Title	Fast-forwarding with NISQ processors without feedback loop
Author(s)	Kian Hwee Lim, Tobias Haug, Leong Chuan Kwek and Kishor Bharti

Fast-Forwarding with NISQ Processors without Feedback Loop

Kian Hwee Lim,[1,*] Tobias Haug,[2] Leong Chuan Kwek,[1,3,4] and Kishor Bharti[1,†]

[1] *Centre for Quantum Technologies, National University of Singapore 117543, Singapore*

[2] *QOLS, Blackett Laboratory, Imperial College London SW7 2AZ, UK*

[3] *National Institute of Education, Nanyang Technological University, 1 Nanyang Walk, Singapore 637616*

[4] *MajuLab, CNRS-UNS-NUS-NTU International Joint Research Unit, UMI 3654, Singapore*

Simulating quantum dynamics is expected to be performed more easily on a quantum computer than on a classical computer. However, the currently available quantum devices lack the capability to implement fault-tolerant quantum algorithms for quantum simulation. Hybrid classical quantum algorithms such as the variational quantum algorithms have been proposed to effectively use current term quantum devices. One promising approach to quantum simulation in the noisy intermediate-scale quantum (NISQ) era is the diagonalisation based approach, with some of the promising examples being the subspace Variational Quantum Simulator (SVQS), Variational Fast Forwarding (VFF), fixed-state Variational Fast Forwarding (fs-VFF), and the Variational Hamiltonian Diagonalisation (VHD) algorithms. However, these algorithms require a feedback loop between the classical and quantum computers, which can be a crucial bottleneck in practical application. Here, we present the Classical Quantum Fast Forwarding (CQFF) as an alternative diagonalisation based algorithm for quantum simulation. CQFF shares some similarities with SVQS, VFF, fs-VFF and VHD but removes the need for a classical-quantum feedback loop and controlled multi-qubit unitaries. The CQFF algorithm does not suffer from the barren plateau problem and the accuracy can be systematically increased. Furthermore, if the Hamiltonian to be simulated is expressed as a linear combination of tensored-Pauli matrices, the CQFF algorithm reduces to the task of sampling some many-body quantum state in a set of Pauli-rotated bases, which is easy to do in the NISQ era. We run the CQFF algorithm on existing quantum processors and demonstrate the promise of the CQFF algorithm for current-term quantum hardware. We compare CQFF with Trotterization for a XY spin chain model Hamiltonian and find that the CQFF algorithm can simulate the dynamics more than 10^5 times longer than Trotterization on current-term quantum hardware. This provides a 10^4 times improvement over the previous record.

I. INTRODUCTION

In the 1980s Feynman suggested that since nature is quantum-mechanical, it would be easier to simulate a quantum system on a quantum computer rather than a classical computer [1]. His idea has far-reaching implications on fields such as chemistry and materials science. Many quantum simulation algorithms have been proposed since then, with Trotterization being one of the most prominent [2]. Trotterization, however, requires an extensive amount of quantum resources and most likely can only be implemented on fault-tolerant quantum computers [3]. Hence, in the current noisy intermediate-scale quantum (NISQ) [4] era when we do not yet have fault-tolerant quantum computers, alternate methods have to be used. In the NISQ era, most algorithms make use of a classical quantum feedback loop [5, 6]. In each iteration, a classical computer runs a classical optimization routine to determine a set of parameters for a parameterized quantum circuit to minimize an appropriate cost function. On the quantum computer, the cost function is calculated and then sent as input to the classical computer for the next iteration. The canonical example of such a quantum algorithm is the variational quantum eigensolver [7, 8]. For the task of quantum simulation, various algorithms have been proposed, such as the Variational Quantum Simulator (VQS) [9, 10], Subspace Variational Quantum Simulator (SVQS) [11], Quantum Assisted Simulator (QAS) [12, 13], Variational Fast Forwarding (VFF) [14], fixed-state Variational Fast Forwarding (fs-VFF) [15], Generalised Quantum Assisted Simulator (GQAS) [16], Variational Hamiltonian Diagonalisation (VHD) [17], projected-Variational Quantum Dynamics (p-VQD) [18] and the truncated Taylor quantum simulator (TTQS) [19]. These algorithms allow to simulate quantum dynamics beyond the coherence time possible with Trotterization in the NISQ era.

In this work, we focus on diagonalisation based approaches, i.e. the SVQS, VFF, fs-VFF and VHD algorithms. The "no fast-forwarding theorem" tells us that for a quantum system, simulating the time evolution with respect to a generic Hamiltonian H for time T requires at least a number of gates that scales linearly with T, which means that in general it is not possible to perform quantum simulation with a sublinear amount of resources [20, 21]. It has been shown that a given Hamiltonian can be fast-forwarded if and only if it corresponds to violations of the time-energy uncertainty relations and

* kianhwee_lim@u.nus.edu

† kishor.bharti1@gmail.com

equivalently allows for precise energy measurements [22]. For a discussion on the implications of asymptotic fast-forwarding on quantum simulation with NISQ devices, refer to [14].

The main idea of the NISQ quantum simulation algorithms based on diagonalisation is to variationally find a unitary transformation into a space such that the time evolution can be easily performed with a fixed circuit structure. We now proceed to review the aforementioned algorithms. First, for the SVQS algorithm, the idea is to first variationally search for a unitary transformation into the subspace spanned by the low-lying energy eigenstates of the Hamiltonian H. Then, the time evolution in that subspace can be easily done by just single-qubit Z-rotations on each qubit. Next, for the VFF algorithm, the idea is to first variationally find the unitary transformation to diagonalise the time evolution operator $e^{-iH\Delta t}$ with a small timestep Δt. Then, one applies the diagonal evolution operator multiple times, which can be done without requiring additional resources, allowing one to fast-forward to large evolution times. The fs-VFF algorithm is a modification of the VFF algorithm, where the observation is made that if the initial state to be evolved $|\psi_0\rangle$ lies in the span of n_{eig} energy eigenstates, then $e^{-iHT}|\psi_0\rangle$ lies in the same span. In the fs-VFF algorithm, diagonalisation of $e^{-iH\Delta t}$ is not done over the entire Hilbert space, but only on the n_{eig} dimensional subspace that $|\psi_0\rangle$ lies in. Finally, for the VHD algorithm, the idea is to first variationally find the unitary transformation to diagonalise the Hamiltonian operator H. Then, H can be easily exponentiated to obtain e^{-iHT}, which which can be done without requiring additional resources. For these diagonalisation based variational quantum simulation algorithms, the bulk of the work is to approximately variationally diagonalise an operator.

The variational approaches described above suffer from a few weaknesses, namely:

1. For variational quantum algorithms, the classical quantum feedback loop can be a major bottleneck when running the algorithm on current cloud-based quantum computers, as each job for the quantum computer has to wait in a queue. For each iteration, one needs to wait for the result from the quantum computer, which can take an extensive amount of time. The SVQS algorithm requires two classical quantum feedback loops, one for the initial state preparation and another for variationally searching for the unitary transform into the space spanned by the low lying energy eigenstates. The VFF/fs-VFF and the VHD algorithms require one classical quantum feedback loop.

2. For variational quantum algorithms, there can potentially be barren plateaus when number of qubits, hardware noise or entanglement increases [23–28], which lead to an exponential decrease in the variance of the gradient and rendering training of these variational approaches very challenging.

3. The fs-VFF algorithm requires the Hadamard test in the computation of n_{eig}, the VHD algorithm requires the Hadamard test in the computation of the cost function and the gradients, and the VFF algorithm requires the use of the Local Hilbert-Schmidt test. The Hadamard test is hard to do in the NISQ era due to it requiring controlled multi-qubit unitaries. The Local Hilbert-Schmidt test does not require controlled multi-qubit unitaries but it requires many controlled single-qubit unitary gates between qubits which are physically far separated. Depending on the type of NISQ quantum computer, this operation may require many SWAP gates and could be a potential bottleneck.

Here, we propose another diagonalisation based algorithm which we call the Classical Quantum Fast Forwarding (CQFF) algorithm. Our algorithm solves all of the aforementioned challenges. Some of the features of our algorithm are:

1. The CQFF algorithm has no classical quantum feedback loop unlike the SVQS, VFF, fs-VFF and VHD algorithms.

2. The CQFF algorithm avoids the barren plateau problem as there is no parametrized quantum circuit that is being updated.

3. The CQFF algorithm has a systematic way of constructing the ansatz and the simulation result can always be improved by considering a higher value of K when computing $\mathbb{CS}_K$.

4. For the CQFF algorithm, the quantum processor's output can be easily computed without requiring any controlled multi-qubit unitaries, such as those required in the Hadamard test.

II. THE CQFF ALGORITHM

Our ansatz is a hybrid state, which is a classical combination of L quantum states

$$|\psi(\boldsymbol{\alpha}(t))\rangle = \sum_{i=1}^{L} \alpha_i(t)\,|\chi_i\rangle\ , \tag{1}$$

where $\boldsymbol{\alpha}(t) \in \mathbb{C}^L$ and $\{|\chi_i\rangle\}_{i=1}^{L}$ is a set of L quantum states. We now want to evolve a state $|\psi(\boldsymbol{\alpha}(t=0))\rangle$ for a time T under a given Hamiltonian H with $i\partial_t|\psi(t)\rangle = H|\psi(t)\rangle$. We assume that the Hamiltonian H that we want to simulate is given as a linear combination of r unitaries

$$H = \sum_{i=1}^{r} \beta_i U_i\,, \tag{2}$$

where $\beta_i \in \mathbb{C}$ and the N-qubit unitaries $U_i \in \mathrm{SU}(2^N \equiv \mathcal{N})$, for $i \in \{1, 2, \ldots r\}$. Moreover, each unitary acts

non-trivially on at most $\mathcal{O}(poly(logN))$ qubits. If the unitaries in Eq. (2) are tensored-Pauli matrices, then we do not need the $\mathcal{O}(poly(logN))$ constraint.

Now, we have to choose the states $|\chi_i\rangle$ in our hybrid ansatz. Our goal is that the states $\{|\chi_i\rangle\}$ span the space of the evolution. We use a NISQ friendly approach first put forward in [29], which is based on the idea of Krylov expansion. Given a matrix A, vector b and some scalar τ, $\exp(\tau A)b$ can be approximated as some order m polynomial which can be reformulated as an element in the following Krylov subspace [30],

$$\mathcal{K}_m = \text{span}\left\{b, Ab, \cdots, A^{m-1}b\right\}. \tag{3}$$

One can improve the approximation accuracy by increasing m. For the case where A is the Hamiltonian and b is a quantum state, the Krylov subspace based approximation for imaginary or real time evolution is a natural choice. This motivates us to construct the states $|\chi_i\rangle$ of our hybrid ansatz using the r unitaries U_i in the Hamiltonian in Eq. (2). We choose states $|\chi_i\rangle$ from the set $\mathbb{CS}_K$, which we call the set of cumulative K-moment states. Briefly speaking, we have $\mathbb{CS}_K = \mathbb{S}_0 \cup \mathbb{S}_1 \cup \cdots \cup \mathbb{S}_K$, where $\mathbb{S}_0 = \{|\phi\rangle\}$, and $\mathbb{S}_p$ for $1 \leq p \leq K$ is defined as

$$\mathbb{S}_p = \{U_{i_p} \ldots U_{i_2} U_{i_1} |\phi\rangle\}_{i_1=1,\ldots i_p=1}^{r}, \tag{4}$$

where the unitaries U_{i_a} are those in the Hamiltonian in Eq. (2). We assume that the state $|\phi\rangle$ can be efficiently prepared on a quantum computer. A formal definition and examples of $\mathbb{CS}_K$ are given in Appendix B. We also justify in Appendix B how the ansatz states $|\chi_i\rangle$ from the set $\mathbb{CS}_K$ capture the time evolution with a given Hamiltonian. The initial state for $t = 0$ can be the quantum state prepared on the quantum computer with $\boldsymbol{\alpha}(t = 0)$ such that $|\psi(\boldsymbol{\alpha}(t = 0)\rangle = |\phi\rangle$, or more generally we can choose $\boldsymbol{\alpha}(t = 0)$ as linear combination of states in $\mathbb{CS}_K$. Note that $|\phi\rangle$ is the only state that needs to be prepared on the quantum computer.

Next, we compute the D and E matrices on the quantum computer, where the matrix elements of D and E matrices are given by

$$D_{ij} = \langle\chi_i|H|\chi_j\rangle = \sum_{a=1}^{r} \beta_a \langle\chi_i|U_a|\chi_j\rangle \tag{5}$$

$$E_{ij} = \langle\chi_i|\chi_j\rangle. \tag{6}$$

We note that if the unitaries U_i are just tensored-Pauli matrices, then the calculation of these matrix elements just reduces to the problem of sampling the state $|\phi\rangle$ in some Pauli-rotated basis. Otherwise, since the unitaries U_i acts trivially only on $\mathcal{O}(poly(log(N)))$ qubits, we can use the methods in [31] to compute the expectation values without the need for Hadamard tests or complicated controlled multi-qubit unitaries. The only task of the quantum computer is to calculate D and E matrix. Thus, if the unitaries U_i are just tensored-Pauli matrices, then we have mapped the task of Hamiltonian simulation to a circuit sampling task.

Once we have the D and E matrices, the job of the quantum computer is done; what remains is the classical post-processing stage with three steps. We give a short summary of the classical post-processing stage. Firstly, we use the D and E matrices to compute a diagonal representation of H. Next, we use the diagonal representation of H to trivially compute a diagonal representation of e^{-iHt}. Lastly, we perform the time evolution to find the vector $\boldsymbol{\alpha}(t)$ in Eq. (1).

We now detail the post-processing steps. The evolution of the hybrid ansatz, which we constructed using the evolution Hamiltonian H, is assumed to be approximately constrained within the space spanned by $\mathbb{CS}_K$. We define the projected Hamiltonian $[H]_{\mathbb{CS}_K}$ (see Appendix A for full derivation) that projects the full H onto the space spanned by hybrid state ansatz given by the cumulative K-moment states $\mathbb{CS}_K$

$$[H]_{\mathbb{CS}_K} = \sum_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\dagger E, \tag{7}$$

where λ_i is the i-th eigenvalue and $\boldsymbol{v}_i$ the i-th eigenvector of the generalized eigenvalue problem defined below [29, 32, 33]

$$D\boldsymbol{v} = \lambda E\boldsymbol{v}. \tag{8}$$

Note that in general E may not be full rank, however the eigenvectors corresponding to the nullspace of E do not contribute in the subsequent equations in the paper and can be safely ignored (see Appendix A). The above generalised eigenvalue problem is related to the following Quadratically Constrained Quadratic Program (QCQP)

$$\begin{aligned} &\min_{\boldsymbol{v}} (\boldsymbol{v}^\dagger D\boldsymbol{v}) \\ &\text{subject to } \boldsymbol{v}^\dagger E\boldsymbol{v} = 1, \end{aligned} \tag{9}$$

which is a well characterised optimisation program. This QCQP has been studied before in the context of finding the ground state energy of a particular Hamiltonian [34]. We can recover the generalised eigenvalue problem from the QCQP by introducing a Lagrange function $L(\boldsymbol{v}, \lambda)$, and finding its stationary points

$$L(\boldsymbol{v}, \lambda) = \boldsymbol{v}^\dagger D\boldsymbol{v} + \lambda(1 - \boldsymbol{v}^\dagger E\boldsymbol{v}) \tag{10}$$

$$\frac{\partial L}{\partial \boldsymbol{v}} = 0 \implies D\boldsymbol{v} = \lambda E\boldsymbol{v}. \tag{11}$$

From Eq. (7) and the observation that $\boldsymbol{v}_i^\dagger E\boldsymbol{v}_j = \delta_{ij}$ (see Appendix A), we can write the evolution unitary within the space spanned by the hybrid states as

$$\left[e^{-iHT}\right]_{\mathbb{CS}_K} = \sum_j e^{-i\lambda_j T} \boldsymbol{v}_j \boldsymbol{v}_j^\dagger E. \tag{12}$$

Using the notation that $[|\psi(\boldsymbol{\alpha}(T))\rangle]_{\mathbb{CS}_K}$ denotes the coordinate vector of $|\psi(\boldsymbol{\alpha}(T))\rangle$ with respect to the set of

states $\mathbb{CS}_K$, we see that the evolution of $\boldsymbol{\alpha}(T)$ to a time T is given by

$$\begin{aligned}\boldsymbol{\alpha}(T) &= [|\psi(\boldsymbol{\alpha}(T))\rangle]_{\mathbb{CS}_K} \\ &= \left[e^{-iHT}\,|\psi(\boldsymbol{\alpha}(0))\rangle\right]_{\mathbb{CS}_K} = [e^{-iHT}]_{\mathbb{CS}_K}\boldsymbol{\alpha}(0)\,. \quad (13)\end{aligned}$$

Eq. (12) is the reason why this method is called "Classical-Quantum Fast Forwarding"; essentially what we are doing is calculating the D and E matrices on the quantum computer, using those matrices to find a diagonal representation of H on a classical computer, and finally using that to get e^{-iHT} with a simple exponentiation of the eigenvalues. After the fast forwarding, Eq. (13) gives us $\boldsymbol{\alpha}(T)$ and hence $|\psi(\boldsymbol{\alpha}(T))\rangle$, which is the expression for the time evolution under H of $|\psi(\boldsymbol{\alpha}(t=0)\rangle$ expressed as a linear combination of the states in $\mathbb{CS}_K$. The accuracy of our simulation can be improved by increasing the value of K in the definition of $\mathbb{CS}_K$.

We shall summarise the CQFF algorithm below as follows:

1. Get the hybrid ansatz state Eq. (1) with classical parameters $\boldsymbol{\alpha}(t)$ and quantum states $|\chi_i\rangle$ generated from an efficiently preferable state $|\phi\rangle$ and Hamiltonian H.
2. Compute the D and E matrices on the quantum computer, with the matrix elements given in Eq. (5) and Eq. (6).
3. Solve the generalised eigenvalue problem in Eq. (8) and use Eq. (13) to get the time evolved parameters of the hybrid state $\boldsymbol{\alpha}(T)$ for some initial $\boldsymbol{\alpha}(0)$.

III. RESULTS

We first use the CQFF algorithm to simulate the time evolution of the Heisenberg model, given by the following Hamiltonian

$$H_1 = \sum_{j=1}^{N-1} X_jX_{j+1} + 2Y_jY_{j+1} + 3Z_jZ_{j+1}\,, \quad (14)$$

where N is the number of qubits. Here, we consider the 2 and 3 qubit cases. Using IBM's quantum processor *ibmq_rome*, we prepare a random initial state $|\phi\rangle$ on the quantum computer (see appendix E for more details). We also used the quantum computer to calculate the matrix elements in Eq. (5) and Eq. (6), and the calculation of each matrix element is done by just sampling the state $|\phi\rangle$ in some Pauli-rotated basis. After obtaining the D and E matrices, the job of the quantum computer is done; to finally obtain the time evolution of $|\psi(\boldsymbol{\alpha}(t=0)\rangle = |\phi\rangle$, we use the classical computer to perform the fast forwarding in accordance with Eq. (12) and Eq. (13). To verify the results of the time evolution, we compare the exact and experimental values found for $\boldsymbol{\alpha}(T)$ by computing

$$|\psi(T)_{\text{theoretical}}\rangle = e^{-iHT}\,|\phi\rangle \quad (15)$$

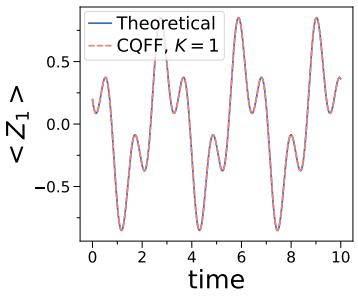

FIG. 1. Time evolution of CQFF for 2 qubits with Hamiltonian H_1, simulated on the IBM quantum processor *ibmq_rome* with 8192 shots. The expectation value $\langle Z_1\rangle$ is shown here. The fidelity of the state remains 1 for the entire time evolution and is hence omitted here. We plot the long time behavior of the fidelity in Appendix G.

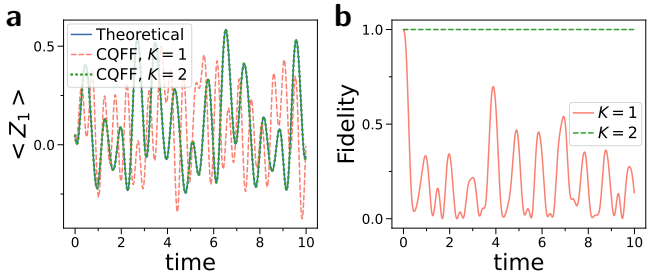

FIG. 2. Time evolution of CQFF on a 3 qubit state with Hamiltonian H_1, simulated on the IBM quantum processor *ibmq_rome* with 8192 shots. **a)** Expectation value $\langle Z_1\rangle$. The CQFF $K = 2$ values completely match with the exact values. **b)** Fidelity of the state. We also plot the long time behavior of the fidelity in Appendix G

classically as well as

$$|\psi(T)_{\text{CQFF}}\rangle = \sum_{i=1}^{L} \alpha_i(T)\,|\chi_i\rangle\,. \quad (16)$$

We calculate fidelity over time given as $F(t) = |\langle\psi(t)_{\text{theory}}|\psi(t)_{\text{CQFF}}\rangle|^2$ and the time variation of the expectation value $\langle Z_1\rangle$, which we denote as $\langle Z_1(t)\rangle$. It is easy to see that we have

$$\langle Z_1(t)\rangle = \sum_{i}^{L}\sum_{j}^{L} \alpha_i(t)^*\,\langle\chi_i|Z_1|\chi_j\rangle\,\alpha_j(t). \quad (17)$$

Here, we focus on the ability of our algorithm to accurately obtain the coefficients of the time evolution $\boldsymbol{\alpha}(T)$ in Eq. (16) using the quantum computer. While $\boldsymbol{\alpha}(T)$ is calculated using the D and E matrices that were computed on the quantum computer, the overlap terms for the expectation values $\langle\chi_i|Z_1|\chi_j\rangle$ are computed classically. For completeness, we also show the 2 qubit case in Appendix G where terms like $\langle\chi_i|Z_1|\chi_j\rangle$ are computed on the quantum computer as well. The results for

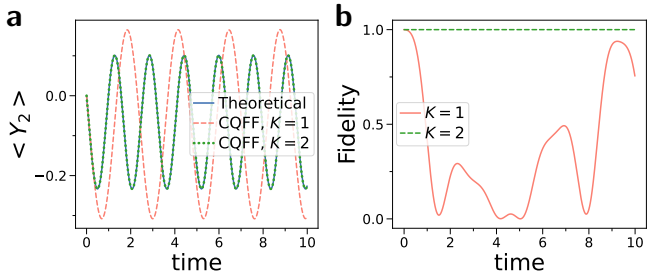

FIG. 3. Time evolution of CQFF on a 4 qubit state with Hamiltonian H_2, simulated on the IBM quantum processor *ibmq_rome* with 8192 shots. Here, we have $J_{zxz} = 1$. **a)** Expectation value of $\langle Y_2 \rangle$. The CQFF $K = 2$ values completely match with the exact values. **b)** Fidelity of the state. We also plot the long time behavior of the fidelity in Appendix G.

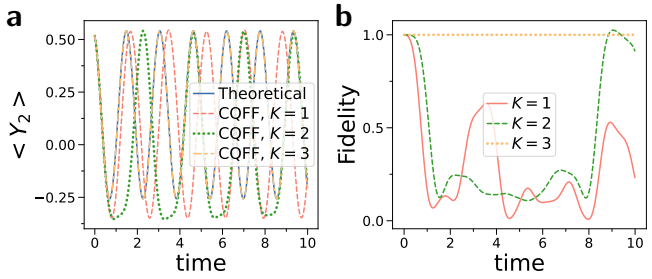

FIG. 4. Time evolution of CQFF on a 5 qubit state with Hamiltonian H_2, simulated on the IBM quantum processor *ibmq_rome* with 8192 shots. Here, we have $J_{zxz} = 1$. **a)** Expectation value $\langle Y_2 \rangle$ **b)** Fidelity of the state. We plot the long time behavior of the fidelity in Appendix G.

both the fidelity $F(t)$ and for $\langle Z_1(t) \rangle$ are shown in Fig. 1 and Fig. 2 for the 2 and 3 qubit cases.

Next, we use the CQFF algorithm to simulate a three-body Hamiltonian proposed in [35]

$$H_2 = J_{zxz} \sum_{k=1}^{N} Z_{k-1} X_k Z_{k+1} \,. \tag{18}$$

We consider the 4 and 5 qubit cases, and follow the same procedure as for the Heisenberg model above on the same quantum computer *ibmq_rome*. The random initial state $|\phi\rangle$ is also prepared the same way. To verify our results, we compute the time variation of the fidelity $F(t)$ as well as the time variation of the expectation value $\langle Y_2 \rangle$, which we denote as $\langle Y_2(t) \rangle$. As before, coefficients $\boldsymbol{\alpha}(T)$ are calculated using the D and E matrices that were computed with the quantum computer, while the terms $\langle \chi_i | Y_2 | \chi_j \rangle$ needed for the expectation values are calculated using classical computers. The results are shown in Fig. 3 and Fig. 4. We see that for both the Hamiltonians H_1 and H_2 considered, we could essentially perform quantum simulation for up to $t = 10$ by considering a large enough value for K.

Next, we benchmark the CQFF algorithm against Trotterization on the IBM quantum processor *ibmq_rome*. To date, the best known fast forwarding on real quantum hardware has been [15], which managed to maintain a fidelity of at least 0.9 for at least 600 Trotter steps. Here, we show that we can surpass this by a factor of about 10^4. Like [15], we consider the time-evolution of the state $|\psi(t = 0)\rangle = |10\rangle$ under the 2-qubit XY spin chain Hamiltonian given by

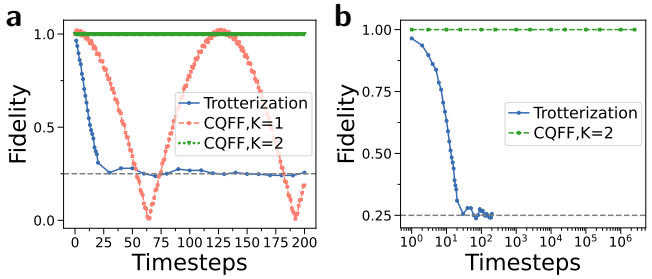

FIG. 5. Time evolution of both CQFF and Trotterization for the 2-qubit state $|10\rangle$ under the Hamiltonian H_3, simulated on the IBM quantum processor *imbq_rome* with 8192 shots. Here, we used the first-order Trotter approximation with a timestep $\Delta t = 0.5$. The grey dashed line at $F = 0.25$ represents the overlap with the maximally mixed state. **a)** Plotted on a linear scale, up to 200 time steps **b)** Plotted on a log scale, up to 2500000 timesteps. The fidelity for CQFF $K = 1$ is still oscillatory even at long time scales and is hence omitted in the log plot.

$$H_3 = X_1 X_2 + Y_1 Y_2 \,. \tag{19}$$

We time evolve the state $|\psi(t = 0)\rangle$ using both Trotterization and CQFF, where for Trotterization, we consider the first-order Trotter approximation with a timestep $\Delta t = 0.5$. We evaluate the quality of our simulations by plotting the fidelity $F = \langle \psi(t) | \rho(t) | \psi(t) \rangle$ against the number of Trotter steps N, where $|\psi(t)\rangle$ here is the exact evolution and $\rho(t)$ is the simulated evolution. For Trotterization the fidelity is computed by first performing tomography, whereas since CQFF uses the hybrid ansatz in Eq. (1), there is no need for tomography. The results are shown in Fig. 5. The grey dashed line at $F = 0.25$ represents the fidelity with the maximally mixed state. As can be seen, Trotterization breaks down at around 25 Trotter steps due to decoherence. On the other hand, for the entire period of the time evolution, CQFF for $K = 1$ exhibits F that is sinusoidally varying between 0 and slightly greater than 1, and CQFF for $K = 2$ has a fidelity of 1. The fidelity being slightly greater than 1 is due to numerical errors in the E-matrix. As seen in the log plot of Fig. 5, the same behavior for $K = 2$ continues even on a longer timescale, at 2500000 Trotter steps. Hence, we can conclude that on current term quantum hardware, the CQFF algorithm with $K = 2$ allows for a fast-forwarding of about 10^5 times compared to the coherence time of the quantum computer, which is essentially indefinitely long.

IV. DISCUSSION

The accuracy of the CQFF algorithm increases as we consider larger K in $\mathbb{CS}_K$. As K increases, the set $\mathbb{CS}_K$ is able to better span the space of states necessary for time evolution of the initial state under H. Furthermore, the quantum computer can calculate the D and E matrices in a single step without any feedback loop. This also means that any noise due to the computation on the quantum computer only affects our algorithm in this single step. More information about error analysis is given in Appendix C.

For the examples considered above, we just needed to sample the state $|\phi\rangle$ in some Pauli-rotated basis, which can be done efficiently on NISQ quantum devices in contrast to more complicated routines like the Hadamard test which requires controlled multi-qubit unitaries. The barren plateau problem is also avoided by construction, as there is no parameterized quantum circuit that is being updated.

For the CQFF algorithm, further studies on the scaling of number of states in $\mathbb{CS}_K$ with the number of qubits need to be performed. The rate of growth in the number of states in $\mathbb{CS}_K$ as the number of qubits increases depends highly on the Hamiltonian; for example, H_2 has lesser number of states in $\mathbb{CS}_K$ as compared to H_1 for the same number of qubits. Here, we propose that since we are using hybrid states for our ansatz, as defined in Eq. (1), whether a Hamiltonian can be fast-forwarded with the CQFF algorithm depends on the growth of the number of states in $\mathbb{CS}_K$ with the number of qubits. It remains an open question to study which Hamiltonians can be fast-forwarded. More work can also be done to study how one might reduce the states in $\mathbb{CS}_K$ for a given Hamiltonian H. Lastly, more work has to be done to study how the choice of the state $|\phi\rangle$ affects the scaling and performance of the CQFF algorithm.

We note that CQFF as formulated above is more similar to VHD than VFF. One can also tweak the CQFF algorithm to make it more similar to VFF. This can be done by using CQFF to diagonalise a small Δt approximation of $U(\Delta t) = e^{-iH\Delta t}$ instead of H. Then, with the diagonal representation of $U(\Delta t)$, we can easily get $U(N\Delta t)$ (See Appendix D for more details). However, this has no advantages in our framework and is instead disadvantageous as the small Δt approximation introduces errors that are avoided by directly diagonalising H as per the original CQFF algorithm.

The CQFF algorithm can also make use of a classical quantum feedback loop if we allow for the quantum states $|\chi_i\rangle$ defining our hybrid ansatz in Eq. (1) to be variationally adjusted. In other words, we can define a hybrid ansatz like

$$|\psi(\boldsymbol{\theta}, \boldsymbol{\alpha}(t))\rangle = \sum_{i=1}^{L} \alpha_i(t) \, |\chi_i(\boldsymbol{\theta})\rangle \tag{20}$$

where $|\chi_i(\boldsymbol{\theta})\rangle = U_i \, |\phi(\boldsymbol{\theta})\rangle$ and U_i here is a product of unitaries in $\mathbb{U}$. Using this idea, we can control our state $|\psi(\boldsymbol{\theta}, \boldsymbol{\alpha}(t))\rangle$ both variationally by updating $\boldsymbol{\theta}$ and classically by updating $\boldsymbol{\alpha}(t)$.

Appendix A: Justification of CQFF algorithm

Here, we first want to find the representation matrix of H with respect to the set $\mathbb{CS}_K$, which we define as $[H]_{\mathbb{CS}_K}$. $[H]_{\mathbb{CS}_K}$ is given by the following equation:

$$[H]_{\mathbb{CS}_K} = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 \dots \boldsymbol{u}_L \end{pmatrix} \tag{A1}$$

where $\boldsymbol{u}_i = [H \, |\chi_i\rangle]_{\mathbb{CS}_K}$ is the coordinate vector of $H \, |\chi_i\rangle$ with respect to the set $\mathbb{CS}_K$. From Eq. (A1), we see that if we define:

$$B = \begin{pmatrix} |\chi_1\rangle & |\chi_2\rangle & \dots & |\chi_L\rangle \end{pmatrix} \tag{A2}$$

we have:

$$\begin{aligned} HB &= H \begin{pmatrix} |\chi_1\rangle & |\chi_2\rangle & \dots & |\chi_L\rangle \end{pmatrix} \\ &= \begin{pmatrix} H \, |\chi_1\rangle & H \, |\chi_2\rangle & \dots & H \, |\chi_L\rangle \end{pmatrix} \\ &= B[H]_{\mathbb{CS}_K} \, . \end{aligned} \tag{A3}$$

Note that defining B in Eq. (A2) also leads us to two very helpful expressions:

$$B^\dagger B = E \tag{A4}$$

$$B^\dagger H B = D \tag{A5}$$

where $B^\dagger$ is:

$$B^\dagger = \begin{pmatrix} \langle\chi_1| \\ \langle\chi_2| \\ \vdots \\ \langle\chi_L| \end{pmatrix} \, . \tag{A6}$$

Hence if we multiply $B^\dagger$ from the left to both sides of Eq. (A3), we will get:

$$D = E[H]_{\mathbb{CS}_K} \, . \tag{A7}$$

We will show next that $[H]_{\mathbb{CS}_K}$ can be related to the eigenvectors $\boldsymbol{v}_i$ obtained from solving the generalised eigenvalue problem in Eq. (8).

When solving the generalised eigenvalue problem in Eq. (8), we note that E in general is not full rank and hence some of the eigenvectors $\{\boldsymbol{v}_i\}$ obtained will belong to the nullspace of E. Since D is a Hermitian matrix and E is a positive semi-definite matrix, the eigenvectors $\boldsymbol{v}_i$ that are in the column space of E can always be chosen to be orthonormal with respect to E. In other words, we can always have:

$$\boldsymbol{v}_i^\dagger E \boldsymbol{v}_j = \delta_{ij} \text{ if } \boldsymbol{v}_i \text{ and } \boldsymbol{v}_j \in \text{Col}(E) \tag{A8}$$

$$\boldsymbol{v}_i^\dagger E \boldsymbol{v}_j = 0 \text{ if } \boldsymbol{v}_i \text{ or } \boldsymbol{v}_j \in \text{Null}(E) \tag{A9}$$

where $\mathrm{Col}(E)$ stands for the column space of E, $\mathrm{Null}(E)$ stands for the nullspace of E, and δ_{ij} here is the Kronecker-Delta symbol. Now, consider the expression:

$$\sum_k \boldsymbol{v}_i^\dagger E \boldsymbol{v}_k \boldsymbol{v}_k^\dagger E \boldsymbol{v}_j = \boldsymbol{v}_i^\dagger E \left(\sum_k \boldsymbol{v}_k \boldsymbol{v}_k^\dagger E \right) \boldsymbol{v}_j \,. \tag{A10}$$

If we then choose $\boldsymbol{v}_i$ and $\boldsymbol{v}_j \in \mathrm{Col}(E)$, we have:

$$\boldsymbol{v}_i^\dagger E \left(\sum_k \boldsymbol{v}_k \boldsymbol{v}_k^\dagger E \right) \boldsymbol{v}_j = \sum_k \delta_{ik}\delta_{kj} = \delta_{ij} \tag{A11}$$

which then immediately leads to the following completeness relation:

$$\sum_j \boldsymbol{v}_j \boldsymbol{v}_j^\dagger E = \mathbb{1} \,. \tag{A12}$$

Note that the expression $\left(\sum_k \boldsymbol{v}_k \boldsymbol{v}_k^\dagger E\right)$ is the same regardless of whether we include the vectors $\boldsymbol{v}_k \in \mathrm{Null}(E)$. Hence from now on, we will ignore these nullspace eigenvectors in Eq. (A12).

Now, what we do is to multiply the D matrix from the left in Eq. (A12) to get

$$\begin{aligned} D &= D \sum_j \boldsymbol{v}_j \boldsymbol{v}_j^\dagger E \\ \implies D &= E \left(\sum_j \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^\dagger E \right) \,. \end{aligned} \tag{A13}$$

Direct comparison of Eq. (A13) with Eq. (A7) gives us

$$[H]_{\mathbb{CS}_K} = \sum_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\dagger E + N \,, \tag{A14}$$

where N is a matrix whose columns are in the nullspace of E. We can then use Eq. (A14) and Eq. (A8) to arrive at

$$\left[e^{-iHT}\right]_{\mathbb{CS}_K} = \sum_i e^{-i\lambda_i T} \boldsymbol{v}_i \boldsymbol{v}_i^\dagger E + N' \,, \tag{A15}$$

where N' is another matrix whose columns are in the nullspace of E. Now, for the initial state $|\psi(\boldsymbol{\alpha}(0))\rangle$ to be a valid quantum state, it must be normalised, which corresponds to the condition $\boldsymbol{\alpha}(0)^\dagger E \boldsymbol{\alpha}(0) = 1$. I.e, $\boldsymbol{\alpha}(0)$ must be in the column space of E. This means that

$$\begin{aligned} \left[e^{-iHT}\right]_{\mathbb{CS}_K} \boldsymbol{\alpha}(0) &= \left(\sum_i e^{-i\lambda_i T} \boldsymbol{v}_i \boldsymbol{v}_i^\dagger E \right) \boldsymbol{\alpha}(0) + N' \boldsymbol{\alpha}(0) \\ &= \left(\sum_i e^{-i\lambda_i T} \boldsymbol{v}_i \boldsymbol{v}_i^\dagger E \right) \boldsymbol{\alpha}(0) + \boldsymbol{n} \,, \end{aligned} \tag{A16}$$

where $\boldsymbol{n}$ is a vector in the nullspace of E. Now, we can safely ignore the vector $\boldsymbol{n}$, since it corresponds to the zero ket in the Hilbert space; to see why, we write

$$|\boldsymbol{n}\rangle = \sum_i n_i |\chi_i\rangle \tag{A17}$$

and compute $\langle \boldsymbol{n}|\boldsymbol{n}\rangle$:

$$\langle \boldsymbol{n}|\boldsymbol{n}\rangle = \boldsymbol{n}^\dagger E \boldsymbol{n} = 0 \,. \tag{A18}$$

Since $\langle \boldsymbol{n}|\boldsymbol{n}\rangle = 0$, $|\boldsymbol{n}\rangle$ must be the zero ket, and hence we can ignore $\boldsymbol{n}$ in Eq. (A16). In the end, we arrive at:

$$\left[e^{-iHT}\right]_{\mathbb{CS}_K} \boldsymbol{\alpha}(0) = \left(\sum_i e^{-i\lambda_i T} \boldsymbol{v}_i \boldsymbol{v}_i^\dagger E \right) \boldsymbol{\alpha}(0) \tag{A19}$$

which is exactly the CQFF time evolution equation given in Eq. (13).

Appendix B: Cumulative K-moment states

Here, we first formally define the $\mathbb{CS}_K$ states before we explain how they can capture Hamiltonian time evolution. The definition of $\mathbb{CS}_K$ here is taken from [29].

Definition 1. (Adapted from [29].) Given a set of unitaries $\mathbb{U} \equiv \{U_i\}_{i=1}^r$, a positive integer K and some quantum state $|\phi\rangle$, the K-moment states is the set of quantum states of the form $\{U_{i_K} \dots U_{i_2} U_{i_1} |\phi\rangle\}_i$ for $U_{il} \in \mathbb{U}$. We denote the aforementioned set by $\mathbb{S}_K$. The singleton set $\{|\phi\rangle\}$ will be referred to as the 0-moment state (denoted by $\mathbb{S}_0$). The cumulative K-moment states $\mathbb{CS}_K$ is defined to be $\mathbb{CS}_K \equiv \cup_{j=0}^K \mathbb{S}_j$.

In this paper, if the Hamiltonian H is given by Eq. (2), we shall let $\mathbb{U}$ be the set of unitaries that make up our Hamiltonian H, and we will use $\mathbb{U}$ to construct $\mathbb{CS}_K$. We then have:

$$\begin{aligned} \mathbb{CS}_0 &= \mathbb{S}_0 \\ &= \{|\phi\rangle\} \\ \mathbb{CS}_1 &= \mathbb{CS}_0 \cup \mathbb{S}_1 \\ &= \{|\phi\rangle\} \cup \{U_{i_1} |\phi\rangle\}_{i_1=1}^r, \\ \mathbb{CS}_2 &= \mathbb{CS}_1 \cup \mathbb{S}_2 \\ &= \{|\phi\rangle\} \cup \{U_{i_1} |\phi\rangle\}_{i_1=1}^r \cup \{U_{i_2} U_{i_1} |\phi\rangle\}_{i_1=1,i_2=1}^r \\ &\vdots \\ \mathbb{CS}_K &= \mathbb{CS}_{K-1} \cup \mathbb{S}_K \end{aligned}$$

where, $U_{i_l} \in \mathbb{U}$.

The motivation for using the set $\mathbb{CS}_K$ to construct our ansatz can be found in [12], but we will reproduce the main gist of their argument here. Starting with $|\psi(t=0)\rangle = |\phi\rangle$, the time evolution $|\psi(t)\rangle = e^{-iHt} |\psi(t=0)\rangle$ can be approximated as

$$|\psi(t)\rangle = e^{-iHt} |\phi\rangle \approx p_{m-1}(-iHt) |\phi\rangle \,, \tag{B1}$$

where p_{m-1} is some $m-1$ degree polynomial[30]. Now, $p_{m-1}(-iHt)\left|\phi\right\rangle$ is an element of the Krylov subspace $\mathcal{K}_m$, which is defined as

$$\mathcal{K}_m = \text{span}\{\left|\phi\right\rangle, H\left|\phi\right\rangle, \ldots, H^{m-1}\left|\phi\right\rangle\}. \tag{B2}$$

Hence, the problem of Hamiltonian time evolution up to a time t can be reframed as the problem of finding a linear combination of vectors in $\mathcal{K}_m$, where the coefficients of that linear combination are functions of t. We note that if the Hamiltonian H has rank r, then the approximation in Eq. (B1) becomes exact for some $m \leq r+1$. Now, if we assume that our Hamiltonian H is a linear combination of unitaries as per Eq. (2), then it is easy to see that

$$\mathcal{K}_{K+1} \subset \mathbb{CS}_K \tag{B3}$$

and hence by using a linear combination of states in $\mathbb{CS}_K$ for our ansatz, we leverage on the power of the Krylov subspace K_{K+1} to approximate the time evolution $e^{-iHt}\left|\phi\right\rangle$.

If the unitaries in our Hamiltonian H are tensored-Pauli operators, then the set $\mathbb{CS}_K$ has a lot more mathematical structure that can be exploited, due to the Lie Algebra structure of the tensored-Pauli operators. This means that in some cases, we need to calculate a lot less overlaps on the quantum computer than what is initially suggested by the construction of the set $\mathbb{CS}_K$.

Appendix C: Error analysis

Due to the lack of the classical-quantum feedback loop in the CQFF algorithm, any error in the CQFF algorithm for a given value of K can be traced back to the computation of the matrix elements $E_{ij} = \langle\chi_i|\chi_j\rangle$ and $D_{ij} = \langle\chi_i|H|\chi_j\rangle$ on the quantum computer, where $\left|\chi_i\right\rangle, \left|\chi_j\right\rangle \in \mathbb{CS}_K$. In this section we first discuss the errors in the computation of those matrix elements before discussing how these errors are propagated in our algorithm.

Firstly, the sources of error involved in the computation of E_{ij} and D_{ij} on the quantum computer are:

1. Shot noise error due to a finite number of shots.
2. Initial state preparation error in the preparation of $\left|\phi\right\rangle$.
3. Final measurement error.

Since the Hamiltonians considered in this paper are tensored-Pauli operators, which we denote by P_k, the matrix elements computed on the quantum computer can be obtained by sampling the initial state $\left|\phi\right\rangle$ in some Pauli-rotated basis. In this case, the shot noise error incurred in the computation $\langle\phi|P_k|\phi\rangle$ can be bounded by using a result from [24], which we reproduce here for convenience:

Theorem 1. (Adapted from [24].) *Let $\epsilon > 0$ and P_k be a tensored-Pauli operator over n qubits. Let multiple copies of an arbitrary n-qubit quantum state $\left|\phi\right\rangle$ be given. The expectation value $\langle\phi|P_k|\phi\rangle$ can be determined to additive accuracy ϵ with failure probability at most δ using $\mathcal{O}(\frac{1}{\epsilon^2}\log(\frac{1}{\delta}))$ copies of $\left|\phi\right\rangle$.*

The final measurement error can be somewhat mitigated by calibrating the POVM matrix, which rotates from an ideal set of counts to one affected by measurment noise. This can be done using the *"CompleteMeasFitter"* function in the IBMQ circuit library.

To summarise thus far, the three sources of error mentioned above are ultimately captured as errors in the matrix elements of the D and E matrices. This means that at the end of the day, we just need to look at the D and E matrices. Then, errors in the E_{ij} and D_{ij} matrix elements are propagated in our algorithm firstly through Eq. (8), which we use to determine the generalised eigenvalues λ_i and the corresponding eigenvectors $\boldsymbol{v}_i$. I.e, these errors in the D_{ij} and E_{ij} matrix elements lead to errors in the generalised eigenvalues λ_i and the generalised eigenvectors $\boldsymbol{v}_i$. Subsequently, errors in λ_i and $\boldsymbol{v}_i$ would lead to errors in $\boldsymbol{\alpha}(T)$ through Eq. (13). We will now make these statements more precise.

Let E, D be the ideal, noiseless E and D matrices and let $\tilde{E}$, $\tilde{D}$ be the matrices whose matrix elements are computed on the quantum computer. Let the corresponding generalised eigenvalue problems be $D\boldsymbol{v} = \lambda E\boldsymbol{v}$ and $\tilde{D}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{E}\tilde{\boldsymbol{v}}$ respectively. If we denote the noiseless CQFF result as

$$\left|\psi(\boldsymbol{\alpha}(T))\right\rangle = \sum_i \alpha_i(T)\left|\chi_i\right\rangle \tag{C1a}$$

$$\boldsymbol{\alpha}(T) = \sum_j e^{-i\lambda_j T}\boldsymbol{v}_j\boldsymbol{v}_j^\dagger E\boldsymbol{\alpha}(0) \tag{C1b}$$

and the noisy CQFF result as

$$\left|\psi(\tilde{\boldsymbol{\alpha}}(T))\right\rangle = \sum_i \tilde{\alpha}_i(T)\left|\chi_i\right\rangle \tag{C2a}$$

$$\tilde{\boldsymbol{\alpha}}(T) = \sum_j e^{-i\tilde{\lambda}_j T}\tilde{\boldsymbol{v}}_j\tilde{\boldsymbol{v}}_j^\dagger \tilde{E}\boldsymbol{\alpha}(0) \tag{C2b}$$

then we have

$$\begin{aligned}\langle\psi(\boldsymbol{\alpha}(T))|\psi(\tilde{\boldsymbol{\alpha}}(T))\rangle &= \boldsymbol{\alpha}(T)^\dagger E\tilde{\boldsymbol{\alpha}}(T) \\ &= \boldsymbol{\alpha}(0)^\dagger K\boldsymbol{\alpha}(0)\end{aligned} \tag{C3}$$

where

$$K = \sum_{jj'} e^{-i(\tilde{\lambda}_{j'}-\lambda_j)T}E\boldsymbol{v}_j\boldsymbol{v}_j^\dagger E\tilde{\boldsymbol{v}}_{j'}\tilde{\boldsymbol{v}}_{j'}^\dagger\tilde{E}. \tag{C4}$$

Hence, we see that the error in λ_i hence leads to an error that is periodic in time, whereas the error in $\boldsymbol{v}_i$ lead to an error that is constant in the simulation time T except when at $T = 0$, where the error vanishes since $\sum_i \tilde{\boldsymbol{v}}_i\tilde{\boldsymbol{v}}_i^\dagger E = I$.

Now, the goal is to find a bound for the difference in eigenvalues $|\lambda_i - \tilde{\lambda}_i|$ and also to find a way to quantify the difference in the corresponding eigenspaces with the eigenvectors $\boldsymbol{v}_j$ and $\tilde{\boldsymbol{v}}_j$. To that end, we can apply results from the well-studied field of matrix perturbation theory [36, 37]. Namely, if certain conditions hold for the D and E matrices, then there are certain bounds that can be established. We shall just give a few examples below to demonstrate what we mean.

Theorem 2. (Adapted from [38].) *Suppose the E and $\tilde{E}$ matrices are positive definite, which allows them to admit the Cholesky decomposition $E = R^\dagger R$ and $\tilde{E} = \tilde{R}^\dagger \tilde{R}$. Define $A = (R^\dagger)^{-1} D R^{-1}$ and $\tilde{A} = (\tilde{R}^\dagger)^{-1} \tilde{D} \tilde{R}^{-1}$, which allows us to write the generalised eigenvalue problems $D\boldsymbol{v} = \lambda E \boldsymbol{v}$ and $\tilde{D}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{E}\tilde{\boldsymbol{v}}$ as $A\boldsymbol{\beta}_i = \lambda_i \boldsymbol{\beta_i}$ and $\tilde{A}\tilde{\boldsymbol{\beta}}_i = \tilde{\lambda}_i \tilde{\boldsymbol{\beta}}_i$. Here, $\boldsymbol{\beta} = R\boldsymbol{v}$ and $\tilde{\boldsymbol{\beta}} = \tilde{R}\tilde{\boldsymbol{v}}$. Without loss of generality, we order the eigenvalues such that $\lambda_i \leq \lambda_j$, $\tilde{\lambda}_i \leq \tilde{\lambda}_j$ for $i < j$. Then, we have:*

$$|\lambda_i - \tilde{\lambda}_i| \leq \|A - \tilde{A}\| \tag{C5}$$

where $\|A - \tilde{A}\|$ denotes the spectral norm.

We see that the trick here is to convert the two generalised eigenvalue problems into normal hermitian eigenvalue problems, before using the well-studied bound for normal hermitian eigenvalue problems. The condition for E and $\tilde{E}$ to be positive definite might seem very strict at first glance, but there are two things that must be said here. Firstly, if the set $\mathbb{CS}_K$ is linearly independent, then the E matrix would be positive definite. If the $\tilde{E}$ matrix is small enough, we can directly check if it is positive definite too before applying this theorem. Secondly, as was mentioned in Appendix A, we can ignore the action of E and $\tilde{E}$ on their corresponding nullspaces when using the CQFF algorithm. This means that it is possible to restrict ourselves to work in the column spaces of the E and $\tilde{E}$ matrices respectively, and in that restricted space, E and $\tilde{E}$ are positive definite. If we want to relax the above condition of E and $\tilde{E}$ being positive definite, we can use an alternative bound for the eigenvalues, known as the Bauer-Fike theorem[39], which we adapt below for the convenience of the reader.

Theorem 3. (Adapted from [39].) *Let E be a Hermitian matrix. For each eigenvalue $\tilde{\lambda}_j$ of $\tilde{E}$, there is an eigenvalue λ of E such that*

$$|\tilde{\lambda}_j - \lambda| < \|\tilde{E} - E\|\,. \tag{C6}$$

where here the norm $\|E - \tilde{E}\|$ is the spectral norm.

Apart from theorem 2 and theorem 3, there are other bounds for the eigenvalues that can be found in [38].

To bound the difference in the various eigenspaces, for the case where both $\tilde{E}$ and E are positive definite, we can use the same transformation as in theorem 2 to convert the two generalised eigenvalue problems $D\boldsymbol{v} = \lambda E\boldsymbol{v}$ and $\tilde{D}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{E}\tilde{\boldsymbol{v}}$ into two regular Hermitian eigenvalue problems before using the well known Davis-Kahan $\sin(\Theta)$ theorem [40] to provide bounds on the eigenspaces corresponding to different eigenspaces of the regular Hermitian eigenvalue problems. For completeness, we will also adapt the Davis-Kahan $\sin(\Theta)$ theorem here.

Theorem 4. (Adapted from [40].) *Suppose the E and $\tilde{E}$ matrices are positive definite, which allows them to admit the Cholesky decomposition $E = R^\dagger R$ and $\tilde{E} = \tilde{R}^\dagger \tilde{R}$. Define $A = (R^\dagger)^{-1} D R^{-1}$ and $\tilde{A} = (\tilde{R}^\dagger)^{-1} \tilde{D} \tilde{R}^{-1}$, which allows us to write the generalised eigenvalue problems $D\boldsymbol{v} = \lambda E\boldsymbol{v}$ and $\tilde{D}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{E}\tilde{\boldsymbol{v}}$ as $A\boldsymbol{\beta}_i = \lambda_i \boldsymbol{\beta_i}$ and $\tilde{A}\tilde{\boldsymbol{\beta}}_i = \tilde{\lambda}_i \tilde{\boldsymbol{\beta}}_i$. Here, $\boldsymbol{\beta} = R\boldsymbol{v}$ and $\tilde{\boldsymbol{\beta}} = \tilde{R}\tilde{\boldsymbol{v}}$. Without loss of generality, let A and $\tilde{A}$ be n by n matrices. Also, treat the vectors $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$ as column matrices. Define*

$$V_0 = \left(\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_l\right) \tag{C7}$$
$$A_0 = diag\left(\lambda_1 \ldots \lambda_l\right) \tag{C8}$$
$$V_1 = \left(\boldsymbol{\beta}_{l+1} \ldots \boldsymbol{\beta}_n\right) \tag{C9}$$
$$A_1 = diag\left(\lambda_{l+1} \ldots \lambda_n\right) \tag{C10}$$

such that we have $A = V_0 A_0 V_0^\dagger + V_1 A_1 V_1^\dagger$. Also define the terms $\tilde{V}_0, \tilde{A}_0, \tilde{V}_1, \tilde{A}_1$ analogously such that we have $\tilde{A} = \tilde{V}_0 \tilde{A}_0 \tilde{V}_0^\dagger + \tilde{V}_1 \tilde{A}_1 \tilde{V}_1^\dagger$. Then, if the eigenvalues of A_0 are contained in an interval (a, b) and the eigenvalues of $\tilde{A}_1$ are excluded from the interval $(a - \delta, b + \delta)$ for some $\delta > 0$, then

$$\|V_0^\dagger \tilde{V}_1\| \leq \frac{\|V_0^\dagger (\tilde{A} - A)\tilde{V}_1\|}{\delta} \tag{C11}$$

for any unitarily invariant norm $\|\cdot\|$.

A corresponding bound for the case where either E is not positive definite or $\tilde{E}$ is not positive definite can be found in [38].

Appendix D: Tweaking CQFF to make it more similar to VFF

Define $U(\Delta t) = e^{-iH\Delta t}$. If Δt is small, we have

$$U(\Delta t) \approx \mathbb{1} - iH\Delta t \tag{D1}$$

and hence

$$U(N\Delta t) = U(\Delta t)^N \approx (\mathbb{1} - iH\Delta t)^N\,. \tag{D2}$$

Note that $U(N\Delta t)$ commutes with H, and hence starting from Eq. (7), we define $[U(N\Delta t)]_{\mathbb{CS}_K}$ through

$$[U(N\Delta t)]_{\mathbb{CS}_K} = \sum_j (1 - i\lambda_j \Delta t)^N \boldsymbol{v}_j \boldsymbol{v}_j^\dagger E \tag{D3}$$

where λ_j, $\boldsymbol{v}_j$ and E have the same meaning as they do in Eq. (7). As mentioned in the main text, this method is generally not preferable as compared with

the original CQFF algorithm, as the approximations in Eq. (D1) and Eq. (D2) introduce errors that can be avoided. Furthermore, time evolution with Eq. (D2) is not norm-preserving, $U(N\Delta t)$ as defined above is not unitary, but only approximately unitary.

Appendix E: Initial state preparation

For the runs on IBM's quantum computer, we prepared a state $|\psi\rangle$ with randomised parameters which we use for the CQFF algorithm. Using the *efficientSU2* function in IBMQ's Qiskit's circuit library, this initial state was produced by 5 layers of gates on $|00...0\rangle$, where each layer comprises of $SU(2)$ operations with randomised rotation angles on all the qubits followed by CNOT gates to entangle all the qubits.

Appendix F: Size of subspaces used in the CQFF algorithm for the different Hamiltonians considered

In Table I, we tabulate the size of the subspaces used in the CQFF algorithm for the different Hamiltonians considered.

	$K=1$	$K=2$	$K=3$
H_1 (2 qubits)	4		
H_1 (3 qubits)	7	16	
H_2 (4 qubits)	3	4	
H_2 (5 qubits)	4	7	8
H_3 (2 qubits)	3		

TABLE I. Comparison of the sizes of $\mathbb{CS}_K$ for the different Hamiltonians considered in this paper.

Appendix G: Supplementary plots

1. Expectation value plot for H_1 with two qubits

As mentioned in the main text we compute an observable O as following

$$\begin{aligned}\langle O(t)\rangle_{CQFF} &= \langle\psi(t)_{CQFF}|O|\psi(t)_{CQFF}\rangle \\ &= \sum_i^L \sum_j^L \alpha_i(t)^* \langle\chi_i|O|\chi_j\rangle \alpha_j(t) \end{aligned} \quad \text{(G1)}$$

The coefficients $\boldsymbol{\alpha}(T)$ as defined in Eq. (16) are computed using a quantum computer by measuring the D and E matrices. These crucial terms describe the evolution of the quantum state in time. To calculate explicit expectation values $\langle O(t)\rangle_{CQFF}$, we additionally need terms like $\langle\chi_i|O|\chi_j\rangle$. In the main text, these terms $\langle\chi_i|O|\chi_j\rangle$ are computed using a classical computer. Here for completeness, we plot the $\langle Z_1(t)\rangle$ for the 2 qubit case of H_1 where the terms $\langle\chi_i|Z_1|\chi_j\rangle$ are computed directly on the quantum computer by sampling the state $|\phi\rangle$ in the corresponding Pauli-rotated basis. Our results are shown in Fig. 6. The computation of $\langle\chi_i|O|\chi_j\rangle$ on the quantum computer introduces a small error in the time evolution due to the noise of the quantum computer.

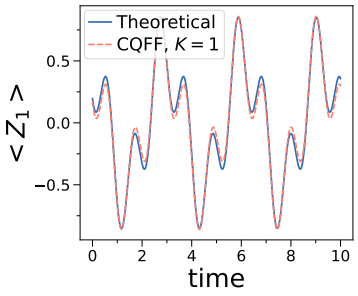

FIG. 6. Time evolution of CQFF on a 2 qubit state with Hamiltonian H_1, simulated on the IBM quantum processor *ibmq_lagos* with 8192 shots. The expectation value $\langle Z_1\rangle$ is shown here. Here, as compared to Fig. 1, terms like $\langle\chi_i|O|\chi_j\rangle$ are computed directly on the quantum computer.

2. Fidelities at long time scales

In Fig. 7, we show the dynamics of the fidelities for long time scales for the Hamiltonians considered in the main text.

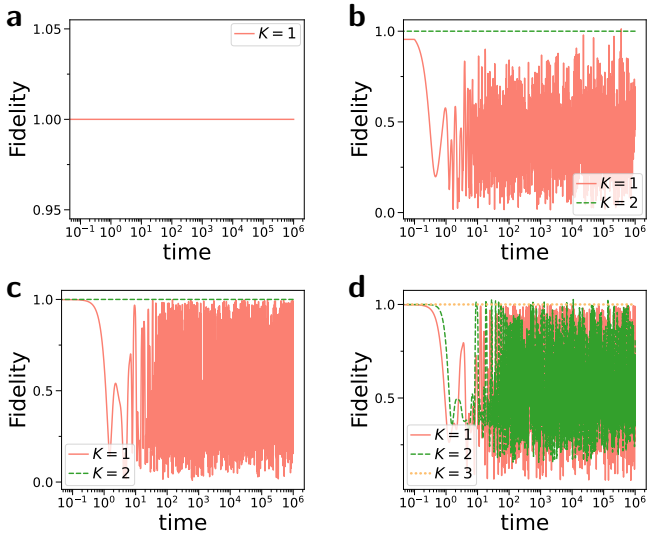

FIG. 7. The long term behaviour of the fidelities for the Hamiltonians considered in the main text. As can be seen, for sufficiently high K, the fidelity essentially remains at 1. For lower values of K, the fidelities remain oscillatory, as reflected in the log plots below.
a) Long term behavior of the fidelity for the Hamiltonian in Fig. 1. **b)** Long term behavior of the fidelity for the Hamiltonian in Fig. 2. **c)** Long term behavior of the fidelity for the Hamiltonian in Fig. 3. **d)** Long term behavior of the fidelity for the Hamiltonian in Fig. 4.

ACKNOWLEDGEMENTS

We are grateful to the National Research Foundation and the Ministry of Education, Singapore for financial support. The authors acknowledge the use of the IBM Quantum Experience devices for this work. This work is supported by a Samsung GRP project and the UK Hub in Quantum Computing and Simulation, part of the UK National Quantum Technologies Programme with funding from UKRI EPSRC grant EP/T001062/1.

DATA AVAILABILITY

The authors declare that the main data supporting the findings of this study are available within the article. Extra data sets are available upon request.

[1] R. P. Feynman, Int. J. Theor. Phys **21** (1982).
[2] S. Lloyd, Science , 1073 (1996).
[3] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer, "The trotter step size required for accurate quantum simulation of quantum chemistry," (2014), arXiv:1406.4920 [quant-ph].
[4] J. Preskill, Quantum **2**, 79 (2018).
[5] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, arXiv:2101.08448 [cond-mat, physics:quant-ph] (2021), arXiv: 2101.08448.
[6] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, arXiv:2012.09265 [quant-ph, stat] (2020), arXiv: 2012.09265.
[7] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, Nature communications **5**, 4213 (2014).
[8] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242 (2017).
[9] Y. Li and S. C. Benjamin, Physical Review X **7**, 021050 (2017).
[10] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. Benjamin, Quantum **3**, 191 (2019), arXiv: 1812.08767.
[11] K. Heya, K. M. Nakanishi, K. Mitarai, and K. Fujii, arXiv:1904.08566 [quant-ph] (2019), arXiv: 1904.08566.
[12] K. Bharti and T. Haug, "Quantum assisted simulator," (2020), arXiv:2011.06911 [quant-ph].
[13] J. W. Z. Lau, K. Bharti, T. Haug, and L. C. Kwek, "Quantum assisted simulation of time dependent hamiltonians," (2021), arXiv:2101.07677 [quant-ph].
[14] C. Cîrstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, npj Quantum Information **6**, 82 (2020).
[15] J. Gibbs, K. Gili, Z. Holmes, B. Commeau, A. Arrasmith, L. Cincio, P. J. Coles, and A. Sornborger, arXiv:2102.04313 [quant-ph, stat] (2021), arXiv: 2102.04313.
[16] T. Haug and K. Bharti, arXiv:2011.14737 [quant-ph] (2020), arXiv: 2011.14737.
[17] B. Commeau, M. Cerezo, Z. Holmes, L. Cincio, P. J. Coles, and A. Sornborger, "Variational hamiltonian diagonalization for dynamical quantum simulation," (2020), arXiv:2009.02559 [quant-ph].
[18] S. Barison, F. Vicentini, and G. Carleo, "An efficient quantum algorithm for the time evolution of parameterized circuits," (2021), arXiv:2101.04579 [quant-ph].
[19] J. W. Z. Lau, T. Haug, L. C. Kwek, and K. Bharti, "Nisq algorithm for hamiltonian simulation via truncated taylor series," (2021), arXiv:2103.05500 [quant-ph].
[20] A. M. Childs and R. Kothari, Quantum Info. Comput. **10**, 669–684 (2010).
[21] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Communications in Mathematical Physics **270**, 359 (2007).

[22] Y. Atia and D. Aharonov, Nature communications **8**, 1 (2017).

[23] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nature communications **9**, 1 (2018).

[24] H.-Y. Huang, K. Bharti, and P. Rebentrost, "Near-term quantum algorithms for linear systems of equations," (2019), arXiv:1909.07344 [quant-ph].

[25] K. Sharma, M. Cerezo, L. Cincio, and P. J. Coles, "Trainability of dissipative perceptron-based quantum neural networks," (2020), arXiv:2005.12458 [quant-ph].

[26] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, "Noise-induced barren plateaus in variational quantum algorithms," (2021), arXiv:2007.14384 [quant-ph].

[27] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost-function-dependent barren plateaus in shallow quantum neural networks," (2020), arXiv:2001.00550 [quant-ph].

[28] T. Haug, K. Bharti, and M. Kim, arXiv preprint arXiv:2102.01659 (2021).

[29] K. Bharti and T. Haug, "Iterative quantum assisted eigensolver," (2020), arXiv:2010.05638 [quant-ph].

[30] Y. Saad, SIAM Journal on Numerical Analysis **29**, 209 (1992).

[31] K. Mitarai and K. Fujii, Phys. Rev. Research **1**, 013006 (2019).

[32] Z. Jia and G. Stewart, Mathematics of computation **70**, 637 (2001).

[33] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, Phys. Rev. A **95**, 042308 (2017).

[34] K. Bharti, "Quantum assisted eigensolver," (2020), arXiv:2009.11001 [quant-ph].

[35] F. Petiziol, M. Sameti, S. Carretta, S. Wimberger, and F. Mintert, "Quantum simulation of three-body interactions in weakly driven quantum systems," (2020), arXiv:2011.03399 [quant-ph].

[36] R. Bhatia, *Perturbation bounds for matrix eigenvalues* (SIAM, 2007).

[37] G. Stewart and J. Sun, Press, Boston MA (1990).

[38] G. Stewart, Linear Algebra and its Applications **23**, 69 (1979).

[39] F. L. Bauer and C. T. Fike, Numerische Mathematik **2**, 137 (1960).

[40] C. Davis and W. M. Kahan, SIAM Journal on Numerical Analysis **7**, 1 (1970).