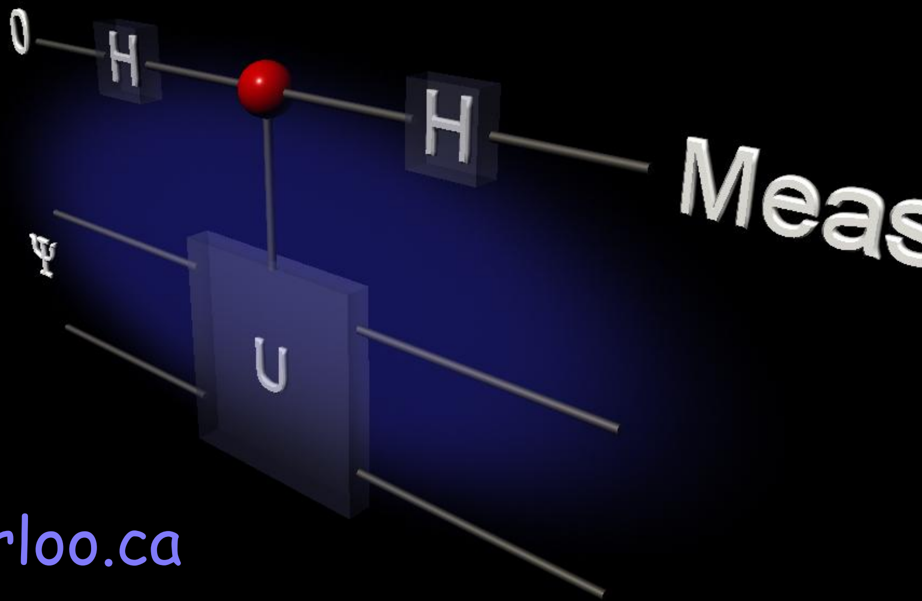
Introduction to Quantum Information Processing



CO481 CS467 PHYS467

Michele Mosca mmosca@iqc.uwaterloo.ca

Tuesdays and Thursdays 10am-11:15am

IQC Institute for Quantum Computing

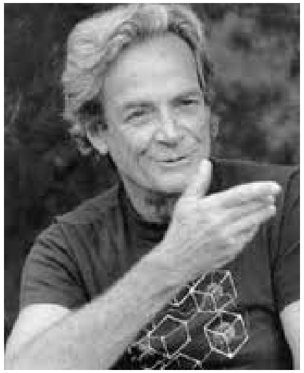
UNIVERSITY OF WATERLOO

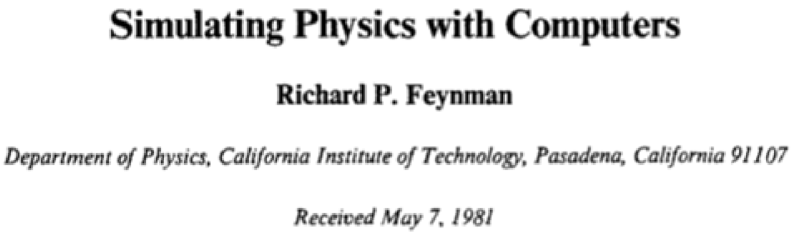
Simulating Quantum Systems

- Motivation
- Hamiltonians
- The Hamiltonian simulation problem
- Simulation for a simple Hamiltonian
- Simulation for local Hamiltonians

Motivation

- Nature is supposed to be quantum, it just looks classical sometimes
- It is also believed that classical systems cannot simulate quantum mechanics efficiently (the Hilbert space is too large!)
- Therefore, quantum computers seem a good candidate for simulating natural systems





Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Applications

- If we can simulate the dynamics of general quantum systems, we can simulate:
  - General quantum mechanical theories (e.g. QFT, QCD)
  - Quantum chemical dynamics (electron structure of large molecules)

- This is of interest to
  - Theoretical physicists
  - Chemical industry
  - Pharmaceutical industry

Applications

- Many supercomputing CPU cycles are spent right now on simulation of these and others quantum systems
- Quantum computing presumably offers a exponential speedup
- For many, this is supposed to be the “killer app” of quantum computing

Hamiltonians

- The physical systems people want to simulate have been characterized in terms of their Hamiltonian
- This is a Hermitian matrix that characterizes the evolution of the system. It might change over time
- The way in which the Hamiltonian characterizes the evolution of the system is given by the Schrödinger equation

Schrödinger Equation

The continuous time-evolution of a closed quantum system follows the Schrödinger equation:

$$i\hbar \frac{d\left|\psi(t)\right\rangle}{dt} = H(t)\left|\psi(t)\right\rangle$$

where $\left|\psi(t)\right\rangle$ is the state of the system at time t

H(t) is the Hamiltonian of the system at time t

$\hbar$ is Planck’s constant

Schrödinger Equation

If the Hamiltonian does not depend on time ("time-independent"), then the solution is

$$\left|\psi(t_2)\right\rangle = e^{-\frac{i}{\hbar}H(t_2-t_1)}\left|\psi(t_1)\right\rangle$$

- We normalize to get rid of the ħ and let $t_1 = 0$. We obtain

$$\left|\psi(t)\right\rangle = e^{-iHt}\left|\psi(0)\right\rangle$$

- We will look at simulation of systems with time independent Hamiltonians
- Time dependent case can be approximated by sufficiently small time-independent intervals. Must be done carefully.

Hamiltonian Simulation Problem

- We want to implement a Hamiltonian in a programmable quantum system

- We have a quantum state, and a Hamiltonian. We want to obtain the state after it evolves according to the Hamiltonian

- We want to achieve this using unitary gates taken from a finite set

Simple time independent Hamiltonian

A spin-$\frac{1}{2}$ particle in a magnetic field oriented along the z-axis has Hamiltonian

$$cZ = \begin{bmatrix} c & 0 \\ 0 & -c \end{bmatrix}$$

So after time t we have

$$|0\rangle \rightarrow e^{-ict}|0\rangle$$

$$|1\rangle \rightarrow e^{ict}|1\rangle$$

Simple time independent Hamiltonian

If we have a system of two non-interacting spin-$\frac{1}{2}$ particles, the Hamiltonian of the 2-qubit system is

$$c_1 Z \otimes I + c_2 I \otimes Z$$

$$= \begin{bmatrix} c_1 + c_2 & 0 & 0 & 0 \\ 0 & c_1 - c_2 & 0 & 0 \\ 0 & 0 & -c_1 + c_2 & 0 \\ 0 & 0 & 0 & -c_1 - c_2 \end{bmatrix}$$

Simple time independent Hamiltonian

If the particles are close enough they interact non-trivially adding the following term to the Hamiltonian

$$j_{12} Z \otimes Z$$

$$= \begin{bmatrix} j_{12} & 0 & 0 & 0 \\ 0 & -j_{12} & 0 & 0 \\ 0 & 0 & -j_{12} & 0 \\ 0 & 0 & 0 & j_{12} \end{bmatrix}$$

Simple time independent Hamiltonian

The Hamiltonian of the whole system is thus the sum of the 3 Hamiltonians:

$$c_1 Z \otimes I + c_2 I \otimes Z + j_{12} Z \otimes Z$$

$$= \begin{bmatrix} c_1 + c_2 + j_{12} & 0 & 0 & 0 \\ 0 & c_1 - c_2 - j_{12} & 0 & 0 \\ 0 & 0 & -c_1 + c_2 - j_{12} & 0 \\ 0 & 0 & 0 & -c_1 - c_2 + j_{12} \end{bmatrix}$$

Easy Hamiltonians

It is easy to simulate Hamiltonians of the following form

$$-H_k = U \Lambda U^{\dagger}$$

where

- U is an easy-to-implement unitary operation
- The diagonal elements of Λ are easy to approximate

Why?

- Note that (from spectral decomposition)

$$e^{-iH_k t} = e^{iU\Lambda U^\dagger t} = Ue^{i\Lambda t}U^\dagger$$

- So if U is an easy-to-implement unitary operation, we only have to worry about simulating $e^{i\Lambda t}$

Why?

If

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_N \end{bmatrix}$$

We want to map

$$| j \rangle \mapsto e^{i\lambda_j t} | j \rangle$$

Approximating this

- We want to map

$$\left| j \right\rangle \mapsto e^{i\lambda_j t} \left| j \right\rangle$$

- Suppose we want precision roughly $\frac{1}{2^n}$

- Suppose we can efficiently compute

$$f(j) \in \{ 0,1,2,\ldots,2^n - 1 \}$$

satisfying

$$\left| \frac{f(j)}{2^n} - \frac{\lambda_j t}{2\pi} \right| < \frac{1}{2^n}$$

Approximating $e^{i\Lambda t}$

- An exercise in reversible computing shows us how to implement U_f

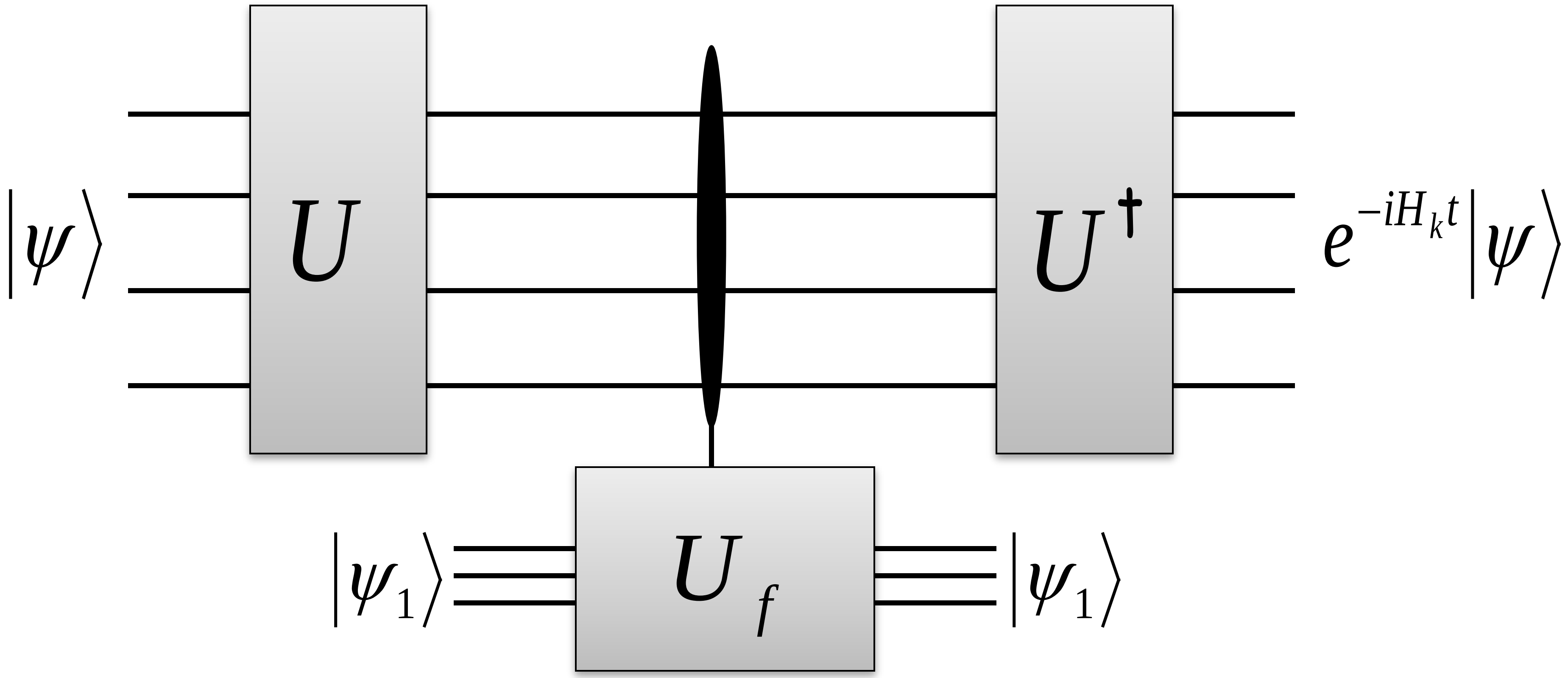
$$|j\rangle|b\rangle \mapsto |j\rangle|b + f(j) \bmod 2^n\rangle$$

- Consider

$$|\psi_1\rangle = QFT_{2^n}^{-1}|1\rangle = \sum_x e^{-2\pi i\frac{x}{2^n}}|x\rangle$$

- Then

$$U_f|j\rangle|\psi_1\rangle = e^{2\pi i\frac{f(j)}{2^n}}|j\rangle|\psi_1\rangle \approx e^{i\lambda_j t}|j\rangle|\psi_1\rangle$$

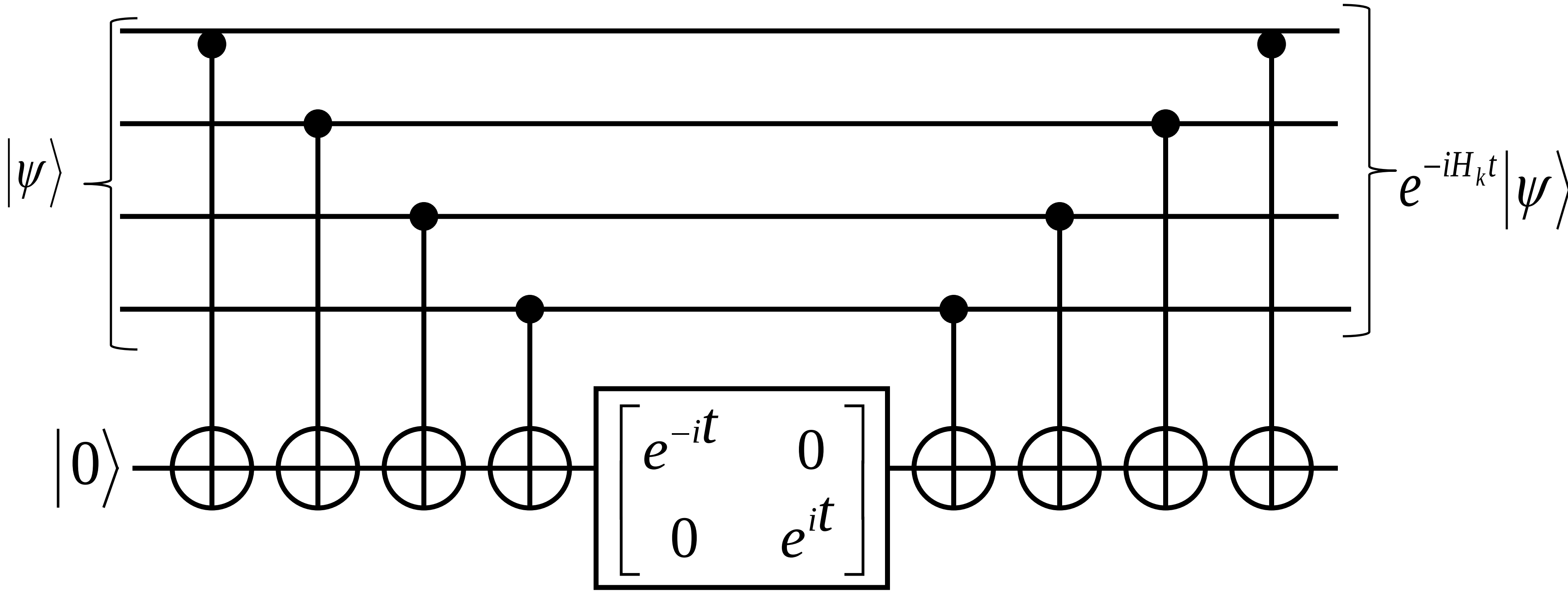

$|\psi\rangle$
U
$U^\dagger$
$e^{-iH_k t}|\psi\rangle$
$|\psi_1\rangle$
U_f
$|\psi_1\rangle$

Another way to do it
(an example)

We do not necessarily have to implement the QFT

e.g.

$$H_k = Z \otimes Z \otimes \cdots \otimes Z$$


$|\psi\rangle$
$|0\rangle$
$\begin{bmatrix} e^{-it} & 0 \\ 0 & e^{it} \end{bmatrix}$
$e^{-iH_k t}|\psi\rangle$

More general kind of Hamiltonian

In general, a collection of physical subsystems forming a larger system usually have Hamiltonians of the form

$$H = \sum_{k=1}^{L} H_k$$

where H_k is a Hamiltonian acting on a small number of nearby subsystems and L is polynomial in the number of subsystems (i.e. logarithmic in the number of possible states)

This kind of Hamiltonian is called a **local Hamiltonian**. Fundamental interactions in nature have this form. Effective Hamiltonians occuring in nature with longer-range interactions can be constructed from Hamiltonians of this form.

More general kind of Hamiltonian

Each $e^{-iH_j t}$ would be easy to simulate. As the dimension of each H_j is constant, we can diagonalize them. Then we can compute the unitary corresponding to the evolution, and implement it using the results about universal sets of gates

Unfortunately, the H_k do not all necessarily commute which means that

$$e^{-iHt} \neq e^{-iH_1 t} \cdot e^{-iH_2 t} \cdots e^{-iH_L t}$$

Trotter Formula

For Hermitian A and B and any real *t*,

$$\lim_{n\to\infty}(e^{iAt/n} \cdot e^{iBt/n})^n = e^{i(A+B)t}$$

More quantitatively

Exercise 4.50 (N&C):

If we let

$$\tilde{U}_t = e^{-iH_1 t} \cdot e^{-iH_2 t} \cdots e^{-iH_L t} e^{-iH_L t} e^{-iH_{L-1} t} \cdot e^{-iH_{L-2} t} \cdots e^{-iH_1 t}$$

then

$$\tilde{U}_t = e^{-2iHt} + O(t^3)$$

$$Error(\tilde{U}_t^{\,m}, e^{-iH(2mt)}) \in O(mt^3)$$

Simulation Algorithm: Inputs

- A local Hamiltonian $H = \sum_{k=1}^{L} H_k$

  where each H_k acts non-trivially on a subsystem of size bounded by a constant

- a description of an easy-to-prepare quantum state $|\Psi_0\rangle$

- A positive accuracy δ

- A time t_f at which the evolved state is desired

Simulation algorithm: Output and Runtime

- A state $\left|\tilde{\Psi}(t_f)\right\rangle$ such that

$$\left|\left\langle\tilde{\Psi}(t_f)\right|e^{-iHt_f}\left|\Psi_0\right\rangle\right|^2 \geq 1-\delta$$

- Runtime $O\left(poly\left(\frac{1}{\delta},L\right)\right)$ steps

Simulation Algorithm for local Hamiltonians

- We want to find a suitable approximation to e^{-iHt_f}
- e.g. Using N&C exercise 4.50, set

$$m \in O\left(\sqrt{\frac{t_f^{\,3}}{\delta}}\right) \qquad t \in O\left(\sqrt{\frac{\delta}{t_f}}\right)$$

so that $mt = t_f$ and $mt^3 \in O(\delta)$

- Approximate each U_t with accuracy in $O\left(\frac{\delta}{m}\right)$

(which can be done by approximating each $e^{-iH_j t}$ with accuracy in $O\left(\frac{\delta}{Lm}\right)$)

Simulation Algorithm

1. Prepare $\left|\tilde{\Psi}_0\right\rangle$ with accuracy in $O(\delta)$

2. Set j=1

3. Compute $\left|\tilde{\Psi}_j\right\rangle = U_t\left|\tilde{\Psi}_{j-1}\right\rangle$

4. If $j < m$, increment j and goto step 3.

   Otherwise output $\left|\tilde{\Psi}_{t_f}\right\rangle = \left|\tilde{\Psi}_m\right\rangle = U_t{}^m\left|\tilde{\Psi}_0\right\rangle$

Simulation Algorithm

- The unitaries preserve length, so the initial $O\left(\delta\right)$ error can propagate, but not get larger. Applying $U_t^{\,m}$ introduces an error of $O(mt^3) \in O(\delta)$. This gives a total $O(\delta)$ error. We can reduce this to δ by adjusting the constants in our algorithm by constant factors.

- The runtime seems not to depend badly in L or $\dfrac{1}{\delta}$. The algorithm indeed runs in time $O\left(poly\left(\dfrac{1}{\delta}, L\right)\right)$

More recent work

- Maybe the state $|\Psi_0\rangle$ whose evolution we want to simulate is not easy to prepare in general. By “simulation” people are often looking for ground-state properties of a given Hamiltonian, and often this problem is NP-hard or QMA-hard, even for local Hamiltonians.

- We can still try to adapt heuristics from classical simulation of Hamiltonians and hope we are not in a hard case

e.g.

Quantum Metropolis Sampling

K. Temme, T.J. Osborne, K. Vollbrecht, D. Poulin and F. Verstraetehttp://arxiv.org/pdf/0911.3635v2

More recent work

- We can also try to generalize results as the one in N&C exercise 4.50 to the time-dependent local Hamiltonian case

  e.g.

  N. Wiebe, D. Berry, P. Høyer and B.C. Sanders

  http://arxiv.org/pdf/0812.0562v3

  “Higher Order Decompositions of Ordered Operator Exponentials”

- We considered a simulation algorithm for a kind of simpler Hamiltonian, local Hamiltonians. There also exist simulation algorithms for other kinds of Hamiltonian

  e.g.

  A. Childs and R. Kothari

  http://arxiv.org/abs/1003.3683

  “Simulating sparse Hamiltonians with star decompositions”

OVERVIEW OF OTHER QUANTUM ALGORITHMS

Non-trivial applications of Amplitude Amplification

For example…

- Minimum/maximum finding
- Collision-finding
- String-matching
- Making quantum algorithms “exact”
- Several graph problems
- Etc. etc.

Generalization of Simon’s problem, order-finding and DLP: “Hidden subgroup problem”

- A unifying framework was developed for these problems

$$f : G \to X$$

$$f(x) = f(y) \quad \text{iff} \quad x + S = y + S$$

for some $S \leq G$

- If G is Abelian, finitely generated, and represented in a reasonable way, we can efficiently find S.

What about non-Abelian HSP

- Consider the symmetric group $G = S_n$
- S$_n$ is the set of permutations of n elements
- Let G be an n-vertex graph
- Let $X_G = \{\pi(G) \mid \pi \in S_n\}$
- Define $f_G : S_n \to X_G \qquad f_G(\pi) = \pi(G)$
- Then $f_G(\pi_1) = f_G(\pi_2) \Leftrightarrow \pi_1 S = \pi_2 S$

where $S = AUT(G) = \{\pi \mid \pi(G) = G\}$

Graph automorphism problem

- So the hidden subgroup of f_G is the automorphism group of G
- This is a difficult problem in NP that is believed not to be in BPP and yet not NP-complete.
- A solution to the graph automorphism problem gives a solution to the graph isomorphism problem.

Generalizations of Abelian HSP

- Already mentioned non-Abelian HSP; various tools include non-Abelian QFT, “pretty good” measurements, “sieving”, and non-trivial reductions to Abelian HSP in some cases.

Generalizations of Abelian HSP

- Can view HSP as a hidden sub-lattice problem for $Z \otimes Z \otimes \cdots \otimes Z = Z^n$.

  One way to generalize the problem, is to find a hidden sub-lattice of $R \otimes R \otimes \cdots \otimes R = R^n$.

  Need to define appropriate ways for specifying/approximating inputs and outputs.

  Applications include solving Pell's equation, Principal Ideal Problem, and finding the unit group of a number field.

Generalizations of Abelian HSP

- Finding Hidden Shifts and Translations
- Can generalize to finding hidden “non-linear” structures. E.g. hidden radius problem, shifted subset problem, hidden polynomial problem
- Estimating “Gauss sums”
- Etc.

“Adiabatic” Algorithms

- Clever idea based on the adiabatic theorem

  - H_0, H_1 - Hamiltonians.
  - ψ_0, ψ_1 - lowest energy states of H_0, H_1.

***Theorem*:**

If we apply a Hamiltonian $H(t)$ that “slowly” changes from H_0, H_1,

then ψ_0 is transformed to a state close to ψ_1

Using adiabatic theorem for computation (Farhi et al.)

$$\begin{tikzcd}
H_0 \arrow[r] & H_1 \\
\Psi_0 \arrow[r] & \Psi_1
\end{tikzcd}$$

- H_0 - easy to compute ground state.
- H_1 - ground state is solution to some problem.
- H_0 and H_1 can both be efficiently implemented.

Slowly change $H_0 \rightarrow H_1$

Quantum walk algorithms

- Can generalize notion of classical random walks
- Can get up to quadratic speed-up for “mixing time”
- Can get up to an exponential speed-up for “hitting time” (“glued-trees” problem)

- For discrete-time versions, it is usually necessary to add a “coin”.
- Applications include:

  Element distinctness, triangle-finding, element k-distinctness, AND-OR trees, MIN-MAX trees, etc.

“Topological” Algorithms

- Original idea (Freedman) was to define a computing model based on topological quantum field theories, since this might allow the evaluation of certain topological invariants (Jones polynomial). An exact solution is NP-hard (in fact, #P-hard).

- It was shown that these computing models are in fact polynomial time equivalent to “standard” quantum computation. However, a topological model might inspire algorithms for topological problems. One can, e.g., approximate the Jones polynomial and Tutte polynomial at certain points.

Future directions

- Better understanding of non-Abelian HSP
- More sophisticated applications of Amplitude Amplification
- More concrete applications of quantum walk algorithms; also unification of quantum walk paradigms
- Does adiabatic optimization give a superpolynomial speed-up??
- Useful algorithm developments in measurement-based paradigm?

Algorithms for quantum tasks

- Other quantum transformations (e.g. Clebsch-Gordan, wavelet)
- Generating general quantum states
- Quantum error correction
- Quantum signature schemes
- Quantum data compression
- Quantum entanglement concentration
- Coset orbit problem
- Etc. etc.

Further reading

- Algorithms Zoo: http://math.nist.gov/quantum/zoo/

Quantum algorithms for algebraic problems

Andrew M. Childs[*]

Department of Combinatorics & Optimization and Institute for Quantum Computing
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Wim van Dam[†]

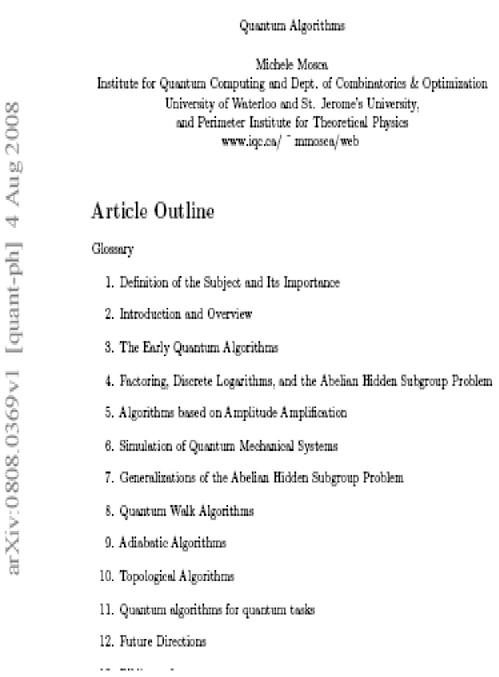
Departments of Computer Science and Physics
University of California, Santa Barbara, California 93106, USA

Quantum computers can execute algorithms that dramatically outperform classical computation. As the best-known example, Shor discovered an efficient quantum algorithm for factoring integers, whereas factoring appears to be difficult for classical computers. Understanding what other computational problems can be solved significantly faster using quantum algorithms is one of the major challenges in the theory of quantum computation, and such algorithms motivate the formidable task of building a large-scale quantum computer. This article reviews the current state of quantum algorithms, focusing on algorithms with superpolynomial speedup over classical computation, and in particular, on problems with an algebraic flavor.

PACS numbers: 03.67.Lx

Contents

[*] amchilds@uwaterloo.ca
[†] vandam@cs.ucsb.edu



Quantum Algorithms

Michele Mosca
Institute for Quantum Computing and Dept. of Combinatorics & Optimization
University of Waterloo and St. Jerome's University,
and Perimeter Institute for Theoretical Physics
www.iqc.ca/ ˜ mmosca/web

Article Outline

Algorithms for Quantum Computers

Jamie Smith and Michele Mosca

1 Introduction

Quantum computing is a new computational paradigm created by reformulating information and computation in a quantum mechanical framework [30, 27]. Since the laws of physics appear to be quantum mechanical, this is the most relevant framework to consider when considering the fundamental limitations of information processing. Furthermore, in recent decades we have seen a major shift from just observing quantum phenomena to actually controlling quantum mechanical systems. We have seen the communication of quantum information over long distances, the "teleportation" of quantum information, and the encoding and manipulation of quantum information in many different physical media. We still appear to be a long way from the implementation of a large-scale quantum computer, however it is a serious goal of many of the world's leading physicists, and progress continues at a fast pace.

In parallel with the broad and aggressive program to control quantum mechanical systems with increased precision, and to control and interact a larger number of subsystems, researchers have also been aggressively pushing the boundaries of what useful tasks one could perform with quantum mechanical devices. These in-

Jamie Smith
Institute for Quantum Computing and Dept. of Combinatorics & Optimization
University of Waterloo,
with support from the Natural Sciences and Engineering Research Council of Canada
e-mail: ja5smith@iqc.ca

Michele Mosca
Institute for Quantum Computing and Dept. of Combinatorics & Optimization
University of Waterloo and St. Jerome's University,
and Perimeter Institute for Theoretical Physics,
with support from the Government of Canada, Ontario-MRI, NSERC, QuantumWorks, MITACS, CIFAR, CRC, ORF, and DTO-ARO
e-mail: mmosca@iqc.ca