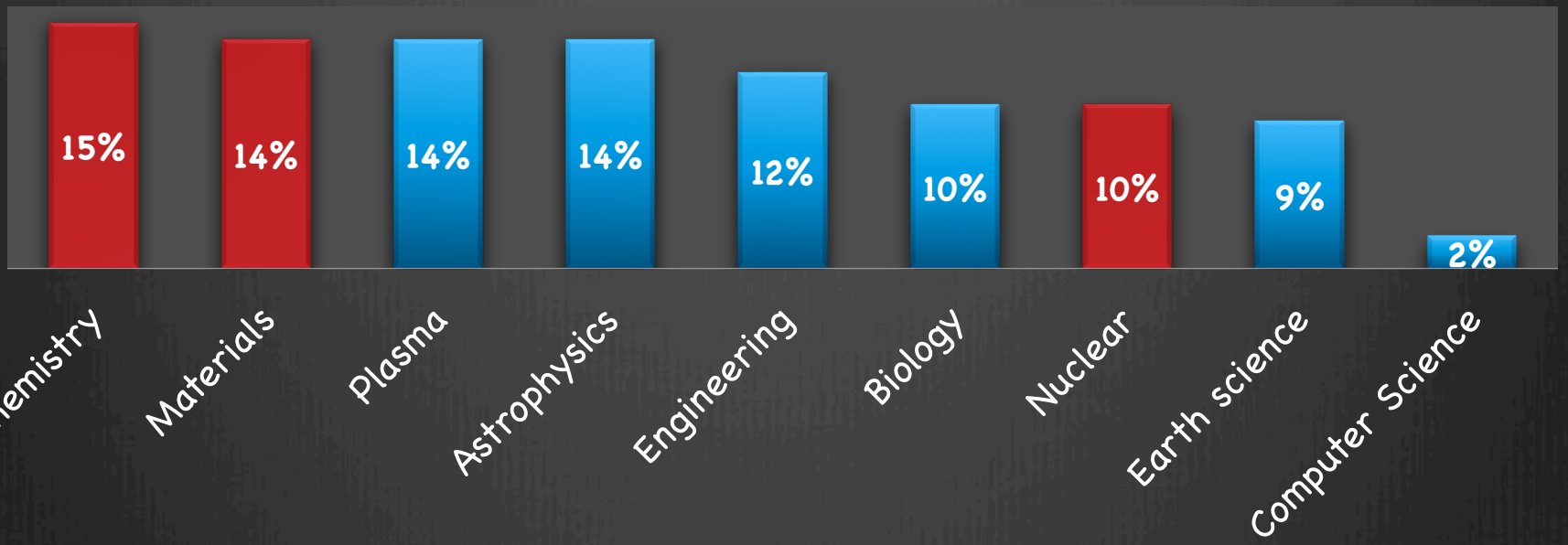
What to do with a small quantum computer?

Aram Harrow (MIT Physics)

IPAM
Dec 6, 2016

simulating quantum mechanics is hard!

DOE (INCITE) supercomputer time by category



Implications:

1. Quantum mechanics is hard to simulate classically.

2. Quantum computers can do this simulation exponentially more efficiently.

3. Maybe they can do other things more efficiently too?

Motivation

Practical

- Social/economic value of computing
- End of Moore's Law

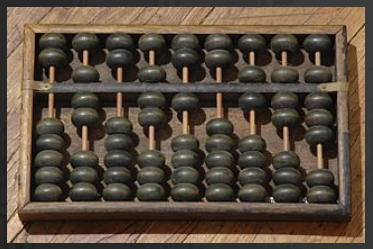
Foundational

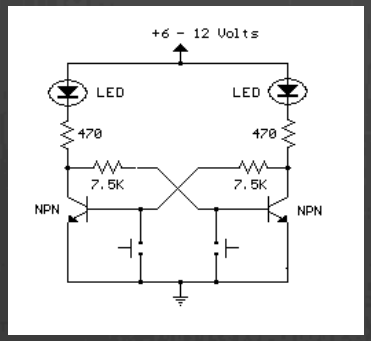
- Nature of information and computing

classical computers

n bits = 2^n states

1	0	1	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	1	0	0	1	1





gates

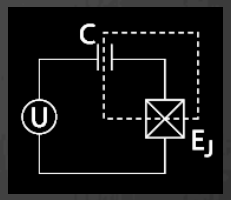
quantum computers

qubit = two-level system

nuclear spin

photon polarization

ion e⁻ states



superconducting charge or flux

...

1 qubit = 2 dimensions $|\psi\rangle \in \mathbb{C}^2$

n qubits = 2^n dimensions $|\psi\rangle \in \mathbb{C}^{2^n}$

Schrödinger's eq:

$$\frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle$$

quantum gates

$$I \otimes \ldots \otimes I \otimes \underbrace{U}_{\text{one- or two-qubit}} \otimes I \otimes \ldots \otimes I$$

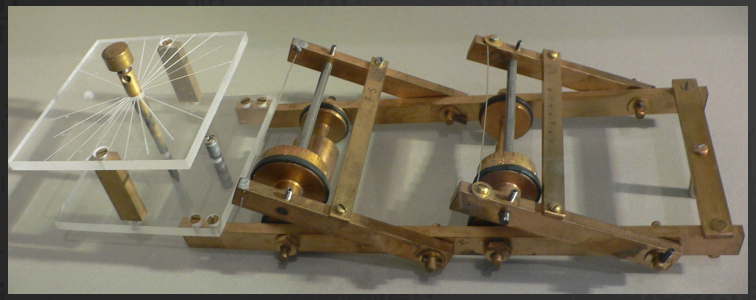
randomized computation

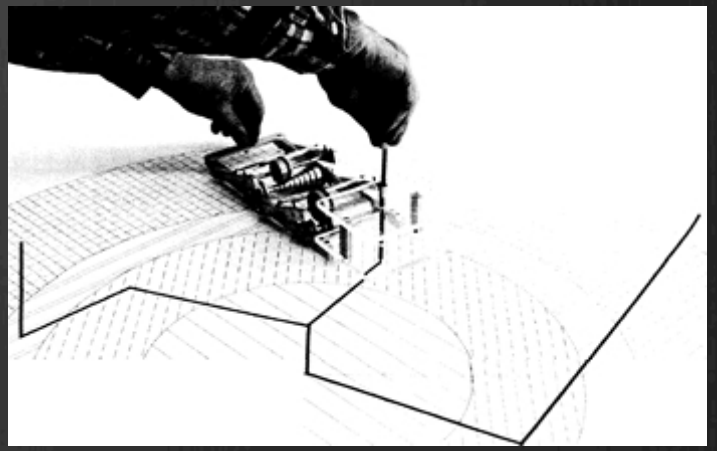
+

$$\begin{pmatrix} p_{0000} \\ p_{0001} \\ \vdots \\ p_{1111} \end{pmatrix} \in \mathbb{R}^{2^n}$$

Application: high-dimensional integrals

$$\iiint \cdots \int dx_1 \ldots dx_n f(x_1, \ldots, x_n)$$





FERMIAC

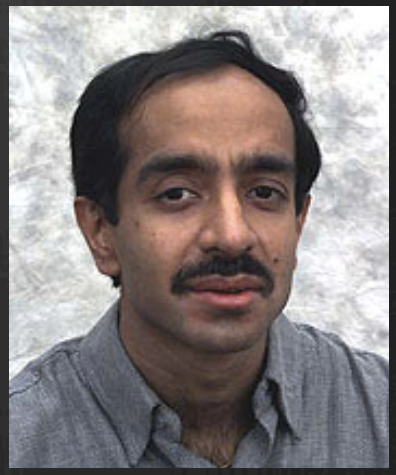
unstructured search

x	f(x)
1	0
2	0
3	0
4	1
5	0
...	...
N	0

Given the ability to compute f, find x such that f(x)=1.

classical: O(N) time needed

Grover's algorithm: (1996)
$O(\sqrt{N})$ on quantum computer



similar speedups for maximizing, approximate counting, collisions, triangle finding, game-tree evaluation, backtracking algorithms ...

factoring

213981239012389123712938721903871239871238917239817391027129083712988964375846896675365871650236478923164871234629187462312389746213487612387461238 = ? × ?

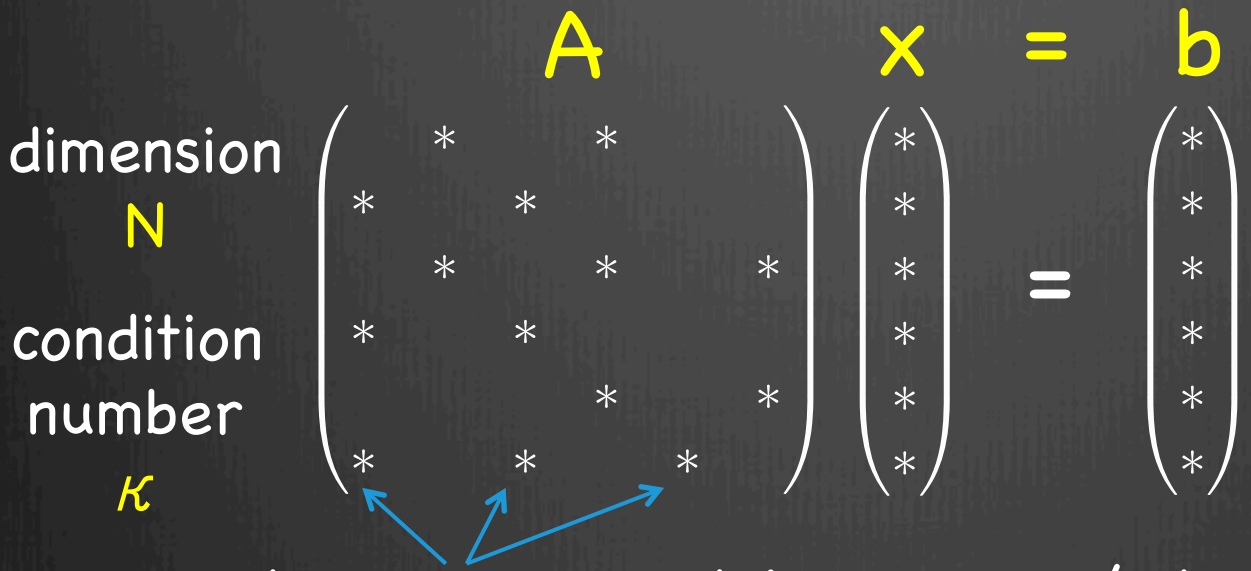
n digits

Best classical algorithm: time $O(\exp(n^{1/3}))$

Shor's algorithm: (1994)
time $O(n^2)$ on a quantum computer

(similar for discrete log, elliptic curves, ...)

linear systems of equations



Classical:
$O(Ns\kappa)$

Quantum:
$O(\log(N)s\kappa)$

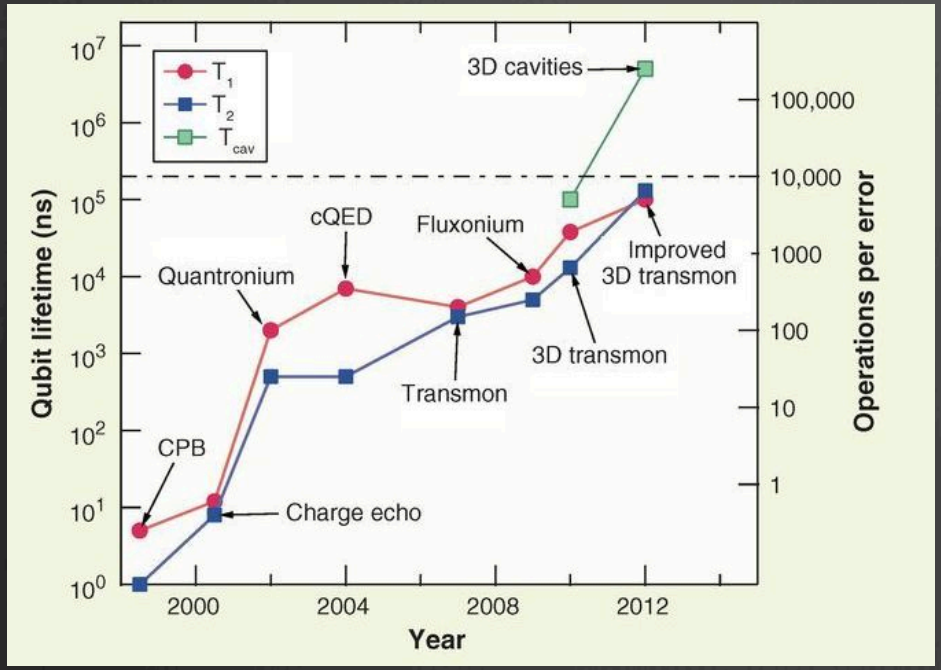
A computable on the fly
input $|b\rangle$ and output $|x\rangle$

Applications:

- ODEs
- PDEs
- regression
- machine learning

experimental QC

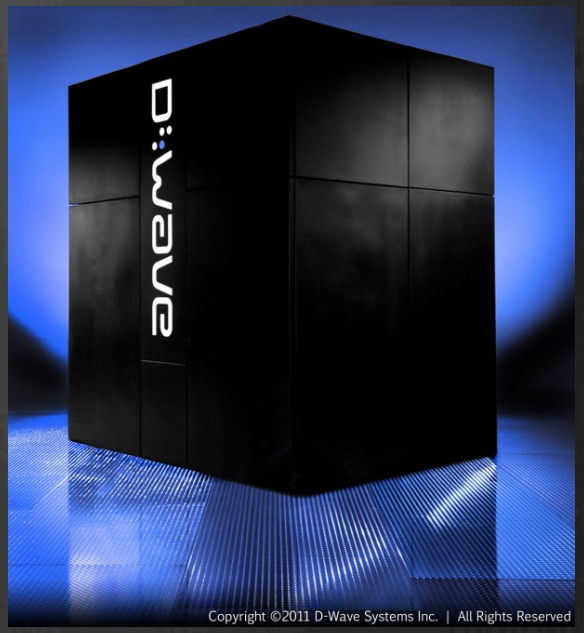
1-2 clean qubits



IBM, Google, Microsoft, Intel, Alibaba, BBN, ...
Rigetti, psitech, ionQ, Quantum Circuits Inc, ...

proposals for ≈50-qubit universal QCs in 2-10 years using superconductors and ion traps

≈1000 noisy qubits



D-Wave
(1qbit, QxBranch, Lockheed Martin)

combinatorial optimization

Problem: minimize

$f(x_1,\ldots,x_n) = \Sigma$ (terms involving a few variables)

Applications:

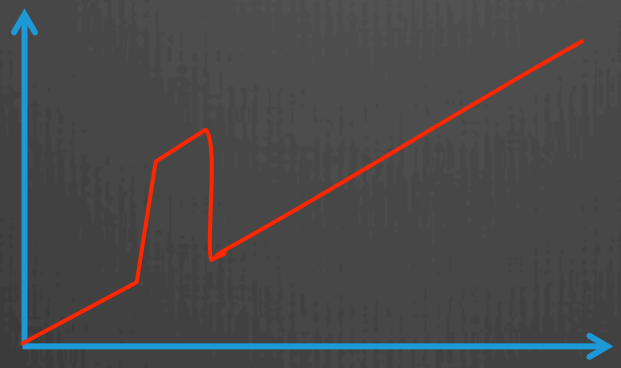
- minimum energy of classical systems
- best way for a protein to fold
- best model in a family explaining observed data
- math proof with fewest errors
- …

Difficulties:

- exponentially many local minima
- NP-complete even for good approximations

rest of the talk

classical optimization algorithms



adiabatic optimization

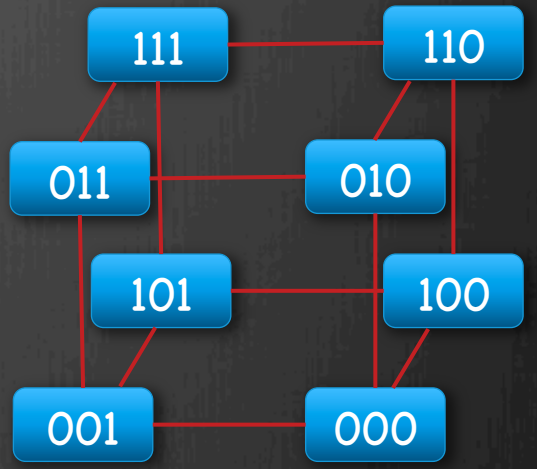
variational algorithms and quantum supremacy

classical local search

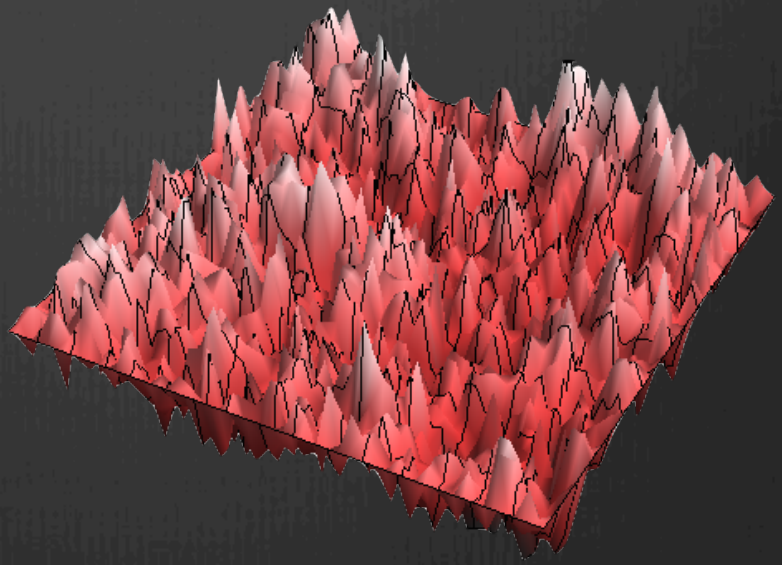
Local search

Start at point x

- Choose a neighbor y
- Move to y if f(x) > f(y)
- Repeat



problem:
local minima



classical simulated annealing

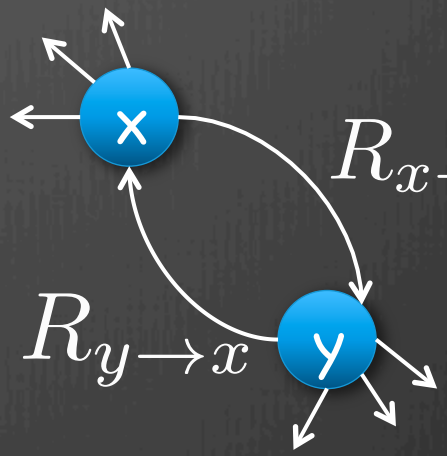
goal: sample from Gibbs distribution

$$\Pr[x] = \frac{e^{-\frac{f(x)}{T}}}{Z}$$

annealing

gradually lower T

Metropolis algorithm



$$\frac{R_{x\to y}}{R_{y\to x}} = e^{\frac{f(x)-f(y)}{T}}$$

T=∞: unbiased random walk

T=0: local search

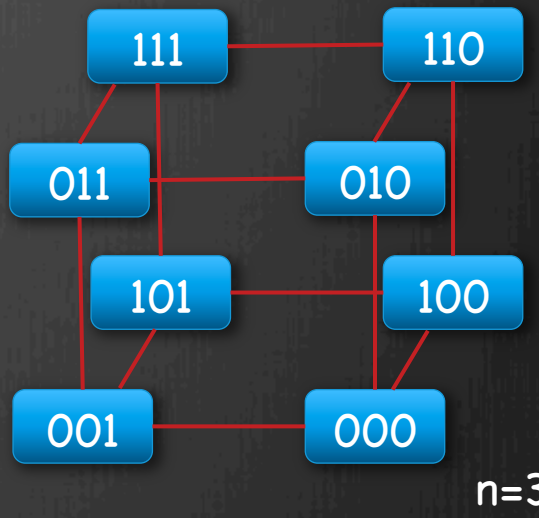
Walks on the hypercube

$\{0,1\}^n$ = "n-dimensional hypercube"

adjacency matrix A

A_{xy} = 1 if (x,y) connected, 0 if not

	000	001	010	011	100	101	110	111
000	0	1	1	0	1	0	0	0
001	1	0	0	1	0	1	0	0
010	1	0	0	1	0	0	1	0
011	0	1	1	0	0	0	0	1
100	1	0	0	0	0	1	1	0
101	0	1	0	0	1	0	0	1
110	0	0	1	0	1	0	0	1
111	0	0	0	1	0	1	1	0



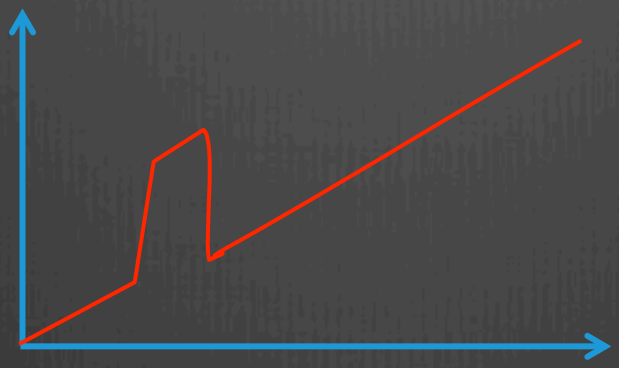
Graph Laplacian: L = n I − A

continuous-time random walk: dp/dt = -Lp ➔ uniform distribution

Simulated annealing: $\frac{dp}{dt} = -\mathrm{diag}(e^{\frac{f}{2T}}) \cdot L \cdot \mathrm{diag}(e^{-\frac{f}{2T}})p$

overview

classical optimization algorithms



adiabatic optimization

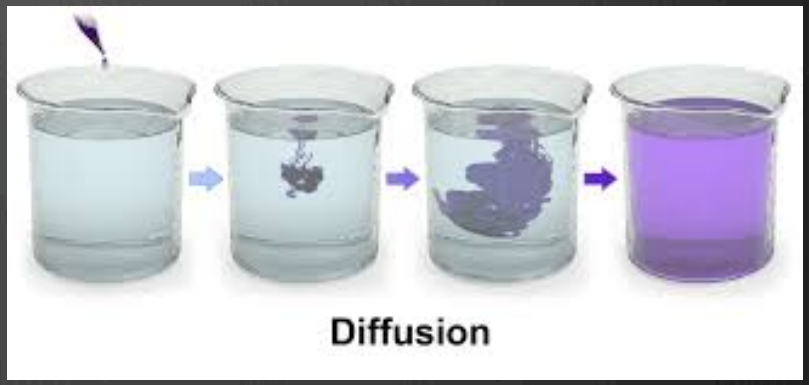
variational algorithms and quantum supremacy

quantum mechanics to the rescue

Diffusion equation:

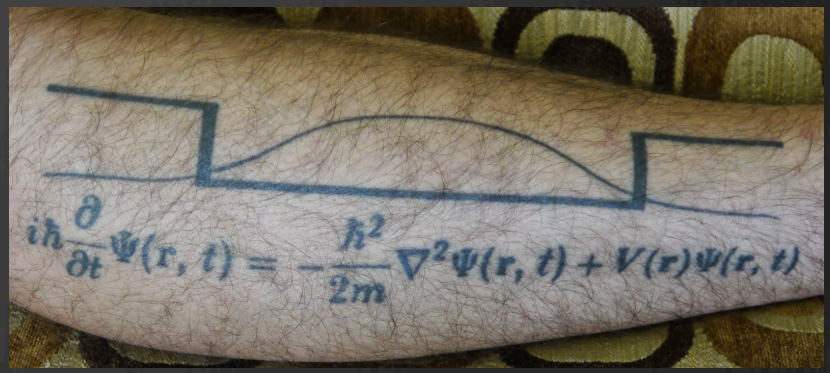
$$\frac{d}{dt}p(\vec{r},t) = -Lp(\vec{r},t)$$

$$L = \vec{\nabla}^2$$



Schrödinger equation:

$$\frac{d}{dt}\psi(\vec{r},t) = iL\psi(\vec{r},t)$$



math-physics dictionary

eigenvectors of H ⟷ quantum states

eigenvalues of H ⟷ energies

eigenvector with lowest eigenvalue ⟷ ground state

Schrödinger's eq:

$$\frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle$$

⟷ energy = frequency

adiabatic algorithm

Schrödinger's eq:

$$\frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle$$

$H_{initial}$ —— $H(t)$ ——→ H_{final}

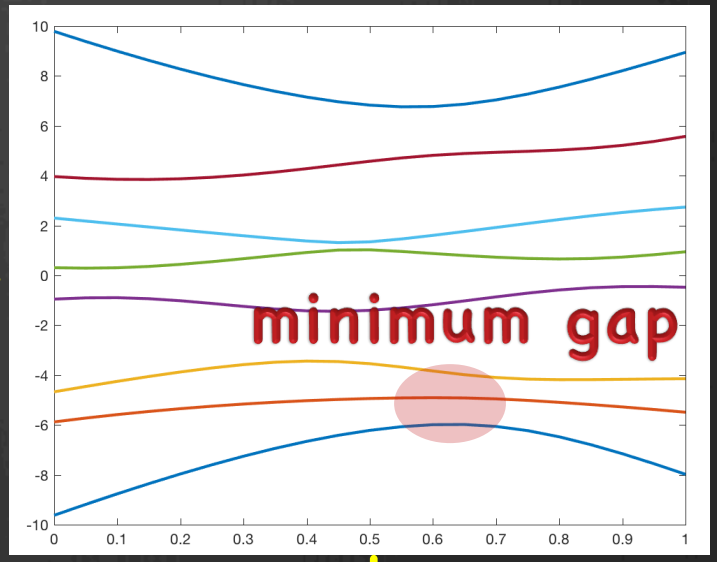
$|$ground state (t=0)$\rangle$ ——→ $|$ground state (t=T_{final})$\rangle$

easy to prepare

desired output

Adiabatic theorem:
Slowly changing H(t)
→ ≈ stay in ground state

$H_{initial}$



H_{final}

adiabatic optimization

Schrödinger's eq:

$$\frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle$$

$$H(t) = \left(1 - \frac{t}{T_{\text{final}}}\right) L + \frac{t}{T_{\text{final}}} \text{diag}(f)$$

$H_{initial} = L$ → $H_{final} = \text{diag}(f)$

$|\text{ground state } (t=0)\rangle$ → $|\text{ground state } (t=T_{final})\rangle$

easy to prepare

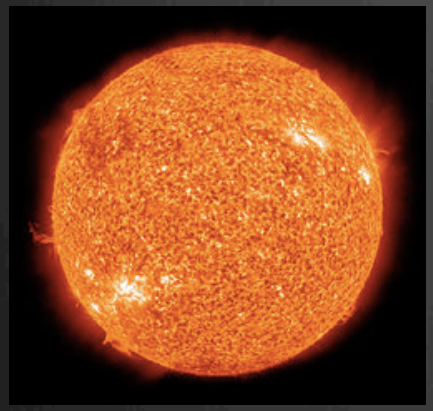
desired output

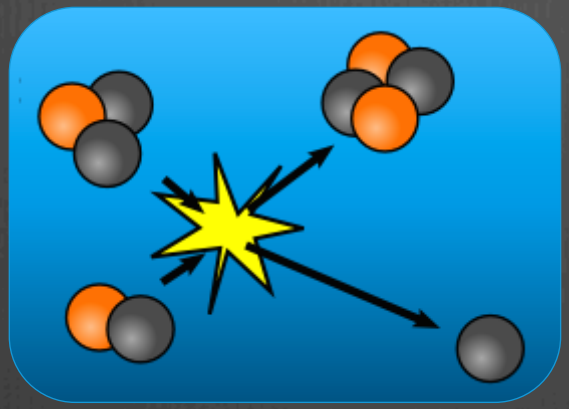
$$\frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

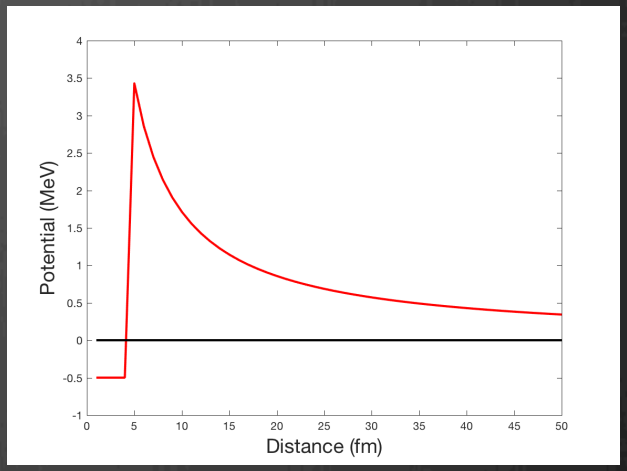
$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$\arg\min_x f(x)$

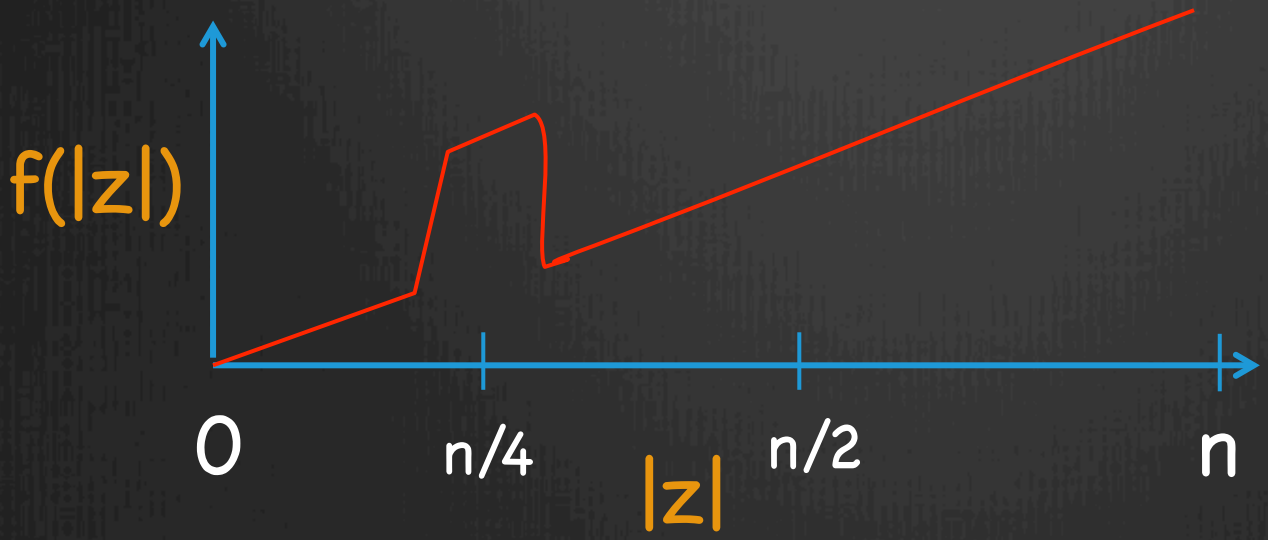
tunneling and optimization





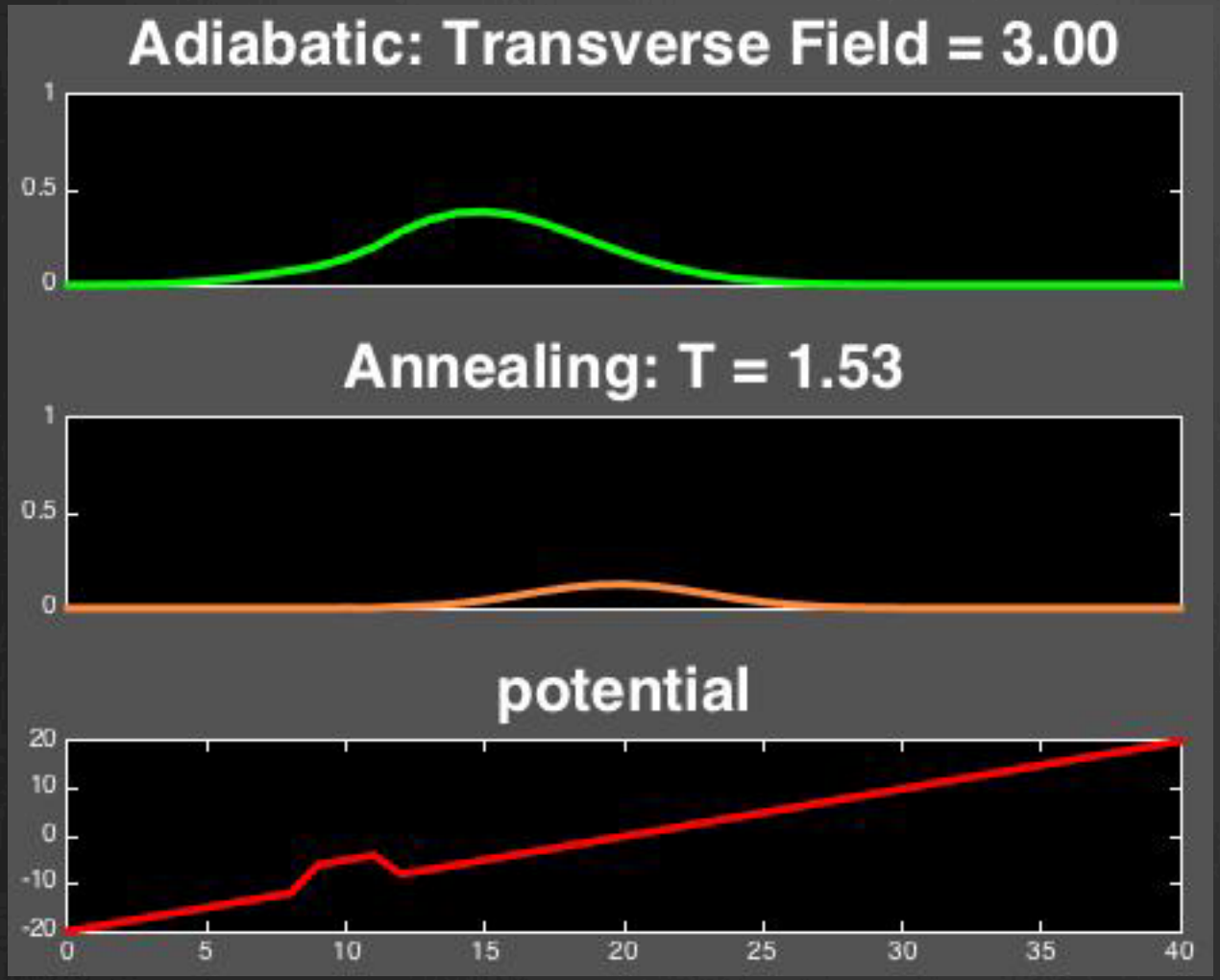


Analogue for adiabatic optimization



|z| = # 1's in string z

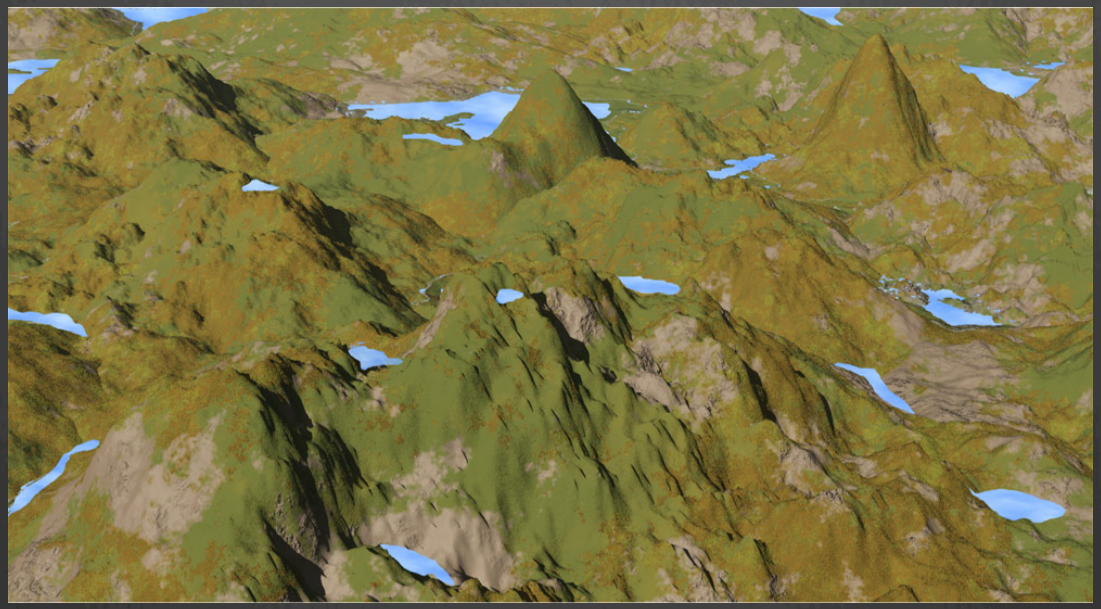
adiabatic tunneling beats simulated annealing



Adiabatic optimization
fast if barrier area $\leq n^{1/2}$

Simulated Annealing
exponential slowdown

quantum advantage in tunneling?



"...classical algorithms can only 'walk over this landscape'. Quantum computers can **tunnel** through the landscape making it faster to find the **lowest point**. The D-Wave processor considers all the possibilities simultaneously to determine the lowest energy..."

from http://www.dwavesys.com/quantum-computing

ground states of stoquastic Hamiltonians

$$H(t) = \left(1 - \frac{t}{T_{\text{final}}}\right) L + \frac{t}{T_{\text{final}}} \text{diag}(f)$$

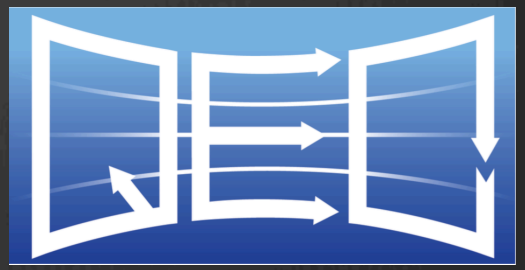
$$= \begin{pmatrix} & \leq 0 & \leq 0 & \cdots \\ \leq 0 & & \leq 0 & \\ \leq 0 & \leq 0 & & \\ \vdots & & & \ddots \end{pmatrix}$$

"sign-problem free"

"stoquastic"

⇒ ground state ψ has only nonnegative entries

⇒ non-universal form of quantum computing

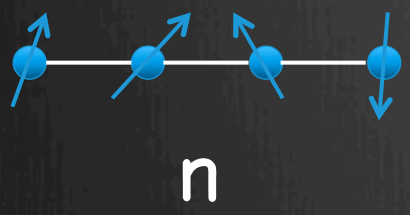


Quantum Enhanced Optimization

quantum Monte Carlo (QMC)

(actually a classical algorithm)

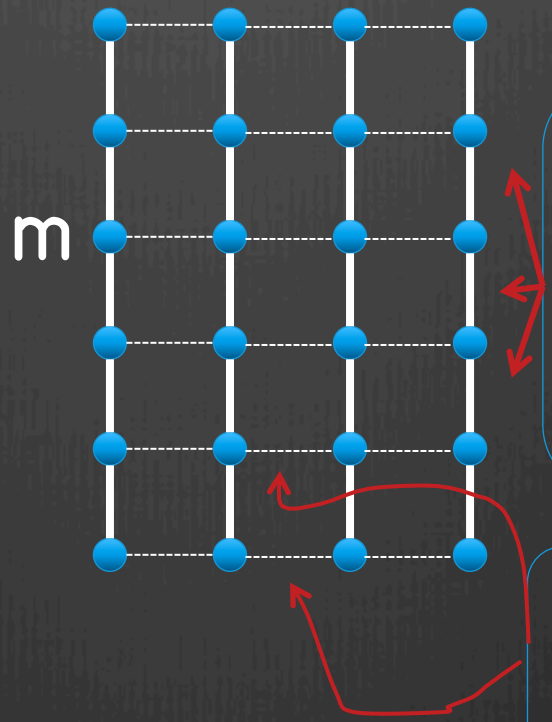
stoquastic



n

$H = f + \Gamma L$

classical



Vertical bonds:
ferromagnetic
energy≈log(m/ Γ)
i.e.disagree prob
O(Γ /m)

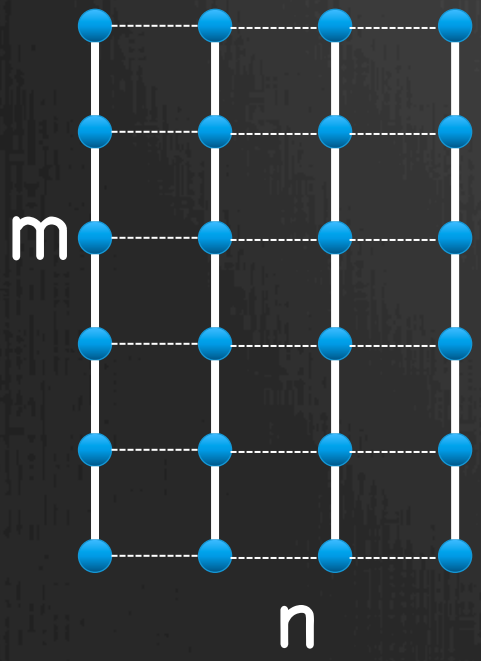
Horizontal bonds:
= f / m

QMC vs barriers

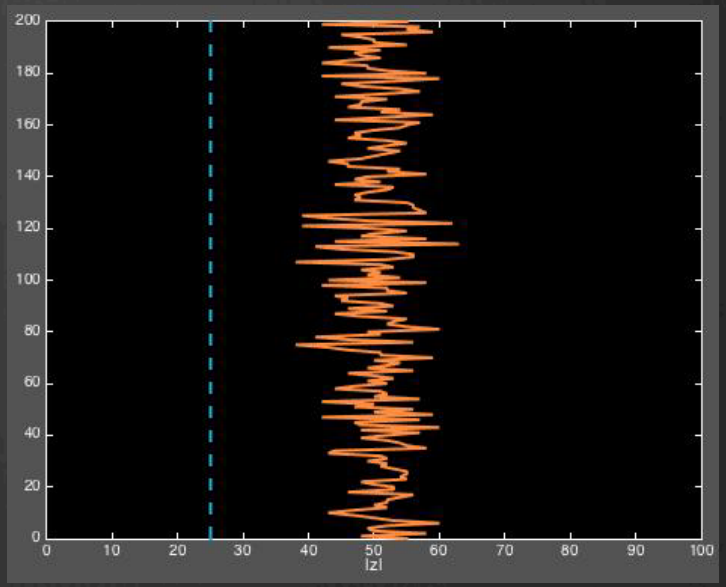
Elizabeth Crosson

Thm: For barrier sizes/shapes where the adiabatic algorithm tunnels through rapidly then quantum Monte Carlo does too.

Proof idea:



Only a few rows on the barrier at once.



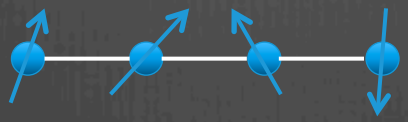
simulated annealing

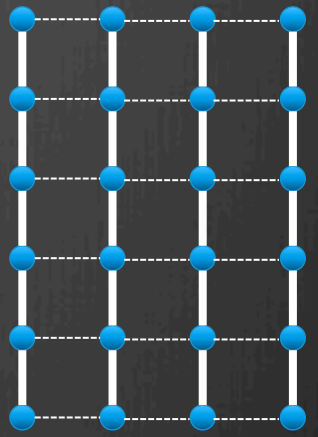
VS

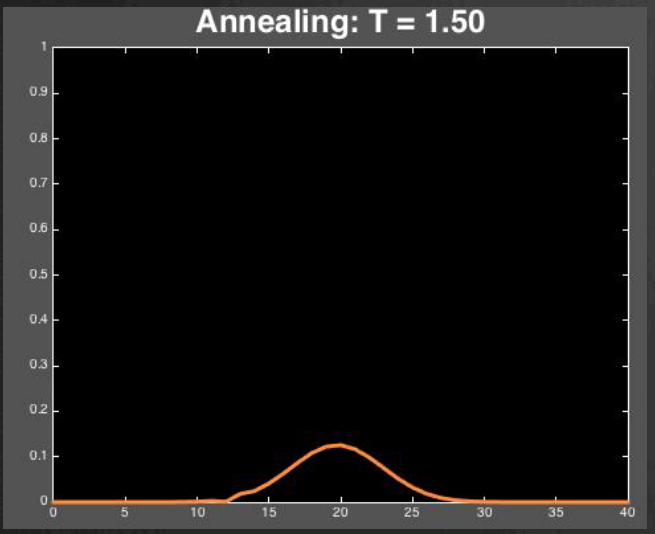
quantum adiabatic optimization

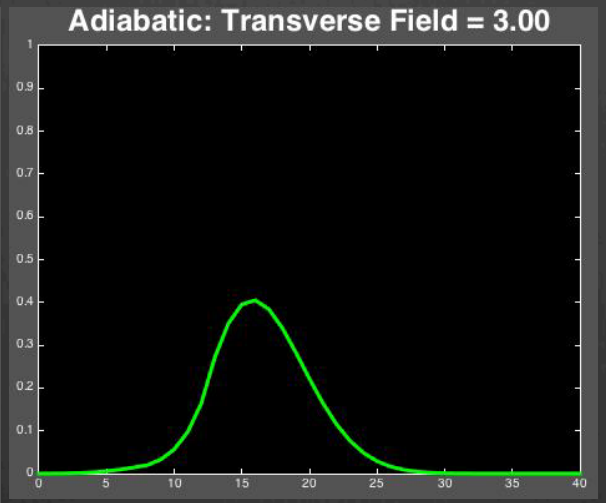
VS

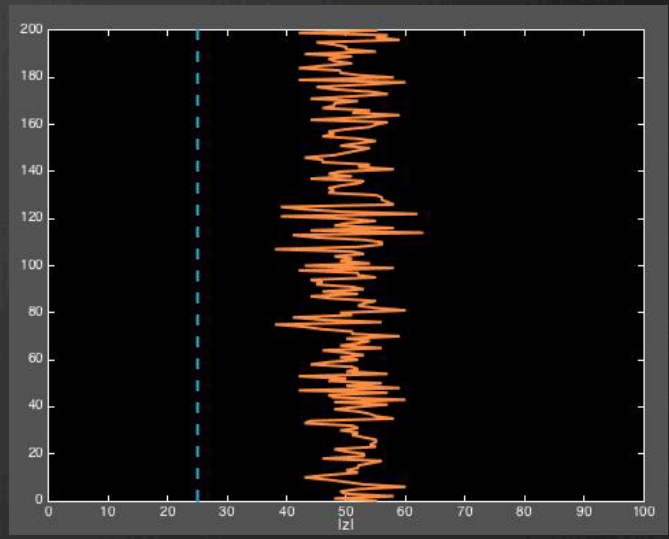
quantum Monte Carlo







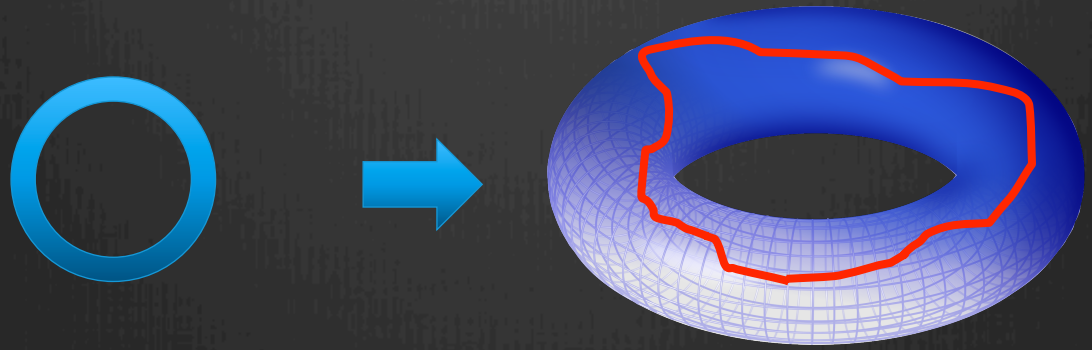




implications

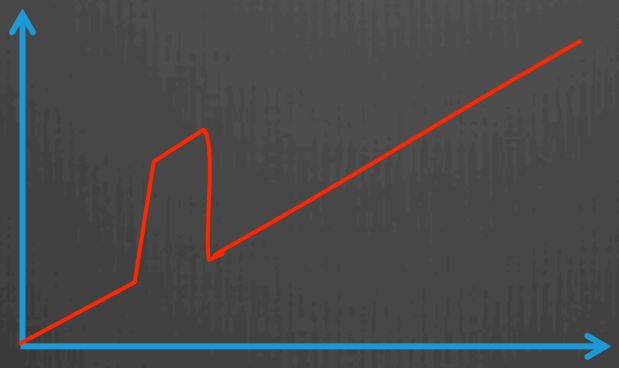
- Evidence against tunneling as a route to quantum speedup.

- Routes to quantum advantage?
  - Non-stoquastic adiabatic optimization
  - Tasks where quantum Monte Carlo fails



overview

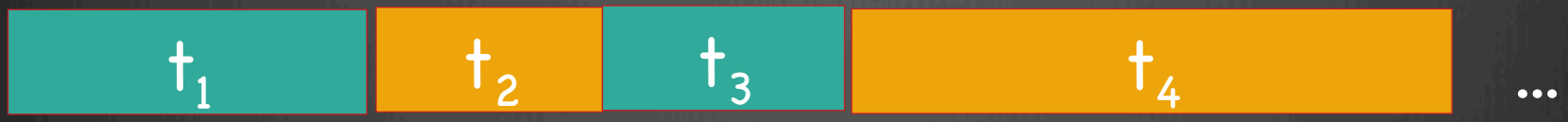
classical optimization algorithms



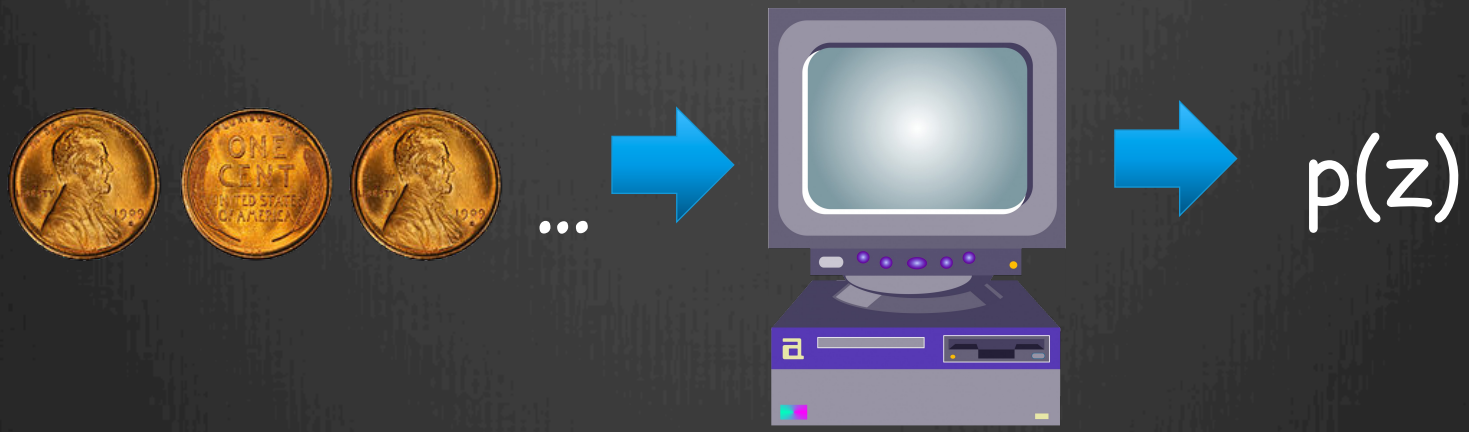
adiabatic optimization

variational algorithms and quantum supremacy

variational quantum algorithms



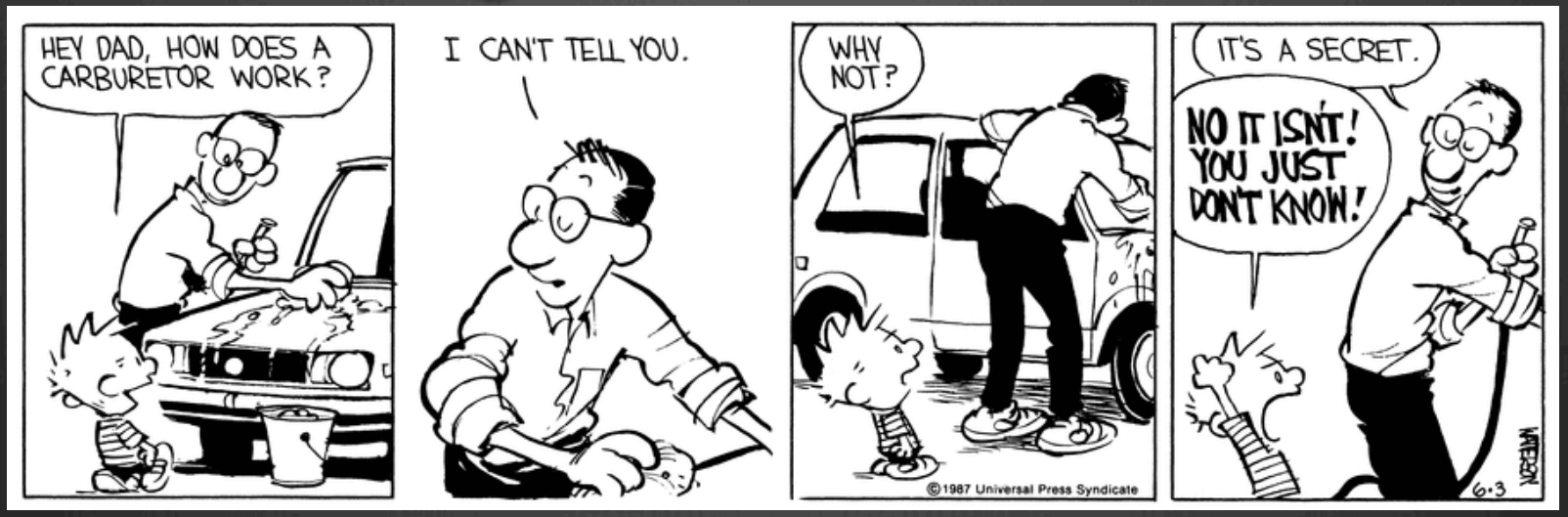
H = L diag(f) L diag(f)



Theorem: "Quantum Supremacy"

Outputs of quantum algorithm are hard to simulate classically.

things I don't know



- Why are there quantum speedups?
  - Exponentially large dimension not enough.
  - Interference needed too.
- What good is a 100-qubit quantum computer?
- Stoquastic adiabatic evolution:
  quantum supremacy or efficient classical simulation?

deleted scenes

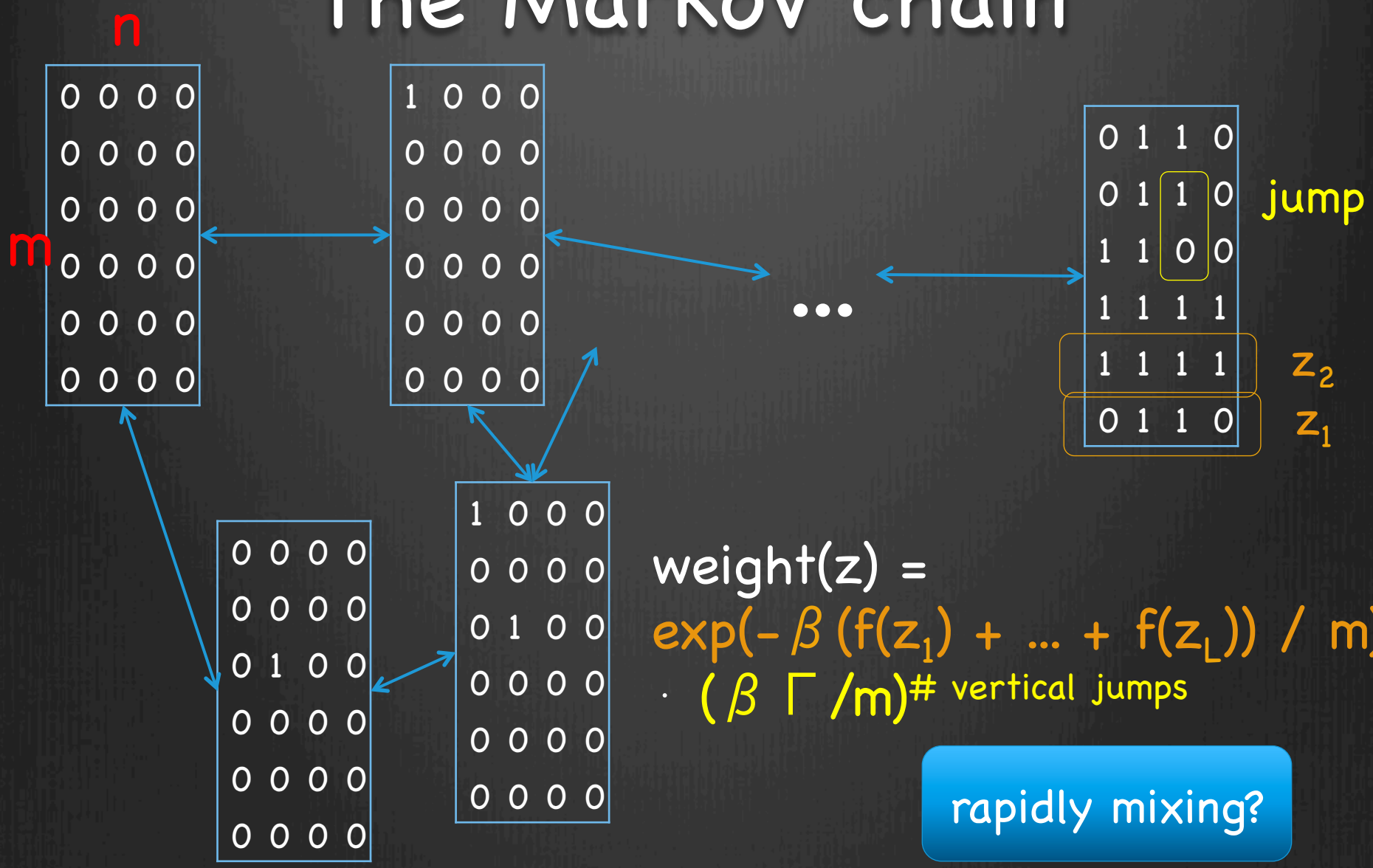
standard part of QMC

1. Use local moves (Glauber or Metropolis) to generate samples from $\pi(z_1, \ldots, z_L)$. Run-time/accuracy tradeoff unknown in general.

2. Use sampling-to-counting equivalence to estimate Z or $\langle O \rangle = \mathrm{tr}[O\, e^{-\beta H}]/Z$.

Problem reduces to bounding mixing time (equiv. gap) of a classical Markov chain.

The Markov chain



weight(z) =

$\exp(-\beta\,(f(z_1) + \ldots + f(z_L))\,/\,m)$

$\cdot\ (\beta\ \Gamma\ /m)^{\#\text{ vertical jumps}}$

rapidly mixing?

the path measure

[see also JSIBMTN 1603.01293]

random walk $z_1,\ldots,z_m$ on hypercube $\{0,1\}^n$

- conditioned to return ($z_{m+1} = z_1$)
- alternatively can use open boundary conditions.
- typically $\approx \beta \Gamma n$ total jumps

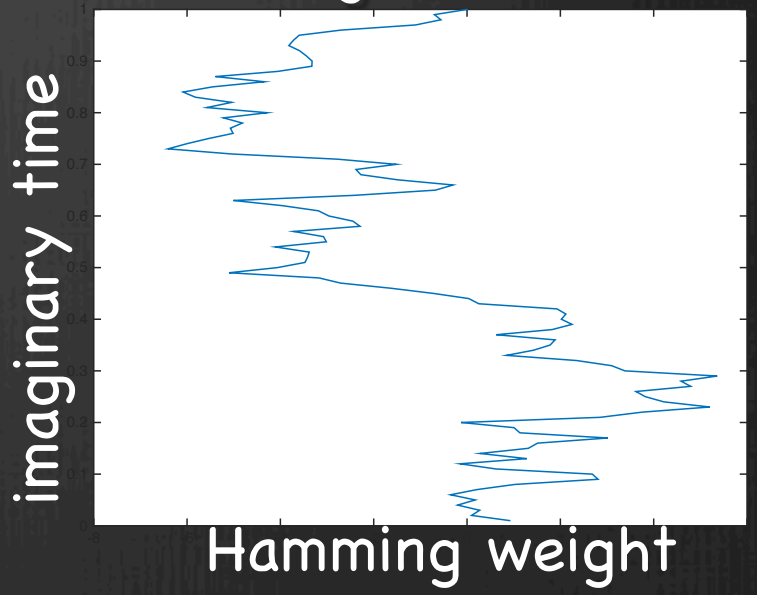
Suppose that f(z) depends only on Hamming weight |z|.

- look only at Hamming weight: $\{0,1\}^n$ -> $\{0,1,\ldots,n\}$.
- take n->∞ and $\{0,1,\ldots,n\} \rightarrow [0,1]$.
- Brownian motion, or with closed B.C., "Brownian bridge"

with local Z fields ->
Brownian motion with drift
"Ornstein-Uhlenbeck bridge"

$dx(t) = \theta(\mu - x(t))\,dt + \sigma\,dB(t)$

θ = drift, μ = mean, σ = diffusion



local times of Brownian motion

Local time: $L^x(t)$ = amount of time Brownian motion $B(t)$ spends at point x.

Lévy's theorem: $\{L^0(t): t \geq 0\}$ and $\{S(t): t \geq 0\}$ have the same distribution, where $S(t) = \sup_{0 \leq s \leq t} B(s)$.

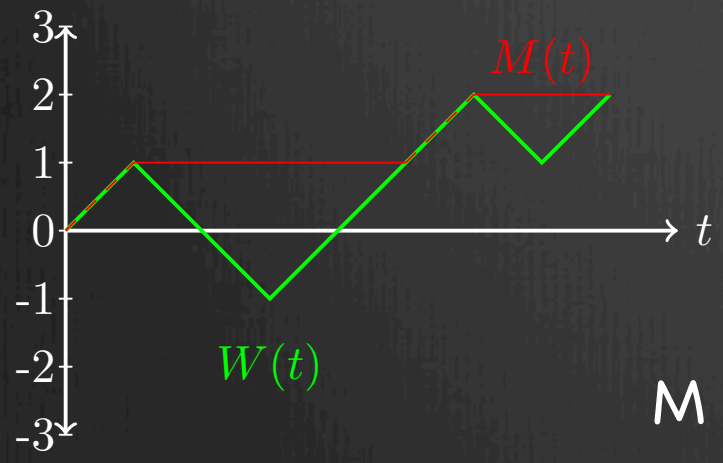
In fact, $(S-B, S) =^d (|B|, L^0)$
Additionally, $S =^d |B|$.

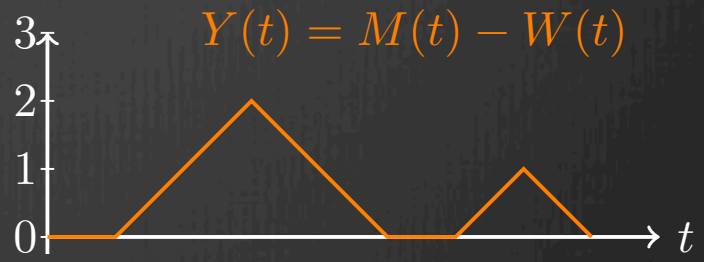
local times of Brownian motion

Local time: $L^x(t)$ = amount of time Brownian motion B([0,t]) spends at point x.

Lévy's theorem: $(S-B, S) =^d (|B|, L^0)$

Proof: consider discrete r walk: W(k) = X(1) + … + X(k) with X(t) = ±1. Let M(k) = max(W(0), …, W(k)).





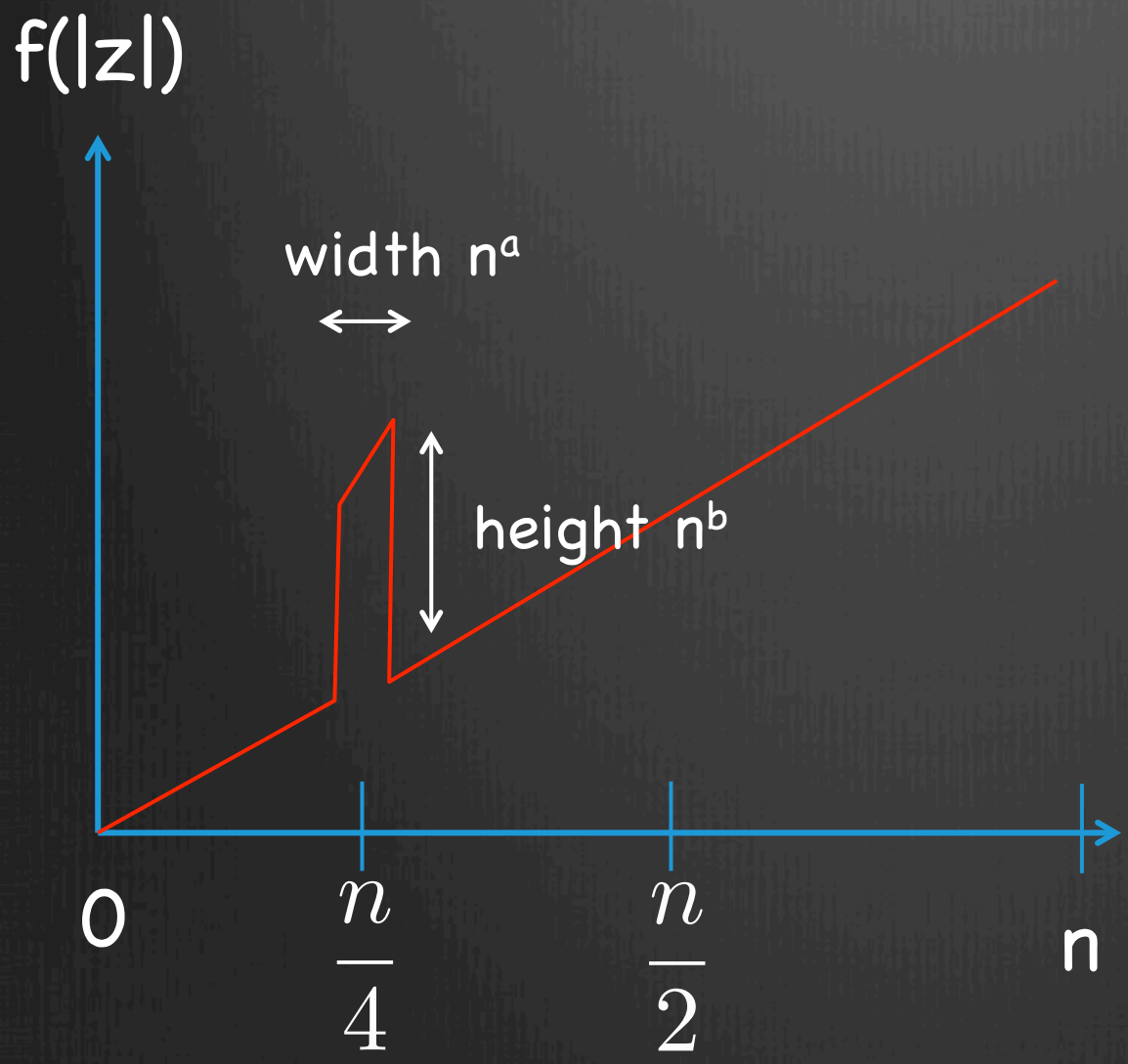
Y(t) = M(t)-W(t) $\approx^d$ |W(t)|

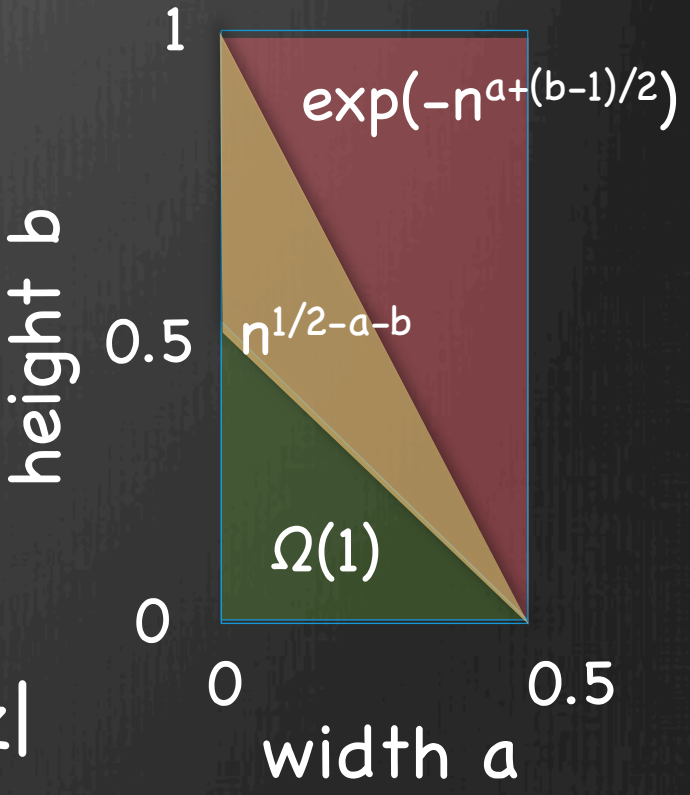
M = # of times M-W remains at 0 $\approx^d L^0$

figure adapted from *Brownian Motion* by Mörters and Peres.

Hamming weight + spike

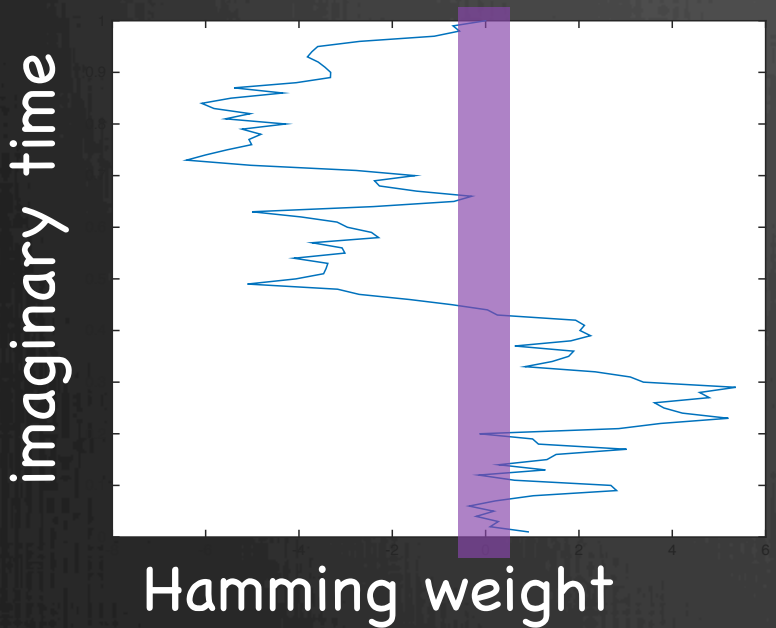
FGG '02
R '04
KC '15
BvD '16
JSIBMTN '16





QMC and tunnelling

spike
(width n^a, height n^b)



ST = normalized spike time
$\approx^d |N(0, n^{a-1/2})|$

proof using either Lévy's thm or quantum-classical correspondence.

Feynman-Kac thm:
$\Pr[\text{path} \mid \text{spike}] = \exp(-\beta\, \text{ST}\, n^b) \Pr[\text{path} \mid \text{no spike}]$

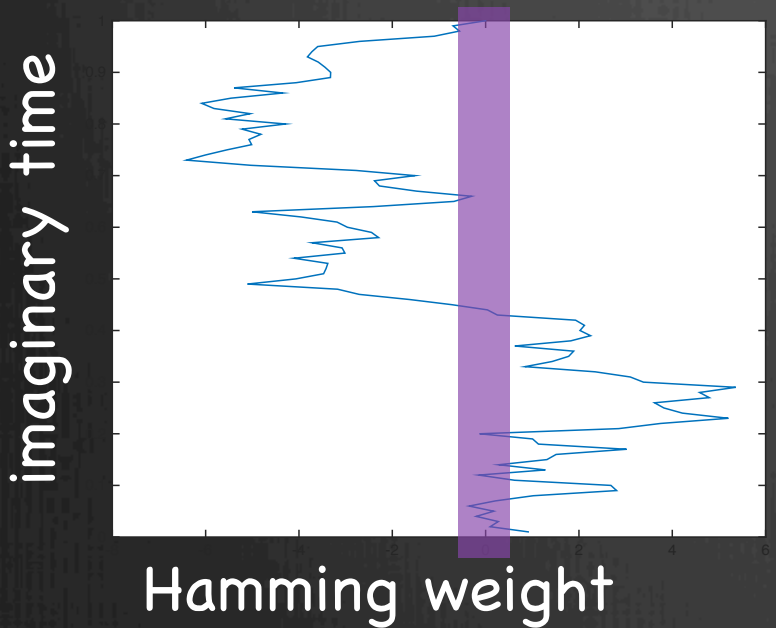
→ if $a+b<1/2$ then typical paths don't notice the spike.

instantons on the cheap

spike
(width n^a, height n^b)

$a<1/2$
$2a+b<1$

cf JSIBMTN'16



steps to traverse spike ≈ n^{2a}

min ST = n^{2a} / $\beta\,\Gamma\,n$

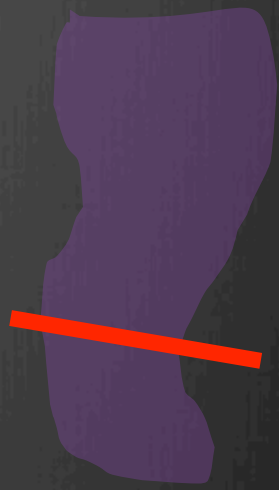
Feynman-Kac →
prob reduced by ≈ $\exp(-n^{2a+b-1})$

∴ $2a+b\leq 1$ is the theshold to cross the spike once.

canonical paths

Given Markov chain P(x,y) with stationary distribution $\pi(x)$ and $Q(x,y) = P(x,y)\,\pi(y) = Q(y,x)$.
TFAE:

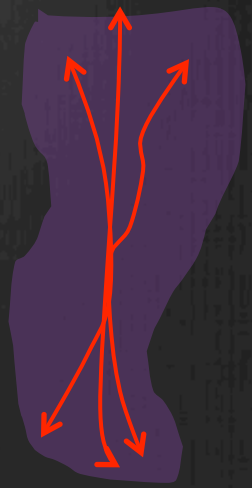
- P has a $\geq 1/\mathrm{poly}(n)$ gap between the top two eigenvalues



conductance

- The conductance Φ is $\geq 1/\mathrm{poly}(n)$.
  $\Phi = \min_S Q(S, S^c) / \pi(S)\,\pi(S^c)$

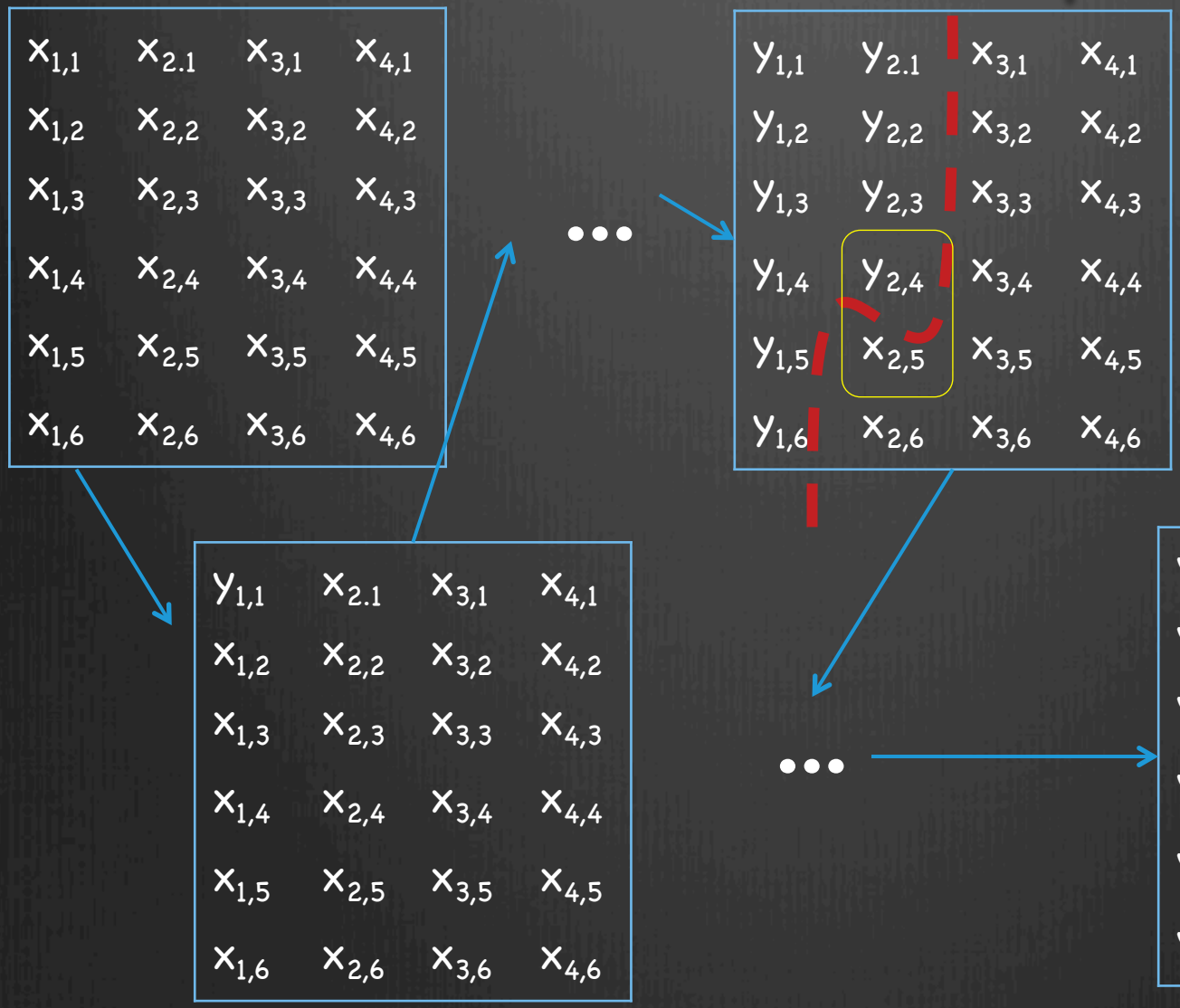
- For any x,y there exists a path γ_{xy} from x -> y routing $\pi(x)\,\pi(y)$ units of flow such that each edge e has load $\leq \mathrm{poly}(n)\,Q(e)$.
  ("canonical paths/flows")



canonical paths

Heuristics analyze some plausible cut.
Proofs analyze all cuts or construct paths.

1-d canonical path



energy penalty:
≤ 2 new jumps
≤ 1 term from H_D
(L bonds each with weight 1/L.)