Amplitude Estimation based on Quantum Signal Processing

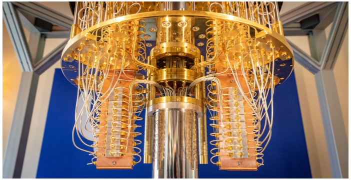
Patrick Rall, Bryce Fuller

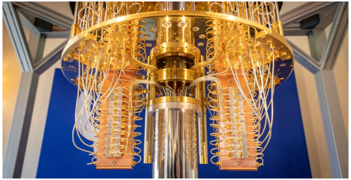
arXiv:2207.08628

IBM **Quantum**

September 2022

- Digital computation with superposition
- Applications: physics, chemistry, cryptography, finance...
- IBM Eagle (2021): 127 Quantum Bits



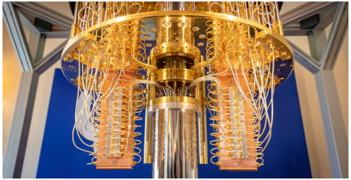


- Digital computation with superposition
- Applications: physics, chemistry, cryptography, finance...
- IBM Eagle (2021): 127 Quantum Bits

- Warm-up: probabilistic computation

$$\text{States: } \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1/2\\1/2\end{bmatrix}, \begin{bmatrix}2/3\\1/3\end{bmatrix} \rightarrow \vec{p} \in \mathbb{R}_+^{2^n}, \ |\vec{p}|_1 = 1$$

- Digital computation with superposition
- Applications: physics, chemistry, cryptography, finance...
- IBM Eagle (2021): 127 Quantum Bits



- Warm-up: probabilistic computation

$$\text{States:} \quad \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1/2\\1/2\end{bmatrix}, \begin{bmatrix}2/3\\1/3\end{bmatrix} \quad \rightarrow \quad \vec{p} \in \mathbb{R}_+^{2^n}, \quad |\vec{p}|_1 = 1$$

$$\text{Transformations:} \quad \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \begin{bmatrix}1/2 & 1/2\\1/2 & 1/2\end{bmatrix}, \begin{bmatrix}1 & 1/2\\0 & 1/2\end{bmatrix} \quad \rightarrow \quad |M\vec{p}|_1 = 1$$

States: $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}+1\\-1\end{bmatrix} \quad \rightarrow \quad \vec{\psi} \in \mathbb{C}^{2^n}, \ |\vec{\psi}|_2 = 1$

A Short Intro to QC

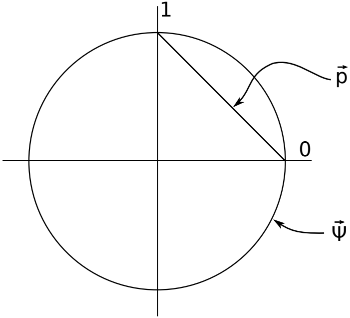
IBM **Quantum**

States: $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}+1\\-1\end{bmatrix} \rightarrow \vec{\psi} \in \mathbb{C}^{2^n}, \ |\vec{\psi}|_2 = 1$

Transformations: $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}, \cancel{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}} \rightarrow |M\vec{\psi}|_2 = 1$

States: $$\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}+1\\-1\end{bmatrix} \rightarrow \vec{\psi} \in \mathbb{C}^{2^n}, |\vec{\psi}|_2 = 1$$

Transformations: $$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}, \xcancel{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}} \rightarrow |M\vec{\psi}|_2 = 1$$

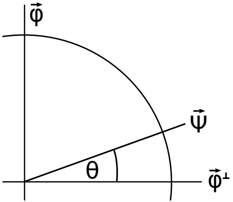


Grover's algorithm

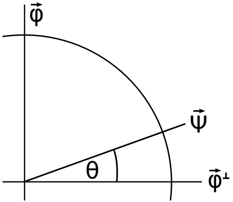
- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$

- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$
- Operations: $Z_\psi = \vec{\psi}\vec{\psi}^* - I$ and $Z_\phi = \vec{\phi}\vec{\phi}^* - I$

- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$
- Operations: $Z_\psi = \vec{\psi}\vec{\psi}^* - I$ and $Z_\phi = \vec{\phi}\vec{\phi}^* - I$



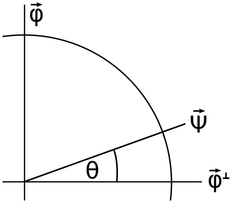
- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$
- Operations: $Z_\psi = \vec{\psi}\vec{\psi}^* - I$ and $Z_\phi = \vec{\phi}\vec{\phi}^* - I$



- Amplitude amplification: transform $\vec{\psi} \to \vec{\phi}$ in $O(|\vec{\psi}^*\vec{\phi}|^{-1})$

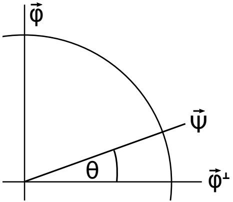
Grover's algorithm

- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$
- Operations: $Z_\psi = \vec{\psi}\vec{\psi}^* - I$ and $Z_\phi = \vec{\phi}\vec{\phi}^* - I$



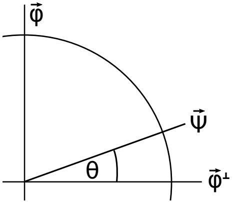
- Amplitude amplification: transform $\vec{\psi} \to \vec{\phi}$ in $O(|\vec{\psi}^*\vec{\phi}|^{-1})$
  - Applications: Search, Ground state and thermal state preparation

- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$
- Operations: $Z_\psi = \vec{\psi}\vec{\psi}^* - I$ and $Z_\phi = \vec{\phi}\vec{\phi}^* - I$



- Amplitude amplification: transform $\vec{\psi} \to \vec{\phi}$ in $O(|\vec{\psi}^*\vec{\phi}|^{-1})$
  - Applications: Search, Ground state and thermal state preparation
- Amplitude estimation: estimate $a := |\vec{\psi}^*\vec{\phi}| = \sin(\theta)$ to accuracy ε in $O(\varepsilon^{-1})$

- All dynamics take place in 2d subspace span$(\vec{\psi}, \vec{\phi}) \in \mathbb{C}^{2^n}$
- Operations: $Z_\psi = \vec{\psi}\vec{\psi}^* - I$ and $Z_\phi = \vec{\phi}\vec{\phi}^* - I$



- Amplitude amplification: transform $\vec{\psi} \rightarrow \vec{\phi}$ in $O(|\vec{\psi}^*\vec{\phi}|^{-1})$
  - Applications: Search, Ground state and thermal state preparation
- Amplitude estimation: estimate $a := |\vec{\psi}^*\vec{\phi}| = \sin(\theta)$ to accuracy ε in $O(\varepsilon^{-1})$
  - Applications: Monte Carlo estimation, partition function estimation

Algorithm idea:

Algorithm idea:

1. Start in the state $\vec{\psi}$

Algorithm idea:

1. Start in the state $\vec{\psi}$
2. Alternatingly apply reflections Z_ϕ and Z_ψ n times

Algorithm idea:

1. Start in the state $\vec{\psi}$
2. Alternatingly apply reflections Z_ϕ and Z_ψ n times
3. Collapse the superposition in the $\vec{\phi}, \vec{\phi}^\perp$ basis

Algorithm idea:

1. Start in the state $\vec{\psi}$
2. Alternatingly apply reflections Z_ϕ and Z_ψ n times
3. Collapse the superposition in the $\vec{\phi}, \vec{\phi}^\perp$ basis

$$\Pr[\text{obtain } \vec{\phi}] = \left| \vec{\phi}^* (Z_\psi Z_\phi)^n \vec{\psi} \right|^2$$

Algorithm idea:

1. Start in the state $\vec{\psi}$
2. Alternatingly apply reflections Z_ϕ and Z_ψ n times
3. Collapse the superposition in the $\vec{\phi}, \vec{\phi}^\perp$ basis

$$\begin{aligned}\text{Pr}[\text{obtain } \vec{\phi}] &= \left|\vec{\phi}^*(Z_\psi Z_\phi)^n \vec{\psi}\right|^2 \\ &= |T_{2n+1}(a)|^2\end{aligned}$$

$$T_d(x) = d\text{'th Chebyshev polynomial}$$

Algorithm idea:

1. Start in the state $\vec{\psi}$
2. Alternatingly apply reflections Z_ϕ and Z_ψ n times
3. Collapse the superposition in the $\vec{\phi}, \vec{\phi}^\perp$ basis

$$\Pr[\text{obtain } \vec{\phi}] = \left| \vec{\phi}^* (Z_\psi Z_\phi)^n \vec{\psi} \right|^2$$
$$= |T_{2n+1}(a)|^2$$

$$T_d(x) = d\text{'th Chebyshev polynomial}$$

Grover’s algorithm

Algorithm idea:

1. Start in the state $\vec{\psi}$
2. Alternatingly apply reflections Z_ϕ and Z_ψ n times
3. Collapse the superposition in the $\vec{\phi}, \vec{\phi}^\perp$ basis

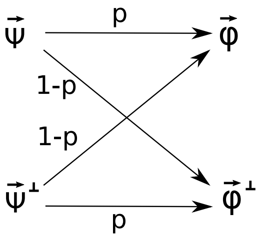
$$\Pr[\text{obtain } \vec{\phi}] = \left| \vec{\phi}^* (Z_\psi Z_\phi)^n \vec{\psi} \right|^2$$

$$= |T_{2n+1}(a)|^2$$

$$T_d(x) = d\text{'th Chebyshev polynomial}$$

Grover’s algorithm

Given $\vec{\psi}$ or $\vec{\psi}^\perp$, prepare either $\vec{\phi}$ or $\vec{\phi}^\perp$ with probability $p = |T_{2n+1}(a)|^2$ using $O(n)$ operations.



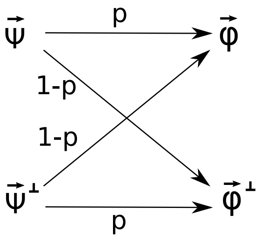
- Alternating reflections $Z_\psi Z_\phi \to$ Chebyshev polynomials $T_n(a)$

- Alternating reflections $Z_\psi Z_\phi \rightarrow$ Chebyshev polynomials $T_n(a)$
- Alternating rotations $e^{i\theta Z_\psi} e^{i\theta' Z_\phi} \rightarrow$ more polynomials $P(a)$

- Alternating reflections $Z_\psi Z_\phi \to$ Chebyshev polynomials $T_n(a)$
- Alternating rotations $e^{i\theta Z_\psi} e^{i\theta' Z_\phi} \to$ more polynomials $P(a)$

Grover's algorithm + Quantum Signal Processing

Say $P(a)$ is a polynomial of degree d. Given $\vec{\psi}$ or $\vec{\psi}^\perp$, prepare either $\vec{\phi}, \vec{\phi}^\perp$ with probability $p = |P(a)|^2$ using $O(d)$ operations.



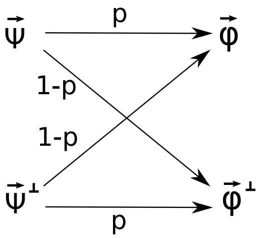
Quantum Signal Processing

IBM **Quantum**

- Alternating reflections $Z_\psi Z_\phi \to$ Chebyshev polynomials $T_n(a)$
- Alternating rotations $e^{i\theta Z_\psi} e^{i\theta' Z_\phi} \to$ more polynomials $P(a)$

Grover's algorithm + Quantum Signal Processing

Say $P(a)$ is a polynomial of degree d. Given $\vec{\psi}$ or $\vec{\psi}^\perp$, prepare either $\vec{\phi}, \vec{\phi}^\perp$ with probability $p = |P(a)|^2$ using $O(d)$ operations.



- Goal: estimate $a = |\vec{\psi}^* \vec{\phi}|$ in as few operations as possible.

Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$

Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$
2. Non-destructive estimation: given exactly one input state $\vec{\psi}$

Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$
2. Non-destructive estimation: given exactly one input state $\vec{\psi}$
3. Hybrid quantum-classical estimation: cost $O(1/\varepsilon^{1+\beta})$, but maximum polynomial degree $O(1/\varepsilon^{1-\beta})$

IBM **Quantum**

Puzzle: estimating a via polynomials

Say $a \in [0, 1]$ is unknown. For polynomials $p : [0, 1] \to [0, 1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$
2. Non-destructive estimation: given exactly one input state $\vec{\psi}$
3. Hybrid quantum-classical estimation: cost $O(1/\varepsilon^{1+\beta})$, but maximum polynomial degree $O(1/\varepsilon^{1-\beta})$
4. Unbiased estimation: output an estimate $\hat{a}$ with $\mathbb{E}[\hat{a}] \approx a$

Fast estimation

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$

Fast estimation

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

Fast estimation

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$

Fast estimation

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:
   1. Pick a Chebyshev polynomial $|T_n(a)|^2$ that's purely increasing/decreasing on $[a_{\text{min}}, a_{\text{max}}]$

Fast estimation

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

Method:

1. Initialize a confidence interval $[a_{\min}, a_{\max}] = [0, 1]$
2. While $a_{\max} - a_{\min} > 2\varepsilon$:
    1. Pick a Chebyshev polynomial $|T_n(a)|^2$ that's purely increasing/decreasing on $[a_{\min}, a_{\max}]$
    2. Estimate $p = |T_n(a)|^2$ to constant precision.

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:
    1. Pick a Chebyshev polynomial $|T_n(a)|^2$ that's purely increasing/decreasing on $[a_{\text{min}}, a_{\text{max}}]$
    2. Estimate $p = |T_n(a)|^2$ to constant precision.
    3. Use the estimate of p to improve the confidence interval.

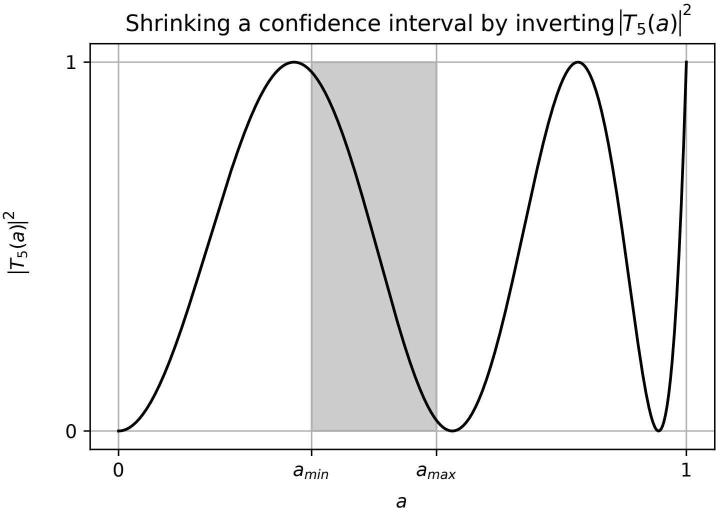
Fast estimation

- Basic idea: Grinko et al. 1912.05559. Achieves $\approx 2.62/\varepsilon$
- We achieve $\approx 1.7/\varepsilon$ by simplifying and fine-tuning their method

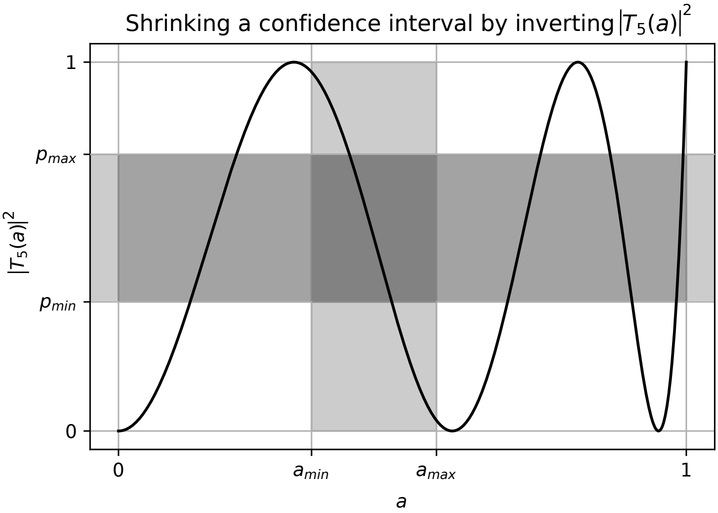
Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:
   1. Pick a Chebyshev polynomial $|T_n(a)|^2$ that's purely increasing/decreasing on $[a_{\text{min}}, a_{\text{max}}]$
   2. Estimate $p = |T_n(a)|^2$ to constant precision.
   3. Use the estimate of p to improve the confidence interval.
3. Return the midpoint of $[a_{\text{min}}, a_{\text{max}}]$.

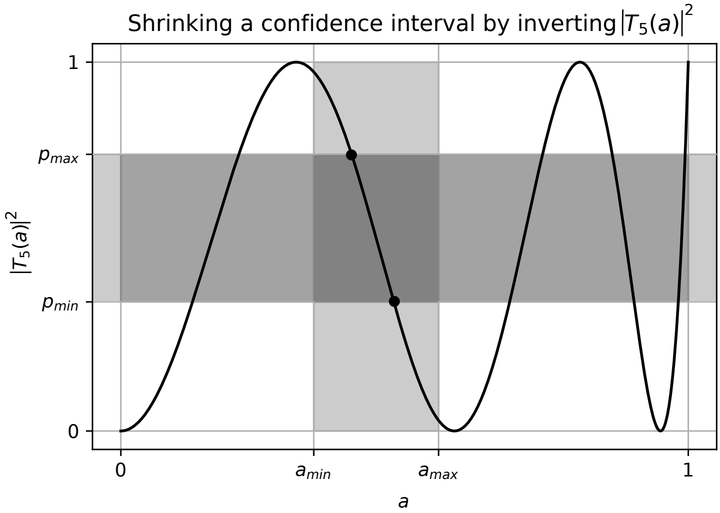
Fast estimation: Interval refinement



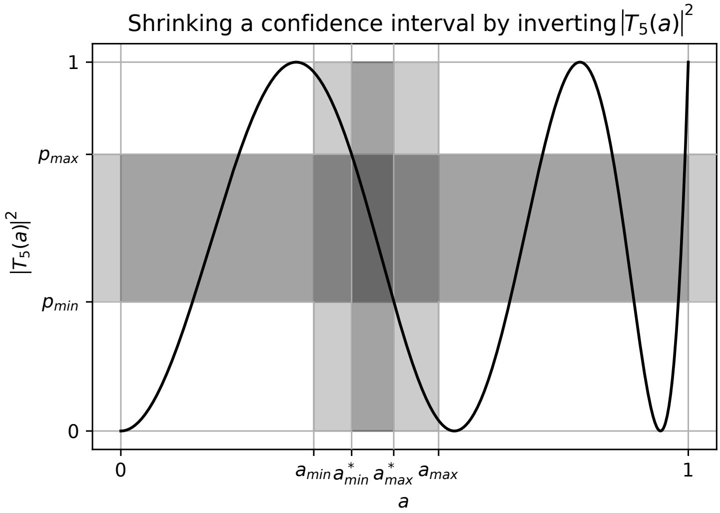
Fast estimation: Interval refinement

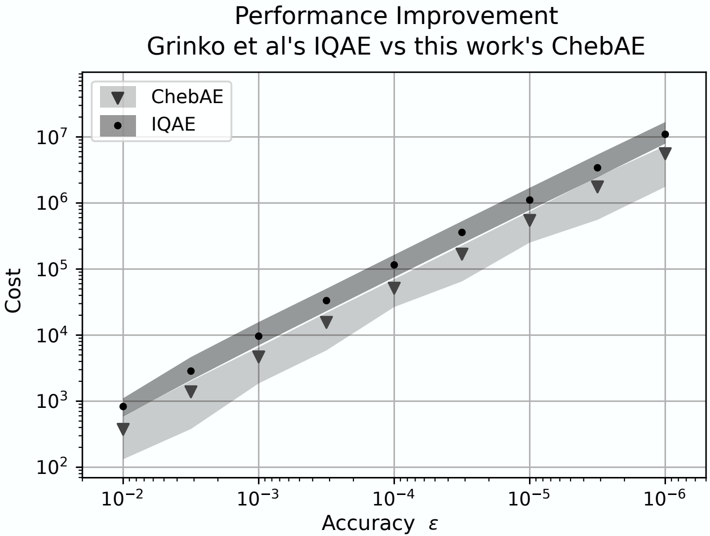


Fast estimation: Interval refinement



Fast estimation: Interval refinement




Performance Improvement
Grinko et al's IQAE vs this work's ChebAE
ChebAE
IQAE
Cost
10^2
10^3
10^4
10^5
10^6
10^7
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
Accuracy ε

Results

Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

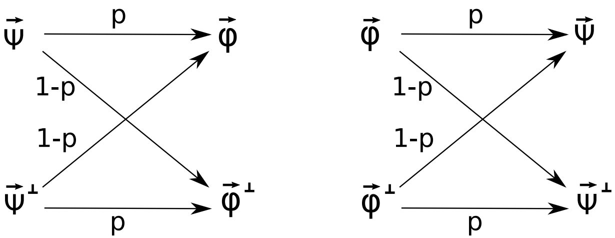
1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$ ✓
2. Non-destructive estimation: given exactly one input state $\vec{\psi}$
3. Hybrid quantum-classical estimation: cost $O(1/\varepsilon^{1+\beta})$, but maximum polynomial degree $O(1/\varepsilon^{1-\beta})$
4. Unbiased estimation: output an estimate $\hat{a}$ with $\mathbb{E}[\hat{a}] \approx a$

Grover's algorithm + Quantum Signal Processing

Say $P(a)$ is a polynomial of degree d. Then:

- We can transform $\vec{\psi}, \vec{\psi}^\perp \to \vec{\phi}, \vec{\phi}^\perp$
- We can transform $\vec{\phi}, \vec{\phi}^\perp \to \vec{\psi}, \vec{\psi}^\perp$

with $p = |P(a)|^2$ using $O(d)$ operations.



Non-destructive Estimation

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

Lemma: post-algorithm state

Say we just ran an amplitude estimation algorithm,
and say the sum of all the degrees of the polynomials was D.
Then, if $a < \sqrt{\delta}/D$ then with probability $\geq 1 - \delta$ we have either $\vec{\psi}$ or $\vec{\phi}^{\perp}$.

Non-destructive Estimation

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

Lemma: post-algorithm state

Say we just ran an amplitude estimation algorithm,
and say the sum of all the degrees of the polynomials was D.
Then, if $a < \sqrt{\delta}/D$ then with probability $\geq 1 - \delta$ we have either $\vec{\psi}$ or $\vec{\phi}^{\perp}$.

Method:

Non-destructive Estimation

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

Lemma: post-algorithm state

Say we just ran an amplitude estimation algorithm,
and say the sum of all the degrees of the polynomials was D.
Then, if $a < \sqrt{\delta}/D$ then with probability $\geq 1 - \delta$ we have either $\vec{\psi}$ or $\vec{\phi}^{\perp}$.

Method:

- If we have $\vec{\psi}$, we are done.

Non-destructive Estimation

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

Lemma: post-algorithm state

Say we just ran an amplitude estimation algorithm,
and say the sum of all the degrees of the polynomials was D.
Then, if $a < \sqrt{\delta}/D$ then with probability $\geq 1 - \delta$ we have either $\vec{\psi}$ or $\vec{\phi}^{\perp}$.

Method:

- If we have $\vec{\psi}$, we are done.
- If we have $\vec{\psi}^{\perp}$, sample $P(a) = a$ to get $\vec{\phi}$ or $\vec{\phi}^{\perp}$

Non-destructive Estimation

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

Lemma: post-algorithm state

Say we just ran an amplitude estimation algorithm,
and say the sum of all the degrees of the polynomials was D.
Then, if $a < \sqrt{\delta}/D$ then with probability $\geq 1 - \delta$ we have either $\vec{\psi}$ or $\vec{\phi}^{\perp}$.

Method:

- If we have $\vec{\psi}$, we are done.
- If we have $\vec{\psi}^{\perp}$, sample $P(a) = a$ to get $\vec{\phi}$ or $\vec{\phi}^{\perp}$
- If we have $\vec{\phi}$, we can assume $a > \sqrt{\delta}/D$.

- Main idea: add a *repair step* to an existing amplitude estimation algorithm. (Assume $a < 1/2$ for simplicity.)

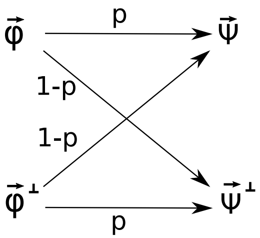
Lemma: post-algorithm state

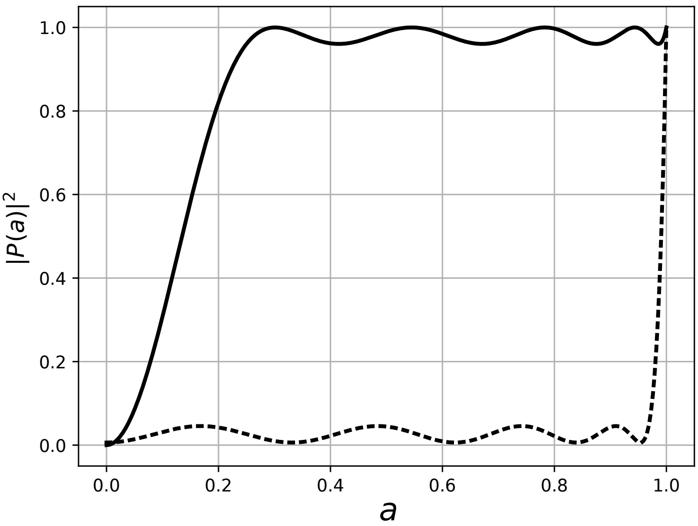
Say we just ran an amplitude estimation algorithm,
and say the sum of all the degrees of the polynomials was D.
Then, if $a < \sqrt{\delta}/D$ then with probability $\geq 1 - \delta$ we have either $\vec{\psi}$ or $\vec{\phi}^{\perp}$.

Method:

- If we have $\vec{\psi}$, we are done.
- If we have $\vec{\psi}^{\perp}$, sample $P(a) = a$ to get $\vec{\phi}$ or $\vec{\phi}^{\perp}$
- If we have $\vec{\phi}$, we can assume $a > \sqrt{\delta}/D$.
- If we have $\vec{\phi}^{\perp}$, we leverage $a < 1/2$.

Converting $\vec{\phi}, \vec{\phi}^\perp \to \vec{\psi}$





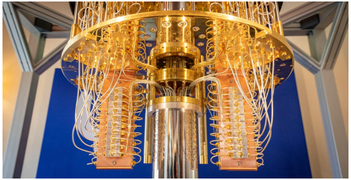
Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

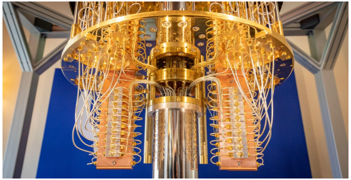
1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$ ✓
2. Non-destructive estimation: given exactly one input state $\vec{\psi}$ ✓
3. Hybrid quantum-classical estimation: cost $O(1/\varepsilon^{1+\beta})$, but maximum polynomial degree $O(1/\varepsilon^{1-\beta})$
4. Unbiased estimation: output an estimate $\hat{a}$ with $\mathbb{E}[\hat{a}] \approx a$

Hybrid quantum-classical estimation

- Current quantum computers are noisy
  $\to$ can only run programs of a certain length
- Program length $\sim$ polynomial degree
- Can think of degree $O(1)$ as classical limit



Hybrid quantum-classical estimation



- Current quantum computers are noisy
  $\rightarrow$ can only run programs of a certain length
- Program length $\sim$ polynomial degree
- Can think of degree $O(1)$ as classical limit

		Max Degree	Total Degree
Quantum		$O(1/\varepsilon)$	$O(1/\varepsilon)$
Classical		$O(1)$	$O(1/\varepsilon^2)$
Hybrid	$\beta \in [0, 1]$	$O(1/\varepsilon^{1-\beta})$	$O(1/\varepsilon^{1+\beta})$

Problem was first posed by Giurgica-Tironc et al 2012.03348.

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:

Amplitude estimation via interval refinement

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:
   1. Let $\Delta = a_{\text{max}} - a_{\text{min}}$. Decide if:

$$a \in [a_{\text{min}} + 0.1\Delta, a_{\text{max}}] \quad \text{(case I)}$$

$$a \in [a_{\text{min}}, a_{\text{max}} - 0.1\Delta] \quad \text{(case II)}$$

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:
   1. Let $\Delta = a_{\text{max}} - a_{\text{min}}$. Decide if:

$$a \in [a_{\text{min}} + 0.1\Delta, a_{\text{max}}] \quad \text{(case I)}$$

$$a \in [a_{\text{min}}, a_{\text{max}} - 0.1\Delta] \quad \text{(case II)}$$

   2. If I, set a_{min} to $a_{\text{min}} + 0.1\Delta$. If II set a_{max} to $a_{\text{max}} - 0.1\Delta$.

Method:

1. Initialize a confidence interval $[a_{\text{min}}, a_{\text{max}}] = [0, 1]$
2. While $a_{\text{max}} - a_{\text{min}} > 2\varepsilon$:
   1. Let $\Delta = a_{\text{max}} - a_{\text{min}}$. Decide if:

$$a \in [a_{\text{min}} + 0.1\Delta, a_{\text{max}}] \quad \text{(case I)}$$

$$a \in [a_{\text{min}}, a_{\text{max}} - 0.1\Delta] \quad \text{(case II)}$$

   2. If I, set a_{min} to $a_{\text{min}} + 0.1\Delta$. If II set a_{max} to $a_{\text{max}} - 0.1\Delta$.
3. Return the midpoint of $[a_{\text{min}}, a_{\text{max}}]$.

Interval refinement with limited degree

- Decide using $O(1/\Delta^{1-\beta})$ maximum degree and $O(1/\Delta^{1+\beta})$ total degree:

$$a \in [a_{\text{min}} + 0.1\Delta, a_{\text{max}}] \quad \text{(case I)}$$

$$a \in [a_{\text{min}}, a_{\text{max}} - 0.1\Delta] \quad \text{(case II)}$$

Interval refinement with limited degree

IBM **Quantum**

- Decide using $O(1/\Delta^{1-\beta})$ maximum degree and $O(1/\Delta^{1+\beta})$ total degree:

$$a \in [a_{\text{min}} + 0.1\Delta, a_{\text{max}}] \quad \text{(case I)}$$

$$a \in [a_{\text{min}}, a_{\text{max}} - 0.1\Delta] \quad \text{(case II)}$$

- Idea: Combine classical amplification with quantum polynomial degree.

Interval refinement with limited degree

- Decide using $O(1/\Delta^{1-\beta})$ maximum degree and $O(1/\Delta^{1+\beta})$ total degree:

$$a \in [a_{\min} + 0.1\Delta, a_{\max}] \quad \text{(case I)}$$

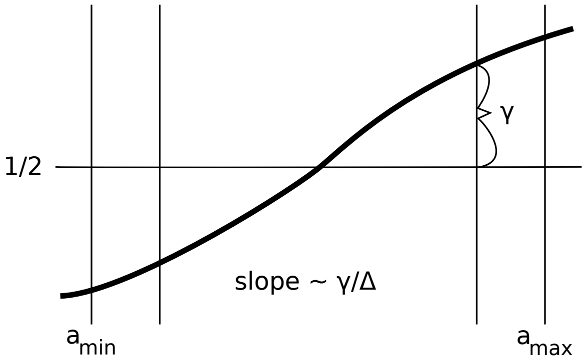
$$a \in [a_{\min}, a_{\max} - 0.1\Delta] \quad \text{(case II)}$$

- Idea: Combine classical amplification with quantum polynomial degree.
- Say we can build a polynomial $p(a)$ such that:

$$a \leq a_{\min} + 0.1\Delta \;\; \rightarrow \;\; p(a) \leq \frac{1}{2} - \gamma$$

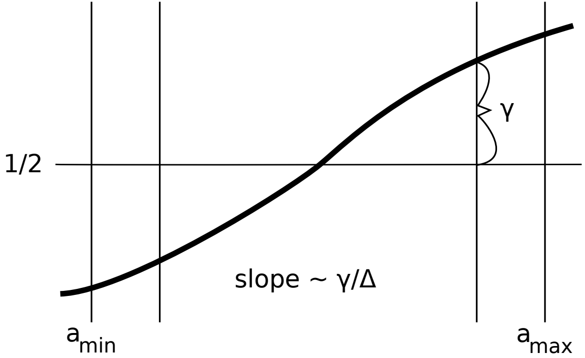
$$a \geq a_{\max} - 0.1\Delta \;\; \rightarrow \;\; p(a) \geq \frac{1}{2} + \gamma$$

$\rightarrow$ then we can distinguish I vs II in $O(1/\gamma^2)$ tries.



1/2
γ
slope ∼ γ/Δ
a_{min}
a_{max}

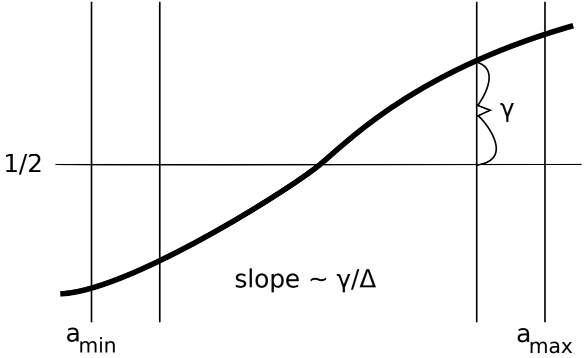
Crafting a polynomial for interval refinement



Polynomial degree $\sim$ slope: $O\left(\frac{\gamma}{\Delta}\right)$ Total cost: $O\left(\frac{1}{\gamma^2}\frac{\gamma}{\Delta}\right)$.

Crafting a polynomial for interval refinement

IBM **Quantum**



Polynomial degree $\sim$ slope: $O\left(\frac{\gamma}{\Delta}\right)$ Total cost: $O\left(\frac{1}{\gamma^2}\frac{\gamma}{\Delta}\right)$.

Achieve $O(1/\Delta^{1-\beta})$ maximum degree and $O(1/\Delta^{1+\beta})$ total degree by selecting $\gamma = \Delta^\beta$.

Puzzle: estimating a via polynomials

Say $a \in [0,1]$ is unknown. For polynomials $p : [0,1] \to [0,1]$, you can toss a coin with bias $p(a)^2$ at cost $\deg(p)$.
Estimate a to precision ε while minimizing cost.

1. Fast estimation: Estimate a to precision ε in $\approx 1.7/\varepsilon$ ✓
2. Non-destructive estimation: given exactly one input state $\vec{\psi}$ ✓
3. Hybrid quantum-classical estimation: cost $O(1/\varepsilon^{1+\beta})$, but maximum polynomial degree $O(1/\varepsilon^{1-\beta})$ ✓
4. Unbiased estimation: output an estimate $\hat{a}$ with $\mathbb{E}[\hat{a}] \approx a$

Unbiased estimation

Goal: sample from a random variable $\hat{a}$ such that:

$$\mathbb{E}[\hat{a}] \approx a \quad \textit{and} \quad \Pr[|\hat{a} - a| > \varepsilon] \leq \delta$$

Goal: sample from a random variable $\hat{a}$ such that:

$$\mathbb{E}[\hat{a}] \approx a \quad \textit{and} \quad \Pr[|\hat{a} - a| > \varepsilon] \leq \delta$$

Approach:

- Say $a \in [a_{\text{min}}, a_{\text{max}}]$ such that $a_{\text{max}} - a_{\text{min}} < \varepsilon$. Pick $\hat{a} = a_{\text{max}}$ with probability

$$\frac{a - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}},$$

otherwise pick $\hat{a} = a_{\text{min}}$. Then $\mathbb{E}[\hat{a}] = a$ and $|\hat{a} - a| < \varepsilon$.

Unbiased estimation

Goal: sample from a random variable $\hat{a}$ such that:

$$\mathbb{E}[\hat{a}] \approx a \quad \textit{and} \quad \Pr[|\hat{a} - a| > \varepsilon] \leq \delta$$

Approach:

- Say $a \in [a_{\text{min}}, a_{\text{max}}]$ such that $a_{\text{max}} - a_{\text{min}} < \varepsilon$. Pick $\hat{a} = a_{\text{max}}$ with probability

$$\frac{a - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}},$$

  otherwise pick $\hat{a} = a_{\text{min}}$. Then $\mathbb{E}[\hat{a}] = a$ and $|\hat{a} - a| < \varepsilon$.
- Construct a polynomial $p(a)$ such that $p(a)^2 \approx \frac{a - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}}$. Then $\mathbb{E}[\hat{a}] \approx a$.

Unbiased estimation

Goal: sample from a random variable $\hat{a}$ such that:

$$\mathbb{E}[\hat{a}] \approx a \quad \textit{and} \quad \Pr[|\hat{a} - a| > \varepsilon] \leq \delta$$

Approach:

- Say $a \in [a_{\text{min}}, a_{\text{max}}]$ such that $a_{\text{max}} - a_{\text{min}} < \varepsilon$. Pick $\hat{a} = a_{\text{max}}$ with probability

$$\frac{a - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}},$$

  otherwise pick $\hat{a} = a_{\text{min}}$. Then $\mathbb{E}[\hat{a}] = a$ and $|\hat{a} - a| < \varepsilon$.
- Construct a polynomial $p(a)$ such that $p(a)^2 \approx \frac{a - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}}$. Then $\mathbb{E}[\hat{a}] \approx a$.
- Apply a recursive argument to the interval refinement algorithm: if all intermediate $\hat{a}$ satisfy $\mathbb{E}[\hat{a}] \approx a$, then so does the algorithm as a whole.

Thank you for your attention!

Special thanks to:

Stefan Woerner, Julien Gacon, and Giacomo Nannicini
John Martyn, Nikitas Stamatopoulos, and Pawel Wocjan

Contact us at `patrickjrall@ibm.com`, `Bryce.Fuller@ibm.com`