Quantum numerical linear algebra

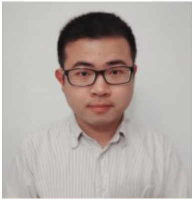
Lin Lin

Department of Mathematics, UC Berkeley
Lawrence Berkeley National Laboratory
Challenge Institute for Quantum Computation

Distinguished Lecture Series of Computational Science,
ETH Zurich, March 2021

Joint work with

Dong An
(Berkeley $\to$ Maryland)



Yulong Dong
(Berkeley)

Yu Tong
(Berkeley)

Nathan Wiebe
(Toronto)

A ritual

There is perhaps a widespread belief that a talk on quantum computation should start with a picture of Feynman..

Figure. A superposition of Feynmans

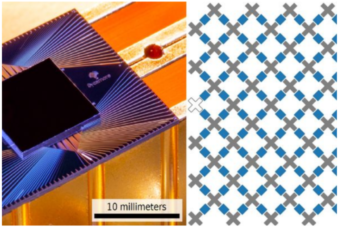
Solve nature with nature:

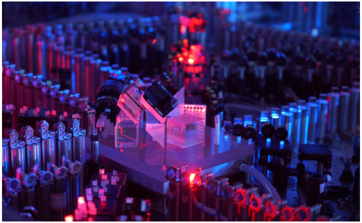
> *... if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.*
>
> *– Richard P. Feynman (1981) 1st Conference on Physics and Computation, MIT*

Quantum computation meets public attention

Google, Nature 2019
Random circuit sampling

USTC, Science 2020
Boson sampling





- After about four decades, quantum supremacy has been reached: the point *where quantum computers can do things that classical computers cannot, regardless of whether those tasks are useful.*
- *Is controlling large-scale quantum systems merely really, really hard, or is it ridiculously hard?* – John Preskill (2012)
- Quantum computer does anything useful? called quantum advantage.

Axes swung at Google

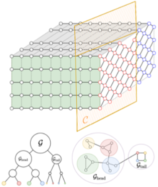
Simulating the Sycamore quantum supremacy circuits

arXiv:2103.03074v1

Feng Pan[1,2] and Pan Zhang[1,*]

[1]*Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China*
[2]*School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*

We propose a general tensor network method for simulating quantum circuits. The method is massively more efficient in computing a large number of correlated bitstring amplitudes and probabilities than existing methods. As an application, we study the sampling problem of Google's Sycamore circuits, which are believed to be beyond the reach of classical supercomputers and have been used to demonstrate quantum supremacy. Using our method, employing a small computational cluster containing 60 graphical processing units (GPUs), we have generated one million correlated bitstrings with some entries fixed, from the Sycamore circuit with 53 qubits and 20 cycles, with *linear cross-entropy benchmark* (XEB) fidelity equals 0.739, which is much higher than those in Google's quantum supremacy experiments.



Another axe swung at the Sycamore

So there's an interesting new paper on the arXiv by Feng Pan and Pan Zhang, entitled "Simulating the Sycamore supremacy circuits." It's about a new tensor contraction strategy for classically simulating Google's 53-qubit quantum supremacy experiment from Fall 2019. Using their approach, and using just 60 GPUs running for a few days, the authors say they managed to generate a million *correlated* 53-bit strings—meaning, strings that all agree on a specific subset of 20 or so bits—that achieve a high linear cross-entropy score.

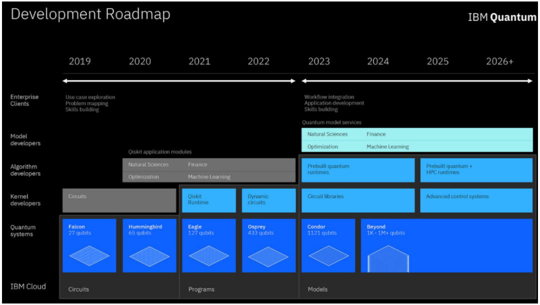
https://www.scottaaronson.com/blog/?p=5371

Quantum computer: current and (near, possible) future

We have a few quantum computers..

...

IBM’s road map (02/2021)



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.
- Normalization condition $\langle\psi|\psi\rangle = 1$.
- $U \in \mathbb{C}^{N\times N}$ is unitary. $U|\psi\rangle$ is efficient to apply ($\mathrm{poly}(n)$).
- Quantum computer: $U_K \cdots U_1 |\psi\rangle$, and then classical output by measuring one or a few qubits M times.
- Quantum cost: $MK\mathrm{poly}(n)$. Potential exponential speedup.

Quantum circuit: "graphical" tensor linear algebra

- State vectors

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

- Pauli matrices

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

$$|0\rangle \; \text{—}\boxed{X}\text{—} \; |1\rangle \qquad |1\rangle \; \text{—}\boxed{Z}\text{—} \; -|1\rangle$$

- Hadamard gate $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) := |+\rangle$

$$|0\rangle \; \text{—}\boxed{H}\text{—} \; |+\rangle$$

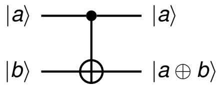
Quantum circuit: “graphical” tensor linear algebra

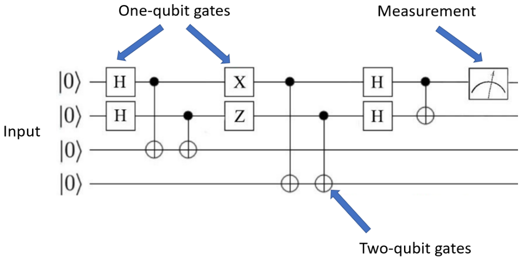
- Tensor product

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \quad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \quad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \quad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

- CNOT gate

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





Numerical linear algebra

- Linear systems of equation $Ax = b$
- Least squares problem $\min_x \|Ax - b\|_2$
- Eigenvalue decomposition $Av_i = \lambda_i v_i$
- Singular value decomposition $Av_i = u_i \sigma_i$
- Preconditioner $M^{-1}Ax = M^{-1}b$
- Matrix exponentiation $\exp(A)b$ and other matrix functions
- Machine learning, e.g. kernel ridge regression $\alpha = (K + \tilde{I})^{-1}y$
- ...

Quantum numerical linear algebra

- Solving numerical linear algebra problems on a quantum computer.
- Many interesting, exciting progresses in the past few years.
- Reasonable way towards "quantum advantage".
- Quantum linear system problem (QLSP)

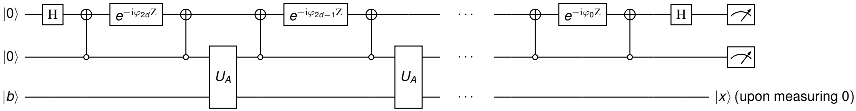
$$A|x\rangle \propto |b\rangle$$

- $A \in \mathbb{C}^{N\times N}$: cost can be $\mathcal{O}(\mathrm{polylog}(N))$.

A toy linear system problem

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

- Goal: obtain $|x\rangle = A^{-1}|b\rangle / \left\| A^{-1}|b\rangle \right\|_2 = \begin{pmatrix} 0.949 \\ -0.316 \end{pmatrix}$.
- (One possible) quantum circuit



- Does not look like any classical direct or iterative algorithm.
- $d = 80$, error of approximation $\begin{pmatrix} -7.020 \times 10^{-11} \\ -2.106 \times 10^{-10} \end{pmatrix}$

How to query A on a quantum computer?

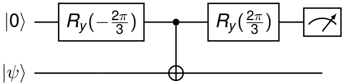
- X, I are unitaries. A is a linear combination of unitaries (LCU), and is itself non-unitary. $\kappa(A) = \|A\|_2 \left\|A^{-1}\right\|_2 = 2$.
- Idea: extend 1-qubit non-unitary matrix to a 2-qubit unitary matrix

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

- Block-encoding (Low-Chuang, 2016; called "standard form" initially)

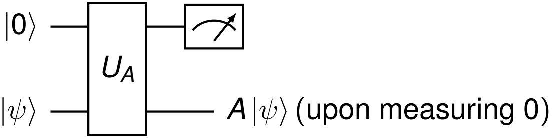
An example of block-encoding U_A

- Unitary. Use 1 ancilla qubit.



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

- U_A should be viewed as a mapping on $(\mathbb{C}^2)^{\otimes 2}$.
- Quantum circuit



Block-encoding the inverse

- Inverse

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

  Note $\left\|A^{-1}\right\| = 2 > 1$, 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

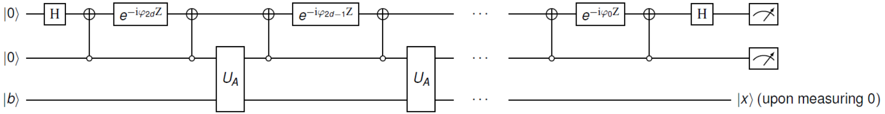
$$U_{A^{-1}} = \begin{pmatrix} 0.075 & -0.025 & 0.0 & 0.0 & 0.271j & 0.728j & -0.442j & 0.442j \\ -0.025 & 0.075 & 0.0 & 0.0 & 0.728j & 0.271j & 0.442j & -0.442j \\ 0.0 & 0.0 & 0.075 & -0.025 & -0.442j & 0.442j & -0.271j & -0.728j \\ 0.0 & 0.0 & -0.025 & 0.075 & 0.442j & -0.442j & -0.728j & -0.271j \\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

- We find

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

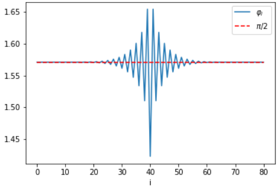
- Use 2 ancilla qubits.

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



(Simplified circuit using that U_A is Hermitian, $A \succ 0$;$\{\varphi_i\}_{i=0}^{2d}$ are called phase factors).

The same circuit works for arbitrarily large matrix ($|b\rangle$ is n-qubit).



$d = 80$. Error for approximating A^{-1}/α

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

What is under the hood?

- Polynomial approximation $A^{-1}/\alpha \approx \sum_{k=0}^{d} c_k T_k(A) := P(x)$ on $[\kappa^{-1}, 1]$ (after possible rescaling)

- Key step: parameterized polynomial representation in SU(2) $\Rightarrow$ Quantum singular value transformation circuit (QSVT) (Gilyén-Su-Low-Wiebe, 2019).

$$\mathrm{Re}\left[e^{\mathrm{i}\phi_0 Z} e^{\mathrm{i}\arccos(x) X} e^{\mathrm{i}\phi_1 Z} e^{\mathrm{i}\arccos(x) X} \cdots e^{\mathrm{i}\phi_{d-1} Z} e^{\mathrm{i}\arccos(x) X} e^{\mathrm{i}\phi_d Z}\right]_{11} = P(x)$$

- Use the same circuit (but different parameters) for various (generalized) matrix functions.

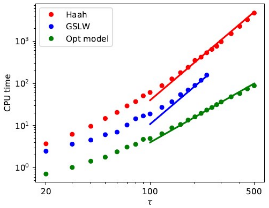
- One of the most interesting developments in quantum algorithms in the past decade.

QSPPACK

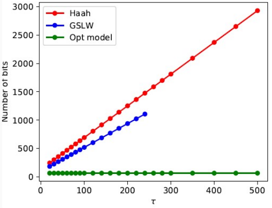
Source Code:
https://github.com/qsppack/qsppack

Example: QSP phase factors for Hamiltonian simulation

```
% time parameter in cos(tau*x), the real part of Hamiltonian
% simulation, e^(i*tau*x)
tau = 1000; parity = 0;

% max expansion order and Chebyshev coeficients in Jacobi-Anger
maxorder = ceil(1.4*tau+log(1e14));
coef = (-1).^(0:maxorder/2)' .* besselj(2*(0:maxorder/2),tau)';
coef(1) = coef(1)/2;

% stopping criteria of the optimization solver
opts.criteria = 1e-12;

% build QSP phase factors via QSPPACK
[phi,out] = QSP_solver(coef,parity,opts);
```



(a)



(b)

Efficient Phase Factor Evaluation in Quantum Signal Processing

Yulong Dong[1,2], Xiang Meng[3], K. Birgitta Whaley[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*

QLSP: quantum and classical (iterative) solvers

- Positive definite matrix. Error in A-norm. $N = 2^n$
- 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: $\widetilde{\mathcal{O}}(\kappa \mathrm{polylog}(1/\epsilon))$ complexity.

Compare the complexities of QLSP solvers

Significant progress in the past few years: Near-optimal complexity matching lower bounds. All with the promise of $\mathrm{poly}(n)$ complexity for matrix of size 2^n.

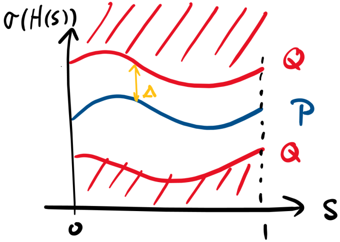
Algorithm	Query complexity
Quantum phase estimation (HHL) (Harrow-Hassidim-Lloyd, 2009)	$\widetilde{\mathcal{O}}(\kappa^2/\epsilon)$
Linear combination of unitaries (LCU) (Childs-Kothari-Somma, 2017)	$\widetilde{\mathcal{O}}(\kappa^2\mathrm{polylog}(1/\epsilon))$
Quantum singular value transformation (QSVT) (Gilyén-Su-Low-Wiebe, 2019)	$\widetilde{\mathcal{O}}(\kappa^2 \log(1/\epsilon))$
Randomization method (RM) (Subasi-Somma-Orsucci, 2019)	$\widetilde{\mathcal{O}}(\kappa/\epsilon)$
Time-optimal adiabatic quantum computing (AQC(exp)) (An-L., 2019, 1909.05500)	$\widetilde{\mathcal{O}}(\kappa\, \mathrm{poly} \log(1/\epsilon))$
Eigenstate filtering (L.-Tong, 1910.14596, Quantum 2020)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$

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...



$$H(s) = (1-s)H_0 + sH_1,$$

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

$|\psi_T(1)\rangle \approx \psi(1)$ (up to a phase factor), T sufficiently large?

Reformulating QLSP into an eigenvalue problem

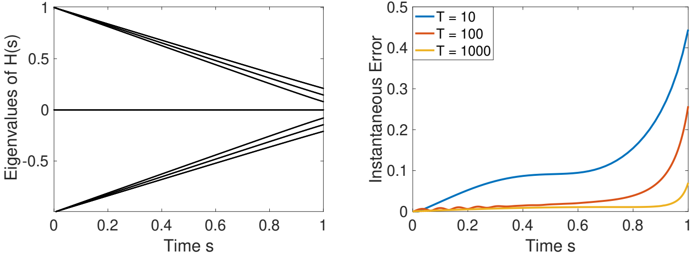
- Weave together linear system, eigenvalue problem, differential equation (Subasi-Somma-Orsucci, 2019)
- $Q_b = I_N - |b\rangle\langle b|$. If $A|x\rangle = |b\rangle \quad \Rightarrow \quad Q_b A|x\rangle = Q_b|b\rangle = 0$
- Then

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

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

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

Slow convergence with respect to T



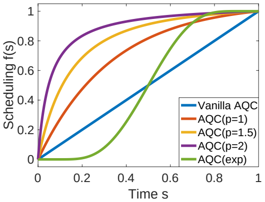
Algorithm	Time complexity (T)
Direct (Jansen-Ruskai-Seiler, 2007)	$\mathcal{O}(\kappa^3/\epsilon)$
Randomization method (RM) (Subasi-Somma-Orsucci, 2019)	$\mathcal{O}(\kappa \log(\kappa)/\epsilon)$
Time-optimal adiabatic quantum computing (AQC(exp)) (An-L., 2019)	$\mathcal{O}(\kappa\, \mathrm{poly}\log(\kappa/\epsilon))$

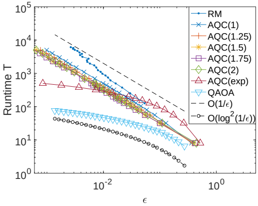
AQC(exp): exponential improvement w.r.t. ϵ

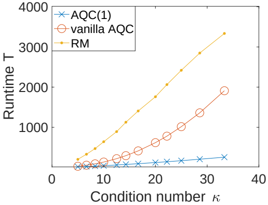
- Adiabatic evolution with $H(f(s)) = (1 - f(s))H_0 + f(s)H_1$

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

- allow $H(f(s))$ to slow down when the gap is close to 0, to cancel with the vanishing gap.

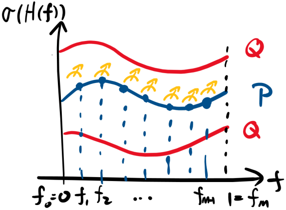




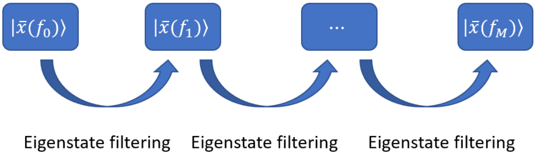


(An-L., 2019, 1909.05500)

Solving QLSP via eigenstate filtering and quantum Zeno effect



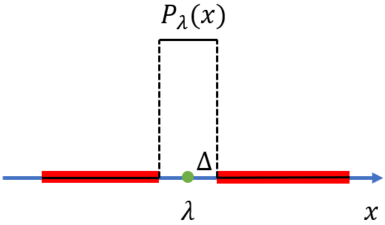
- QZE: variant of adiabatic computation. Frequent measurement $\Rightarrow$ Projection
- Measure the state $|\bar{x}(f_{j-1})\rangle$ in the eigenbasis of $H(f_j)$.
- Fidelity approaches 1 as step size decreases.
- Replace measurement with eigenstate filtering (projection).

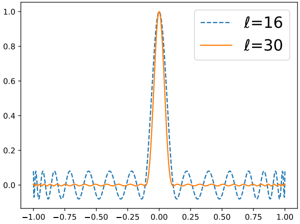


(L.-Tong, 1910.14596, Quantum 2020)

Eigenstate filtering problem

- H is Hermitian. λ is an eigenvalue of H, separated from the rest of the spectrum by a gap Δ.
- P_λ: projection operator into the λ-eigenspace of H. How to find a polynomial P to approximate P_λ?
- Requirement: $P(\lambda) = 1$ and $|P(\lambda')|$ is small for $\lambda' \in \sigma(H)\backslash\{\lambda\}$.





Application of eigenstate filtering: Solving QLSP via quantum Zeno effect (QZE)

Theorem (L.-Tong, 1910.14596)

A is a d-sparse Hermitian matrix with condition number κ, $\|A\|_2 \le 1$. Then $|x\rangle \propto A^{-1}|b\rangle$ can be obtained with fidelity $1-\epsilon$ using

1. $\mathcal{O}\left(d\kappa\left(\log(\kappa)\log\log(\kappa) + \log(\frac{1}{\epsilon})\right)\right)$ queries to $A, |b\rangle$,

2. $\mathcal{O}\left(nd\kappa\left(\log(\kappa)\log\log(\kappa) + \log(\frac{1}{\epsilon})\right)\right)$ other primitive gates,

3. $\mathcal{O}(n)$ qubits.

- Fully-gate based implementation.
- No need for time-dependent Hamiltonian simulation.
- Successive projection along the carefully scheduled adiabatic path.
- Near-optimal complexity!

Preconditioned quantum linear system solver

- QLSP with $\|A\| \gg \|B\|$

$$(A + B)\,|x\rangle \sim |b\rangle$$

- Preconditioner: A^{-1}

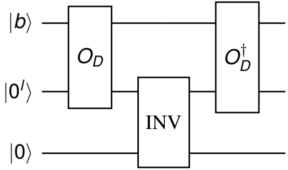
$$(I + A^{-1}B)\,|x\rangle \sim A^{-1}\,|b\rangle$$

- Condition number:
  $\kappa(I + A^{-1}B) \leq \left(1 + \|(A+B)^{-1}\|\|B\|\right)\left(1 + \|A^{-1}\|\|B\|\right)$

- Circuit depth: independent of $\|A\|$

Fast inversion of diagonal matrices

- $D = \mathrm{diag}(D_{ii})$: $\|D^{-1}\| = \min|D_{ii}| = \Omega(1)$, $\|D\| = \max|D_{ii}| \gg 1$
- Assume $O_D \ket{i}\ket{0^l} = \ket{i}\ket{D_{ii}}, \quad i \in [N]$
- Circuit U'_D for the block-encoding of D^{-1} (classical arithmetic)



- Circuit depth is independent of $\|D\|$

(Tong-An-Wiebe-L., 2008.13295)

Example: elliptic partial differential equation

- Consider a 1D Poisson's equation:

$$-\Delta u(r) + u(r) = b(r), \quad r \in \Omega = [0, 1]. \tag{1}$$

- Discretize under planewave (Fourier) basis $\exp(2\pi \mathrm{i} k r)$:

$$\begin{pmatrix} 1 & & & \\ & 1+(2\pi)^2 & & \\ & & \ddots & \\ & & & 1+(2\pi N)^2 \end{pmatrix} \begin{pmatrix} \widehat{u}_0 \\ \widehat{u}_1 \\ \vdots \\ \widehat{u}_N \end{pmatrix} = \begin{pmatrix} \widehat{b}_0 \\ \widehat{b}_1 \\ \vdots \\ \widehat{b}_N \end{pmatrix}$$

- Circuit depth of unpreconditioned method depends on $\kappa(D) = \mathcal{O}(N^2)$
- Circuit depth of fast inversion: $\mathcal{O}(1)$.

Fast matrix function evaluation: Gibbs state preparation

- Prepare $\rho_\beta = \frac{1}{Z_\beta} e^{-\beta H}$, $Z_\beta = \mathrm{Tr}(e^{-\beta H})$.
- Purified Gibbs state $|\Psi\rangle = \frac{1}{\sqrt{Z_\beta}} \sum_{x \in [N]} |x\rangle \left(e^{-\beta H/2} |x\rangle\right)$: trace out first register $\quad \Rightarrow \quad$ Obtain ρ_β
- Two new approaches (convert to linear system problems):
  - Cauchy's contour integral formula:

$$e^{-\beta H} = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma e^{-\beta z} (z - H)^{-1} \, \mathrm{d}z$$

  - Inverse transform:

$$e^{-\beta H} = e^{-\beta (H^{-1})^{-1}}$$

Fast algorithm for preparing $\propto e^{-H}|b\rangle$

	Algorithm	Query complexities
w.o. preconditioner	Phase estimation (Poulin-Wocjan, 2009)	$\widetilde{\mathcal{O}}(\frac{\alpha_H}{\xi\epsilon})$
	LCU (van Apeldoorn et al, 2020)	$\widetilde{\mathcal{O}}(\frac{\alpha_H}{\xi}\log(\frac{1}{\epsilon}))$
w. preconditioner	This work (contour integral)	$\widetilde{\mathcal{O}}(\frac{\alpha_B}{\xi\widetilde{\sigma}'^2_{\min}}\log(\frac{1}{\epsilon}))$
	This work (inverse transformation)	$\widetilde{\mathcal{O}}\left(\frac{\alpha_B}{\xi\widetilde{\sigma}^2_{\min}}\left[\log\left(\frac{1}{\epsilon}\right)\right]^5\right)$

$H = A + B, \|A\| \gg \|B\|, \alpha_H \sim \|H\|, \alpha_B \sim \|B\|$

$\xi = \|e^{-H}|b\rangle\|, \widetilde{\sigma}'_{\min} = \Omega(1/\alpha_B), \widetilde{\sigma}_{\min} = \Omega(1/(1 + \|(A+B)^{-1}\|\|B\|))$

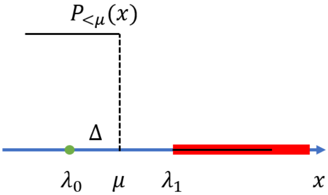
(Tong-An-Wiebe-L., 2008.13295)

Ground state energy

- $H|\psi_0\rangle = \lambda_0 |\psi_0\rangle$. w.o. assumption the problem is QMA-complete.
- Initial state $|\phi_0\rangle$ prepared by unitary U_I, w. assumptions

  (P1) Lower bound for the overlap: $|\langle\phi_0|\psi_0\rangle| \geq \gamma$,

  (P2) Bounds for the ground energy and spectral gap:
  $\lambda_0 \leq \mu - \Delta/2 < \mu + \Delta/2 \leq \lambda_1$.



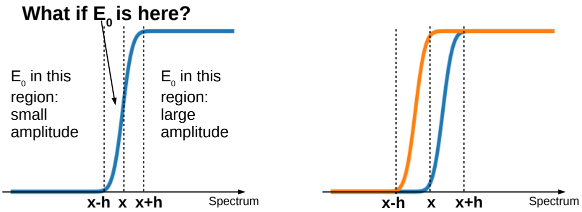
Near-optimal algorithm for finding ground state energy

		Preparation (bound known)	Ground energy	Preparation (bound unknown)
U_H	Our work	$\mathcal{O}\left(\frac{\alpha}{\gamma\Delta}\log(\frac{1}{\epsilon})\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha}{\gamma h}\log(\frac{1}{\vartheta})\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha}{\gamma\Delta}\log(\frac{1}{\vartheta\epsilon})\right)$
	GTC19	$\widetilde{\mathcal{O}}\left(\frac{\alpha}{\gamma\Delta}\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha^{3/2}}{\gamma h^{3/2}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{\alpha^{3/2}}{\gamma\Delta^{3/2}}\right)$
U_I	Our work	$\mathcal{O}\left(\frac{1}{\gamma}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\log(\frac{\alpha}{h})\log(\frac{1}{\vartheta})\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\log(\frac{\alpha}{\Delta})\log(\frac{1}{\vartheta})\right)$
	GTC19	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\sqrt{\frac{\alpha}{h}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\sqrt{\frac{\alpha}{\Delta}}\right)$
Extra qubits	Our work	$\mathcal{O}(1)$	$\mathcal{O}(\log(\frac{1}{\gamma}))$	$\mathcal{O}(\log(\frac{1}{\gamma}))$
	GTC19	$\mathcal{O}(\log(\frac{1}{\Delta}\log(\frac{1}{\epsilon})))$	$\mathcal{O}(\log(\frac{1}{h}))$	$\mathcal{O}(\log(\frac{1}{\Delta}\log(\frac{1}{\epsilon})))$

Well-known result: phase estimation; Previous best results: (Ge-Tura-Cirac, 2019)
h: precision of the ground energy estimate; $1-\vartheta$: success probability
Lower bound for the overlap: $|\langle\phi_0|\psi_0\rangle| \geq \gamma$,
Bounds for the ground energy and spectral gap: $\lambda_0 \leq \mu - \Delta/2 < \mu + \Delta/2 \leq \lambda_1$.

(L.-Tong, 2002.12508, Quantum 2020)

Binary search for ground energy



- Construct filtering polynomial with cost $\mathcal{O}(\frac{1}{h}\log(\frac{1}{\epsilon}))$ by approximating erf (Low-Chuang, 2017)
- Apply two shifted polynomials. Return with high confidence: $E_0 \geq x - h$ or $E_0 \leq x + h$.
- Perform binary search for E_0.

How to select the TOP500 quantum computers?

First, how to do the job for classical supercomputers?

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	RIKEN Center for Computational Science Japan	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D Fujitsu	7,299,072	415,530.0	513,854.7	28,335
2	DOE/SC/Oak Ridge National Laboratory United States	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM	2,414,592	148,600.0	200,794.9	10,096
3	DOE/NNSA/LLNL United States	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM / NVIDIA / Mellanox	1,572,480	94,640.0	125,712.0	7,438
4	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
5	National Super Computer Center in Guangzhou China	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 NUDT	4,981,760	61,444.5	100,678.7	18,482

NEWS

Japan Captures TOP500 Crown with Arm-Powered Supercomputer

June 22, 2020

FRANKFURT, Germany; BERKELEY, Calif.; and KNOXVILLE, Tenn.—The 55th edition of the TOP500 saw some significant additions to the list, spearheaded by a new number one system from Japan. The latest rankings also reflect a steady growth in aggregate performance and power efficiency.

The new top system, Fugaku, turned in a High Performance Linpack (HPL) result of 415.5 petaflops, besting the now second-place Summit system by a factor of 2.8x. Fugaku, is powered by Fujitsu's 48-core A64FX SoC, becoming the first number one system on the list to be powered by ARM processors. In single or further reduced precision, which are often used in machine learning and AI applications, Fugaku's peak performance is over 1,000 petaflops (1 exaflops). The new system is installed at RIKEN Center for Computational Science (R-CCS) in Kobe, Japan.

What is LINPACK? Why LINPACK?

[1] https://www.top500.org/, 55th edition of the TOP500, June 2020

Climbing the Quantum Mount Everest

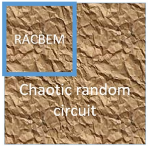
Quantum advantage

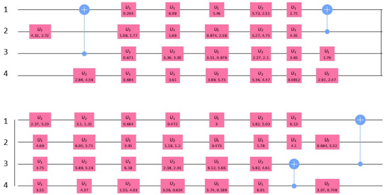
Quantum LINPACK benchmark

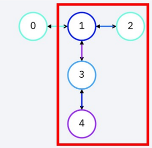
STEPS away

Quantum supremacy

RAndom Circuit Block-Encoded Matrix (RACBEM)







Qubit 1 takes $|0\rangle$
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.
  $A = (\langle 0| \otimes I_n)\, U_A\, (|0\rangle \otimes I_n)$

(Dong, L., 2006.04010)

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

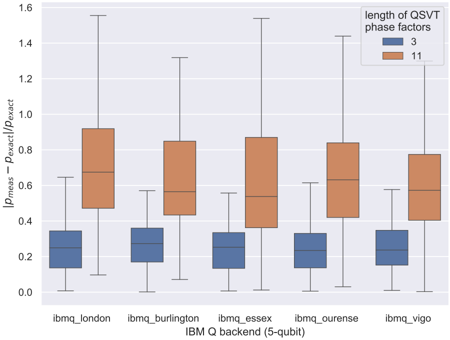
Example:

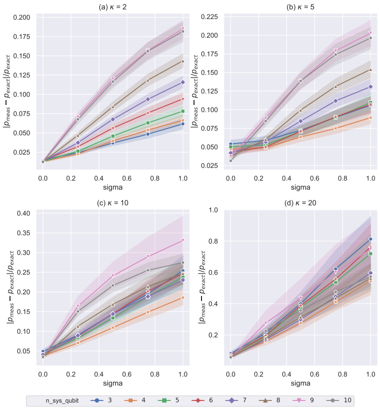
```python
from racbem import *
from qiskit import execute
import numpy as np

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
```

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)

Well conditioned linear system





Conclusion

- Large-scale fully error-corrected quantum computer remains really, really hard in the near future. Think about both near-term and long-term quantum algorithms.
- Many interesting, exciting progresses in the past few years on quantum linear algebra. Many more are coming.
- Linear system, preconditioning, eigenvalue problem, benchmark
- Future? Maybe problem with more structural information. More efficient implementation (not only asymptotic scaling)
- Heisenberg-limited ground state energy estimation for early fault-tolerant quantum computers (L.-Tong, 2102.11340)
- Quantum LINPACK benchmark on Sycamore

Advertisements

IPAM workshop on “Quantum numerical linear algebra”

Quantum Numerical Linear Algebra

JANUARY 24 - 27, 2022

OVERVIEW | LODGING | APPLICATION & REGISTRATION

Overview

With the rapid development of quantum computers, a number of quantum algorithms have been developed and tested on both superconducting qubits based machines and trapped-ion hardware. The recent development of quantum algorithms has significantly pushed forward the frontier of using quantum computers for performing a wide range of numerical linear algebra tasks, such as solving linear systems, eigenvalue decomposition, singular value decomposition, matrix function evaluation etc. While many quantum algorithms aim at future fault-tolerant quantum architecture, some of such numerical linear algebra algorithms have already demonstrated promise for being implemented on near term quantum devices. This workshop brings together leading experts in quantum numerical linear algebra, to discuss the recent development of quantum algorithms to perform linear algebra tasks for solving challenging problems in science and engineering and for various industrial and technological applications.

This workshop will include a poster session; a request for posters will be sent to registered participants in advance of the workshop.

ORGANIZING COMMITTEE

Aram Harrow (Massachusetts Institute of Technology)
Lin Lin (University of California, Berkeley (UC Berkeley), Mathematics)
Thomas Vidick (California Institute of Technology)
Nathan Wiebe (University of Toronto)

Acknowledgment

Thank you for your attention!

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

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

Block-encoding

Definition

Given an n-qubit matrix A, if we can find $\alpha, \epsilon \in \mathbb{R}_+$, and an $(m+n)$-qubit unitary matrix U_A so that that

$$\|A - \alpha \left(\langle 0^m| \otimes I_n\right) U_A \left(|0^m\rangle \otimes I_n\right)\| \leq \epsilon,$$

then U_A is called an (α, m, ϵ)-block-encoding of A.

- A "gray box" for the read-in problem.
- Many examples of block-encoding: density operators, POVM operators, d-sparse matrices, addition and multiplication of block-encoded matrices (Gilyén-Su-Low-Wiebe, 2019)