Title: Exponential quantum speedup in simulating coupled classical oscillators

Speakers: Nathan Wiebe

Series: Perimeter Institute Quantum Discussions

Date: April 19, 2023 - 11:00 AM

URL: https://pirsa.org/23040079

Abstract: We present a quantum algorithm for simulating the classical dynamics of 2^n coupled oscillators (e.g., masses coupled by springs). Our approach leverages a mapping between the Schrodinger equation and Newton's equations for harmonic potentials such that the amplitudes of the evolved quantum state encode the momenta and displacements of the classical oscillators. When individual masses and spring constants can be efficiently queried, and when the initial state can be efficiently prepared, the complexity of our quantum algorithm is polynomial in n, almost linear in the evolution time, and sublinear in the sparsity. As an example application, we apply our quantum algorithm to efficiently estimate the kinetic energy of an oscillator at any time, for a specification of the problem that we prove is BQP-complete. Thus, our approach solves a potentially practical application with an exponential speedup over classical computers. Finally, we show that under similar conditions our approach can efficiently simulate more general classical harmonic systems with 2^n modes.

Zoom link: https://pitp.zoom.us/j/91882209363?pwd=UndJRVdaZW04RGtpL0M2SE52RDJwZz09

Simulating Harmonic Oscillators Using Quantum Computers

Ryan Babbush, Dominic Berry, Robin Kothari, Rolando Somma, Nathan Wiebe

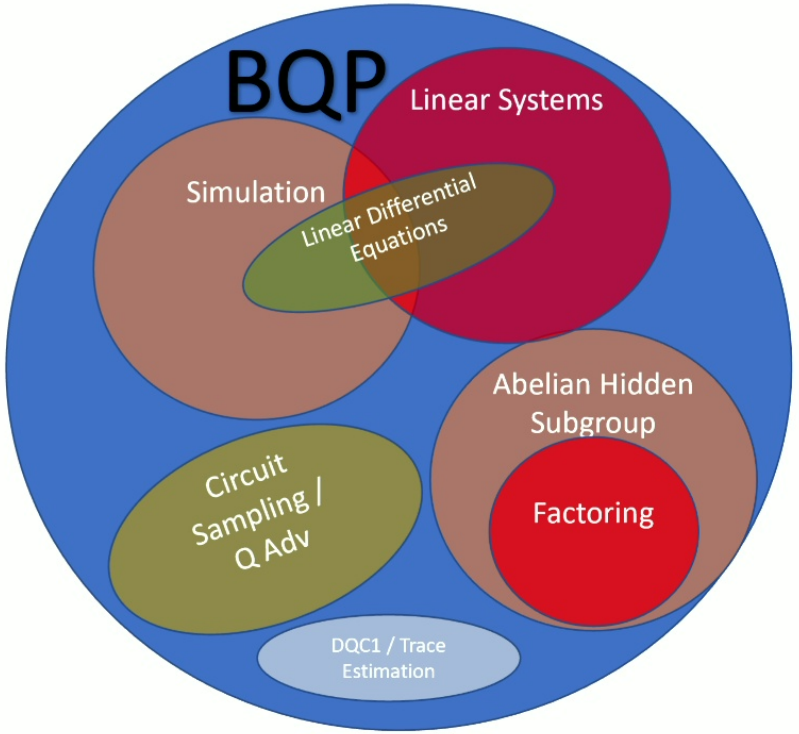
Exponential / Suspected Exponential Speedups for Quantum Computing

Examples of new classes of problems are rare.

Applications are often hard to find because of the limitations that quantum imposes.

Many examples have been found to be dequantizable: (Recommendation Systems, Nearest Centroid Classifiers, Quantum PCA, Topological Data Analysis??)

We want to understand what makes a problem hard to dequantize and in turn understand what tasks are true exponential speedups for quantum.



Finding Speedups for Differential Equations

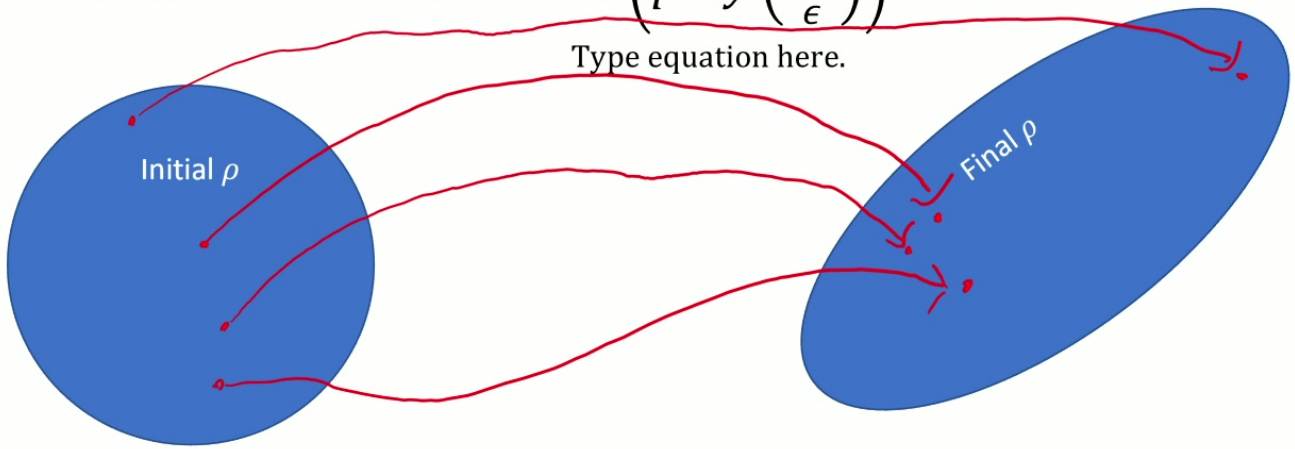
- Nearly everything can be modeled by a linear differential equation.
- Let ρ be a probability density of N particles in D dimensions.
- $\log(\dim(\rho)) \in O(ND)$, meaning distribution is exponentially large
- Dynamics of probability density is given by

$$\frac{\partial \rho}{\partial t} = -\{\rho, H\}$$

- Classical dynamics can be described by a linear differential eq.
- Awesome! Does this mean we get an exponential speedup for all classical dynamics?

No! No! No! we totally don't…

- The differential equation for a single trajectory is generically non-linear but can be solved in time $O\left(poly\left(\frac{ND}{\epsilon}\right)\right)$.



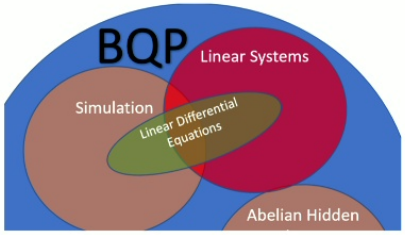
- Monte Carlo sampling can be used to get samples from distribution.

Ok Weird so are there any ODEs that give a quantum advantage

- Good News: The answer is totally yes

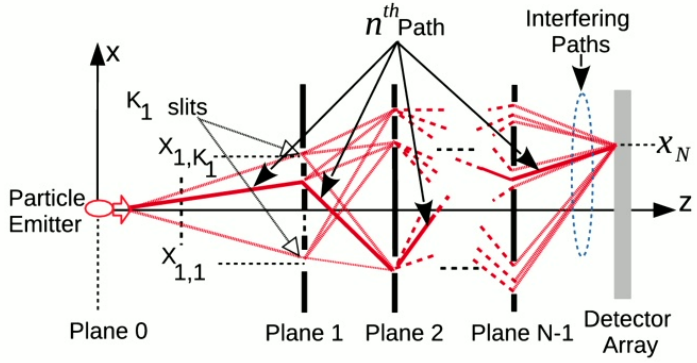
- Bad News: The most straight forward answer is the Schrodinger Equation.

$$\frac{\partial \rho}{\partial t} = -\{\rho, H\} \longrightarrow \frac{\partial \rho}{\partial t} = -i[\rho, H]$$



Why does this dequantization strategy fail for the Schrodinger equation?

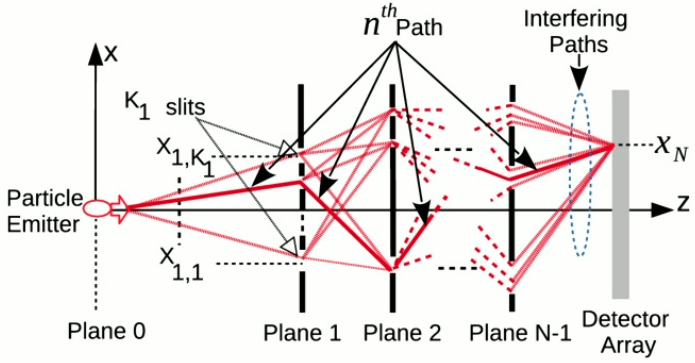
- Path Integration is the natural way to find trajectories for quantum systems.



- Each path can be computed in polynomial time.
- Phases nearly exactly cancel and variance in monte-carlo estimate is exponentially large (sign problem).

Why does this dequantization strategy fail for the Schrodinger equation?

- Path Integration is the natural way to find trajectories for quantum systems.



It's Interference!

(IE contextuality to you foundations nerds)

- Each path can be computed in polynomial time.
- Phases nearly exactly cancel and variance in monte-carlo estimate is exponentially large (sign problem).

Is interference all you need?

LOV K. GROVER AND ANIRVAN M. SENGUPTA

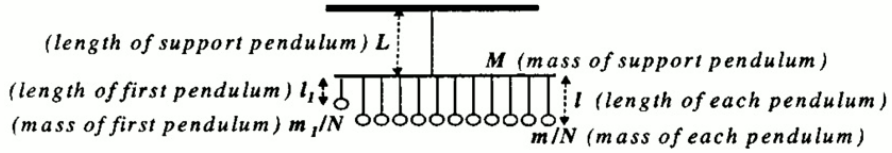
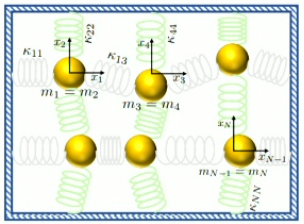


FIG. 1. *N* pendulums are suspended from a single pendulum.

- Grover showed that mechanical interference can be used to achieve a Grover speedup.
- This speedup is impractical because of engineering constraints / inability to error correct.
- Can we argue that mechanical interference is equivalent to quantum?

Intuition for Mechanical Oscillators

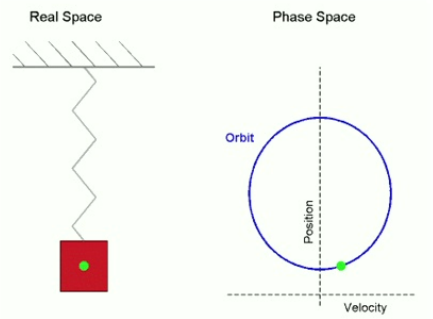


- Problem: Assume we have a system of 2^n oscillators such that

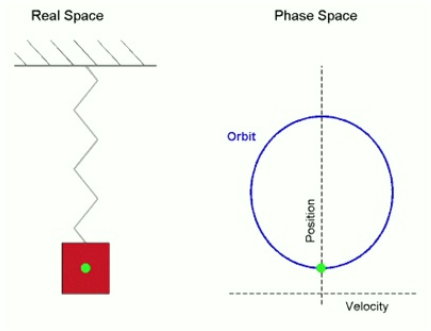
$$\mathrm{E} = \sum_i m_i \dot{x}_i{}^2/2 + \sum_{i,j} \kappa_{ij}\left(x_i - x_j\right)^2/2 + \sum_j \kappa_{jj} x_j^2/2$$

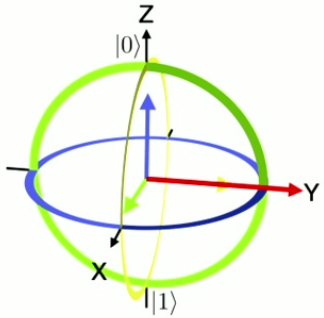
- The dynamics is given by
- $\partial_t x_i = \dot{x}_i, \quad \partial_t\, \dot{x}_i = -\sum_j \kappa_{ij}\left(x_i - x_j\right) - \kappa_{ii} x_i$

- In one dimension:



Why is this a good match with quantum: It's symplectic just like quantum





- The motion of an oscillator rotates a point in a circle.
- Quantum y-axis rotations do exactly the same thing.

How do we handle this generically?

- There are many different representations that we could use for the dynamics and they lead to different complexity.
- Unitary dynamics preserves the 2-norm of the vector.

$$\mathrm{E} = \sum_i m_i \dot{x}_i{}^2/2 + \sum_{i,j} \kappa_{ij}\left(x_i - x_j\right)^2/2 + \sum_j \kappa_{jj} x_j^2/2$$

- $|\psi\rangle = \begin{bmatrix} \sqrt{m_1/2E}\,\dot{x}_1 \\ \vdots \\ \sqrt{m_{2^n}/2E}\,\dot{x}_{2^n} \\ \sqrt{\frac{\kappa_{12}}{2E}}\,(x_1 - x_2) \\ \sqrt{\frac{\kappa_{13}}{2E}}\,(x_1 - x_3) \\ \vdots \\ \sqrt{\frac{\kappa_{2^n-1,2^n}}{2E}}\,(x_{2^n-1} - x_{2^n}) \end{bmatrix}$

This unit vector corresponds to a set of oscillators with fixed internal energy.

The time evolution of this vector under the classical equations of motions preserves the two-norm.

Unitarity

- Equation of motion for $\dot{x}$ can be written as

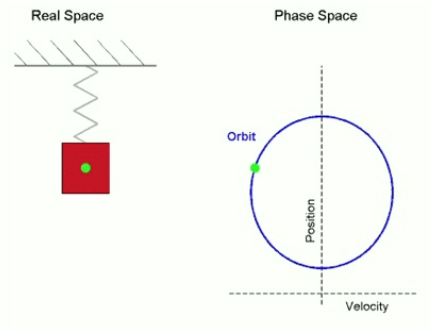
$$m_j \ddot{x}_j(t) = \sum_{k \neq j} \kappa_{jk} \big(x_k(t) - x_j(t) \big) - \kappa_{jj} x_j(t) \ .$$

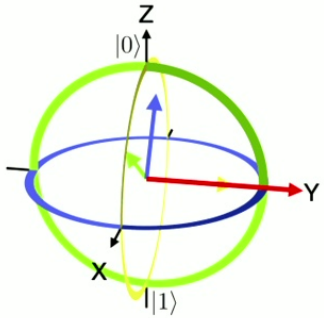
- This is equivalent to $\mathbf{M} \ddot{\vec{x}}(t) = -\mathbf{F} \vec{x}(t)$

- If we define $\vec{y} = \sqrt{M} \vec{x}$, $\mathbf{A} := \sqrt{\mathbf{M}}^{-1} \mathbf{F} \sqrt{\mathbf{M}}^{-1} \succcurlyeq 0$ then $\ddot{\vec{y}}(t) = -\mathbf{A} \vec{y}(t)$
- This implies $\ddot{\vec{y}}(t) + i\sqrt{\mathbf{A}} \dot{\vec{y}}(t) = i\sqrt{\mathbf{A}} \Big(\dot{\vec{y}}(t) + i\sqrt{\mathbf{A}} \vec{y}(t) \Big)$
- Then in turn we can see that in our encoding we have unitarity:

$$\dot{\vec{y}}(t) + i\sqrt{\mathbf{A}} \vec{y}(t) = e^{it\sqrt{\mathbf{A}}} \Big(\dot{\vec{y}}(0) + i\sqrt{\mathbf{A}} \vec{y}(0) \Big)$$

Why is this a good match with quantum: It's symplectic just like quantum





- The motion of an oscillator rotates a point in a circle.
- Quantum y-axis rotations do exactly the same thing.

Simplest Way to Simulate (OG HHL Method)

- Dynamics of Harmonic Oscillators can then be simulated by simulating $\sqrt{A}$ on an initial quantum state.
- We formally deal with this using
- Algorithm:
  - Prepare initial state $|\psi\rangle$
  - Apply phase estimation coherently on state using e^{-iAt} for $t \leq \frac{\pi|A|}{2}$.
    - Let $A|v\rangle = \lambda_v|v\rangle,\ \sum_v a_v\,|v\rangle|0\rangle \mapsto \sum_v a_v\,|v\rangle|\lambda_v\rangle$
    - Apply a square root operation reversibly: $\sum_v a_v\,|v\rangle|\lambda_v\rangle \mapsto \sum_v a_v\,|v\rangle|\lambda_v\rangle|\sqrt{\lambda_v}\rangle$.
    - Apply phase $\sum_v a_v\,|v\rangle|\lambda_v\rangle\left|\sqrt{\lambda_v}\right\rangle \mapsto \sum_v a_v\,e^{-i\sqrt{\lambda_v}t}|v\rangle|\lambda_v\rangle\left|\sqrt{\lambda_v}\right\rangle$.
    - Invert transformations.

Unitarity

- Equation of motion for $\dot{x}$ can be written as

$$m_j \ddot{x}_j(t) = \sum_{k \neq j} \kappa_{jk} \big(x_k(t) - x_j(t) \big) - \kappa_{jj} x_j(t) \ .$$

- This is equivalent to $\mathbf{M} \ddot{\vec{x}}(t) = -\mathbf{F} \vec{x}(t)$

- If we define $\vec{y} = \sqrt{M} \vec{x}$, $\mathbf{A} := \sqrt{\mathbf{M}}^{-1} \mathbf{F} \sqrt{\mathbf{M}}^{-1} \succcurlyeq 0$ then $\ddot{\vec{y}}(t) = -\mathbf{A} \vec{y}(t)$
- This implies $\ddot{\vec{y}}(t) + i\sqrt{\mathbf{A}} \dot{\vec{y}}(t) = i\sqrt{\mathbf{A}} \Big(\dot{\vec{y}}(t) + i\sqrt{\mathbf{A}} \vec{y}(t) \Big)$
- Then in turn we can see that in our encoding we have unitarity:

$$\dot{\vec{y}}(t) + i\sqrt{\mathbf{A}} \vec{y}(t) = e^{it\sqrt{\mathbf{A}}} \Big(\dot{\vec{y}}(0) + i\sqrt{\mathbf{A}} \vec{y}(0) \Big)$$

Complexity of Algorithm

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \tag{23}$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \tag{24}$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \mathrm{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.

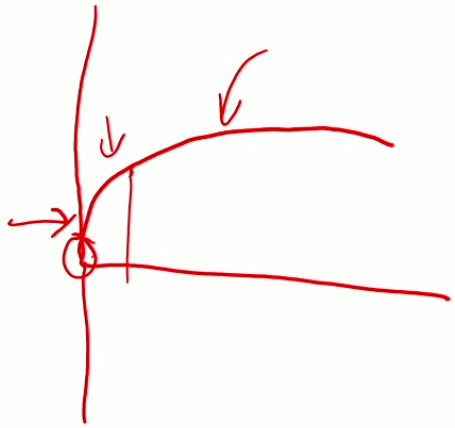
Complexity of Algorithm

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \tag{23}$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

$\frac{d}{dx}$



Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \tag{24}$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \mathrm{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.

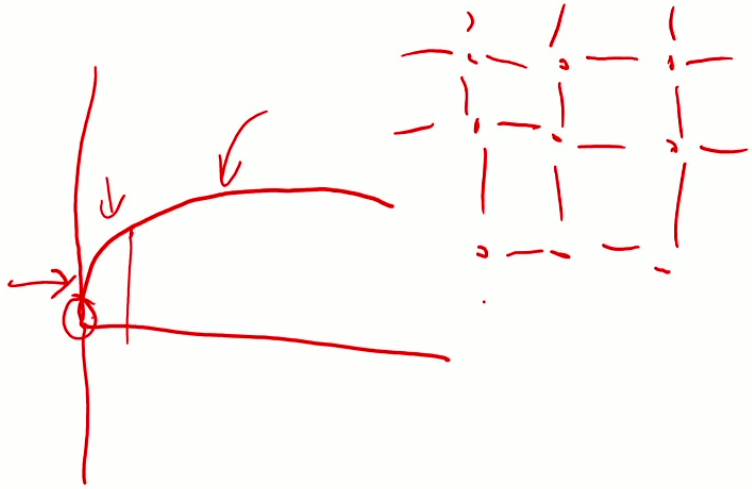
Complexity of Algorithm

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \tag{23}$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

$\frac{d}{dx}$



Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \tag{24}$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \mathrm{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.

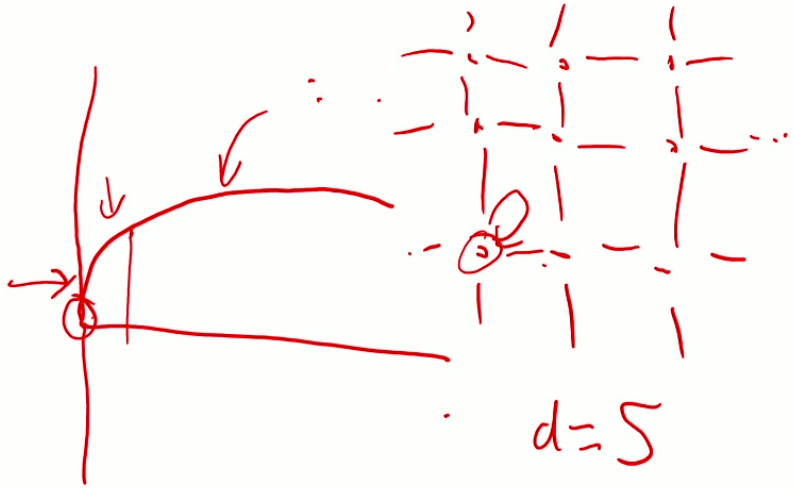
Complexity of Algorithm

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \tag{23}$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

$\frac{d}{dx}$



Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \tag{24}$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \mathrm{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.

Complexity of Algorithm

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

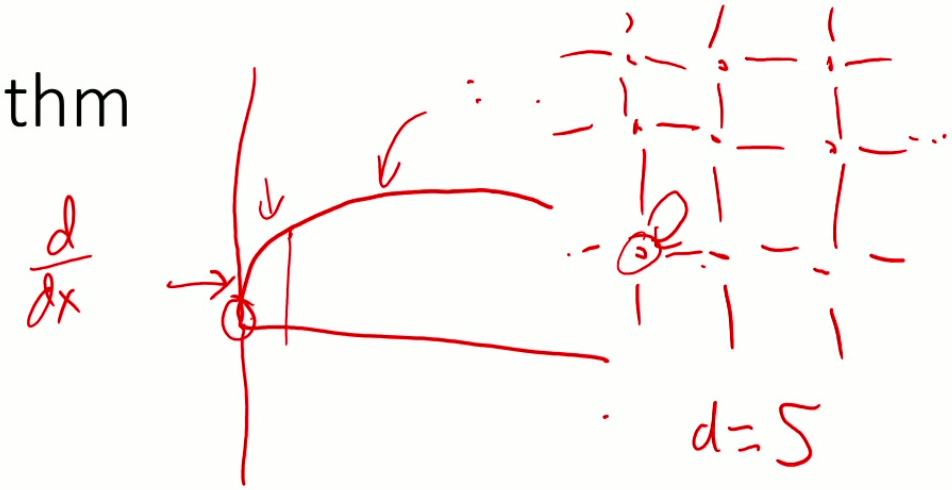
$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix}, \tag{23}$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \tag{24}$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \mathrm{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.



$$\left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \right\|_{\max} = 4$$

Complexity of Algorithm

Problem 4. *Let $\mathbf{A} \succcurlyeq 0$ be an $N \times N$ real-symmetric, PSD, d-sparse matrix. Define the normalized state*

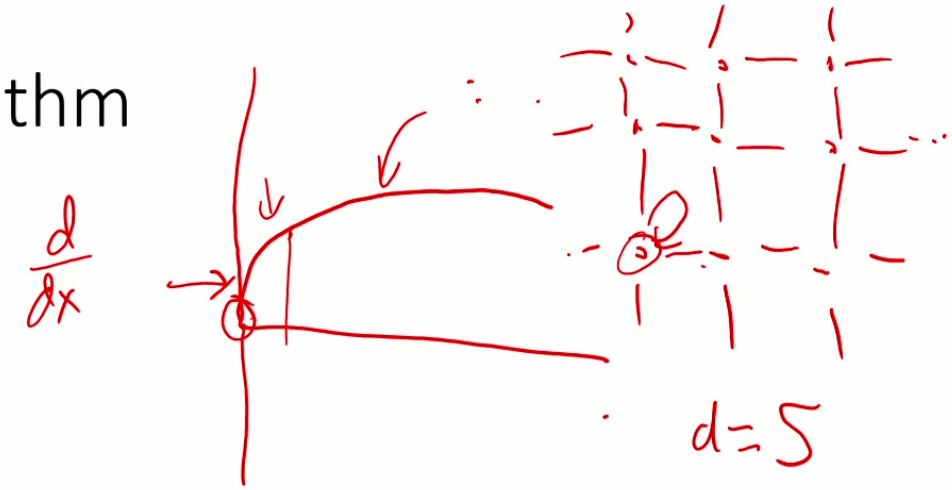
$$|\psi(t)\rangle := \frac{1}{\sqrt{2E}} \begin{pmatrix} \dot{\vec{y}}(t) \\ i\vec{\mu}(t) \end{pmatrix} , \tag{23}$$

where $E > 0$ is a constant and $\vec{\mu}(t) := \sqrt{\mathbf{A}}\vec{y}(t) \in \mathbb{C}^N$. Assume we are given oracle access to $\mathbf{A}$ and oracle access to a unitary $\mathcal{W}$ that prepares the initial state $|\psi(0)\rangle$. Given t and ϵ, the goal is to output a state that is ϵ-close to $|\psi(t)\rangle$ in Euclidean norm.

Theorem 4. *Problem 4 can be solved with a quantum algorithm that makes*

$$Q = \mathcal{O}\left(\|\mathbf{A}\|_{\max} d \log(1/\epsilon) \min\left(\frac{t\sqrt{\|\mathbf{A}^{-1}\|}}{\epsilon}, \frac{t^2}{\epsilon^2} \right) \right) \tag{24}$$

queries to the oracles, uses $\widetilde{\mathcal{O}}(Q \times \mathrm{polylog}(N/\epsilon))$ two-qubit gates, and uses $\mathcal{W}$ once.



$$\left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \right\|_{\max} = 4$$

Special Case with Positive Couplings

- A far simpler algorithm can be employed in the case where $\kappa_{ij} \geq 0$.
- The central idea behind it is to use a simplified idea for building the square root of *A*.
- Let B be a matrix such that $B^{+}B = A$. Note B does not need to be square.
- We choose $B \in C^{N^2 \times N}$ such that for $j \leq k$

$$\sqrt{\mathbf{MB}}\,|j,k\rangle = \begin{cases} \sqrt{\kappa_{jj}}\,|j\rangle\,, & \text{if } j = k \\ \sqrt{\kappa_{jk}}(|j\rangle - |k\rangle), & \text{if } j < k \end{cases}.$$

Special Case with Positive Couplings

- A far simpler algorithm can be employed in the case where $\kappa_{ij} \geq 0$.
- The central idea behind it is to use a simplified idea for building the square root of *A*.
- Let B be a matrix such that $B^{+}B = A$. Note B does not need to be square.
- We choose $B \in C^{N^2 \times N}$ such that for $j \leq k$

$$\sqrt{\mathbf{MB}}\,|j,k\rangle = \begin{cases} \sqrt{\kappa_{jj}}\,|j\rangle\,, & \text{if } j = k \\ \sqrt{\kappa_{jk}}(|j\rangle - |k\rangle), & \text{if } j < k \end{cases}.$$

Implementing Operator

- An operator of the form $\sqrt{\mathbf{MB}}\,|j,k\rangle = \begin{cases} \sqrt{\kappa_{jj}}\,|j\rangle\,, & \text{if } j = k \\ \sqrt{\kappa_{jk}}(|j\rangle - |k\rangle), & \text{if } j < k \end{cases}$. can be block encoded with error ϵ using $O\left(\log^2\left(\frac{1}{\epsilon}\right)\right)$ operations. Block encoding const: $\sqrt{2\aleph d}$
- Idea: Construct a large superposition over many ancilla states $\frac{1}{2^{r/2}}\sum_{x=1}^{2^r}|x\rangle$
  - Perform inequality tests to see if $\frac{\kappa_{\max}\bar{\kappa}_{jk}}{2^{r_\kappa}} \leq \frac{x^2}{2^{2r}}\aleph\frac{m_{\max}\bar{m}_j}{2^{r_m}}$
  - Store result in qubit, uncompute arithmetic and swap j,k if 1.
  - Apply additional Hadamard / Z gates to get mixture.
- Log scaling comes from cost of multiplication.

Implementing Operator

$x \leq \sqrt{y}$

$x^2 \leq y$

- An operator of the form $\sqrt{\mathbf{MB}}\,|j,k\rangle = \begin{cases} \sqrt{\kappa_{jj}}\,|j\rangle\,, & \text{if } j = k \\ \sqrt{\kappa_{jk}}(|j\rangle - |k\rangle), & \text{if } j < k \end{cases}$ can be block encoded with error ϵ using $O\left(\log^2\left(\frac{1}{\epsilon}\right)\right)$ operations. Block encoding const: $\sqrt{2\aleph d}$
- Idea: Construct a large superposition over many ancilla states $\frac{1}{2^{r/2}}\sum_{x=1}^{2^r}|x\rangle$
  - Perform inequality tests to see if $\frac{\kappa_{\max}\bar{\kappa}_{jk}}{2^{r_\kappa}} \leq \frac{x^2}{2^{2r}}\aleph\frac{m_{\max}\bar{m}_j}{2^{r_m}}$
  - Store result in qubit, uncompute arithmetic and swap j,k if 1.
  - Apply additional Hadamard / Z gates to get mixture.
- Log scaling comes from cost of multiplication.

Full Simulation Algorithm

- Next let $\mathbf{H} := -\begin{pmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^\dagger & \mathbf{0} \end{pmatrix}$
- We then have that $\begin{pmatrix} \dot{\vec{y}}(t) \\ i\mathbf{B}^\dagger \vec{y}(t) \end{pmatrix} = e^{-it\mathbf{H}} \begin{pmatrix} \dot{\vec{y}}(0) \\ i\mathbf{B}^\dagger \vec{y}(0) \end{pmatrix}$

- This shows that we can simulate the dynamics using Hamiltonian simulation ideas.
- Using qubitization (with QSP to remove the arccos) to simulate the dynamics through the block-encoding of B, we obtain

Theorem 1. *Problem 1 can be solved with a quantum algorithm that makes $Q = \mathcal{O}(\tau + \log(1/\epsilon))$ queries to the oracles for $\mathbf{K}$ and $\mathbf{M}$, uses*

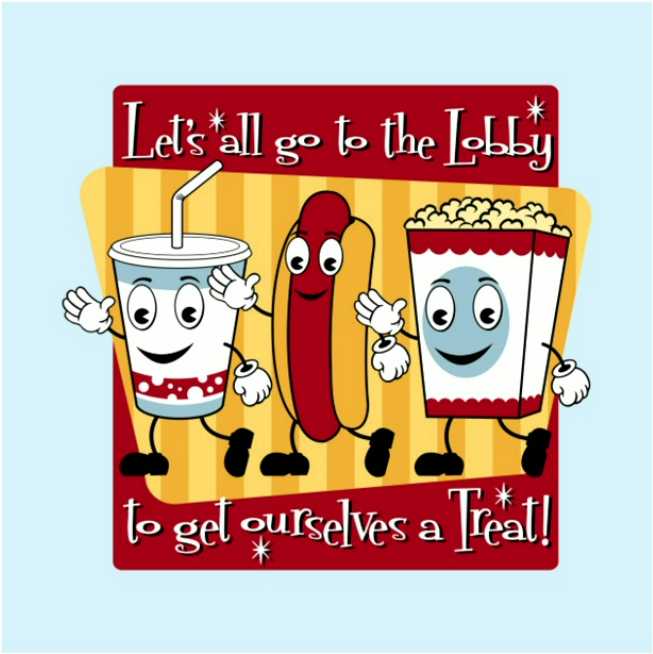
$$G = \mathcal{O}\left(Q \log^2\left(\frac{N\tau}{\epsilon}\frac{m_{\max}}{m_{\min}}\right)\right) \tag{3}$$

two-qubit gates, and uses $\mathcal{W}$ once, where $\tau := t\sqrt{\aleph d} \geq 1$, $\aleph := \kappa_{\max}/m_{\min}$, $m_{\max} \geq m_j \geq m_{\min} > 0$ and $\kappa_{\max} \geq \kappa_{jk}$ for all $j, k \in [N]$ are known quantities.

Intermission

- We showed that we can simulate exponentially many harmonic oscillators in polynomial time.

- Potential applications: Chemistry, Mechanical Eng, Aeronautics…



Intermission

- We showed that we can simulate exponentially many harmonic oscillators in polynomial time.

- Potential applications: Chemistry, Mechanical Eng, Aeronautics…

Intermission

- We showed that we can simulate exponentially many harmonic oscillators in polynomial time.

- Potential applications: Chemistry, Mechanical Eng, Aeronautics…

Does this actually give a speedup?

Claim: Simulating Exponentially Large Systems of Oscillators is BQP-Complete

- Proof: We showed the simulation is in BQP already.
  - Next step: show that it is BQP-Hard.
  - Strategy: Show that if you had a box that simulates any oscillator system, then you could simulate a quantum computer with it.

- BQP Complete Problem: Given an input state $|0\rangle$, apply a sequence of gates from a universal gate set $\{H, Toff\}^*$.
- We choose H,Toff because the gate set is real-valued and can be mapped to the oscillators easily.

Feynman-Kitaev Clock

- Idea is to take a Hamiltonian whose dynamics implements this circuit and map it to an oscillator.

Problems

1. The off-diagonal entries of the Hamiltonian would not be real negative numbers, i.e., they cannot be related to spring constants;

2. The evolution of the oscillators is induced by the $\sqrt{\mathbf{A}}$ rather than $\mathbf{A}$, so the evolution property of the Hamiltonian does not apply.

- Target operator:

$$\mathbf{A} = 4\mathbb{1}_N - \sum_{l=1}^{L} (|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes W_l \ .$$

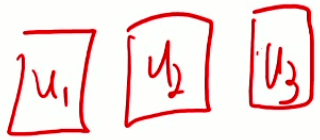
Feynman-Kitaev Clock

- Idea is to take a Hamiltonian whose dynamics implements this circuit and map it to an oscillator.

Problems

1. The off-diagonal entries of the Hamiltonian would not be real negative numbers, i.e., they cannot be related to spring constants;

2. The evolution of the oscillators is induced by the $\sqrt{\mathbf{A}}$ rather than $\mathbf{A}$, so the evolution property of the Hamiltonian does not apply.



- Target operator:

$$\mathbf{A} = 4\mathbb{1}_N - \sum_{l=1}^{L} (|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes W_l \ .$$

Dealing with negativity

$$\mathbf{A} = 4\mathbb{1}_N - \sum_{l=1}^{L} (|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes W_l \ .$$

To deal with square-roots we need to come up with a way to address the fact that that elements are negative. Negative elements prevent us from working with our faster method for square roots.

If we want gate l to be X/TOFF we choose W to be $X \otimes I, Toff \otimes I$.

If the gate you want at stage l is a Hadamard choose the gate to be $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$.

Ok so clearly something's up: no negative numbers appear at all.
Trick is that we act on the states $|0\rangle|-\rangle, |1\rangle|-\rangle$.

This means that the "Hadamard" gate above does the correct thing while only involving positive coefficients.

Thus this Hamiltonian corresponds to an oscillator system and can be used to simulate an arbitrary quantum comp.

Dealing with negativity

$$\mathbf{A} = 4\mathbb{1}_N - \sum_{l=1}^{L} (|l\rangle\langle l+1| + |l+1\rangle\langle l|) \otimes W_l \ .$$

To deal with square-roots we need to come up with a way to address the fact that that elements are negative.
Negative elements prevent us from working with our faster method for square roots.

If we want gate l to be X/TOFF we choose W to be $X \otimes I, Toff \otimes I$.

If the gate you want at stage l is a Hadamard choose the gate to be

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Ok so clearly something's up: no negative numbers appear at all.
Trick is that we act on the states $|0\rangle|-\rangle, |1\rangle|-\rangle$.

This means that the "Hadamard" gate above does the correct thing while only involving positive coefficients.

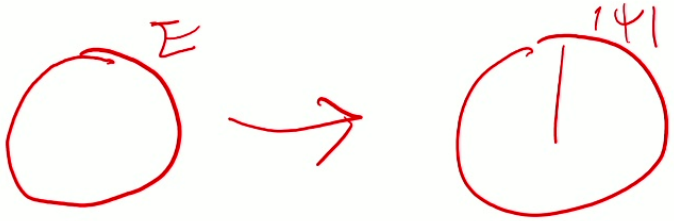
Thus this Hamiltonian corresponds to an oscillator system and can be used to simulate an arbitrary quantum comp.

Conclusions

- We provide a quantum algorithm for simulating systems of coupled classical harmonic oscillators.
- We show an exponential speedup relative to classical methods.
- If our method can be dequantized then BPP=BQP.

- Future Work:
  - Dissipative Quantum Dynamics
  - Non-Quadratic Hamiltonians
  - Do practical quantum advantages exist for our algorithm?

Conclusions



- We provide a quantum algorithm for simulating systems of coupled classical harmonic oscillators.
- We show an exponential speedup relative to classical methods.
- If our method can be dequantized then BPP=BQP.

- Future Work:
  - Dissipative Quantum Dynamics
  - Non-Quadratic Hamiltonians
  - Do practical quantum advantages exist for our algorithm?