Nuclear structure (and reactions) with Quantum Computers - IV

Alessandro Roggero

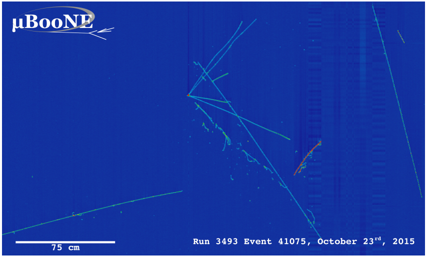


figure credit: μBooNE collab.

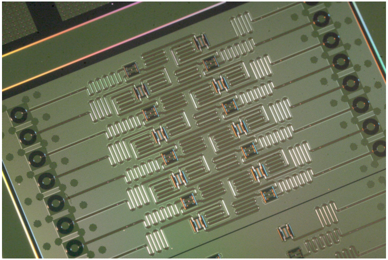


figure credit: IBM

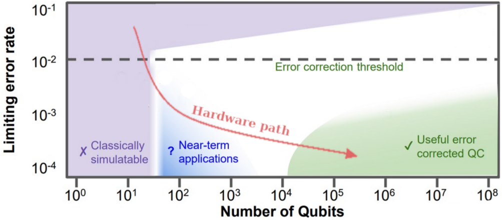
QC and QIS for NP

JLAB – 18 March, 2020

The plan for today

- nuclear dynamics, computation of scattering cross sections
  - EXAMPLE: neutrino-^{40}Ar cross section for DUNE
  - inclusive scattering and the response function
  - calculation of two-point functions
  - direct calculation of response in frequency space
- complexity of these calculations, can we actually run them on current/near-term NISQ devices?

from Google AI



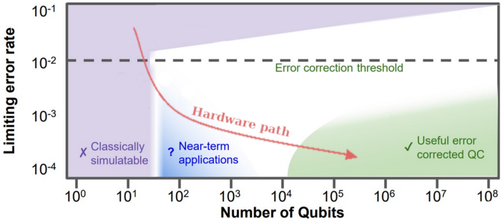
- advanced algorithms

- Fermionic Swap Networks
- Linear Combination of Unitaries
- Amplitude Amplification
- Qubitization

The plan for today

- nuclear dynamics, computation of scattering cross sections
  - EXAMPLE: neutrino-^{40}Ar cross section for DUNE
  - inclusive scattering and the response function
  - calculation of two-point functions
  - direct calculation of respone in frequency space
- complexity of these calculations, can we actually run them on current/near-term NISQ devices?

from Google AI



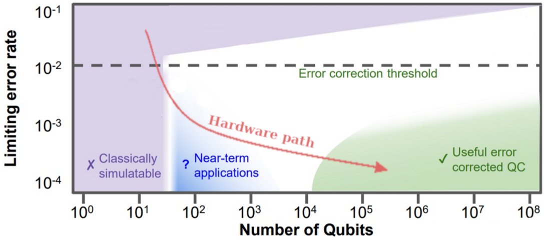
- advanced algorithms

- Fermionic Swap Networks
- Linear Combination of Unitaries
- Amplitude Amplification
- Qubitization

Final recap of first day

1. quantum computers can simulate efficiently the time-evolution operator $U(\tau) = \exp(i\tau H)$ for r-local Hamiltonians
   - for target error ϵ this requires $\mathcal{O}(\text{poly}(n, \tau, 1/\epsilon)4^r)$ gates
   - Jordan Wigner on n qubits leads to n-local terms!
     - SPOILER: this might not be a problem in practice
   - tomorrow we'll generalize this and find better scaling
2. if we can prepare an energy eigenstate $|\phi\rangle$ we can use this to measure it's phase with accuracy Δ using a total propagation time $\tau \sim 1/\Delta$
3. if $|\Psi\rangle$ has overlap $\alpha = |\langle\phi|\Psi\rangle|^2$, we just add $\mathcal{O}(1/\alpha)$ repetitions

from Google AI

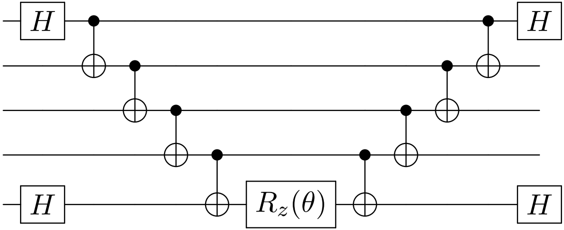


Problem of non-locality of Jordan-Wigner mapping

The Jordan-Wigner mapping stores the information about the fermionic parity into strings of Z operators between fermionic modes:

$$a_p^\dagger a_q = \frac{X_p X_q + Y_p Y_q}{2} \left(Z_{p+1} \cdots Z_{q-1} \right)$$

This leads to large CNOT circuits to compute the parity, for instance



$$= \exp\left(i \frac{\theta}{2} X_1 X_5 Z_2 Z_3 Z_4 \right)$$

When executing the full propagator many phases cancel [Hastings et al. (2014)]

$$U_1(\tau) = \prod_{p,q} \exp(i\tau h_{p,q} a_p^\dagger a_q)$$

Problem of non-locality of Jordan-Wigner mapping II

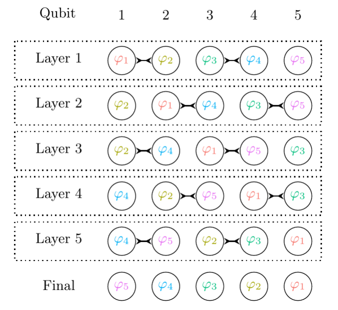
When executing the full propagator many phases cancel [Hastings et al. (2014)]

$$U_1(\tau) = \prod_{p,q}^{n} \exp(i\tau h_{p,q} a_p^\dagger a_q) = \prod_{p,q}^{n} u_{p,q}(\tau)$$

Fermionic Swap Network

Babbush et al. (2017), Kivlichan et al. (2018)

We can implement $U_1(\tau)$ exactly using only $3\binom{n}{2}$ two qubit gates



- n layers of $\frac{n-1}{2}$ two qubit gates

$$W_{i,i+1} = u_{i,i+1}(\tau) f_{SWAP}$$

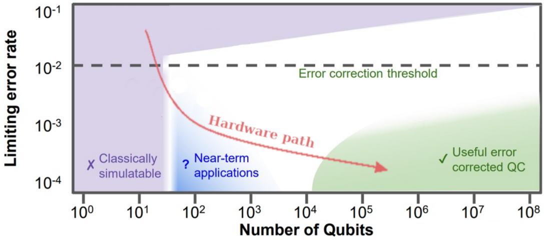
$$f_{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

- final qubit order is reversed

Final recap of first day

1. quantum computers can simulate efficiently the time-evolution operator $U(\tau) = \exp(i\tau H)$ for r-local Hamiltonians
    - for target error ϵ this requires $\mathcal{O}(\text{poly}(n, \tau, 1/\epsilon)4^r)$ gates
    - Jordan Wigner on n qubits leads to n-local terms!
        - SPOILER: this might not be a problem in practice
    - tomorrow we'll generalize this and find better scaling
2. if we can prepare an energy eigenstate $|\phi\rangle$ we can use this to measure it's phase with accuracy Δ using a total propagation time $\tau \sim 1/\Delta$
3. if $|\Psi\rangle$ has overlap $\alpha = |\langle\phi|\Psi\rangle|^2$, we just add $\mathcal{O}(1/\alpha)$ repetitions

from Google AI



Time evolution: short time approximations

Baker–Campbell–Hausdorff formula

$$\exp\left(it\left(A+B\right)\right)=\exp\left(itA\right)\exp\left(itB\right)+\mathcal{O}\left([A,B]t^2\right)$$

- can be extended to large time intervals by slicing $[0,t]$ into L intervals

$$\exp\left(it\left(A+B\right)\right)=\left[\exp\left(i\frac{t}{L}A\right)\exp\left(i\frac{t}{L}B\right)\right]^L+\mathcal{O}\left([A,B]\frac{t^2}{L}\right)$$

and we need: $L=\mathcal{O}\left(t^2/\epsilon\right)$

Time evolution: short time approximations

Baker–Campbell–Hausdorff formula

$$\exp\left(it\left(A+B\right)\right)=\exp\left(itA\right)\exp\left(itB\right)+\mathcal{O}\left([A,B]t^2\right)$$

- can be extended to large time intervals by slicing $[0,t]$ into L intervals

$$\exp\left(it\left(A+B\right)\right)=\left[\exp\left(i\frac{t}{L}A\right)\exp\left(i\frac{t}{L}B\right)\right]^L+\mathcal{O}\left([A,B]\frac{t^2}{L}\right)$$

  and we need: $L=\mathcal{O}\left(t^2/\epsilon\right)$

- we can define higher order formulas with better scaling Suzuki (1991)

$$S_2(t)=\exp\left(i\frac{t}{2}B\right)\exp\left(itA\right)\exp\left(i\frac{t}{2}B\right)\Rightarrow\epsilon=\mathcal{O}\left(t^3\right)$$

In general formulas with error $\mathcal{O}(t^\gamma)$ will require $L=\mathcal{O}\left(t^{\frac{\gamma}{\gamma-1}}\left(\frac{1}{\epsilon}\right)^{\frac{1}{\gamma-1}}\right)$

- is it possible to get a better scaling? Say $\mathcal{O}\left(t+\log(1/\epsilon)\right)$?

Efficient time evolution with qubitization

Assume the Hamiltonian can be decomposed efficiently as

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha_k > 0$$

EFFICIENT: the number of terms $L = \text{poly}(n)$, gate cost of U_k is $\text{poly}(n)$

Time evolution using Quantum Signal Processing Low & Chuang (2016)

The time evolution operator $U(t)$ can be approximated with error less than ϵ using $\mathcal{O}\left(t + \log(1/\epsilon)\right)$ calls to a quantum operation, the **qubiterate** W_Q, that can be implemented with $\mathcal{O}(\text{poly}(n))$ gates

Efficient time evolution with qubitization

Assume the Hamiltonian can be decomposed efficiently as

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha_k > 0$$

EFFICIENT: the number of terms $L = \text{poly}(n)$, gate cost of U_k is $\text{poly}(n)$

Time evolution using Quantum Signal Processing — Low & Chuang (2016)

The time evolution operator $U(t)$ can be approximated with error less than ϵ using $\mathcal{O}(t + \log(1/\epsilon))$ calls to a quantum operation, the **qubiterate** W_Q, that can be implemented with $\mathcal{O}(\text{poly}(n))$ gates

- is this the limit or can we hope for something better?

No Fast-Forward Theorem — Atia & Aharonov (2017)

Without additional details on the Hamiltonian, you can't beat $\mathcal{O}(t)$

Appetizer: the Linear Combination of Unitaries method

Assume the Hamiltonian can be decomposed efficiently as

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha_k > 0$$

and define $\alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$.

Linear Combination of Unitaries (LCU) — Childs & Wiebe (2012)

We can apply the operation $H_\alpha = H/\alpha$ to a state $|\Psi\rangle$ with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$, $m = \lceil \log_2(L) \rceil$ ancilla qubits and $\mathcal{O}(\text{poly}(L, n))$ gates

Appetizer: the Linear Combination of Unitaries method

Assume the Hamiltonian can be decomposed efficiently as

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha_k > 0$$

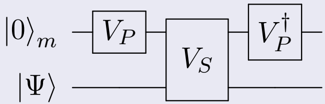
and define $\alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$.

Linear Combination of Unitaries (LCU)

Childs & Wiebe (2012)

We can apply the operation $H_\alpha = H/\alpha$ to a state $|\Psi\rangle$ with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$, $m = \lceil \log_2(L) \rceil$ ancilla qubits and $\mathcal{O}(\text{poly}(L,n))$ gates

prepare $\quad V_P |0\rangle_m = \sum_{k=0}^{L-1} \sqrt{\frac{\alpha_{k+1}}{\alpha}} |k\rangle \qquad$ select $\quad V_S = \sum_{k=0}^{L-1} |k\rangle\langle k| \otimes U_{k+1}$



the final state is

$$|\Phi\rangle = \frac{H}{\alpha} |\Psi\rangle \otimes |0\rangle_m + \left|0^\perp\right\rangle$$

where $\langle 0^\perp | 0\rangle_m = 0$.

Appetizer: the Linear Combination of Unitaries method

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

Linear Combination of Unitaries (LCU) — Childs & Wiebe (2012)

We can apply the operation $H_\alpha = H/\alpha$ to a state $|\Psi\rangle$ with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$, $m = \lceil \log_2(L) \rceil$ ancilla qubits and $\mathcal{O}(\text{poly}(L, n))$ gates

Expand time-evolution operator as Taylor series

$$\exp(itH) \approx \sum_{k=0}^{K} \frac{(it)^k}{k!} H^k = \sum_{k=0}^{K} \frac{(it)^k}{k!} \sum_{q_0 \cdots q_k = 1}^{L} \alpha_{q_0} \cdots \alpha_{q_k} U_{q_0} \cdots U_{q_k}$$

Appetizer: the Linear Combination of Unitaries method

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

Linear Combination of Unitaries (LCU)

Childs & Wiebe (2012)

We can apply the operation $H_\alpha = H/\alpha$ to a state $|\Psi\rangle$ with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$, $m = \lceil \log_2(L) \rceil$ ancilla qubits and $\mathcal{O}(\text{poly}(L,n))$ gates

Expand time-evolution operator as Taylor series

$$\exp(itH) \approx \sum_{k=0}^{K} \frac{(it)^k}{k!} H^k = \sum_{k=0}^{K} \frac{(it)^k}{k!} \sum_{q_0 \cdots q_k = 1}^{L} \alpha_{q_0} \cdots \alpha_{q_k} U_{q_0} \cdots U_{q_k}$$

Time evolution with truncated Taylor series

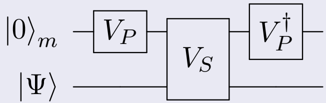
Berry et al. (2015)

We need $\mathcal{O}\left(\alpha t \frac{\log(\alpha t/\epsilon)}{\log(\log(\alpha t/\epsilon))}\right)$ calls to V_S, V_P costing $\approx \mathcal{O}(L^2)$ gates

- use **oblivious amplitude amplification** to boost probability $P \approx 1$

Amplitude Amplification

Brassard & Hoyer (1997), Grover (1998)



$$\Rightarrow |\Phi\rangle = U|\Psi\rangle|0\rangle_m = \frac{H}{\alpha}|\Psi\rangle|0\rangle_m + |0^\perp\rangle$$

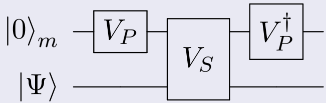
More generally we can consider the situation where

$$|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^\perp\rangle\ ,$$

then we can use the reflections $R_\Psi = \mathbb{1} - 2|\Psi\rangle\langle\Psi|$ and $R_\Phi = 2|\Phi\rangle\langle\Phi| - \mathbb{1}$ to rotate in the 2D subspace spanned by $|\Psi\rangle$ and $|\Psi^\perp\rangle$.

Amplitude Amplification

Brassard & Hoyer (1997), Grover (1998)

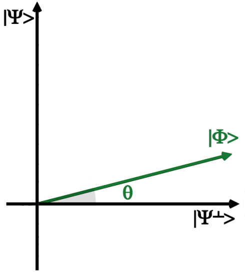


$$\Rightarrow |\Phi\rangle = U|\Psi\rangle|0\rangle_m = \frac{H}{\alpha}|\Psi\rangle|0\rangle_m + |0^\perp\rangle$$

More generally we can consider the situation where

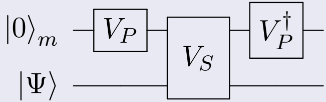
$$|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^\perp\rangle \ ,$$

then we can use the reflections $R_\Psi = \mathbb{1} - 2|\Psi\rangle\langle\Psi|$ and $R_\Phi = 2|\Phi\rangle\langle\Phi| - \mathbb{1}$ to rotate in the 2D subspace spanned by $|\Psi\rangle$ and $|\Psi^\perp\rangle$.



Amplitude Amplification

Brassard & Hoyer (1997), Grover (1998)

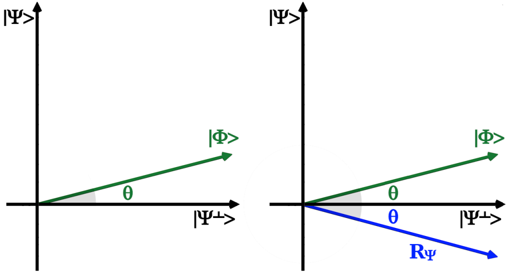


$$\Rightarrow |\Phi\rangle = U\,|\Psi\rangle\,|0\rangle_m = \frac{H}{\alpha}\,|\Psi\rangle\,|0\rangle_m + \left|0^{\perp}\right\rangle$$

More generally we can consider the situation where

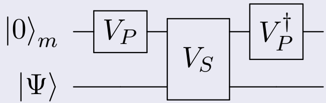
$$|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^{\perp}\rangle\ ,$$

then we can use the reflections $R_{\Psi} = \mathbb{1} - 2|\Psi\rangle\langle\Psi|$ and $R_{\Phi} = 2|\Phi\rangle\langle\Phi| - \mathbb{1}$ to rotate in the 2D subspace spanned by $|\Psi\rangle$ and $|\Psi^{\perp}\rangle$.



Amplitude Amplification

Brassard & Hoyer (1997), Grover (1998)

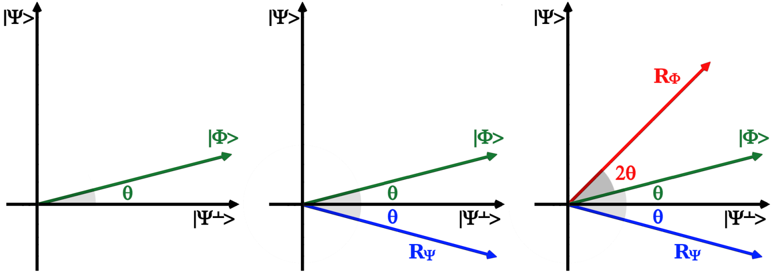


$$\Rightarrow |\Phi\rangle = U\,|\Psi\rangle\,|0\rangle_m = \frac{H}{\alpha}\,|\Psi\rangle\,|0\rangle_m + \left|0^{\perp}\right\rangle$$

More generally we can consider the situation where

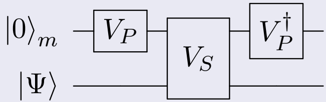
$$|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^{\perp}\rangle\;,$$

then we can use the reflections $R_\Psi = \mathbb{1} - 2|\Psi\rangle\langle\Psi|$ and $R_\Phi = 2|\Phi\rangle\langle\Phi| - \mathbb{1}$ to rotate in the 2D subspace spanned by $|\Psi\rangle$ and $|\Psi^{\perp}\rangle$.



Amplitude Amplification

Brassard & Hoyer (1997), Grover (1998)



$$\Rightarrow |\Phi\rangle = U\,|\Psi\rangle\,|0\rangle_m = \frac{H}{\alpha}\,|\Psi\rangle\,|0\rangle_m + \left|0^\perp\right\rangle$$

More generally we can consider the situation where

$$|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^\perp\rangle\ ,$$

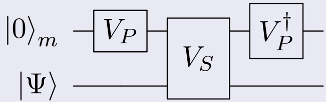
then we can use the reflections $R_\Psi = \mathbb{1} - 2|\Psi\rangle\langle\Psi|$ and $R_\Phi = 2|\Phi\rangle\langle\Phi| - \mathbb{1}$ to rotate in the 2D subspace spanned by $|\Psi\rangle$ and $|\Psi^\perp\rangle$. We have

$$W^n\,|\Phi\rangle = \sin\left((2n+1)\theta\right)|\Psi\rangle + \cos\left((2n+1)\theta\right)|\Psi^\perp\rangle \quad W = R_\Phi R_\Psi$$

and will need $n \approx \pi/4\theta$ iterations to reach maximum success probability.

Amplitude Amplification

Brassard & Hoyer (1997), Grover (1998)



$$\Rightarrow |\Phi\rangle = U|\Psi\rangle|0\rangle_m = \frac{H}{\alpha}|\Psi\rangle|0\rangle_m + |0^\perp\rangle$$

More generally we can consider the situation where

$$|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^\perp\rangle\ ,$$

then we can use the reflections $R_\Psi = \mathbb{1} - 2|\Psi\rangle\langle\Psi|$ and $R_\Phi = 2|\Phi\rangle\langle\Phi| - \mathbb{1}$ to rotate in the 2D subspace spanned by $|\Psi\rangle$ and $|\Psi^\perp\rangle$. We have

$$W^n|\Phi\rangle = \sin\left((2n+1)\theta\right)|\Psi\rangle + \cos\left((2n+1)\theta\right)|\Psi^\perp\rangle \quad W = R_\Phi R_\Psi$$

and will need $n \approx \pi/4\theta$ iterations to reach maximum success probability.

Oblivious amplitude amplification

Berry et. al (2014)

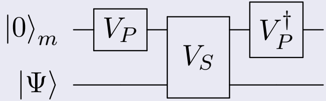
For unitary H we only need $W = -UR_0U^\dagger R_0$ where R_0 reflects over $|0\rangle_m$

Quick recap on this last part

Brassard & Hoyer (1997), Grover (1998), Childs & Wiebe (2012), Berry et. al (2014-2015)

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

- Linear Combination of Unitaries (LCU)



$$\Rightarrow |\Phi\rangle = U\,|\Psi\rangle\,|0\rangle_m = \frac{H}{\alpha}\,|\Psi\rangle\,|0\rangle_m + \left|0^\perp\right\rangle$$

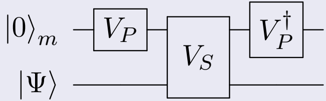
$\Rightarrow$ the algorithm succeeds with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$.

Quick recap on this last part

Brassard & Hoyer (1997), Grover (1998), Childs & Wiebe (2012), Berry et. al (2014-2015)

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

- Linear Combination of Unitaries (LCU)



$$\Rightarrow |\Phi\rangle = U\,|\Psi\rangle\,|0\rangle_m = \frac{H}{\alpha}\,|\Psi\rangle\,|0\rangle_m + \left|0^\perp\right\rangle$$

$\Rightarrow$ the algorithm succeeds with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$.

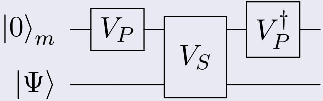
- Amplitude Amplification can boost probability at cost $\mathcal{O}\left(1/\sqrt{P}\right)$

Quick recap on this last part

Brassard & Hoyer (1997), Grover (1998), Childs & Wiebe (2012), Berry et. al (2014-2015)

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

- Linear Combination of Unitaries (LCU)



$$\Rightarrow |\Phi\rangle = U|\Psi\rangle|0\rangle_m = \frac{H}{\alpha}|\Psi\rangle|0\rangle_m + |0^\perp\rangle$$

$\Rightarrow$ the algorithm succeeds with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$.

- Amplitude Amplification can boost probability at cost $\mathcal{O}\left(1/\sqrt{P}\right)$
- Time evolution with Taylor exp. can achieve scaling $\mathcal{O}\left(\alpha t \log(\alpha t/\epsilon)\right)$

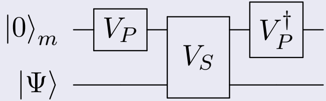
Standard Trotter-like approaches can only give $\mathcal{O}\left(t^{1+\frac{1}{\eta}}/\epsilon^{\frac{1}{\eta}}\right)$ with $\eta \geq 1$

Quick recap on this last part

Brassard & Hoyer (1997), Grover (1998), Childs & Wiebe (2012), Berry et. al (2014-2015)

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

- Linear Combination of Unitaries (LCU)



$$\Rightarrow |\Phi\rangle = U |\Psi\rangle |0\rangle_m = \frac{H}{\alpha} |\Psi\rangle |0\rangle_m + |0^\perp\rangle$$

$\Rightarrow$ the algorithm succeeds with probability $P = \langle\Psi|H^2|\Psi\rangle/\alpha^2$.

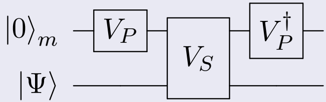
- Amplitude Amplification can boost probability at cost $\mathcal{O}\left(1/\sqrt{P}\right)$
- Time evolution with Taylor exp. can achieve scaling $\mathcal{O}\left(\alpha t \log(\alpha t/\epsilon)\right)$

Standard Trotter-like approaches can only give $\mathcal{O}\left(t^{1+\frac{1}{\eta}}/\epsilon^{\frac{1}{\eta}}\right)$ with $\eta \geq 1$

- use Oblivious Amplitude Amplification to achieve this deterministically with much smaller prefactors (simpler reflection operators)

Back to qubitization

Low & Chuang (2016)



$$\Rightarrow |\Phi\rangle = U|\Psi\rangle|0\rangle_m = \frac{H}{\alpha}|\Psi\rangle|0\rangle_m + \left|0^\perp\right\rangle$$

If we denote $|\phi_P\rangle = V_P|0\rangle_m$ the **qubiterate** W_Q can be defined as

$$W_Q = (2|\phi_P\rangle\langle\phi_P| - \mathbb{1})\, V_S$$

one can show that if $H/\alpha = \sum_n \lambda_n |n\rangle\langle n|$ then we can write

$$W_Q|n\rangle|\phi_P\rangle = W_Q|P_n\rangle = \lambda_n|P_n\rangle + \sqrt{1-\lambda_n^2}\left|P_n^\perp\right\rangle$$

for **every** eigenstate $\Rightarrow W_Q$ generates rotations in $span\{|P_n\rangle, \left|P_n^\perp\right\rangle\}\ \forall n$

$$W_Q = \bigoplus_n \begin{pmatrix} \lambda_n & -\sqrt{1-\lambda_n^2} \\ \sqrt{1-\lambda_n^2} & \lambda_n \end{pmatrix} = \bigoplus_n e^{iY\arccos(\lambda_n)}$$

What is this useful for?

$$W_Q = \bigoplus_n \begin{pmatrix} \lambda_n & -\sqrt{1-\lambda_n^2} \\ \sqrt{1-\lambda_n^2} & \lambda_n \end{pmatrix} = \bigoplus_n e^{iY \arccos(\lambda_n)} \sim e^{i\widetilde{Y} \arccos\left(\frac{H}{\alpha}\right)}$$

- we have a 2D invariant subspace for every eigenvalue ⇒ powers of the qubiterate W_Q will generate rotations in these subspaces

Quantum Signal Processing

Low & Chuang (2016)

We can use this to generate polynomial functions of the Hamiltonian. Take

$$W_Q(\theta) = \bigoplus_n \begin{pmatrix} \lambda_n & -ie^{-i\theta}\sqrt{1-\lambda_n^2} \\ ie^{i\theta}\sqrt{1-\lambda_n^2} & \lambda_n \end{pmatrix} = R(\theta) W_Q R^\dagger(\theta)$$

where $R(\theta)$ uses V_P, then we can use this **phased iterate** to generate

$$\prod_{j=1}^{N} W_Q(\theta_j) = \sum_{j=1}^{N/2} \left[a_j\left(\vec{\theta}\right) + ib_j\left(\vec{\theta}\right)\right] \left(\frac{H}{\alpha}\right)^j$$

- (QSP+Taylor) ⇒ optimal scaling algorithm for time evolution

What else is this useful for?

$$W_Q = \bigoplus_n \begin{pmatrix} \lambda_n & -\sqrt{1-\lambda_n^2} \\ \sqrt{1-\lambda_n^2} & \lambda_n \end{pmatrix} = \bigoplus_n e^{iY_n \arccos(\lambda_n)} \sim e^{i\widetilde{Y} \arccos\left(\frac{H}{\alpha}\right)}$$

- the spectrum of W_Q is the (almost) the same as the Hamiltonian

$$\forall \lambda_n = E_n/\alpha \quad \text{we have} \quad \eta_\pm = \exp\left(i \pm \arccos(\lambda_n)\right)$$

- the qubiterate W_Q can be implemented **exactly** using $\mathcal{O}(L^2)$ gates

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

What else is this useful for?

$$W_Q = \bigoplus_n \begin{pmatrix} \lambda_n & -\sqrt{1-\lambda_n^2} \\ \sqrt{1-\lambda_n^2} & \lambda_n \end{pmatrix} = \bigoplus_n e^{iY_n \arccos(\lambda_n)} \sim e^{i\widetilde{Y} \arccos\left(\frac{H}{\alpha}\right)}$$

- the spectrum of W_Q is the (almost) the same as the Hamiltonian

$$\forall \lambda_n = E_n/\alpha \quad \text{we have} \quad \eta_\pm = \exp\left(i \pm \arccos(\lambda_n)\right)$$

- the qubiterate W_Q can be implemented **exactly** using $\mathcal{O}(L^2)$ gates

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

Quantum Phase Estimation with no time evolution
Berry et al. (2018)

We can use W_Q instead of $\exp\left(itH\right)$ to get the energy spectrum

What else is this useful for?

$$W_Q = \bigoplus_n \begin{pmatrix} \lambda_n & -\sqrt{1-\lambda_n^2} \\ \sqrt{1-\lambda_n^2} & \lambda_n \end{pmatrix} = \bigoplus_n e^{iY_n \arccos(\lambda_n)} \sim e^{i\widetilde{Y} \arccos\left(\frac{H}{\alpha}\right)}$$

- the spectrum of W_Q is the (almost) the same as the Hamiltonian

$$\forall \lambda_n = E_n/\alpha \quad \text{we have} \quad \eta_\pm = \exp\left(i \pm \arccos(\lambda_n)\right)$$

- the qubiterate W_Q can be implemented **exactly** using $\mathcal{O}(L^2)$ gates

$$H = \sum_{k=1}^{L} \alpha_k U_k \quad \text{with} \quad U_k^\dagger U_k = \mathbb{1} \quad \text{and} \quad \alpha = \sum_{k=1}^{L} \alpha_k \geq \|H\|$$

Quantum Phase Estimation with no time evolution Berry et al. (2018)

We can use W_Q instead of $\exp(itH)$ to get the energy spectrum

- can we use W_Q to compute scattering-cross sections?

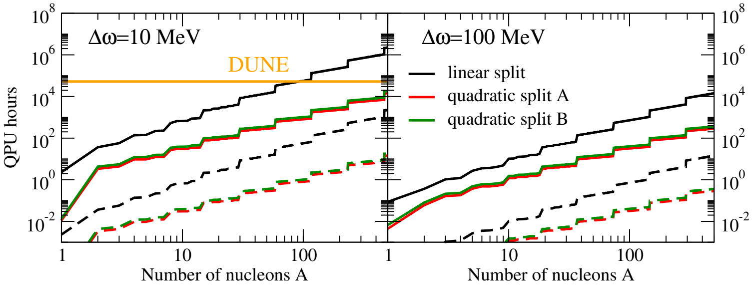
Qubitization for scattering cross section

Algorithm 1 — Roggero & Carlson (2018)

- prepare target state
- apply excitation operator
- measure energy using $U(t)$
- final time evolution
- measurement

Algorithm 2 — Roggero, Li et al. (2019)

- prepare target state
- apply excitation operator
- measure energy using W_Q
- final time evolution
- measurement



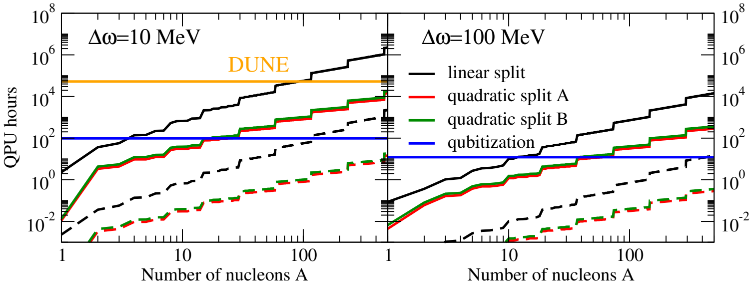
Qubitization for scattering cross section

Algorithm 1 — Roggero & Carlson (2018)

- prepare target state
- apply excitation operator
- measure energy using $U(t)$
- final time evolution
- measurement

Algorithm 2 — Roggero, Li et al. (2019)

- prepare target state
- apply excitation operator
- measure energy using W_Q
- final time evolution
- measurement



What have we learned in these couple days

- Quantum Computers are good at simulating time evolution for Hamiltonians with 2 and 3-body interactions (and possibly others)

$$H = \sum_{ij} K_{ij} a_i^\dagger a_j + \sum_{ijkl} U_{ijkl} a_i^\dagger a_j^\dagger a_k a_l + \sum_{ijklmn} V_{ijklmn} a_i^\dagger a_j^\dagger a_k^\dagger a_l a_m a_n$$

What have we learned in these couple days

- Quantum Computers are good at simulating time evolution for Hamiltonians with 2 and 3-body interactions (and possibly others)

$$H = \sum_{ij} K_{ij} a_i^\dagger a_j + \sum_{ijkl} U_{ijkl} a_i^\dagger a_j^\dagger a_k a_l + \sum_{ijklmn} V_{ijklmn} a_i^\dagger a_j^\dagger a_k^\dagger a_l a_m a_n$$

- Quantum Phase Estimation uses this to compute eigenvalues exponentially faster than with (exact) classical methods
  - we can use this to prepare eigenstates
  - we can use this to compute cross sections
  - ...

What have we learned in these couple days

- Quantum Computers are good at simulating time evolution for Hamiltonians with 2 and 3-body interactions (and possibly others)

$$H = \sum_{ij} K_{ij} a_i^\dagger a_j + \sum_{ijkl} U_{ijkl} a_i^\dagger a_j^\dagger a_k a_l + \sum_{ijklmn} V_{ijklmn} a_i^\dagger a_j^\dagger a_k^\dagger a_l a_m a_n$$

- Quantum Phase Estimation uses this to compute eigenvalues exponentially faster than with (exact) classical methods
  - we can use this to prepare eigenstates
  - we can use this to compute cross sections
  - ...
- Quantum Computers are even better at evolving using the qubiterate

$$W_Q \approx \exp\left(i \arccos\left(\frac{H}{\alpha}\right)\right)$$

  - we can use this to do time evolution faster: $\mathcal{O}\left(\alpha t + \log(1/\epsilon)\right)$
  - we can use this to compute cross sections faster
  - ...

What have we learned in these couple days

- Quantum Computers are good at simulating time evolution for Hamiltonians with 2 and 3-body interactions (and possibly others)

$$H = \sum_{ij} K_{ij} a_i^\dagger a_j + \sum_{ijkl} U_{ijkl} a_i^\dagger a_j^\dagger a_k a_l + \sum_{ijklmn} V_{ijklmn} a_i^\dagger a_j^\dagger a_k^\dagger a_l a_m a_n$$

- Quantum Phase Estimation uses this to compute eigenvalues exponentially faster than with (exact) classical methods
  - we can use this to prepare eigenstates
  - we can use this to compute cross sections
  - ...
- Quantum Computers are even better at evolving using the qubiterate

$$W_Q \approx \exp\left(i \arccos\left(\frac{H}{\alpha}\right)\right)$$

  - we can use this to do time evolution faster: $\mathcal{O}\left(\alpha t + \log(1/\epsilon)\right)$
  - we can use this to compute cross sections faster
  - ...

We will need to reduce gate counts considerably!