SUPPLEMENTARY MATERIALS: VARIATIONAL QUANTUM SIMULATION OF GENERAL PROCESSES

REVIEW: VARIATIONAL QUANTUM SIMULATION OF REAL AND IMAGINARY TIME EVOLUTION

We first review the variational quantum algorithms for simulating real and imaginary time evolution introduced in Ref. [1] and [2], respectively, as we generalise these algorithms to propose our new algorithm for general processes. We refer the reader to Ref. [3] for a comprehensive derivation of quantum variational algorithms from various variational principles — the Dirac and Frenkel variational principle, the McLachlan's variational principle, and the time-dependent variational principle.

The real time evolution is described by the Schrödinger equation,

$$\frac{d\left|\psi(t)\right\rangle}{dt} = -iH\left|\psi(t)\right\rangle, \tag{1}$$

with Hermitian Hamiltonian H. Instead of directly simulating the real time dynamics with the Hamiltonian simulation algorithms [4–8], the variational method assumes that the quantum state $|\psi(t)\rangle$ is prepared by a parametrised quantum circuit, $|\varphi(\vec{\theta}(t))\rangle = R_N(\theta_N)\dots R_k(\theta_k)\dots R_1(\theta_1)\left|\bar{0}\right\rangle$ with each gate $R_k(\theta_k)$ controlled by the real parameter θ_k and the reference state $|\bar{0}\rangle$. Here, we denote $\vec{\theta} = (\theta_1, \theta_2, \dots, \theta_N)$. According to McLachlan's variational principle [9], the real time dynamics of $|\psi(t)\rangle$ can be mapped to the evolution of the parameters $\vec{\theta}(t)$ by minimising the distance between the ideal evolution and the evolution induced of the parametrised trial state,

$$\delta\|(\partial/\partial t + iH)\left|\varphi(\vec{\theta}(t))\right\rangle\| = 0, \tag{2}$$

where $\|\left|\varphi\right\rangle\| = \sqrt{\langle\varphi|\varphi\rangle}$. The solution is

$$\sum_j M_{k,j}\dot{\theta}_j = V_k, \tag{3}$$

with coefficients

$$\begin{aligned} M_{k,j} &= \mathrm{Re}\left(\frac{\partial\left\langle\varphi(\vec{\theta}(t))\right|}{\partial\theta_k}\frac{\partial\left|\varphi(\vec{\theta}(t))\right\rangle}{\partial\theta_j}\right), \\ V_k &= \mathrm{Im}\left(\left\langle\varphi(\vec{\theta}(t))\right| H\frac{\partial\left|\varphi(\vec{\theta}(t))\right\rangle}{\partial\theta_k}\right). \end{aligned} \tag{4}$$

For imaginary time evolution, the normalised Wick-rotated Shrödinger equation is obtained by replacing $t = i\tau$ in Eq. (1),

$$\frac{d\left|\psi(\tau)\right\rangle}{d\tau} = -(H - \langle\psi(\tau)|H|\psi(\tau)\rangle)\left|\psi(\tau)\right\rangle. \tag{5}$$

Applying a similar procedure for real time evolution, the imaginary time evolution is mapped to the evolution of the parameters via McLachlan's principle,

$$\delta\|(\partial/\partial\tau + H - \langle H\rangle)\left|\varphi(\vec{\theta}(t))\right\rangle\| = 0. \tag{6}$$

The evolution of the parameters is

$$\sum_j M_{k,j}\dot{\theta}_j = C_k, \tag{7}$$

with M given in Eq. (4) and C defined by

$$C_k = -\mathrm{Re}\left(\left\langle\varphi(\vec{\theta}(t))\right| H\frac{\partial\left|\varphi(\vec{\theta}(t))\right\rangle}{\partial\theta_k}\right). \tag{8}$$

The M, V, and C terms can be efficiently measured with quantum circuits. Considering gate based circuits, the derivative of the each parameterised gate can be expressed as

$$\frac{\partial R_k}{\partial \theta_k} = \sum_i g_{k,i} R_k \sigma_{k,i}, \tag{9}$$

where $\sigma_{k,i}$ are unitary operators and $g_{k,i}$ are complex coefficients. The derivative of the trial state can be written as

$$\frac{\partial \left|\varphi(\vec{\theta}(t))\right\rangle}{\partial \theta_k} = \sum_i g_{k,i} R_{k,i} \left|\bar{0}\right\rangle, \tag{10}$$

where

$$R_{k,i} = R_N R_{N-1} \cdots R_{k+1} R_k \sigma_{k,i} \cdots R_2 R_1. \tag{11}$$

The $M_{k,j}$ terms can be expressed as

$$M_{k,j} = \sum_{i,j} \Re\left(g^*_{k,p} g_{j,q} \left\langle \bar{0} \right| R^\dagger_{k,p} R_{j,q} \left| \bar{0} \right\rangle \right). \tag{12}$$

Similarly, considering sparse Hamiltonian with decomposition $H = \sum_j \lambda_j \sigma_j$, $\lambda_j \in \mathbb{R}$, we have C_k and V_k as

$$\begin{aligned} V_k &= \sum_{i,j} \mathrm{Re}\left(i g^*_{k,i} \lambda_j \left\langle \bar{0} \right| R^\dagger_{k,i} \sigma_j R \left| \bar{0} \right\rangle \right), \\ C_k &= -\sum_{i,j} \mathrm{Re}\left(g^*_{k,i} \lambda_j \left\langle \bar{0} \right| R^\dagger_{k,i} \sigma_j R \left| \bar{0} \right\rangle \right), \end{aligned} \tag{13}$$

All the M, C, and V terms can be written in the form

$$a \mathrm{Re}\left(e^{i\theta} \left\langle \bar{0} \right| U \left| \bar{0} \right\rangle \right),$$

where $a, \theta \in \mathbb{R}$ depend on the coefficients, and U is a unitary operator of either $R^\dagger_{k,p} R_{j,q}$ or $R^\dagger_{k,i} \sigma_j R$. We can calculate M, C, and V by using the quantum circuit shown in Fig. 1.

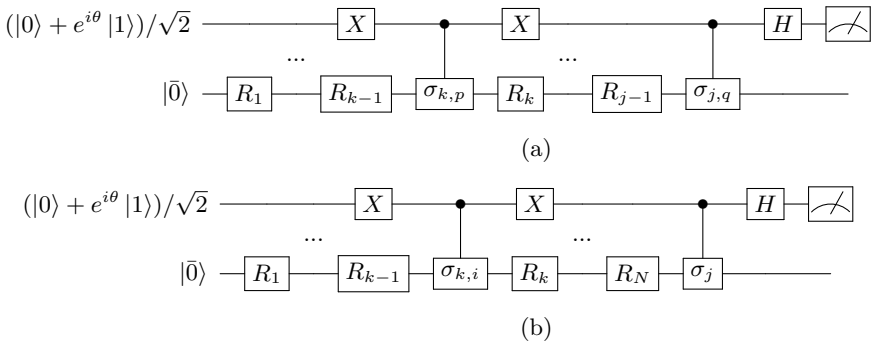

FIG. 1. Quantum circuits that evaluate (a) $\mathrm{Re}(e^{i\theta} \langle \bar{0}| R^\dagger_{k,p} R_{j,q} |\bar{0}\rangle)$ and (b) $\mathrm{Re}(e^{i\theta} \langle \bar{0}| R^\dagger_{k,i} \sigma_j R |\bar{0}\rangle)$.

DERIVATION FOR VARIATIONAL SIMULATION OF GENERALISED TIME EVOLUTION EQUATION

Now, we consider variational simulation of the generalised time evolution equation,

$$B(t) \frac{d}{dt} \left|v(t)\right\rangle = \sum_j A_j(t) \left|v'_j(t)\right\rangle. \tag{14}$$

By parametrising $|v(t)\rangle$ and $|v'_j(t)\rangle$ as $|v(\vec{\theta}(t))\rangle$ and $|v'(\vec{\theta}'_j(t))\rangle$, with McLachlan's principle, we have

$$\delta \left\| B(t) \sum_i \frac{\partial |v(\vec{\theta}(t))\rangle}{\partial \theta_i} \dot{\theta}_i - \sum_j A_j(t) |v'(\vec{\theta}'_j(t))\rangle \right\| = 0. \tag{15}$$

This is equivalent to

$$\begin{aligned} &\frac{\partial}{\partial \dot{\theta}_k} \left\| B(t) \sum_i \frac{\partial |v(\vec{\theta}(t))\rangle}{\partial \theta_i} \dot{\theta}_i - \sum_j A_j(t) |v'(\vec{\theta}'_j(t))\rangle \right\| \\ &= \frac{\partial}{\partial \dot{\theta}_k} \left(\sum_i \frac{\partial \langle v(\vec{\theta}(t))|}{\partial \theta_i} \dot{\theta}_i B^\dagger(t) - \sum_j \langle v'(\vec{\theta}'_j(t))| A_j^\dagger(t) \right) \\ &\left(\sum_l B(t) \frac{\partial |v(\vec{\theta}(t))\rangle}{\partial \theta_l} \dot{\theta}_l - \sum_j A_j(t) |v'(\vec{\theta}'_j(t))\rangle \right) = 0 \ \forall k. \end{aligned} \tag{16}$$

Hence, we have

$$\begin{aligned} &\sum_j \left(\frac{\partial \langle v(\vec{\theta}(t))|}{\partial \theta_k} B^\dagger(t) B(t) \frac{\partial |v(\vec{\theta}(t))\rangle}{\partial \theta_j} + \frac{\partial \langle v(\vec{\theta}(t))|}{\partial \theta_j} B^\dagger(t) B(t) \frac{\partial |v(\vec{\theta}(t))\rangle}{\partial \theta_k} \right) \dot{\theta}_j \\ &= \sum_j \frac{\partial \langle v(\vec{\theta}(t))|}{\partial \theta_k} B^\dagger(t) A_j(t) |v'_j(\vec{\theta}'_j(t))\rangle + h.c, \end{aligned} \tag{17}$$

which leads to Eq. (3) in the main text.

$$\sum_j \tilde{M}_{k,j} \dot{\theta}_j = \tilde{V}_k.$$

By substituting $|v(\vec{\theta}(t))\rangle = \alpha(\vec{\theta_0}(t)) |\varphi(\vec{\theta}_1(t))\rangle$ and $|v'(\vec{\theta}'_j(t))\rangle = \alpha'(\vec{\theta}'_{0j}(t)) |\varphi(\vec{\theta}'_{1j}(t))\rangle$, we have

$$\begin{aligned} \tilde{M}_{k,j} &= \mathrm{Re} \left(|\alpha(\vec{\theta}_0(t))|^2 \frac{\partial \langle \varphi(\vec{\theta}_1(t))|}{\partial \theta_k} B^\dagger(t) B(t) \frac{\partial |\varphi(\vec{\theta}_1(t))\rangle}{\partial \theta_j} \right) \\ &+ \mathrm{Re} \left(\frac{\partial \alpha^*(\vec{\theta}_0(t))}{\partial \theta_k} \alpha(\vec{\theta}_0(t)) \langle \varphi(\vec{\theta}_1(t))| B(t) \frac{\partial |\varphi(\vec{\theta}_1(t))\rangle}{\partial \theta_j} \right) \\ &+ \mathrm{Re} \left(\frac{\partial \alpha^*(\vec{\theta}_0(t))}{\partial \theta_j} \alpha(\vec{\theta}_0(t)) \langle \varphi(\vec{\theta}_1(t))| B(t) \frac{\partial |\varphi(\vec{\theta}_1(t))\rangle}{\partial \theta_k} \right) \\ &+ \mathrm{Re} \left(\frac{\partial \alpha(\vec{\theta}_0(t))}{\partial \theta_k} \frac{\partial \alpha^*(\vec{\theta}_0(t))}{\partial \theta_j} \langle \varphi(\vec{\theta}_1(t))| B^\dagger(t) B(t) |\varphi(\vec{\theta}_1(t))\rangle \right), \\ \tilde{V}_k &= \sum_j \mathrm{Re} \left(\frac{\partial \alpha^*(\vec{\theta}_0(t))}{\partial \theta_k} \alpha'(\vec{\theta}'_{0j}(t)) \langle \varphi(\vec{\theta}_1(t))| B^\dagger(t) A_j(t) |\varphi'_j(\vec{\theta}'_{1j}(t))\rangle \right). \\ &+ \sum_j \mathrm{Re} \left(\alpha^*(\vec{\theta}_0(t)) \alpha'(\vec{\theta}'_{0j}(t)) \frac{\partial \langle \varphi(\vec{\theta}_1(t))|}{\partial \theta_k} B^\dagger(t) A_j(t) |\varphi'_j(\vec{\theta}'_{1j}(t))\rangle \right). \end{aligned} \tag{18}$$

The first term of $\tilde{M}_{k,j}$ can written as

$$\sum_{i,q,l} \mathrm{Re} \Big(|\alpha(\vec{\theta}_0(t))|^2 g^*_{k,i} g_{j,q} \beta_l \langle \bar{0}| R^\dagger_{k,i} \sigma_l R_{j,q} |\bar{0}\rangle \Big), \tag{19}$$

where we set $B^\dagger(t)B(t) = \sum_l \beta_l \sigma_l$, and σ_l is a Pauli operator. Each term of $\tilde{M}_{k,j}$ can be written in the form of $a\,\mathrm{Re}(e^{i\theta}\langle\bar{0}|R_{k,i}^\dagger\sigma_l R_{j,q}|\bar{0}\rangle)$, where $a,\theta\in\mathbb{R}$. The quantum circuit to compute this value is shown in Fig. 2. It is worth mentioning that this quantum circuit only necessitates three controlled operations from an ancilla qubit. The second and third terms can be written as $\sum_{ij}\mathrm{Re}\big(\frac{\partial\alpha^*(\vec{\theta}_0(t))}{\partial\theta_k}\alpha(\vec{\theta}_0(t))\gamma_j g_{k,i}^*\langle\bar{0}|R_{k,i}^\dagger\sigma_j R|\bar{0}\rangle\big)$, where we set $B(t)=\sum_j\gamma_j\sigma_j$. We can express each term in the form of $a\,\mathrm{Re}(e^{i\theta}\langle\bar{0}|R_{k,i}^\dagger\sigma_l R|\bar{0}\rangle)$, which can be evaluated by using the quantum circuit shown in Fig. 1 (b). The fourth term can be simply computed by measuring the expectation value of $B^\dagger(t)B(t)=\sum_l\beta_l\sigma_l$ for $|\varphi(\vec{\theta}_1)\rangle$.

The term $\tilde{V}$ can be computed as follows. We denote $\tilde{U}_t$ and $\tilde{U}_t'$ to be the unitary circuit to prepare $|\varphi(\vec{\theta}_1(t))\rangle = \tilde{U}_t|\bar{0}\rangle$ and $|\varphi_j'(\vec{\theta}_{1j}'(t))\rangle = \tilde{U}_t'|\bar{0}\rangle$. Replacing $B^\dagger(t)A_j(t) = \sum_l\Lambda_l^j(t)\sigma_l$, the first term of each $\tilde{V}_k$ can be written as $\sum_{jl}\mathrm{Re}\big(\alpha^*(\vec{\theta}_0(t))\alpha'(\vec{\theta}_{0j}'(t))\Lambda_l^j\langle\bar{0}|\tilde{U}_t^\dagger\sigma_l\tilde{U}_t'|\bar{0}\rangle\big)$. Each term can be expressed in the form of $a\mathrm{Re}\big(e^{i\theta}\langle\bar{0}|\tilde{U}_t^\dagger\sigma_l\tilde{U}_t'|\bar{0}\rangle\big)$ and computed using the quantum circuit shown in Fig. 3. Meanwhile, the second term can be expressed as $\sum_{ijl}\mathrm{Re}(\alpha^*(\vec{\theta}_0(t))\alpha'(\vec{\theta}_{0j}(t))\Lambda_l^j(t)g_{k,i}^*\langle\bar{0}|R_{k,i}^\dagger\sigma_l\tilde{U}_t'|\bar{0}\rangle)$, and each term can be written in the form of $a\mathrm{Re}\big(e^{i\theta}\langle\bar{0}|R_{k,i}^\dagger\sigma_l\tilde{U}_t'|\bar{0}\rangle\big)$, which can be computed by using the quantum circuit shown in Fig. 4.

Note that, although the quantum circuits to evaluate $\tilde{V}$ generally need controlled U operations, in the case $|v'(\vec{\theta}_j'(t))\rangle = |v(\vec{\theta}(t))\rangle$, e.g., open quantum simulation and solving linear equations, it can be computed by the quantum circuit shown in Fig. 1 (b), which only needs two controlled operations.

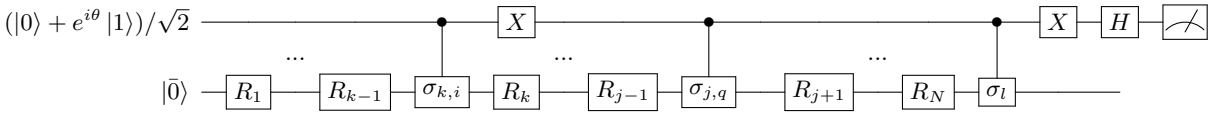

FIG. 2. Quantum circuits that evaluate $\mathrm{Re}\Big(g_{k,i}^* g_{j,q}\langle\bar{0}|R_{k,i}^\dagger\sigma_l R_{j,q}|\bar{0}\rangle\Big)$.

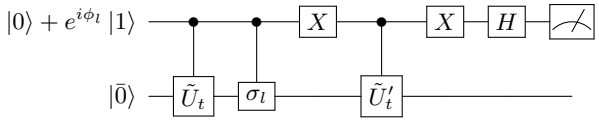

FIG. 3. The quantum circuit for evaluating $\tilde{V}_k$.

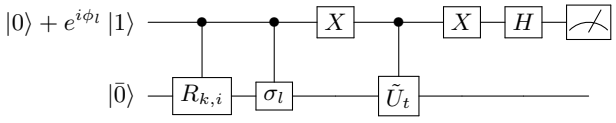

FIG. 4. The quantum circuit for evaluating $\tilde{V}_k$.

VARIATIONAL SIMULATION FOR LINEAR ALGEBRA TASKS

Matrix evolution with normalised states

Here, we also consider the case where we are only interested in the normalised final state $|\psi(t)\rangle = \mathcal{M}|v_0\rangle/\|\mathcal{M}|v_0\rangle\|$. By extrapolating from $|v_0\rangle/\||v_0\rangle\|$ to $|\psi(t)\rangle$, we can similarly have an evolution of the state $|\psi(t)\rangle$ as

$$|\psi(t)\rangle = E'(t)|\psi_0\rangle\,, \tag{20}$$

with

$$E'(t) = N(t)\Big(\frac{t}{T}\mathcal{M} + (1-\frac{t}{T})I\Big), \tag{21}$$

and a normalisation factor

$$N(t) = \frac{1}{\sqrt{\left\|\left(\frac{t}{T}\mathcal{M} + (1-\frac{t}{T})I\right)|\psi_0\rangle\right\|}}. \tag{22}$$

The normalisation factor $N(t)$ can be measured from the expectation values of $\mathcal{M}^\dagger + \mathcal{M}$ and $\mathcal{M}^\dagger\mathcal{M}$ for $|\psi_0\rangle$. Given the definition of the state $|\psi(t)\rangle$ at time t, the corresponding derivative equation is

$$\frac{d}{dt}|\psi(t)\rangle = \frac{\dot{N}(t)}{N(t)}|\psi(t)\rangle + N(t)G|\psi(0)\rangle. \tag{23}$$

Such an equation is also described by the generalised time evolution equation with $|v(t)\rangle = |\psi(t)\rangle$, $A_1(t) = \frac{\dot{N}(t)}{N(t)}I$, $A_2(t) = N(t)G$, $|v_1'(t)\rangle = |\psi(t)\rangle$, $|v_2'(t)\rangle = |\psi(0)\rangle$, and $B(t) = 1$.

Solving linear equations

We now discuss how to solve linear equations, defined by

$$\mathcal{M}|v_{\mathcal{M}^{-1}}\rangle = |v_0\rangle,$$

with vector $|v_0\rangle$, invertible matrix $\mathcal{M}$, and solution $|v_{\mathcal{M}^{-1}}\rangle = \mathcal{M}^{-1}|v_0\rangle$. We also introduce two algorithms for this problem where the first does not assume the structure of the matrix $\mathcal{M}$ and the second one assumes tensor product structure of $\mathcal{M}$.

For the first type of algorithm, we convert the static problem into a dynamical problem. We consider a path from the initial state $|v_0\rangle$ to $|v_{\mathcal{M}^{-1}}\rangle$ as

$$E(t)|v(t)\rangle = |v_0\rangle$$

with

$$E(t) = t/T\cdot\mathcal{M} + \big(1-t/T\big)I,$$

$|v(0)\rangle = |v_0\rangle$, and $|v(T)\rangle = |v_{\mathcal{M}^{-1}}\rangle$. That is, we can evolve the state $|v_0\rangle$ at time $t=0$ to the state $|v_{\mathcal{M}^{-1}}\rangle$ at time $t=T$. The derivation equation of $|v(t)\rangle$ is given by

$$E(t)\frac{d}{dt}|v(t)\rangle = -G(t)|v(t)\rangle$$

with $G(t) = \big(\mathcal{M}-I\big)/T$. It is a special case of a generalised time evolution defined in Eq. (1), with $B(t) = E(t)$, $A_j(t) = -G(t)$, and $|v_j'(\vec{\theta}_j'(t))\rangle = |v(\vec{\theta}(t))\rangle$. Therefore, we have the evolution equation as Eq. (3) with coefficients

$$\tilde{M}_{k,j} = \mathrm{Re}\left(\frac{\partial\langle v(\vec{\theta}(t))|}{\partial\theta_k}E^\dagger(t)E(t)\frac{\partial|v(\vec{\theta}(t))\rangle}{\partial\theta_j}\right),\ \tilde{V}_k = -\mathrm{Re}\left(\frac{\partial\langle v(\vec{\theta}(t))|}{\partial\theta_k}E^\dagger(t)G(t)|v(\vec{\theta}(t))\rangle\right).$$

We can calculate $\mathcal{M}^{-1}|v_0\rangle$ by measuring the coefficients and evolving the parameters accordingly. It is important to note that $\tilde{V}_k$ can be efficiently computed with the quantum circuit shown in Fig. 1. Therefore, linear equations can be solved with shallow quantum circuits which necessitate two or three controlled operations from an ancilla qubit.

For the second type of algorithm, we assume that $\mathcal{M} = UDV$ and the solution is given by

$$|v_{\mathcal{M}^{-1}}\rangle = V^\dagger D^{-1}U^\dagger|v_0\rangle.$$

With similar methods for realising matrix multiplication, we can also realise the matrix inversion operation with real and imaginary time evolutions.

RESOURCE ESTIMATION

In this section, we discuss the complexity of the variational algorithms. We give the resource estimation for achieving a desired simulation accuracy.

Resource estimation for simulating general processes

We first discuss the resource required for implementing the variational algorithm for the generalised time evolution. Suppose we aim to simulate the evolution

$$B(t)\frac{d}{dt}\left|v(t)\right\rangle = \sum_j A_j(t)\left|v_j'(t)\right\rangle, \tag{24}$$

from time $t=0$ to $t=T$. We approximate $|v(t)\rangle$ by $|v(\vec{\theta}(t))\rangle$ and evolve the parameters according to

$$\sum_j \tilde{M}_{k,j}\dot{\theta}_j = \tilde{V}_k, \tag{25}$$

with a time step δt. Here the definition of the coefficients can be found in the main text. The error of the algorithm can be described by

$$\varepsilon = D\left(\left|v(T)\right\rangle, \left|v(\vec{\theta}(T))\right\rangle\right), \tag{26}$$

where $D(\rho,\sigma) = \frac{1}{2}\mathrm{Tr}\left[|\rho-\sigma|\right]$ is the trace distance between two states. Denote $\mathcal{E}(t-\delta t,t)$ as the evolution from time $t-\delta t$ to t, then

$$\begin{aligned}
\varepsilon &= D\left(\mathcal{E}(T-\delta t,T)(|v(T-\delta t)\rangle), |v(\vec{\theta}(T))\rangle\right),\\
&\leq D\left(\mathcal{E}(T-\delta t,T)(|v(T-\delta t)\rangle), \mathcal{E}(T-\delta t,T)(|v(\vec{\theta}(T-\delta t))\rangle)\right) + D\left(\mathcal{E}(T-\delta t,T)(|v(\vec{\theta}(T-\delta t))\rangle), |v(\vec{\theta}(T))\rangle\right), \quad (27)\\
&\leq D\left(\left|v(T-\delta t)\right\rangle, \left|v(\vec{\theta}(T-\delta t))\right\rangle\right) + D\left(\mathcal{E}(T-\delta t,T)(|v(\vec{\theta}(T-\delta t))\rangle), |v(\vec{\theta}(T))\rangle\right).
\end{aligned}$$

Here the second line is due to the triangle inequality of the distance and the third line assumes that the evolution is dissipative in the sense that it can only decrease the distance. This is true for both real and imaginary time evolution and the open system simulation. Whether this is true for linear algebra problems depends on the problem. As this work mainly focuses on the simulation of open quantum systems, we leave a detailed discussion of the resource estimation for linear algebra problems in a future work.

Following a recursive procedure, we thus arrive at the final upper bound for the error of the simulation algorithm

$$\varepsilon \leq \sum_{t=\delta t:\delta t:T} D\left(\mathcal{E}(t-\delta t,t)(|v(\vec{\theta}(t-\delta t))\rangle), |v(\vec{\theta}(t))\rangle\right), \tag{28}$$

where we use $D\left(\left|v(0)\right\rangle, \left|v(\vec{\theta}(0))\right\rangle\right) = 0$ by assuming that the ansatz can perfectly represent the initial state at time $t=0$. For each term, we can also further divide it into two parts

$$D\left(\mathcal{E}(t-\delta t,t)(|v(\vec{\theta}(t-\delta t))\rangle), |v(\vec{\theta}(t))\rangle\right) = \delta\varepsilon_I + \delta\varepsilon_A, \tag{29}$$

with

$$\begin{aligned}
\delta\varepsilon_I &= D\left(\left|v(\vec{\theta}_0(t))\right\rangle, \left|v(\vec{\theta}(t))\right\rangle\right),\\
\delta\varepsilon_A &= D\left(\mathcal{E}(t-\delta t,t)(|v(\vec{\theta}(t-\delta t))\rangle), |v(\vec{\theta}_0(t))\rangle\right).
\end{aligned} \tag{30}$$

Here $|v(\vec{\theta}_0(t))\rangle$ is the ideal state obtained with exact values of $\tilde{M}$ and $\tilde{V}$. Therefore $\delta\varepsilon_I$ characterises the implementation error induced from imperfect $\tilde{M}$ and $\tilde{V}$; $\delta\varepsilon_A$ characterises the algorithmic error due to finite time step and insufficient ansatz. Overall, the total error is upper bounded by

$$\varepsilon \leq \varepsilon_I + \varepsilon_A = \sum_{t=\delta t:\delta t:T} (\delta\varepsilon_I + \delta\varepsilon_A), \tag{31}$$

with $\varepsilon_I = \sum_{t=\delta t:\delta t:T}\delta\varepsilon_I$ and $\varepsilon_A = \sum_{t=\delta t:\delta t:T}\delta\varepsilon_A$. In the following, we analyse these two terms separately.

Implementation error

The implementation error mainly comes from the imprecise estimation of $\tilde{M}$ and $\tilde{V}$ owing to either physical error or shot noise. Denote the exact value of $\tilde{M}$ and $\tilde{V}$ by $\tilde{M}_0$ and $\tilde{V}_0$, respectively. Then we have the exact derivative $\dot{\vec{\theta}}_0 = \tilde{M}_0^{-1}\tilde{V}_0$ and the practically measure derivative $\dot{\vec{\theta}} = \tilde{M}^{-1}\tilde{V}$. Denote $\tilde{M} = \tilde{M}_0 + \delta\tilde{M}$, $\tilde{V} = \tilde{V}_0 + \delta\tilde{V}$, and $\dot{\vec{\theta}} = \dot{\vec{\theta}}_0 + \delta\dot{\vec{\theta}}$ with $\delta\tilde{M}$, $\delta\tilde{V}$, and $\delta\dot{\vec{\theta}}$ representing the small noise perturbation. In Ref. [1], it is shown that the implementation error can be estimated by

$$\delta\varepsilon_I = D\left(|v(\vec{\theta}_0(t))\rangle, |v(\vec{\theta}(t))\rangle\right) = \sqrt{\delta\dot{\vec{\theta}}^T \mathcal{B}\delta\dot{\vec{\theta}}\delta t^2 + O(\delta t^3)}, \tag{32}$$

where

$$\mathcal{B}_{k,k'} = \frac{\partial \langle v(\vec{\theta}(t))|}{\partial\theta_k}\frac{\partial |v(\vec{\theta}(t))\rangle}{\partial\theta_{k'}} - \frac{\partial \langle v(\vec{\theta}(t))|}{\partial\theta_k}|v(\vec{\theta}(t))\rangle\langle v(\vec{\theta}(t))|\frac{\partial |v(\vec{\theta}(t))\rangle}{\partial\theta_{k'}}. \tag{33}$$

Here we assumed that the evolution of Eq. (24) is normalised, i.e., the state vector $|v(\vec{\theta}(t))\rangle$ is normalised.

The error of the parameters can be also represented as a function of $\delta\tilde{M}$ and $\delta\tilde{V}$,

$$\begin{aligned}
\delta\dot{\vec{\theta}} &= \tilde{M}^{-1}\tilde{V} - \tilde{M}_0^{-1}\tilde{V}_0, \\
&= \left(\tilde{M}_0(I + \tilde{M}_0^{-1}\delta\tilde{M})\right)^{-1}(\tilde{V}_0 + \delta\tilde{V}) - \tilde{M}_0^{-1}\tilde{V}_0 \\
&= (I + \tilde{M}_0^{-1}\delta\tilde{M})_0^{-1}\tilde{M}^{-1}(\tilde{V}_0 + \delta\tilde{V}) - \tilde{M}_0^{-1}\tilde{V}_0 \\
&\approx (I - \tilde{M}_0^{-1}\delta\tilde{M})(\tilde{M}_0^{-1}\tilde{V}_0 + \tilde{M}_0^{-1}\delta\tilde{V}) - \tilde{M}_0^{-1}\tilde{V}_0 \\
&\approx \tilde{M}_0^{-1}\delta\tilde{V} - \tilde{M}_0^{-1}\delta\tilde{M}\dot{\vec{\theta}}_0.
\end{aligned} \tag{34}$$

Here in the fourth line we consider a Taylor expansion with $\|\tilde{M}_0^{-1}\delta\tilde{M}\| \leq \|\tilde{M}_0^{-1}\|\|\delta\tilde{M}\| \ll 1$ and we omit higher order errors in the fifth line.

Therefore we can upper bound the implementation error as

$$\begin{aligned}
\delta\varepsilon_I = \sqrt{\delta\dot{\vec{\theta}}^T \mathcal{B}\delta\dot{\vec{\theta}}\delta t^2 + O(\delta t^3)} &\lesssim \sqrt{\|\mathcal{B}\|}\|\delta\dot{\vec{\theta}}\|\delta t, \\
&\lesssim \sqrt{\|\mathcal{B}\|}(\|\tilde{M}_0^{-1}\|\|\delta\tilde{V}\| + \|\tilde{M}_0^{-1}\|^2\|\tilde{V}_0\|\|\delta\tilde{M}\|)\delta t.
\end{aligned} \tag{35}$$

Suppose the dominant error is the shot noise, we can have the relationship between the implementation error and the number of samples. Given the number of measurements N_s for each term of $\tilde{M}$ and $\tilde{V}$, we have

$$\|\delta\tilde{M}\| \approx \frac{\sqrt{\sum_{j,k}\tilde{m}_{k,j}^2}}{\sqrt{N_S}}, \quad \|\delta\tilde{V}\| \approx \frac{\sqrt{\sum_k \tilde{v}_k^2}}{\sqrt{N_S}} \tag{36}$$

where

$$\begin{aligned}
\tilde{m}_{k,j} &= |\alpha(\vec{\theta}_0(t))|^2 \sum_{i,q}|g_{k,i}^* g_{j,q}|\|B(t)\|^2 + \left|\frac{\partial\alpha^*(\vec{\theta}_0(t))}{\partial\theta_k}\right||\alpha(\vec{\theta}_0(t))|\sum_i |g_{j,i}|\|B(t)\| \\
&+ \left|\frac{\partial\alpha^*(\vec{\theta}_0(t))}{\partial\theta_j}\right||\alpha(\vec{\theta}_0(t))|\sum_i |g_{k,i}|\|B(t)\| + \left|\frac{\partial\alpha^*(\vec{\theta}_0(t))}{\partial\theta_k}\right|\left|\frac{\partial\alpha^*(\vec{\theta}_0(t))}{\partial\theta_j}\right|\|B(t)\|^2 \\
\tilde{v}_k &= \sum_j \left[\left|\frac{\partial\alpha^*(\vec{\theta}_0(t))}{\partial\theta_k}\right||\alpha^*(\vec{\theta}'_{0j}(t))|\ \|A_j(t)\| + |\alpha^*(\vec{\theta}_0(t))|\ |\alpha^*(\vec{\theta}'_{0j}(t))|\sum_i |g_{k,i}|\ \|A_j(t)\|\right]\|B(t)\|,
\end{aligned} \tag{37}$$

and $g_{k,i}$ comes from

$$\frac{\partial |\varphi(\vec{\theta}(t))\rangle}{\partial\theta_k} = \sum_i g_{k,i}R_{k,i}|\bar{0}\rangle. \tag{38}$$

Therefore, $\|\delta\dot{\vec{\theta}}\|$ can be upper bounded by

$$\|\delta\dot{\vec{\theta}}\| \leq \frac{\Delta}{\sqrt{N_S}} \tag{39}$$

with

$$\Delta = \|\tilde{M}_0^{-1}\|^{-2}\|\tilde{V}_0\|\sqrt{\sum_{j,k}\tilde{m}_{k,j}^2} + \|\tilde{M}_0^{-1}\|\sqrt{\sum_k \tilde{v}_k^2}. \tag{40}$$

The implementation error at each step is thus upper bounded by

$$\delta\varepsilon_I \leq \sqrt{\|\mathcal{B}\|}\frac{\Delta}{\sqrt{N_S}}\delta t, \tag{41}$$

and the total implementation error is

$$\varepsilon_I \leq \sqrt{\|\mathcal{B}\|_{\max}}\frac{\Delta_{\max}}{\sqrt{N_S}}T, \tag{42}$$

where $\|\mathcal{B}\|_{\max}$ and $\Delta_{\max}$ denote the maximal possible value at different times. Therefore, in order to have an implementation error ε_I, we find that we can write

$$N_S = \|\mathcal{B}\|_{\max}\Delta_{\max}^2 T^2/\varepsilon_I^2.$$

Algorithmic error

The algorithmic error mainly comes from finite time step and imperfect ansatz. Following the definition of the time evolution, we have

$$\delta\varepsilon_A = D\left(\mathcal{E}(t-\delta t,t)(|v(\vec{\theta}(t-\delta t))\rangle), |v(\vec{\theta}_0(t))\rangle\right) = \sqrt{\Delta_2\delta t^2 + \Delta_3\delta t^3 + O(\delta t^4)} \lesssim \sqrt{\Delta_2^{(\max)}}\delta t + \sqrt{\Delta_3^{(\max)}\delta t}\delta t, \tag{43}$$

where the first term comes from imperfection of ansatz and the second term comes from finite time step [1]. Here

$$\Delta_2 = \langle\delta v_1(\vec{\theta})|\delta v_1(\vec{\theta})\rangle\,,\ \Delta_3 = \langle\delta v_1(\vec{\theta})|\delta v_2(\vec{\theta})\rangle + \langle\delta v_2(\vec{\theta})|\delta v_1(\vec{\theta})\rangle\,, \tag{44}$$

with

$$\begin{aligned}
|\delta v_1(\vec{\theta}(t))\rangle &= |dv(t)\rangle - \sum_j \dot{\theta}_j\frac{\partial}{\partial\theta_j}|v(\vec{\theta}(t))\rangle \\
|\delta v_2(\vec{\theta}(t))\rangle &= \frac{1}{2}\big(\frac{d}{dt}|dv(t)\rangle - \sum_{jj'}\dot{\theta}_j\dot{\theta}_{j'}\frac{\partial^2}{\partial\theta_j\partial\theta_{j'}}|v(\vec{\theta}(t))\rangle\big).
\end{aligned} \tag{45}$$

And $\Delta_2^{(\max)}$ and $\Delta_3^{(\max)}$ are the maximal possible values at time t, respectively.

Assuming that the ansatz can always represent the target state with $\Delta_2^{(\max)} = 0$, the total algorithmic error is upper bounded by

$$\varepsilon_A \lesssim \sqrt{\Delta_3^{(\max)}\delta t}T. \tag{46}$$

To suppress the effect of the second term to ε_A, we need to have $\delta t \approx \varepsilon_A^2/(\Delta_3^{(\max)}T^2)$, hence the number of steps required is $N_A = T/\delta t \approx \Delta_3^{(\max)}T^3/\varepsilon_A^2$.

Resource estimation

Now, we combine the results and show the overall resource cost. At each step, the number of required individual quantum circuits N_I is

$$N_I = N_{B^\dagger B} N_P^2 N_D^2 + 2N_B N_D N_P + N_{B^\dagger B} + N_B N_{A_j} N'_{A_j} (N_P N_D + 1), \tag{47}$$

where the first term and the second terms correspond to the number of individual quantum circuits to populate each element of $\tilde{M}$ and $\tilde{V}$, respectively. Here, N_P is the number of parameters, N_D is the number of terms in Eq. (38), N_{A_j}, N_B and $N_{B^\dagger B}$ are the number of Pauli terms to decompose the $A_j(t)$, $B(t)$ and $B^\dagger(t)B(t)$ matrices in Eq. (24), respectively, and N'_{A_j} is the number of $A_j(t)$ matrices (the number of the terms in the right hand side of Eq. (24)). Thus the total number of measurements N_{tot} required to suppress the implementation error by shot noise to ε_I and the effect of algorithmic error to ε_A is

$$\begin{aligned} N_{\text{tot}} &= N_I \times N_A \times N_S \\ &\approx \frac{\|\mathcal{B}\|_{\max} \Delta_{\max}^2 \Delta_3^{(\max)} T^5}{\varepsilon_A^2 \varepsilon_I^2} (N_{B^\dagger B} N_P^2 N_D^2 + 2N_B N_D N_P + N_{B^\dagger B} + N_B N_{A_j} N'_{A_j} (N_P N_D + 1)). \end{aligned} \tag{48}$$

By taking $\varepsilon_I = \varepsilon_A = \varepsilon/2$ with the total error ε, we have

$$\begin{aligned} N_{\text{tot}} &\approx N_I \times N_A \times N_S \\ &= \frac{16\|\mathcal{B}\|_{\max} \Delta_{\max}^2 \Delta_3^{(\max)} T^5}{\varepsilon^4} (N_{B^\dagger B} N_P^2 N_D^2 + 2N_B N_D N_P + N_{B^\dagger B} + N_B N_{A_j} N'_{A_j} (N_P N_D + 1)). \end{aligned} \tag{49}$$

Note that this resource estimation is only a pessimistic asymptotic estimation, while the realistic resource required can be much less for a given problem further with the optimisation of measurement schemes

Resource estimation for matrix multiplication via singular value decomposition method

In this section, we show the resource required for realising matrix multiplication via the singular value decomposition method, which is used in the variational algorithm for simulating open quantum systems. We leave the detailed resource analysis for the other variational algorithms of linear algebra problems in a future work.

Suppose the matrix $\mathcal{M}$ can be decomposed as $\mathcal{M} = UDV$, where U and V are unitary matrices and D is a diagonal matrix with non-negative entries. Suppose that U, V and D can be decomposed as $U = \exp(-iH^U T^U)$ and $V = \exp(-iH^V T^V)$ and $D \approx \exp(-H^D T^D)$, and these operations can be realised by variational real and imaginary time simulation algorithms. As we only approximate $D = \sum_j a_j |j\rangle\langle j|$ by $D \approx \exp(-H^D T^D)$ with $-H^D T = \sum_{a_j \neq 0} \log(a_j) |j\rangle\langle j| - \alpha \sum_{a_j=0} |j\rangle\langle j|$, we first analyse the error introduced from this approximation.

Accuracy of approximating D

Denote $D_\alpha = D + \Delta_\alpha$ with $D = \sum_{a_j \neq} a_j |j\rangle\langle j|$ and $\Delta_\alpha = \sum_{a_j=0} e^{-\alpha} |j\rangle\langle j|$. We use D_α to approximate D, for a given vector $|v\rangle$, the error of the matrix can propagate to the state as

$$\begin{aligned} \varepsilon_D =& \left\| \frac{D_\alpha |v\rangle\langle v| D_\alpha}{\langle v| D_\alpha^2 |v\rangle} - \frac{D |v\rangle\langle v| D}{\langle v| D^2 |v\rangle} \right\|_1 \\ =& \frac{1}{\langle v| D^2 |v\rangle} \left\| \frac{\langle v| D^2 |v\rangle}{\langle v| D_\alpha^2 |v\rangle} D_\alpha |v\rangle\langle v| D_\alpha - D |v\rangle\langle v| D \right\|_1, \\ =& \frac{1}{\langle v| D^2 |v\rangle} \left\| \frac{\langle v| D^2 |v\rangle}{\langle v| D_\alpha^2 |v\rangle} (D + \Delta_\alpha) |v\rangle\langle v| (D + \Delta_\alpha) - D |v\rangle\langle v| D \right\|_1, \\ =& \frac{1}{\langle v| D^2 |v\rangle} \left\| \left(\frac{\langle v| D^2 |v\rangle}{\langle v| D_\alpha^2 |v\rangle} - 1 \right) D |v\rangle\langle v| D + \frac{\langle v| D^2 |v\rangle}{\langle v| D_\alpha^2 |v\rangle} ((D + \Delta_\alpha) |v\rangle\langle v| (D + \Delta_\alpha) - D |v\rangle\langle v| D) \right\|_1, \\ \leq& \left| \frac{\langle v| D^2 |v\rangle}{\langle v| D_\alpha^2 |v\rangle} - 1 \right| + \frac{1}{\langle v| D^2 |v\rangle} \|(D + \Delta_\alpha) |v\rangle\langle v| (D + \Delta_\alpha) - D |v\rangle\langle v| D\|_1 . \end{aligned} \tag{50}$$

Here $\|\rho\|_1 = \mathrm{Tr}|\rho|$ is the trace norm of ρ. The first term can be bounded as

$$\left|\frac{\langle v|\, D^2\, |v\rangle}{\langle v|\, D_\alpha^2\, |v\rangle} - 1\right| = \left|\frac{\langle v|\, D_\alpha^2\, |v\rangle - \langle v|\, D^2\, |v\rangle}{\langle v|\, D_\alpha^2\, |v\rangle}\right| = \left|\frac{\langle v|\, \Delta_\alpha^2\, |v\rangle}{\langle v|\, D_\alpha^2\, |v\rangle}\right| \leq \frac{e^{-2\alpha}}{C}, \tag{51}$$

where $C = \langle v|\, D^2\, |v\rangle$ determines the norm after multiplying the matrix to the vector, which can be assumed to be lower bounded by a constant. As the value of C can be experimentally measured, the cases with an exponential small C can be experimentally identified. In the case with an exponentially small C, it just indicate that the output vector is a zero vector.

For the second term, we can bound it as

$$\begin{aligned}
&\frac{1}{\langle v|\, D^2\, |v\rangle} \left\|(D+\Delta_\alpha)\, |v\rangle \langle v|\, (D+\Delta_\alpha) - D\, |v\rangle \langle v|\, D\right\|_1 \\
\leq &\frac{1}{C} \left\|\Delta_\alpha\, |v\rangle \langle v|\, D + D\, |v\rangle \langle v|\, \Delta_\alpha + \Delta_\alpha\, |v\rangle \langle v|\, \Delta_\alpha\right\|_1 \\
\leq &\frac{1}{C} \left\|\Delta_\alpha\, |v\rangle \langle v|\, D\right\|_1 + \left\|D\, |v\rangle \langle v|\, \Delta_\alpha\right\|_1 + \left\|\Delta_\alpha\, |v\rangle \langle v|\, \Delta_\alpha\right\|_1 \\
= &\frac{e^{-2\alpha}}{C},
\end{aligned} \tag{52}$$

where we used $\|\Delta_\alpha\, |v\rangle \langle v|\, D\|_1 = \|D\, |v\rangle \langle v|\, \Delta_\alpha\|_1 = 0$. Therefore, the total error induced from the approximation of D_α is

$$\varepsilon_D = \frac{2e^{-2\alpha}}{C}, \tag{53}$$

and we can choose $\alpha = \ln(2/C\varepsilon_D)/2$. Therefore, when $C \geq 1/\mathrm{Poly}(n)$ with n denoting the number of qubits, we have

$$\alpha = O\left(\log\left(\frac{1}{\varepsilon_D}\right) + \log \mathrm{Poly}(n)\right). \tag{54}$$

We note that even when C is exponentially small with respect to n as $C = 1/c^n$ with constant c, we can still choose α to be a constant. Nevertheless, we should always measure the value of C at the beginning of the algorithm and directly output a zero vector when C is too small.

Resource analysis

According to our resource analysis for the generalised time evolution together with the error for approximating D, the implementation error is

$$\varepsilon_I \lesssim \sqrt{\|\mathcal{B}\|_{\max}} \frac{\Delta_{\max} T_{SVD}}{\sqrt{N_S}}, \tag{55}$$

and the algorithmic error is

$$\varepsilon_A \lesssim \sqrt{\Delta_3^{(\max)}} \delta t T_{SVD} + \varepsilon_D, \tag{56}$$

where $T_{SVD} = T^U + T^V + T^D$. Therefore, the total number of measurements N_{tot}^{SVD} required to suppress the effect of shot noise to ε_I and the effect of algorithmic error to ε_A is

$$N_{\mathrm{tot}}^{SVD} \approx \frac{\|\mathcal{B}\|_{\max} \Delta_{\max}^2 \Delta_3^{(\max)} T_{SVD}^5}{(\varepsilon_A - \varepsilon_D)^2 \varepsilon_I^2} (N_P^2 N_D^2 + N_P N_H N_D), \tag{57}$$

where N_H is the largest number of terms when H^U, H^V, and H^D is decomposed as a linear combination of Pauli operators.

Resource estimation for stochastic Schödinger equation

The resource estimation for the variational algorithm of open quantum systems is more involved. The simulation error consists of the following parts

1. The algorithmic error from approximating the Lindblad master equation with the stochastic Schrödinger equation with finite number of trajectory samples.

2. For each trajectory, we approximate the continuous evolution with the generalised time evolution and the jump process with the matrix multiplication algorithm.

3. For each jump, we need to estimate the jump probability and determining the jump time, whose error can also cause implementation errors.

For the first type of error, the error can be upper bounded to a small value with a sufficiently large number of sample proportional to a polynomial function of the inverse of the accuracy. For the second type of error, we can bound the error with the analysis for the generalised time evolution. For the jump process, we can use the analysis for matrix multiplication for each jump and we show in the following that the number of jumps is proportional to the evolution time. For the third type error, we can also bound it with a similar analysis for the generalised time evolution. In the following, we show how to estimate the number of jumps and we leave the detailed analysis of resource estimation in a future work.

Number of jumps

One trajectory of the stochastic Schrödinger equation is composed of the continuous evolution and jumps processes. The resource cost of the continuous evolution is similar to the one for real time evolution discussed in Ref. [1], which is shown to be polynomial to the evolution time and system size. For the jump processes, as each jump is simulated with real and imaginary time evolution, the resource cost of each jump is therefore also polynomial to the evolution time and system size.

Then we discuss how many jump process occur on average in the simulation of the Stochastic Schrödinger equation. The averaged number of jump events during time from t to $t + dt$ is

$$\langle\psi_c(t)| \sum_k L_k^\dagger L_k |\psi_c(t)\rangle \, dt. \tag{58}$$

Therefore, the average number of jump events from $t = 0$ to $t = T$

$$\begin{aligned} N_{\text{jump}} &= \int_0^T \langle\psi_c(t)| \sum_k L_k^\dagger L_k |\psi_c(t)\rangle \, dt, \\ &\leq T \left\| \sum_k L_k^\dagger L_k \right\|_\infty, \\ &\leq T \sum_k \left\| L_k^\dagger L_k \right\|_\infty, \end{aligned} \tag{59}$$

where $\|L\|_\infty$ is the operator norm, which is the largest singular value of L. For physical systems, we generally have $\|L_k^\dagger L_k\|_\infty$ and the number of Lindblad terms equal $O(\text{Poly}(n))$, where n is the system size and $\text{Poly}(n)$ is a polynomial function of n. Therefore, the averaged number of jumps is

$$N_{\text{jump}} = O(T \cdot \text{Poly}(n)). \tag{60}$$

The number of jumps is much fewer when considering the case where each Lindblad operator only locally acts on a constant subsystem. That is, we assume that $L_k^\dagger L_k$ has orthogonal support to each other and $\|L_k^\dagger L_k\| = O(1)$. In this case, we have $\left\| \sum_k L_k^\dagger L_k \right\|_\infty = O(1)$ and hence

$$N_{\text{jump}} = O(T). \tag{61}$$

To simulate the stochastic Schrödinger equation, we also need to sample different random trajectories. When we measure an observable O and hope to suppress the sampling error to $\epsilon = 1/\sqrt{M}$, we need M random samples. Therefore, the overall cost should also be multiplied by M. However, it is worth noting that every trajectory is exactly parallel so the overhead M can be also reduced by a constant factor.

[1] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).
[2] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, npj Quantum Information **5**, 1 (2019).
[3] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).
[4] S. Lloyd, Science , 1073 (1996).
[5] M. Suzuki, Journal of Mathematical Physics **32**, 400 (1991).
[6] A. M. Childs, A. Ostrander, and Y. Su, Quantum **3**, 182 (2019).
[7] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Physical review letters **114**, 090502 (2015).
[8] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019).
[9] A. McLachlan, Molecular Physics **8**, 39 (1964).