Quantum Algorithms for Systems of Linear Equations

Rolando Somma

Theoretical Division
Los Alamos National Laboratory

Joint work with

Andrew Childs
Maryland

Robin Kothari
Microsoft

Davide Orsucci
Vienna

Yigit Subasi
Los Alamos

Workshop at the Intersection of Machine Learning and Quantum Information
University of Maryland
September 2018

LDRD
innovation for our nation

Los Alamos
NATIONAL LABORATORY
EST.1943

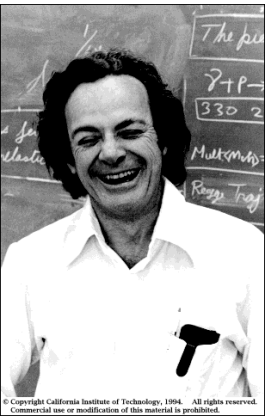
Quantum Algorithms for Systems of Linear Equations

References:

- “Quantum linear systems algorithm with exponentially improved dependence on precision”, A.M. Childs, R. Kothari, and **R.D. Somma**, SIAM J. Comp. **46**, 1920 (2017).

- “Quantum algorithms for linear systems inspired by adiabatic quantum computing”, Y. Subasi, **R.D. Somma**, and D. Orsucci, arXiv:1805.10549 (2018).

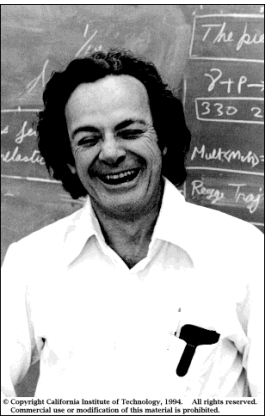
A brief history of results in quantum computing


© Copyright California Institute of Technology, 1994. All rights reserved.
Commercial use or modification of this material is prohibited.

- Simulating quantum systems was the main motivation behind Feynman's idea of a quantum computer (1982).

- For example, algorithms for simulating the dynamics of n spin systems with classical computers have complexity that is exponential in n. Quantum algorithms, in principle, have only complexity that is polynomial in n.

A brief history of results in quantum computing

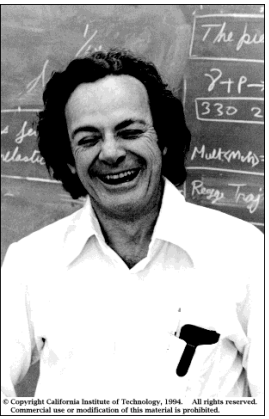

© Copyright California Institute of Technology, 1994. All rights reserved. Commercial use or modification of this material is prohibited.

- Simulating quantum systems was the main motivation behind Feynman's idea of a quantum computer (1982).

- For example, algorithms for simulating the dynamics of n spin systems with classical computers have complexity that is exponential in n. Quantum algorithms, in principle, have only complexity that is polynomial in n.

- Peter Shor discovers a quantum algorithm for efficient factorization of integers with important applications to cryptography. Shor's algorithm results in a superpolynomial quantum speedup (1994). His result was a main motivation for the discovery of other quantum algorithms.

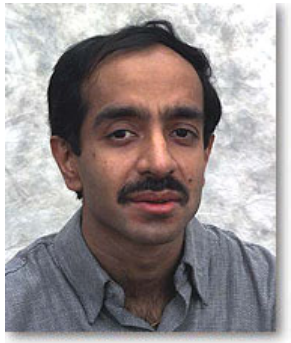
A brief history of results in quantum computing


© Copyright California Institute of Technology, 1994. All rights reserved. Commercial use or modification of this material is prohibited.

- Simulating quantum systems was the main motivation behind Feynman's idea of a quantum computer (1982).

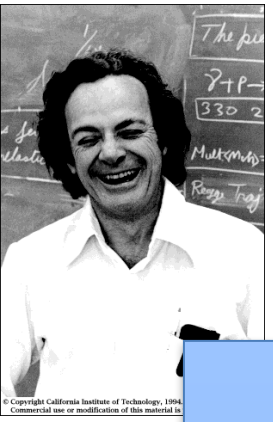
- For example, algorithms for simulating the dynamics of n spin systems with classical computers have complexity that is exponential in n. Quantum algorithms, in principle, have only complexity that is polynomial in n.

- Peter Shor discovers a quantum algorithm for efficient factorization of integers with important applications to cryptography. Shor's algorithm results in a superpolynomial quantum speedup (1994). His result was a main motivation for the discovery of other quantum algorithms.



- L. Grover discovers a quantum algorithm for unstructured search resulting in a polynomial (quadratic) quantum speedup (1997).A main idea in Grover's result (amplitude amplification) has been extensively used in other quantum algorithms for problems such as optimization, search, and more.

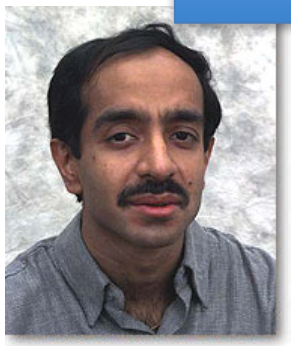
A brief history of results in quantum computing



- Simulating quantum systems was the main motivation behind Feynman's idea of a quantum computer (1982).

- For example, algorithms for simulating the dynamics of n spin systems with classical computers have complexity that is

Other quantum algorithms for linear algebra problems?



- L. Grover discovers a quantum algorithm for unstructured search resulting in a polynomial (quadratic) quantum speedup (1997).A main idea in Grover's result (amplitude amplification) has been extensively used in other quantum algorithms for problems such as optimization, search, and more.

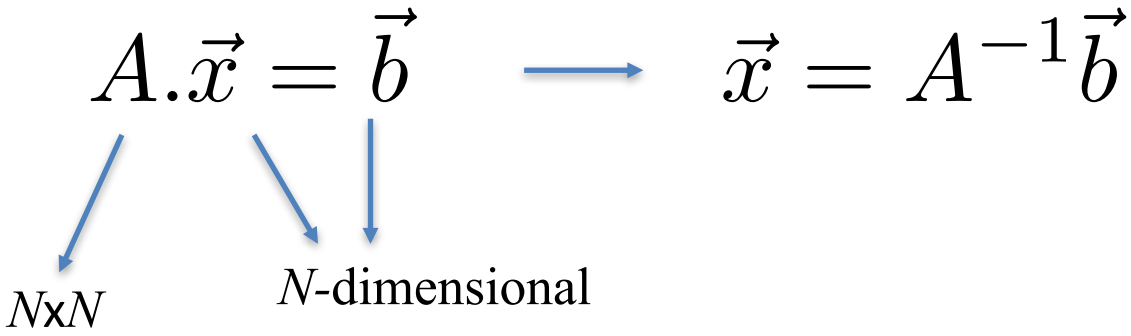
Linear Equations: An important problem

Let's consider the problem of solving a system of linear equations or the related problem of inverting a matrix:

$$A.\vec{x} = \vec{b} \quad \longrightarrow \quad \vec{x} = A^{-1}\vec{b}$$

Linear Equations: An important problem

Let's consider the problem of solving a system of linear equations or the related problem of inverting a matrix:



Linear Equations: An important problem

Let's consider the problem of solving a system of linear equations or the related problem of inverting a matrix:

$$A.\vec{x} = \vec{b} \quad \longrightarrow \quad \vec{x} = A^{-1}\vec{b}$$

- There is a variety of classical algorithms to solve this problem. Nevertheless, even when the matrix A and vector $\vec{b}$ are sparse, the complexity of ``exact'' classical algorithms is at least linear in N.

Linear Equations: An important problem

Let's consider the problem of solving a system of linear equations or the related problem of inverting a matrix:

$$A.\vec{x} = \vec{b} \quad \longrightarrow \quad \vec{x} = A^{-1}\vec{b}$$

- There is a variety of classical algorithms to solve this problem. Nevertheless, even when the matrix A and vector $\vec{b}$ are sparse, the complexity of ``exact'' classical algorithms is at least linear in N.

- A result [HHL08]: Quantum computers can prepare a quantum state proportional to the solution of the system in time that is polynomial in the condition number, inverse of precision, and the logarithm of the dimension (under some assumptions).

Linear Equations: An important problem

Let's consider the problem of solving a system of linear equations or the related problem of inverting a matrix:

$$A.\vec{x} = \vec{b} \quad \longrightarrow \quad \vec{x} = A^{-1}\vec{b}$$

- There is a variety of classical algorithms to solve this problem. Nevertheless, even when the matrix A and vector $\vec{b}$ are sparse, the complexity of ``exact'' classical algorithms is at least linear in N.

- A result [HHL08]: Quantum computers can prepare a quantum state proportional to the solution of the system in time that is polynomial in the condition number, inverse of precision, and the logarithm of the dimension (under some assumptions).

- Note: This is a somewhat different problem (QLSP) and classical algorithms may do better in this case. However, the QLSP is BQP-Complete.

Quantum Linear System Problem (QLSP)

$$A.\vec{x} = \vec{b}$$

Assumptions

- A is Hermitian of dimension NxN
- A is s-sparse
- A is invertible and its condition number is $\kappa<\infty$
- The spectral norm of A is bounded by 1

Quantum Linear System Problem (QLSP)

$$A.\vec{x} = \vec{b}$$

Assumptions

- A is Hermitian of dimension NxN
- A is s-sparse
- A is invertible and its condition number is $\kappa<\infty$
- The spectral norm of A is bounded by 1

We define the quantum states $|b\rangle := \frac{\sum_{i=0}^{N-1} b_i|i\rangle}{\|\sum_{i=0}^{N-1} b_i|i\rangle\|}$ and $|x\rangle := \frac{\sum_{i=0}^{N-1} x_i|i\rangle}{\|\sum_{i=0}^{N-1} x_i|i\rangle\|}$

Quantum Linear System Problem (QLSP)

$$A.\vec{x} = \vec{b}$$

Assumptions

- A is Hermitian of dimension NxN
- A is s-sparse
- A is invertible and its condition number is $\kappa<\infty$
- The spectral norm of A is bounded by 1

We define the quantum states $|b\rangle := \frac{\sum_{i=0}^{N-1} b_i|i\rangle}{\|\sum_{i=0}^{N-1} b_i|i\rangle\|}$ and $|x\rangle := \frac{\sum_{i=0}^{N-1} x_i|i\rangle}{\|\sum_{i=0}^{N-1} x_i|i\rangle\|}$

Proportional to $\vec{b}$

Proportional to $\vec{x}$

Quantum Linear System Problem (QLSP)

$$A.\vec{x} = \vec{b}$$

Assumptions

- A is Hermitian of dimension NxN
- A is s-sparse
- A is invertible and its condition number is $\kappa<\infty$
- The spectral norm of A is bounded by 1

We define the quantum states $|b\rangle := \frac{\sum_{i=0}^{N-1} b_i|i\rangle}{\|\sum_{i=0}^{N-1} b_i|i\rangle\|}$ and $|x\rangle := \frac{\sum_{i=0}^{N-1} x_i|i\rangle}{\|\sum_{i=0}^{N-1} x_i|i\rangle\|}$

Let U_A be a procedure that runs in time T_A and computes the entries of A

Let U_b be a procedure that runs in time T_b and prepares the state $|b\rangle$

Quantum Linear System Problem (QLSP)

$$A.\vec{x} = \vec{b}$$

Assumptions

- A is Hermitian of dimension NxN
- A is s-sparse
- A is invertible and its condition number is $\kappa<\infty$
- The spectral norm of A is bounded by 1

We define the quantum states $|b\rangle := \frac{\sum_{i=0}^{N-1} b_i|i\rangle}{\|\sum_{i=0}^{N-1} b_i|i\rangle\|}$ and $|x\rangle := \frac{\sum_{i=0}^{N-1} x_i|i\rangle}{\|\sum_{i=0}^{N-1} x_i|i\rangle\|}$

Let $C_A(t,\epsilon)$ be the cost of simulating e^{-iAt} with precision ϵ

Let U_b be a procedure that runs in time T_b and prepares the state $|b\rangle$

The goal is to prepare a quantum state $|\tilde{x}\rangle$ such that $\||\tilde{x}\rangle - |x\rangle\| \leq \epsilon$ with probability $\geq 1/2$, with a flag indicating success

Quantum Linear System Problem (QLSP)

$$A.\vec{x} = \vec{b}$$

Assumptions

- A is Hermitian of dimension NxN
- A is s-sparse
- A is invertible and its condition number is $\kappa<\infty$
- The spectral norm of A is bounded by 1

We define the quantum states $|b\rangle := \frac{\sum_{i=0}^{N-1} b_i|i\rangle}{\|\sum_{i=0}^{N-1} b_i|i\rangle\|}$ and $|x\rangle := \frac{\sum_{i=0}^{N-1} x_i|i\rangle}{\|\sum_{i=0}^{N-1} x_i|i\rangle\|}$

Let $C_A(t,\epsilon)$ be the cost of simulating e^{-iAt} with precision ϵ

Let U_b be a procedure that runs in time T_b and prepares the state $|b\rangle$

The goal is to prepare a mixed state ρ_x such that $\frac{1}{2}\mathrm{Tr}\,|\rho_x - |x\rangle\langle x|\,| \leq \epsilon$

Hamiltonian simulation

Note: Recent advances in Hamiltonian simulation resulted in

$$C_A(t,\epsilon) = \tilde{O}(tsT_A \log(t/\epsilon))$$

- Complexity almost linear in the evolution time
- Complexity is polylogarithmic in the inverse of a precision parameter

D. Berry, A. Childs, R. Cleve, R. Kothari, and RDS, PRL 114, 090502 (2015)
D. Berry, A. Childs, and R. Kothari, FOCS 2015, 792 (2015)
G.H. Low and I. Chuang, PRL 118, 010501 (2017)

Quantum Linear System Problem (QLSP)

Some applications:

- In physics, where the goal is to compute the expectation value of the inverse of a matrix. This idea was used in [1] for obtaining the resistance of a network.

- In stat mech, where, e.g., estimating the hitting time of a Markov chain also reduces to computing the expectation value of the inverse of a matrix [2]

- In ML, for solving problems related to least-squares estimation [3], by applying the pseudoinverse: $\arg\min_{\lambda\in\mathbb{C}^M} \|F\vec{\lambda} - \vec{y}\| \to \lambda = \frac{1}{F^\dagger F} F^\dagger \vec{y}\,.$

- For solving certain linear differential equations [4]: $\dot{\vec{x}}(t) = A(t)\vec{x}(t) + \vec{b}(t)$

[1] G. Wang, arXiv:1311.1851 (2013). [2] A. Chowdhury and R. Somma, QIC 17, 0041 (2017)
[3] N. Wiebe, D. Braun, and S. Lloyd, PRL 109, 050505 (2012).
[4] D. Berry, A. Childs, A. Ostrander, and G. Wang, CMP 356, 1057 (2017)

Quantum Linear System Problem (QLSP)

A note: Even for those applications, a number of assumptions must be made in order to obtain quantum speedups. These assumptions include efficient preparation of certain states (of exp many amplitudes), nice scaling of the condition number, and solving certain problems like computing expectation values. For these reasons, shown quantum speedups are typically polynomial.

The HHL Algorithm for the QLSP [5]

[HHL08] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa/\epsilon, \epsilon/\kappa))\right]$$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

The HHL Algorithm for the QLSP [5]

[HHL08] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa/\epsilon, \epsilon/\kappa))\right]$$

Considering that many Hamiltonians can be simulated efficiently on quantum computers, the complexity dependence on the dimension is small (e.g., logarithmic)

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

Improvements of the HHL Algorithm [6]

[HHL08] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa/\epsilon, \epsilon/\kappa))\right]$$

Considering that many Hamiltonians can be simulated efficiently on quantum computers, the complexity dependence on the dimension is small (e.g., logarithmic)
Further improvements by Ambainis (Variable Time Amplitude Amplification or VTAA):

[6] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa T_b + C_A(\kappa/\epsilon^3, \epsilon)\right]$$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)
[6] A. Ambainis, STACS 14, 636 (2012)

Improvements of the HHL Algorithm [6]

[HHL08] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa/\epsilon, \epsilon/\kappa))\right]$$

Considering that many Hamiltonians can be simulated efficiently on quantum computers, the complexity dependence on the dimension is small (e.g., logarithmic)
Further improvements by Ambainis (Variable Time Amplitude Amplification or VTAA):

[6] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa T_b + C_A(\kappa/\epsilon^3, \epsilon)\right] \longrightarrow \text{Almost linear in } \kappa!$$

- Note that the best Hamiltonian simulation methods have query and gate complexities almost linear in evolution time and logarithmic in precision

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)
[6] A. Ambainis, STACS 14, 636 (2012)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

Let $A|v_i\rangle = \lambda_i|v_i\rangle$. Then, $|b\rangle = \sum_{j=0}^{N-1} c_i|v_i\rangle$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

Let $A|v_i\rangle = \lambda_i|v_i\rangle$. Then, $|b\rangle = \sum_{j=0}^{N-1} c_i|v_i\rangle$

We now use phase estimation to estimate the eigenvalues as follows:

$$|b\rangle \rightarrow \sum_{j=0}^{N-1} c_j|v_j\rangle_I|\tilde{\lambda}_j\rangle_E$$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)
[6] A. Ambainis, STACS 14, 636 (2012)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

Let $A|v_i\rangle = \lambda_i |v_i\rangle$. Then, $|b\rangle = \sum_{j=0}^{N-1} c_i |v_i\rangle$

We now use phase estimation to estimate the eigenvalues as follows:

$$|b\rangle \rightarrow \sum_{j=0}^{N-1} c_j |v_j\rangle_I |\tilde{\lambda}_j\rangle_E$$

This register contains the eigenvalue estimate (superposition):

- It suffices to have the estimate with relative precision ϵ
- Order $\log(\kappa/\epsilon)$ ancillary qubits

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

Let $A|v_i\rangle = \lambda_i|v_i\rangle$. Then, $|b\rangle = \sum_{j=0}^{N-1} c_i|v_i\rangle$

We now use phase estimation to estimate the eigenvalues as follows:

$$|b\rangle \rightarrow \sum_{j=0}^{N-1} c_j|v_j\rangle_I|\tilde{\lambda}_j\rangle_E$$

Then we implement the conditional rotation: $|\tilde{\lambda}_j\rangle_E \rightarrow |\tilde{\lambda}_j\rangle_E \left(\frac{1}{\kappa\tilde{\lambda}_j}|0\rangle_O + \sqrt{1 - \frac{1}{\kappa^2\tilde{\lambda}_j^2}}|1\rangle_O\right)$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

Let $A|v_i\rangle = \lambda_i|v_i\rangle$. Then, $|b\rangle = \sum_{j=0}^{N-1} c_i|v_i\rangle$

We now use phase estimation to estimate the eigenvalues as follows:

$$|b\rangle \rightarrow \sum_{j=0}^{N-1} c_j|v_j\rangle_I|\tilde{\lambda}_j\rangle_E$$

Then we implement the conditional rotation: $|\tilde{\lambda}_j\rangle_E \rightarrow |\tilde{\lambda}_j\rangle_E \left(\frac{1}{\kappa\tilde{\lambda}_j}|0\rangle_O + \sqrt{1 - \frac{1}{\kappa^2\tilde{\lambda}_j^2}}|1\rangle_O\right)$

Undo phase estimation

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

HHL has three registers: I that holds the state $|b\rangle$, E that holds an estimate of the eigenvalue, and O that is an ancilla qubit. $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_E \otimes \mathcal{H}_O$

Let $A|v_i\rangle = \lambda_i|v_i\rangle$. Then, $|b\rangle = \sum_{j=0}^{N-1} c_i|v_i\rangle$

We now use phase estimation to estimate the eigenvalues as follows:

$$|b\rangle \rightarrow \sum_{j=0}^{N-1} c_j|v_j\rangle_I|\tilde{\lambda}_j\rangle_E$$

Then we implement the conditional rotation: $|\tilde{\lambda}_j\rangle_E \rightarrow |\tilde{\lambda}_j\rangle_E \left(\frac{1}{\kappa\tilde{\lambda}_j}|0\rangle_O + \sqrt{1 - \frac{1}{\kappa^2\tilde{\lambda}_j^2}}|1\rangle_O\right)$

Undo phase estimation

Amplitude amplification for amplifying the amplitude of the $|0\rangle_O$ state

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

Roughly, the scaling of the HHL algorithm can be analyzed from the worst case:

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

Roughly, the scaling of the HHL algorithm can be analyzed from the worst case:

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

The action of 1/ A will roughly create the equal superposition state, so both are equally important

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

Roughly, the scaling of the HHL algorithm can be analyzed from the worst case:

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

The action of 1/ A will roughly create the equal superposition state, so both are equally important

For the desired precision we need to evolve with A for time of order κ/ϵ

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

Roughly, the scaling of the HHL algorithm can be analyzed from the worst case:

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

The action of 1/ A will roughly create the equal superposition state, so both are equally important

For the desired precision we need to evolve with A for time of order κ/ϵ

The action of 1/ ($\kappa\, A$) on the state reduces its amplitude by order 1/ κ and order κ amplitude amplification rounds are needed

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

Roughly, the scaling of the HHL algorithm can be analyzed from the worst case:

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

The action of 1/ A will roughly create the equal superposition state, so both are equally important

For the desired precision we need to evolve with A for time of order κ/ϵ

The action of 1/ (κ A) on the state reduces its amplitude by order 1/ κ and order κ amplitude amplification rounds are needed

From here we see that we need to evolve with A for time that is, at least, order κ^2/ϵ

[5] Harrow, Hassidim, Lloyd, PRL 103, 150502 (2009)

A quick view of the HHL algorithm and VTAA

How can we improve this time complexity to something that is almost linear in the condition number?

[6] A. Ambainis, STACS 14, 636 (2012)

A quick view of the HHL algorithm and VTAA

How can we improve this time complexity to something that is almost linear in the condition number?

One answer is via Variable Time Amplitude Amplification (VTAA) [6]

[6] A. Ambainis, STACS 14, 636 (2012)

A quick view of the HHL algorithm and VTAA

How can we improve this time complexity to something that is almost linear in the condition number?

One answer is via Variable Time Amplitude Amplification (VTAA) [6]

The rough idea is as follows (again, considering the worst case):

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

- First we do a bad-precision phase estimation to distinguish large from small eigenvalues. This may be done evolving with A for time independent of κ
- Then we implement a rough approximation of $1/\kappa A$ to eigenstates of large eigenvalue
- We need order κ amplitude amplification steps
- We implement an accurate approximation of $1/\kappa A$ to eigenstates of small eigenvalue
- Amplitude amplification for order 1 steps
- Undo phase estimation or apply the Fourier transform

[6] A. Ambainis, STACS 14, 636 (2012)

A quick view of the HHL algorithm and VTAA

How can we improve this time complexity to something that is almost linear in the condition number?

One answer is via Variable Time Amplitude Amplification (VTAA) [6]

The rough idea is as follows (again, considering the worst case):

$$|b\rangle = (1/\kappa)|v_{1/\kappa}\rangle + \sqrt{1 - 1/\kappa^2}|v_1\rangle$$

- First we do a bad-precision phase estimation to distinguish large from small eigenvalues. This may be done evolving with A for time independent of κ
- Then we implement a rough approximation of $1/\kappa A$ to eigenstates of large eigenvalue
- We need order κ amplitude amplification steps
- We implement an accurate approximation of $1/\kappa A$ to eigenstates of small eigenvalue
- Amplitude amplification for order 1 steps
- Undo phase estimation or apply the Fourier transform

The complexity of VTAA in terms of precision is worse than that of HHL

[6] A. Ambainis, STACS 14, 636 (2012)

This talk: two quantum algorithms for the QSLP

- I will present two quantum algorithms for the QLSP that improve previous results in different ways:

[7] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon/\kappa))\right]$$

[7] A. Childs, R. Kothari, **RDS**, SIAM J. Comp. **46**, 1920 (2017).

This talk: two quantum algorithms for the QSLP

- I will present two quantum algorithms for the QLSP that improve previous results in different ways:

[7] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon/\kappa))\right]$$

- This results in an exponential improvement on the precision parameter

[7] A. Childs, R. Kothari, **RDS**, SIAM J. Comp. **46**, 1920 (2017).

This talk: two quantum algorithms for the QSLP

- I will present two quantum algorithms for the QLSP that improve previous results in different ways:

[7] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon/\kappa))\right]$$

- This results in an exponential improvement on the precision parameter
- It can be improved using a version of VTAA to:

[7] There exists a quantum algorithm that solves the QLSP with complexity

$$\tilde{O}\left[\kappa T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon))\right]$$

[7] A. Childs, R. Kothari, **RDS**, SIAM J. Comp. **46**, 1920 (2017).

This talk: two quantum algorithms for the QSLP

Why these improvements are important?

- The previous result allowed us to prove a polynomial quantum speedup for hitting time estimation in terms of the spectral gap of a Markov chain and precision (A. Chowdhury, R.D. Somma, QIC 17, 0041 (2017)).

- Having a small complexity dependence on precision is important for, e.g., computing expectation values of observables at the quantum metrology limit.

This talk: two quantum algorithms for the QSLP

- I will present two quantum algorithms for the QLSP that improve previous results in different ways:

[8] There exists a quantum algorithm that solves the QLSP by evolving with Hamiltonians that are linear combinations of (products of) A, the projector in the initial state, and Pauli matrices. The overall evolution time is $\tilde{O}(\kappa/\epsilon)$

[8] Y. Subasi, **RDS**, D. Orsucci, arXiv:1805.10549 (2018).

This talk: two quantum algorithms for the QSLP

- I will present two quantum algorithms for the QLSP that improve previous results in different ways:

[8] There exists a quantum algorithm that solves the QLSP by evolving with Hamiltonians that are linear combinations of (products of) A, the projector in the initial state, and Pauli matrices. The overall evolution time is $\tilde{O}(\kappa/\epsilon)$

Using Hamiltonian simulation, this transfers to complexity $\tilde{O}(\kappa T_b/\epsilon + C_A(\kappa/\epsilon, \epsilon))$

[8] Y. Subasi, **RDS**, D. Orsucci, arXiv:1805.10549 (2018).

This talk: two quantum algorithms for the QSLP

- I will present two quantum algorithms for the QLSP that improve previous results in different ways:

[8] There exists a quantum algorithm that solves the QLSP by evolving with Hamiltonians that are linear combinations of (products of) A, the projector in the initial state, and Pauli matrices. The overall evolution time is $\tilde{O}(\kappa/\epsilon)$

Using Hamiltonian simulation, this transfers to complexity $\tilde{O}(\kappa T_b/\epsilon + C_A(\kappa/\epsilon, \epsilon))$

- The method is very different and based on adiabatic evolutions. It does not require of complicated subroutines such as phase estimation and variable time amplitude amplification, therefore reducing the number of ancillary qubits substantially.

[8] Y. Subasi, **RDS**, D. Orsucci, arXiv:1805.10549 (2018).

This talk: two quantum algorithms for the QSLP

Why this improvement is important?

- Phase estimation and VTAA require several ancillary qubits (beyond those needed for Hamiltonian simulation)
- Within two weeks of posting our result, a group implemented our algorithm in NMR, claiming that it is the largest simulated instance so far (8x8) [9]

[9] J. Wen, et.al., arXiv:1806.0329 (2018)

First algorithm: A Fourier approach for solving the QSLP

First algorithm: A Fourier approach for solving the QSLP

Recall that the main goal is achieved by implementing $1/A$ to the initial state $|b\rangle$

First algorithm: A Fourier approach for solving the QSLP

Recall that the main goal is achieved by implementing $1/A$ to the initial state $|b\rangle$

- $1/A$ is not unitary and we need to find a unitary implementation for it. We then go through a sequence of approximations:

$$\frac{1}{x} = \int_0^\infty dy\; xye^{-(xy)^2/2}\;,\;\; xye^{-(xy)^2/2} = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dz\; ze^{-z^2/2} e^{-ixyz}$$

First algorithm: A Fourier approach for solving the QSLP

Recall that the main goal is achieved by implementing $1/A$ to the initial state $|b\rangle$

- $1/A$ is not unitary and we need to find a unitary implementation for it. We then go through a sequence of approximations:

$$\frac{1}{x} = \int_0^\infty dy\; xye^{-(xy)^2/2}\;,\;\; xye^{-(xy)^2/2} = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dz\; ze^{-z^2/2} e^{-ixyz}$$

$$\frac{1}{A} \approx_\varepsilon \frac{i\delta y\, \delta z}{\sqrt{2\pi}} \sum_{j=0}^{J} \sum_{k=-K}^{K} z_k\, e^{-(z_k)^2/2}\, e^{-iA(y_j z_k)}$$

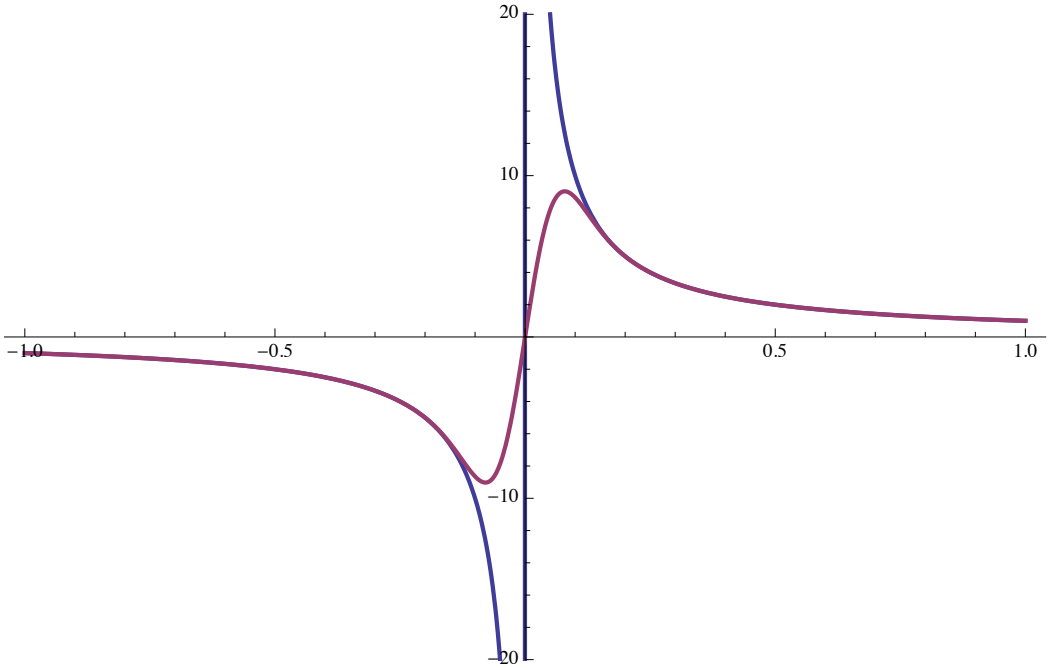
we are getting closer: Linear combination of unitaries

First algorithm: A Fourier approach for solving the QSLP

$$\frac{1}{x} \approx \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J} \Delta y \sum_{k=-K}^{K} \Delta z \; z_k e^{-z_k^2/2} e^{-ixy_j z_k}$$

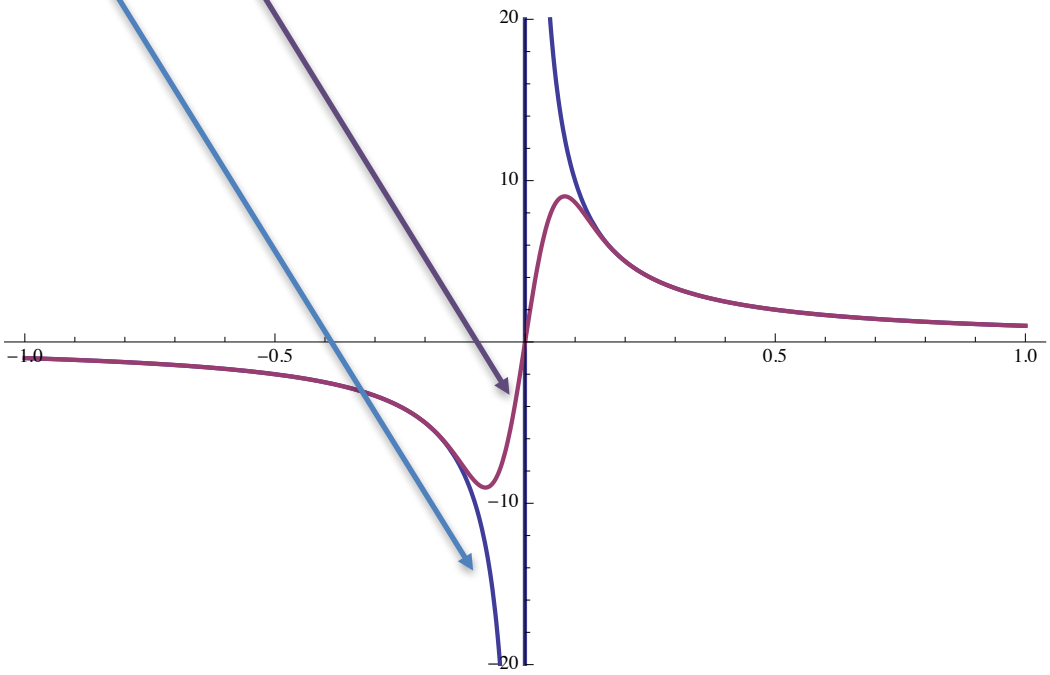
First algorithm: A Fourier approach for solving the QSLP

$$\frac{1}{x} \approx \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J} \Delta y \sum_{k=-K}^{K} \Delta z \; z_k e^{-z_k^2/2} e^{-ixy_j z_k}$$



First algorithm: A Fourier approach for solving the QSLP

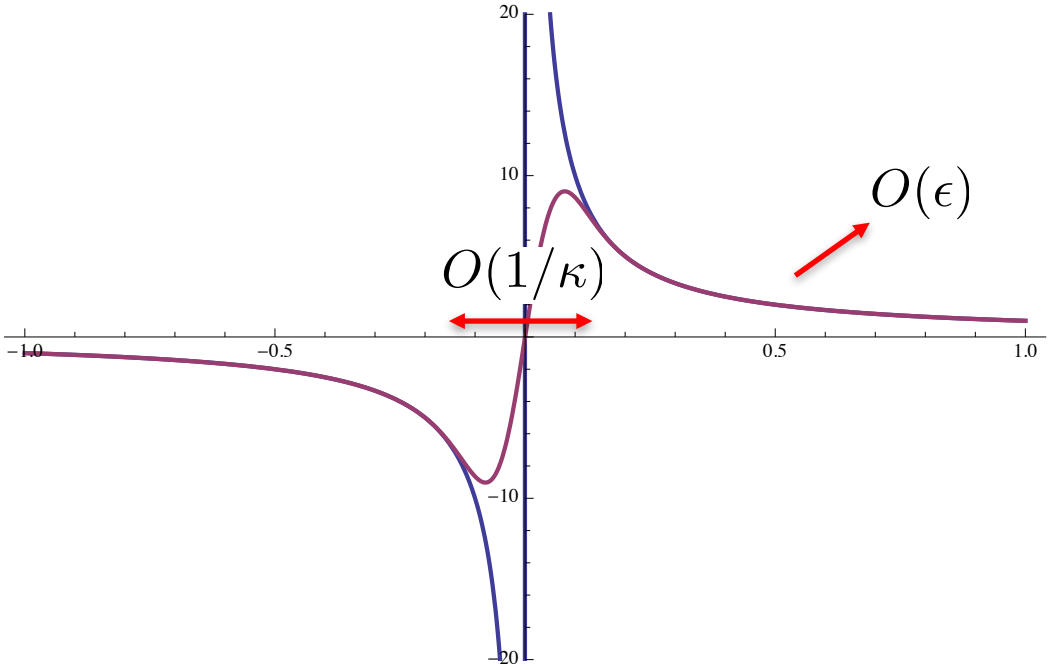
$$\frac{1}{x} \approx \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J} \Delta y \sum_{k=-K}^{K} \Delta z \; z_k e^{-z_k^2/2} e^{-ixy_j z_k}$$



The accuracy of the approximation will determine $J, K, \Delta y, \Delta z$

First algorithm: A Fourier approach for solving the QSLP

$$\frac{1}{x} \approx \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J} \Delta y \sum_{k=-K}^{K} \Delta z \; z_k e^{-z_k^2/2} e^{-ixy_j z_k}$$



$$J = \tilde{O}(\kappa/\epsilon)$$
$$K = \tilde{O}(\kappa)$$
$$\Delta y = \tilde{\Omega}(\epsilon)$$
$$\Delta z = \tilde{\Omega}(1/\kappa)$$

The maximum "evolution time" under A in the approximation of $1/A$ is

$$(J\Delta y)(K\Delta z) = O(\kappa \log(\kappa/\epsilon))$$

First algorithm: A Fourier approach for solving the QSLP

- So far we approximated $1/A$, within the desired accuracy, by a finite linear combination of unitaries. Each unitary corresponds to evolving with A for certain time, and the max evolution time is almost linear in the condition number

First algorithm: A Fourier approach for solving the QSLP

- So far we approximated $1/A$, within the desired accuracy, by a finite linear combination of unitaries. Each unitary corresponds to evolving with A for certain time, and the max evolution time is almost linear in the condition number

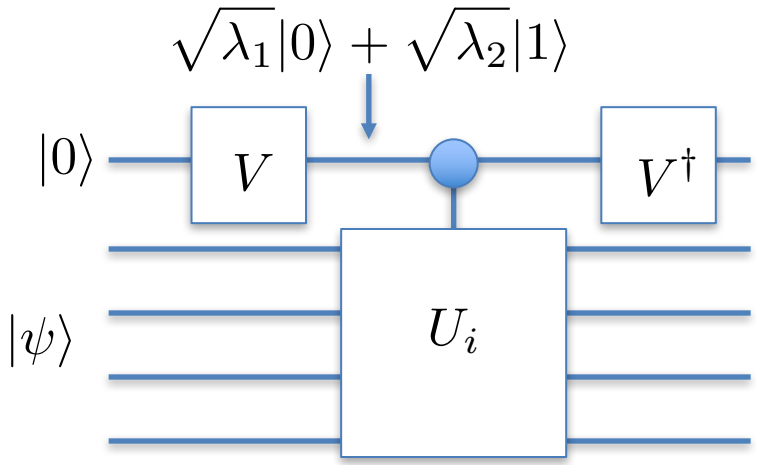
QLSP  Hamiltonian simulation

Implementing a linear combination of unitaries

Suppose we want to implement the operator $\lambda_1 U_1 + \lambda_2 U_2$ to some state $|\psi\rangle$ where $\lambda_i \geq 0$, $\lambda_1 + \lambda_2 = 1$, and U_i unitary

Implementing a linear combination of unitaries

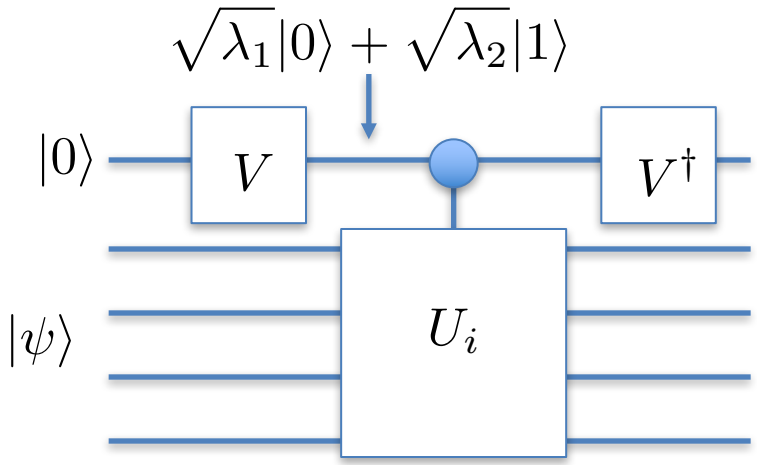
Suppose we want to implement the operator $\lambda_1 U_1 + \lambda_2 U_2$ to some state $|\psi\rangle$ where $\lambda_i \geq 0$, $\lambda_1 + \lambda_2 = 1$, and U_i unitary



$$(\lambda_1 U_1 + \lambda_2 U_2)|\psi\rangle|0\rangle + \sqrt{\lambda_1 \lambda_2}(U_1 - U_2)|\psi\rangle|1\rangle$$

Implementing a linear combination of unitaries

Suppose we want to implement the operator $\lambda_1 U_1 + \lambda_2 U_2$ to some state $|\psi\rangle$ where $\lambda_i \geq 0$, $\lambda_1 + \lambda_2 = 1$, and U_i unitary



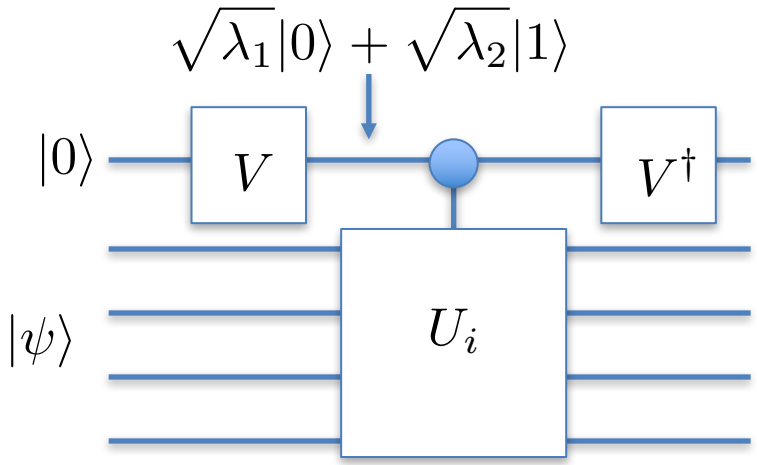
$$(\lambda_1 U_1 + \lambda_2 U_2)|\psi\rangle|0\rangle + \sqrt{\lambda_1\lambda_2}(U_1 - U_2)|\psi\rangle|1\rangle$$

Correct part

Incorrect part (orthogonal)

Implementing a linear combination of unitaries

Suppose we want to implement the operator $\lambda_1 U_1 + \lambda_2 U_2$ to some state $|\psi\rangle$ where $\lambda_i \geq 0$, $\lambda_1 + \lambda_2 = 1$, and U_i unitary



$$(\lambda_1 U_1 + \lambda_2 U_2)|\psi\rangle|0\rangle + \sqrt{\lambda_1\lambda_2}(U_1 - U_2)|\psi\rangle|1\rangle$$

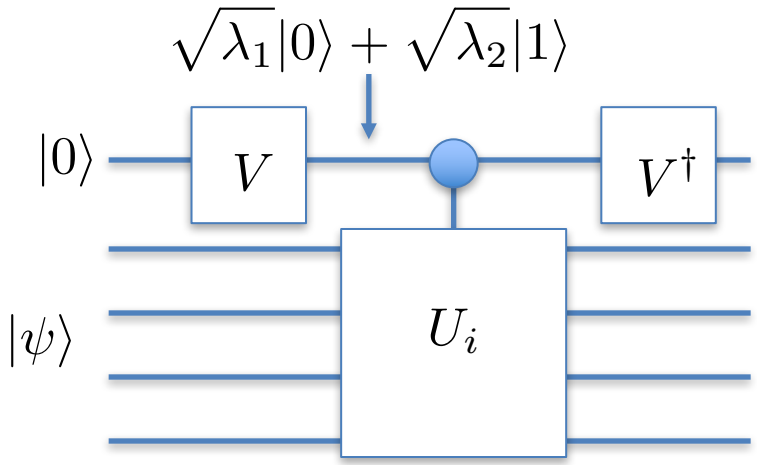
Correct part

Incorrect part (orthogonal)

This idea can be generalized to the case where the goal is to implement $\sum_{i=0}^{M-1} \lambda_i U_i$

Implementing a linear combination of unitaries

Suppose we want to implement the operator $\lambda_1 U_1 + \lambda_2 U_2$ to some state $|\psi\rangle$ where $\lambda_i \geq 0$, $\lambda_1 + \lambda_2 = 1$, and U_i unitary



$$(\lambda_1 U_1 + \lambda_2 U_2)|\psi\rangle|0\rangle + \sqrt{\lambda_1\lambda_2}(U_1 - U_2)|\psi\rangle|1\rangle$$

Correct part

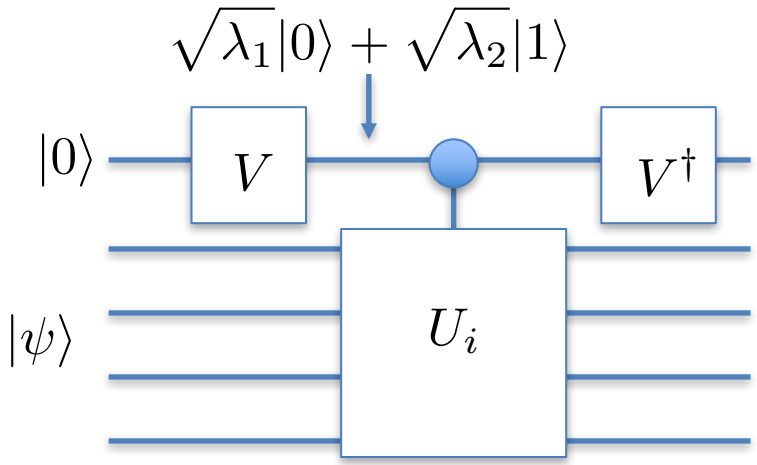
Incorrect part (orthogonal)

This idea can be generalized to the case where the goal is to implement $\sum_{i=0}^{M-1} \lambda_i U_i$

$$\frac{1}{\lambda}\left(\sum_{i=0}^{M-1} \lambda_i U_i\right)|\psi\rangle|0\ldots0\rangle + |\xi^\perp\rangle$$

Implementing a linear combination of unitaries

Suppose we want to implement the operator $\lambda_1 U_1 + \lambda_2 U_2$ to some state $|\psi\rangle$ where $\lambda_i \geq 0$, $\lambda_1 + \lambda_2 = 1$, and U_i unitary



$$(\lambda_1 U_1 + \lambda_2 U_2)|\psi\rangle|0\rangle + \sqrt{\lambda_1\lambda_2}(U_1 - U_2)|\psi\rangle|1\rangle$$

Correct part

Incorrect part (orthogonal)

This idea can be generalized to the case where the goal is to implement $\sum_{i=0}^{M-1} \lambda_i U_i$

$$\Rightarrow \quad \frac{1}{\lambda}\left(\sum_{i=0}^{M-1} \lambda_i U_i\right)|\psi\rangle|0\ldots0\rangle + |\xi^\perp\rangle \quad \Rightarrow$$

Amplitude amplification to obtain the correct part

First algorithm: A Fourier approach for solving the QSLP

- We use the LCU approach to implement the Fourier approximation of $1/A$

First algorithm: A Fourier approach for solving the QSLP

- We use the LCU approach to implement the Fourier approximation of $1/A$
- Note: We assume that the gate complexity of the operation V is small with respect to other complexities
- Adding up all the coefficients in the linear combination of unitaries, we obtain

$$\lambda = \tilde{O}(\kappa)$$

First algorithm: A Fourier approach for solving the QSLP

- We use the LCU approach to implement the Fourier approximation of $1/A$
- Note: We assume that the gate complexity of the operation V is small with respect to other complexities
- Adding up all the coefficients in the linear combination of unitaries, we obtain

$$\lambda = \tilde{O}(\kappa)$$

- This is also the number of amplitude amplification rounds

First algorithm: A Fourier approach for solving the QSLP

- We use the LCU approach to implement the Fourier approximation of 1/A
- Note: We assume that the gate complexity of the operation V is small with respect to other complexities
- Adding up all the coefficients in the linear combination of unitaries, we obtain

$$\lambda = \tilde{O}(\kappa)$$

- This is also the number of amplitude amplification rounds
- Then, the overall complexity of this approach is

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon/\kappa))\right]$$

First algorithm: A Fourier approach for solving the QSLP

- We use the LCU approach to implement the Fourier approximation of 1/A
- Note: We assume that the gate complexity of the operation V is small with respect to other complexities
- Adding up all the coefficients in the linear combination of unitaries, we obtain

$$\lambda = \tilde{O}(\kappa)$$

- This is also the number of amplitude amplification rounds
- Then, the overall complexity of this approach is

$$\tilde{O}\left[\kappa(T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon/\kappa))\right]$$

This is almost quadratic in the condition number. To improve it to almost linear we use a version of VTAA that doesn't ruin the logarithmic scaling in precision

First algorithm: A Fourier approach for solving the QSLP

- VTAA for HHL relies heavily on phase estimation, bringing a prohibitive complexity dependence on precision
- But in our case we only need to distinguish the regions for the eigenvalues with high confidence, so the scaling in precision is logarithmic

First algorithm: A Fourier approach for solving the QSLP

- VTAA for HHL relies heavily on phase estimation, bringing a prohibitive complexity dependence on precision
- But in our case we only need to distinguish the regions for the eigenvalues with high confidence, so the scaling in precision is logarithmic

- The final algorithm is VTAA applied to another algorithm that is built upon a sequence of steps.
- At each step we do the following: i) We determine the region of the eigenvalue with high confidence. ii) We apply 1/A within the necessary precision for that region (replacing the condition number)

First algorithm: A Fourier approach for solving the QSLP

- VTAA for HHL relies heavily on phase estimation, bringing a prohibitive complexity dependence on precision
- But in our case we only need to distinguish the regions for the eigenvalues with high confidence, so the scaling in precision is logarithmic

- The final algorithm is VTAA applied to another algorithm that is built upon a sequence of steps.
- At each step we do the following: i) We determine the region of the eigenvalue with high confidence. ii) We apply 1/A within the necessary precision for that region (replacing the condition number)

- The overall complexity of this approach is

$$\tilde{O}\left[\kappa T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon))\right]$$

First algorithm: A Fourier approach for solving the QSLP

- VTAA for HHL relies heavily on phase estimation, bringing a prohibitive complexity dependence on precision
- But in our case we only need to distinguish the regions for the eigenvalues with high confidence, so the scaling in precision is logarithmic

- The final algorithm is VTAA applied to another algorithm that is built upon a sequence of steps.
- At each step we do the following: i) We determine the region of the eigenvalue with high confidence. ii) We apply 1/A within the necessary precision for that region (replacing the condition number)

- The overall complexity of this approach is

$$\tilde{O}\left[\kappa T_b + C_A(\kappa \log(\kappa/\epsilon, \epsilon))\right]$$

Using the best Hamiltonian simulation methods, this is almost linear in the condition number and polylog in inverse of precision

Second algorithm: An “adiabatic” approach for the QSLP

Second algorithm: An “adiabatic” approach for the QSLP

- The idea here is to prepare the eigenstate of a Hamiltonian by preparing a sequence continuously related eigenstates of a family of Hamiltonians

Second algorithm: An “adiabatic” approach for the QSLP

- The idea here is to prepare the eigenstate of a Hamiltonian by preparing a sequence continuously related eigenstates of a family of Hamiltonians
- We want the eigenstate to be the desired quantum state (after tracing out ancillary systems)

Second algorithm: An “adiabatic” approach for the QSLP

- The idea here is to prepare the eigenstate of a Hamiltonian by preparing a sequence continuously related eigenstates of a family of Hamiltonians
- We want the eigenstate to be the desired quantum state (after tracing out ancillary systems)

$$P_b^{\perp} A.\vec{x} = P_b^{\perp}\vec{b} = 0$$

$$B^{\dagger}B|x\rangle = 0\ ,\ B = P_b^{\perp}.A$$

Second algorithm: An "adiabatic" approach for the QSLP

- The idea here is to prepare the eigenstate of a Hamiltonian by preparing a sequence continuously related eigenstates of a family of Hamiltonians
- We want the eigenstate to be the desired quantum state (after tracing out ancillary systems)

$$P_b^{\perp} A.\vec{x} = P_b^{\perp}\vec{b} = 0$$

$$\overset{H}{\Rightarrow} \quad B^{\dagger}B|x\rangle = 0\ ,\ B = P_b^{\perp}.A$$

The following properties can be proven:

- The desired state is the unique ground state of H
- The eigenvalue gap is order $1/\kappa^2$

Second algorithm: An "adiabatic" approach for the QSLP

- The idea here is to prepare the eigenstate of a Hamiltonian by preparing a sequence continuously related eigenstates of a family of Hamiltonians
- We want the eigenstate to be the desired quantum state (after tracing out ancillary systems)

$$P_b^{\perp} A.\vec{x} = P_b^{\perp}\vec{b} = 0$$

$$\xrightarrow{H} \quad B^{\dagger}B|x\rangle = 0 \ , \ B = P_b^{\perp}.A$$

The following properties can be proven:

- The desired state is the unique ground state of H
- The eigenvalue gap is order $1/\kappa^2$

- We now seek the family of interpolating Hamiltonians

Second algorithm: An “adiabatic” approach for the QSLP

- We assume for the moment that A>1/ κ

Second algorithm: An “adiabatic” approach for the QSLP

- We assume for the moment that A>1/ κ
- We define the interpolating matrix $A(s) = (1-s)I + sA\ ,\ 0 \leq s \leq 1$

Second algorithm: An “adiabatic” approach for the QSLP

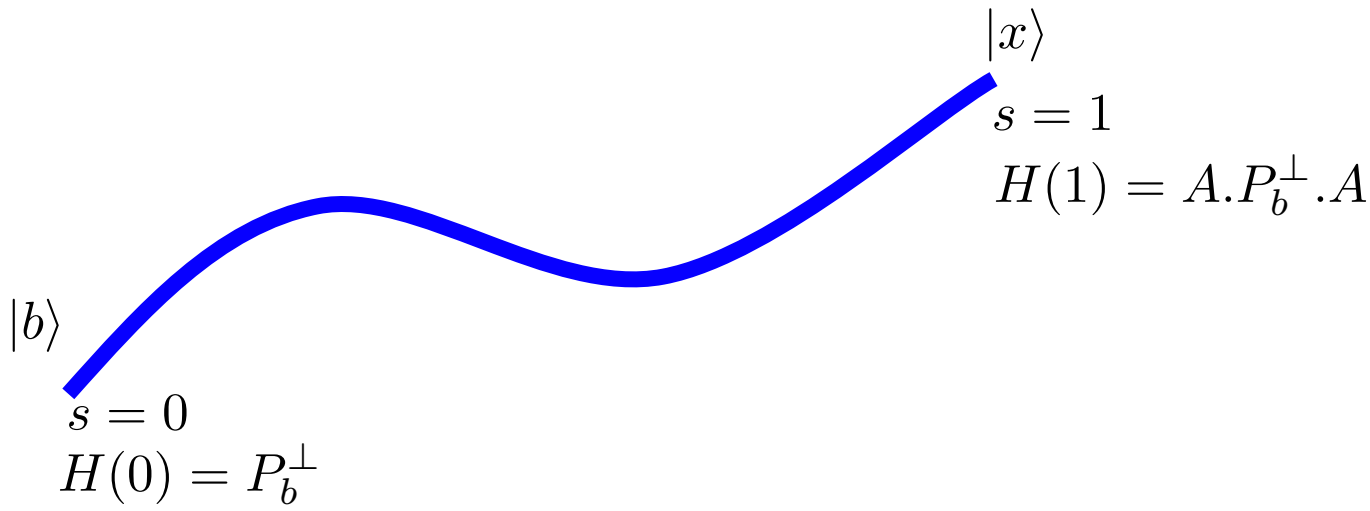
- We assume for the moment that A>1/ κ
- We define the interpolating matrix $A(s) = (1-s)I + sA\ ,\ 0 \leq s \leq 1$
- Similarly, we define $H(s) = B^{\dagger}(s)B(s)\ ,\ B(s) = P_b^{\perp}A(s)$

Second algorithm: An “adiabatic” approach for the QSLP

- We assume for the moment that A>1/ κ
- We define the interpolating matrix $A(s) = (1-s)I + sA\ ,\ 0 \leq s \leq 1$
- Similarly, we define $H(s) = B^{\dagger}(s)B(s)\ ,\ B(s) = P_b^{\perp}A(s)$
- This is like solving an increasingly difficult system of linear equations!

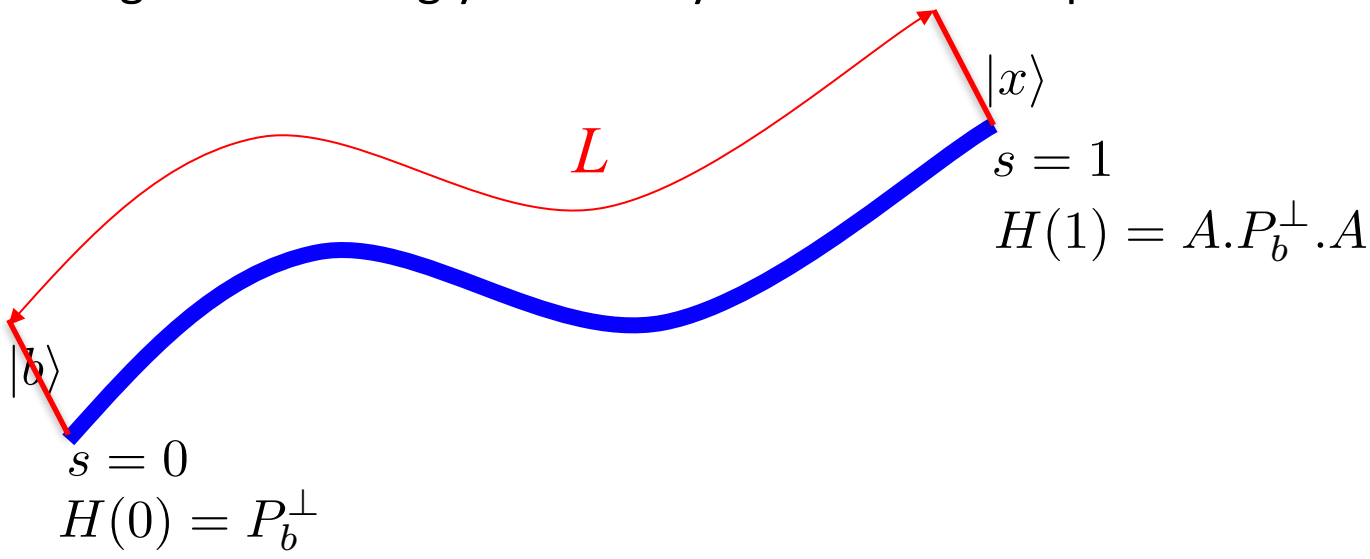
Second algorithm: An "adiabatic" approach for the QSLP

- We assume for the moment that A>1/ κ
- We define the interpolating matrix $A(s) = (1-s)I + sA\ ,\ 0 \leq s \leq 1$
- Similarly, we define $H(s) = B^{\dagger}(s)B(s)\ ,\ B(s) = P_b^{\perp}A(s)$
- This is like solving an increasingly difficult system of linear equations!



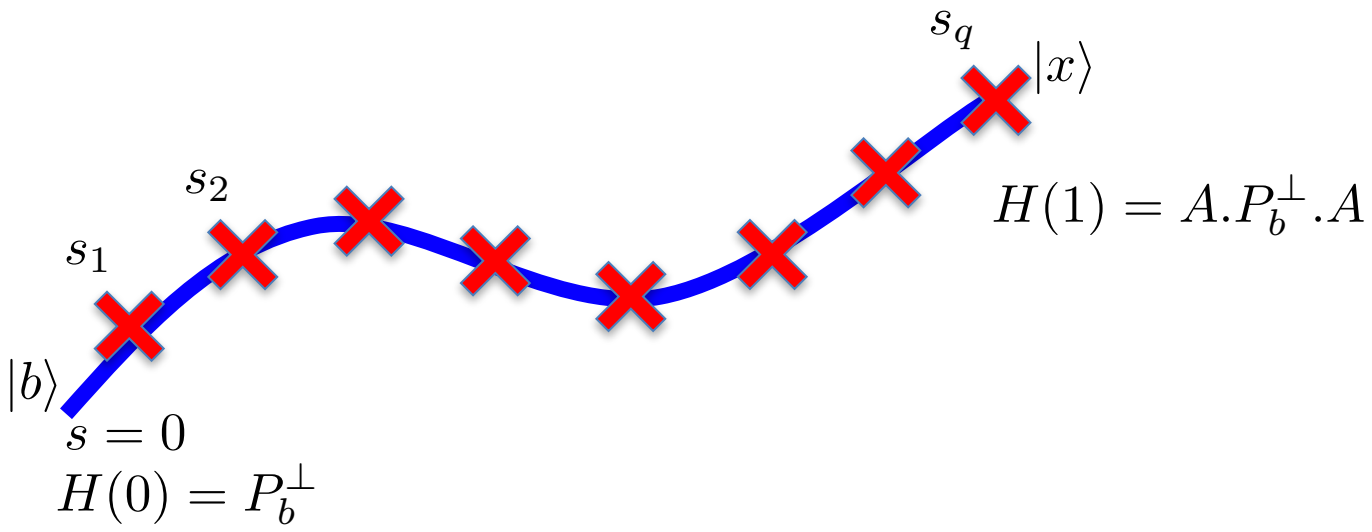
Second algorithm: An "adiabatic" approach for the QSLP

- We assume for the moment that A>1/ κ
- We define the interpolating matrix $A(s) = (1-s)I + sA\ ,\ 0 \le s \le 1$
- Similarly, we define $H(s) = B^{\dagger}(s)B(s)\ ,\ B(s) = P_b^{\perp}A(s)$
- This is like solving an increasingly difficult system of linear equations!



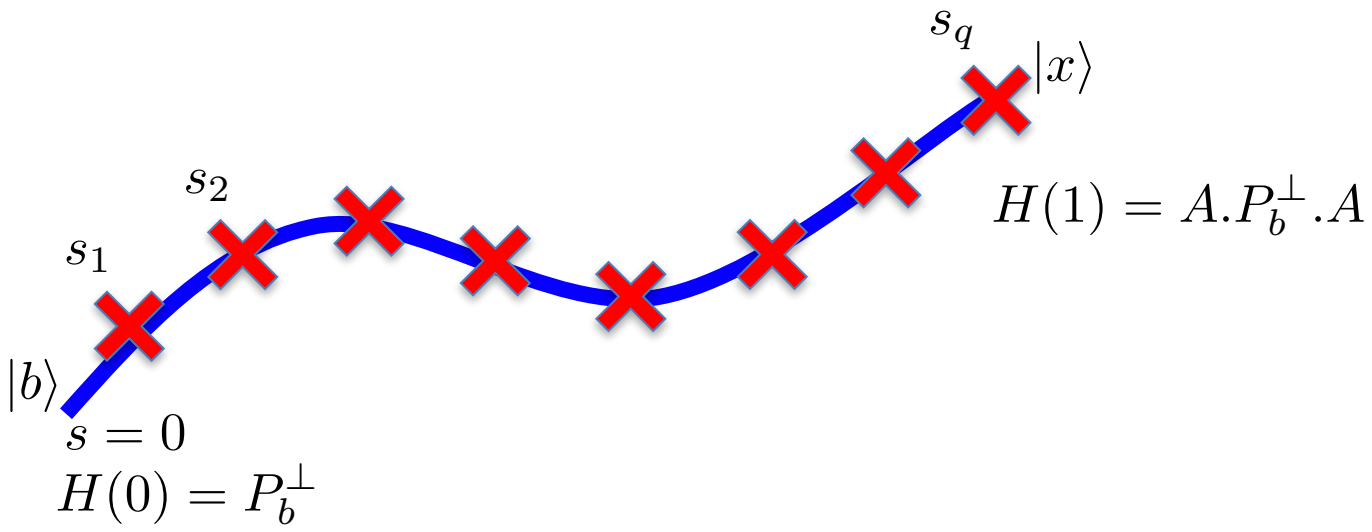
- The minimum eigenvalue gap is order 1/ κ^2 and the length of the path L is log(κ)

The randomization method to prepare eigenstates []



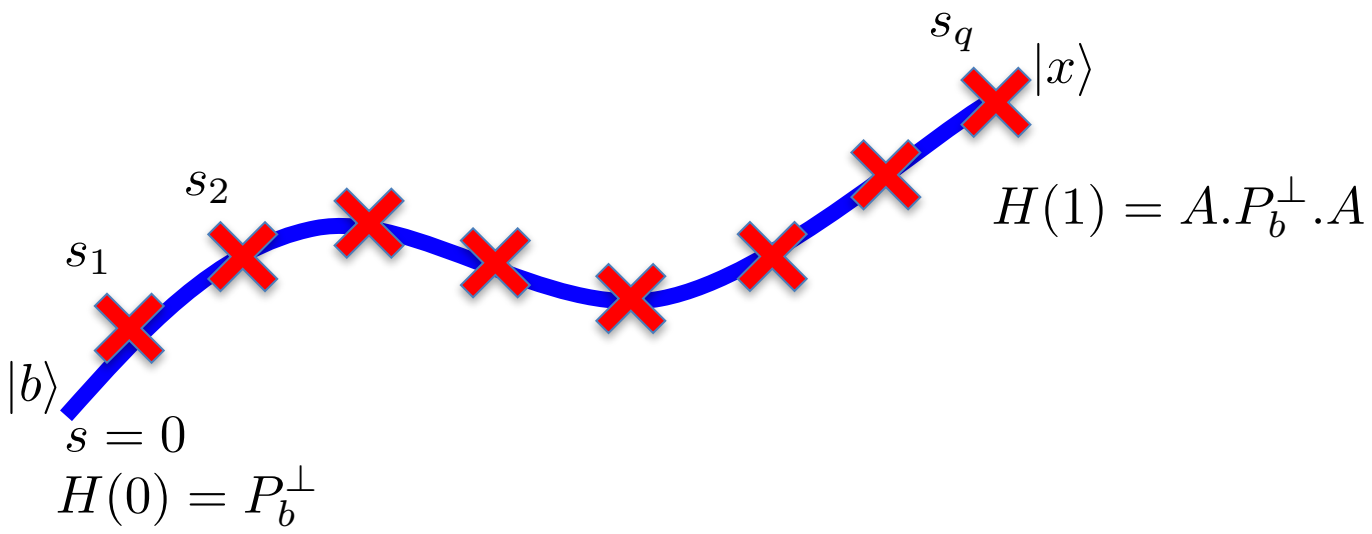
- By performing a sequence of projective measurements at sufficiently close points, we can prepare the related eigenstates with high probability

The randomization method to prepare eigenstates []



- By performing a sequence of projective measurements at sufficiently close points, we can prepare the related eigenstates with high probability
- Each measurement can be simulated by evolving with the corresponding Hamiltonian for random time. This reduces coherences between eigenstates

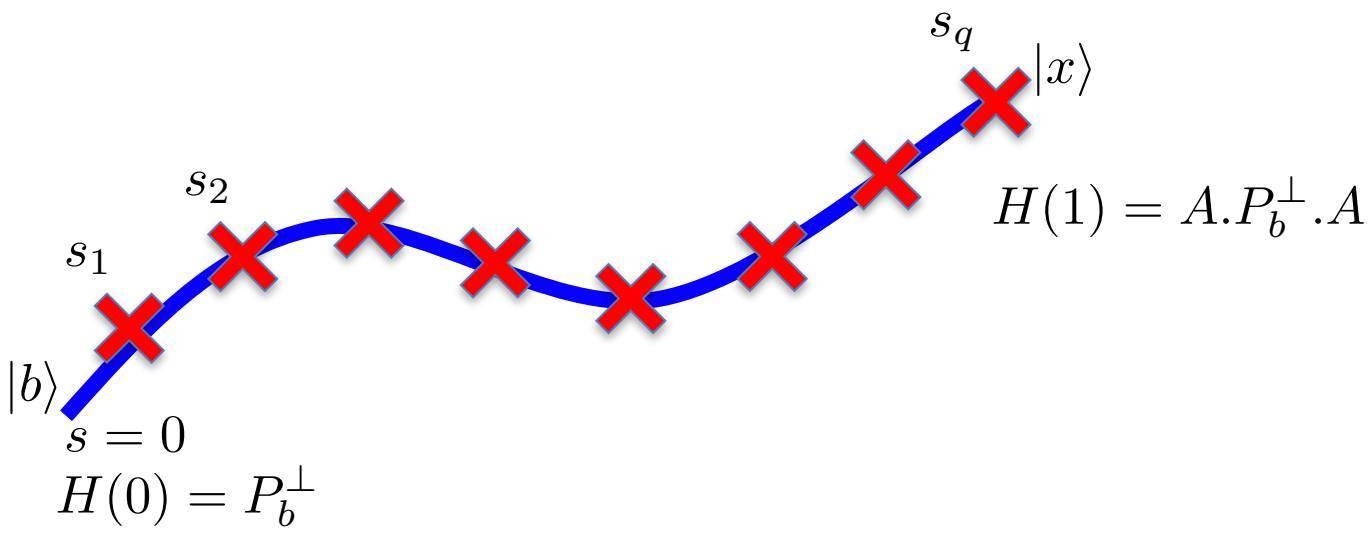
The randomization method to prepare eigenstates []



- By performing a sequence of projective measurements at sufficiently close points, we can prepare the related eigenstates with high probability
- Each measurement can be simulated by evolving with the corresponding Hamiltonian for random time. This reduces coherences between eigenstates
- The expected evolution time with the Hamiltonians in the randomization method satisfies

$$T_{RM} = O\left(\frac{L^2}{\epsilon\Delta}\right)$$

The randomization method to prepare eigenstates []

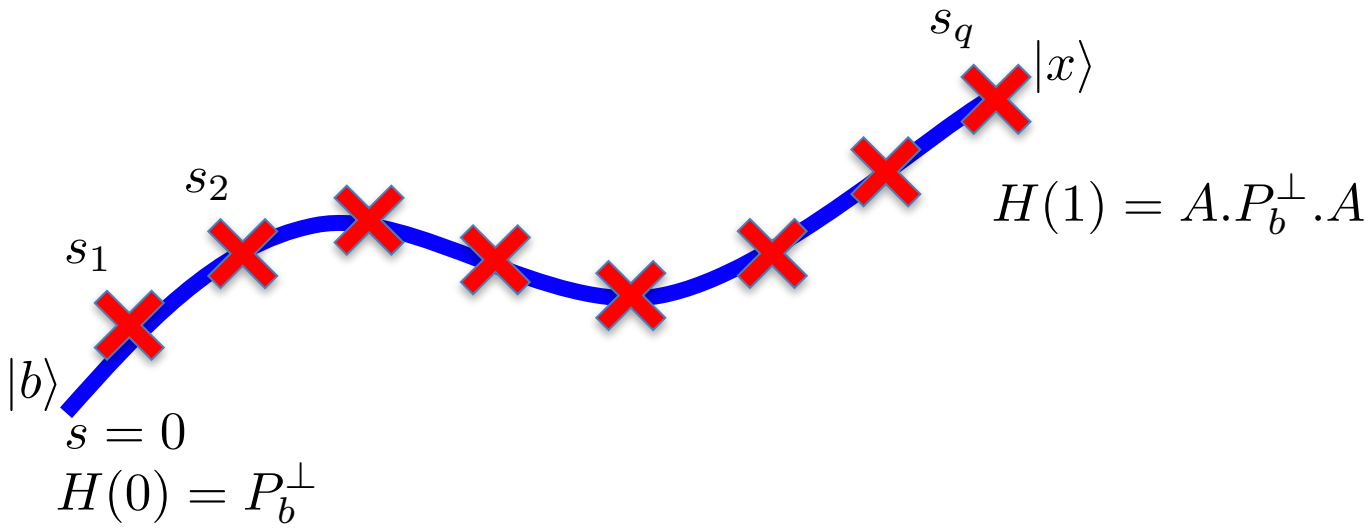


- By performing a sequence of projective measurements at sufficiently close points, we can prepare the related eigenstates with high probability
- Each measurement can be simulated by evolving with the corresponding Hamiltonian for random time. This reduces coherences between eigenstates
- The expected evolution time with the Hamiltonians in the randomization method satisfies

$$T_{RM} = O\left(\frac{L^2}{\epsilon\Delta}\right)$$

L is the path length
Δ is the min gap
ϵ is the error (trace norm)

The randomization method for the QLSP []



- By performing a sequence of projective measurements at sufficiently close points, we can prepare the related eigenstates with high probability
- Each measurement can be simulated by evolving with the corresponding Hamiltonian for random time. This reduces coherences between eigenstates
- The expected evolution time with the Hamiltonians in the randomization method satisfies

$$T_{RM} = O\left(\frac{\kappa^2 \log^2(\kappa)}{\epsilon}\right)$$

The randomization method for the QLSP []

- The strong dependence of the evolution time with the spectral gap suggests one to consider other Hamiltonians that have the same eigenstate but a larger eigenvalue gap

The randomization method for the QLSP []

- The strong dependence of the evolution time with the spectral gap suggests one to consider other Hamiltonians that have the same eigenstate but a larger eigenvalue gap

- For this problem, spectral gap amplification [10] is useful:

$$H(s) \to H'(s) = B^\dagger(s) \otimes \sigma^- + B(s) \otimes \sigma^+ = \begin{pmatrix} 0 & B(s) \\ B^\dagger(s) & 0 \end{pmatrix}$$

[10] R.D. Somma and S. Boixo, SIAM J. Comp. 593 (2013)

The randomization method for the QLSP []

- The strong dependence of the evolution time with the spectral gap suggests one to consider other Hamiltonians that have the same eigenstate but a larger eigenvalue gap

- For this problem, spectral gap amplification [10] is useful:

$$H(s) \to H'(s) = B^\dagger(s) \otimes \sigma^- + B(s) \otimes \sigma^+ = \begin{pmatrix} 0 & B(s) \\ B^\dagger(s) & 0 \end{pmatrix}$$

- Some results:

Let $|x(s)\rangle$ be the eigenstate of 0-eigenvalue of $H(s)$.
Then, $|x(s)\rangle|1\rangle$ is an eigenstate of 0-eigenvalue of $H'(s)$.
This eigenstate is separated from others by an eigenvalue gap $\sqrt{\Delta(s)}$

[10] R.D. Somma and S. Boixo, SIAM J. Comp. 593 (2013)

The randomization method for the QLSP []

- The strong dependence of the evolution time with the spectral gap suggests one to consider other Hamiltonians that have the same eigenstate but a larger eigenvalue gap

- For this problem, spectral gap amplification is useful:

$$H(s) \rightarrow H'(s) = B^{\dagger}(s) \otimes \sigma^{-} + B(s) \otimes \sigma^{+} = \begin{pmatrix} 0 & B(s) \\ B^{\dagger}(s) & 0 \end{pmatrix}$$

- Some results:

Let $|x(s)\rangle$ be the eigenstate of 0-eigenvalue of $H(s)$.
Then, $|x(s)\rangle|1\rangle$ is an eigenstate of 0-eigenvalue of $H'(s)$.
This eigenstate is separated from others by an eigenvalue gap $\sqrt{\Delta(s)}$

- Note that the path length did not change. The only change for the RM is the use of a different Hamiltonian.

The randomization method for the QLSP []

- Using the randomization method with the new Hamiltonian, the expected evolution time is

$$T_{RM} = O\left(\frac{\kappa \log^2(\kappa)}{\epsilon}\right)$$

The randomization method for the QLSP []

- Using the randomization method with the new Hamiltonian, the expected evolution time is

$$T_{RM} = O\left(\frac{\kappa \log^2(\kappa)}{\epsilon}\right)$$

- The case of non-positive matrix A can be analyzed similarly using

$$A(s) = (1-s)(\sigma_z^{anc} \otimes I) + s(\sigma_x^{anc} \otimes A)$$

The randomization method, the QLSP, and the gate model

For $A > 0$, the Hamiltonian is $H'(s) = (I - |b\rangle\langle b|)((1-s)I + sA)\sigma^- + H.c.$

The randomization method, the QLSP, and the gate model

For $A > 0$, the Hamiltonian is $H'(s) = (I - |b\rangle\langle b|)((1-s)I + sA)\sigma^- + H.c.$

If $|b\rangle$ is sparse and A is sparse, then $H'(s)$ is also sparse

The randomization method, the QLSP, and the gate model

For $A > 0$, the Hamiltonian is $H'(s) = (I - |b\rangle\langle b|)((1-s)I + sA)\sigma^- + H.c.$

If $|b\rangle$ is sparse and A is sparse, then $H'(s)$ is also sparse

We can use a Hamiltonian simulation method to build a quantum circuit that simulates the evolution. The quantum circuit will use queries.

- The complexity in terms of queries for $|b\rangle\langle b|$ is $\tilde{O}(\kappa T_b/\epsilon)$
- The complexity in terms of queries for A is almost order $C_A(\kappa/\epsilon, \epsilon)$

The randomization method, the QLSP, and the gate model

For $A > 0$, the Hamiltonian is $H'(s) = (I - |b\rangle\langle b|)((1-s)I + sA)\sigma^- + H.c.$

If $|b\rangle$ is sparse and A is sparse, then $H'(s)$ is also sparse

We can use a Hamiltonian simulation method to build a quantum circuit that simulates the evolution. The quantum circuit will use queries.

- The complexity in terms of queries for $|b\rangle\langle b|$ is $\tilde{O}(\kappa T_b/\epsilon)$
- The complexity in terms of queries for A is almost order $C_A(\kappa/\epsilon, \epsilon)$
- The scaling in the precision parameter can be done polyligarithmic by using faster methods for eigenpath traversal [11]

[11] S. Boixo, E. Knill, and R.D. Somma, arXiv:1005.3034 (2010)

Some conclusions and observations

- Quantum computing seems promising. Several quantum algorithms for problems in linear algebra with significant speedups exist

- I presented quantum algorithms to solve the quantum linear systems problem. The techniques can be generalized to apply other operators (other than the inverse of a matrix) to quantum states.

- The advantages of the first algorithm are in that the complexity dependence on precision is only polylogarithmic, exponentially improving previous algorithms for this problem

- The advantages of the second algorithm are in that it doesn't require many ancillary qubits and the problem reduces to a simple Hamiltonian simulation problem

- It would be important to understand the applicability of this algorithm to scientific problems beyond the ones I mentioned. How important are the problems and algorithms?