Essay submitted for the course

Introduction to Quantum Computing

Lecturer: Dr. Petros Wallden
University of Edinburgh, Semester 1, 2015/16

Quantum Simulation

Michael Krause

s1572730@sms.ed.ac.uk

October 29, 2015

Contents

1 Introduction

This essay deals with a technique called *Quantum Simulation*, which means simulating one quantum system through another. It begins by motivating the benefits one might obtain from simulating quantum systems and by introducing the main challenges one faces in the process. In section 2, an algorithm that employs quantum simulation to solve linear equations is reviewed and possible applications are discussed.

1.1 Why Quantum Simulation?

Since the middle of the 20th century, the development of ever more powerful computing devices has introduced a new method to the sciences: computer simulation. With the help of computers, scientists can model natural phenomena utilizing thousands of parameters, thereby gaining a better understanding of the laws that give rise to these phenomena. Today, simulations play an integral part in many disciplines of science and engineering and to some extents - for example through weather forecasts - even in our daily lives (see also [7]).

Quantum mechanics with its substantial mathematical foundations and counter-intuitive predictions seems to be a natural domain for computer simulation. Experimental setups to analyse quantum effects may be complex and error prone and researchers would benefit greatly from the ability to simulate these effects on a more easily controllable system.

In a paper from 1982, Richard Feynman first discussed the challenges of simulating quantum systems on classical computers (see [3] and [2]). In order to simulate the behaviour of a system over time, its initial state must first be represented in some way by the device that is to carry out the simulation. In the quantum mechanical case, the state of a system is a unit vector in complex Hilbert space whose dimension N is the number of particles in the system. Thus, the system state may be described by a vector $|\phi\rangle = \Sigma_i \alpha_i |i\rangle$, where the $|i\rangle$ form a basis of the system space and the α_i are complex numbers representing the corresponding probability amplitudes. Therefore, storing this state involves storing $i = 2^N$ complex numbers, a number which grows exponentially with the number of particles in the system.

The evolution of a quantum system over time is described by the Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} |\phi\rangle = H |\phi\rangle$$

where H is the system's Hamiltonian. This poses another problem for simulation: If H stays constant over time, the solution of this equation at time t for an initial state $|\phi(0)\rangle$ is

$$|\phi(t)\rangle = e^{-i\hbar H t} |\phi(0)\rangle$$

Approximations for exponentiating H are possible in some cases, but this operation is usually intractable on classical computers (see [5] and [3]).

In his paper, Feynman suggested that quantum computers might be able to simulate quantum systems efficiently as they act as quantum systems themselves ([3]). The state of a quantum computer is itself a superposition of its qubits, so a quantum computer might store the state of an N particle system using only an equivalent amount of qubits, thereby avoiding the exponential growth in memory requirement that a classical computer suffers from. Additionally, an important result by Lloyd from 1996 showed that the circuits of a quantum computer could indeed be used to simulate any Hamiltonian acting on a state (and that this is efficiently possible in some important special cases). Subsection 1.2 sketches how these properties might be used for quantum simulation.

Quantum computing has applications that go beyond the simulation of other quantum system (for instance Shor's algorithm for factoring prime numbers, as well as other algorithms, see [5]). However, since quantum computers with a sufficient amount of qubits for practical use are not available yet, researchers have also developed alternative approaches for specific domains, where the simulating and the simulated system are very similar (see the sections on analogue quantum simulation in [3]). The rest of this essay will discuss quantum simulation in the context of digital quantum simulation, i.e: using quantum circuits.

1.2 How to Simulate a Quantum System

Generally, simulating a quantum system involves three steps (as seen in [3]). For a simulation to be deemed efficient, these steps must be performed in time polynomial in the size of the input problem.

1. *Preparing the initial state:* The simulation begins by establishing some mapping between the state of the simulated system and the state of the simulator. This is difficult, because it means gathering some data from the simulated system and engineering circuits that transform the simulating system into a similar state. For some simulation problems, efficient circuits for this task can be found (see [3]).

2. *Simulating the evolution of the system:* Once simulated and simulating system are in a similar state, it must be assured that they evolve in parallel. As established by Lloyd, quantum circuits can emulate the actions of any Hamiltonian on a system and can do so efficiently if H can be written as a sum of local Hamiltonians (operators that only apply to some of the qubits of the system, see [5] and [3]). However, while simulation is efficiently possible in many cases in theory, finding a combination of quantum gates that simulates a Hamiltonian in practice is very hard and generally yields only approximations of the 'real' evolution.

3. *Measuring the final state:* If the initial states as well as the evolutions of the simulated and the simulating system were sufficiently alike, so are their final states. Thus, by measuring some property of the simulator, insight into the simulated system can be gained.

As quantum computers become more powerful, quantum simulation may be applied to solve real-world problems. Many problems from various disciplines of physics have been proven to be efficiently solvable through a quantum simulator. The main source of this essay ([3]) provides a general overview of these applications. In contrast, the following section presents an algorithm that employs quantum simulation as a subroutine to solve linear equations.

2 Quantum simulation for linear algebra

One of the most basic tasks in linear algebra is finding the solution to a system of linear equations. That is, for a given $n \times n$ matrix A and an n vector b, find x with $Ax = b$. If the inverse of A is known, this reduces to a matrix-multiplication $x = A^{-1}b$. This problem must often be solved as part of a larger computation.

In a paper from 2009, Harrow, Hassidim and Lloyd ([4]) introduced a quantum algorithm that solves some systems of linear equations with exponential speedup compared to the best known classical algorithm. They note that any algorithm computing an appropriate x would have to run in at least $n^2 + n$ steps to read every entry of A and b and take additional n steps to write out x. In contrast, their algorithm (from here on referred to as HHL, as in [1]) solves the problem under some conditions in time logarithmic in n. In the following, these conditions will be clarified and the algorithm and its applications will be discussed.

2.1 Solving linear equations

The HHL algorithm takes as input a vector b, encoded as the amplitudes of a quantum state

$$|b\rangle = \Sigma_i b_i |i\rangle$$

and a sparse $n \times n$ Hermitian matrix A. Using Hamiltonian simulation of A and phase estimation, b is rewritten with regard to an eigenbasis $|u_i\rangle$ of A. Another mapping is applied to multiply with the inverse eigenvalues corresponding to these $|u_i\rangle$. As this mapping might fail, it is repeatedly applied to obtain a state $|x\rangle = \Sigma_i x_i |u_i\rangle$, where the x_i form a vector proportional to $A^{-1}b$. Finally, a measurement may be applied to readout some property of the solution x.

That A is sparse means that it contains at most s non-zero entries per row for a constant $s \ll n$. Furthermore, A is assumed to be well-conditioned, so its condition number $\kappa_A = \frac{\lambda_{max_A}}{\lambda_{min_A}}$ is not much greater than 1, where λ_{max_A} and λ_{min_A} denote the biggest and smallest eigenvalue of A respectively. Greater κ mean that a matrix is closer to being non-invertible. The whole procedure runs in time polynomial in $log(n) \cdot \kappa_A^2 \cdot \epsilon^{-1}$, where ϵ is the desired error in $|x\rangle$.

The HHL algorithm takes advantage of the fact that for sparse, well-conditioned A and times t there are procedures to apply e^{-iAt} to $|b\rangle$ in time logarithmic in n. It also works for any non-sparse A that can be simulated efficiently.

Additionally, the authors suggest that for ill-conditioned matrices A the algorithm may be modified to invert only parts of A. Alternatively, a common

technique from numerical calculus may be used, where the condition number of the problem is improved by solving $AB^{-1}x' = b$, with $x' = Bx$ for a matrix B so that $\kappa_{AB^{-1}} < \kappa_A$.

A central result from the HHL paper states that any quantum circuit can be simulated by solving a system of linear equations. Therefore, if there was a classical algorithm capable of achieving similar performance in solving linear equations, this would imply $\mathsf{BPP} = \mathsf{BQP}$: quantum computers would be no more powerful than probabilistic polynomial time Turing machines.

2.2 Applications

The HHL algorithm encodes its solution as the amplitudes of a quantum state, which are not easily accessible. Instead, only some properties of the solution can be revealed through measurements. Despite the algorithms limited scope, researchers have found a number of interesting applications for it, often addressing machine learning problems such as clustering data sets. An overview of these is given by Aaronson in [1]. For example, [6] gives an algorithm that approximates the parameters of a least-squares fit of a set of data points using an optimized version of the HHL algorithm.

As Aaronson notes, however, most of these applications might not yield the expected performance benefits unless efficient procedures are found by which the input data (A and b) are calculated or loaded into the quantum computers memory. Furthermore, efficient quantum algorithms do not by themselves rule out the possibility that equally efficient classical algorithms might exist.

Thus, while the algorithm by Harrow et al. does demonstrate exponential speedup in some cases, it has its limitations. Still, researchers find exciting uses for it and since it was introduced fairly recently, one can expect further results in the future.

References

[1] S. Aaronson. Quantum Machine Learning Algorithms: Read the Fine Print. *Nature Physics*, 11:291–293, 2015.

[2] G. Benenti and G. Strini. Quantum simulation of the single-particle Schrödinger equation. *American Journal of Physics*, 76:657–662, July 2008.

[3] I. M. Georgescu, S. Ashhab, and F. Nori. Quantum simulation. *Reviews of Modern Physics*, 86:153–185, January 2014.

[4] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15):150502, October 2009.

[5] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[6] N. Wiebe, D. Braun, and S. Lloyd. Quantum Algorithm for Data Fitting. *Physical Review Letters*, 109(5):050505, August 2012.

[7] Eric Winsberg. Computer simulations in science. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2015 edition, 2015.