Quantum Algorithms and Monte Carlo Estimation

Patrick Rall

The University of Texas at Austin
Quantum Information Center

October 22 2021

Feynman's vision

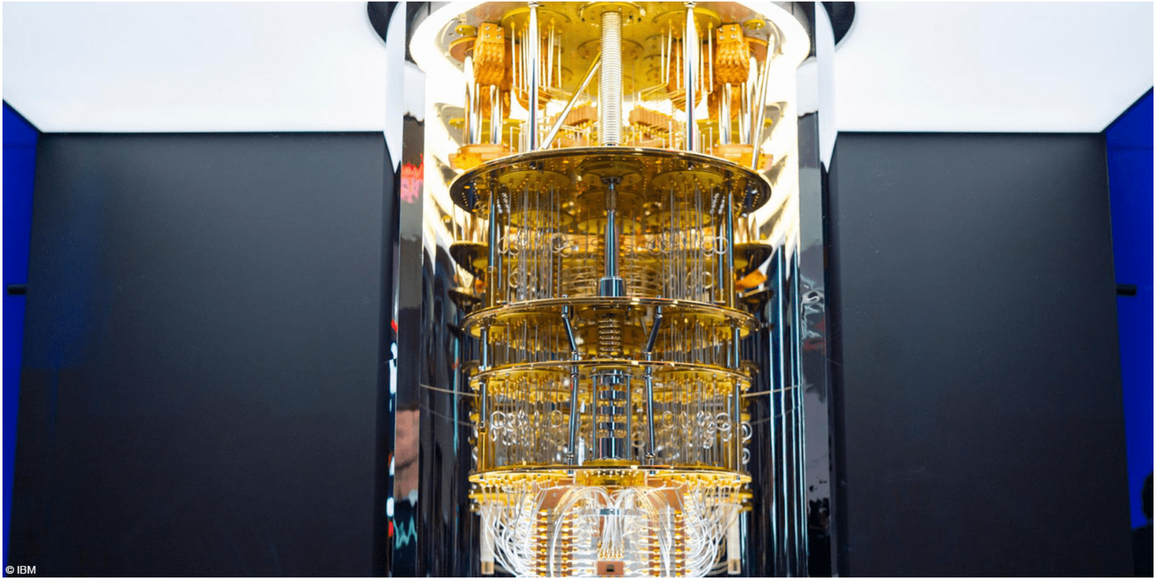
"What kind of computer are we going
to use to simulate physics?"

'Simulating Physics with Computers' 1981

"What kind of computer are we going
to use to simulate physics?"

"If you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

'Simulating Physics with Computers' 1981


© IBM

How well *can* a quantum computer simulate physics?

How well *can* a quantum computer simulate physics?

- ‘Dynamics Questions’:

How well *can* a quantum computer simulate physics?

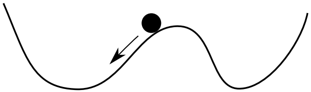
- 'Dynamics Questions':

- 'Statics Questions':

How well *can* a quantum computer simulate physics?

- ‘Dynamics Questions’:

  Input: Current state of the system.
  What is the state of the system 10 seconds later?

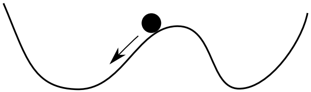


- ‘Statics Questions’:

How well *can* a quantum computer simulate physics?

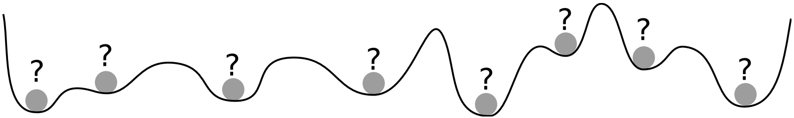
- 'Dynamics Questions':

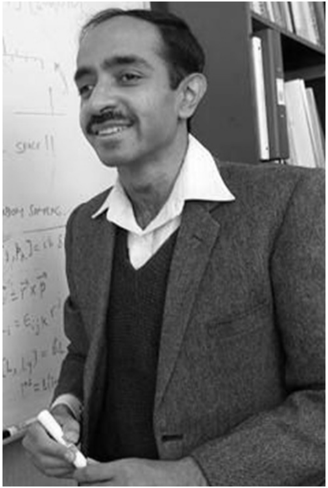
  Input: Current state of the system.
  What is the state of the system 10 seconds later?



- 'Statics Questions':

  Input: Description of the energy landscape.
  In what state are we likely to find the system?



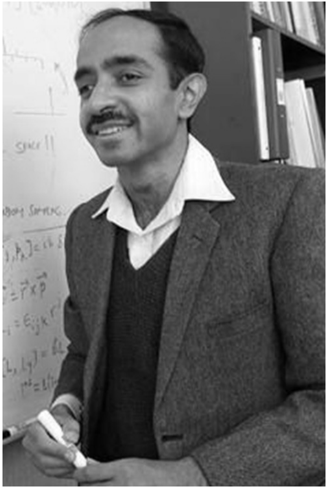


Lov Grover

“Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple possibilities.”

‘A fast quantum mechanical algorithm for database search’ 1996

Quantum Computers and Monte Carlo



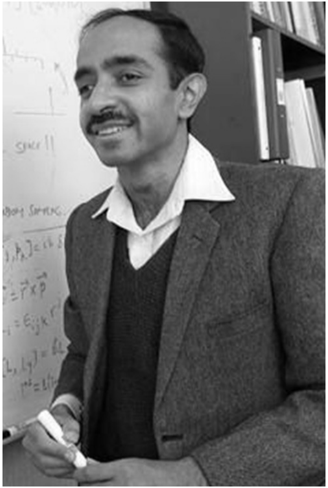
Lov Grover

"Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple possibilities."

'A fast quantum mechanical algorithm for database search' 1996

Say an event happens with probability p.

Quantum Computers and Monte Carlo



Lov Grover

"Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple possibilities."

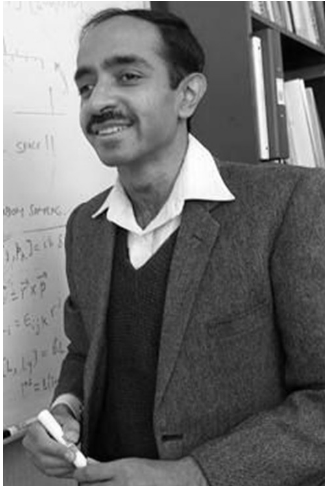
'A fast quantum mechanical algorithm for database search' 1996

Say an event happens with probability p.

Making the event happen:

$$\text{classical} \sim \frac{1}{p} \qquad \text{quantum} \sim \frac{1}{\sqrt{p}}$$

Quantum Computers and Monte Carlo



Lov Grover

"Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple possibilities."

'A fast quantum mechanical algorithm for database search' 1996

Say an event happens with probability p.

Making the event happen:

$$\text{classical} \sim \frac{1}{p} \qquad \text{quantum} \sim \frac{1}{\sqrt{p}}$$

Estimating p to accuracy ε:

$$\text{classical} \sim \frac{1}{\varepsilon^2} \qquad \text{quantum} \sim \frac{1}{\varepsilon}$$

Brassard, Høyer, Mosca, Tapp (BHMT) 1998

Outline

1. An introduction to quantum computing and Grover's algorithm

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
   Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
   Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32
3. An introduction to block encodings

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
   Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32
3. An introduction to block encodings
4. Quantum algorithms for estimating physical quantities
   Rall, arXiv:2004.06832, Phys. Rev. A 102, 022408

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
   Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32
3. An introduction to block encodings
4. Quantum algorithms for estimating physical quantities
   Rall, arXiv:2004.06832, Phys. Rev. A 102, 022408
5. A quantum algorithm for measuring in the energy eigenbasis
   Rall, arXiv:2103.09717, Quantum 5 556

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
   Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32
3. An introduction to block encodings
4. Quantum algorithms for estimating physical quantities
   Rall, arXiv:2004.06832, Phys. Rev. A 102, 022408
5. A quantum algorithm for measuring in the energy eigenbasis
   Rall, arXiv:2103.09717, Quantum 5 556

In the dissertation but not in the defense:
Efficiently simulating certain quantum circuits using classical Monte Carlo
Rall, Liang, Cook, Kretschmer, arXiv:1901.09070, Phys. Rev. A 99, 062337

Quantum Circuits

$$C := \;|0\rangle \qquad \left(\text{the column vector } \begin{bmatrix}1\\0\end{bmatrix}\right)$$

Quantum Circuits

$$C := \quad |0\rangle \qquad \left(\text{the column vector } \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$$

$$\text{or} \quad U \qquad \text{(a unitary matrix)}$$

Quantum Circuits

$$
\begin{aligned}
C := \;& |0\rangle && \left(\text{the column vector } \begin{bmatrix}1\\0\end{bmatrix}\right)\\
\text{or }\;& U && (\text{a unitary matrix})\\
\text{or }\;& C \cdot C && (\text{matrix multiplication})
\end{aligned}
$$

Quantum Circuits

$$
\begin{aligned}
C :=\ & |0\rangle && \left(\text{the column vector } \begin{bmatrix}1\\0\end{bmatrix}\right)\\
\text{or }\ & U && \text{(a unitary matrix)}\\
\text{or }\ & C \cdot C && \text{(matrix multiplication)}\\
\text{or }\ & C \otimes C && \text{(tensor product)}
\end{aligned}
$$

Quantum Circuits

$$
\begin{aligned}
C := \;& |0\rangle && \left(\text{the column vector } \begin{bmatrix}1\\0\end{bmatrix}\right)\\
\text{or } \;& U && \text{(a unitary matrix)}\\
\text{or } \;& C \cdot C && \text{(matrix multiplication)}\\
\text{or } \;& C \otimes C && \text{(tensor product)}\\
\text{or } \;& C^\dagger && \text{(adjoint / conjugate transpose).}
\end{aligned}
$$

The power of quantum circuits

$$|+\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot |0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The power of quantum circuits

$$|+\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot |0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$|+\rangle \otimes |+\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The power of quantum circuits

$$|+\rangle := \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\cdot|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

$$|+\rangle\otimes|+\rangle = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \qquad |+\rangle\otimes|+\rangle\otimes|+\rangle = \frac{1}{2\sqrt{2}}\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}$$

The power of quantum circuits

- General Monte Carlo situation: An event happens with probability p.

The power of quantum circuits

- General Monte Carlo situation: An event happens with probability p.
- Useful example: N possibilities, K considered 'good'.
  A random possibility is good with probability $p = K/N$.

The power of quantum circuits

- General Monte Carlo situation: An event happens with probability p.
- Useful example: N possibilities, K considered 'good'.
  A random possibility is good with probability $p = K/N$.
- Superposition of 2^n many possibilities using $|+\rangle^{\otimes n} = |+\rangle \otimes |+\rangle \otimes |+\rangle \ldots$.

The power of quantum circuits

- General Monte Carlo situation: An event happens with probability p.
- Useful example: N possibilities, K considered 'good'.
  A random possibility is good with probability $p = K/N$.
- Superposition of 2^n many possibilities using $|+\rangle^{\otimes n} = |+\rangle \otimes |+\rangle \otimes |+\rangle \ldots$.

"Try all possibilities at once?"

The power of quantum circuits

- General Monte Carlo situation: An event happens with probability p.
- Useful example: N possibilities, K considered 'good'.
  A random possibility is good with probability $p = K/N$.
- Superposition of 2^n many possibilities using $|+\rangle^{\otimes n} = |+\rangle \otimes |+\rangle \otimes |+\rangle \ldots$.

"Try all possibilities at once?"

Not quite so simple.

The power of quantum circuits

- General Monte Carlo situation: An event happens with probability p.
- Useful example: N possibilities, K considered 'good'.
  A random possibility is good with probability $p = K/N$.
- Superposition of 2^n many possibilities using $|+\rangle^{\otimes n} = |+\rangle \otimes |+\rangle \otimes |+\rangle \ldots$.

"Try all possibilities at once?"

Not quite so simple.

"The only difference between a probabilistic classical world
and the equations of the quantum world is that
somehow or other it appears as if **the probabilities would have to go negative.**"

Feynman 1981

Grover's search and negative probability

$|\Psi\rangle := |+\rangle^{\otimes n} \sim$ superposition over all possibilities $\quad \rightarrow \quad R_\Psi :=$ reflection about Ψ

$\Pi :=$ projection matrix onto good possibilities $\quad \rightarrow \quad R_\Pi :=$ reflection about Π

Grover's search and negative probability

$|\Psi\rangle := |+\rangle^{\otimes n} \sim$ superposition over all possibilities $\quad \rightarrow \quad R_\Psi :=$ reflection about Ψ

$\Pi :=$ projection matrix onto good possibilities $\quad \rightarrow \quad R_\Pi :=$ reflection about Π

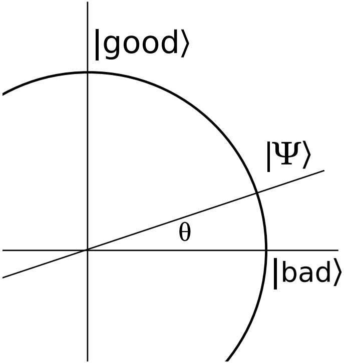
$$p := |\Pi |\Psi\rangle |^2 = \sin^2(\theta)$$

$$|\Psi\rangle = \sin(\theta) |\text{good}\rangle + \cos(\theta) |\text{bad}\rangle$$

Grover's search and negative probability

$$|\Psi\rangle := |+\rangle^{\otimes n} \sim \text{superposition over all possibilities} \quad \rightarrow \quad R_\Psi := \text{reflection about } \Psi$$

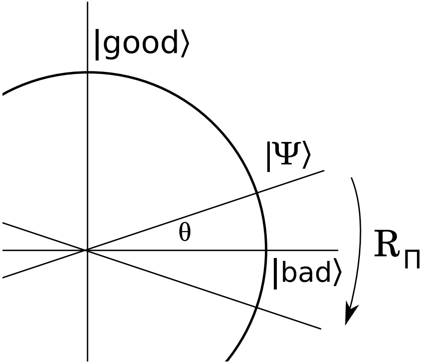
$$\Pi := \text{projection matrix onto good possibilities} \quad \rightarrow \quad R_\Pi := \text{reflection about } \Pi$$



Grover's search and negative probability

$$|\Psi\rangle := |+\rangle^{\otimes n} \sim \text{superposition over all possibilities} \quad \rightarrow \quad R_\Psi := \text{reflection about } \Psi$$

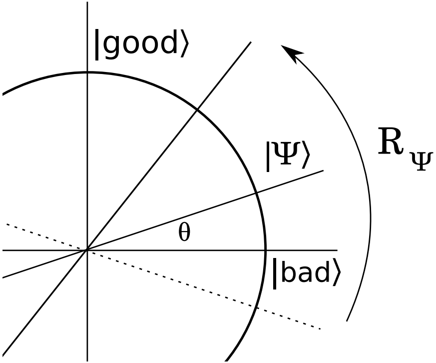
$$\Pi := \text{projection matrix onto good possibilities} \quad \rightarrow \quad R_\Pi := \text{reflection about } \Pi$$



Grover's search and negative probability

$|\Psi\rangle := |+\rangle^{\otimes n} \sim$ superposition over all possibilities $\quad\rightarrow\quad R_\Psi :=$ reflection about Ψ

$\Pi :=$ projection matrix onto good possibilities $\quad\rightarrow\quad R_\Pi :=$ reflection about Π



Grover's search and negative probability

$$|\Psi\rangle := |+\rangle^{\otimes n} \sim \text{superposition over all possibilities} \quad \rightarrow \quad R_\Psi := \text{reflection about } \Psi$$

$$\Pi := \text{projection matrix onto good possibilities} \quad \rightarrow \quad R_\Pi := \text{reflection about } \Pi$$

$$R_\Psi \cdot R_\Pi \cdot |\Psi\rangle = \sin(3\theta)\,|\text{good}\rangle + \cos(3\theta)\,|\text{bad}\rangle$$

$$(R_\Psi \cdot R_\Pi)^k \cdot |\Psi\rangle = \sin((2k+1)\theta)\,|\text{good}\rangle + \cos((2k+1)\theta)\,|\text{bad}\rangle$$

Grover's search and negative probability

$|\Psi\rangle := |+\rangle^{\otimes n} \sim$ superposition over all possibilities $\rightarrow$ $R_\Psi :=$ reflection about Ψ

$\Pi :=$ projection matrix onto good possibilities $\rightarrow$ $R_\Pi :=$ reflection about Π

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- An event happens with probability p.

$$\theta = \arcsin(\sqrt{1/p})$$

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- An event happens with probability p.

$$\theta = \arcsin(\sqrt{1/p})$$

- Useful example: N possibilities, K considered 'good'. Random guess: $p = K/N$.

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- An event happens with probability p.

$$\theta = \arcsin(\sqrt{1/p})$$

- Useful example: N possibilities, K considered 'good'. Random guess: $p = K/N$.
- **Search:** Make the event happen / find a good possibility.

$$r \sim 1/\theta \sim \sqrt{1/p}$$

$$\sin^2(r\theta) \sim \text{const}$$

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- An event happens with probability p.

$$\theta = \arcsin(\sqrt{1/p})$$

- Useful example: N possibilities, K considered 'good'. Random guess: $p = K/N$.
- **Counting:** Estimate p or K to accuracy ε.

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- An event happens with probability p.

$$\theta = \arcsin(\sqrt{1/p})$$

- Useful example: N possibilities, K considered 'good'. Random guess: $p = K/N$.
- **Counting:** Estimate p or K to accuracy ε.
- Many estimation tasks reducible to probability estimation. Montanaro arXiv:1504.06987
    - Statistical physics: canonical ensemble, grand canonical ensemble, etc.
    - Machine learning: stochastic integration, Bayesian networks
    - Finance: stochastic option pricing, financial derivatives
    - In this talk: quantum observables, density of states

Grover's Algorithm and Monte Carlo Estimation

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- An event happens with probability p.

$$\theta = \arcsin(\sqrt{1/p})$$

- Useful example: N possibilities, K considered 'good'. Random guess: $p = K/N$.
- **Counting:** Estimate p or K to accuracy ε.
- Many estimation tasks reducible to probability estimation. Montanaro arXiv:1504.06987
  - Statistical physics: canonical ensemble, grand canonical ensemble, etc.
  - Machine learning: stochastic integration, Bayesian networks
  - Finance: stochastic option pricing, financial derivatives
  - In this talk: quantum observables, density of states
- Traditional method: BHMT 1998

  Very complicated: uses quantum Fourier transform, conditional rotations, median amplification.

Quantum Approximate Counting, Simplified Aaronson, SOSA '20 24-32

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

Quantum Approximate Counting, Simplified Aaronson, SOSA '20 24-32

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- Estimating p is like estimating θ. Keep track of $\theta_{\text{min}}, \theta_{\text{max}}$ such that $\theta \in [\theta_{\text{min}}, \theta_{\text{max}}]$.

Quantum Approximate Counting, Simplified Aaronson, SOSA '20 24-32

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- Estimating p is like estimating θ. Keep track of $\theta_{\text{min}}, \theta_{\text{max}}$ such that $\theta \in [\theta_{\text{min}}, \theta_{\text{max}}]$.
- Start with $[\theta_{\text{min}}, \theta_{\text{max}}] = [0, \pi/2]$ and iteratively improve.

Quantum Approximate Counting, Simplified Aaronson, SOSA '20 24-32

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- Estimating p is like estimating θ. Keep track of $\theta_{\text{min}}, \theta_{\text{max}}$ such that $\theta \in [\theta_{\text{min}}, \theta_{\text{max}}]$.
- Start with $[\theta_{\text{min}}, \theta_{\text{max}}] = [0, \pi/2]$ and iteratively improve.
- Pick $r \sim 1/(\theta_{\text{max}} - \theta_{\text{min}})$ so that:

$$\begin{aligned} r\theta_{\text{min}} &\approx \pi k & r\theta_{\text{max}} &\approx \pi k + \frac{\pi}{2} \\ \sin^2(r\theta_{\text{min}}) &\approx 0 & \sin^2(r\theta_{\text{max}}) &\approx 1 \end{aligned}$$

Quantum Approximate Counting, Simplified [Aaronson, SOSA '20 24-32]

Grover coin

For every odd r, we can toss a coin:

$$\Pr[\text{coin comes up heads}] = \sin^2(r\theta)$$

- Estimating p is like estimating θ. Keep track of $\theta_{\text{min}}, \theta_{\text{max}}$ such that $\theta \in [\theta_{\text{min}}, \theta_{\text{max}}]$.
- Start with $[\theta_{\text{min}}, \theta_{\text{max}}] = [0, \pi/2]$ and iteratively improve.
- Pick $r \sim 1/(\theta_{\text{max}} - \theta_{\text{min}})$ so that:

$$r\theta_{\text{min}} \approx \pi k \qquad r\theta_{\text{max}} \approx \pi k + \frac{\pi}{2}$$

$$\sin^2(r\theta_{\text{min}}) \approx 0 \qquad \sin^2(r\theta_{\text{max}}) \approx 1$$

- Toss r-coin very many times to test if:

$$\sin^2(r\theta) \leq \frac{1}{3} \quad \text{or} \quad \sin^2(r\theta) \geq \frac{2}{3}$$

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
3. An introduction to block encodings
4. Quantum algorithms for estimating physical quantities
5. A quantum algorithm for measuring in the energy eigenbasis

Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
3. An introduction to block encodings
4. Quantum algorithms for estimating physical quantities
5. A quantum algorithm for measuring in the energy eigenbasis

How to study physics on a quantum computer?

Want to evaluate quantities like:

- Expectations of observables:

$$\langle O \rangle = \mathrm{Tr}(\rho O)$$

How to study physics on a quantum computer?

Want to evaluate quantities like:

- Expectations of observables:

$$\langle O \rangle = \mathrm{Tr}(\rho O)$$

$$\langle O_1(t_1) O_2(t_2) \ldots O_n(t_n) \rangle$$

How to study physics on a quantum computer?

Want to evaluate quantities like:

- Expectations of observables:

$$\langle O \rangle = \mathrm{Tr}(\rho O)$$

$$\langle O_1(t_1) O_2(t_2) \dots O_n(t_n) \rangle$$

- Density of states:

$$\rho(E) = \frac{1}{D} \sum_i \delta(E_i - E)$$

How to study physics on a quantum computer?

Want to evaluate quantities like:

- Expectations of observables:

$$\langle O \rangle = \mathrm{Tr}(\rho O)$$

$$\langle O_1(t_1) O_2(t_2) \ldots O_n(t_n) \rangle$$

- Density of states:

$$\rho(E) = \frac{1}{D} \sum_i \delta(E_i - E)$$

$\rightarrow$ Lots of non-unitary matrices: cannot write into a quantum circuit.

Block Encodings

- Main idea: encode M into top left corner of a unitary matrix.

$$U_M = \begin{bmatrix} M & \cdot \\ \cdot & \cdot \end{bmatrix} \qquad (\langle 0| \otimes I) U_M (|0\rangle \otimes I) = M$$

Block Encodings

- Main idea: encode M into top left corner of a unitary matrix.

$$U_M = \begin{bmatrix} M & \cdot \\ \cdot & \cdot \end{bmatrix} \qquad (\langle 0| \otimes I) U_M (|0\rangle \otimes I) = M$$

- Block encoding circuits: Gilyen, Low, Su, Wiebe, arXiv:1806.01838, STOC '19

$$B := M \qquad \text{(any matrix)}$$ New!

Block Encodings

- Main idea: encode M into top left corner of a unitary matrix.

$$U_M = \begin{bmatrix} M & \cdot \\ \cdot & \cdot \end{bmatrix} \qquad\qquad (\langle 0| \otimes I) U_M (|0\rangle \otimes I) = M$$

- Block encoding circuits: Gilyen, Low, Su, Wiebe, arXiv:1806.01838, STOC '19

$$\begin{aligned} B :=&\ M && \text{(any matrix)} \quad \boxed{\text{New!}} \\ \text{or}&\ B + B && \text{(matrix addition)} \quad \boxed{\text{New!}} \end{aligned}$$

Block Encodings

- Main idea: encode M into top left corner of a unitary matrix.

$$U_M = \begin{bmatrix} M & \cdot \\ \cdot & \cdot \end{bmatrix} \qquad\qquad (\langle 0| \otimes I) U_M (|0\rangle \otimes I) = M$$

- Block encoding circuits: Gilyen, Low, Su, Wiebe, arXiv:1806.01838, STOC '19

$$\begin{aligned} B := \quad & M && \text{(any matrix)} \quad \boxed{\text{New!}} \\ \text{or} \quad & B + B && \text{(matrix addition)} \quad \boxed{\text{New!}} \\ \text{or} \quad & B \cdot B && \text{(matrix multiplication)} \\ \text{or} \quad & B \otimes B && \text{(tensor product)} \\ \text{or} \quad & B^\dagger && \text{(adjoint / conjugate transpose)} \end{aligned}$$

Block Encodings

- Main idea: encode M into top left corner of a unitary matrix.

$$U_M = \begin{bmatrix} M & \cdot \\ \cdot & \cdot \end{bmatrix} \qquad\qquad (\langle 0| \otimes I) U_M (|0\rangle \otimes I) = M$$

- Block encoding circuits: Gilyen, Low, Su, Wiebe, arXiv:1806.01838, STOC '19

$$\begin{aligned}
B :=\ & M && \text{(any matrix)} \quad \boxed{\text{New!}} \\
\text{or}\ & B + B && \text{(matrix addition)} \quad \boxed{\text{New!}} \\
\text{or}\ & B \cdot B && \text{(matrix multiplication)} \\
\text{or}\ & B \otimes B && \text{(tensor product)} \\
\text{or}\ & B^\dagger && \text{(adjoint / conjugate transpose)} \\
\text{or}\ & p(B) && \text{(transform eigenvalues by a polynomial } p\text{).} \quad \boxed{\text{New!}}
\end{aligned}$$

Block Encodings

- Main idea: encode M into top left corner of a unitary matrix.

$$U_M = \begin{bmatrix} M & \cdot \\ \cdot & \cdot \end{bmatrix} \qquad\qquad (\langle 0| \otimes I) U_M (|0\rangle \otimes I) = M$$

- Block encoding circuits: Gilyen, Low, Su, Wiebe, arXiv:1806.01838, STOC '19

$$\begin{aligned}
B := \ & M && \text{(any matrix)} \quad \text{New!} \\
\text{or} \ & B + B && \text{(matrix addition)} \quad \text{New!} \\
\text{or} \ & B \cdot B && \text{(matrix multiplication)} \\
\text{or} \ & B \otimes B && \text{(tensor product)} \\
\text{or} \ & B^\dagger && \text{(adjoint / conjugate transpose)} \\
\text{or} \ & p(B) && \text{(transform eigenvalues by a polynomial } p\text{).} \quad \text{New!}
\end{aligned}$$

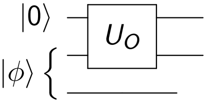
- Linear combinations of Paulis $\rightarrow$ block encodings of any Hamiltonian or observable

Estimating observables: $\mathrm{Tr}(\rho O)$ Rall, arXiv:2004.06832, Phys. Rev. A 102, 022408

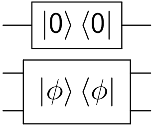
- Given a block encoding U_O of O, and $|\phi\rangle$, a purification of ρ

- Given a block encoding U_O of O, and $|\phi\rangle$, a purification of ρ

$|\Psi\rangle :=$

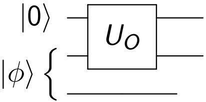


$\Pi :=$

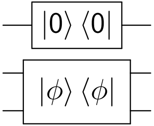


- Given a block encoding U_O of O, and $|\phi\rangle$, a purification of ρ

$|\Psi\rangle :=$



$\Pi :=$



- "Approximate counting simplified" gives an estimate of:

$$\begin{aligned} p := |\Pi\,|\Psi\rangle\,|^2 &= |\,\langle 0|\,\langle\phi|\,(U_O \otimes I)\,|0\rangle\,|\phi\rangle\,|^2 \\ &= |\,\langle\phi|\,(O \otimes I)\,|\phi\rangle\,|^2 \\ &= |\mathrm{Tr}\,[|\phi\rangle\,\langle\phi|\,(O \otimes I)]\,|^2 = |\mathrm{Tr}\,(\rho O)\,|^2 \end{aligned}$$

n-time correlation functions

- An observable in Heisenberg picture:

$$O_i(t_i) = e^{iHt_i} O_i e^{-iHt_i}$$

n-time correlation functions

- An observable in Heisenberg picture:

$$O_i(t_i) = e^{iHt_i} O_i e^{-iHt_i}$$

- To estimate $\langle O_1(t_1) O_2(t_2) ... O_n(t_n) \rangle$, construct block encoding of:

$$\Gamma = \prod_i O_i(t_i) = e^{iHt_1} O_1 e^{iH(t_2 - t_1)} O_2 e^{iH(t_3 - t_2)} ... O_n e^{iHt_n}$$

n-time correlation functions

- An observable in Heisenberg picture:

$$O_i(t_i) = e^{iHt_i} O_i e^{-iHt_i}$$

- To estimate $\langle O_1(t_1) O_2(t_2) \ldots O_n(t_n) \rangle$, construct block encoding of:

$$\Gamma = \prod_i O_i(t_i) = e^{iHt_1} O_1 e^{iH(t_2 - t_1)} O_2 e^{iH(t_3 - t_2)} \ldots O_n e^{iHt_n}$$

- Γ is not Hermitian, so expectation is complex.

$$\text{Real part:} \qquad \mathrm{Tr}\left(\rho \frac{\Gamma + \Gamma^\dagger}{2}\right)$$

$$\text{Imaginary part:} \qquad \mathrm{Tr}\left(\rho \frac{\Gamma - \Gamma^\dagger}{2i}\right)$$

Density of states

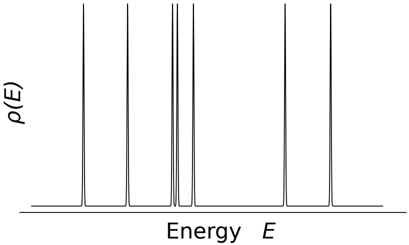
- Density of states:

$$\rho(E) = \frac{1}{D} \sum_i \delta(E_i - E)$$

Density of states

- Density of states:

$$\rho(E) = \frac{1}{D}\sum_i \delta(E_i - E)$$



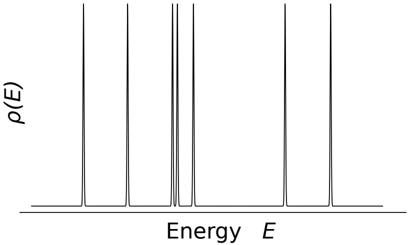
Density of states

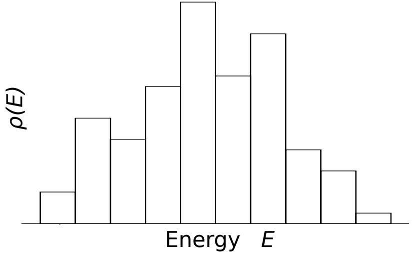
- Density of states:

$$\rho(E) = \frac{1}{D}\sum_i \delta(E_i - E)$$

- Histogram bin:

$$\int_{E_1}^{E_2} \rho(E)dE$$





Density of states

- Density of states:

$$\rho(E) = \frac{1}{D}\sum_i \delta(E_i - E)$$

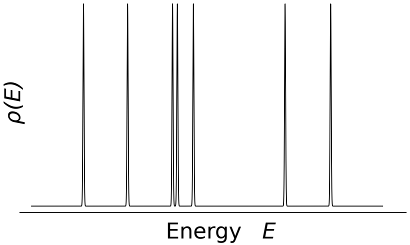
- Histogram bin:

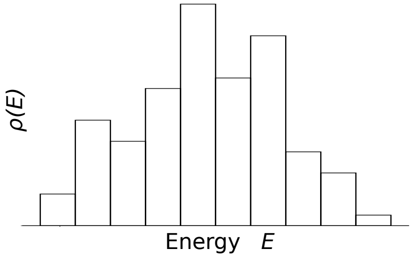
$$\int_{E_1}^{E_2} \rho(E)dE$$

- Map to observable estimation:

$$\int_{E_1}^{E_2} \rho(E)dE = \mathrm{Tr}\left(\frac{I}{D}\cdot w(H)\right)$$

$w(E) :=$ polynomial indicating $E_1 \leq E \leq E_2$





Outline

1. An introduction to quantum computing and Grover's algorithm
2. Simplified quantum Monte Carlo estimation
3. An introduction to block encodings
4. Quantum algorithms for estimating physical quantities
5. A quantum algorithm for measuring in the energy eigenbasis

Outline

Estimating Energy in Superposition

- Previous methods estimate the *average* energy $\langle H \rangle$.

Estimating Energy in Superposition

- Previous methods estimate the *average* energy $\langle H \rangle$.
- What about *measuring* the Hamiltonian H?

Estimating Energy in Superposition

- Previous methods estimate the *average* energy $\langle H \rangle$.
- What about *measuring* the Hamiltonian H?
  Input:

$$H = \sum_j E_j \left|\psi_j\right\rangle \left\langle\psi_j\right|$$

Estimating Energy in Superposition

- Previous methods estimate the *average* energy $\langle H \rangle$.
- What about *measuring* the Hamiltonian H?
  Input:

$$H = \sum_j E_j \left|\psi_j\right\rangle \left\langle\psi_j\right|$$

Output: a quantum circuit that transforms

$$\sum_j \alpha_j \left|\psi_j\right\rangle \quad \rightarrow \quad \left|\psi_j\right\rangle \left|E_j\right\rangle \text{ with probability } |\alpha_j|^2$$

$$\left|E_j\right\rangle := n\text{-bit estimate of } E_j$$

Estimating Energy in Superposition

- Previous methods estimate the *average* energy $\langle H \rangle$.
- What about *measuring* the Hamiltonian H?
  Input:

$$H = \sum_j E_j \left|\psi_j\right\rangle \left\langle\psi_j\right|$$

Output: a quantum circuit that transforms

$$\sum_j \alpha_j \left|\psi_j\right\rangle \quad \rightarrow \quad \left|\psi_j\right\rangle \left|E_j\right\rangle \text{ with probability } |\alpha_j|^2$$

$$\left|E_j\right\rangle := n\text{-bit estimate of } E_j$$

- 'Textbook' method: Phase Estimation Nielsen, Chuang
  Uses Hamiltonian simulation, quantum Fourier transform, median amplification, and a quantum sorting network.

Block Measurement Lemma

Say we have a block encoding of a projector Π.
Say $|\psi\rangle$ splits into:

$$|\psi\rangle = \alpha\,|\text{good}\rangle + \beta\,|\text{bad}\rangle$$

Then, we can measure the projector Π:

$$|\psi\rangle \quad \to \quad |\text{good}\rangle\,|1\rangle \;\text{ or }\; |\text{bad}\rangle\,|0\rangle$$

Block Measurement Lemma

Say we have a block encoding of a projector Π.
Say $|\psi\rangle$ splits into:

$$|\psi\rangle = \alpha\,|\text{good}\rangle + \beta\,|\text{bad}\rangle$$

Then, we can measure the projector Π:

$$|\psi\rangle \quad \rightarrow \quad |\text{good}\rangle\,|1\rangle \;\text{ or }\; |\text{bad}\rangle\,|0\rangle$$

- Goal: n-bit approximation of E_j

Block Measurement Lemma

Say we have a block encoding of a projector Π.
Say $|\psi\rangle$ splits into:

$$|\psi\rangle = \alpha\,|\text{good}\rangle + \beta\,|\text{bad}\rangle$$

Then, we can measure the projector Π:

$$|\psi\rangle \quad \to \quad |\text{good}\rangle\,|1\rangle \;\text{ or }\; |\text{bad}\rangle\,|0\rangle$$

- Goal: n-bit approximation of E_j
- Make a projector for each bit $k \in \{1, \ldots, n\}$:

$$\Pi_k = \sum_j k\text{'th bit of } E_j \cdot |\psi_j\rangle\langle\psi_j|$$

Block Measurement Lemma

Say we have a block encoding of a projector Π.
Say $|\psi\rangle$ splits into:

$$|\psi\rangle = \alpha\,|\text{good}\rangle + \beta\,|\text{bad}\rangle$$

Then, we can measure the projector Π:

$$|\psi\rangle \quad \to \quad |\text{good}\rangle\,|1\rangle \;\text{ or }\; |\text{bad}\rangle\,|0\rangle$$

- Goal: n-bit approximation of E_j
- Make a projector for each bit $k \in \{1, \ldots, n\}$:

$$\Pi_k = \sum_j k\text{'th bit of } E_j \cdot |\psi_j\rangle\langle\psi_j| \approx \sum_j p_k(E_j) \cdot |\psi_j\rangle\langle\psi_j|$$

Block Measurement Lemma

Say we have a block encoding of a projector Π.
Say $|\psi\rangle$ splits into:

$$|\psi\rangle = \alpha\,|\text{good}\rangle + \beta\,|\text{bad}\rangle$$

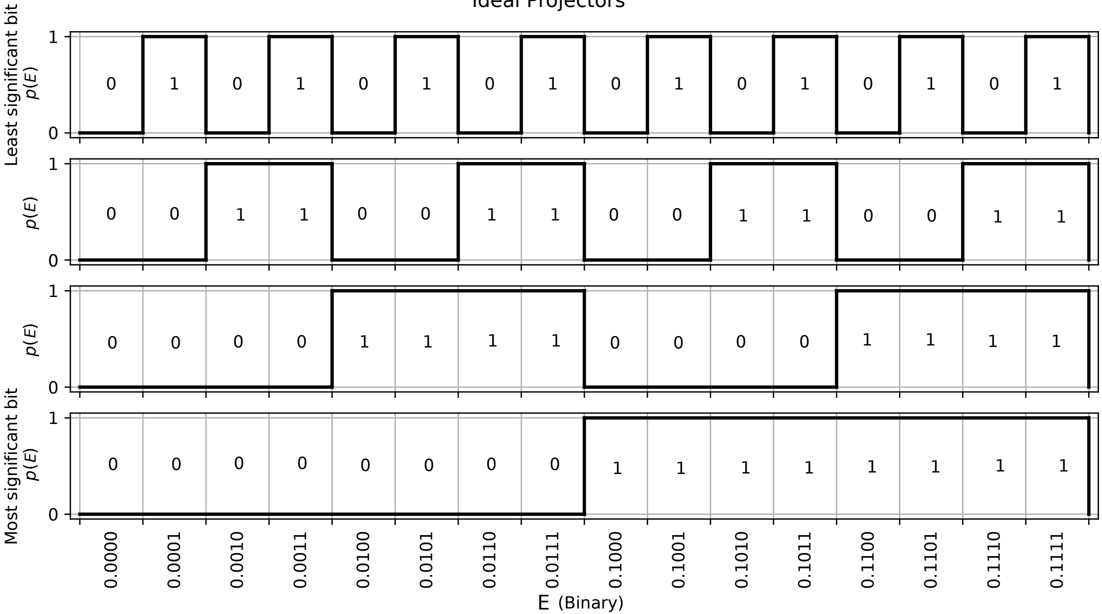
Then, we can measure the projector Π:

$$|\psi\rangle \quad \to \quad |\text{good}\rangle\,|1\rangle \;\text{ or }\; |\text{bad}\rangle\,|0\rangle$$

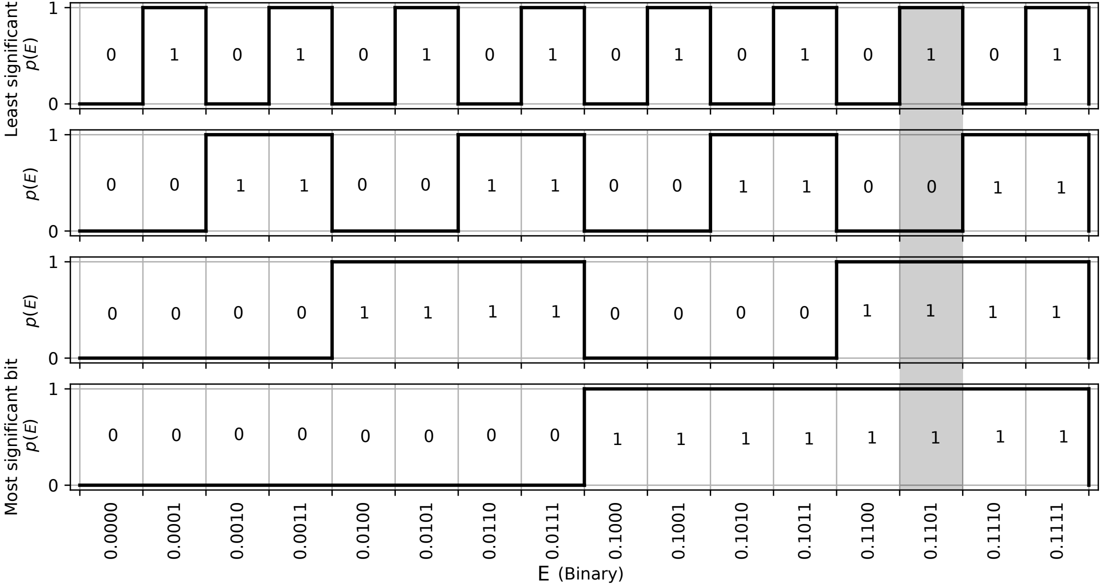
- Goal: n-bit approximation of E_j
- Make a projector for each bit $k \in \{1, ..., n\}$:

$$\Pi_k = \sum_j k\text{'th bit of } E_j \cdot |\psi_j\rangle\langle\psi_j| \approx \sum_j p_k(E_j) \cdot |\psi_j\rangle\langle\psi_j| = p_k(H)$$

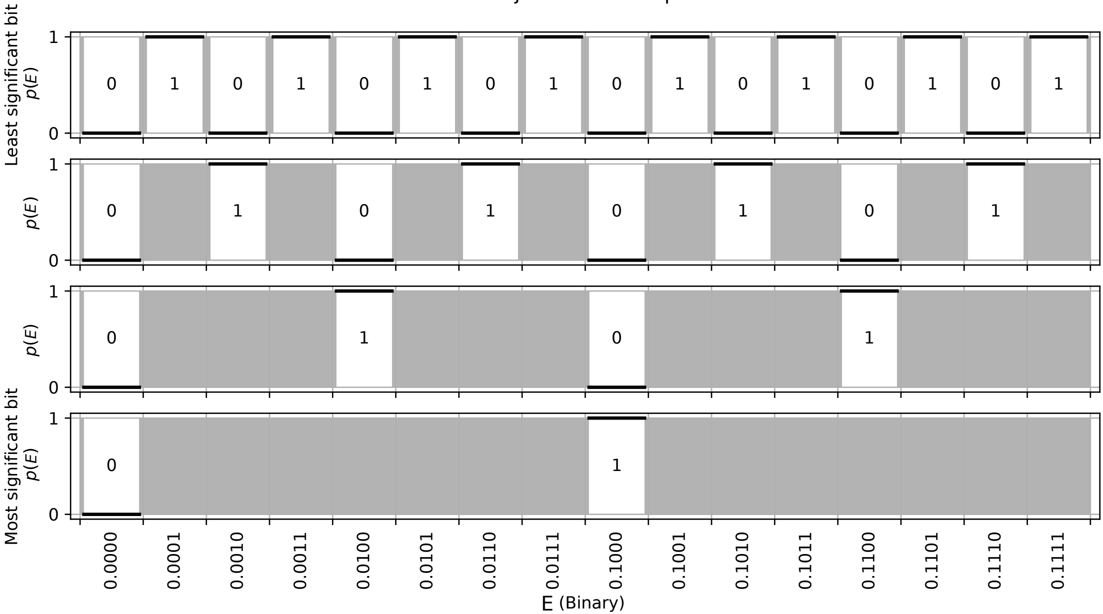
Ideal Projectors



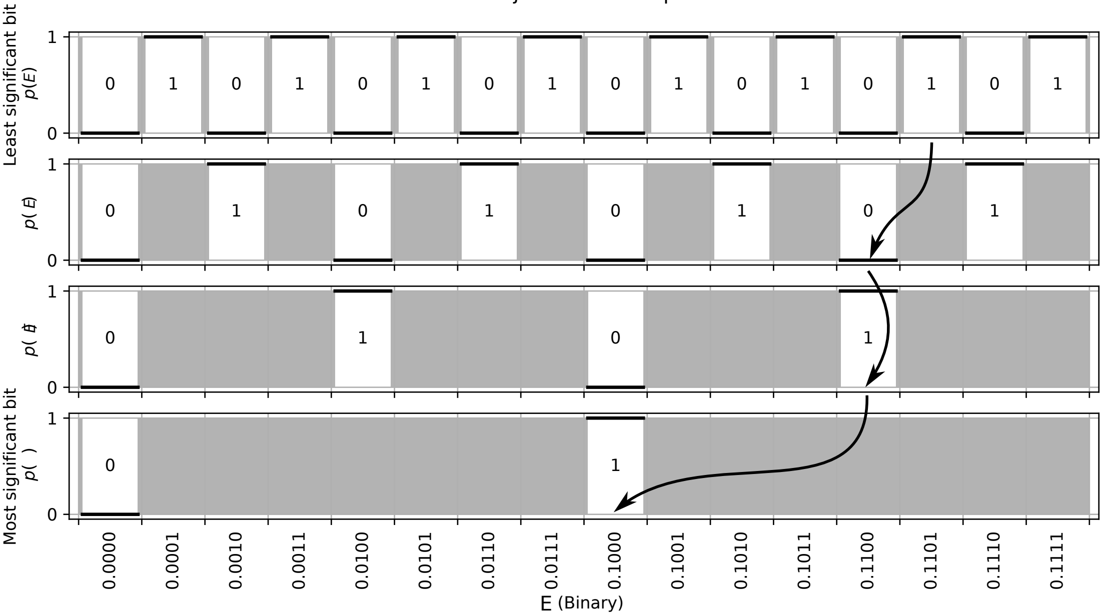
Ideal Projectors

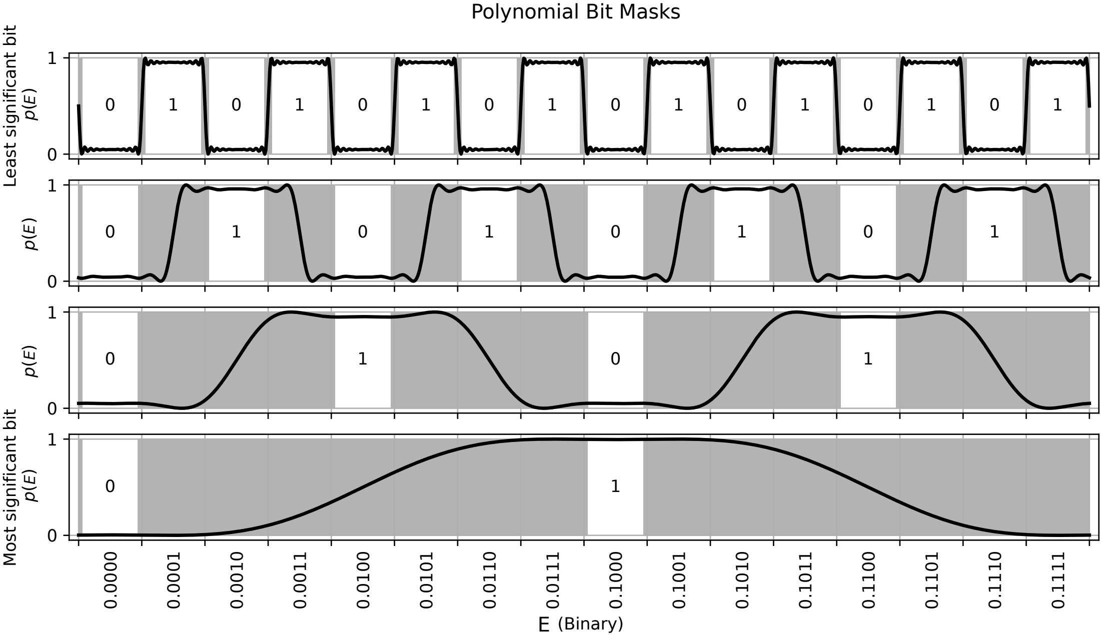


Projectors with Gaps



Projectors with Gaps





Polynomial Bit Masks
Least significant bit
p(E)
Most significant bit
0
1
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1
0 1 0 1
0 1
0.0000
0.0001
0.0010
0.0011
0.0100
0.0101
0.0110
0.0111
0.1000
0.1001
0.1010
0.1011
0.1100
0.1101
0.1110
0.1111
E (Binary)

Advantages over traditional energy estimation:

- $\sim 20x$ constant factor speedup

Performance and Applications

Advantages over traditional energy estimation:

- $\sim 20x$ constant factor speedup
- Ancilla cost: $O(n)$, independent of accuracy

Performance and Applications

Advantages over traditional energy estimation:

- $\sim 20x$ constant factor speedup
- Ancilla cost: $O(n)$, independent of accuracy

Applications:

- Thermal state preparation: Quantum Metropolis Sampling
  Temme et al arXiv:0911.3635 Yung, Guzik arXiv:1011.1468 Lemieux et al arXiv:1910.01659

Performance and Applications

Advantages over traditional energy estimation:

- $\sim 20x$ constant factor speedup
- Ancilla cost: $O(n)$, independent of accuracy

Applications:

- Thermal state preparation: Quantum Metropolis Sampling
  Temme et al arXiv:0911.3635 Yung, Guzik arXiv:1011.1468 Lemieux et al arXiv:1910.01659
- Non-Destructive Amplitude Estimation: Partition function estimation, Bayesian inference
  Harrow, Wei arXiv:1908.10856 Arunachalam et al arXiv:2009.11270

Takeaways

- Quantum computers offer a general purpose speedup for Monte Carlo estimation

  Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32

  Next steps: non-destructive estimation, hybrid quantum-classical estimation

Takeaways

- Quantum computers offer a general purpose speedup for Monte Carlo estimation
  Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32
  Next steps: non-destructive estimation, hybrid quantum-classical estimation

- Block encodings make it easy to estimate quantities from physics on a quantum computer
  Rall, arXiv:2004.06832, Phys. Rev. A 102, 022408
  Next steps: enhanced block encodings with quantum channels, and multivariate polynomials

Takeaways

- Quantum computers offer a general purpose speedup for Monte Carlo estimation
  Aaronson, Rall arXiv:1908.10846, SOSA '20 24-32
  Next steps: non-destructive estimation, hybrid quantum-classical estimation
- Block encodings make it easy to estimate quantities from physics on a quantum computer
  Rall, arXiv:2004.06832, Phys. Rev. A 102, 022408
  Next steps: enhanced block encodings with quantum channels, and multivariate polynomials
- Energy measurement is simpler and more efficient using eigenvalue transformation
  Rall, arXiv:2103.09717, Quantum 5 556
  Next steps: modernize the quantum algorithms literature with eigenvalue transformation

Thank you for your attention!

Special thanks to:

Scott Aaronson,
All of my family, especially Stella Wang and Anna Maria Rall,
Eric Price, Elena Caceres, Allan MacDonald,
Elaine Li, Brian La Cour, Antia Lamas-Linares
Corey Ostrove, Bryce Fuller, Daniel Liang, William Kretschmer,
Andrew Tan, Adrian Trejo Nuñes, Justin Yirka, Yosi Atia, Chunhao Wang,
and many others.

I was supported by Dr. Aaronson's Vannevar Bush Faculty Fellowship.

Contact me at `patrickjrall@gmail.com`