Quantum numerical linear algebra

Lin Lin

Department of Mathematics, UC Berkeley
Lawrence Berkeley National Laboratory

Joint work with Dong An, Yulong Dong, Yu Tong (Berkeley)

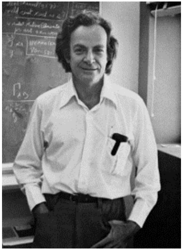
Distinguished Seminar Series in
Computational Science and Engineering,

MIT, October 2020

Solve nature with nature: Where are we today after four decades?

Nature evolves quantum mechanics in real time.

Use nature to solve quantum mechanical problems.



"... if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."'

– Richard P. Feynman (1981) 1st Conference on Physics and Computation, MIT

What does nature do

- Evolution of the Schrödinger equation

$$\mathrm{i}\frac{\partial}{\partial t}\left|\psi(t)\right\rangle = H\left|\psi(t)\right\rangle$$

- Solution

$$\left|\psi(t)\right\rangle = e^{-\mathrm{i}Ht}\left|\psi(0)\right\rangle := U(t)\left|\psi(0)\right\rangle .$$

- $U(t)$: unitary matrix.

Shor's algorithm for prime factorization

- $n = p \cdot q$ (p, q are prime numbers)
- Classical algorithm with best asymptotic complexity (super-polynomial): General number field sieve
  $\mathcal{O}\left(\exp\left[c(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}\right]\right)$
- (Shor, 1994) Quantum algorithm achieves polynomial complexity
  $\mathcal{O}\left((\log n)^2(\log\log n)(\log\log\log n)\right)$
- "First wave" of interests on quantum computing

MIT and quantum computation

Chuang, Farhi, Goldstone, Harrow, Lloyd, Shapiro, Shor..

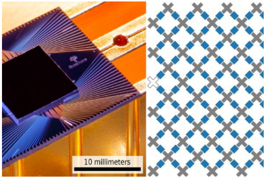
Quantum supremacy

- John Preskill coined the term “Quantum supremacy” in 2012.
- Describe the point *where quantum computers can do things that classical computers cannot, regardless of whether those tasks are useful.*
- It was not clear whether this “moderate” goal can be reached..

  Is controlling large-scale quantum systems merely really, really hard, or is it ridiculously hard? – John Preskill (2012)
- Noisy Intermediate-Scale Quantum (NISQ) technology.

Quantum supremacy

- Quantum supremacy has been reached! (It is merely really, really hard) (Martinis et al, Nature, 2019) Google's 54-qubit machine *Sycamore*

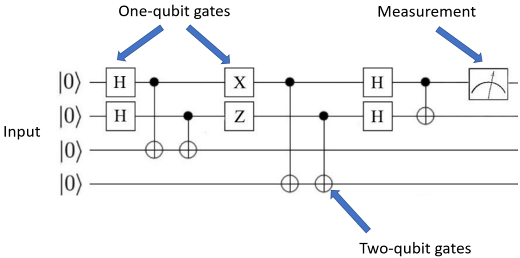


- Finding the probability of bit-strings from a random quantum circuit. Exponentially hard for classical computer w.r.t. n.
- When quantum computer can do something useful, it is called "Quantum advantage".
- We are in the "second wave" of interests on quantum computing.

What is a quantum computer (mathematically)

- $|\psi\rangle \in \mathbb{C}^N \cong (\mathbb{C}^2)^{\otimes n}, \quad N = 2^n$. n : number of qubits.
- $U \in \mathbb{C}^{N\times N}$ is unitary. $U\,|\psi\rangle$ is efficient to apply.
- $U_N \cdots U_1\,|\psi\rangle$, and then measure one or a few qubits (with output $0/1$).
- Quantum computer: cost $\mathrm{poly}(n)$, potential exponential speedup.

Quantum circuit: “graphical” linear algebra



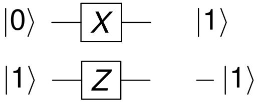
Quantum circuit: “graphical” linear algebra

- State vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Pauli matrices

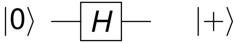
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



Quantum circuit: "graphical" linear algebra

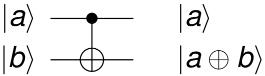
- Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) := |+\rangle$$



- CNOT gate

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



- See more in the classical textbook (Nielsen-Chuang, 2000), or simply take a course at MIT..

Quantum numerical linear algebra

- Solving linear systems, eigenvalue problems, matrix exponentials, least square problems, singular value decompositions etc on a quantum computer.
- Many interesting, exciting progresses in the past few years.
- Reasonable way towards “quantum advantage”.
- Related to “Quantum machine learning”.
- Solving linear equations

$$Ax = b$$

- Quantum linear system problem (QLSP)

$$A|x\rangle \propto |b\rangle$$

Our recent works on quantum numerical linear algebra

Near-optimal QLSP solver (adiabatic computing)	(An-**L.**, 1909.05500)
Near-optimal QLSP solver (eigenstate filtering)	(**L.**-Tong, 1910.14596)
Near-optimal ground energy solver	(**L.**-Tong, 2002.12508)
Quantum signal process phase factor	(Dong-Meng-Whaley-**L.**, 2002.11649)
Policy gradient method and QAOA	(Yao-Bukov-**L.**, 2002.01068)
Proposal of quantum LINPACK benchmark	(Dong-**L.**, 2006.04010)
Fast inversion and preconditioned linear solver	(Tong-An-Wiebe-**L.**, 2008.13295)

A concrete, toy example

$$A = \frac{1}{4}X + \frac{3}{4}I = \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}, \quad |b\rangle = |0\rangle .$$

- X, I are unitaries. A is a linear combination of unitaries (LCU), and is itself non-unitary. $\kappa(A) = 2$.
- Idea: extend 1-qubit non-unitary matrix to a 2-qubit unitary matrix

$$U_A = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}$$

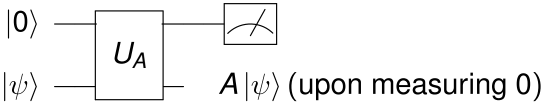
- Block-encoding

A concrete, toy example

- An example of block-encoding. Unitary. Use 1 ancilla qubit.

$$U_A = \begin{pmatrix} \boxed{\begin{matrix} 0.750 & 0.250 \\ 0.250 & 0.750 \end{matrix}} & \begin{matrix} 0.433 & -0.433 \\ -0.433 & 0.433 \end{matrix} \\ \begin{matrix} 0.433 & -0.433 \\ -0.433 & 0.433 \end{matrix} & \begin{matrix} 0.250 & 0.750 \\ 0.750 & 0.250 \end{matrix} \end{pmatrix}$$

- U_A should be viewed as a mapping on $(\mathbb{C}^2)^{\otimes 2}$.
- $(\langle 0| \otimes I) U_A (|0\rangle \otimes I) = A$.



- U_A is our oracle.

A concrete, toy example

- Inverse

$$A^{-1} = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 1.5 \end{pmatrix}$$

Note $\|A^{-1}\| = 2 > 1$, no hope to have

$$U_{A^{-1}} = \begin{pmatrix} A^{-1} & \cdot \\ \cdot & \cdot \end{pmatrix}$$

- How about (with $\alpha > 1$)

$$U_{A^{-1}} \approx \begin{pmatrix} A^{-1}/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix}$$

- Construct $U_{A^{-1}}$ using U_A, $U_A^\dagger$, and simple quantum gates (in this case $U_A = U_A^\dagger$).

Such an $U_{A^{-1}}$ exists

$$U_{A^{-1}} = \begin{pmatrix} 0.075 & -0.025 & 0.0 & 0.0 & 0.271j & 0.728j & -0.442j & 0.442j \\ -0.025 & 0.075 & 0.0 & 0.0 & 0.728j & 0.271j & 0.442j & -0.442j \\ 0.0 & 0.0 & 0.075 & -0.025 & -0.442j & 0.442j & -0.271j & -0.728j \\ 0.0 & 0.0 & -0.025 & 0.075 & 0.442j & -0.442j & -0.728j & -0.271j \\ 0.271j & 0.728j & -0.442j & 0.442j & 0.075 & -0.025 & 0.0 & 0.0 \\ 0.728j & 0.271j & 0.442j & -0.442j & -0.025 & 0.075 & 0.0 & 0.0 \\ -0.442j & 0.442j & -0.271j & -0.728j & 0.0 & 0.0 & 0.075 & -0.025 \\ 0.442j & -0.442j & -0.728j & -0.271j & 0.0 & 0.0 & -0.025 & 0.075 \end{pmatrix}$$

- We find

$$A^{-1}/\alpha = \begin{pmatrix} 0.075 & -0.025 \\ -0.025 & 0.075 \end{pmatrix}, \quad \alpha = 20.$$

- Use 2 ancilla qubits.

Procedure to construct $U_{A^{-1}} |b\rangle$

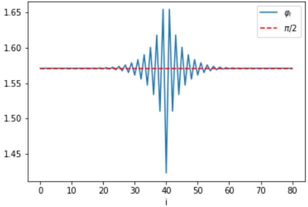
Does not look like any classical direct or iterative algorithm.

- Start from $|0\rangle |0\rangle |b\rangle \equiv (b_1, b_2, 0, 0, 0, 0, 0, 0)^\top$
- Apply $H \otimes I \otimes I$ and obtain $|+\rangle |0\rangle |b\rangle$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- For $i = 0, \ldots, 2d - 1$[1]
  Apply $U_{\varphi_i} \otimes I$, where $U_\varphi = e^{\mathrm{i}\varphi Z} \otimes |0\rangle \langle 0| + e^{-\mathrm{i}\varphi Z} \otimes |1\rangle \langle 1|$.
  Apply $I \otimes U_A$
- Apply $U_{\varphi_{2d}} \otimes I$
- Apply $H \otimes I \otimes I$
- (Optionally, multiply a global phase factor $(-1)^d$)
- Measure the ancilla qubits, i.e. $(\langle 0^2| \otimes I) |\psi\rangle \approx A^{-1}/\alpha |b\rangle$.

[1] This is a simplified procedure using that U_A is Hermitian, $A \succ 0$;$\{\varphi_i\}_{i=0}^{2d}$ are called phase factors.

Accuracy

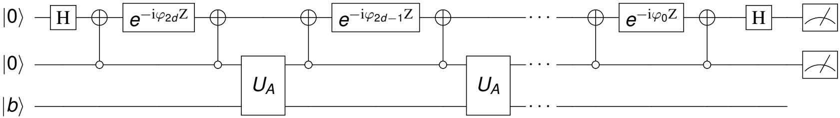
Take $d = 80$, plot the phase factors



- Decay of the phase factor away from the center
- Error for approximating A^{-1}/α

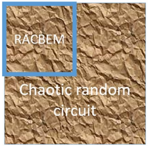
$$\begin{pmatrix} -2.046 \times 10^{-11} & 2.532 \times 10^{-11} \\ 2.532 \times 10^{-11} & -2.046 \times 10^{-11} \end{pmatrix}$$

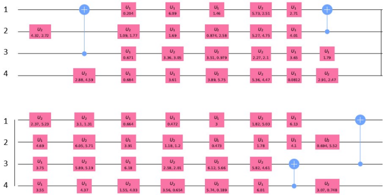
Simplifying the presentation using a quantum circuit

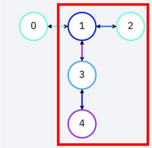


- The same circuit works for arbitrarily large matrix.
- A special case of quantum signal processing (Low-Chuang, 2017) and quantum singular value transformation circuit (QSVT) (Gilyén-Su-Low-Wiebe, 2019).
- One of the most interesting developments in quantum algorithms in the past decade.
- Polynomial eigenvalue transformation and singular value transformation with a definite parity.

RAndom Circuit Block-Encoded Matrix (RACBEM)







Qubit 1 takes |0⟩
Get a 3-qubit RACBEM

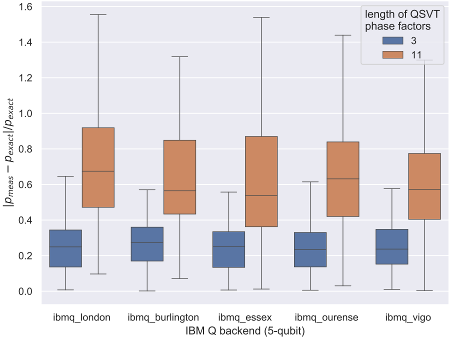
$$A = \begin{pmatrix} -0.066-0.055i & -0.301-0.093i & -0.133-0.118i & 0.046+0.016i & -0.024-0.027i & -0.119-0.056i & 0.393+0.403i & -0.141-0.061i \\ -0.301-0.099i & -0.067-0.041i & 0.042+0.014i & -0.152-0.095i & -0.119-0.058i & -0.025-0.021i & -0.130-0.054i & 0.459+0.333i \\ -0.079-0.188i & -0.105+0.155i & -0.178+0.097i & -0.195-0.104i & 0.396-0.069i & -0.253-0.270i & 0.028+0.124i & -0.096+0.099i \\ -0.109+0.149i & -0.107-0.176i & -0.197-0.104i & -0.154+0.128i & -0.239-0.276i & 0.386-0.128i & -0.097+0.101i & 0.050+0.115i \\ -0.135+0.033i & -0.412+0.299i & 0.037+0.272i & -0.038-0.065i & -0.003+0.045i & 0.053+0.156i & -0.278+0.162i & 0.053-0.070i \\ -0.419+0.294i & -0.118+0.047i & -0.035-0.059i & 0.083+0.263i & 0.051+0.158i & 0.003+0.041i & 0.048-0.065i & -0.248+0.208i \\ -0.071-0.125i & -0.060+0.118i & 0.029+0.367i & -0.316+0.245i & -0.150+0.247i & 0.265-0.009i & 0.038+0.066i & -0.038+0.074i \\ -0.064+0.114i & -0.090-0.114i & -0.319+0.249i & 0.097+0.350i & 0.262+0.001i & -0.113+0.269i & -0.038+0.075i & 0.049+0.057i \end{pmatrix}$$

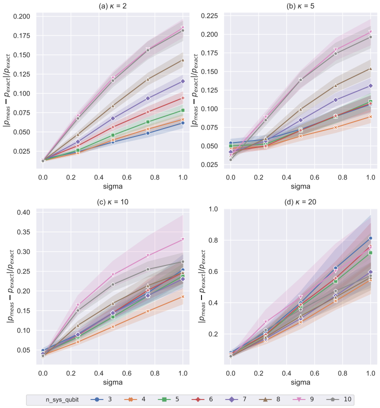
- A very flexible way to construct a non-unitary matrix with respect to any coupling map of the quantum architecture.
- Take upper-left diagonal block: measure one-qubit.
  $A = (\langle 0| \otimes I_n)\, U_A \,(|0\rangle \otimes I_n)$

[1] (Dong, **L.**, arXiv: 2006.04010)

Solving linear system on IBM Q and QVM

Compute $\left\|\mathfrak{H}^{-1}\left|0^n\right\rangle\right\|_2^2$. (sigma: noise level on QVM)





RACBEM

Source Code:
https://github.com/qsppack/racbem

Random circuit block-encoded matrix and
a proposal of quantum LINPACK benchmark

Yulong Dong[1,2] and Lin Lin[3,4*]
[1] Berkeley Center for Quantum Information and Computation, Berkeley, California 94720 USA
[2] Department of Chemistry, University of California, Berkeley, California 94720 USA
[3] Department of Mathematics, University of California, Berkeley, California 94720 USA and
[4] Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

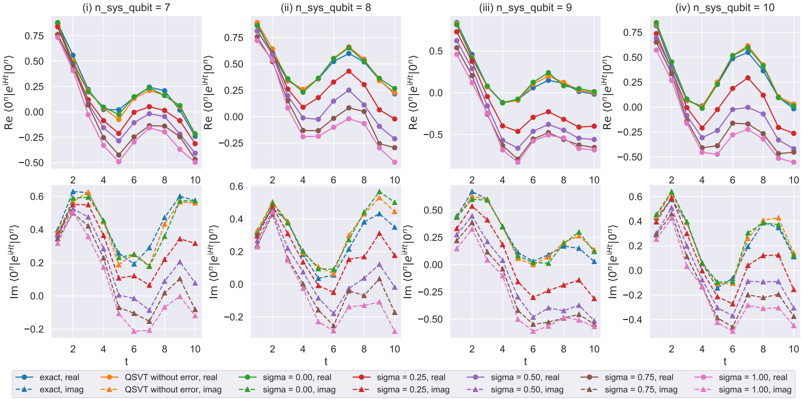
Example:

```
from racbem import *
from qiskit import execute
import numpy as np

n_sys_qubit = 3             # the number of system qubits
n_be_qubit = 1              # the number of block-encoding qubit
n_sig_qubit = 1             # the number of signal qubit
n_tot_qubit = n_sig_qubit+n_be_qubit+n_sys_qubit
n_depth = 15                # the depth of random circuit
prob_one_q_op = 0.5         # the probability of selecting a one-qubit
                            # operation when two_q_op is allowed
n_shots = 2**13             # the number of shots used in measurements

# instances of RACBEM
be = BlockEncoding(n_be_qubit, n_sys_qubit)
qsp = QSPCircuit(n_sig_qubit, n_be_qubit, n_sys_qubit)

# build random circuit with respect to the architecture, including
# 'basis_gates' and 'coupling_map', retrieved from the hardware
be.build_random_circuit(n_depth, basis_gates=basis_gates,
        prob_one_q_op=prob_one_q_op, coupling_map=be_map)

# retrieve the block-encoded matrix
UA = retrieve_unitary_matrix(be.qc)
A = UA[0:2**n_sys_qubit, 0:2**n_sys_qubit]

# build the inverse unitary matrix
be.build_dag()

# build Quantum Signal Processing (QSP) circuit with respect to the
# given set of phase factors 'phi_seq', which determines the functionality
qsp.build_circuit(be.qc, be.qc_dag, phi_seq, realpart=True, measure=True)

# execute the compiled circuit on the quantum backend 'noisy_backend'
job = execute(qsp.qcircuit, backend=noisy_backend, shots=n_shots)

# get the result from the backend
result = job.result()
counts = result.get_counts(qsp.qcircuit)

# success probability of the QSP circuit
prob_meas = np.float(counts['00']) / n_shots
```

Time series (no Trotter)

$$s(t) = \langle \psi | e^{i\mathfrak{H}t} | \psi \rangle$$

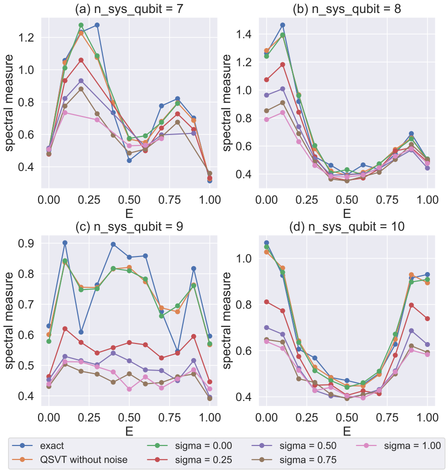
QVM:



Spectral measure

$$s(E) = \langle\psi|\delta(\mathfrak{H} - E)|\psi\rangle \approx \frac{1}{\pi} \operatorname{Im} \langle\psi|(\mathfrak{H} - E - \mathrm{i}\eta)^{-1}|\psi\rangle$$

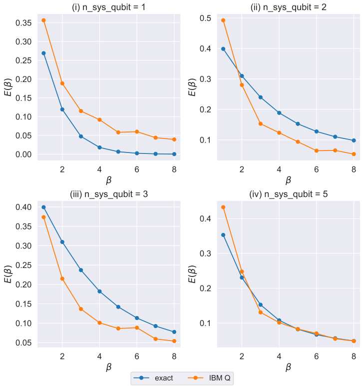
QVM:

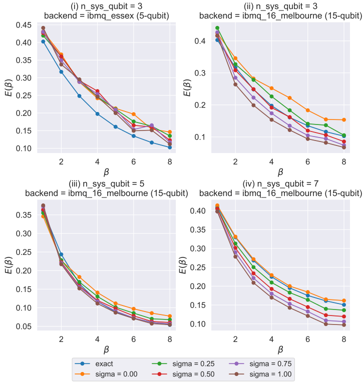


Thermal energy

$$E(\beta) = \frac{\mathrm{Tr}[\mathfrak{H} e^{-\beta \mathfrak{H}}]}{\mathrm{Tr}[e^{-\beta \mathfrak{H}}]}$$

IBM Q (left) and QVM (right)





---

[1] Use the minimally entangled typical thermal state (METTS) algorithm (White, 2009) (Motta et al, 2020)

FAQ (usually from a math audience)

1. *Is quantum linear algebra a real thing?* Yes and no (usually works with complex arithmetic..)

2. *How do you get the matrix / vector into the computer?* Read-in problem, e.g. RACBEM, LCU, block-encoding, Trotter

3. *Which quantity do you measure?* Read-out problem. e.g. some success probabilities and/or access to samples

4. *How do you know your answer is correct?* Verification problem. Performance guarantee versus *a posteriori* verification

Quantum linear system problem (QLSP)

- All vectors must be normalized. $A \in \mathbb{C}^{N\times N}, |b\rangle \in \mathbb{C}^N, N = 2^n$.
- $\||b\rangle\|^2 := \langle b|b\rangle = 1$.
- Solution vector
$$|x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}.$$
- How to get $A, |b\rangle$ into a quantum computer? read-in problem. Oracular assumption.
- Query complexity: the number of oracles used.

Quantum speedup

- κ: condition number of A. ϵ: target accuracy. Proper assumptions on A (e.g. d-sparse) so that oracles cost $\mathrm{poly}(n)$.
- (Harrow-Hassadim-Lloyd, 2009): $\mathcal{O}(\text{poly}(n)\kappa^2/\epsilon)$. Exponential speedup with respect to n.
- (Childs-Kothari-Somma, 2017): Linear combination of unitary (LCU). $\mathcal{O}(\text{poly}(n)\kappa^2\text{poly}(\log(\kappa/\epsilon)))$
- (Low-Chuang, 2017) (Gilyén-Su-Low-Wiebe, 2019): Quantum signal processing (QSP). $\mathcal{O}(\text{poly}(n)\kappa^2\text{poly}(\log(\kappa/\epsilon)))$

Comparison with classical iterative solvers

- Positive definite matrix. Error in A-norm
- Steepest descent: $\mathcal{O}(N\kappa\log(1/\epsilon))$; Conjugate gradient: $\mathcal{O}(N\sqrt{\kappa}\log(1/\epsilon))$
- Quantum algorithms can scale better in N but worse in κ.
- Lower bound: Quantum solver cannot generally achieve $\mathcal{O}(\kappa^{1-\delta})$ complexity for any $\delta > 0$ (Harrow-Hassadim-Lloyd, 2009)
- Goal of near-optimal quantum linear solver: $\mathcal{O}(\mathrm{poly}(n)\kappa\ \mathrm{polylog}(\kappa/\epsilon))$ complexity.

Compare the complexities of QLSP solvers

Significant progress in the past few years: Near-optimal complexity matching lower bounds.

Algorithm	Query complexity	Remark
HHL,(Harrow-Hassidim-Lloyd, 2009)	$\widetilde{\mathcal{O}}(\kappa^2/\epsilon)$	w. VTAA, complexity becomes $\widetilde{\mathcal{O}}(\kappa/\epsilon^3)$ (Ambainis 2010)
Linear combination of unitaries (LCU),(Childs-Kothari-Somma, 2017)	$\widetilde{\mathcal{O}}(\kappa^2 \mathrm{polylog}(1/\epsilon))$	w. VTAA, complexity becomes $\widetilde{\mathcal{O}}(\kappa \,\mathrm{poly}\log(1/\epsilon))$
Quantum singular value transformation (QSVT) (Gilyén-Su-Low-Wiebe, 2019)	$\widetilde{\mathcal{O}}(\kappa^2 \log(1/\epsilon))$	Queries the RHS only $\widetilde{\mathcal{O}}(\kappa)$ times
Randomization method (RM) (Subasi-Somma-Orsucci, 2019)	$\widetilde{\mathcal{O}}(\kappa/\epsilon)$	Prepares a mixed state; w. repeated phase estimation, complexity becomes $\widetilde{\mathcal{O}}(\kappa \,\mathrm{poly}\log(1/\epsilon))$
Time-optimal adiabatic quantum computing (AQC(exp)) (An-**L.**, 2019, 1909.05500)	$\widetilde{\mathcal{O}}(\kappa \,\mathrm{poly}\log(1/\epsilon))$	No need for any amplitude amplification. Use time-dependent Hamiltonian simulation.
Eigenstate filtering (**L.**-Tong, 2019, 1910.14596)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$	No need for any amplitude amplification. Does not rely on any complex subroutines.

Reformulating QLSP into an eigenvalue problem

- Weave together linear system, eigenvalue problem, differential equation (Subasi-Somma-Orsucci, 2019)
- $Q_b = I_N - |b\rangle\langle b|$. If $A|x\rangle = |b\rangle \quad \Rightarrow \quad Q_b A|x\rangle = Q_b|b\rangle = 0$
- Then

$$H_1 = \begin{pmatrix} 0 & AQ_b \\ Q_b A & 0 \end{pmatrix}, \quad |\widetilde{x}\rangle = |0\rangle|x\rangle = \begin{pmatrix} x \\ 0 \end{pmatrix}$$

$$\mathrm{Null}(H_1) = \mathrm{span}\{|\widetilde{x}\rangle, |\bar{b}\rangle\}, \quad |\bar{b}\rangle = |1\rangle|b\rangle = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

- QLSP $\quad \Rightarrow \quad$ Find an eigenvector of H_1 with eigenvalue 0.

Adiabatic computation

- Known eigenstate $H_0 \left|\psi_0\right\rangle = \lambda_0 \left|\psi_0\right\rangle$ for some H_0.
- Interested in some eigenstate $H_1 \left|\psi_1\right\rangle = \lambda_1 \left|\psi_1\right\rangle$
- $H(s) = (1 - s)H_0 + sH_1$,

$$\frac{1}{T}\mathrm{i}\partial_s \left|\psi_T(s)\right\rangle = H(s) \left|\psi_T(s)\right\rangle, \quad \left|\psi_T(0)\right\rangle = \left|\psi_0\right\rangle$$

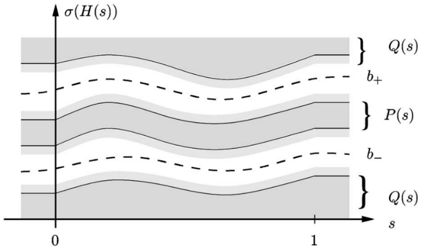
- $\left|\psi_T(1)\right\rangle \approx \psi(1)$ (up to a phase factor), T sufficiently large?
- Gate-based implementation: time-dependent Trotter, for near-optimal complexity (Low-Wiebe, 2019)

Adiabatic computation

- (Born-Fock, 1928)

  A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

- Albash, Avron, Babcock, Cirac, Cerf, Elgart, Hagedorn, Jansen, Kato, Lidar, Nenciu, Roland, Ruskai, Seiler, Wiebe...



Adiabatic quantum computation (AQC) for QLSP

- Introduce

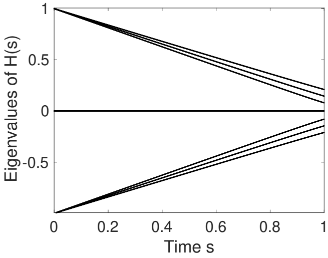
$$H_0 = \begin{pmatrix} 0 & Q_b \\ Q_b & 0 \end{pmatrix}, \quad \mathrm{Null}(H_0) = \mathrm{span}\{|\widetilde{b}\rangle, |\bar{b}\rangle\}$$

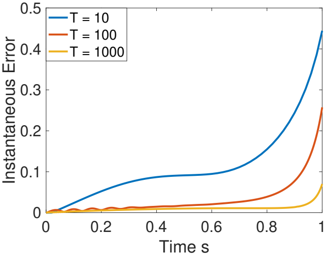
$$|\widetilde{b}\rangle = |0\rangle|b\rangle = \begin{pmatrix} b \\ 0 \end{pmatrix}, \quad |\bar{b}\rangle = |1\rangle|b\rangle = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

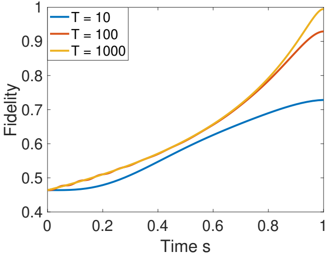
- Adiabatically connecting $|\widetilde{b}\rangle$ (zero eigenvector of H_0) to $|\widetilde{x}\rangle$ (zero eigenvector of H_1) (Subasi-Somma-Orsucci, 2019)

- Only one eigenvector in the null space is of interest: transition to $|\bar{b}\rangle$ is prohibited during dynamics

Eigenvalue gap and fidelity







Fidelity:

$$F(|\varphi\rangle, |\psi\rangle) = |\langle\varphi|\psi\rangle|^2 = \text{Tr}[P_\varphi P_\psi].$$

Adiabatic quantum computation

Theorem (Jansen-Ruskai-Seiler, 2007)

Hamiltonian $H(s)$, $P(s)$ projector to eigenspace of $H(s)$ separated by a gap $\Delta(s)$ from the rest of the spectrum of $H(s)$

$$|1 - \langle \psi_T(s) | P(s) | \psi_T(s) \rangle| \leq \eta^2(s), \quad 0 \leq s \leq 1$$

where

$$\eta(s) = \frac{C}{T} \Big\{ \frac{\|H^{(1)}(0)\|_2}{\Delta^2(0)} + \frac{\|H^{(1)}(s)\|_2}{\Delta^2(s)} + \int_0^s \left(\frac{\|H^{(2)}(s')\|_2}{\Delta^2(s')} + \frac{\|H^{(1)}(s')\|_2^2}{\Delta^3(s')} \right) ds' \Big\}.$$

T: time complexity; $1/T$ convergence.
$\Delta(s) \geq \Delta_*$, $T \sim \mathcal{O}((\Delta_*)^{-3}/\epsilon)$ (worst case)

Implication in QLSP

- Lower bound of gap (Assume $A \succ 0$ for now, can be relaxed)

$$\Delta(s) \geq \Delta_*(s) = 1 - s + s/\kappa \geq \kappa^{-1}$$

- Worst-case time complexity $T \sim \mathcal{O}(\kappa^3/\epsilon)$
- AQC inspired algorithm: randomization method (Subasi-Somma-Orsucci, 2019),

$$T \sim \mathcal{O}(\kappa \log(\kappa)/\epsilon)$$

ϵ : 2-norm error of the density matrix.

- Rescheduled dynamics.

Accelerate AQC for QLSP: Scheduling

- Goal: improve the scaling AQC w.r.t. κ.
- Adiabatic evolution with $H(f(s)) = (1 - f(s))H_0 + f(s)H_1$

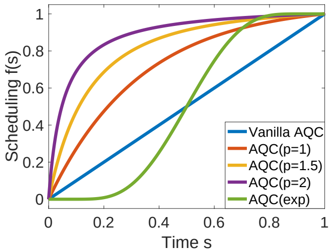
$$\frac{1}{T}\mathrm{i}\partial_s \left|\psi_T(s)\right\rangle = H(f(s)) \left|\psi_T(s)\right\rangle , \quad \left|\psi_T(0)\right\rangle = |\widetilde{b}\rangle$$

- $f(s)$: scheduling function. $0 \leq f(s) \leq 1, f(0) = 0, f(1) = 1$.
- allow $H(f(s))$ to slow down when the gap is close to 0, to cancel with the vanishing gap.
- (Roland-Cerf, 2002) for time-optimal AQC of Grover search.

Choice of scheduling function: AQC(exp)

- AQC(exp): modified schedule (slow at beginning and end)

$$f(s) = c_e^{-1} \int_0^s \exp\left(-\frac{1}{s'(1-s')}\right) \mathrm{d}s'$$



- Intuition: error bound of (Jansen-Ruskai-Seiler, 2007) and integration by parts (Wiebe-Babcock, 2012)

- Rigorous proof of exponential convergence: follow the idea of (Nenciu, 1993), asymptotic expansion of $P(s)$

Near-optimal time complexity

Theorem (An-**L.**, 1909.05500)

$A \succ 0$, condition number κ. Then for large enough $T > 0$, the error of the AQC(exp) scheme is

$$\| P_T(1) - |\widetilde{x}\rangle\langle\widetilde{x}| \|_2 \le C \log(\kappa) \exp\left(-C \left(\frac{\kappa \log^2 \kappa}{T} \right)^{-\frac{1}{4}} \right).$$

Therefore the runtime $T = \mathcal{O}\left(\kappa \log^2(\kappa) \log^4\left(\frac{\log \kappa}{\epsilon}\right)\right)$.

Near-optimal complexity (up to $\mathrm{polylog}$ factors).
Similar results for Hermitian indefinite and non-Hermitian matrices.

Conclusion

- Large-scale fully error-corrected quantum computer remains at least really, really, really hard in the near future. Think about both near-term and long-term for quantum linear algebra.
- Quantum signal processing and quantum singular value transformation: polynomial approximation theory in SU(2).
- Decay properties of phase factors.
- Statistics of random-circuit block-encoded matrix: truncated random unitaries and classical hardness
- Fast thermal state preparation.

NSF Quantum Leap Challenge Institutes, 2021-2025

Full name	Department affiliation	Institutional affiliation
Aaronson, Scott	Computer Science	UT Austin
Altman, Ehud	Physics	UC Berkeley
Campbell, Wesley	Physics	UC Los Angeles
Endres, Manuel	Physics	Caltech
Goldwasser, Shafi	EECS	UC Berkeley
Harrow, Aram	Physics	MIT
Head-Gordon, Martin	Chemistry	UC Berkeley
Hudson, Eric (co-PI)	Physics	UC Los Angeles
Häffner, Hartmut (co-PI)	Physics	UC Berkeley
Jayich, Ania	Physics	UC Santa Barbara
Kanté, Boubacar	EECS	UC Berkeley
Kubiatowicz, John	EECS	UC Berkeley
Lin, Lin	Mathematics	UC Berkeley
Mahadev, Urmila	Computing and Math	Caltech
Moore, Joel	Physics	UC Berkeley
Palsberg, Jens	Computer Science	UC Los Angeles
Reichardt, Ben	Electrical Engineering	Univ. of Southern California
Shor, Peter	Mathematics	MIT
Stamper-Kurn, Dan (PI)	Physics	UC Berkeley
Vazirani, Umesh (co-PI)	EECS	UC Berkeley
Vidick, Thomas	Computing and Math	Caltech
Weld, David	Physics	UC Santa Barbara
Whaley, K. Birgitta (co-PI)	Chemistry	UC Berkeley
Wiebe, Nathan	Physics	University of Washington
Wu, Ming	EECS	UC Berkeley
Yao, Norman	Physics	UC Berkeley

Acknowledgment

Thank you for your attention!

Lin Lin
https://math.berkeley.edu/~linlin/

U.S. DEPARTMENT OF ENERGY | Office of Science
QAT4Chem
NSF
Google AI