Preparation of Thermal States of Quantum Systems by Dimension Reduction

Ersen Bilgin

Institute for Quantum Information
California Institute of Technology

Joint work with Sergio Boixo
Outline

1. Introduction
   - Motivation
   - Main Results

2. The Algorithm
   - Overview
   - How it works

3. Summary
Motivation

- Very few quantum systems have analytical solutions.
- Have to resort to numerical simulations in many cases
  - Brute force calculations take $O(e^N)$ time and memory for N-particle systems.
  - Classical algorithms to approximate solutions (DMRG, PEPS, BP, etc) only work for specific cases.
- What about quantum computers?
Very few quantum systems have analytical solutions.

Have to resort to numerical simulations in many cases:

- Brute force calculations take $O(e^N)$ time and memory for N-particle systems.
- Classical algorithms to approximate solutions (DMRG, PEPS, BP, etc) only work for specific cases.

What about quantum computers?
Motivation

- Very few quantum systems have analytical solutions.
- Have to resort to numerical simulations in many cases
  - Brute force calculations take $O(e^N)$ time and memory for $N$-particle systems.
  - Classical algorithms to approximate solutions (DMRG, PEPS, BP, etc) only work for specific cases.
- What about quantum computers?
Very few quantum systems have analytical solutions.

Have to resort to numerical simulations in many cases

- Brute force calculations take $O(e^N)$ time and memory for N-particle systems.
- Classical algorithms to approximate solutions (DMRG, PEPS, BP, etc) only work for specific cases.

What about quantum computers?
Motivation

- Very few quantum systems have analytical solutions.
- Have to resort to numerical simulations in many cases
  - Brute force calculations take $O(e^N)$ time and memory for $N$-particle systems.
  - Classical algorithms to approximate solutions (DMRG, PEPS, BP, etc) only work for specific cases.
- What about quantum computers?
Feynman suggested quantum machines to simulate quantum systems.

Quantum computers are very good at simulating unitary evolutions (Lloyd).

Initial state preparation is still a difficult problem.

Several Proposals:
- Evolving with a bath (Terhal and DiVincenzo)
- Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
- Thermalization algorithm (Poulin and Wocjan)
Feynman suggested quantum machines to simulate quantum systems.

Quantum computers are very good at simulating unitary evolutions (Lloyd).

Initial state preparation is still a difficult problem.

Several Proposals:
- Evolving with a bath (Terhal and DiVincenzo)
- Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
- Thermalization algorithm (Poulin and Wocjan)
Previous work

- Feynman suggested quantum machines to simulate quantum systems.
- Quantum computers are very good at simulating unitary evolutions (Lloyd).
- Initial state preparation is still a difficult problem.

Several Proposals:
- Evolving with a bath (Terhal and DiVincenzo)
- Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
- Thermalization algorithm (Poulin and Wocjan)
Feynman suggested quantum machines to simulate quantum systems.

Quantum computers are very good at simulating unitary evolutions (Lloyd).

Initial state preparation is still a difficult problem.

Several Proposals:
- Evolving with a bath (Terhal and DiVincenzo)
- Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
- Thermalization algorithm (Poulin and Wocjan)
Feynman suggested quantum machines to simulate quantum systems.

Quantum computers are very good at simulating unitary evolutions (Lloyd).

Initial state preparation is still a difficult problem.

Several Proposals:
- Evolving with a bath (Terhal and DiVincenzo)
- Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
- Thermalization algorithm (Poulin and Wocjan)
Previous work

- Feynman suggested quantum machines to simulate quantum systems.
- Quantum computers are very good at simulating unitary evolutions (Lloyd).
- Initial state preparation is still a difficult problem.
- Several Proposals:
  - Evolving with a bath (Terhal and DiVincenzo)
  - Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
  - Thermalization algorithm (Poulin and Wocjan)
Feynman suggested quantum machines to simulate quantum systems.

Quantum computers are very good at simulating unitary evolutions (Lloyd).

Initial state preparation is still a difficult problem.

Several Proposals:
- Evolving with a bath (Terhal and DiVincenzo)
- Quantum Metropolis Sampling (Temme et al., Yung and Aspuru-Guzik)
- Thermalization algorithm (Poulin and Wocjan)
### Main Results

<table>
<thead>
<tr>
<th></th>
<th>Poulin-Wocjan</th>
<th>Dimension Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D systems</td>
<td>$O(e^{\alpha N})$</td>
<td>$O(N^\beta | h |)$</td>
</tr>
<tr>
<td>D-dimensions</td>
<td>$O(\exp(\alpha N^D))$</td>
<td>$O(\exp(N^{D-1}))$</td>
</tr>
</tbody>
</table>

- e.g. For 5x5 lattice, $e^{25} \rightarrow e^5$ ($10^{10} \rightarrow 150$).
- We exploit the geometry of the underlying system.
Main Results

<table>
<thead>
<tr>
<th></th>
<th>Poulin-Wocjan</th>
<th>Dimension Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D systems</td>
<td>$\mathcal{O}(e^{\alpha N})$</td>
<td>$\mathcal{O}(N^\beta</td>
</tr>
<tr>
<td>D-dimensions</td>
<td>$\mathcal{O}(\exp(\alpha N^D))$</td>
<td>$\mathcal{O}(\exp(N^{D-1}))$</td>
</tr>
</tbody>
</table>

- e.g. For 5x5 lattice, $e^{25} \rightarrow e^5$ ($10^{10} \rightarrow 150$).
- We exploit the geometry of the underlying system.
### Main Results

<table>
<thead>
<tr>
<th></th>
<th>Poulin-Wocjan</th>
<th>Dimension Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1D systems</strong></td>
<td>$O(e^{\alpha N})$</td>
<td>$O(N^\beta |h|)$</td>
</tr>
<tr>
<td><strong>D-dimensions</strong></td>
<td>$O(\exp(\alpha N^D))$</td>
<td>$O(\exp(N^{D-1}))$</td>
</tr>
</tbody>
</table>

- e.g. For 5x5 lattice, $e^{25} \rightarrow e^5$ ($10^{10} \rightarrow 150$).
- We exploit the geometry of the underlying system.
Thermalization using QPE

- Given $H = \sum_a E_a |a\rangle\langle a|$, we want $\rho \propto \sum_a e^{-\beta E_a} |a\rangle\langle a|$. 

- Now, instead of $|a\rangle$, we input $I = \sum_a |a\rangle\langle a|$

  $\rightarrow \sum_a e^{-\beta E_a} |a\rangle\langle a| \otimes |E_a\rangle\langle E_a| \otimes |0\rangle\langle 0| + \ldots$

- Projecting onto $|0\rangle\langle 0|$ gives $\rho$ and succeeds with $p \sim e^{-\beta \|H\|}$. 
Thermalization using QPE

- Given $H = \sum_a E_a |a⟩⟨a|$, we want $\rho \propto \sum_a e^{-\beta E_a} |a⟩⟨a|.$

- Now, instead of $|a⟩$, we input $I = \sum_a |a⟩⟨a|$

  $$\rightarrow \sum_a e^{-\beta E_a} |a⟩⟨a| \otimes |E_a⟩⟨E_a| \otimes |0⟩⟨0| + \ldots$$

- Projecting onto $|0⟩⟨0|$ gives $\rho$ and succeeds with $p \sim e^{-\beta \|H\|}.$
Thermalization using QPE

Given $H = \sum_a E_a |a\rangle \langle a|$, we want $\rho \propto \sum_a e^{-\beta E_a} |a\rangle \langle a|$. 

Now, instead of $|a\rangle$, we input $I = \sum_a |a\rangle \langle a|$

$$\sum_a e^{-\beta E_a} |a\rangle \langle a| \otimes |E_a\rangle \langle E_a| \otimes |0\rangle \langle 0| + \ldots$$

Projecting onto $|0\rangle \langle 0|$ gives $\rho$ and succeeds with $\rho \sim e^{-\beta \|H\|}$. 
Thermalization using QPE

Given $H = \sum_a E_a |a\rangle\langle a|$, we want $\rho \propto \sum_a e^{-\beta E_a} |a\rangle\langle a|$. 

Now, instead of $|a\rangle$, we input $I = \sum_a |a\rangle\langle a| \rightarrow \sum_a e^{-\beta E_a} |a\rangle\langle a| \otimes |E_a\rangle\langle E_a| \otimes |0\rangle\langle 0| + \ldots$

Projecting onto $|0\rangle\langle 0|$ gives $\rho$ and succeeds with $p \sim e^{-\beta \|H\|}$. 
Introduction

The Algorithm

Summary

Dimension Reduction Overview

- Projecting everything in one step costs $O(e^{\beta \|H\|}) \sim O(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$. 
Dimension Reduction Overview

- Projecting everything in one step costs $O(e^{\beta \|H\|}) \sim O(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$. 
Dimension Reduction Overview

- Projecting everything in one step costs $O(e^{\beta \|H\|}) \sim O(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$. 

Dimension Reduction Overview

- Projecting everything in one step costs $O(e^{\beta \|H\|}) \sim O(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$. 

$\quad I$
Dimension Reduction Overview

- Projecting everything in one step costs $O(e^{\beta \|H\|}) \sim O(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$.

$$e^{-\beta h_{1,2}} , e^{-\beta h_{3,4}} , e^{-\beta h_{5,6}} , e^{-\beta h_{7,8}}$$

$$p \sim e^{-\beta \|h\|}$$
Dimension Reduction Overview

- Projecting everything in one step costs $\mathcal{O}(e^\beta \|H\|) \sim \mathcal{O}(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$.

\[ e^{-\beta (h_{1,2} + h_{2,3} + h_{3,4})}, \quad e^{-\beta (h_{5,6} + h_{6,7} + h_{7,8})} \]

\[ p \sim e^{-\beta \|h\|} \]
- Projecting everything in one step costs $O(e^{\beta\|H\|}) \sim O(e^N)$.
- We want to break up the projections so that only a small section needs to be restarted after a failure.
- e.g. For one-dimensional systems, $H = \sum_j h_{j,j+1}$.

$$e^{-\beta(h_{1,2}+h_{2,3}+h_{3,4}+h_{4,5}+h_{5,6}+h_{6,7}+h_{7,8})}$$

$$p \sim e^{-\beta\|h\|}, \text{ Total cost: } O(N^\beta\|h\|)$$
Perturbative Hamiltonian Update

- We need the map \( e^{-\beta H} \rightarrow e^{-\beta(H+h)} \).

- Defining \( \rho^{(\epsilon)} \propto e^{-\beta(H+\epsilon h)} \), we want the sequence:

\[
\rho^{(0)} \rightarrow \rho^{(\epsilon)} \rightarrow \rho^{(2\epsilon)} \rightarrow \ldots \rightarrow \rho^{(1)}
\]

- Each step is correct up to an error of \( O(\epsilon^2) \), resulting an overall error of \( O(\epsilon) \).
We need the map $e^{-\beta H} \rightarrow e^{-\beta (H + h)}$.

Defining $\rho^{(\epsilon)} \propto e^{-\beta (H + \epsilon h)}$, we want the sequence:

$$\rho^{(0)} \rightarrow \rho^{(\epsilon)} \rightarrow \rho^{(2\epsilon)} \rightarrow \ldots \rightarrow \rho^{(1)}$$

Each step is correct up to an error of $O(\epsilon^2)$, resulting an overall error of $O(\epsilon)$. 
We need the map $e^{-\beta H} \rightarrow e^{-\beta (H+h)}$.

Defining $\rho^{(\epsilon)} \propto e^{-\beta (H+\epsilon h)}$, we want the sequence:

$$\rho^{(0)} \rightarrow \rho^{(\epsilon)} \rightarrow \rho^{(2\epsilon)} \rightarrow \ldots \rightarrow \rho^{(1)}$$

Each step is correct up to an error of $O(\epsilon^2)$, resulting an overall error of $O(\epsilon)$. 
Perturbative Hamiltonian Update - Eigenvalues

- Update the eigenvalues using QPE:

\[
\begin{align*}
|a\rangle & \xrightarrow{\text{QPE}(H)} |a\rangle \\
|0\rangle & \quad |E_a\rangle \\
|0\rangle & \quad e^{-\beta E_a/2} |0\rangle + \ldots |1\rangle
\end{align*}
\]

\[\rho \rightarrow e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2}\]

with probability \( p \geq e^{-\epsilon \beta \|h\|} \).

- This procedure updates all the eigenvalues of \( \rho \) correctly to the leading order in \( \epsilon \).
Perturbative Hamiltonian Update - Eigenvalues

- Update the eigenvalues using QPE:

\[
\rho \rightarrow e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2}
\]

with probability \( p \geq e^{-\epsilon \beta \|h\|} \).

This procedure updates all the eigenvalues of \( \rho \) correctly to the leading order in \( \epsilon \).
Update the eigenvalues using QPE:

\[ \rho \rightarrow e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2} \]

with probability \( p \geq e^{-\epsilon \beta \| h \|} \).

This procedure updates all the eigenvalues of \( \rho \) correctly to the leading order in \( \epsilon \).
Perturbative Hamiltonian Update - Eigenvalues

- Update the eigenvalues using QPE:

\[ \rho \rightarrow e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2} \]

with probability \( p \geq e^{-\epsilon \|h\|} \).

- This procedure updates all the eigenvalues of \( \rho \) correctly to the leading order in \( \epsilon \).
Perturbative Hamiltonian Update - Eigenstates

- We dephase in the eigenbasis of the new Hamiltonian, $H + \epsilon h$.
- After the QPE circuit, we had $\rho_{\text{prob}} \propto e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2}$.
- After dephasing, we get

$$\sum_{k^\epsilon} P_{k^\epsilon} \rho_{\text{prob}} P_{k^\epsilon} = e^{-\beta(H+\epsilon h)} + O(\epsilon^2)$$
We dephase in the eigenbasis of the new Hamiltonian, $H + \epsilon h$.

After the QPE circuit, we had $\rho_{\text{prob}} \propto e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2}$.

After dephasing, we get

$$\sum_{k^\epsilon} P_{k^\epsilon} \rho_{\text{prob}} P_{k^\epsilon} = e^{-\beta(H+\epsilon h)} + O(\epsilon^2)$$
We dephase in the eigenbasis of the new Hamiltonian, $H + \epsilon h$.

After the QPE circuit, we had $\rho_{\text{prob}} \propto e^{-\epsilon \beta h/2} \rho e^{-\epsilon \beta h/2}$.

After dephasing, we get

$$\sum_{k^\epsilon} P_{k^\epsilon} \rho_{\text{prob}} P_{k^\epsilon} = e^{-\beta (H + \epsilon h)} + O(\epsilon^2)$$
Putting Everything Together

- We can now implement the map $e^{-\beta H} \rightarrow e^{-\beta(H+h)}$ using the sequence:

  $\rho^{(0)} \rightarrow \rho^{(\epsilon)} \rightarrow \rho^{(2\epsilon)} \rightarrow \ldots \rightarrow \rho^{(1)}$

- This succeeds with probability $\sim (e^{-\epsilon \beta \|h\|})^{1/\epsilon} \sim e^{-\beta \|h\|}$.
Putting Everything Together

- We can now implement the map $e^{-\beta H} \rightarrow e^{-\beta (H+h)}$ using the sequence:

  $\rho^{(0)} \rightarrow \rho^{(\epsilon)} \rightarrow \rho^{(2\epsilon)} \rightarrow \ldots \rightarrow \rho^{(1)}$

- This succeeds with probability $\sim (e^{-\epsilon\beta\|h\|})^{1/\epsilon} \sim e^{-\beta\|h\|}$. 
To thermalize a chain of $N = 2^k$ qubits, we use the recursive merging procedure from earlier:

$$
\tau(k) = \alpha 2 \tau(k - 1) + m
$$

- For an error $\bar{\epsilon}$, running time for 1D: $\tau \sim \beta N^{\beta \|h\|} / \bar{\epsilon}^2$
- For D-dimensions: $\tau \sim \beta e^{2\beta \|h\|D^{D-1}} / \bar{\epsilon}^2$
To thermalize a chain of $N = 2^k$ qubits, we use the recursive merging procedure from earlier:

$$\tau(k) = \alpha 2 \tau(k - 1) + m$$

For an error $\bar{\epsilon}$, running time for 1D: $\tau \sim \beta N^{\beta\|h\|}/\bar{\epsilon}^2$

For D-dimensions: $\tau \sim \beta e^{2\beta\|h\|D^{D-1}}/\bar{\epsilon}^2$
To thermalize a chain of $N = 2^k$ qubits, we use the recursive merging procedure from earlier:

\[ \tau(k) = \alpha 2 \tau(k - 1) + m \]

For an error $\bar{\epsilon}$, running time for 1D: $\tau \sim \beta N^\beta \|h\| / \bar{\epsilon}^2$

For D-dimensions: $\tau \sim \beta e^{2\beta \|h\| D^{D-1}} / \bar{\epsilon}^2$
To thermalize a chain of $N = 2^k$ qubits, we use the recursive merging procedure from earlier:

$$\tau(k) = \alpha 2^k \tau(k-1) + m$$

For an error $\bar{\epsilon}$, running time for 1D: $\tau \sim \beta N^\beta \| h \| / \bar{\epsilon}^2$

For D-dimensions: $\tau \sim \beta e^{2\beta \| h \| D N^{D-1}} / \bar{\epsilon}^2$
With dimension reduction, we get:

<table>
<thead>
<tr>
<th></th>
<th>Poulin-Wocjan</th>
<th>Dimension Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D systems</td>
<td>$\mathcal{O}(e^{\alpha N})$</td>
<td>$\mathcal{O}(N^{\beta | h |})$</td>
</tr>
<tr>
<td>D-dimensions</td>
<td>$\mathcal{O}(\exp(\alpha N^D))$</td>
<td>$\mathcal{O}(\exp(N^{D-1}))$</td>
</tr>
</tbody>
</table>
Summary

- With dimension reduction, we get:

<table>
<thead>
<tr>
<th></th>
<th>Poulin-Wocjan</th>
<th>Dimension Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D systems</td>
<td>$\mathcal{O}(e^{\alpha N})$</td>
<td>$\mathcal{O}(N^{\beta</td>
</tr>
<tr>
<td>D-dimensions</td>
<td>$\mathcal{O}(\exp(\alpha N^D))$</td>
<td>$\mathcal{O}(\exp(N^{D-1}))$</td>
</tr>
</tbody>
</table>

- Made possible by recursively merging smaller regions using QPE and dephasing