Carnegie Mellon University

18-819F: Introduction to Quantum Computing 47-779/785: Quantum Integer Programming & Quantum Machine Learning

Quantum Phase Estimation and Linear Algebraic Systems

Lecture 15

2022.11.02

Electrical & Computer ENGINEERING

TEPPER

USRA Universities Space Research Association

Agenda

- Linear algebraic systems in Hilbert space
  - Eigenvalues of unitary matrices
  - Relationship of the phase of a state vector acted on by a unitary operator to produce an eigenvalue
  - Phase estimation

- Revisiting the quantum Fourier transform
  - Solution of linear algebraic systems using the quantum phase estimation method
  - Harrow, Hassidim and Llyod (HHL) quantum linear systems algorithm

Linear Algebraic Systems

- Some of the most widely used techniques in computing are based on the solution of a system of linear equations that read

$$Ax = b,\ A \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n \text{ and } b \in \mathbb{R}^n, \quad \text{Eqn. (15.1).}$$

- Such a system of equations may commonly arise when attempting to solve machine learning problems.

- When A is a small matrix and invertible, it is generally easy to find the unknown list (components of the vector) in x; such solutions are obtained from

$$x = A^{-1}b \text{ Eqn. (15.2).}$$

- Popular methods of solution (for reasonably sized) matrices include the Gaussian elimination method and the conjugate gradient method. However, when the system of equations exhibited in (15.1) is generated from practical problems of interest such as in machine learning or discretized dynamical differential equations governing behaviors of objects such as aircraft wings, solving such systems of equations can be very challenging.

- There is a growing body of evidence that quantum computing can offer algorithms that can conceivably solve linear systems faster and at lower computational cost.

Linear Systems in Complex Vector Spaces

- Since vector spaces (Hilbert spaces) in quantum computing are complex, the linear system of equations becomes

$$Ax = \lambda x,\ A \in \mathbb{C}^{2n\times 2n}, x \in \mathbb{C}^{2n},\ \text{and } \lambda \in \mathbb{C},\ \text{Eqn. (15.3).}$$

- Here are a few important facts we will use later for the complex linear systems we discuss:

$$Ax = \lambda x \Longrightarrow x = A^{-1}\lambda x \text{ and } A^2 x = \lambda^2 x \ \text{ Eqn. (15.4).}$$

  - Reflections have phases of ± 1; rotations are implemented by $e^{\pm j\vartheta}$, and projections have eigenvalue $\lambda = 1$.

- For a unitary operator U with an eigenvector $|x\rangle$ that operates on a set of m qubits, we can write

$$U|x\rangle = \lambda|x\rangle = e^{2\pi j\vartheta}|x\rangle \quad \text{Eqn. (15.5).}$$

- Because of the unitarity of U, its eigenvalues have the norm 1 or $|\lambda| = 1$, which is equivalent to writing

$$\lambda = e^{2\pi j\vartheta} \text{ Eqn. (15.6).}$$

- Finding the eigenvalues of U is tantamount to determining the primitive roots of unity and translates to figuring out the values of ϑ such that $0 < \vartheta < 1$. This is the phase estimation problem.

Binary Decimals

- A decimal in binary notation can be written as $\vartheta = 0.\vartheta_1\vartheta_2 \dots \vartheta_n$, where each ϑ_i is either 0 or 1.

- One can therefore write

$$\vartheta = 0.\vartheta_1\vartheta_2 \dots \vartheta_n \leftrightarrow \vartheta = \sum_{\ell=1}^{n} \vartheta_\ell 2^{-\ell}$$

- The number 0.75 in decimal is equivalent to the binary 0.11 because we can write

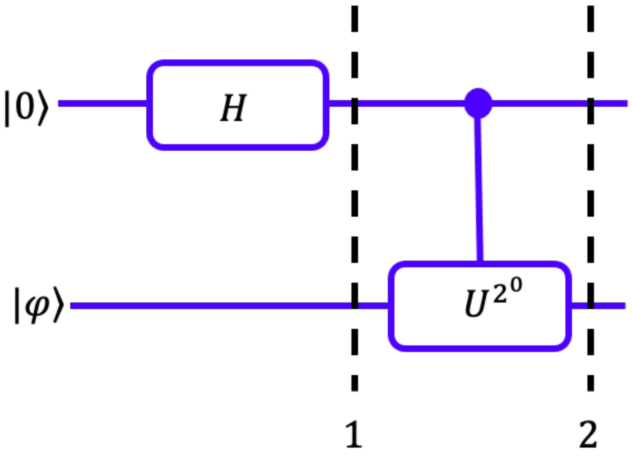
$$0.11 = 1\times2^{-1} + 1\times2^{-2} = 0.75.$$

Phase Estimation

- Since the phase in (15.5) is constrained to be constrained, i.e., $0 < \vartheta < 1$, we can write it as a decimal binary number, thus
$$\vartheta = 0.\vartheta_1\vartheta_2\vartheta_3 \ldots \vartheta_n = \sum_{\ell=1}^{n} \vartheta_\ell 2^{-\ell} \text{ Eqn. (15.6).}$$

- If we approximate $\vartheta \approx 0.\vartheta_1$, then the constraint above implies we can rewrite (15.5) as
$$U|x\rangle = \lambda|x\rangle = e^{2\pi j 0.\vartheta_1}|x\rangle \text{ Eqn. (15.7).}$$

- For simplicity, we take it that $|x\rangle$ is an input to the operator U. A simple circuit that implements the process of making $|x\rangle$ available to U is illustrated on the right. If the input is initialized to $|0\rangle$, we will use a Hadamard to create a superposition of the input state with ($|1\rangle$), thus
$$H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}.$$



- At the plane 1, the state vector available for $U = U^{2^0}$ to operate on is the tensor product
$$\psi_1 = \frac{(|0\rangle+|1\rangle)}{\sqrt{2}} \otimes |\varphi\rangle = \frac{|0\rangle|\varphi\rangle+|1\rangle|\varphi\rangle}{\sqrt{2}} \text{ Eqn. (15.8).}$$

- Recalling that a controlled-operator only changes what it operates on when the control qubit is 1, we see that at plane 2 we will have
$$U\left(\frac{|0\rangle|\varphi\rangle+|1\rangle|\varphi\rangle}{\sqrt{2}}\right) = \frac{|0\rangle|\varphi\rangle+|1\rangle U|\varphi\rangle}{\sqrt{2}} = \frac{|0\rangle|\varphi\rangle+\exp(2\pi j 0.\vartheta_1)|1\rangle|\varphi\rangle}{\sqrt{2}} \quad \text{Eqn. (15.9).}$$

Phase Estimation

- The effect of U on the input to the circuit on the previous slide is Eqn. (15.9), which we rewrite as

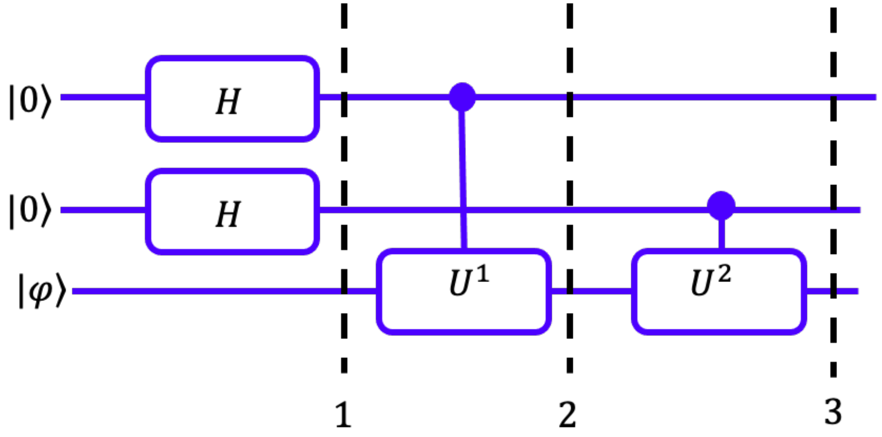
$$\psi_2 = \frac{|0\rangle|\varphi\rangle + \exp(2\pi j 0.\vartheta_1)|1\rangle|\varphi\rangle}{\sqrt{2}} = \left(\frac{|0\rangle + \exp(2\pi j 0.\vartheta_1)|1\rangle}{\sqrt{2}}\right) \otimes |\varphi\rangle \text{ Eqn. (15.10).}$$

- We see that U has written phase information into the first qubit but left the second qubit intact.

- The main idea here is that if we keep applying controlled-U operations at successively higher powers of $2^0, 2^1 \ldots 2^{n-1}$, which means $U^{2^0}, U^{2^1}, \ldots U^{2^{n-1}}(U^1, U^2,\ U^4 \ldots)$, we will encode phase information into the input qubits.

- We will next analyze what happens if we have a two-qubit input register and the operators U^1 and U^2.

Result of Adding a Second Controlled-U Operator



- Adding a second qubit to the the circuit and another controlled-U gate that is squared leads to the circuit on the left. With two qubits, we will also assume that $\vartheta = 0.\vartheta_1\vartheta_2$.

- If we look at plane 2, the state vector is now a tensor that includes the following

$$|\chi_2\rangle = \left(\frac{|0\rangle + \exp(2\pi j 0.\vartheta_1\vartheta_2)|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes |\varphi\rangle \quad \text{Eqn. (15.11).}$$

- The first term is a result of the operation of the Hadamard on the first input qubit $|0\rangle$ and the subsequent effect of operating on the result with a controlled-U^1. The second term is merely a result of operating with the Hadamard on the second input qubit, and the third term is the qubit in the last register.

Action of the Controlled-U^2 Operator

- Action of the controlled-U^2on the second qubit in the circuit on the previous slide can be written as

$$U^2\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\otimes|\varphi\rangle\right)=\frac{|0\rangle+\exp(2\pi j(2\vartheta)|1\rangle)}{\sqrt{2}}\otimes|\varphi\rangle \quad \text{Eqn. (15.12).}$$

- We note that $2\vartheta = 2(0.\vartheta_1\vartheta_2) = 2(\vartheta_1 2^{-1} + \vartheta_2 2^{-2}) = \vartheta_1 + \vartheta_2 2^{-1} = \vartheta_1.\vartheta_2$. Inserting this result into the phase term of (15.12) yields $\exp(2\pi j(\vartheta_1.\vartheta_2)) = \exp(2\pi j(\vartheta_1+0.\vartheta_2) = \exp(2\pi j\vartheta_1)\exp(2\pi j0.\vartheta_2)$.

- The term $\exp(2\pi j\vartheta_1) = 1$ because ϑ_1 is an integer. This means the effect of the Hadamard and the controlled-U^2 operator on the second qubit at plane 3 is

$$\frac{|0\rangle+\exp(2\pi j0.\vartheta_2)|1\rangle}{\sqrt{2}}\otimes|\varphi\rangle \qquad \text{Eqn. (15.13).}$$

- Combining our results, we get the overall state vector at plane 3 for the circuit as

$$|\chi_3\rangle = \left(\frac{|0\rangle+\exp(2\pi j0.\vartheta_1\vartheta_2)|1\rangle}{\sqrt{2}}\right)\otimes\frac{|0\rangle+\exp(2\pi j0.\vartheta_2)|1\rangle}{\sqrt{2}}\otimes|\varphi\rangle \quad \text{Eqn. (15.14).}$$

Generalizing the Effect of Controlled-U^{2^n} Operator

- Clearly, multiple applications of the controlled-U operator raised to various powers of 2 lead to phase encoding into the appropriate qubits in the input register. The effect is as follows

$$\left.\begin{array}{rcl} \exp(2\pi j(2^0\vartheta)) & = & \exp(2\pi j 0.\vartheta_1) \\ \exp(2\pi j(2^1\vartheta)) & = & \exp(2\pi j(0.\vartheta_1\vartheta_2)) \\ \exp\left(2\pi j(2^\ell\vartheta)\right) & = & \exp\left(2\pi j(0.\vartheta_\ell\vartheta_{\ell+1}\ldots)\right) \end{array}\right\} \quad \text{Eqn. (15.15).}$$

- We now know the combined effects of the Hadamard and the controlled-U operators of various powers of 2 encode phase information into the input qubits, but how do we read (get) that information out?

- Since expression (15.14) is reminiscent of the quantum Fourier transform, perhaps one can read the phase values by inverting expression (15.14). Such inversion could be accomplished by the inverse Fourier transform.

Revisiting the Quantum Fourier Transform

- We learned that the QFT of a state vector $|x\rangle^n$, where n indicates the number of bits is

$$QFT^{(2^n)}|x\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left(\frac{2\pi jxy}{2^n}\right)|y\rangle$$

- We can rewrite the equation above as

$$QFT^{(2^n)}|x\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left(\frac{2\pi jxy}{2^n}\right)|y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left(2\pi jx \sum_{\ell=0}^{2^n-1} y_\ell 2^{-\ell}\right)|y\rangle$$

- Rewriting again, we get

$$QFT^{(2^n)}|x\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{y_1=0}^{1} \sum_{y_2=0}^{1} \dots \sum_{y_n=0}^{1} \prod_{\ell=1}^{n} \exp\left(\frac{2\pi jxy_\ell}{2^\ell}\right)|y_k\rangle$$

- And again,

$$QFT^{(2^n)}|x\rangle^n = \frac{1}{\sqrt{2^n}} \prod_{\ell=1}^{n} \left(|0\rangle + \exp\left(\frac{2\pi jxy_\ell}{2^\ell}\right)|1\rangle\right) \quad \text{Eqn. (15.16).}$$

QFT Revisited

- Eqn. (15.16) is a tensor product that can be written out as

$$QFT^{(2^n)}|x\rangle^n = \frac{1}{\sqrt{2^n}}\left(|0\rangle + \exp\left(\frac{2\pi jx}{2^1}\right)|1\rangle\right)\left(|0\rangle + \exp\left(\frac{2\pi jx}{2^2}\right)|1\rangle\right)\ldots\left(|0\rangle + \exp\left(\frac{2\pi jx}{2^n}\right)|1\rangle\right) \text{ Eqn. (15.17).}$$

- By noting that in binary we can write

$$\frac{x}{2^n} = \frac{x_1x_2\ldots x_n}{2^n} = \frac{x_1}{2^1} + \frac{x_2}{2^2} + \cdots \frac{x_n}{2^n} = 0.x_1x_2 \ldots x_n \text{ Eqn. (15.18).}$$

- And that

$$\frac{x}{2^{n-1}} = 2\frac{x_1x_2\ldots x_n}{2^n} = x_1 + 0.x_2x_3 \ldots x_n \text{ Eqn. (15.19).}$$

- Eqn. (15.19) allows us to conclude that

$$\exp\left(\frac{2\pi jx}{2^{n-1}}\right) = \exp(2\pi jx_1)\exp(0.x_2x_3 \ldots x_n) = \exp(0.x_2x_3 \ldots x_n) \text{ Eqn. (15.20)}$$

- Eqns. (15.18), (15.19) and (15.20) permit use to rewrite (15.17) as

$$QFT^{(2^n)}|x\rangle^n = \frac{1}{\sqrt{2^n}}(|0\rangle + \exp(2\pi j0.x_n)|1\rangle)(|0\rangle + \exp(2\pi j0.x_{n-1}x_n)|1\rangle)\ldots(|0\rangle + \exp(2\pi j0.x_1x_2 \ldots x_n)|1\rangle) \text{ Eqn. (15.21).}$$

Relationship of Input Register Data to QFT

- Eqn. (15.21) shows that if our input register has qubits written as $|\vartheta_1\rangle|\vartheta_2\rangle\ldots|\vartheta_n\rangle$, then the QFT would be

$$QFT|\vartheta_1\vartheta_2\ldots\vartheta_n\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + \exp(2\pi j0.\vartheta_1)\,|1\rangle)\,(|0\rangle + \exp(2\pi j0.\vartheta_1\vartheta_2)\,|1\rangle)\ldots\,(|0\rangle + \exp(2\pi j0.\vartheta_1\vartheta_2\ldots\vartheta_n)\,|1\rangle)$$

- Notice that the state vector expression above is similar to our result of the phase estimation Eqn. (15.14) but in reverse order.

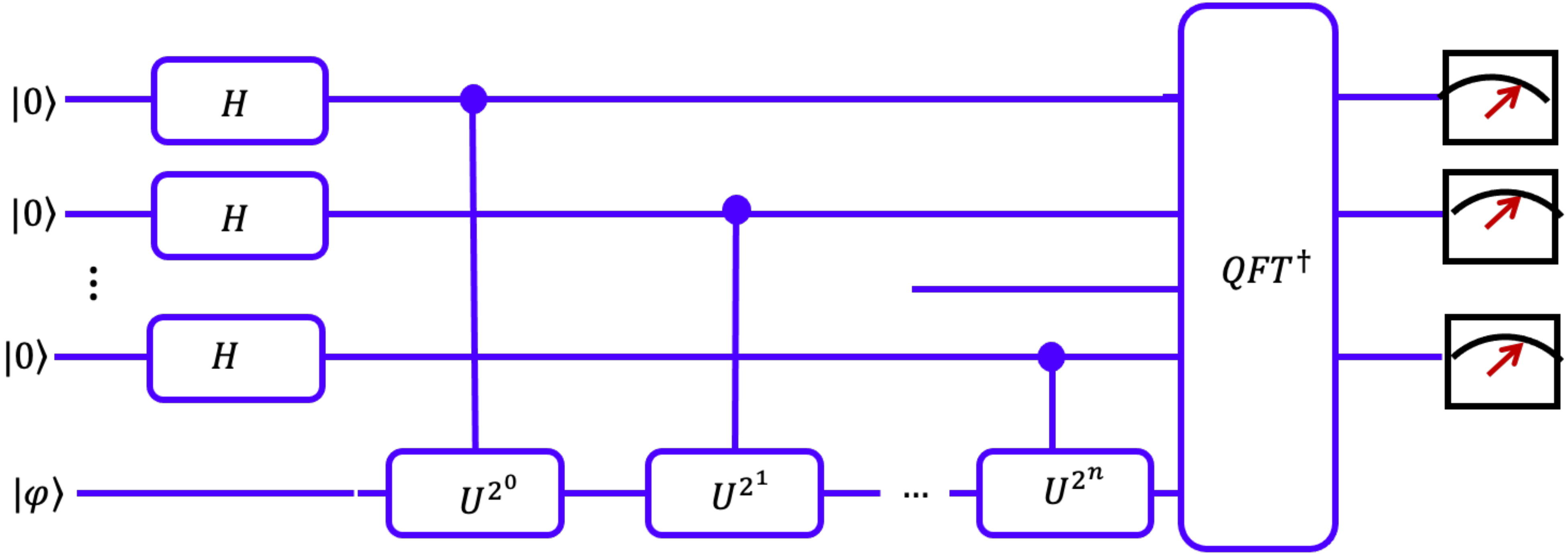
- This identification tells us that to read out the phase information encoded into the input register we simply take the inverse Fourier transform.

- Copying from (15.14), we therefore write

$$QFT^{\dagger}\left(\left(\frac{|0\rangle + \exp(2\pi j0.\vartheta_1\vartheta_2)|1\rangle}{\sqrt{2}}\right)\otimes\frac{|0\rangle + \exp(2\pi j0.\vartheta_2)|1\rangle}{\sqrt{2}}\otimes|\varphi\rangle\right) = |\vartheta_2\rangle\otimes|\vartheta_1\rangle\otimes|\varphi\rangle \text{ Eqn. (15.22)}$$

General Phase Estimation Circuit

- Based on the simple circuits for the one- and two-qubit inputs that were acted on by the Hadamard and successive controlled-U operators raised to increasing powers of 2, a general circuit for phase estimation can be composed as shown below.



Sequence of Equations for the General QPE

- The action of the Hadamard operators on the input register results in

$$H^{\otimes n}|0\rangle^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \cdots \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \quad \text{Eqn. (15.23).}$$

- After applying controlled-U operators, the circuit gives

$$U\left(H^{\otimes n}|0\rangle^{\otimes n}\right)$$
$$= \frac{1}{\sqrt{2^n}}\left(|0\rangle + \exp(2\pi j 0.\vartheta_1\vartheta_2 \ldots \vartheta_n)\,|1\rangle\right) \otimes \left(|0\rangle + \exp(2\pi j 0.\vartheta_2 \ldots \vartheta_n)\,|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + \exp(2\pi j 0.\vartheta_n)|1\rangle\right)$$

- Taking the inverse quantum Fourier transform of the expression above results in

$$QFT^{\dagger}\left(U\left(H^{\otimes n}|0\rangle^{\otimes n}\right)\right) = |\vartheta_n\rangle \otimes |\vartheta_{n-1}\rangle \otimes \ldots \otimes |\vartheta_1\rangle \text{ Eqn. (15.24).}$$

- A quantum measurement after the step above should then yield the estimated phase.

Linear Algebra System of Equations

- We stated earlier that a linear algebraic system of equations is given by

$$Ax = b,\ A \in \mathbb{C}^{2n\times 2n}, x \in \mathbb{C}^{2n},\ \text{and b} \in \mathbb{C},\ \text{Eqn. (15.3).}$$

- In order to solve this problem with a quantum computer we must prepare the input state vector(s) and Hermitian operators.

- We require that the system be rewritten as

$$A|x\rangle = |b\rangle\ \text{Eqn. (15.25).}$$

- Since A is Hermitian, it can be written in a spectrally decomposed form, thus

$$A = \sum_{k=0}^{N-1} \lambda_k |u_k\rangle\langle u_k|,\ \lambda_k \in \mathbb{R}\ \ \text{Eqn. (15.26)}$$

- In (15.26), u_k are the eigenvectors of A and the λ_k are the corresponding eigenvalues. In principle, one can immediately determine the inverse of A, thus

$$A^{-1} = \sum_{k=0}^{N-1} \lambda_k^{-1} |u_k\rangle\langle u_k|,\ \lambda_k \in \mathbb{R}\ \ \text{Eqn. (15.27)}$$

Solution of the Linear Algebraic System of Equation

- The state vector $|b\rangle$ on the right-hand side of (15.25) can be expanded as linear combination of basis states, thus

$$|b\rangle = \sum_{k=0}^{N-1} b_k |u_k\rangle \ \ b_k \in \mathbb{C} \quad \text{Eqn. (15.28).}$$

- Our goal is to eventually write the solution of the problem in the form

$$|x\rangle = A^{-1}|b\rangle = \sum_{k=0}^{N-1} \lambda_k^{-1} |u_k\rangle \langle u_k|\, b_k |u_k\rangle = \sum_{k=0}^{N-1} \lambda_k^{-1} b_k |u_k\rangle, \ \text{Eqn. (15.29).}$$

- Note that in(15.29), we have used the fact that $\langle u_k | u_k \rangle = 1$.

- Since this is a quantum system, all state vectors are (or must be) normalized and the operator must be unitary.

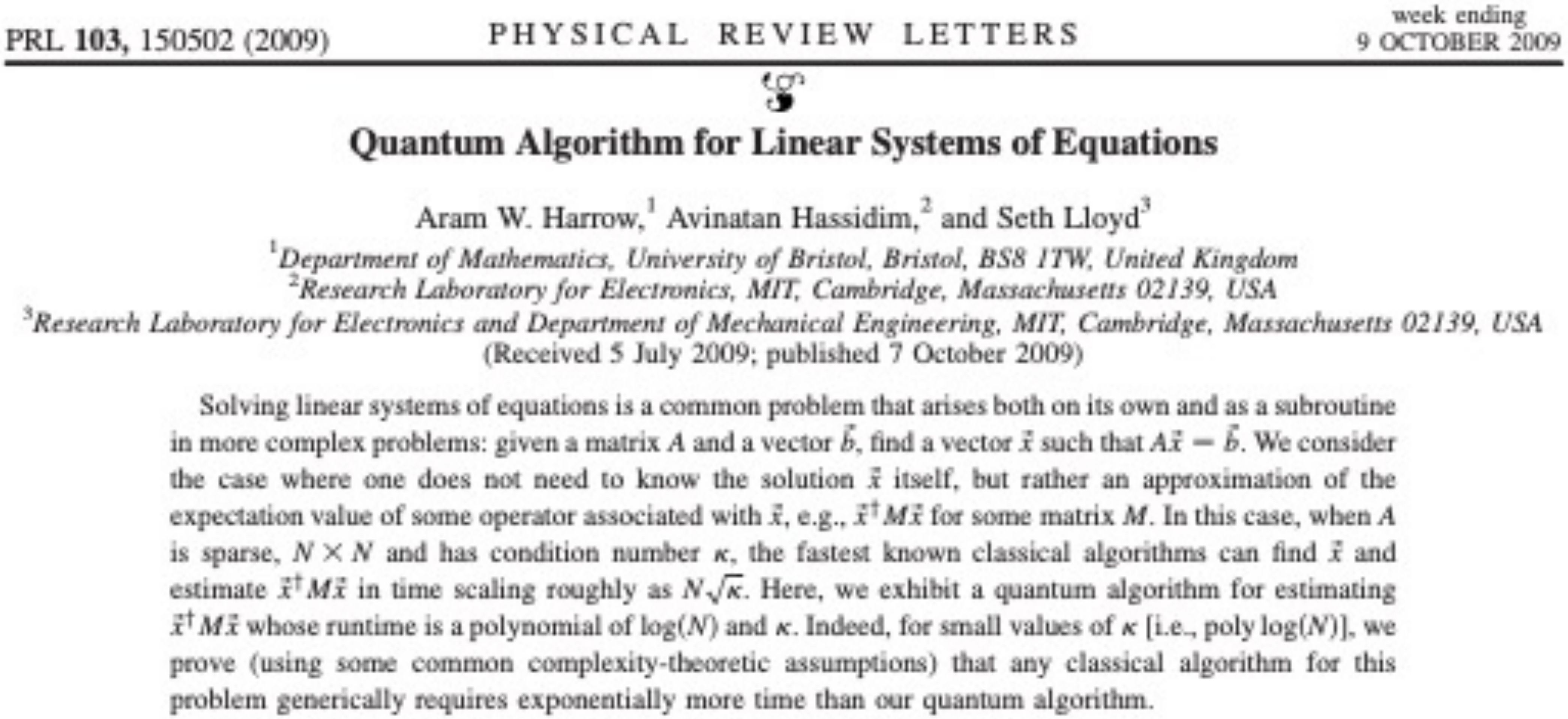
Quantum Algorithm for Solving Linear Systems

- There is a quantum algorithm called the HHL algorithm, named after its creators (Harrow, Hassadim and Lloyd) that was developed to solve linear systems.

- The algorithm assumes that a quantum computer is available. It allocates 3 quantum registers initialized to $|0\rangle$ for the task: register 1 is labeled as n_λ and it stores the eigenvalues of the matrix A; register 2 is labeled n_b for storing the solution vector. It is usually also assumed that the problem is of size $N = 2^{n_b}$. There is also usually a third register that is used for storing intermediate steps in the calculations.

- The HHL algorithm is one of the most widely studied approach to solving a linear systems of equations. Although it doesn't actually give a final solution, it provides a framework for how to go about solving such problems.

Original Source for Discussion of HHL Algorithm



PRL **103,** 150502 (2009) PHYSICAL REVIEW LETTERS week ending 9 OCTOBER 2009

Quantum Algorithm for Linear Systems of Equations

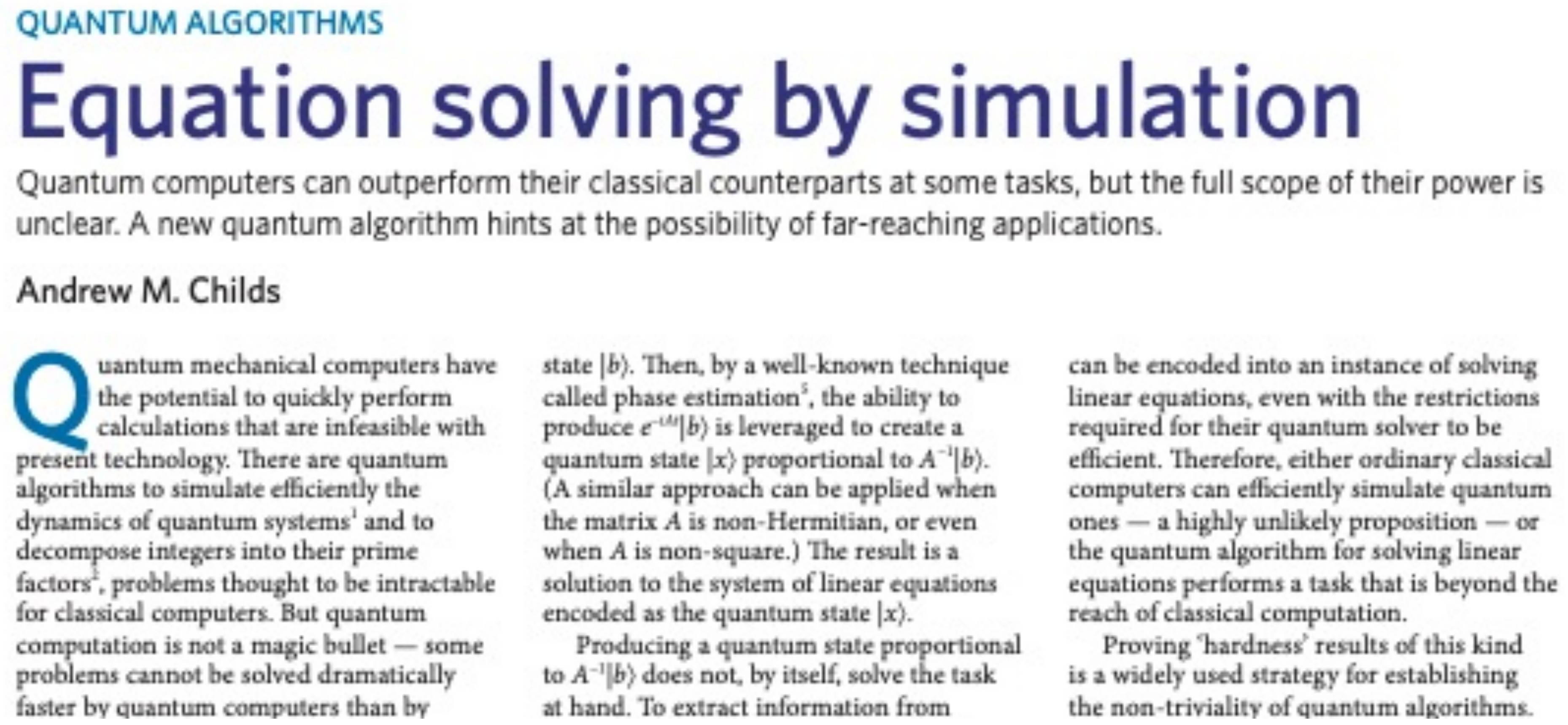
Aram W. Harrow,[1] Avinatan Hassidim,[2] and Seth Lloyd[3]

[1]*Department of Mathematics, University of Bristol, Bristol, BS8 1TW, United Kingdom*
[2]*Research Laboratory for Electronics, MIT, Cambridge, Massachusetts 02139, USA*
[3]*Research Laboratory for Electronics and Department of Mechanical Engineering, MIT, Cambridge, Massachusetts 02139, USA*

(Received 5 July 2009; published 7 October 2009)

Solving linear systems of equations is a common problem that arises both on its own and as a subroutine in more complex problems: given a matrix A and a vector $\vec{b}$, find a vector $\vec{x}$ such that $A\vec{x} = \vec{b}$. We consider the case where one does not need to know the solution $\vec{x}$ itself, but rather an approximation of the expectation value of some operator associated with $\vec{x}$, e.g., $\vec{x}^\dagger M\vec{x}$ for some matrix M. In this case, when A is sparse, $N \times N$ and has condition number κ, the fastest known classical algorithms can find $\vec{x}$ and estimate $\vec{x}^\dagger M\vec{x}$ in time scaling roughly as $N\sqrt{\kappa}$. Here, we exhibit a quantum algorithm for estimating $\vec{x}^\dagger M\vec{x}$ whose runtime is a polynomial of $\log(N)$ and κ. Indeed, for small values of κ [i.e., poly $\log(N)$], we prove (using some common complexity-theoretic assumptions) that any classical algorithm for this problem generically requires exponentially more time than our quantum algorithm.

Commentary on the HHL Algorithm



QUANTUM ALGORITHMS

Equation solving by simulation

Quantum computers can outperform their classical counterparts at some tasks, but the full scope of their power is unclear. A new quantum algorithm hints at the possibility of far-reaching applications.

Andrew M. Childs

Quantum mechanical computers have the potential to quickly perform calculations that are infeasible with present technology. There are quantum algorithms to simulate efficiently the dynamics of quantum systems[1] and to decompose integers into their prime factors[2], problems thought to be intractable for classical computers. But quantum computation is not a magic bullet — some problems cannot be solved dramatically faster by quantum computers than by

state $|b\rangle$. Then, by a well-known technique called phase estimation[5], the ability to produce $e^{-iAt}|b\rangle$ is leveraged to create a quantum state $|x\rangle$ proportional to $A^{-1}|b\rangle$. (A similar approach can be applied when the matrix A is non-Hermitian, or even when A is non-square.) The result is a solution to the system of linear equations encoded as the quantum state $|x\rangle$.

Producing a quantum state proportional to $A^{-1}|b\rangle$ does not, by itself, solve the task at hand. To extract information from

can be encoded into an instance of solving linear equations, even with the restrictions required for their quantum solver to be efficient. Therefore, either ordinary classical computers can efficiently simulate quantum ones — a highly unlikely proposition — or the quantum algorithm for solving linear equations performs a task that is beyond the reach of classical computation.

Proving 'hardness' results of this kind is a widely used strategy for establishing the non-triviality of quantum algorithms.

Nature Physics | Vol 5| December 2009 | Page 861

HHL Algorithm

- Key steps of the HHL Algorithm:

1. Load data from vector $|b\rangle \in \mathbb{C}^N$ into register n_b so that $|0\rangle_{n_b} \longrightarrow |b\rangle_{n_b}$.

2. Calculate eigenvalues of A via the operation $U = e^{2\pi jt} := \sum_{k=0}^{N-1} \exp(j\lambda_k t)\, |u_k\rangle\langle u_k|$.

3. The register quantum state is now in the eigen basis of A and is written as $\sum_{k=0}^{N-1} b_k |\, \lambda_k\rangle_{n_\lambda} |u_k\rangle_{n_b}$. Note that $|\lambda_k\rangle_{n_\lambda}$ is a n_λ-bit binary representation of the eigenvalues of A. One then adds an auxiliary qubit before a rotation is applied; the rotation is dependent on the values of the eigenvalues, λ_k.

…HHL Algorithm

- The auxiliary qubit is usually written as

$$\sum_{k=0}^{N-1} b_k|\lambda_k\rangle_{n_\lambda}|u_k\rangle_{n_b}\left(\sqrt{1-\frac{C^2}{\lambda_k^2}}\,|0\rangle+\frac{C}{\lambda_k}|1\rangle\right)$$

- In the auxiliary qubit, C is a normalization constant that is assumed to obey the condition $|C| < \lambda_{min}$.

4. The inverse quantum phase estimation $(QPE^\dagger)$ algorithm is then applied to the auxiliary qubit to give

$$\sum_{k=0}^{N-1} b_k|0\rangle_{n_\lambda}|u_k\rangle_{n_b}\left(\sqrt{1-\frac{C^2}{\lambda_k^2}}\,|0\rangle+\frac{C}{\lambda_k}|1\rangle\right)$$

…HHL Algorithm

- 5. One then usually measures the auxiliary qubit in the computational basis. If the measurement collapses the state to the $|1\rangle$ basis, then one obtains the normalized result

$$\left(\sqrt{\frac{1}{\sum_{k=0}^{N-1}\frac{|b_k|^2}{|\lambda_k|^2}}}\right)\sum_{k=0}^{N-1}\frac{b_k}{\lambda_k}|0\rangle_{n_\lambda}|u_k\rangle_{n_b}$$

- This is the desired solution up to a normalization constant that can be appropriately scaled for the final solution.

- It is often the case that one does not need the actual solution but some function that depends on certain components of the of the solution vector. In such cases, a measurable observable M might be suitably applied for calculating the desired function, thus,

$$f(x) = \langle x|M|x\rangle.$$

Summary

- Developed the quantum phase estimation (QPE) method
  - Related QPE to the QFT
  - Review quantum gate circuit for QPE
- Discussed a quantum computing method for solving a linear algebraic system
  - Reviewed the HHL algorithm