Fermion determinants on quantum computers

2407.13080

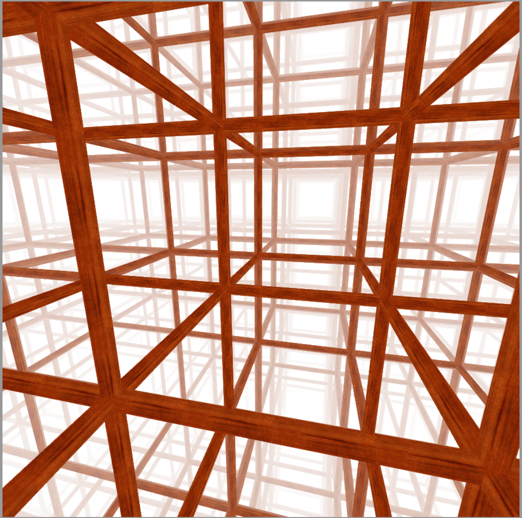
George Fleming Prasanth Shyamsundar **Judah Unmuth-Yockey**

Theory Division, Fermi National Accelerator Laboratory, IL

November 15, 2024

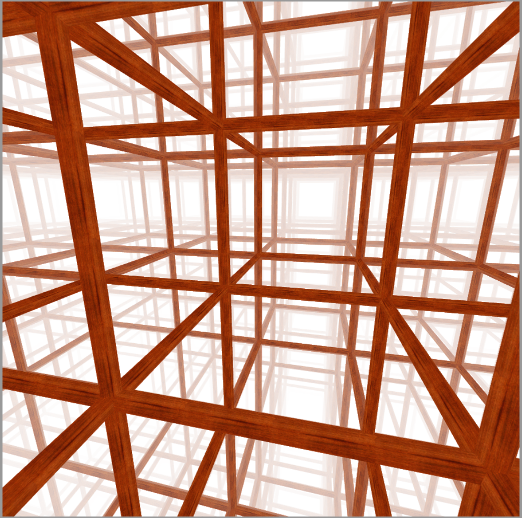
Introduction

Lattice field theory



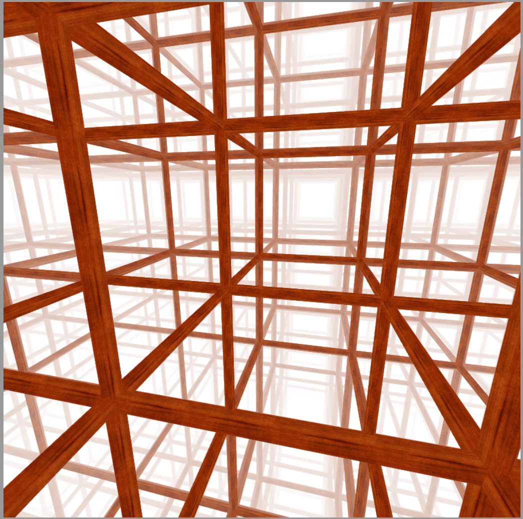
Lattice field theory

- Lattice QCD



Lattice field theory

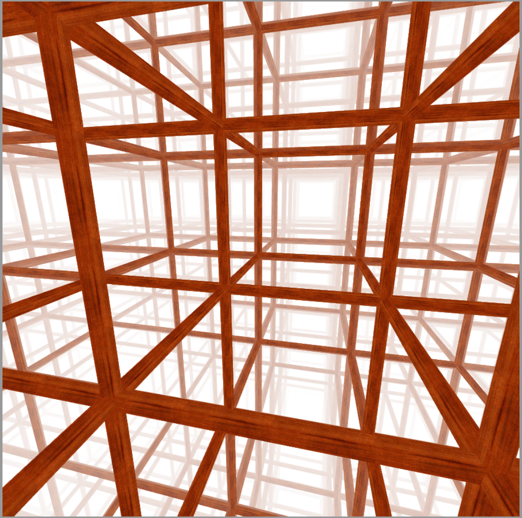
- Lattice QCD
- Fermion fields



Lattice field theory

- Lattice QCD
- Fermion fields

  $\rightarrow$ Grassmann variables

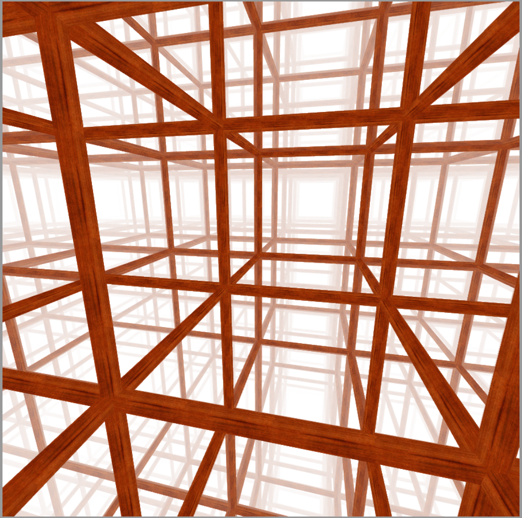


Lattice field theory

- Lattice QCD
- Fermion fields

  $\rightarrow$ Grassmann variables
- Effective action

$$S_{\text{eff}} = -\log\det(W)$$



Dealing with fermions

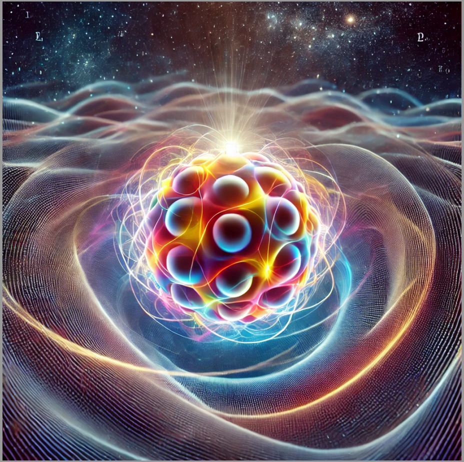


Figure: A fermion from the mind of AI

Dealing with fermions

- Monte Carlo Sampling

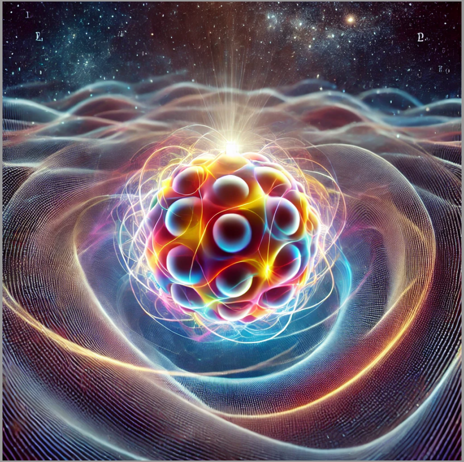


Figure: A fermion from the mind of AI

Dealing with fermions

- Monte Carlo Sampling
- $S_{\text{eff}} \sim O(V^3)$

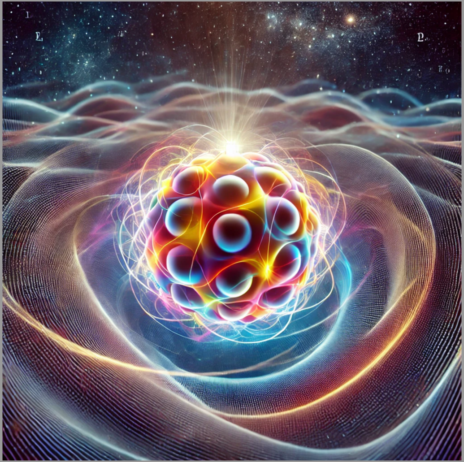


Figure: A fermion from the mind of AI

Dealing with fermions

- Monte Carlo Sampling
- $S_{\text{eff}} \sim O(V^3)$

  $\rightarrow$ pseudofermion method

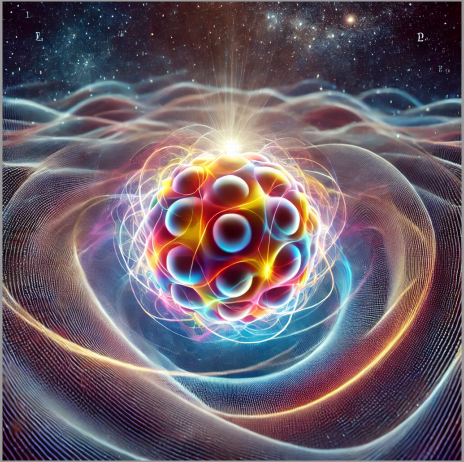


Figure: A fermion from the mind of AI

Dealing with fermions

- Monte Carlo Sampling
- $S_{\text{eff}} \sim O(V^3)$

  $\rightarrow$ pseudofermion method
- Improved cost

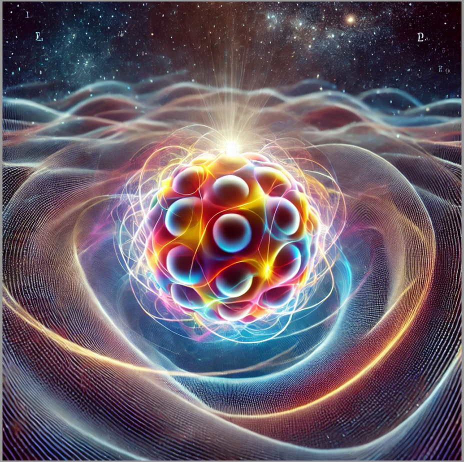


Figure: A fermion from the mind of AI

Dealing with fermions

- Monte Carlo Sampling
- $S_{\text{eff}} \sim O(V^3)$

  $\rightarrow$ pseudofermion method
- Improved cost

  $\rightarrow$ matrix inversion (with tricks)

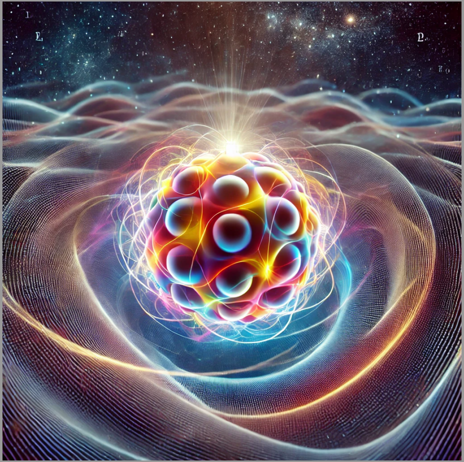


Figure: A fermion from the mind of AI

Dealing with fermions

- Monte Carlo Sampling
- $S_{\text{eff}} \sim O(V^3)$

  $\rightarrow$ pseudofermion method
- Improved cost

  $\rightarrow$ matrix inversion (with tricks)
- Severe sign problem

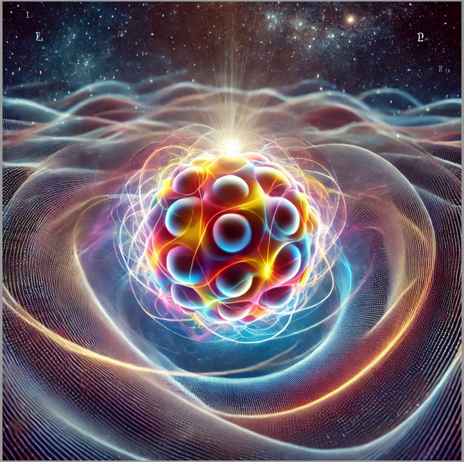


Figure: A fermion from the mind of AI

Use quantum computers?

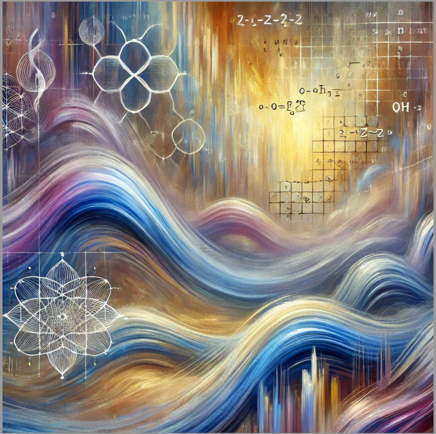


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)

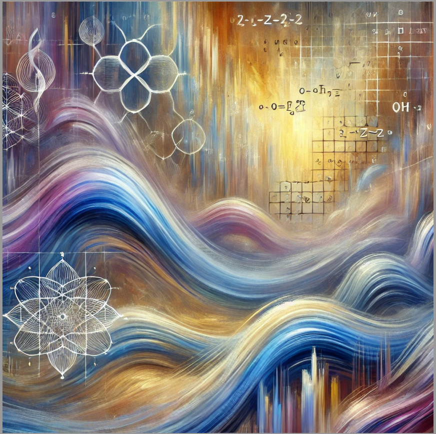


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)
- Matrix arithmetic (Qubitization/Quantum signal processing)

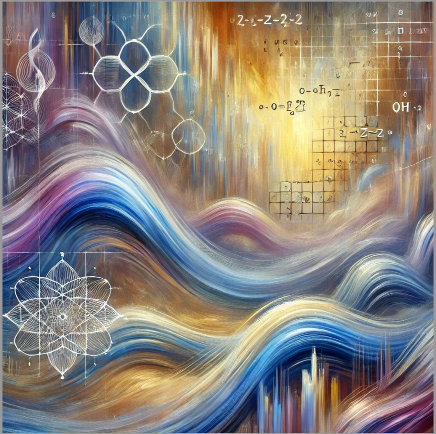


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)
- Matrix arithmetic (Qubitization/Quantum signal processing)
- Sums/means (Quantum phase estimation)

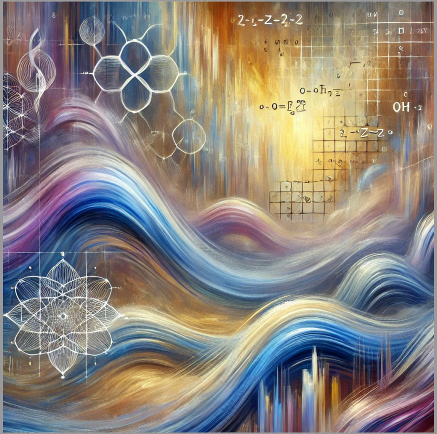


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)
- Matrix arithmetic (Qubitization/Quantum signal processing)
- Sums/means (Quantum phase estimation)
- Real-time dynamics (Hamiltonian simulation)

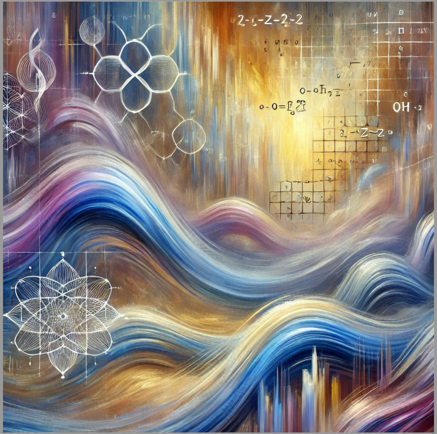


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)
- **Matrix arithmetic (Qubitization/Quantum signal processing)**
- Sums/means (Quantum phase estimation)
- Real-time dynamics (Hamiltonian simulation)

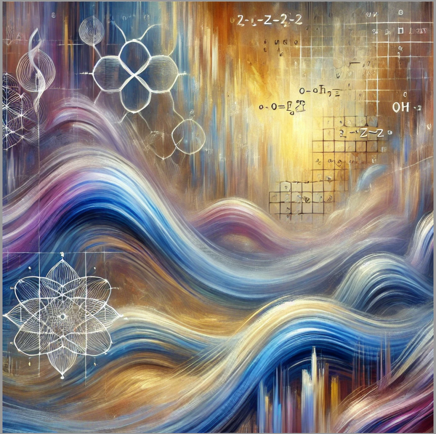


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)
- **Matrix arithmetic (Qubitization/Quantum signal processing)**
- **Sums/means (Quantum phase estimation)**
- Real-time dynamics (Hamiltonian simulation)

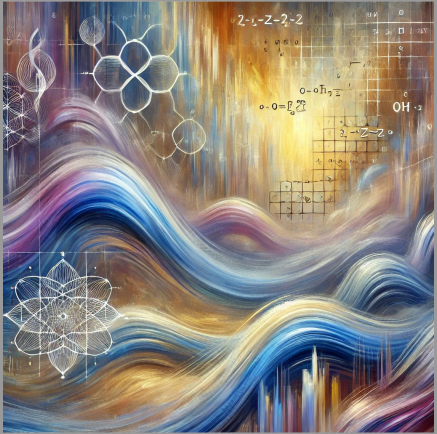


Figure: Quantum Fourier transform from the mind of AI

Use quantum computers?

- Fourier transform (Quantum fourier transform)
- **Matrix arithmetic (Qubitization/Quantum signal processing)**
- **Sums/means (Quantum phase estimation)**
- Real-time dynamics (Hamiltonian simulation)
- $O(V\log(V)) < O(V^3)$

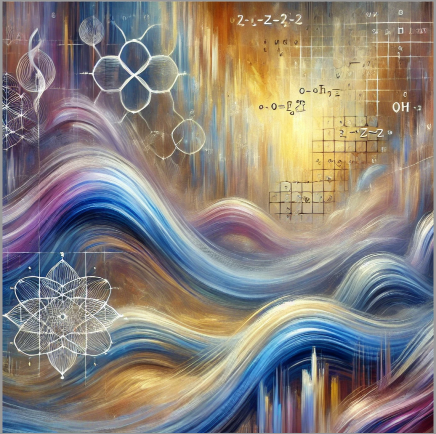
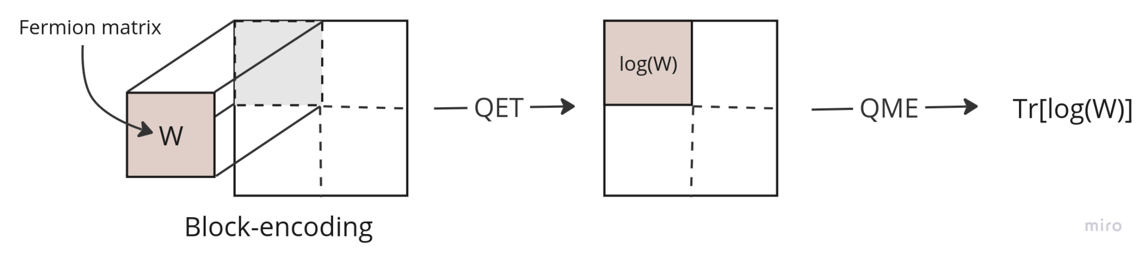


Figure: Quantum Fourier transform from the mind of AI

How are we going to do it?



Block encoding

What is block encoding?

Some **nonunitary** matrix, W. Some larger unitary matrix, U

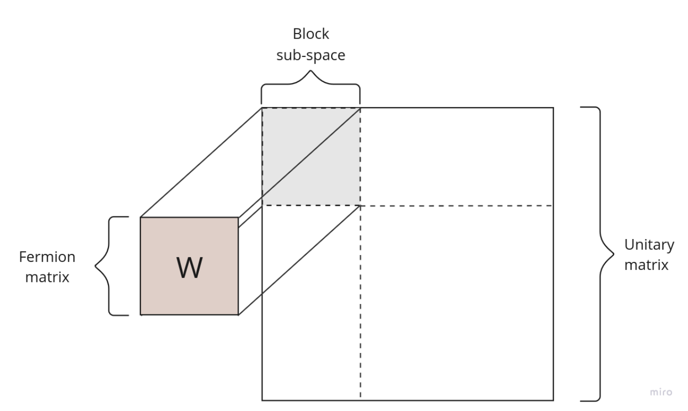
$$\langle 0_{\text{anc}}| \langle n| U |0_{\text{anc}}\rangle |m\rangle = W_{nm}$$

What is block encoding?

Some nonunitary matrix, W. Some larger **unitary** matrix, U

$$\langle 0_{\mathrm{anc}}| \langle n| U |0_{\mathrm{anc}}\rangle |m\rangle = W_{nm}$$

Block encoding visualized



$$\langle 0_{\text{anc}}|\, \langle n|\, U\, |0_{\text{anc}}\rangle\, |m\rangle = W_{nm}$$

Is this practically possible?

[Camps et al., 2023]

Is this practically possible?

Yes.

[Camps et al., 2023]

Is this practically possible?

Yes.

1. Fermion matrix is sparse: $O(V)$ entries.

[Camps et al., 2023]

Is this practically possible?

Yes.

1. Fermion matrix is **sparse**: $O(V)$ entries.

$$
\begin{aligned}
M_{mn} = \frac{K}{2} \sum_{\mu} & \eta_\mu(m) U_\mu(m) \delta_{n,m+\hat{\mu}} \\
& - \eta_\mu(m) U^*_\mu(m-\hat{\mu}) \delta_{n,m-\hat{\mu}} \\
& + m_0 \delta_{n,m}.
\end{aligned}
$$

[Camps et al., 2023]

Is this practically possible?

Yes.

1. Fermion matrix is **sparse**: $O(V)$ entries.
2. Small number of interactions.

$$M_{mn} = \frac{K}{2}\sum_{\mu} \eta_\mu(m) U_\mu(m) \delta_{n,m+\hat{\mu}} - \eta_\mu(m) U^*_\mu(m-\hat{\mu}) \delta_{n,m-\hat{\mu}} + m_0 \delta_{n,m}.$$

[Camps et al., 2023]

Is this practically possible?

Yes.

1. Fermion matrix is **sparse**: $O(V)$ entries.
2. Small number of interactions.
3. Parameterize with locality.

$$M_{mn} = \frac{K}{2}\sum_{\mu} \eta_\mu(m)U_\mu(m)\delta_{n,m+\hat{\mu}} - \eta_\mu(m)U_\mu^*(m-\hat{\mu})\delta_{n,m-\hat{\mu}} + m_0\delta_{n,m}.$$

[Camps et al., 2023]

Is this practically possible?

Yes.

1. Fermion matrix is **sparse**: $O(V)$ entries.
2. Small number of interactions.
3. Parameterize with locality.
4. Insert matrix elements one-by-one.

[Camps et al., 2023]

$$
\begin{aligned}
M_{mn} = \frac{K}{2}\sum_{\mu} &\eta_\mu(m)U_\mu(m)\delta_{n,m+\hat{\mu}} \\
&- \eta_\mu(m)U_\mu^*(m-\hat{\mu})\delta_{n,m-\hat{\mu}} \\
&+ m_0\delta_{n,m}.
\end{aligned}
$$

How do we block encode?

How do we block encode?

$$|n\rangle \leftrightarrow \underbrace{|n+\hat{\mu}\rangle}_{\text{neighbors}}$$

How do we block encode?

$$|n\rangle \leftrightarrow \underbrace{|n+\hat{\mu}\rangle}_{\text{neighbors}}$$

- Enumerate neighbors (nonzero row entries), ℓ

How do we block encode?

$$|n\rangle \leftrightarrow \underbrace{|n+\hat{\mu}\rangle}_{\text{neighbors}}$$

- Enumerate neighbors (nonzero row entries), ℓ

$$\begin{cases} |n+\hat{x}\rangle & \ell = 0 \\ \vdots & \\ |n-\hat{t}\rangle & \ell = 7 \\ |n\rangle & \ell = 8 \end{cases}$$

Encode sparse structure

Encode sparse structure

e.g.:
$|n+\hat{y}\rangle \mapsto |1\rangle\,|n\rangle$
$|n-\hat{t}\rangle \mapsto |7\rangle\,|n\rangle$

Encode sparse structure

e.g.:
$|n+\hat{y}\rangle \mapsto |1\rangle\,|n\rangle$
$|n-\hat{t}\rangle \mapsto |7\rangle\,|n\rangle$

Nonzero entries in column n

Encode sparse structure

e.g.:
$|n + \hat{y}\rangle \mapsto |1\rangle\,|n\rangle$
$|n - \hat{t}\rangle \mapsto |7\rangle\,|n\rangle$

Nonzero entries in column n
Set-up:

$$\underbrace{|0\rangle}_{\text{projects block}} \quad \overbrace{|\ell\rangle}^{\text{nonzero elements}} \quad \underbrace{|n\rangle}_{\text{location}}$$

Preparing $|\ell\rangle$

Preparing $|\ell\rangle$

$$|0\rangle \underbrace{|\ell\rangle} |n\rangle$$

Preparing $|\ell\rangle$

$$|0\rangle \underbrace{|\ell\rangle} |n\rangle$$

Equal super-position:

Preparing $|\ell\rangle$

$|0\rangle \underbrace{|\ell\rangle} |n\rangle$

Equal super-position:

$$H|0/1\rangle = |+/-\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

Preparing $|\ell\rangle$

$$|0\rangle \underbrace{|\ell\rangle} |n\rangle$$

Equal super-position:

$$H|0/1\rangle = |+/-\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

$$\underbrace{H \otimes \cdots \otimes H}_{k} |\underbrace{0 \cdots 0}_{k}\rangle = |+\cdots+\rangle = \frac{1}{2^{k/2}}\left(|0\cdots\rangle + \cdots + |1\cdots\rangle\right)$$

Preparing $|\ell\rangle$

$$|0\rangle \underbrace{|\ell\rangle} |n\rangle$$

Equal super-position:

$$H|0/1\rangle = |+/-\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

$$\underbrace{H \otimes \cdots \otimes H}_{k} |\underbrace{0 \cdots 0}_{k}\rangle = |+ \cdots +\rangle = \frac{1}{2^{k/2}}\left(|0 \cdots\rangle + \cdots + |1 \cdots\rangle\right)$$

$$D_s \equiv H \otimes \cdots \otimes H$$

Map of locality

Map of locality

Design a unitary matrix O_c

$$O_c |0\rangle |\ell\rangle |n\rangle = |0\rangle |\ell\rangle |n + \hat{\mu}(\ell)\rangle$$

Insert non-zero values

Insert non-zero values

Design a unitary matrix O_A

$$O_A |0\rangle |\ell\rangle |n\rangle = \left(W_{c(n,\ell)n} |0\rangle + \sqrt{1 - |W_{c(n,\ell)n}|^2} |1\rangle \right) |\ell\rangle |n\rangle$$

Work-flow

$O_c \, O_A \, D_s \, |0\rangle \, |0\rangle \, |n\rangle$

Work-flow

$$O_c O_A D_s |0\rangle |0\rangle |n\rangle = O_c O_A |0\rangle \frac{1}{\sqrt{N}} \left[|0\rangle + \cdots + |N-1\rangle \right] |n\rangle$$

Work-flow

$$O_c O_A D_s |0\rangle |0\rangle |n\rangle = O_c O_A |0\rangle \frac{1}{\sqrt{N}} [|0\rangle + \cdots + |N-1\rangle] |n\rangle$$

$$= O_c \frac{1}{\sqrt{N}} [O_A |0\rangle |0\rangle + \cdots + O_A |0\rangle |N-1\rangle] |n\rangle$$

Work-flow

$$
\begin{aligned}
O_c O_A D_s |0\rangle |0\rangle |n\rangle &= O_c O_A |0\rangle \frac{1}{\sqrt{N}} [|0\rangle + \cdots + |N-1\rangle] |n\rangle \\
&= O_c \frac{1}{\sqrt{N}} [O_A |0\rangle |0\rangle + \cdots + O_A |0\rangle |N-1\rangle] |n\rangle \\
&= O_c \frac{1}{\sqrt{N}} \left[\left(W_{c(0,n)n} |0\rangle + \sqrt{1 - |W_{c(0,n)n}|^2} |1\rangle \right) |0\rangle + \cdots \right] |n\rangle
\end{aligned}
$$

Work-flow

$$
\begin{aligned}
O_c O_A D_s |0\rangle |0\rangle |n\rangle &= O_c O_A |0\rangle \frac{1}{\sqrt{N}} \left[|0\rangle + \cdots + |N-1\rangle\right] |n\rangle \\
&= O_c \frac{1}{\sqrt{N}} \left[O_A |0\rangle |0\rangle + \cdots + O_A |0\rangle |N-1\rangle\right] |n\rangle \\
&= O_c \frac{1}{\sqrt{N}} \left[\left(W_{c(0,n)n} |0\rangle + \sqrt{1 - |W_{c(0,n)n}|^2} |1\rangle\right) |0\rangle + \cdots\right] |n\rangle \\
&= \frac{1}{\sqrt{N}} \left[\left(W_{c(0,n)n} |0\rangle + \sqrt{1 - |W_{c(0,n)n}|^2} |1\rangle\right) |0\rangle |n+\hat{\mu}(0)\rangle + \cdots\right]
\end{aligned}
$$

Work-flow

$$\left(\langle 0|\,\langle 0|\,\langle m|\,D_s^\dagger\right)\left(O_c O_A D_s\,|0\rangle\,|0\rangle\,|n\rangle\right)$$

Work-flow

$$\left(\langle 0|\,\langle 0|\,\langle m|\,D_s^\dagger\right)\left(O_c O_A D_s\,|0\rangle\,|0\rangle\,|n\rangle\right)$$

$$= \left(\langle 0|\,\langle 0|\,\langle m|\,D_s^\dagger\right)\frac{1}{\sqrt{N}}\left[\left(W_{c(0,n)n}\,|0\rangle + \sqrt{1-|W_{c(0,n)n}|^2}\,|1\rangle\right)|0\rangle\,|n+\hat{\mu}(0)\rangle + \cdots +\right]$$

Work-flow

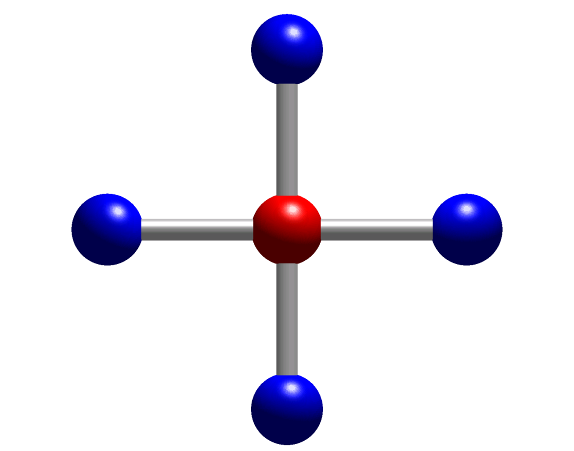
$$\left(\langle 0|\,\langle 0|\,\langle m|\,D_s^\dagger\right)\left(O_c O_A D_s\,|0\rangle\,|0\rangle\,|n\rangle\right)$$

$$= \left(\langle 0|\,\langle 0|\,\langle m|\,D_s^\dagger\right)\frac{1}{\sqrt{N}}\left[\left(W_{c(0,n)n}\,|0\rangle + \sqrt{1-|W_{c(0,n)n}|^2}\,|1\rangle\right)|0\rangle\,|n+\hat{\mu}(0)\rangle + \cdots +\right]$$

$$B \equiv D_s O_c O_A D_s$$

Example: scalar Laplacian

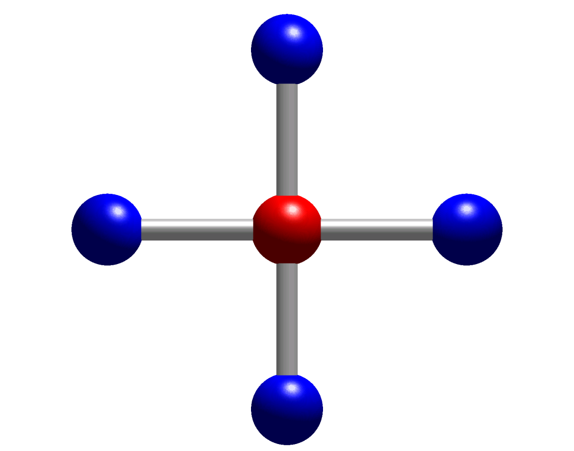
Two dimensions, Δ_{xy}/s:



Example: scalar Laplacian

Two dimensions, Δ_{xy}/s:

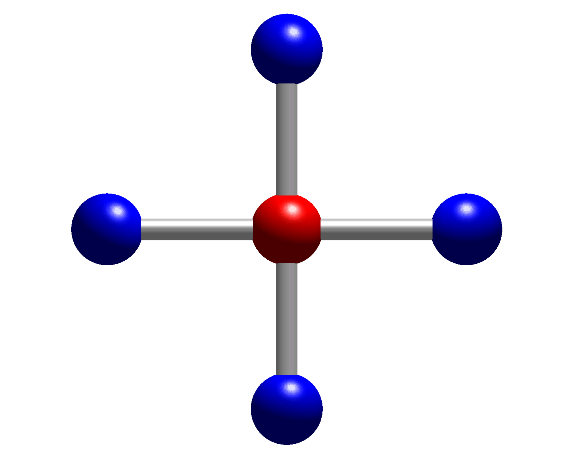
- Four nearest-neighbors + diagonal



Example: scalar Laplacian

Two dimensions, Δ_{xy}/s:

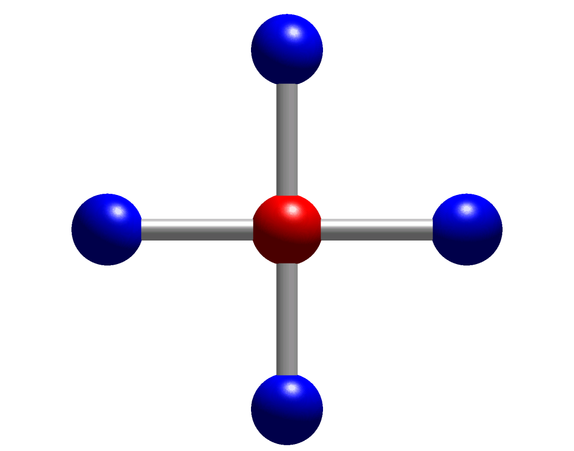
- Four nearest-neighbors + diagonal
- $(4 + m_0^2)/s$ along the diagonal



Example: scalar Laplacian

Two dimensions, Δ_{xy}/s:

- Four nearest-neighbors + diagonal
- $(4 + m_0^2)/s$ along the diagonal
- $-1/s$ on the off-diagonal at nearest neighbors



Example: scalar Laplacian

Example: scalar Laplacian

$$c(n,\ell) = \begin{cases} \ell = 0: & |n+\hat{x}\rangle \\ \ell = 1: & |n+\hat{y}\rangle \\ \ell = 2: & |n-\hat{x}\rangle \\ \ell = 3: & |n-\hat{y}\rangle \\ 4 \le \ell < 8: & |n\rangle \end{cases}$$

Example: scalar Laplacian

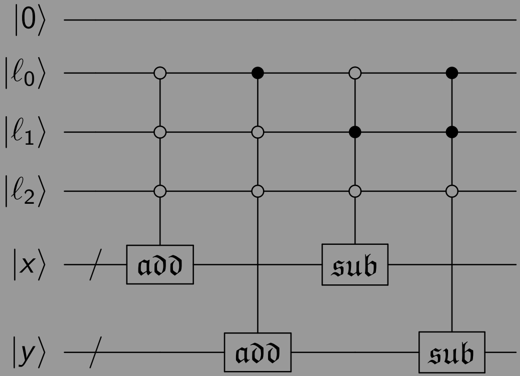
$$c(n,\ell) = \begin{cases} \ell = 0 : & |n+\hat{x}\rangle \\ \ell = 1 : & |n+\hat{y}\rangle \\ \ell = 2 : & |n-\hat{x}\rangle \\ \ell = 3 : & |n-\hat{y}\rangle \\ 4 \leq \ell < 8 : & |n\rangle \end{cases}$$

$$O_c\,|0\rangle\,|0\rangle\,|n\rangle = |0\rangle\,|0\rangle\,|n+\hat{x}\rangle$$
$$O_c\,|0\rangle\,|1\rangle\,|n\rangle = |0\rangle\,|1\rangle\,|n+\hat{y}\rangle$$
$$O_c\,|0\rangle\,|2\rangle\,|n\rangle = |0\rangle\,|2\rangle\,|n-\hat{x}\rangle$$
$$O_c\,|0\rangle\,|3\rangle\,|n\rangle = |0\rangle\,|3\rangle\,|n-\hat{y}\rangle$$

Example: scalar Laplacian

$$c(n,\ell) = \begin{cases} \ell = 0: & |n+\hat{x}\rangle \\ \ell = 1: & |n+\hat{y}\rangle \\ \ell = 2: & |n-\hat{x}\rangle \\ \ell = 3: & |n-\hat{y}\rangle \\ 4 \le \ell < 8: & |n\rangle \end{cases}$$

$$O_c\,|0\rangle\,|0\rangle\,|n\rangle = |0\rangle\,|0\rangle\,|n+\hat{x}\rangle$$
$$O_c\,|0\rangle\,|1\rangle\,|n\rangle = |0\rangle\,|1\rangle\,|n+\hat{y}\rangle$$
$$O_c\,|0\rangle\,|2\rangle\,|n\rangle = |0\rangle\,|2\rangle\,|n-\hat{x}\rangle$$
$$O_c\,|0\rangle\,|3\rangle\,|n\rangle = |0\rangle\,|3\rangle\,|n-\hat{y}\rangle$$



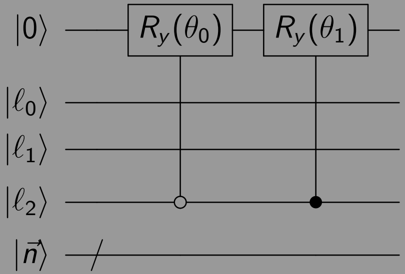
Example: scalar Laplacian

Example: scalar Laplacian

$$O_A |0\rangle |\ell\rangle |n\rangle = \begin{cases} \left(-1/s |0\rangle + \sqrt{1 - |1/s|^2} |1\rangle\right) |\ell\rangle |n\rangle & 0 \le \ell \le 3 \\ \left((4 + m_0^2)/s |0\rangle + \sqrt{1 - |(4 + m_0^2)/s|^2} |1\rangle\right) |\ell\rangle |n\rangle & 4 \le \ell < 8 \end{cases}$$

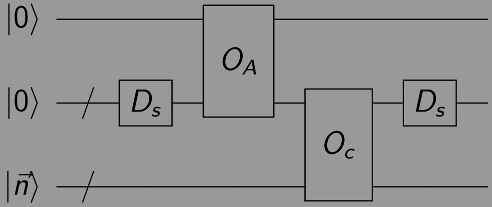
Example: scalar Laplacian

$$O_A |0\rangle |\ell\rangle |n\rangle = \begin{cases} \left(-1/s\,|0\rangle + \sqrt{1-|1/s|^2}\,|1\rangle\right)|\ell\rangle\,|n\rangle & 0 \le \ell \le 3 \\ \left((4+m_0^2)/s\,|0\rangle + \sqrt{1-|(4+m_0^2)/s|^2}\,|1\rangle\right)|\ell\rangle\,|n\rangle & 4 \le \ell < 8 \end{cases}$$



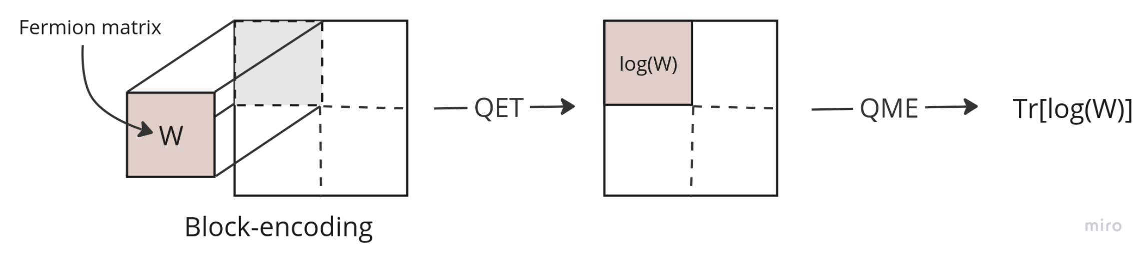
Example: scalar Laplacian

Example: scalar Laplacian



Matrix log

Matrix logarithm



Quantum eigenvalue transform (QET)

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum eigenvalue transform (QET)

- A **unitary** transformation

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum eigenvalue transform (QET)

- A **unitary** transformation
- Coherent

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum eigenvalue transform (QET)

- A **unitary** transformation
- Coherent
- **A polynomial transform** of the eigenvalues

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum eigenvalue transform (QET)

- A **unitary** transformation
- Coherent
- **A polynomial transform** of the eigenvalues
- **No knowlege** of the eigenvalues is necessary

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum eigenvalue transform (QET)

- A **unitary** transformation
- Coherent
- **A polynomial transform** of the eigenvalues
- **No knowlege** of the eigenvalues is necessary

$$\begin{pmatrix} W & \cdots \\ \vdots & \ddots \end{pmatrix} \longrightarrow \begin{pmatrix} f(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum eigenvalue transform (QET)

- A **unitary** transformation
- Coherent
- **A polynomial transform** of the eigenvalues
- **No knowlege** of the eigenvalues is necessary

$$\begin{pmatrix} W & \cdots \\ \vdots & \ddots \end{pmatrix} \longrightarrow \begin{pmatrix} f(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

How does it do it?

[Low and Chuang, 2019, Gilyén et al., 2019, Martyn et al., 2021]

Quantum signal processing (QSP)

Quantum signal processing (QSP)

$$R(w) = e^{i \arccos(w) X}$$

Quantum signal processing (QSP)

$$R(w) = e^{i\arccos(w)X} = \begin{pmatrix} w & i\sqrt{1-w^2} \\ i\sqrt{1-w^2} & w \end{pmatrix}.$$

Quantum signal processing (QSP)

$$R(w) = e^{i \arccos(w) X} = \begin{pmatrix} w & i\sqrt{1-w^2} \\ i\sqrt{1-w^2} & w \end{pmatrix}.$$

This is a **block-encoding** of $w \in [-1, 1]$.

Quantum signal processing (QSP)

$$R(w) = e^{i\,\mathrm{arccos}(w)X} = \begin{pmatrix} w & i\sqrt{1-w^2} \\ i\sqrt{1-w^2} & w \end{pmatrix}.$$

This is a **block-encoding** of $w \in [-1, 1]$.

$$\prod_{n=1}^{d} R(w)$$

Quantum signal processing (QSP)

$$R(w) = e^{i\,\mathrm{arccos}(w) X} = \begin{pmatrix} w & i\sqrt{1-w^2} \\ i\sqrt{1-w^2} & w \end{pmatrix}.$$

This is a **block-encoding** of $w \in [-1, 1]$.

$$\prod_{n=1}^{d} R(w) = \begin{pmatrix} T_d(w) & \cdot \\ \cdot & \cdot \end{pmatrix}$$

Quantum signal processing (QSP)

Quantum signal processing (QSP)

$$U_\Phi = e^{i\phi_0 Z} \prod_{n=1}^{d} R(w) e^{i\phi_n Z}$$

Quantum signal processing (QSP)

$$
\begin{aligned}
U_\Phi &= e^{i\phi_0 Z} \prod_{n=1}^{d} R(w) e^{i\phi_n Z} \\
&= \begin{pmatrix} P(w) & iQ(w)\sqrt{1-w^2} \\ iQ(w)\sqrt{1-w^2} & P^*(w) \end{pmatrix},
\end{aligned}
$$

QSP conditions

QSP conditions

(1) $\deg(P) \leq d$, $\deg(Q) \leq (d-1)$,

QSP conditions

(1) $\deg(P) \leq d$, $\deg(Q) \leq (d-1)$,

(2) P has parity $d \mod 2$, and Q has parity $(d-1) \mod 2$,

QSP conditions

(1) $\deg(P) \leq d$, $\deg(Q) \leq (d-1)$,

(2) P has parity $d \mod 2$, and Q has parity $(d-1) \mod 2$,

(3) $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1,\ \forall x \in [-1,1]$.

QSP conditions

(1) $\deg(P) \leq d$, $\deg(Q) \leq (d-1)$,

(2) P has parity $d \mod 2$, and Q has parity $(d-1) \mod 2$,

(3) $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1, \forall x \in [-1,1]$.

How does this help us?

Block-encoding

Block-encoding

$$R(w) = e^{i\,\mathrm{arccos}(w) X} = \begin{pmatrix} w & i\sqrt{1-w^2} \\ i\sqrt{1-w^2} & w \end{pmatrix}.$$

Block-encoding

$$R(w) = e^{i\,\mathrm{arccos}(w)X} = \begin{pmatrix} w & i\sqrt{1-w^2} \\ i\sqrt{1-w^2} & w \end{pmatrix}.$$

$$B = \begin{array}{cc} & \langle 0_{\mathrm{anc}}| \\ |0_{\mathrm{anc}}\rangle & \begin{pmatrix} W & \cdots \\ \vdots & \ddots \end{pmatrix} \end{array}$$

Projector

Projector

$$R_z(\phi) = e^{i\phi Z} \qquad Z = 2\left|0\right\rangle\left\langle 0\right| - \mathbb{1}$$

Projector

$$R_z(\phi) = e^{i\phi Z} \qquad Z = 2\left|0\right\rangle\left\langle 0\right| - \mathbb{1}$$

$$U_\Pi = e^{i\phi\Pi} \qquad \Pi = 2\left|0_{\text{anc}}\right\rangle\left\langle 0_{\text{anc}}\right| \otimes \mathbb{1} - \mathbb{1}$$

Quantum eigenvalue transform

Quantum eigenvalue transform

$$\mathrm{QET} = U_\Pi(\phi_0) \prod_{i=1}^{d} B\, U_\Pi(\phi_i)$$

Quantum eigenvalue transform

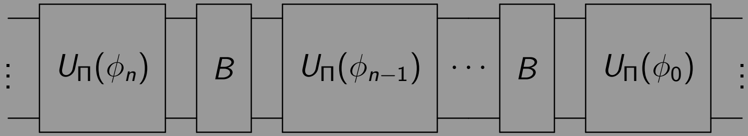
$$\mathrm{QET} = U_\Pi(\phi_0) \prod_{i=1}^{d} B\, U_\Pi(\phi_i)$$

$$\to \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

Quantum eigenvalue transform

$$\mathrm{QET} = U_\Pi(\phi_0) \prod_{i=1}^{d} B\, U_\Pi(\phi_i)$$

$$\rightarrow \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$



Choose a polynomial

Choose a polynomial

$$U_\Pi(\phi_0) \prod_{i=1}^{d} B\, U_\Pi(\phi_i) = \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

Choose a polynomial

$$U_\Pi(\phi_0) \prod_{i=1}^{d} B \, U_\Pi(\phi_i) = \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

- $f(x) \equiv \log(x)$

Choose a polynomial

$$U_\Pi(\phi_0)\prod_{i=1}^{d} B\, U_\Pi(\phi_i) = \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

- $f(x) \equiv \log(x)$
  - $\longrightarrow$ there are caveats here so that QSP works

Choose a polynomial

$$U_\Pi(\phi_0)\prod_{i=1}^{d} B\, U_\Pi(\phi_i) = \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

- $f(x) \equiv \log(x)$
  - $\longrightarrow$ there are caveats here so that QSP works
- want $P(x) \approx f(x)$

Choose a polynomial

$$U_\Pi(\phi_0)\prod_{i=1}^{d} B\, U_\Pi(\phi_i) = \begin{pmatrix} P(W) & \cdots \\ \vdots & \ddots \end{pmatrix}$$

- $f(x) \equiv \log(x)$
  - $\longrightarrow$ there are caveats here so that QSP works
- want $P(x) \approx f(x)$

$$P(x) = \sum_{k=0}^{d} a_k T_k(x) \approx f(x)$$

Solve for the phases

Solve for the phases

We used **least-squares** [Dong et al., 2021]

Solve for the phases

We used **least-squares** [Dong et al., 2021]

$$L(\{\phi\}) = \frac{1}{\tilde{d}} \sum_{j=1}^{\tilde{d}} \left| \langle 0| \, e^{i\phi_0 Z} \prod_{n=1}^{d} R(x_j) e^{i\phi_n Z} \, |0\rangle - P(x_j) \right|^2$$

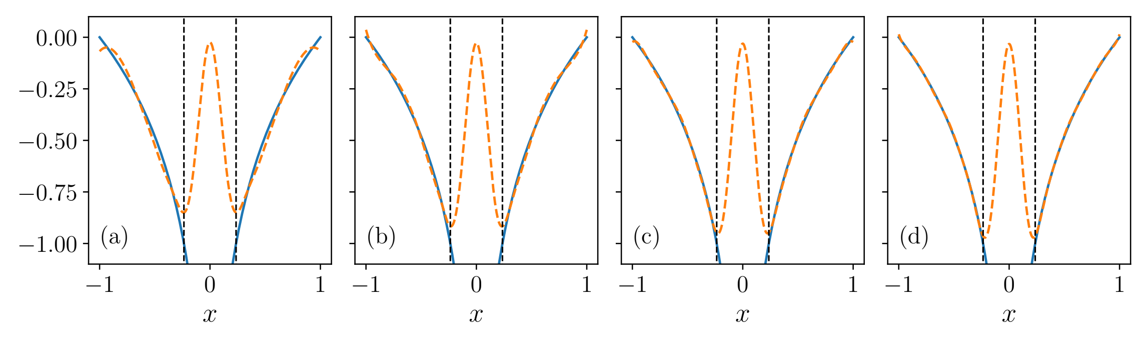
Solve for the phases

We used **least-squares** [Dong et al., 2021]

$$L(\{\phi\}) = \frac{1}{\tilde{d}} \sum_{j=1}^{\tilde{d}} \left| \langle 0 | \, e^{i\phi_0 Z} \prod_{n=1}^{d} R(x_j) e^{i\phi_n Z} \, | 0 \rangle - P(x_j) \right|^2$$

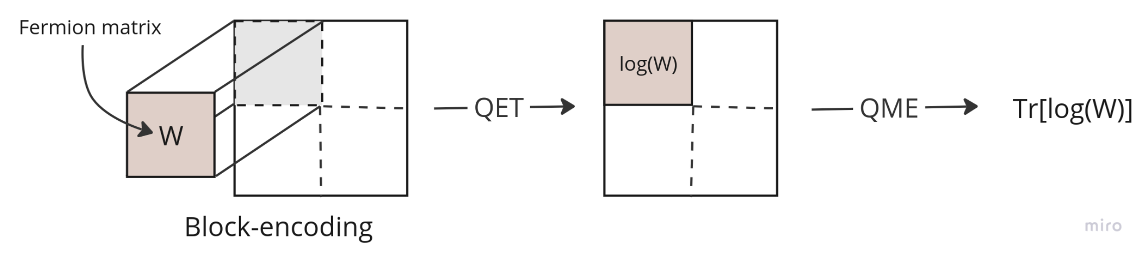
Minimize L w.r.t. $\{\phi\}$.

Polynomial approximation



Matrix trace

Workflow



Quantum mean estimation

[Shyamsundar, 2021, Gustafson et al., 2023, Montanaro, 2015, Hamoudi, 2021]

Quantum mean estimation

- Compute normalized sums (means)

[Shyamsundar, 2021, Gustafson et al., 2023, Montanaro, 2015, Hamoudi, 2021]

Quantum mean estimation

- Compute normalized sums (means)
- **Quadradically faster**

[Shyamsundar, 2021, Gustafson et al., 2023, Montanaro, 2015, Hamoudi, 2021]

Quantum mean estimation

- Compute normalized sums (means)
- **Quadradically faster**
- Uses **quantum phase estimation** (QPE)

[Shyamsundar, 2021, Gustafson et al., 2023, Montanaro, 2015, Hamoudi, 2021]

What is QPE?

Quantum phase estimation

Given a unitary matrix $\mathcal{Q}$, and an eigenvector of $\mathcal{Q}$, $|q\rangle$, QPE returns the corresponding eigenvalue of $|q\rangle$: $e^{i\phi_q}$.

$$\mathcal{Q}\,|q\rangle = e^{i\phi_q}\,|q\rangle$$

How does QPE work?

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}} \sum_{m=0}^{M-1} |m\rangle$$

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}}\sum_{m=0}^{M-1}|m\rangle = \frac{1}{2^{M/2}}(|0\rangle + |1\rangle + |2\rangle + \dots)$$

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}} \left(|0\rangle + |1\rangle + |2\rangle + \ldots \right)$$

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}}\left(|0\rangle + |1\rangle + |2\rangle + \ldots\right) \rightarrow \frac{1}{2^{M/2}}\left(e^{2\pi i(0)\bar{x}/M}|0\rangle + |1\rangle + \ldots\right)$$

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}}\left(|0\rangle + |1\rangle + |2\rangle + \ldots\right) \rightarrow \frac{1}{2^{M/2}}\left(e^{2\pi i(0)\bar{x}/M}|0\rangle + e^{2\pi i(1)\bar{x}/M}|1\rangle + \ldots\right)$$

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}}\left(|0\rangle + |1\rangle + |2\rangle + \ldots\right) \to \frac{1}{2^{M/2}}\left(e^{2\pi i(0)\bar{x}/M}|0\rangle + e^{2\pi i(1)\bar{x}/M}|1\rangle + e^{2\pi i(2)\bar{x}/M}|2\rangle + \ldots\right)$$

How does QPE work?

Step 1) Build Fourier series

$$\frac{1}{2^{M/2}}\sum_{m=0}^{M-1} e^{2\pi i m\bar{x}/M}\,|m\rangle$$

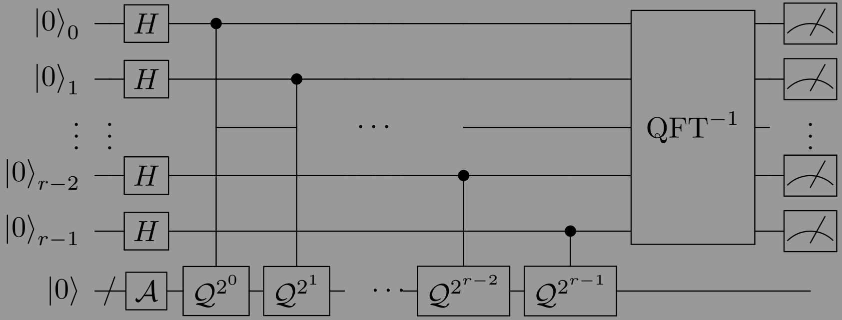
How does QPE work?

Step 2) Inverse Fourier transform

$$\mathrm{QFT}^{-1}\left[\frac{1}{2^{M/2}}\sum_{m=0}^{M-1} e^{2\pi i m\bar{x}/M}\left|m\right\rangle\right] = \left|\bar{x}\right\rangle$$

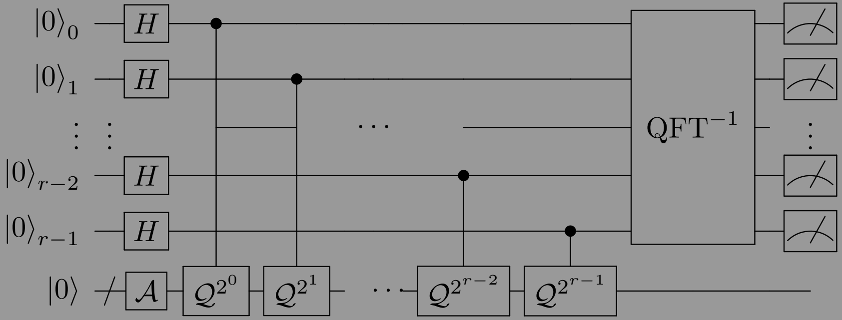
QPE to QME

$$\mathcal{A}\,|0\rangle = |q\rangle, \qquad \mathcal{Q}\,|q\rangle = e^{i\phi_q}\,|q\rangle$$



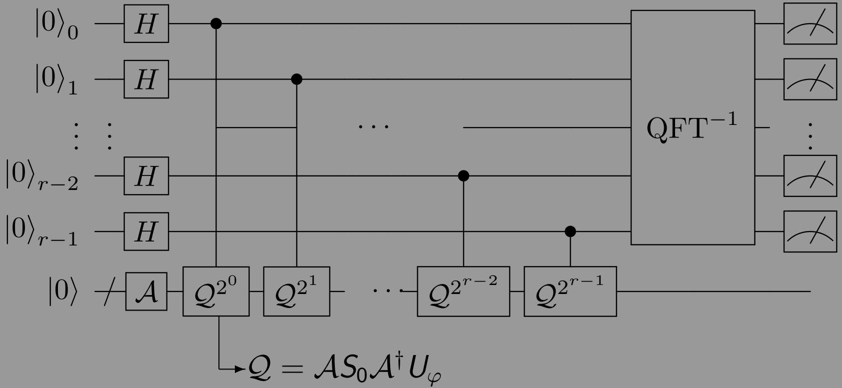
QPE to QME

How does **mean** estimation enter?



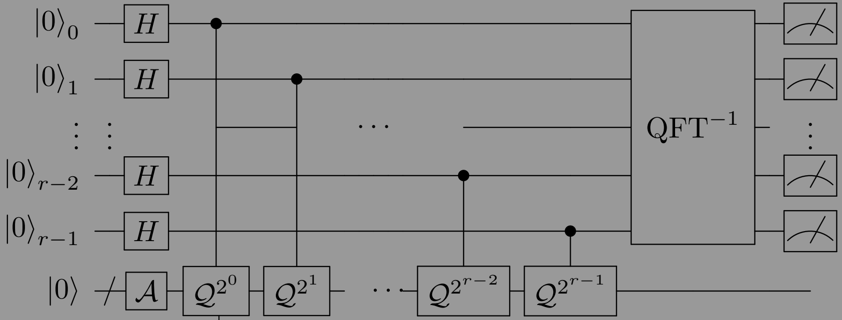
QPE to QME

How does **mean** estimation enter?



QPE to QME

How does **mean** estimation enter?

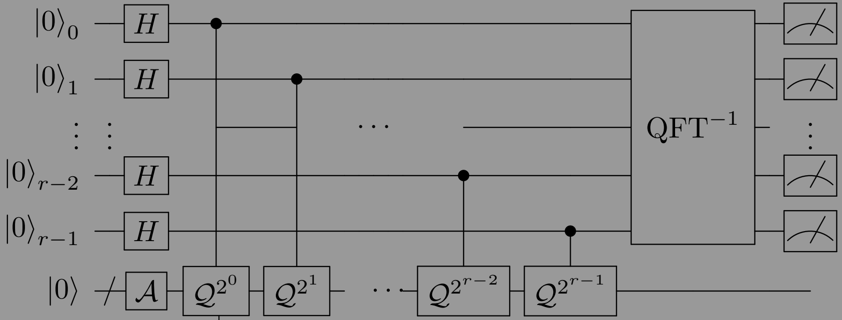


$$\mathcal{Q} = \mathcal{A} S_0 \mathcal{A}^\dagger U_\varphi$$

$$\mathcal{Q}\,|q\rangle \sim e^{i\bar{x}}\,|q\rangle$$

QPE to QME

How does **mean** estimation enter?



$$\mathcal{Q} = \mathcal{A} S_0 \mathcal{A}^\dagger U_\varphi$$

$$\mathcal{Q} |q\rangle \sim e^{i\bar{x}} |q\rangle$$

$$\bar{x} \sim \langle 0| \mathcal{A}^\dagger U_\varphi \mathcal{A} |0\rangle = \langle q| U_\varphi |q\rangle$$

Trace as a mean

With block-encoding

$$\Gamma \equiv \frac{1}{V}\mathrm{Tr}\left[U\right]$$

Trace as a mean

With block-encoding

$$\Gamma \equiv \frac{1}{V}\mathrm{Tr}\left[U\right] = \frac{1}{V}\sum_{i=0}^{V-1} U_{ii}$$

Trace as a mean

With block-encoding

$$\Gamma \equiv \frac{1}{V}\mathrm{Tr}\left[U\right] = \frac{1}{V}\sum_{i=0}^{V-1} U_{ii} = \frac{1}{V}\sum_{i=0}^{V-1} \langle 0| \langle i \,|\, \mathbb{V} \,|0\rangle \,|\, i\rangle$$

Trace as a mean

With block-encoding

$$\Gamma \equiv \frac{1}{V}\mathrm{Tr}\left[U\right] = \frac{1}{V}\sum_{i=0}^{V-1} U_{ii} = \frac{1}{V}\sum_{i=0}^{V-1} \langle 0| \langle i \,|\, \mathbb{V} \; |0\rangle \,|\, i\rangle$$

$$= \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \delta_{ij} \langle 0| \langle j \,|\, \mathbb{V} \; |0\rangle \,|\, i\rangle$$

Trace as a mean

With block-encoding

$$\Gamma \equiv \frac{1}{V}\mathrm{Tr}\left[U\right] = \frac{1}{V}\sum_{i=0}^{V-1} U_{ii} = \frac{1}{V}\sum_{i=0}^{V-1} \langle 0| \langle i | \mathbb{V} \; |0\rangle \, | i\rangle$$

$$= \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \delta_{ij} \langle 0| \langle j | \mathbb{V} \; |0\rangle \, | i\rangle$$

$$= \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j | i\rangle \langle 0| \langle j | \mathbb{V} \; |0\rangle \, | i\rangle$$

Trace as a mean

With block-encoding

$$\Gamma \equiv \frac{1}{V}\mathrm{Tr}\left[U\right] = \frac{1}{V}\sum_{i=0}^{V-1} U_{ii} = \frac{1}{V}\sum_{i=0}^{V-1} \langle 0| \langle i | \mathbb{V} |0\rangle | i\rangle$$

$$= \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \delta_{ij} \langle 0| \langle j | \mathbb{V} |0\rangle | i\rangle$$

$$= \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j | i\rangle \langle 0| \langle j | \mathbb{V} |0\rangle | i\rangle$$

$$= \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j| \langle 0| \langle j| \mathbb{1}\otimes\mathbb{V} |i\rangle |0\rangle |i\rangle$$

Pieces

$$\Gamma = \frac{1}{V} \sum_{i=0}^{V-1} \sum_{j=0}^{V-1} \langle j| \langle 0| \langle j| \; \mathbb{1} \otimes \mathbb{V} \; |i\rangle |0\rangle |i\rangle$$

Pieces

$$\Gamma = \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j| \langle 0| \langle j| \; \mathbb{1} \otimes \mathbb{V} \; |i\rangle \, |0\rangle \, |i\rangle$$

$$\sim \langle 0| \mathcal{A}^{\dagger} U_{\varphi} \mathcal{A} |0\rangle$$

Pieces

$$\Gamma = \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j| \langle 0| \langle j| \; \mathbb{1} \otimes \mathbb{V} \; |i\rangle |0\rangle |i\rangle$$

$$\sim \langle 0| \mathcal{A}^\dagger U_\varphi \mathcal{A} |0\rangle$$

$$\mathcal{A}|0\rangle = \frac{1}{\sqrt{V}} \sum_{j=0}^{V-1} |j\rangle |0\rangle |j\rangle$$

Pieces

$$\Gamma = \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j| \langle 0| \langle j| \ \mathbb{1} \otimes \mathbb{V} \ |i\rangle |0\rangle |i\rangle$$

$$\sim \langle 0| \mathcal{A}^\dagger U_\varphi \mathcal{A} |0\rangle$$

$$\mathcal{A}|0\rangle = \frac{1}{\sqrt{V}}\sum_{j=0}^{V-1} |j\rangle |0\rangle |j\rangle$$

$$U_\varphi \sim \mathbb{1} \otimes \mathbb{V}$$

Pieces

$$\Gamma = \frac{1}{V}\sum_{i=0}^{V-1}\sum_{j=0}^{V-1} \langle j| \langle 0| \langle j| \; \mathbb{1} \otimes \mathbb{V} \; |i\rangle |0\rangle |i\rangle$$

$$\sim \langle 0| \mathcal{A}^\dagger U_\varphi \mathcal{A} |0\rangle$$

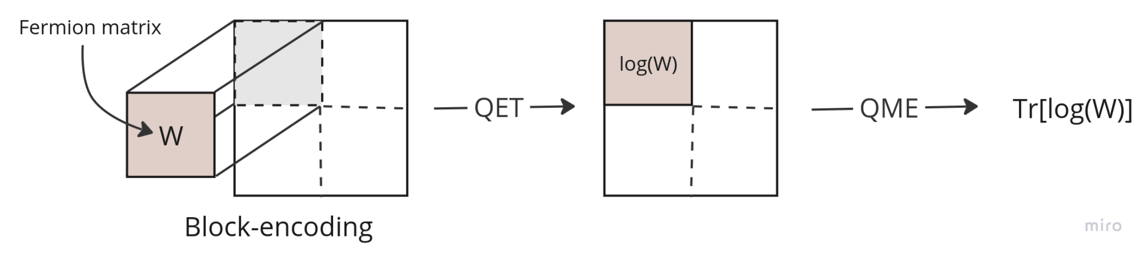
$$\mathcal{A}|0\rangle = \frac{1}{\sqrt{V}}\sum_{j=0}^{V-1} |j\rangle |0\rangle |j\rangle$$

$$U_\varphi \sim \mathbb{1} \otimes \mathbb{V}$$

Use QPE!

Determinant

Putting it all together



Putting it all together

Putting it all together

- Need $U_\varphi = \mathbb{1} \otimes \mathbb{V}$ for QME

Putting it all together

- Need $U_\varphi = \mathbb{1} \otimes \mathbb{V}$ for QME
- $\mathbb{V} = ?$

Putting it all together

- Need $U_\varphi = \mathbb{1} \otimes \mathbb{V}$ for QME
- $\mathbb{V} = \ \log(W)$

$$\mathbb{V} = \text{QET}\,(\text{block-encode})$$

Putting it all together

- Need $U_\varphi = \mathbb{1} \otimes \mathbb{V}$ for QME
- $\mathbb{V} = \log(W)$

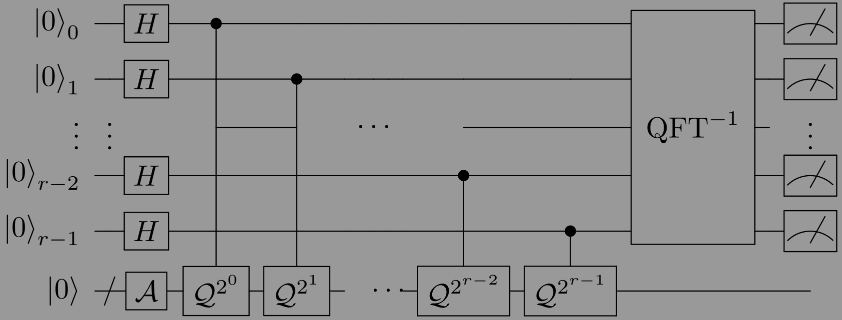
$$\begin{aligned}\mathbb{V} &= \text{QET}\,(\text{block-encode}) \\ &= \text{QET}(\underbrace{D_s O_c O_A D_s}_{B})\end{aligned}$$

Putting it all together

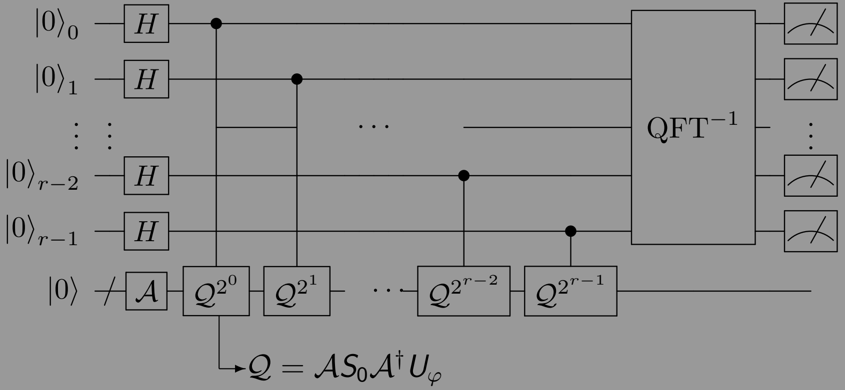
- Need $U_\varphi = \mathbb{1} \otimes \mathbb{V}$ for QME
- $\mathbb{V} = \log(W)$

$$
\begin{aligned}
\mathbb{V} &= \text{QET}\,(\text{block-encode}) \\
&= \text{QET}(\underbrace{D_s O_c O_A D_s}_{B}) \\
&= U_\Pi(\phi_0) \prod_{i=1}^{d} B\, U_\Pi(\phi_i)
\end{aligned}
$$

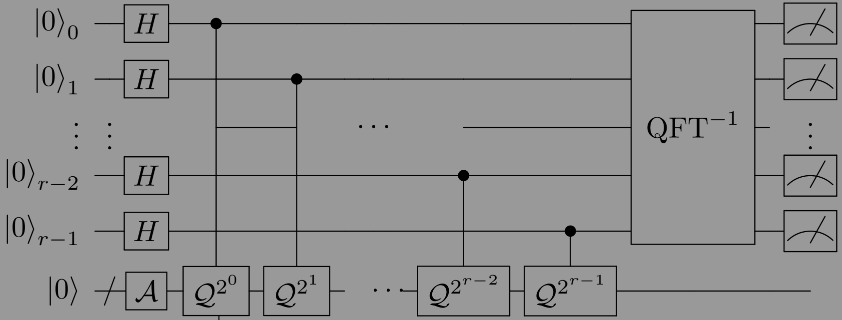
Putting it all together



Putting it all together



Putting it all together



$\mathcal{Q} = \mathcal{A} S_0 \mathcal{A}^\dagger U_\varphi$

$$U_\varphi = \mathbb{1} \otimes U_\Pi(\phi_0) \prod_{i=1}^{d} B\, U_\Pi(\phi_i)$$

What is the cost?

Cost of block-encoding + matrix log:

What is the cost?

Cost of block-encoding + matrix log:

$$\mathbb{V} = \overbrace{U_\Pi(\phi_0) \prod_{i=1}^{d} (\underbrace{D_s O_c O_A D_s}_{O(V \log V)}) \, U_\Pi(\phi_i)}^{d}$$

What is the cost?

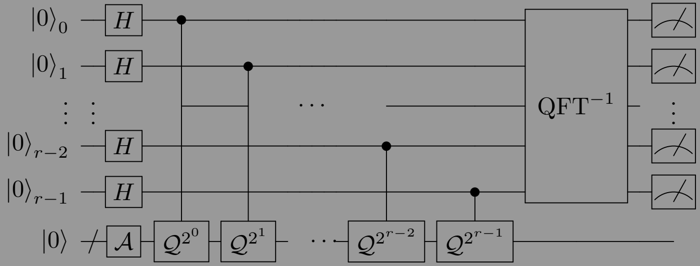
Cost of block-encoding + matrix log:

$$\mathbb{V} = \overbrace{U_{\Pi}(\phi_0)\prod_{i=1}^{d}(\underbrace{D_s O_c O_A D_s}_{O(V\log V)})\, U_{\Pi}(\phi_i)}^{d}$$

$\mathbb{V}$ cost is $O(dV\log V)$

What is the cost?

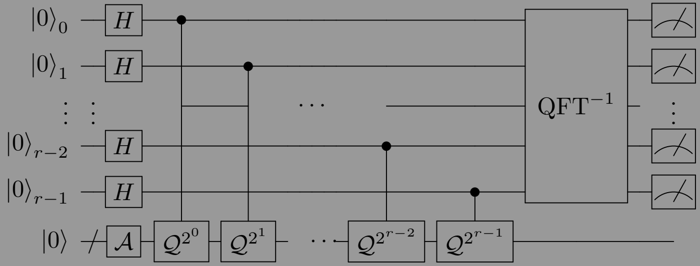
Cost of quantum mean estimation



What is the cost?

Cost of quantum mean estimation

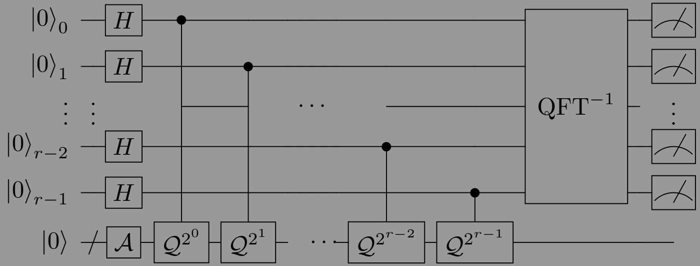
- QPE runtime: $O(1/\epsilon)$



What is the cost?

Cost of quantum mean estimation

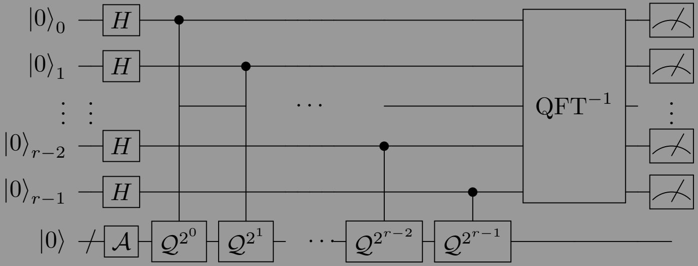
- QPE runtime: $O(1/\epsilon)$
- ϵ is error on the trace estimation



What is the cost?

Cost of quantum mean estimation

- QPE runtime: $O(1/\epsilon)$
- ϵ is error on the trace estimation
  - Note: classically $O(1/\epsilon^2)$



What is the cost?

Total cost

What is the cost?

Total cost

- block-encode + matrix log: $O(dV \log V)$

What is the cost?

Total cost

- block-encode + matrix log: $O(dV \log V)$
- trace: $O(1/\epsilon)$

What is the cost?

Total cost

- block-encode + matrix log: $O(dV \log V)$
- trace: $O(1/\epsilon)$
- total: $O(dV \log V/\epsilon)$

Conclusions

Conclusions

Conclusions

- Quantum algorithm for $\log\det$

Conclusions

- Quantum algorithm for $\log \det$
- Scales $O(V \log V)$. Compare $O(V^3)$ classically.

Conclusions

- Quantum algorithm for $\log\det$
- Scales $O(V \log V)$. Compare $O(V^3)$ classically.
- For sparse matrices

Conclusions

- Quantum algorithm for $\log\det$
- Scales $O(V \log V)$. Compare $O(V^3)$ classically.
- For sparse matrices
- Uses $\log\det \sim \mathrm{Tr}\log$

Conclusions

- Quantum algorithm for $\log \det$
- Scales $O(V \log V)$. Compare $O(V^3)$ classically.
- For sparse matrices
- Uses $\log \det \sim \mathrm{Tr} \log$
- 2407.13080

Conclusions

- Quantum algorithm for log det
- Scales $O(V \log V)$. Compare $O(V^3)$ classically.
- For sparse matrices
- Uses $\log\det \sim \mathrm{Tr}\log$
- 2407.13080

Thank you!

References I

Camps, D., Lin, L., Beeumen, R. V., and Yang, C. (2023).
Explicit quantum circuits for block encodings of certain sparse matrices.

Dong, Y., Meng, X., Whaley, K. B., and Lin, L. (2021).
Efficient phase-factor evaluation in quantum signal processing.
Phys. Rev. A, 103:042419.

Gilyén, A., Su, Y., Low, G. H., and Wiebe, N. (2019).
Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics.
In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19. ACM.

References II

Gustafson, E. J., Lamm, H., and Unmuth-Yockey, J. (2023).
Quantum mean estimation for lattice field theory.
Phys. Rev. D, 107:114511.

Hamoudi, Y. (2021).
Quantum algorithms for the monte carlo method.
PhD thesis, Université de Paris.

Low, G. H. and Chuang, I. L. (2019).
Hamiltonian simulation by qubitization.
Quantum, 3:163.

Martyn, J. M., Rossi, Z. M., Tan, A. K., and Chuang, I. L. (2021).
Grand unification of quantum algorithms.
PRX Quantum, 2:040203.

References III

Montanaro, A. (2015).
Quantum speedup of monte carlo methods.
Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471(2181):20150301.

Shyamsundar, P. (2021).
Non-boolean quantum amplitude amplification and quantum mean estimation.