Quantum algorithms for Dynamics Simulation: Hamiltonian Simulation and Linear Differential Equations

Di Fang

Department of Mathematics
Duke Quantum Center
Duke University

IPAM Tutorial, 2023

Outline

1. Time-dependent Ham. Sim.
2. Quantum Linear Differential Equations Solvers

Revisit: Summary of Hamiltonian Simulation

- Hamiltonian simulation: motivation; set-up
- Expected cost: No-fast-forwarding theorem and BQP-hardness
- Trotterization
- truncated Taylor series
- QSVT

Important take-home message:

QSVT + OAA $\Rightarrow$ Optimal Hamiltonian Simulation

Time-dependent Hamiltonian Simulation

$$\left\| \mathcal{U}_{\mathsf{app}} - \mathcal{T} e^{-\mathrm{i} \int_0^t H(s)\, ds} \right\| \leq \epsilon.$$

[1] [Low-Wiebe 2018]

Time-dependent Hamiltonian Simulation

$$\left\| \mathcal{U}_{\mathsf{app}} - \mathcal{T} e^{-\mathrm{i} \int_0^t H(s)\, ds} \right\| \leq \epsilon.$$

Why time-dependent?

Besides its own applications, it can also arise from time-independent problems under the Interaction Picture [1]

[1] [Low-Wiebe 2018]

Time-dependent Hamiltonian Simulation

$$\left\| \mathcal{U}_{\text{app}} - \mathcal{T} e^{-\mathrm{i} \int_0^t H(s)\, ds} \right\| \leq \epsilon.$$

Why time-dependent?

Besides its own applications, it can also arise from time-independent problems under the Interaction Picture [1]

$H = A + B$ and A has a much larger spectral norm.

[1] [Low-Wiebe 2018]

Time-dependent Hamiltonian Simulation

$$\left\| \mathcal{U}_{\mathsf{app}} - \mathcal{T} e^{-\mathrm{i} \int_0^t H(s)\, ds} \right\| \leq \epsilon.$$

Why time-dependent?

Besides its own applications, it can also arise from time-independent problems under the Interaction Picture [1]

$H = A + B$ and A has a much larger spectral norm.

Motivation: e.g.,

$$H = -\frac{1}{2}\Delta + V(x), \quad \|\Delta_h\| \gg \|V\|$$

Time-dependent Hamiltonian Simulation

$$\left\| \mathcal{U}_{\text{app}} - \mathcal{T} e^{-\mathrm{i}\int_0^t H(s)\,ds} \right\| \le \epsilon.$$

Why time-dependent?

Besides its own applications, it can also arise from time-independent problems under the Interaction Picture [1]

$H = A + B$ and A has a much larger spectral norm.

$$\partial_t \psi + \mathrm{i} A \psi = -\mathrm{i} B \psi$$

$$\xrightarrow[\text{factor}]{\text{Integrating}} e^{\mathrm{i}At}\partial_t \psi + \mathrm{i} e^{\mathrm{i}At} A \psi = -\mathrm{i} e^{\mathrm{i}At} B \psi$$

$$\Rightarrow \partial_t \Big(\underbrace{e^{\mathrm{i}At}\psi}_{\psi_I} \Big) = -\mathrm{i} e^{\mathrm{i}At} B \psi = -\mathrm{i} \underbrace{e^{\mathrm{i}At} B e^{-\mathrm{i}At}}_{H_I(t)} \psi_I$$

1 [Low-Wiebe 2018]

Time-dependent Hamiltonian Simulation

$$\left\| \mathcal{U}_{\text{app}} - \mathcal{T} e^{-\mathrm{i}\int_0^t H(s)\,ds} \right\| \leq \epsilon.$$

Why time-dependent?

Besides its own applications, it can also arise from time-independent problems under the Interaction Picture [1]

$H = A + B$ and A has a much larger spectral norm.

$$\partial_t \psi + \mathrm{i} A \psi = -\mathrm{i} B \psi$$

$$\xrightarrow[\text{factor}]{\text{Integrating}} e^{\mathrm{i}At} \partial_t \psi + \mathrm{i} e^{\mathrm{i}At} A \psi = -\mathrm{i} e^{\mathrm{i}At} B \psi$$

$$\Rightarrow \partial_t \Big(\underbrace{e^{\mathrm{i}At} \psi}_{\psi_I} \Big) = -\mathrm{i} e^{\mathrm{i}At} B \psi = -\mathrm{i} \underbrace{e^{\mathrm{i}At} B e^{-\mathrm{i}At}}_{H_I(t)} \psi_I$$

$H_I(t) := e^{\mathrm{i}At} B e^{-\mathrm{i}At}$, $\psi_I := e^{\mathrm{i}At}\psi$ and $\mathrm{i}\partial_t \psi_I = H_I \psi_I$

[1] [Low-Wiebe 2018]

Time-dependent Hamiltonian Simulation

Set-up

$$H(\tau) \quad \tau \in [0, t]$$

Desire feature: log-dependence on $\|\partial_t H(t)\|$.

[2] continuous qDRIFT [Berry-Childs-Su-Wang-Wiebe 2020], Monte-Carlo type [Poulin-Qarry-Somma-Verstraete 2011]
[3] [Berry-Childs-Cleve-Kothari-Somma 2015], [Low-Wiebe 2018], [Kieferova-Scherer-Berry 2018], [An-Fang-Lin 2022], etc

Time-dependent Hamiltonian Simulation

Set-up

$$H(\tau) \quad \tau \in [0, t]$$

Desire feature: log-dependence on $\|\partial_t H(t)\|$.

Idea: divide the time-interval into short intervals $0 < t_1 < \cdots < t_L = t$, and apply numerical integrators on sub-intervals, followed by OAA.

[2] continuous qDRIFT [Berry-Childs-Su-Wang-Wiebe 2020], Monte-Carlo type [Poulin-Qarry-Somma-Verstraete 2011]

[3] [Berry-Childs-Cleve-Kothari-Somma 2015], [Low-Wiebe 2018], [Kieferova-Scherer-Berry 2018], [An-Fang-Lin 2022], etc

Time-dependent Hamiltonian Simulation

Set-up

$$H(\tau) \quad \tau \in [0, t]$$

Desire feature: log-dependence on $\|\partial_t H(t)\|$.

Idea: divide the time-interval into short intervals $0 < t_1 < \cdots < t_L = t$, and apply numerical integrators on sub-intervals, followed by OAA.

Methods (short-time integrators):

- Trotterization, if $H(\tau) = \sum_{k=1}^{L} H_k(\tau)$. $\times$
- Randomized methods[2], e.g., continuous qDRIFT (sample and weak convergence) ✓
- LCU + Series truncation [3]: truncated Dyson series, rescaled Dyson series, truncated Magnus series, e.g., qHOP, etc ✓

[2] continuous qDRIFT [Berry-Childs-Su-Wang-Wiebe 2020], Monte-Carlo type [Poulin-Qarry-Somma-Verstraete 2011]

[3] [Berry-Childs-Cleve-Kothari-Somma 2015], [Low-Wiebe 2018], [Kieferova-Scherer-Berry 2018], [An-Fang-Lin 2022], etc

Time-dependent Hamiltonian Simulation

- Trotterization: $H = H_1(t) + H_2(t)$

$$\mathcal{T} e^{-\mathrm{i}\int_{t_j}^{t_{j+1}} H(s)\,ds} \approx e^{-\mathrm{i}hH_2(\tau_j)} e^{-\mathrm{i}hH_1(\tau_j)},$$

where $\tau_j \in [t_j, t_{j+1}]$ are chosen according to Suzuki construction. The number of unitaries depends on $\|\partial_t H(t)\|$.

Time-dependent Hamiltonian Simulation

- Trotterization: $H = H_1(t) + H_2(t)$

$$\mathcal{T} e^{-\mathrm{i}\int_{t_j}^{t_{j+1}} H(s)\, ds} \approx e^{-\mathrm{i}hH_2(\tau_j)} e^{-\mathrm{i}hH_1(\tau_j)},$$

where $\tau_j \in [t_j, t_{j+1}]$ are chosen according to Suzuki construction. The number of unitaries depends on $\|\partial_t H(t)\|$.

High-order (p-th) generalization $\left(\sum_{j=1}^{m} \|\partial_t^p H_j\|\right)^{1/(p+1)}$ [4]

Time-dependent Hamiltonian Simulation

- Trotterization:

  High-order (p-th) generalization $\left(\sum_{j=1}^{m} \|\partial_t^p H_j\|\right)^{1/(p+1)}$ [4]

- Randomized algorithms (first-order accuracy and weak conv) e.g., continuous qDRIFT, hybridized methods, etc [5]

---

4 [Wiebe-Berry-Hoyer-Sanders 2010]

5 [Poulin-Qarry-Somma-Verstraete 2011], [Berry-Childs-Su-Wang-Wiebe 2020], [Rajput-Roggero-Wiebe 2021]

Time-dependent Hamiltonian Simulation

- Trotterization:

  High-order (p-th) generalization $\left(\sum_{j=1}^{m} \|\partial_t^p H_j\|\right)^{1/(p+1)}$ [4]

- Randomized algorithms (first-order accuracy and weak conv) e.g., continuous qDRIFT, hybridized methods, etc [5]

$$
\begin{aligned}
&\mathcal{E}(t,0)(\rho) = U(t,0)\rho U^\dagger(t,0) \\
=&\mathcal{T}_{\rightarrow} \exp\left(-i\int_0^t \mathrm{d}\tau\, H(\tau)\right)\rho \mathcal{T}_{\rightarrow} \exp\dagger\left(-i\int_0^t \mathrm{d}\tau\, H(\tau)\right) \\
\approx &\,\mathcal{U}(t,0)(\rho) = \int_0^t \mathrm{d}\tau\, p(\tau) e^{-i\frac{H(\tau)}{p(\tau)}}\rho e^{i\frac{H(\tau)}{p(\tau)}},
\end{aligned}
$$

where $p(\tau) := \frac{\|H(\tau)\|}{\int_0^t \|H(\tau)\|\,d\tau} := \frac{\|H(\tau)\|}{\|H\|_{\infty,1}}$ is a probability density function defined for $0 \le \tau \le t$.

[4] [Wiebe-Berry-Hoyer-Sanders 2010]

[5] [Poulin-Qarry-Somma-Verstraete 2011], [Berry-Childs-Su-Wang-Wiebe 2020], [Rajput-Roggero-Wiebe 2021]

Time-dependent Hamiltonian Simulation

- Series truncation (LCU) based e.g., truncated Dyson series, rescaled Dyson series, truncated Magnus series, etc. [6]

$$
\begin{aligned}
&\mathcal{T}_{\to} e^{-\mathrm{i}\int_0^t H(s)\,ds} \\
=&\sum_{n=0}^{\infty} \frac{(-i)^n}{n!} \int_0^t dt_n \int_0^t dt_{n-1} \cdots \int_0^t dt_1\, \mathcal{T} H(t_n) H(t_{n-1}) \cdots H(t_1). \\
=&I - \mathrm{i} \int_0^t dt_1 H(t_1) - \int_0^t dt_2 \int_0^{t_2} dt_1 H(t_2) H(t_1) + \cdots
\end{aligned}
$$

6 [Berry-Childs-Cleve-Kothari-Somma 2015], [Low-Wiebe 2018], [Kieferova-Scherer-Berry 2018], [An-Fang-Lin 2022], etc

Time-dependent Hamiltonian Simulation

- Series truncation (LCU) based e.g., truncated Dyson series, rescaled Dyson series, truncated Magnus series, etc. [6]

$$
\begin{aligned}
&\mathcal{T}_{\rightarrow} e^{-\mathrm{i}\int_0^t H(s)\,ds} \\
=&\sum_{n=0}^{\infty} \frac{(-i)^n}{n!} \int_0^t dt_n \int_0^t dt_{n-1} \cdots \int_0^t dt_1\, \mathcal{T} H(t_n) H(t_{n-1}) \cdots H(t_1). \\
=&I - \mathrm{i} \int_0^t dt_1 H(t_1) - \int_0^t dt_2 \int_0^{t_2} dt_1 H(t_2) H(t_1) + \cdots
\end{aligned}
$$

Why? How to derive?

$$\partial_t U(t,0) = -\mathrm{i} H(t) U(t,0), \quad U(0,0) = I$$

6 [Berry-Childs-Cleve-Kothari-Somma 2015], [Low-Wiebe 2018], [Kieferova-Scherer-Berry 2018], [An-Fang-Lin 2022], etc

Time-dependent Hamiltonian Simulation

- Series truncation (LCU) based e.g., truncated Dyson series, rescaled Dyson series, truncated Magnus series, etc. [6]

$$
\begin{aligned}
&\mathcal{T}_{\to} e^{-\mathrm{i}\int_0^t H(s)\,ds} \\
=&\sum_{n=0}^{\infty} \frac{(-i)^n}{n!} \int_0^t dt_n \int_0^t dt_{n-1} \cdots \int_0^t dt_1\, \mathcal{T} H(t_n) H(t_{n-1}) \cdots H(t_1). \\
=&I - \mathrm{i} \int_0^t dt_1 H(t_1) - \int_0^t dt_2 \int_0^{t_2} dt_1 H(t_2) H(t_1) + \cdots
\end{aligned}
$$

Why? How to derive?

$$\partial_t U(t,0) = -\mathrm{i} H(t) U(t,0), \quad U(0,0) = I$$

$$
\begin{aligned}
U(t,0) =& I - \mathrm{i} \int_0^t dt_1 H(t_1) U(t_1, 0) \\
& I - \mathrm{i} \int_0^t dt_1 H(t_1) + (-\mathrm{i})^2 \int_0^t H(t_2) dt_2 \int_0^{t_2} dt_1 H(t_1) U(t_1, 0)
\end{aligned}
$$

Time-dependent Hamiltonian Simulation

- Series truncation (LCU) based e.g., truncated Dyson series, rescaled Dyson series, truncated Magnus series, etc. [6]

$$\mathcal{T}_{\rightarrow} e^{-\mathrm{i}\int_0^t H(s)\,ds}$$
$$=\sum_{n=0}^{\infty}\frac{(-i)^n}{n!}\int_0^t dt_n \int_0^t dt_{n-1}\cdots\int_0^t dt_1\,\mathcal{T} H(t_n)H(t_{n-1})\cdots H(t_1).$$
$$=I-\mathrm{i}\int_0^t dt_1 H(t_1)-\int_0^t dt_2\int_0^{t_2} dt_1 H(t_2)H(t_1)+\cdots$$

Why? How to derive?

$$\partial_t U(t,0)=-\mathrm{i}H(t)U(t,0),\quad U(0,0)=I$$

$$U(t,0)=I-\mathrm{i}\int_0^t dt_1 H(t_1)U(t_1,0)$$

$$I-\mathrm{i}\int_0^t dt_1 H(t_1)+(-\mathrm{i})^2\int_0^t H(t_2)dt_2\int_0^{t_2} dt_1 H(t_1)U(t_1,0)$$

6 [Berry-Childs-Cleve-Kothari-Somma 2015], [Low-Wiebe 2018], [Kieferova-Scherer-Berry 2018], [An-Fang-Lin 2022], etc

Part 2: General Linear Differential Equations (non-unitary dynamics)

Outline of Quantum Linear Differential Equation Solvers

- Definition of the task
- Challenges
- Ways to address
  - QSLA + Padding
  - Time-marching
  - LCHS

Linear differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = A(t) |\psi(t)\rangle , \quad |\psi(0)\rangle = |\psi_0\rangle ,$$

Task: To prepare a quantum state that is proportional to the final solution $|\psi(T)\rangle$ with certain precision ϵ.

To prepare a quantum state $|\widetilde{\psi}(T)\rangle$ that satisfies

$$\left\| \frac{|\psi(T)\rangle}{\| |\psi(T)\rangle \|} - \frac{|\widetilde{\psi}(T)\rangle}{\| |\widetilde{\psi}(T)\rangle \|} \right\| = \mathcal{O}(\epsilon).$$

Note that

$$|\psi(T)\rangle = \mathcal{T} e^{\int_0^T A(s)\mathrm{d}s} |\psi_0\rangle ,$$

and hence it is reasonable to construct $\mathcal{T} e^{\int_0^T A(s)\mathrm{d}s}$ and then apply it to the quantum state.

Challenge

Question: How to solve linear ODEs classically?

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$).

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$,

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.
  Block-encoding is needed!

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}} \psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.

  Block-encoding is needed!

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix}$$

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.

  Block-encoding is needed!

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

  The success probability vanishes exponentially in the number of time steps L!

Question: Why not use Amplitude Amplification at each time step?

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.

  Block-encoding is needed!

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

  The success probability vanishes exponentially in the number of time steps L!

Question: Why not use Amplitude Amplification at each time step?
Needs access the initial state $\mathcal{O}(\frac{1}{\sqrt{p}})$ times per step $\Rightarrow \mathcal{O}((\frac{1}{\sqrt{p}})^L)$

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.

  Block-encoding is needed!

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

  The success probability vanishes exponentially in the number of time steps L!

Question: Why not use Amplitude Amplification at each time step?

Needs access the initial state $\mathcal{O}(\frac{1}{\sqrt{p}})$ times per step $\Rightarrow \mathcal{O}((\frac{1}{\sqrt{p}})^L)$

Question: Why it works for Hamiltonian Simulation?

Challenge

Question: How to solve linear ODEs classically?

- Step 1: divide the interval $[0, t]$ into small pieces $0 < t_1 < \cdots < t_L$ with $t_k = kt/L$ and step size $\Delta t = t/L$.
- Step 2: For each time step $[t_k, t_{k+1}]$, apply some numerical integrators (denote as $\psi^{k+1} = V_{\text{num}}\psi^k$). e.g, $\psi^{k+1} = \psi^k + \Delta t A(t_k)\psi^k = (I + \Delta t A(t_k))\psi^k$, $\psi^{k+1} = e^{A(t_k)\Delta t}\psi^k$.

  Block-encoding is needed!

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

  The success probability vanishes exponentially in the number of time steps L!

Question: Why not use Amplitude Amplification at each time step?

Needs access the initial state $\mathcal{O}(\frac{1}{\sqrt{p}})$ times per step $\Rightarrow \mathcal{O}((\frac{1}{\sqrt{p}})^L)$

Question: Why it works for Hamiltonian Simulation?

For general A, V_{num} are NOT close to unitaries! Cannot use OAA!

Challenge

Major Challenge:

Short time $\Rightarrow$ Long time?

Sidenote for simple cases

Consider the simple time-independent case $A(t) \equiv A$. We seek to implement e^{tA}.

- When A is anti-Hermitian, the task becomes Hamiltonian simulation and QSVT gives the best asymptotic scaling.

[7] [Tong-An-Wiebe-Lin 2021], [Takahira-Ohashi-Sogabe-Usuda 2021], [Fang-Lin-Tong 2023]

Sidenote for simple cases

Consider the simple time-independent case $A(t) \equiv A$. We seek to implement e^{tA}.

- When A is anti-Hermitian, the task becomes Hamiltonian simulation and QSVT gives the best asymptotic scaling.
- For general A, a direct application of QSVT should just work?

[7] [Tong-An-Wiebe-Lin 2021], [Takahira-Ohashi-Sogabe-Usuda 2021], [Fang-Lin-Tong 2023]

Sidenote for simple cases

Consider the simple time-independent case $A(t) \equiv A$. We seek to implement e^{tA}.

- When A is anti-Hermitian, the task becomes Hamiltonian simulation and QSVT gives the best asymptotic scaling.
- For general A, a direct application of QSVT should just work?

  Issue: matrix exponential is defined via eigenvalue decomposition that does not agree with singular value decomposition unless the matrix is normal.
  Remedy: **Contour integral formulation** [7].

$$e^A = \frac{1}{2\pi i}\oint_\Gamma e^z (z-A)^{-1} \mathrm{d}z \approx \frac{1}{K}\sum_{k=0}^{K-1} e^{z_k} z_k (z_k - A)^{-1}.$$

[7] [Tong-An-Wiebe-Lin 2021], [Takahira-Ohashi-Sogabe-Usuda 2021], [Fang-Lin-Tong 2023]

Challenge

Major Challenge:

Short time $\Rightarrow$ Long time?

Ideas:

- Postpone A.A. as much as possible
- Device new ways to boost success probability at each time step

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [8]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

[8] [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [8]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ -(I+\Delta t A_1) & I & 0 & \cdots & 0 & 0 \\ 0 & -(I+\Delta t A_2) & I & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & -(I+\Delta t A_{L-1}) & I \end{pmatrix} \begin{pmatrix} \psi^1 \\ \psi^2 \\ \psi^3 \\ \vdots \\ \psi^L \end{pmatrix} = \begin{pmatrix} (I \end{pmatrix}$$

[8] [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [8]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ -(I+\Delta t A_1) & I & 0 & \cdots & 0 & 0 \\ 0 & -(I+\Delta t A_2) & I & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & -(I+\Delta t A_{L-1}) & I \end{pmatrix} \begin{pmatrix} \psi^1 \\ \psi^2 \\ \psi^3 \\ \vdots \\ \psi^L \end{pmatrix} = \begin{pmatrix} (I \end{pmatrix}$$

Cost of applying QLSA:

$$\mathcal{O}(s\kappa \, \mathrm{polylog}(Ns\kappa/\epsilon))$$

queries to oracles for $\mathcal{L}$ and $\mathbf{b}$

8 [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [8]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ -(I+\Delta t A_1) & I & 0 & \cdots & 0 & 0 \\ 0 & -(I+\Delta t A_2) & I & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & -(I+\Delta t A_{L-1}) & I \end{pmatrix} \begin{pmatrix} \psi^1 \\ \psi^2 \\ \psi^3 \\ \vdots \\ \psi^L \end{pmatrix} = \begin{pmatrix} (I \end{pmatrix}$$

Cost of applying QLSA:

$$\mathcal{O}(s\kappa \, \mathrm{polylog}(Ns\kappa/\epsilon))$$

queries to oracles for $\mathcal{L}$ and $\mathbf{b}$ (which in turn gives query complexity to oracles for A and for preparing ψ^0).
$\Rightarrow$ estimate the condition number of $\mathcal{L} \sim 1/\Delta t^\alpha$ for $\alpha > 0$.

[8] [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [9]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ -(I+\Delta t A_1) & I & 0 & \cdots & 0 & 0 \\ 0 & -(I+\Delta t A_2) & I & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & -(I+\Delta t A_{L-1}) & I \end{pmatrix} \begin{pmatrix} \psi^1 \\ \psi^2 \\ \psi^3 \\ \vdots \\ \psi^L \end{pmatrix} = \begin{pmatrix} (I \\ \\ \\ \\ \end{pmatrix}$$

So far we only get the history state $\mathbf{x} \Rightarrow$ still needs to extract information of ψ^L out of $\mathbf{x}$.

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [9]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ -(I+\Delta t A_1) & I & 0 & \cdots & 0 & 0 \\ 0 & -(I+\Delta t A_2) & I & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & -(I+\Delta t A_{L-1}) & I \end{pmatrix} \begin{pmatrix} \psi^1 \\ \psi^2 \\ \psi^3 \\ \vdots \\ \psi^L \end{pmatrix} = \begin{pmatrix} (I \\ \\ \\ \\ \end{pmatrix}$$

So far we only get the history state $\mathbf{x} \Rightarrow$ still needs to extract information of ψ^L out of $\mathbf{x}$.

Success probability issue: padding + A.A.

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [9]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T(\psi^2)^T \cdots (\psi^L)^T\right)^T$$

Success probability issue: padding + A.A.

Generic Result: Under certain assumptions (sparse smooth/analytic dissipative $A(t) = V(t)\Lambda(t)V(t)^{-1}$ is diagonalizable for all time and the condition number of $V(t)$ has a uniform(-in-t) upper bound; all derivatives of the solution have a uniform upper bound),

$$\mathcal{O}(qT\,\mathrm{polylog}(T/\epsilon)) \quad (q := \|\psi(0)\|/\|\psi(t)\|)$$

queries to both Oracle for A and Oracle preparing the initial state.

[9] [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

[10] [An-Liu-Wang-Zhao 2022]

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [9]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

Success probability issue: padding + A.A.

Generic Result: Under certain assumptions (sparse smooth/analytic dissipative $A(t) = V(t)\Lambda(t)V(t)^{-1}$ is diagonalizable for all time and the condition number of $V(t)$ has a uniform(-in-t) upper bound; all derivatives of the solution have a uniform upper bound),

$$\mathcal{O}(qT\,\mathrm{polylog}(T/\epsilon)) \quad (q := \|\psi(0)\|/\|\psi(t)\|)$$

queries to both Oracle for A and Oracle preparing the initial state.

Lower bound: linear in q is needed, but the optimal queries to the oracle preparing the initial state is $\mathcal{O}(q)$. [10]

[9] [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

[10] [An-Liu-Wang-Zhao 2022]

Short time to Long time: Approach 1

Approach 1: Apply QLSA + padding to history state [9]

$$\mathcal{L}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} = \left((\psi^1)^T (\psi^2)^T \cdots (\psi^L)^T\right)^T$$

Success probability issue: padding + A.A.

Generic Result: Under certain assumptions (sparse smooth/analytic dissipative $A(t) = V(t)\Lambda(t)V(t)^{-1}$ is diagonalizable for all time and the condition number of $V(t)$ has a uniform(-in-t) upper bound; all derivatives of the solution have a uniform upper bound),

$$\mathcal{O}(qT \operatorname{polylog}(T/\epsilon)) \quad (q := \|\psi(0)\|/\|\psi(t)\|)$$

queries to both Oracle for A and Oracle preparing the initial state.

Lower bound: linear in q is needed, but the optimal queries to the oracle preparing the initial state is $\mathcal{O}(q)$. [10]

Natural Question: Can we achieve optimal state preparation cost?

9 [Berry 2014], [Berry-Childs-Ostrander-Wang 2017], [Childs-Liu 2019], [Krovi 2023], etc

10 [An-Liu-Wang-Zhao 2022]

Short time to Long time: Approach 2

Approach 2: Remedy time-marching strategy by Uniform Singular Value Amplification (USVA) [11]

Motivation:

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

Success probability: $\Omega\left(\frac{\|\psi(t)\|^2}{\|\psi(0)\|^2(1+\Delta t\|A\|)^{2L}}\right)$

11 [Fang-Lin-Tong 2023]

Short time to Long time: Approach 2

Approach 2: Remedy time-marching strategy by Uniform Singular Value Amplification (USVA) [11]

Motivation:

$$\begin{pmatrix} \frac{I+\Delta tA}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta tA)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

Success probability: $\Omega\left(\frac{\|\psi(t)\|^2}{\|\psi(0)\|^2(1+\Delta t\|A\|)^{2L}}\right)$

$$\begin{pmatrix} \frac{I+\Delta tA}{\|I+\Delta tA\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta tA)^L}{\|I+\Delta tA\|^L} & * \\ * & * \end{pmatrix}$$

[11] [Fang-Lin-Tong 2023]

Short time to Long time: Approach 2

Approach 2: Remedy time-marching strategy by Uniform Singular Value Amplification (USVA) [11]

Motivation:

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

Success probability: $\Omega\left(\frac{\|\psi(t)\|^2}{\|\psi(0)\|^2(1+\Delta t\|A\|)^{2L}}\right)$

$$\begin{pmatrix} \frac{I+\Delta t A}{\|I+\Delta t A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{\|I+\Delta t A\|^L} & * \\ * & * \end{pmatrix}$$

Observation: $\|I + \Delta t A\|^L \approx \left\|e^{\Delta t A}\right\|^L \sim \left\|e^{tA}\right\|$. For dissipative system and unitary dynamics, it is ok!

[11] [Fang-Lin-Tong 2023]

Short time to Long time: Approach 2

Approach 2: Remedy time-marching strategy by Uniform Singular Value Amplification (USVA) [11]

Motivation:

$$\begin{pmatrix} \frac{I+\Delta t A}{1+\Delta t\|A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{(1+\Delta t\|A\|)^L} & * \\ * & * \end{pmatrix}$$

Success probability: $\Omega\left(\frac{\|\psi(t)\|^2}{\|\psi(0)\|^2(1+\Delta t\|A\|)^{2L}}\right)$

$$\begin{pmatrix} \frac{I+\Delta t A}{\|I+\Delta t A\|} & * \\ * & * \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{(I+\Delta t A)^L}{\|I+\Delta t A\|^L} & * \\ * & * \end{pmatrix}$$

Observation: $\|I+\Delta t A\|^L \approx \left\|e^{\Delta t A}\right\|^L \sim \left\|e^{tA}\right\|$. For dissipative system and unitary dynamics, it is ok!

Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xRightarrow{\text{oblivious to the state}} \begin{pmatrix} \frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

[11] [Fang-Lin-Tong 2023]

Short time to Long time: Approach 2

Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xrightarrow[\text{QSVT}]{\text{oblivious to the state}} \begin{pmatrix} \frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

Short time to Long time: Approach 2

Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xrightarrow[\text{QSVT}]{\text{oblivious to the state}} \begin{pmatrix} \frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

Remark: For unitary dynamics, OAA works the magic! OAA only needs to boost a singular value, say $1/2$ to 1 using $3x - 4x^3$.)

Short time to Long time: Approach 2

Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xrightarrow[\text{QSVT}]{\text{oblivious to the state}} \begin{pmatrix} \frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

Remark: For unitary dynamics, OAA works the magic! OAA only needs to boost a singular value, say $1/2$ to 1 using $3x - 4x^3$.)
Here we need to boost all singular values uniformly.

Denote $\gamma := \alpha / \|\Xi\|$.

We seek for a polynomial approximation of γx on $[-\gamma^{-1}, \gamma^{-1}]$.

Short time to Long time: Approach 2

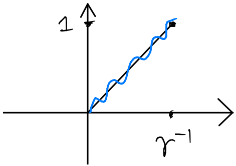
Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xrightarrow[\text{QSVT}]{\text{oblivious to the state}} \begin{pmatrix} \frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

Remark: For unitary dynamics, OAA works the magic! OAA only needs to boost a singular value, say $1/2$ to 1 using $3x - 4x^3$.) Here we need to boost all singular values uniformly.

Denote $\gamma := \alpha / \|\Xi\|$.

We seek for a polynomial approximation of γx on $[-\gamma^{-1}, \gamma^{-1}]$.



Short time to Long time: Approach 2

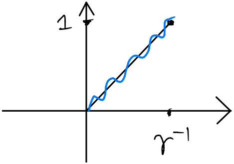
Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xrightarrow[\text{QSVT}]{\text{oblivious to the state}} \begin{pmatrix} (1-\delta)\frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

Remark: For unitary dynamics, OAA works the magic! OAA only needs to boost a singular value, say $1/2$ to 1 using $3x - 4x^3$.)
Here we need to boost all singular values uniformly.

Denote $\gamma := \alpha / \|\Xi\|$.

We seek for a polynomial approximation of $(1-\delta)\gamma x$ on $[-\gamma^{-1}, \gamma^{-1}]$.



Gibbs phenomena

Short time to Long time: Approach 2

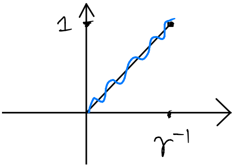
Needs:

$$\begin{pmatrix} \frac{\Xi}{\alpha} & * \\ * & * \end{pmatrix} \xrightarrow[\text{QSVT}]{\text{oblivious to the state}} \begin{pmatrix} (1-\delta)\frac{\Xi}{\|\Xi\|} & * \\ * & * \end{pmatrix}$$

Remark: For unitary dynamics, OAA works the magic! OAA only needs to boost a singular value, say $1/2$ to 1 using $3x - 4x^3$.) Here we need to boost all singular values uniformly.

Denote $\gamma := \alpha / \|\Xi\|$.

We seek for a polynomial approximation of $(1-\delta)\gamma x$ on $[-\gamma^{-1}, \gamma^{-1}]$.



Gibbs phenomena

Lemma(Uniform Singular Value Amplification)
$d = \mathcal{O}(\delta^{-1}\gamma \log(\gamma/\epsilon))$.

Short time to Long time: Approach 2

Algorithm per time step:

1. Numerical integrator (Dyson, Magnus, Euler, etc)
2. Uniform Singular Value Amplification (QSVT)

[Fang-Lin-Tong 2023 / arXiv 2022]

Short time to Long time: Approach 2

Algorithm per time step:

1. Numerical integrator (Dyson, Magnus, Euler, etc)
2. Uniform Singular Value Amplification (QSVT)

Construct a block-encoding of long-time evolution by compression gadget.

Generic Result for dissipative or near-unitary dynamics: The quantum algorithm makes

$$\mathcal{O}\left(qT^2 \,\mathrm{polylog}(T/\epsilon)\right)$$

queries to the oracle for A and

$$\mathcal{O}\left(q\right)$$

queries to the oracle preparing the initial state. (smoothness is not required; bounded variation is sufficient)

[Fang-Lin-Tong 2023 / arXiv 2022]

Short time to Long time: Approach 3

Approach 3: Complex analysis (Linear combination of Hamiltonian simulation) [12]

$$A(t) = L(t) + iH(t), \quad L(t) = \frac{A(t) + A^\dagger(t)}{2}, \quad H(t) = \frac{A(t) - A^\dagger(t)}{2i}.$$

Assume $L(t) \preceq 0$ for all $t \in \mathcal{I}$.

$$\mathcal{T}e^{-\int_0^t A(s)ds} = \int_{\mathbb{R}} \frac{1}{\pi(1+k^2)} \mathcal{T}e^{-i\int_0^t (H(s)+kL(s))ds} dk, \quad t \in \mathcal{I}.$$

12 [An-Lin-Liu 2023]

Short time to Long time: Approach 3

Approach 3: Complex analysis (Linear combination of Hamiltonian simulation) [12]

$$A(t) = L(t) + iH(t), \quad L(t) = \frac{A(t) + A^\dagger(t)}{2}, \quad H(t) = \frac{A(t) - A^\dagger(t)}{2i}.$$

Assume $L(t) \preceq 0$ for all $t \in \mathcal{I}$.

$$\mathcal{T} e^{-\int_0^t A(s)ds} = \int_{\mathbb{R}} \frac{1}{\pi(1+k^2)} \mathcal{T} e^{-i\int_0^t (H(s)+kL(s))ds} dk, \quad t \in \mathcal{I}.$$

Generic Result for dissipative dynamics: The quantum algorithm makes

$$\mathcal{O}\left(q^{2+2/p} T^{1+1/p} / \epsilon^{1+2/p}\right)$$

queries to the oracle for H and L and

$$\mathcal{O}\left(q\right)$$

queries to the oracle preparing the initial state.

[12] [An-Lin-Liu 2023]

Improved solver for special cases

Everything we discussed is on general quantum solvers (for general $A(t)$) and attaining lower-bound in the worst-case scenario.

This doesn't mean that the scaling for a specific case can not be further improved.

13 [Babbush-Berry-Kothari-Somma-Wiebe 2023]

Improved solver for special cases

Everything we discussed is on general quantum solvers (for general $A(t)$) and attaining lower-bound in the worst-case scenario.

This doesn't mean that the scaling for a specific case can not be further improved.

Simulating coupled classical oscillators [13]
a specific linear system with conservation laws.

[13] [Babbush-Berry-Kothari-Somma-Wiebe 2023]

Improved solver for special cases

Everything we discussed is on general quantum solvers (for general $A(t)$) and attaining lower-bound in the worst-case scenario.

This doesn't mean that the scaling for a specific case can not be further improved.

Simulating coupled classical oscillators [13]
a specific linear system with conservation laws.

For simplicity, consider $E(0) \equiv E(t) = \sum_{j=1}^{N} \frac{\dot{x}_j(t)^2}{2} + \frac{x_j(t)^2}{2}$.
Hamiltonian ODE: $\dot{y} = Ay$, $y = (x_1, \cdots, x_N, \dot{x}_1, \cdots, \dot{x}_N)$.

[13] [Babbush-Berry-Kothari-Somma-Wiebe 2023]

Improved solver for special cases

Everything we discussed is on general quantum solvers (for general $A(t)$) and attaining lower-bound in the worst-case scenario.

This doesn't mean that the scaling for a specific case can not be further improved.

Simulating coupled classical oscillators [13]
a specific linear system with conservation laws.

For simplicity, consider $E(0) \equiv E(t) = \sum_{j=1}^{N} \frac{\dot{x}_j(t)^2}{2} + \frac{x_j(t)^2}{2}$.
Hamiltonian ODE: $\dot{y} = Ay$, $y = (x_1, \cdots, x_N, \dot{x}_1, \cdots, \dot{x}_N)$.

Results (informal):

- It can be mapped to Hamiltonian simulation, and the quantum algorithm makes

$$\mathcal{O}(t + \log(1/\epsilon))$$

queries to the oracles representing the coefficient matrix.

[13] [Babbush-Berry-Kothari-Somma-Wiebe 2023]

Improved solver for special cases

Everything we discussed is on general quantum solvers (for general $A(t)$) and attaining lower-bound in the worst-case scenario.

This doesn't mean that the scaling for a specific case can not be further improved.

Simulating coupled classical oscillators [13]
a specific linear system with conservation laws.

For simplicity, consider $E(0) \equiv E(t) = \sum_{j=1}^{N} \frac{\dot{x}_j(t)^2}{2} + \frac{x_j(t)^2}{2}$.
Hamiltonian ODE: $\dot{y} = Ay$, $y = (x_1, \cdots, x_N, \dot{x}_1, \cdots, \dot{x}_N)$.

Results (informal):

- It can be mapped to Hamiltonian simulation, and the quantum algorithm makes

$$\mathcal{O}(t + \log(1/\epsilon))$$

  queries to the oracles representing the coefficient matrix.
- still BQP-hard.

[13] [Babbush-Berry-Kothari-Somma-Wiebe 2023]

Summary of Quantum Linear Differential Equation Solvers

- Definition of the task
- Challenges
- Ways to address (general solvers)
  - QSLA + Padding
  - Time-marching
  - LCHS
- Improved solver for specific cases, e.g., simulating coupled classical oscillators

Thank you for your attention!