential term.

While nuclear physics problems are relatively unexplored on quantum computers, a few investigations have been made. Among these is the determination of **the ground state energy of the deuteron** using a one-body Hamiltonian derived from quantum field theory. This calculation was made using the **variational quantum eigensolver (VQE)** on IBM cloud quantum computers.

3.1 Method of solving the np bound state

Hamiltonian and model space.——Introduce and tailor a deuteron **Hamiltonian** from pionless effective field theory (EFT).

Mapping the deuteron onto qubits.——Quantum computers manipulate qubits by operations based on Pauli matrices. Write out the Hamiltonian explicitly in matrix form in the oscillator basis and decomposing directly into tensor products of **Pauli spin matrices**.

Preparing the variational wave function.——In quantum computing, a popular approach to determine the ground-state energy of a Hamiltonian is to use UCC ansatz in tandem with the VQE algorithm.

Reducing the cost.——Reduce the number of qubits and circuit depth as much as possible. Because the implementation of quantum many-body simulation algorithms on these devices faces *gate and measurement errors*, and *qubit decoherence.*

Quantum computation.——Use **the VQE quantum-classical hybrid algorithm** to minimize the Hamitonian expectation value or the variance for our wave function ansatz, to obtain the energy of ground state and excited state.

3.2 Pauli spin matrix decomposition of real symmetric matrices

There are many ways to encode a Hamiltonian in terms of Pauli matrices, and here we present a method to represent an $N \times N$ real symmetric matrix as the combination of Pauli matrices that composes the given matrix through *Kronecker products*.

As a sample application, we apply the method to a Hamiltonian describing the deuteron using effective field theory as previously implemented on a quantum computer under the *Jordan-Wigner mapping.*

First, we started by 2×2 matrix. Including the Identity I, the Pauli matrices can be written as

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

An arbitrary linear combination of the I, X and Z with real coefficients, is

$$a_0 I + a_1 + a_3 Z = \begin{pmatrix} a_0 + a_3 & a_1 \\ a_1 & a_0 - a_3 \end{pmatrix}$$

One can see that any real 2×2 symmetric matrix can be represented in this form.

Then we consider general case, $N \times N$ matrix. As an example, we work through a case of a 3×3 matrix. We take the following matrix, which represents the energy of the deuteron in an oscillator basis restricted to three oscillator states.

$$a_0 I + a_1 + a_3 Z = \begin{pmatrix} -0.43658111 & -4.28660705 & 0 \\ -4.28660705 & 12.15 & -7.82623792 \\ 0 & -7.82623792 & 19.25 \end{pmatrix}$$

a) *If H is not of a power-of-two order, increase the size of H and pad the new elements with zero.*

This is extended to the next power of two order, 4×4 by padding with zeros:

$$a_0 I + a_1 + a_3 Z = \begin{pmatrix} -0.43658111 & -4.28660705 & 0 & 0 \\ -4.28660705 & 12.15 & -7.82623792 & 0 \\ 0 & -7.82623792 & 19.25 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

b) *From the set I, X, Y, Z, Generate the set of Kronecker products of the permutations, and store in a dictionary.*

Since the order $4 = 2^2$ we make a list of all pairs of Pauli matrices which do not have an odd numbers of Y matrices. This list is

$$II, IX, IZ, XI, XX, XZ, YY, ZI, ZX, ZZ.$$

Each of the elements in the list has a 4×4 matrix representation made, which are:

$$II = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad IX = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad IZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \ldots$$

c) *Loop through the elements of the Hamiltonian H.*

Encode the equation as one row of M in the matrix-vector equation $Ma = h$ where a is a vector of all the unknowns and h is the Hamiltonian reshaped as a column vector.

The full equation for our example is

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{II} \\ a_{IX} \\ a_{IZ} \\ a_{XI} \\ a_{XX} \\ a_{XZ} \\ a_{YY} \\ a_{ZI} \\ a_{ZX} \\ a_{ZZ} \end{pmatrix} = \begin{pmatrix} -0.43658111 \\ -4.28660705 \\ 0 \\ 0 \\ 12.15 \\ -7.82623792 \\ 0 \\ 19.25 \\ 0 \\ 0 \end{pmatrix}$$

d) *Solve set of equations for unknowns a_n.*

$$H = 7.766II - 2.143IX + 1.641IZ - 3.913XX - 3.913YY - 1.859ZI - 2.143ZX - 7.984ZZ$$

3.3 Variational quantum eigensolver (VQE)

VQE is an approach to **finding Hamiltonian eigenvalues**, in which a quantum processor is used as part of a hybrid quantum-classical algorithm.

4 steps to formulate a physical problem:

a) Establish a **correspondence** between the *physical states* and the *multi-qubit states* of a quantum computer,

b) Prepare a **parameterized ansatz state** $|\psi(\vec{\theta})\rangle = U(\vec{\theta})\,|\psi_0\rangle$ on quantum computer,

c) Evaluate **Hamiltonian expectation value** $E(\vec{\theta}) = \langle\psi(\vec{\theta})|\,\hat{H}\,|\psi(\vec{\theta})\rangle$ by sampling on the *quantum computer*,

d) Send the estimated value $E(\vec{\theta})$ to the *classical optimizer* to **determine the set of parameters** for the next iteration of the algorithm.