Quantum computation of Green's functions

Lin Lin

Department of Mathematics, UC Berkeley
Lawrence Berkeley National Laboratory

IMSI,

February, 2021

Joint work with

Yu Tong
(Berkeley)

Dong An
(Berkeley $\rightarrow$ Maryland)

Nathan Wiebe
(Toronto)

Fast inversion, preconditioned quantum linear system solvers, fast Green's function computation, and fast evaluation of matrix functions, (Tong, An, Wiebe, **L.**, 2008.13295)

A ritual

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

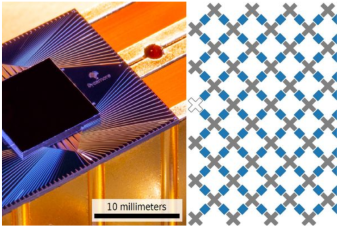
Figure. A superposition of Feynmans

Solve nature with nature:

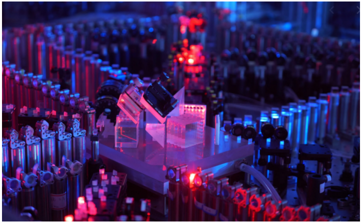
> *... if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.*
>
> *– Richard P. Feynman (1981) 1st Conference on Physics and Computation, MIT*

Quantum computation meets the public's attention

Google, Nature 2019
Random circuit sampling



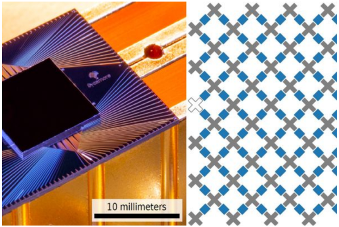
USTC, Science 2020
Boson sampling



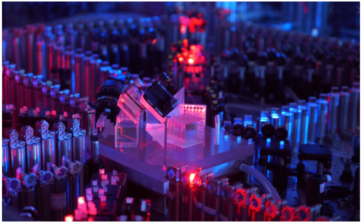
- After about four decades, quantum supremacy has been reached: the point *where quantum computers can do things that classical computers cannot, regardless of whether those tasks are useful.*

Quantum computation meets the public's attention

Google, Nature 2019
Random circuit sampling



USTC, Science 2020
Boson sampling

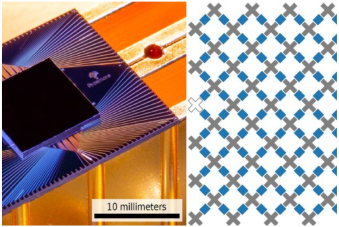


- After about four decades, quantum supremacy has been reached: the point *where quantum computers can do things that classical computers cannot, regardless of whether those tasks are useful.*

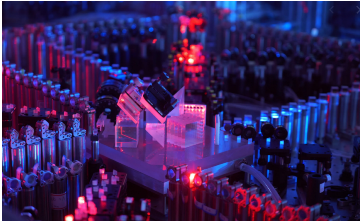
- *Is controlling large-scale quantum systems merely really, really hard, or is it ridiculously hard?* – John Preskill (2012)

Quantum computation meets the public's 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.

Quantum computer: current and (near, possible) future

We have a few quantum computers..

Google AI Quantum

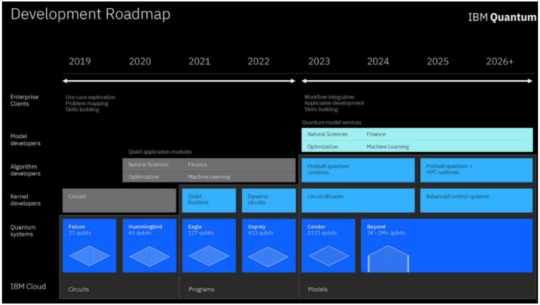
IBM

rigetti

Berkeley UNIVERSITY OF CALIFORNIA

...

IBM's road map (02/2021)



Quantum numerical linear algebra

- Solving linear systems, eigenvalue problems, matrix exponentials, least square problems, singular value decompositions etc on a quantum computer.

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.

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”.

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".

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 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)

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

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)

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

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

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 $\widetilde{\mathcal{O}}(\kappa \operatorname{poly}\log(1/\epsilon))$
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 $\widetilde{\mathcal{O}}(\kappa \operatorname{poly}\log(1/\epsilon))$
Time-optimal adiabatic quantum computing (AQC(exp)) (An-**L.**, 2019, 1909.05500)	$\widetilde{\mathcal{O}}(\kappa \operatorname{poly}\log(1/\epsilon))$	No need for any amplitude amplification. Use time-dependent Hamiltonian simulation.
Eigenstate filtering (**L.**-Tong, 1910.14596, Quantum 2020)	$\widetilde{\mathcal{O}}(\kappa \log(1/\epsilon))$	No need for any amplitude amplification. Does not rely on any complex subroutines.

Electron excitation

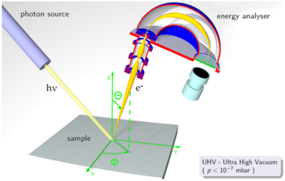
- Photoemission spectroscopy

$$\hbar\omega + E_N^0 = E_{N-1}^i + E_{kin}$$ ⇐ Energy conservation

E_N^0: Ground state energy of N-electron system
E_{N-1}^i: i-th excited state of $N-1$ electron system
E_{kin}: Kinetic energy of out-going electron (measurement)

- Quasi-particle energy

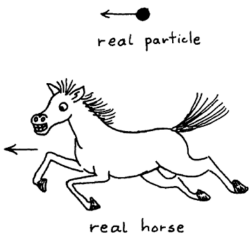
$$\varepsilon_i = E_N^0 - E_{N-1}^i = E_{kin} - \hbar\omega.$$

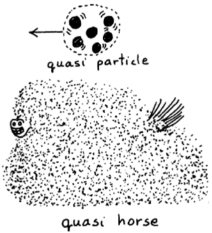


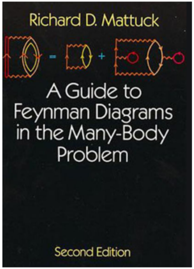
Source: Wikipedia

Quasi-particle and quasi-horse

"Quasi-horse": bare horse + response of dust (Mattuck, 1976)







Quasi-particle: bare particle + response of material

Quasi-electron: added electron + response

Quasi-hole: removed electron + response ⟵ Photoemission experiment!

Chemistry and materials

- Ionization potential (minimal energy to remove an electron)

$$I = E_{N-1}^0 - E_N^0$$

- Electron affinity (maximal energy released to add an electron)

$$A = E_N^0 - E_{N+1}^0$$

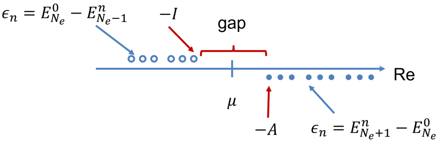
- Fundamental band gap

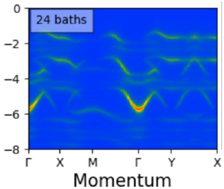
$$E_g = I - A = E_{N+1}^0 - 2E_N^0 + E_{N-1}^0$$

Curvature-like quantity

- Key quantity in chemistry and materials

Spectroscopic information and Green's function





Spectral function, 2D Hubbard model.
$A(\mathbf{k}, \omega) = -\frac{1}{\pi}\,\mathrm{Im}(G(\mathbf{k}, \omega))$

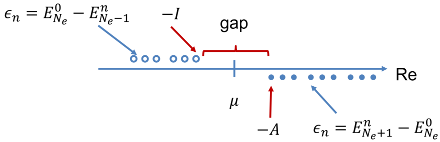
DMFT calculation: [Mejuto-Zaera, Zepeda-Nunez, Lindsey, Tubman, Whaley, **L.**, 2020]

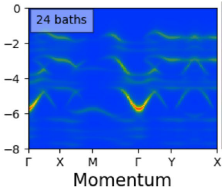
- Lehmann representation of the single-particle Green's function

$$G(z) = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n + \mathrm{i}\eta\,\mathrm{sgn}(\varepsilon_n - \mu)}, \quad \eta = 0^+.$$

ε_n: quasi-particle energy; f_n: quasi-particle wavefunction

Spectroscopic information and Green's function





Spectral function, 2D Hubbard model.
$A(\mathbf{k},\omega) = -\frac{1}{\pi}\,\mathrm{Im}(G(\mathbf{k},\omega))$

DMFT calculation: [Mejuto-Zaera, Zepeda-Nunez, Lindsey, Tubman, Whaley, **L.**, 2020]

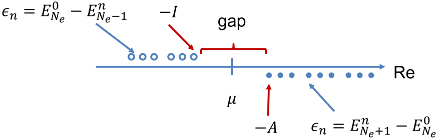
- Lehmann representation of the single-particle Green's function

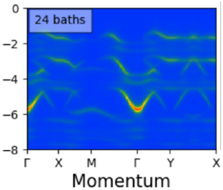
$$G(z) = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n + \mathrm{i}\eta\,\mathrm{sgn}(\varepsilon_n - \mu)}, \quad \eta = 0^+.$$

ε_n: quasi-particle energy; f_n: quasi-particle wavefunction

- Poles: ionization potential, electron affinity.

Spectroscopic information and Green's function





Spectral function, 2D Hubbard model.
$A(\mathbf{k}, \omega) = -\frac{1}{\pi} \operatorname{Im}(G(\mathbf{k}, \omega))$

DMFT calculation: [Mejuto-Zaera, Zepeda-Nunez, Lindsey, Tubman, Whaley, **L.**, 2020]

- Lehmann representation of the single-particle Green's function

$$G(z) = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n + \mathrm{i}\eta\, \mathrm{sgn}(\varepsilon_n - \mu)}, \quad \eta = 0^+.$$

ε_n: quasi-particle energy; f_n: quasi-particle wavefunction

- Poles: ionization potential, electron affinity.
- Many experiments: photoemission spectroscopy; inverse photoemission spectroscopy; ARPES...

Quantum many-body problem

- N sites (spin-orbitals)

Quantum many-body problem

- N sites (spin-orbitals)
- $\hat{a}_i, \hat{a}_i^\dagger, \hat{n}_i$: annihilation, creation, number operator at site i.

Quantum many-body problem

- N sites (spin-orbitals)
- $\hat{a}_i, \hat{a}_i^\dagger, \hat{n}_i$: annihilation, creation, number operator at site i.
- Many-body Hamiltonian (dimension: 2^N)

$$\hat{H} = \underbrace{\hat{H}_0}_{\text{non-interacting}} + \underbrace{\hat{H}_1}_{\text{interacting}}$$

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k.$$

Quantum many-body problem

- N sites (spin-orbitals)
- $\hat{a}_i, \hat{a}_i^\dagger, \hat{n}_i$: annihilation, creation, number operator at site i.
- Many-body Hamiltonian (dimension: 2^N)

$$\hat{H} = \underbrace{\hat{H}_0}_{\text{non-interacting}} + \underbrace{\hat{H}_1}_{\text{interacting}}$$

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k.$$

- $|\Psi_0\rangle$: ground state with N_e electrons ($N_e \leq 2N$)
  E_0: ground state energy.

Green's function

- Time-ordered single-particle Green's function (or Green's function for short) in the frequency domain: map $\mathbb{C} \to \mathbb{C}^{N\times N}$

$$G(z) = G^{(+)}(z) + G^{(-)}(z).$$

Green's function

- Time-ordered single-particle Green's function (or Green's function for short) in the frequency domain: map $\mathbb{C} \to \mathbb{C}^{N \times N}$

$$G(z) = G^{(+)}(z) + G^{(-)}(z).$$

- Advanced ($G^{(+)}$) and retarded ($G^{(-)}$) Green's functions

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \middle| \Psi_0 \right\rangle$$

$$G_{ij}^{(-)}(z) := \left\langle \Psi_0 \middle| \hat{a}_j^\dagger \left(z + \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_i \middle| \Psi_0 \right\rangle .$$

Green's function

- Time-ordered single-particle Green's function (or Green's function for short) in the frequency domain: map $\mathbb{C} \to \mathbb{C}^{N \times N}$

$$G(z) = G^{(+)}(z) + G^{(-)}(z).$$

- Advanced ($G^{(+)}$) and retarded ($G^{(-)}$) Green's functions

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \middle| \Psi_0 \right\rangle$$

$$G_{ij}^{(-)}(z) := \left\langle \Psi_0 \middle| \hat{a}_j^\dagger \left(z + \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_i \middle| \Psi_0 \right\rangle .$$

- Assume $|\mathrm{Im}(z)| \geq \eta > 0$ (broadening parameter)

Simplest setting: non-interacting system

- $\hat{H} = \hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j$.

Simplest setting: non-interacting system

- $\hat{H} = \hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j$.
- Very simple analytic solution via a (small) matrix inversion

$$G_0(z) = (z - T)^{-1} = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n}, \quad T f_n = \varepsilon_n f_n.$$

Simplest setting: non-interacting system

- $\hat{H} = \hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j$.
- Very simple analytic solution via a (small) matrix inversion

$$G_0(z) = (z - T)^{-1} = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n}, \quad Tf_n = \varepsilon_n f_n.$$

- Bare Green's function (bare horse)

Simplest setting: non-interacting system

- $\hat{H} = \hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j$.
- Very simple analytic solution via a (small) matrix inversion

$$G_0(z) = (z - T)^{-1} = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n}, \quad T f_n = \varepsilon_n f_n.$$

- Bare Green's function (bare horse)
- With interaction $\hat{H} = \hat{H}_0 + \hat{H}_1$. $G(z)$: quasi-horse

Simplest setting: non-interacting system

- $\hat{H} = \hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j$.
- Very simple analytic solution via a (small) matrix inversion

$$G_0(z) = (z - T)^{-1} = \sum_n \frac{f_n f_n^\dagger}{z - \varepsilon_n}, \quad T f_n = \varepsilon_n f_n.$$

- Bare Green's function (bare horse)
- With interaction $\hat{H} = \hat{H}_0 + \hat{H}_1$. $G(z)$: quasi-horse
- Self energy

$$\Sigma(z) := G^{-1}(z) - G_0^{-1}(z).$$

Next simplest setting: quantum impurity

Example: Single-impurity Anderson model (SIAM)

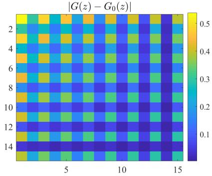
$$\hat{H} = \underbrace{\sum_{\sigma} \epsilon_f \hat{f}^\dagger_\sigma \hat{f}_\sigma + \sum_{\langle j,j' \rangle \sigma} t_{jj'} \hat{c}^\dagger_{j\sigma} \hat{c}_{j'\sigma} + \sum_{j,\sigma} \left(V_j \hat{f}^\dagger_\sigma \hat{c}_{j\sigma} + V_j^* \hat{c}^\dagger_{j\sigma} \hat{f}_\sigma \right)}_{\hat{H}_0} + \underbrace{U \hat{f}^\dagger_\uparrow \hat{f}_\uparrow \hat{f}^\dagger_\downarrow \hat{f}_\downarrow}_{\hat{H}_1}$$

Next simplest setting: quantum impurity

Example: Single-impurity Anderson model (SIAM)

$$\hat{H} = \underbrace{\sum_{\sigma} \epsilon_f \hat{f}_\sigma^\dagger \hat{f}_\sigma + \sum_{\langle j,j' \rangle \sigma} t_{jj'} \hat{c}_{j\sigma}^\dagger \hat{c}_{j'\sigma} + \sum_{j,\sigma} \left(V_j \hat{f}_\sigma^\dagger \hat{c}_{j\sigma} + V_j^* \hat{c}_{j\sigma}^\dagger \hat{f}_\sigma \right)}_{\hat{H}_0} + \underbrace{U \hat{f}_\uparrow^\dagger \hat{f}_\uparrow \hat{f}_\downarrow^\dagger \hat{f}_\downarrow}_{\hat{H}_1}$$

- Perturbation to the Green's function is global.

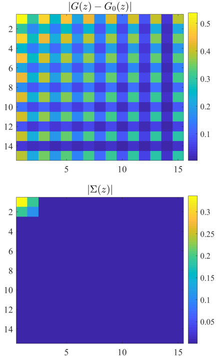


Next simplest setting: quantum impurity

Example: Single-impurity Anderson model (SIAM)

$$\hat{H} = \underbrace{\sum_{\sigma} \epsilon_f \hat{f}_\sigma^\dagger \hat{f}_\sigma + \sum_{\langle j,j' \rangle \sigma} t_{jj'} \hat{c}_{j\sigma}^\dagger \hat{c}_{j'\sigma} + \sum_{j,\sigma} \left(V_j \hat{f}_\sigma^\dagger \hat{c}_{j\sigma} + V_j^* \hat{c}_{j\sigma}^\dagger \hat{f}_\sigma \right)}_{\hat{H}_0} + \underbrace{U \hat{f}_\uparrow^\dagger \hat{f}_\uparrow \hat{f}_\downarrow^\dagger \hat{f}_\downarrow}_{\hat{H}_1}$$

- Perturbation to the Green's function is global.
- Self energy $\Sigma(z)$ is only nonzero on the impurity.

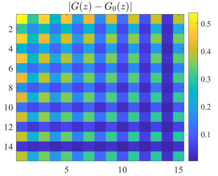


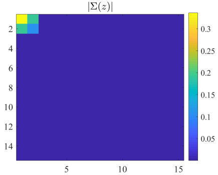
Next simplest setting: quantum impurity

Example: Single-impurity Anderson model (SIAM)

$$\hat{H} = \underbrace{\sum_{\sigma} \epsilon_f \hat{f}_\sigma^\dagger \hat{f}_\sigma + \sum_{\langle j,j' \rangle \sigma} t_{jj'} \hat{c}_{j\sigma}^\dagger \hat{c}_{j'\sigma} + \sum_{j,\sigma} \left(V_j \hat{f}_\sigma^\dagger \hat{c}_{j\sigma} + V_j^* \hat{c}_{j\sigma}^\dagger \hat{f}_\sigma \right)}_{\hat{H}_0} + \underbrace{U \hat{f}_\uparrow^\dagger \hat{f}_\uparrow \hat{f}_\downarrow^\dagger \hat{f}_\downarrow}_{\hat{H}_1}$$

- Perturbation to the Green's function is global.
- Self energy $\Sigma(z)$ is only nonzero on the impurity.
- Foundation of DMFT / CT-QMC etc. "Folk theorem" at least since Feynman (with diagrammatic arguments)



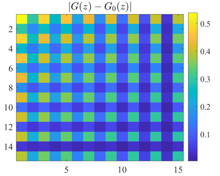


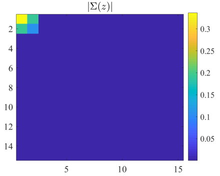
Next simplest setting: quantum impurity

Example: Single-impurity Anderson model (SIAM)

$$\hat{H} = \underbrace{\sum_{\sigma} \epsilon_f \hat{f}_\sigma^\dagger \hat{f}_\sigma + \sum_{\langle j,j' \rangle \sigma} t_{jj'} \hat{c}_{j\sigma}^\dagger \hat{c}_{j'\sigma} + \sum_{j,\sigma} \left(V_j \hat{f}_\sigma^\dagger \hat{c}_{j\sigma} + V_j^* \hat{c}_{j\sigma}^\dagger \hat{f}_\sigma \right)}_{\hat{H}_0} + \underbrace{U \hat{f}_\uparrow^\dagger \hat{f}_\uparrow \hat{f}_\downarrow^\dagger \hat{f}_\downarrow}_{\hat{H}_1}$$

- Perturbation to the Green's function is global.
- Self energy $\Sigma(z)$ is only nonzero on the impurity.
- Foundation of DMFT / CT-QMC etc. "Folk theorem" at least since Feynman (with diagrammatic arguments)
- Non-perturbative proof (for general impurities): [**L.**-Lindsey, Ann. Henri Poincare 2020]





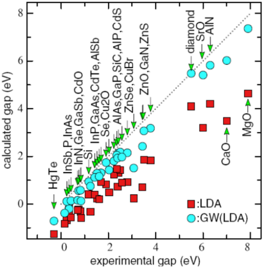
Computing Green's functions (with general interaction)

- With HF/DFT: essentially a non-interacting picture

Computing Green's functions (with general interaction)

- With HF/DFT: essentially a non-interacting picture
- Small $\hat{H}_1$: many-body perturbation theory (MBPT). GF2, GW, SOSEX, GFCC..

van Schilfgaarde et al PRL 96 226402 (2008)

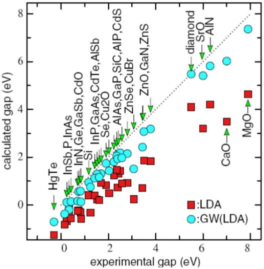


Computing Green's functions (with general interaction)

- With HF/DFT: essentially a non-interacting picture

- Small $\hat{H}_1$: many-body perturbation theory (MBPT). GF2, GW, SOSEX, GFCC..

van Schilfgaarde et al PRL 96 226402 (2008)



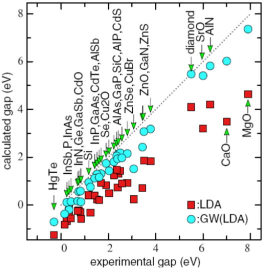
- Large $\hat{H}_1$: exact diagonalization / CI, QMC, DMRG..

Computing Green's functions (with general interaction)

- With HF/DFT: essentially a non-interacting picture

- Small $\hat{H}_1$: many-body perturbation theory (MBPT). GF2, GW, SOSEX, GFCC..

van Schilfgaarde et al PRL 96 226402 (2008)



- Large $\hat{H}_1$: exact diagonalization / CI, QMC, DMRG..

- Quantum computer

Direct computation of G: strategy 1

- Focus on $G^{(+)}$ ($G^{(-)}$ is similar)

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \left| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^{\dagger} \right| \Psi_0 \right\rangle$$

Direct computation of G: strategy 1

- Focus on $G^{(+)}$ ($G^{(-)}$ is similar)

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \middle| \Psi_0 \right\rangle$$

- $|\Phi_j\rangle := \hat{a}_j^\dagger |\Psi_0\rangle$: (sparse) matrix-vector multiplication.

Direct computation of G: strategy 1

- Focus on $G^{(+)}$ ($G^{(-)}$ is similar)

$$G^{(+)}_{ij}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}^\dagger_j \middle| \Psi_0 \right\rangle$$

- $|\Phi_j\rangle := \hat{a}^\dagger_j |\Psi_0\rangle$: (sparse) matrix-vector multiplication.
- $|\zeta_j\rangle := \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} |\Phi_j\rangle$: solve a big linear system. Most challenging

Direct computation of G: strategy 1

- Focus on $G^{(+)}$ ($G^{(-)}$ is similar)

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \middle| \Psi_0 \right\rangle$$

- $|\Phi_j\rangle := \hat{a}_j^\dagger |\Psi_0\rangle$: (sparse) matrix-vector multiplication.
- $|\zeta_j\rangle := \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} |\Phi_j\rangle$: solve a big linear system. Most challenging
- $G_{ij}^{(+)}(z) = \langle \Phi_i | \zeta_j \rangle$: inner product.

Direct computation of G: strategy 1

- Focus on $G^{(+)}$ ($G^{(-)}$ is similar)

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^{\dagger} \middle| \Psi_0 \right\rangle$$

- $|\Phi_j\rangle := \hat{a}_j^{\dagger} |\Psi_0\rangle$: (sparse) matrix-vector multiplication.

- $|\zeta_j\rangle := \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} |\Phi_j\rangle$: solve a big linear system. Most challenging

- $G_{ij}^{(+)}(z) = \langle \Phi_i | \zeta_j \rangle$: inner product.

- Seems to be a lucid approach.

Direct computation of G: strategy 2

- Rewrite

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \middle| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \middle| \Psi_0 \right\rangle := \langle \Psi_0 | A | \Psi_0 \rangle$$

Direct computation of G: strategy 2

- Rewrite

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \left| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \right| \Psi_0 \right\rangle := \langle \Psi_0 | A | \Psi_0 \rangle$$

- Perform “matrix-multiplication”

$$A = \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger$$

Direct computation of G: strategy 2

- Rewrite

$$G_{ij}^{(+)}(z) := \left\langle \Psi_0 \left| \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger \right| \Psi_0 \right\rangle := \langle \Psi_0 | A | \Psi_0 \rangle$$

- Perform “matrix-multiplication”

$$A = \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger$$

- Sounds worse / crazier, but this is what we are going to do.

Block-encoding

- Quantum gates have to be unitary.

Block-encoding

- Quantum gates have to be unitary.
- $A = \hat{a}_i \left(z - \left[\hat{H} - E_0 \right] \right)^{-1} \hat{a}_j^\dagger$ is not unitary.

Block-encoding

- Quantum gates have to be unitary.
- $A = \hat{a}_i \left(z - \left[\hat{H} - E_0\right]\right)^{-1} \hat{a}_j^\dagger$ is not unitary.
- Idea: extend n-qubit non-unitary matrix to a $(n+m)$-qubit unitary matrix (Low-Chuang, 2016; called “standard form” initially)

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

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.

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)

Block-encoding for Green's function computation

- Jordan-Wigner transformation

$$\hat{a}_i = Z^{\otimes(i-1)} \otimes \frac{1}{2}(X + \mathrm{i}Y) \otimes I^{\otimes(N-i)},$$

$$\hat{a}_i^\dagger = Z^{\otimes(i-1)} \otimes \frac{1}{2}(X - \mathrm{i}Y) \otimes I^{\otimes(N-i)}, \quad \hat{n}_i = \frac{1}{2}(I - Z_i).$$

Block-encoding for Green's function computation

- Jordan-Wigner transformation

$$\hat{a}_i = Z^{\otimes(i-1)} \otimes \frac{1}{2}(X + \mathrm{i}Y) \otimes I^{\otimes(N-i)},$$

$$\hat{a}_i^\dagger = Z^{\otimes(i-1)} \otimes \frac{1}{2}(X - \mathrm{i}Y) \otimes I^{\otimes(N-i)}, \quad \hat{n}_i = \frac{1}{2}(I - Z_i).$$

- $\hat{a}_i, \hat{a}_j^\dagger, \hat{n}_i$ are not unitary, but X, Y, Z, I are (Pauli-matrices).

Block-encoding for Green's function computation

- Jordan-Wigner transformation

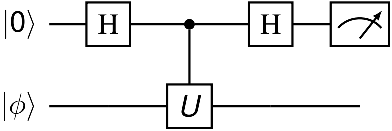
$$\hat{a}_i = Z^{\otimes(i-1)} \otimes \frac{1}{2}(X + \mathrm{i}Y) \otimes I^{\otimes(N-i)},$$

$$\hat{a}_i^\dagger = Z^{\otimes(i-1)} \otimes \frac{1}{2}(X - \mathrm{i}Y) \otimes I^{\otimes(N-i)}, \quad \hat{n}_i = \frac{1}{2}(I - Z_i).$$

- $\hat{a}_i, \hat{a}_j^\dagger, \hat{n}_i$ are not unitary, but X, Y, Z, I are (Pauli-matrices).
- Provide a $(1,1,0)$-block-encodings of $\hat{a}_i, \hat{a}_j^\dagger, \hat{n}_i$.

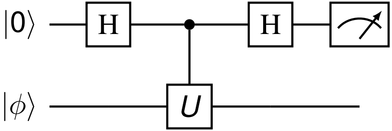
Basic quantum strategy: Hadamard test

- *U* is an *n*-qubit unitary matrix. Hadamard gate $\mathrm{H} := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$



Basic quantum strategy: Hadamard test

- U is an n-qubit unitary matrix. Hadamard gate $\mathrm{H} := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

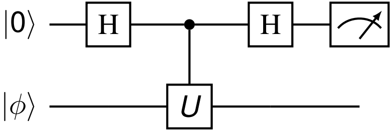


- Success probability of measuring 0 is

$$p(0) = \frac{1}{2}(1 + \mathrm{Re}\,\langle\phi|U|\phi\rangle)$$

Basic quantum strategy: Hadamard test

- U is an n-qubit unitary matrix. Hadamard gate $\mathrm{H} := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$



- Success probability of measuring 0 is

$$p(0) = \frac{1}{2}(1 + \mathrm{Re}\,\langle\phi|U|\phi\rangle)$$

- A similar circuit with success probability $\frac{1}{2}(1 + \mathrm{Im}\,\langle\phi|U|\phi\rangle)$
  $\Rightarrow$ Obtain $\langle\phi|U|\phi\rangle$

Hadamard test for Green's function computation

- If we can block-encode the inverse: $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$.

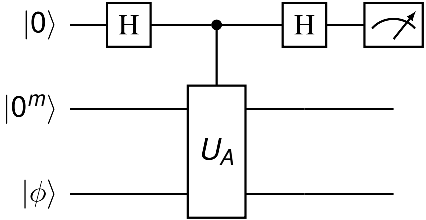
  Product of block-encoded matrices $A = \hat{a}_i \left(z - \left[\hat{H} - E_0\right]\right)^{-1} \hat{a}_j^{\dagger}$, call it U_A, which is a $(1, m, \epsilon)$-block-encoding.

Hadamard test for Green's function computation

- If we can block-encode the inverse: $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$.

  Product of block-encoded matrices $A = \hat{a}_i \left(z - \left[\hat{H} - E_0\right]\right)^{-1} \hat{a}_j^\dagger$, call it U_A, which is a $(1, m, \epsilon)$-block-encoding.

- Hadamard test circuit

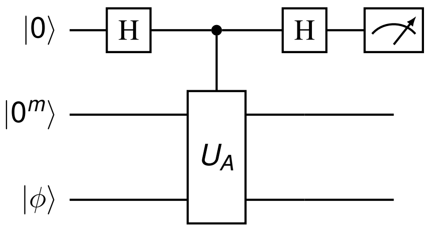


Hadamard test for Green's function computation

- If we can block-encode the inverse: $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$.

  Product of block-encoded matrices $A = \hat{a}_i \left(z - \left[\hat{H} - E_0\right]\right)^{-1} \hat{a}_j^\dagger$, call it U_A, which is a $(1, m, \epsilon)$-block-encoding.

- Hadamard test circuit



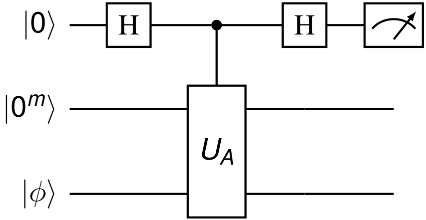
- Success probability $p(0) = \frac{1}{2}(1 + \mathrm{Re}\,\langle\phi|A|\phi\rangle)$.

Hadamard test for Green's function computation

- If we can block-encode the inverse: $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$.

  Product of block-encoded matrices $A = \hat{a}_i \left(z - \left[\hat{H} - E_0\right]\right)^{-1} \hat{a}_j^{\dagger}$, call it U_A, which is a $(1, m, \epsilon)$-block-encoding.

- Hadamard test circuit



- Success probability $p(0) = \frac{1}{2}(1 + \text{Re}\langle\phi|A|\phi\rangle)$.

- Cost: dominated by the circuit depth of U_A.

Estimate the circuit depth

- Determined by the depth of block-encoding $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$, assume well conditioned

Estimate the circuit depth

- Determined by the depth of block-encoding $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$, assume well conditioned
- If we can query a block-encoding of $\hat{H}$, then the circuit depth $\propto \alpha_H \sim \left\|\hat{H}\right\|$ (dependence on other parameters are omitted)

Estimate the circuit depth

- Determined by the depth of block-encoding $\left(z - \left[\hat{H} - E_0\right]\right)^{-1}$, assume well conditioned
- If we can query a block-encoding of $\hat{H}$, then the circuit depth $\propto \alpha_H \sim \left\|\hat{H}\right\|$ (dependence on other parameters are omitted)
- Basically, this is due to the polynomial approximation of $x \mapsto x^{-1}$ on the interval $\left[1, \left\|\hat{H}\right\|\right]$.

Problem: large block-encoding factor α_H

- Recall

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{pqrs} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k.$$

Problem: large block-encoding factor α_H

- Recall

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{pqrs} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k.$$

- Planewave / real space refined spatial discretization: $\left\|\hat{H}\right\| \approx \left\|\hat{H}_0\right\| \gg \left\|\hat{H}_1\right\|$

Problem: large block-encoding factor α_H

- Recall

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{pqrs} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k.$$

- Planewave / real space refined spatial discretization: $\left\| \hat{H} \right\| \approx \left\| \hat{H}_0 \right\| \gg \left\| \hat{H}_1 \right\|$

- Hubbard model, large U limit: $\left\| \hat{H} \right\| \approx \left\| \hat{H}_1 \right\| \gg \left\| \hat{H}_0 \right\|$

Problem: large block-encoding factor α_H

- Recall

$$\hat{H}_0 = \sum_{ij=1}^{N} T_{ij} \hat{a}_i^\dagger \hat{a}_j, \quad \hat{H}_1 = \sum_{ijkl=1}^{N} V_{pqrs} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k.$$

- Planewave / real space refined spatial discretization: $\left\|\hat{H}\right\| \approx \left\|\hat{H}_0\right\| \gg \left\|\hat{H}_1\right\|$
- Hubbard model, large U limit: $\left\|\hat{H}\right\| \approx \left\|\hat{H}_1\right\| \gg \left\|\hat{H}_0\right\|$
- Let us write $\hat{H} = \hat{A} + \hat{B}$, where $\left\|\hat{A}\right\| \gg \left\|\hat{B}\right\|$.

Green's functions of quantum many-body systems

Main result (informal)

Algorithm	Queries to block-encodings
HHL	$\widetilde{\mathcal{O}}(\frac{\lvert z\rvert+\alpha_H}{\eta^3\epsilon^2})$
LCU/QSVT	$\widetilde{\mathcal{O}}(\frac{\lvert z\rvert+\alpha_H}{\eta^2\epsilon})$
Our work	$\widetilde{\mathcal{O}}(\frac{\alpha_B}{\widetilde{\sigma}_{\min}^2\epsilon})$

- $\hat{H} = \hat{A} + \hat{B}$, with $\widetilde{\sigma}_{\min} = \Omega(\eta/\alpha_B)$, and $\left\|\hat{A}\right\| \gg \left\|\hat{B}\right\|$.
- Block-encodings in our work involves fast inversion.

Fast inversion

- Key idea: instead of block-encode a matrix A, if $\|A\|$ is large but $\|A^{-1}\|$ is small, try to directly block-encode A^{-1}, instead of relying on a standard QLSP solver.

Fast inversion

- Key idea: instead of block-encode a matrix A, if $\|A\|$ is large but $\|A^{-1}\|$ is small, try to directly block-encode A^{-1}, instead of relying on a standard QLSP solver.
- Fast block-encoding of the inverse (there is a subtle difference from fast solution of the linear system)

Fast inversion

- Key idea: instead of block-encode a matrix A, if $\|A\|$ is large but $\|A^{-1}\|$ is small, try to directly block-encode A^{-1}, instead of relying on a standard QLSP solver.
- Fast block-encoding of the inverse (there is a subtle difference from fast solution of the linear system)
- Parallel to fast-forwarding.

Fast inversion

- Key idea: instead of block-encode a matrix A, if $\|A\|$ is large but $\|A^{-1}\|$ is small, try to directly block-encode A^{-1}, instead of relying on a standard QLSP solver.

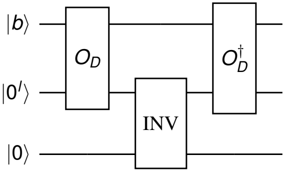
- Fast block-encoding of the inverse (there is a subtle difference from fast solution of the linear system)

- Parallel to fast-forwarding.

- Not violating lower bound by (Harrow-Hassidim-Lloyd, 2009)

Fast inversion of diagonal matrices

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



- Circuit depth is independent of $\left\|D^{-1}\right\|, \|D\|$

Fast inversion of diagonal matrices

- The inversion circuit INV (with $\alpha'_D \geq \|D^{-1}\|$):

$$\mathrm{INV}\,|\zeta\rangle\,|0\rangle = |\zeta\rangle \left(\frac{1}{\alpha'_D \zeta}\,|0\rangle + \sqrt{1 - \left|\frac{1}{\alpha'_D \zeta}\right|^2}\,|1\rangle \right).$$

- Output ($\alpha'_D \sim \|D^{-1}\|$):

$$U'_D\,|b\rangle\,|0^l\rangle\,|0\rangle = \alpha'_D \sum_i (D_{ii})^{-1} b_i\,|i\rangle\,|0^l\rangle\,|0\rangle + \sum_i \sqrt{1 - |(\alpha'_D D_{ii})^{-1}|^2}\, b_i\,|i\rangle\,|0^l\rangle\,|1\rangle\,.$$

- U'_D is an $(\alpha'_D, m'_D, 0)$-block-encoding of D^{-1} with $\alpha'_D = \mathcal{O}(\|D^{-1}\|)$ and $m'_D = \mathcal{O}(l + \mathrm{poly}\log(N))$

Example: elliptic partial differential equation

- Consider a 1D Poisson's equation:

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

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

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

- $\alpha_D = \mathcal{O}(N^2)$, $\alpha'_D = \mathcal{O}(1)$, $\kappa(D) = \mathcal{O}(N^2)$
- $\widehat{b}_j$ decays rapidly as $j \to \infty$: $\|D^{-1} b\| = \Theta(1)$
- Cost: $\mathcal{O}(\alpha'_D / \|D^{-1} |b\rangle \|) = \mathcal{O}(1)$
- QSVT still scales $\mathcal{O}(N^2)$

Fast inversion beyond diagonal matrices

- Diagonal matrices D: U'_D
- 1-sparse matrices $A = \Pi D$
  - if we have access to Π^{-1}: $A^{-1} = D^{-1}\Pi^{-1}$
  - Also fast-invertible if we only have query access to the column of the single nonzero element in each row as well as to the value of the each element
- Normal matrices $A = VDV^\dagger$

$$U'_A = (V \otimes I_{l+1})U'_D(V^\dagger \otimes I_{l+1}).$$

Preconditioned quantum linear system solver

- Consider

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

- Assume very large $\|A\|$ and moderate $\|B\|, \|A^{-1}\|, \|(A+B)^{-1}\|$, thus $\kappa(A+B) \sim \mathcal{O}(\|A\|)$

- An example: $-\Delta u(r) + V(r)u(r) = b(r)$

- Oracles:
  - U_A': an $(\alpha_A', m_A', 0)$-block-encoding of A^{-1} prepared by the fast inversion procedure
  - U_B: an $(\alpha_B, m_B, 0)$-block-encoding of B
  - U_b: $\left|b\right\rangle = U_b\left|0^n\right\rangle$

- Preconditioner: A^{-1}

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

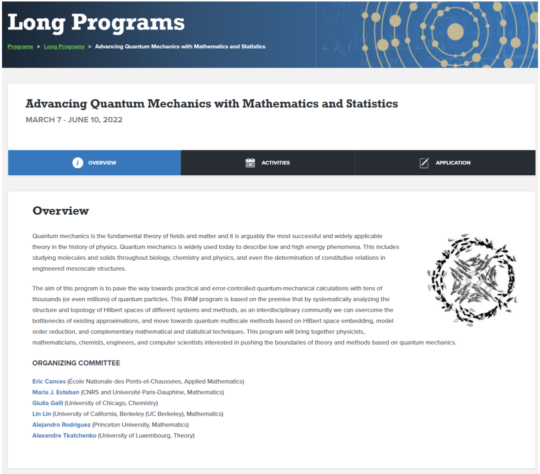
Preconditioned quantum linear system solver

$$A^{-1} \to A^{-1}B \to I + A^{-1}B \to (I + A^{-1}B)^{-1}$$
$$\to (I + A^{-1}B)^{-1}A^{-1} = (A+B)^{-1}$$

- A $\left(\frac{4\alpha'_A}{3\widetilde{\sigma}_{\min}}, 2m'_A + m_B + 3, \delta'\right)$-block-encoding of $(A+B)^{-1}$ using $\mathcal{O}\left(\frac{\alpha'_A\alpha_B}{\widetilde{\sigma}_{\min}} \log\left(\frac{\alpha'_A}{\delta'\widetilde{\sigma}_{\min}}\right)\right)$ queries, with $\widetilde{\sigma}_{\min} \geq 1/(1 + \|(A+B)^{-1}\|\|B\|)$.
- Solving $(A+B)\left|x\right\rangle \sim \left|b\right\rangle$: $\mathcal{O}\left(\frac{{\alpha'_A}^2\alpha_B}{\xi\widetilde{\sigma}^2_{\min}} \log\left(\frac{\alpha'_A}{\widetilde{\sigma}_{\min}\xi\epsilon}\right)\right)$ queries, with $\xi = \|(A+B)^{-1}\left|b\right\rangle\|$
- Worst case: $\xi \sim \|(A+B)\|^{-1} \sim \Omega(1/\kappa((A+B)))$
- Best case: $\xi \sim \mathcal{O}(1)$
- Outperform QSVT in both worst and best case

Advertisements

1. IPAM Long Program, 3/7-6/10, 2022



Long Programs

Programs > Long Programs > Advancing Quantum Mechanics with Mathematics and Statistics

Advancing Quantum Mechanics with Mathematics and Statistics

MARCH 7 - JUNE 10, 2022

OVERVIEW | ACTIVITIES | APPLICATION

Overview

Quantum mechanics is the fundamental theory of fields and matter and it is arguably the most successful and widely applicable theory in the history of physics. Quantum mechanics is widely used today to describe low and high energy phenomena. This includes studying molecules and solids throughout biology, chemistry and physics, and even the determination of constitutive relations in engineered mesoscale structures.

The aim of this program is to pave the way towards practical and error-controlled quantum-mechanical calculations with tens of thousands (or even millions) of quantum particles. This IPAM program is based on the premise that by systematically analyzing the structure and topology of Hilbert spaces of different systems and methods, as an interdisciplinary community we can overcome the bottlenecks of existing approximations, and move towards quantum multiscale methods based on Hilbert space embedding, model order reduction, and complementary mathematical and statistical techniques. This program will bring together physicists, mathematicians, chemists, engineers, and computer scientists interested in pushing the boundaries of theory and methods based on quantum mechanics.

ORGANIZING COMMITTEE

Eric Cances (École Nationale des Ponts-et-Chaussées, Applied Mathematics)
Maria J. Esteban (CNRS and Université Paris-Dauphine, Mathematics)
Giulia Galli (University of Chicago, Chemistry)
Lin Lin (University of California, Berkeley (UC Berkeley), Mathematics)
Alejandro Rodriguez (Princeton University, Mathematics)
Alexandre Tkatchenko (University of Luxembourg, Theory)

2. IPAM workshop on "Quantum numerical linear algebra", 1/24-1/27, 2022

- Aram Harrow, MIT
- Lin Lin, UC Berkeley
- Thomas Vidick, Caltech
- Nathan Wiebe, University of Toronto

(Website available soon)

A >< = qubit

Acknowledgment

Thank you for your attention!

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

A concrete, toy example

$$A = \frac{1}{4}X + \frac{3}{4}I = \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}$$

- X, I are unitaries. A is a linear combination of unitaries (LCU), and is itself non-unitary. $\kappa(A) = 2$ (invertible)

A concrete, toy example

$$A = \frac{1}{4}X + \frac{3}{4}I = \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}$$

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

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

A concrete, toy example

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

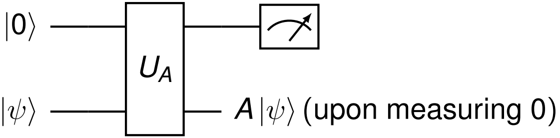
$$U_A = \begin{pmatrix} \boxed{\begin{matrix} 0.750 & 0.250 \\ 0.250 & 0.750 \end{matrix}} & \begin{matrix} 0.433 & -0.433 \\ -0.433 & 0.433 \end{matrix} \\ \begin{matrix} 0.433 & -0.433 \\ -0.433 & 0.433 \end{matrix} & \begin{matrix} 0.250 & 0.750 \\ 0.750 & 0.250 \end{matrix} \end{pmatrix}$$

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 $(\mathbb{C}^2)^{\otimes 2}$.



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

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

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 $\alpha > 1$)

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

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

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

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.

Cost analysis

Lemma (Tong, An, Wiebe, **L.**)

Given

1. *State* $|\phi\rangle$ *prepared with trace-distance error* ς *by a unitary circuit* U_ϕ *with probability at least* p
2. A *is given through its* $(\alpha, m, 0)$*-block-encoding* U_A*,*

Then $\langle\phi|A|\phi\rangle$ *can be estimated to precision* $2\alpha\varsigma + \epsilon$ *with probability at least* $1 - \delta$*, using*

1. $\mathcal{O}((\alpha/\epsilon)\log(1/\delta))$ *applications of* U_A *and its inverse*
2. $\mathcal{O}((\alpha/\sqrt{p}\epsilon)\log(1/\varsigma)\log(1/\delta))$ *applications of* U_ϕ *and its inverse*
3. $\mathcal{O}((\alpha/\sqrt{p}\epsilon)\log(1/\varsigma)\log(1/\delta))$ *other one- and two-qubit gates.*

- Compute Green's function, using amplitude estimation to improve dependence on ϵ (Brassard-Høyer-Mosca-Tapp, 2002)

Cost analysis

Lemma (Tong, An, Wiebe, **L.**)

Given

1. *State* $|\phi\rangle$ *prepared with trace-distance error* ς *by a unitary circuit* U_ϕ *with probability at least p*
2. *A is given through its* $(\alpha, m, 0)$*-block-encoding* U_A*,*

Then $\langle\phi|A|\phi\rangle$ *can be estimated to precision* $2\alpha\varsigma + \epsilon$ *with probability at least* $1 - \delta$*, using*

1. $\mathcal{O}((\alpha/\epsilon)\log(1/\delta))$ *applications of* U_A *and its inverse*
2. $\mathcal{O}((\alpha/\sqrt{p}\epsilon)\log(1/\varsigma)\log(1/\delta))$ *applications of* U_ϕ *and its inverse*
3. $\mathcal{O}((\alpha/\sqrt{p}\epsilon)\log(1/\varsigma)\log(1/\delta))$ *other one- and two-qubit gates.*

- Compute Green's function, using amplitude estimation to improve dependence on ϵ (Brassard-Høyer-Mosca-Tapp, 2002)

- There is some (but not a whole lot) of rooms to maneuver, but we can ask what is the circuit depth for U_A.