Quantum Information Processing (2019) 18:357
https://doi.org/10.1007/s11128-019-2469-1

# Context-aware quantum simulation of a matrix stored in quantum memory

Ammar Daskin[1] · Teng Bian[2,3,4] · Rongxin Xia[2,3,4] · Sabre Kais[2,3,4]

Received: 2 April 2019 / Accepted: 9 October 2019

## Abstract

In this paper, a storage method and a context-aware circuit simulation idea are presented for the sum of block diagonal matrices. Using the design technique for a generalized circuit for the Hamiltonian dynamics through the truncated series, we generalize the idea to (0–1) matrices and discuss the generalization for the real matrices. The presented circuit requires $O(n)$ number of quantum gates and yields the correct output with the success probability depending on the number of elements: For matrices with $poly(n)$, the success probability is $1/poly(n)$. Since the operations on the circuit are controlled by the data itself, the circuit can be considered as a context-aware computing gadget. In addition, it can be used in variational quantum eigensolver and in the simulation of molecular Hamiltonians.

**Keywords** Quantum circuits · Circuit simulation of Hamiltonian · Context-aware computing · Quantum memory

## 1 Introduction

A quantum algorithm can be described through matrix-vector transformations. The number of two- and single-qubit quantum gates required to implement these transformations as a quantum circuit describes the computational complexity of the algorithm. It is known that an $N \times N$ matrix that depends on $N^2$ independent parameters requires $O(N^2)$ quantum gates [1]. When the matrix is sparse with only $polylog(N)$ nonzero elements, then it is possible to design matrix specific quantum circuits with

✉ Ammar Daskin
adaskin25@gmail.com

[1] Department of Computer Engineering, Istanbul Medeniyet University, Üsküdar, Istanbul, Turkey

[2] Department of Chemistry, Purdue University, West Lafayette, IN, USA

[3] Department of Physics and Astronomy, Purdue University, West Lafayette, IN, USA

[4] Birck Nanotechnology Center, Purdue University, West Lafayette, IN, USA

Published online: 20 October 2019

$polylog(N)$ quantum gates. The common circuit design approach is to write the matrix $\mathcal{H}$ as a sum $\sum_j H_j$, where $e^{iH_jt}$ should be easy to compute for any $H_j$, and approximate $\exp(i\mathcal{H}t)$ by using the Trotter formula for the exponentiation, i.e., $(\prod_j \exp(iH_jt/r)^r$. This idea is used in previous works such as [2,3] to find the most efficient circuits for sparse matrices.

Solutions of many problems relate to finding the lowest (or highest) eigenvalue in magnitude and its associated eigenvector. This includes finding the ground state of the Hamiltonian of a quantum system in quantum chemistry [4–9]. Recent methods such as variational quantum eigensolver [10,11] and quantum signal processing [12] have proved that the exponential $e^{i\mathcal{H}t}$ is not needed to find the eigenpair of $\mathcal{H}$. For these methods, it is sufficient to find a direct circuit for $\mathcal{H}$ that can generate $\mathcal{H}\left|\psi\right\rangle$ for any quantum state $\left|\psi\right\rangle$ (example circuits [13,14]).

In this paper, we describe a storage method for (0–1) matrices on quantum memory and show an efficient circuit design that loads the data from quantum memory as a superpositioned state and generates the output $\mathcal{H}\left|\psi\right\rangle$ for any $\mathcal{H}$ by using quantum operations that are controlled by the data itself. When a system uses the context to provide relevant information, it may be considered as a context-aware system [15]. Therefore, we believe this work will pave the way for quantum context-aware computing. In terms of the computational complexity, the circuit uses only $O(n)$ number of quantum gates. The success probability of the method—as expected—scales with the number of elements and is $1/poly(n)$ for the matrices with $poly(n)$ number of nonzero elements.

This paper is organized as follows: We explain the approach for (0–1) block diagonal matrices in Sect. 2. In Sect. 3, we generalize the idea to matrices with (0–1) elements. The computational complexity and the success probability in the case of sparse matrices are shown in Sect. 4. In Sect. 5, we explain how the quantum random-access memory is employed and analyze the storage complexity for sparse matrices with (0–1) elements. Section 6 briefly discusses how the idea generalizes to a matrix in real space and how the circuit can be used with variational quantum eigensolver by applying the circuit to find the ground state of the hydrogen molecule.

## 2 The approach for (0–1) block diagonal matrices

Assume that we want the circuit implementation of the following $N \times N$ matrix:

$$\mathcal{H} = \sum_i Q_i \tag{1}$$

where $Q_i$ is a block diagonal matrix defined as the direct sum of quantum gates: For instance, we will assume $Q_i = \bigoplus_k^K \sigma_k$ with $\sigma_k \in \{X, Z, -Z, XZ, ZX, -I, I, Zero\}$; here $Zero$ represents a $2 \times 2$ zero matrix and $\bigoplus$ is used for the direct sum of the matrices. (It generates a block diagonal matrix.). We use a quantum state whose elements encode the gate information of the matrices on the diagonal:

$$|g_i\rangle = \frac{1}{\eta}\sum_{k=0}^{K-1} |g_{ik}\rangle\, |k\rangle\,, \tag{2}$$

where $\eta$ is the normalization constant and $|g_{ik}\rangle$ is a vector of size eight from the standard basis since there are eight different gates in the gate set. As an example, if

$$Q_i = \begin{pmatrix} Z & & \\ & X & \\ & & Zero \end{pmatrix}, \tag{3}$$

then

$$\begin{aligned} &|g_{i0}\rangle = |\mathbf{1}\rangle\,, |g_{i1}\rangle = |\mathbf{0}\rangle\,,\\ &|g_{i2}\rangle = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ and } \eta = \frac{1}{\sqrt{2}}. \end{aligned} \tag{4}$$

Here, $|\mathbf{i}\rangle$ represents the $i$th vector in the standard basis. The zero vector for $|g_{i2}\rangle$ means that it is not included in the summation given in (2). Notice that $|g_i\rangle$ involves $n+2$ number of qubits: The last $n-1$ qubits indicate the value $k$, and the first three qubits are for the gate-type $g_{ik}$.

For an arbitrary given $n$-qubit state $|\psi\rangle$ and $|g_i\rangle$, consider the operation $|\beta\rangle = Q_i\,|\psi\rangle$. The circuit in Fig. 1 can be used to produce $1/(\eta\sqrt{8})\,|\beta\rangle$ on the amplitudes of the following states (see "Appendix A" for the validation of the circuit):

$$\begin{aligned} \frac{1}{\eta\sqrt{8}}\,|000\rangle \sum_{j=0}^{N/2-1} \big(&\beta_{2j}\,|\mathbf{j}\rangle_k\,|\mathbf{j}0\rangle_{\text{system}}\\ &+\beta_{2j+1}\,|\mathbf{j}\rangle_k\,|\mathbf{j}1\rangle_{\text{system}}\big), \end{aligned} \tag{5}$$

where $|\mathbf{j}\rangle$ represents the $j$th vector in the basis. The coefficient $1/(\eta\sqrt{8})$ comes from (2) and the three Hadamard gates used in the circuit. Therefore, the overall success probability is $||\,|\beta\rangle\,/(\eta\sqrt{8})||^2$.

Now let us generalize this to $\sum_i Q_i\,|\psi\rangle$: First we add one more register to represent $|i\rangle$, and then, we give the following initial state to the circuit:

$$|g\rangle = \frac{1}{\zeta}\sum_{i,k} |i\rangle\,|g_{ik}\rangle\,|k\rangle\,|\psi\rangle\,, \tag{6}$$

where $\zeta$ is the normalization constant so that all the nonzero elements in $|i\rangle\,|g_{ik}\rangle\,|k\rangle$ are equal to $\frac{1}{\zeta}$. To get the sum, we apply Hadamard gates to the register representing $|i\rangle$. The resulting circuit is drawn in Fig. 2 (see "Appendix B" for the validation). If $|g_{ik}\rangle$s are stored in the quantum memory, the circuit takes only $O(n)$ time. This complexity does not change much if we change the size of the basis to add more gates to the set.

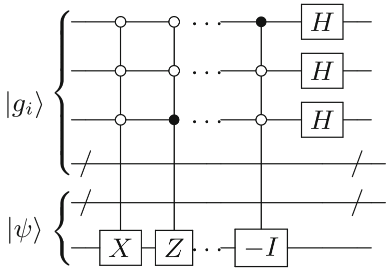

**Fig. 1** Quantum circuit for the implementation of $Q_i |\psi\rangle$. The circuit includes $2n + 2$ qubits and implements the gate set $\{X, Z, -Z, XZ, ZX, -I, I, I\}$

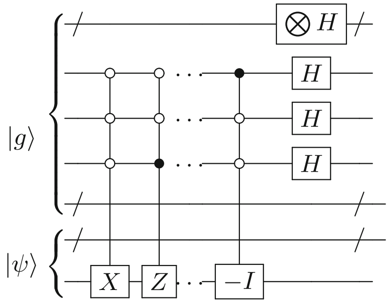

**Fig. 2** Quantum circuit for the implementation of $\sum Q_i |\psi\rangle$

In this case, the size of the first register $|i\rangle$ determines the number of Hadamard gates and the success probability. We can define the success probability in this general case as:

$$P_{\text{success}} = \left\| \frac{\sum_i Q_i |\psi\rangle}{C\zeta} \right\|^2, \tag{7}$$

where $C$ is the coefficient determined by the number of Hadamard gates on the circuit. If the number of Hadamard gates is close to $n$, then we obtain $P_{\text{success}} \approx 1/N$ which is exponentially small in the number of qubits. However, in the sparse case one can expect the number of Hadamards to be much smaller than $n$ as shown in Sect. 4.

## 3 General $\mathcal{H}$ with 0–1 elements

In [16], a method is presented to write a general matrix as a sum of unitary matrices. In this method, first without changing the location of any element, two indices $i$ and $k$ are assigned to all $2 \times 2$ submatrices inside the matrix. For instance,

$$\mathcal{H} = \begin{pmatrix} H_{00} & H_{10} & H_{20} & H_{30} \\ H_{11} & H_{01} & H_{31} & H_{21} \\ H_{22} & H_{32} & H_{02} & H_{12} \\ H_{33} & H_{23} & H_{13} & H_{03} \end{pmatrix}_{8\times 8}. \tag{8}$$

Then for $i = 0, \ldots, N/2-1$, the block diagonal matrix $H_i = \otimes_{k=0}^{N/2-1}$ is constructed. Here, $H_i$ includes one submatrix from each row and $k$ corresponds to the row index of the submatrix (a larger matrix given in (C1), notice the symmetries.). $\mathcal{H}$ is expressed as a sum of $H_i$s in the following form:

$$\mathcal{H} = \sum_{i=0}^{N/2} H_i P_i. \tag{9}$$

Here $P_i$ is a permutation matrix described by using the binary form $i = (b_0 \dots b_{n-1})_2$ as:

$$P_i = \left( \bigotimes_{j=0}^{n-1} X^{b_j} \right) \otimes I. \tag{10}$$

That means an $X$ gate is put on qubit $j$ if there is 1 in the binary representation of $i$. This results in at most $n-1$ single $X$ gates on the circuit. Here note that while $H_i$ may not be a symmetric matrix, $H_i P_i$ is one.

As in the previous section, we would like to have a circuit where based on the data elements of $H_i$, a control register can choose the set of quantum gates. We will consider sparse matrices with 0–1 elements and assume that any $H_{ik}$ is in the following form (Note that since the following matrices form a basis, it automatically generalizes to any $H_{ik}$ with 0–1 elements.):

$$\begin{aligned}
G_0 &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{I+Z}{2} \\
G_1 &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \frac{X+XZ}{2} \\
G_2 &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \frac{X+ZX}{2} \\
G_3 &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \frac{I-Z}{2} \\
Zero &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = Zero + Zero
\end{aligned} \tag{11}$$

Using the above assumption, $H_i$ can be defined as a sum of two block diagonal matrices similarly to $Q_i$ in (3): $H_i = (Q_{ia} + Q_{ib})$. Then, we rewrite $\mathcal{H}$ as:

$$\mathcal{H} = \sum_{i=0}^{N/2-1} \left( Q_{ia} P_i + Q_{ib} P_i \right). \tag{12}$$

For simplicity, we will again assume $Q_{ia}$ and $Q_{ib}$ consist only the gates from the set $\{X, Z, -Z, XZ, ZX, -I, I, Zero\}$ and use the same encoding as in (4). Applying $\mathcal{H}$ to a general quantum state $|\psi\rangle$ leads to a superpositioned state:

$$\mathcal{H} |\psi\rangle = \sum_{i=0}^{N/2-1} H_i P_i |\psi\rangle \tag{13}$$

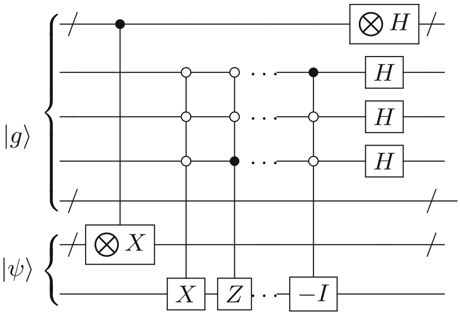

**Fig. 3** Quantum circuit implementation of (13). The circuit requires $O(n)$ quantum operations. The success probability is given in (7) for the general case and in (14) for the sparse case

We can construct this superposition state by using a similar circuit to Fig. 2, but in this case we have to control the X gates that implement $P_i$ by the first register representing $|i\rangle$. The resulting circuit is shown in Fig. 3.

## 4 Sparse $\mathcal{H}$ with $poly(n)$ number of elements

For $\mathcal{H}$ with $poly(n)$ number of 1s, the following two observations can be made:

1. The number of $H_i$s with only 0–1 elements cannot be more than $poly(n)$. Otherwise, the matrix has more than $poly(n)$ elements or has elements with values different than 0–1s. Therefore, we load only nonzero $H_i$s and adjust the control bits of the $X$ gates that implements $P_i$s. This requires only $poly(log(n))$ number of qubits for representing $|i\rangle$ and Hadamard gates.
2. $\zeta$ cannot be more than $poly(n)$. Otherwise, the matrix has more than $poly(n)$ number of nonzero elements.

Combining these two observations, we can conclude that

$$P_{\text{success}} = \left\| \frac{\sum_i Q_i \, |\psi\rangle}{poly(n)} \right\|^2. \tag{14}$$

Therefore, if $\left\| \sum_i Q_i \, |\psi\rangle \right\|$ is not too small, the circuit in Fig. 3 with $O(n)$ number of quantum operations is able to simulate any sparse matrix with $poly(n)$ number of 1s with $O(1/poly(n))$ success probability.

## 5 Storing $\mathcal{H}$ in quantum memory

A random-access memory (RAM) with $2^n$ memory cells can be addressed by using an $n$-bit input string. In the classical architectures, for a given address information in the input register, a RAM outputs the data stored in the addressed memory cell. Similarly, a quantum RAM model (qRAM) based on the bucket brigade addressing scheme (a treelike structure) is proposed by Giovanetti et al. [17,18]. For a given address information on the quantum register, qRAM returns the information on the addressed memory cell onto the second quantum register. This process requires $O(n)$ number of memory call. (That is, the number of switches activated in the tree-structured address scheme to reach the data. See [19] for the discussion on its robustness.). Therefore, we can consider its complexity as $O(n)$ per memory call. In contrast to classical RAM, the input register in qRAM can be also a superposition of the addresses. In that case, it returns the superposition of the addressed memory cells in $O(n)$ time complexity, i.e.,

$$\sum_j \alpha_j |\mathbf{j}\rangle |\mathbf{0}\rangle \xrightarrow{\text{qRAM}} \sum_j \alpha_j |\mathbf{j}\rangle \left|\mathbf{m_j}\right\rangle, \tag{15}$$

where $\sum_j \alpha_j |\mathbf{j}\rangle$ represents the superposition of the addresses and $\left|\mathbf{m_j}\right\rangle$ is the data (either classical or quantum) in the $j$th memory cell. In this paper, we assume the data are classically preprocessed in a way that matrices on the diagonal of $H_i$ are stored as a vector in qRAM. In particular, we store the vectorized form of $H_{ik}$:

$$\text{for } H_{ik} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ |g_{ik}\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle\,. \tag{16}$$

Here, the size of $|g_{ik}\rangle$ depends on the number of quantum gates that are defined in the gate set. Since in the case of sparse matrices we consider the gate set $\{G_0, G_1, G_2, G3\}$ defined in Eq. (11), the size of $|g_{ik}\rangle$ is four. It is important to note that in the physical implementation, the indices of the nonzero $H_{ik}$s and the related data $|g_{ik}\rangle$s can be stored classically; however, the address register is a quantum register and prepared by using the classically stored indices associated with the nonzero $H_{ik}$s.

In addition to not storing and indexing $H_{ik}$s without any nonzero element; $H_i$s which do not have any nonzero $H_{ik}$ are also not stored or loaded. When $\mathcal{H}$ has $poly(n)$ number of nonzero elements, most of $H_i$s must be zero and there can be at most $poly(n)$ number of nonzero $H_{ik}$s and $H_i$s (otherwise $\mathcal{H}$ must have more than $poly(n)$ number of nonzero elements). Therefore, loading the data of $\mathcal{H}$ from qRAM requires preparing a quantum input register of $log(poly(n))$ qubits into the equal superposition state of classically stored $poly(n)$ number of indices associated with the nonzero elements. This can be done in $O(n)$ time. Furthermore, a single memory call with this superpositioned-address state is enough to load the data of $\mathcal{H}$, which requires $O(n)$ number of activations as stated above. Therefore, the initial data load from qRAM requires $O(n)$ time complexity in total for the sparse matrices with $poly(n)$ number of nonzero elements.

Note that the analysis above mainly considers the sparse matrices. In the case of dense matrices with $O(2^n)$ elements, the complexity is governed by preparing an input register that has $n$ qubits, which requires $O(2^n)$ time complexity.

# 6 Discussion

## 6.1 Generalization for $\mathcal{H} \in R$

Suppose we have

$$|g_{ik}\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}, \tag{17}$$

where $\alpha$s are the real-valued elements of a $H_{ik}$. The quantum gates $G_0 \ldots G_3$s given in (11) form a computational basis. That means for $|g_{ik}\rangle$ above, it applies the superpositioned of the quantum gates with the probabilities defined by the elements of $|g_{ik}\rangle$. Therefore, after the Hadamard gates on the chosen state, the correct normalized output $\mathcal{H}\,|\psi\rangle$ can still be obtained from Fig. 3 with the probability given in (7). However, in this case, one needs to store quantum data on qRAM, which is not an easy task with the near-term quantum technology.

## 6.2 Use in variational quantum eigensolver

Variational quantum eigensolver [10,11] is generally applied to the quantum chemistry problems that are represented by the electronic Hamiltonian in the second quantization by transforming the Hamiltonian to the sum of Pauli operators, which are the products of the Pauli spin matrices(e.g., [7,20,21]). Assume the electronic Hamiltonian is $\mathcal{H} = \sum_i h_i \mathcal{H}_i$, where $\mathcal{H}_i$ is a Pauli operator and $h_i$ is the corresponding coefficient. The algorithm starts with a state $|\psi(\theta)\rangle$ defined by the vector of parameters $\theta$ and tries to optimize these parameters by minimizing the outcome $\langle\psi(\theta)|\,\mathcal{H}\,|\psi(\theta)\rangle = \sum_i \langle\psi(\theta)|\,h_i\mathcal{H}_i\,|\psi(\theta)\rangle$. Since each $\mathcal{H}_i$ is assumed to be a product of the Pauli spin matrices, they are implemented by a separate quantum module efficiently. Therefore, the algorithm involves a quantum part which computes the outcome and a classical part responsible for the optimization by computing the sum of individual outcomes and updating the parameters that forms $|\psi\rangle$. The circuit we describe here can be directly used in the quantum part of the algorithm. Since each operator $h_i\mathcal{H}_i$ can be written as $c_i H_i P_i$, where $c_i$ is $h_i$ or $-h_i$ based on the form of $\mathcal{H}_i$, $H_i$ is a block diagonal matrix and $P_i$ is the permutation matrix. In that case, we can simply get:

$$\langle\psi(\theta)|\,\mathcal{H}\,|\psi(\theta)\rangle = \sum_i \langle\psi(\theta)|\,c_i H_i P_i\,|\psi(\theta)\rangle\,. \tag{18}$$

This outcome of the circuit can be used in the classical optimization routine to update the parameters of the input state. One can also run each $\langle\psi(\theta)|\,c_i H_i P_i\,|\psi(\theta)\rangle$ on separate modules and sum the outcomes in the classical subroutine as done in the original quantum eigensolver.

As an example to show how to implement the modified VQE, we will take $h_i\mathcal{H}_i = h_i XYYZ$. Since $h_i XYYZ$ can be rewritten as $c_i H_i P_i$, in which $P_i = XXXI$, $H_i = IZZZ$ and $c_i = -h_i$. Thus $\langle\psi(\theta)|\,h_i XYYZ\,|\psi(\theta)\rangle = c_i\,\langle\psi(\theta)|\,H_i P_i\,|\psi(\theta)\rangle$. We can first apply $X$ gates to specific qubits to obtain $P_i\,|\psi(\theta)\rangle$. Then, the context-

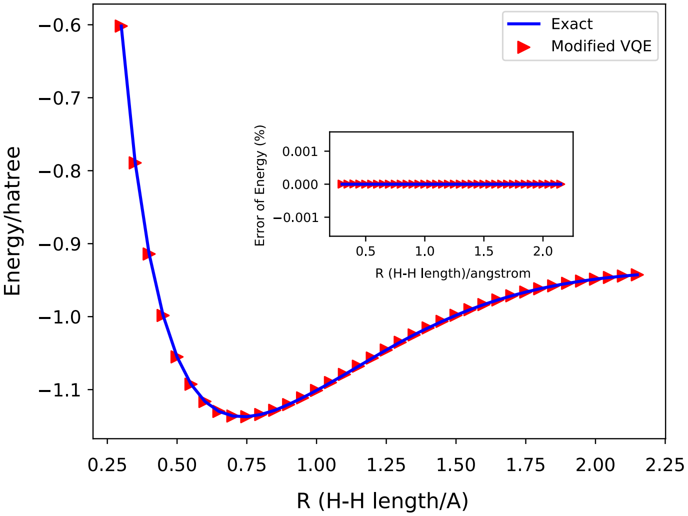

**Fig. 4** The ground state energy for the $H_2$ molecule as a function of internuclear distance R using the modified VQE based on the contex aware algorithm

aware algorithm will implement the block diagonal matrix $H_i$ and output the state $H_i P_i |\psi(\theta)\rangle$. By preparing another $|\psi(\theta)\rangle$ and doing quantum fingerprinting [22] using swap gate, we obtain $\langle\psi(\theta)| H_i P_i |\psi(\theta)\rangle$. Summing all the terms with coefficients $c_i$ gives us the energy of state $|\psi(\theta)\rangle$. Then, we can use classical optimization methods to update $\theta$ and finally get the minimum energy of the system. Figure 4 shows the ground state energy curve of $H_2$ obtained by the simulation based on this modified VQE. In the simulation, the 4-qubit Hamiltonian of $H_2$ is calculated by openfermion package [23] using STO-3G basis set, and the hardware-efficient ansatz is prepared by 3-layer pairwise design in [7].

## 7 Conclusion

In this paper, using an abstract indexing to indicate a matrix as a sum of permuted block diagonal matrices, we have described a storage method for (0–1) diagonal matrices on quantum memory and shown an efficient circuit design with $O(n)$ number of quantum gates: the circuit that loads the data from quantum memory as a superpositioned state and generates the output $\mathcal{H}|\psi\rangle$; the success probability is $1/poly(n)$ for $\mathcal{H}$ with $poly(n)$ number of nonzero elements. The quantum operations in the circuit are controlled by the data itself. Therefore, this idea can be used for context-aware quantum computing.

The circuit can be also employed with the known quantum algorithms for any matrix-related problems. We have discussed how it can be used with variational quantum eigensolver for the simulation of molecular Hamiltonians. As an example, we have used the molecule $H_2$ and computed the ground state energy curve, which is in a complete agreement with the exact diagonalization results. Since any arbitrary real matrix can be decomposed into block diagonal matrices by the described method, this method may provide a new way to evolve quantum state by unitary/non-unitary operators through quantum circuits.

**Acknowledgements** One of us, S.K, would like to acknowledge the partial support from Purdue Integrative Data Science Initiative and the U.S. Department of Energy, Office of Basic Energy Sciences, under Award Number DE-SC0019215.

## Appendix A: Validation of the circuit in Fig. 1

The controlled gate set in the circuit has the following matrix form:

$$\begin{pmatrix} I^{\otimes 2n-2}\otimes X & & & \\ & I^{\otimes 2n-2}\otimes Z & & \\ & & \ddots & \\ & & & I^{\otimes 2n-2}\otimes I \end{pmatrix} \tag{A1}$$

We can represent the whole circuit more concisely by using the direct sum of matrices:

$$\left(H^{\otimes 3}\otimes I^{\otimes 2n-1}\right)\left(\bigoplus_{j=0}^{N/2} X \oplus \bigoplus_{j=0}^{N/2} Z \oplus \bigoplus_{j=0}^{N/2} -Z \oplus \cdots \bigoplus_{j=0}^{N/2} I\right) \tag{A2}$$

To illustrate the action of the circuit, we will use the example matrix $Q_i = Z \oplus XZ$ which leads to the following $|g_{ik}\rangle\,|k\rangle$ s:

$$|g_{i0}\rangle\,|0\rangle = |\mathbf{1}\rangle\,|0\rangle \text{ and } |g_{i1}\rangle\,|1\rangle = |\mathbf{3}\rangle\,|1\rangle\,. \tag{A3}$$

Then, we form the following 6-qubit initial state:

$$|g_i\rangle\,|\psi\rangle = \frac{1}{\sqrt{2}}\left(|\mathbf{1}\rangle\,|0\rangle\,|\psi\rangle + |\mathbf{3}\rangle\,|1\rangle\,|\psi\rangle\right). \tag{A4}$$

After applying the controlled gates ($CG$) to the initial state, we obtain:

$$CG\,|g_i\rangle\,|\psi\rangle = \frac{1}{\sqrt{2}}\left(|\mathbf{1}\rangle\,|0\rangle\,(Z\otimes Z)\,|\psi\rangle + |\mathbf{3}\rangle\,|1\rangle\,(XZ\otimes XZ)\,|\psi\rangle\right). \tag{A5}$$

Applying the Hadamard gates to the first three qubits produces the following final state:

$$\begin{aligned}\frac{1}{4}\Big(&\big(|000\rangle - |001\rangle + |010\rangle - |011\rangle + |100\rangle - |101\rangle \\ &+ |110\rangle - |111\rangle\big)\,|0\rangle\,(Z\otimes Z)\,|\psi\rangle \\ &+\big(|000\rangle - |001\rangle - |010\rangle + |011\rangle + |100\rangle - |101\rangle \\ &- |110\rangle + |111\rangle\big)\,|1\rangle\,(XZ\otimes XZ)\,|\psi\rangle\Big).\end{aligned} \tag{A6}$$

Here, the states where the first three qubits are in $|000\rangle$ includes the expected output which are:

$$\frac{1}{4}\big(|000\rangle\,|0\rangle\,(Z\otimes Z)\,|\psi\rangle + |000\rangle\,|1\rangle\,(XZ\otimes XZ)\,|\psi\rangle\big). \tag{A7}$$

The equivalent of $Q_i \left|\psi\right\rangle$ is produced on the amplitudes of the states:

$$\{\left|000000\right\rangle, \left|000001\right\rangle, \left|000110\right\rangle, \left|000111\right\rangle\}. \tag{A8}$$

## Appendix B: Validations of the circuits in Figs. 2 and 3

In the circuit in Fig. 2, we have the superpositioned input state $\left|g\right\rangle$. Before the Hadamard gates on the first register of the circuit, for different $\left|i\right\rangle$ on the output we have normalized $Q_i \left|\psi\right\rangle$ on the same states as given in (A7) and (A8). That means for $\left|\mathbf{0}\right\rangle$ on the first register we have normalized $Q_0 \left|\psi\right\rangle$ on the chosen states, and for $\left|\mathbf{1}\right\rangle$ we have $Q_1 \left|\psi\right\rangle$, and so on. By applying the Hadamard gates to the first register, for $\left|i\right\rangle = \left|\mathbf{0}\right\rangle$, we generate the normalized summation $\sum_i Q_i \left|\psi\right\rangle$ on the same chosen states.

Figure 3 is just the generalization of Fig. 2 and acts the same way.

## Appendix C: A larger Hamiltonian divided into submatrices

$$\mathcal{H} = \begin{pmatrix} H_{00} & H_{10} & H_{20} & H_{30} & H_{40} & H_{50} & H_{60} & H_{70} \\ H_{11} & H_{01} & H_{31} & H_{21} & H_{51} & H_{41} & H_{71} & H_{61} \\ H_{22} & H_{32} & H_{02} & H_{12} & H_{62} & H_{72} & H_{42} & H_{52} \\ H_{33} & H_{23} & H_{13} & H_{03} & H_{73} & H_{63} & H_{53} & H_{43} \\ H_{44} & H_{54} & H_{64} & H_{70} & H_{04} & H_{14} & H_{24} & H_{34} \\ H_{55} & H_{45} & H_{75} & H_{61} & H_{15} & H_{05} & H_{35} & H_{25} \\ H_{66} & H_{76} & H_{46} & H_{52} & H_{26} & H_{36} & H_{06} & H_{16} \\ H_{77} & H_{67} & H_{57} & H_{43} & H_{37} & H_{27} & H_{17} & H_{07} \end{pmatrix}_{16\times 16}. \tag{C1}$$

## References

1. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2010)
2. Childs, A.M., Kothari, R.: Simulating sparse Hamiltonians with star decompositions. Conference on Quantum Computation. Communication, and Cryptography, pp. 94–103. Springer, Berlin (2010)
3. Berry, D.W., Childs, A.M., Cleve, R., Kothari, R., Somma, R.D.: Exponential improvement in precision for simulating sparse Hamiltonians. In: Forum of Mathematics, Sigma, vol. 5. Cambridge University Press, Cambridge (2017)
4. Kassal, I., Whitfield, J.D., Perdomo-Ortiz, A., Yung, M.-H., Aspuru-Guzik, A.: Simulating chemistry using quantum computers. Ann. Rev. Phys. Chem. **62**, 185–207 (2011)
5. Olson, J., Cao, Y., Romero, J., Johnson, P., Dallaire-Demers, P.-L., Sawaya, N., Narang, P., Kivlichan, I., Wasielewski, M., Aspuru-Guzik, A.: Quantum information and computation for chemistry. arXiv preprint arXiv:1706.05413 (2017)
6. Kais, S.: Quantum Information and Computation for Chemistry: Advances in Chemical Physics, vol. 154, p. 224109. Wiley, Hoboken (2014)
7. Bian, T., Murphy, D., Xia, R., Daskin, A., Kais, S.: Quantum computing methods for electronic states of the water molecule. Mol. Phys. **117**, 2069–2082 (2019)
8. Xia, R., Bian, T., Kais, S.: Electronic structure calculations and the ising Hamiltonian. J. Phys. Chem. B **122**, 3384–3395 (2017)

9. Xia, R., Kais, S.: Quantum machine learning for electronic structure calculations. Nat. Commun. **9**, 4195 (2018)
10. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nat. Commun. **5**, 4213 (2014)
11. McClean, J.R., Romero, J., Babbush, R., Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms. New J. Phys. **18**, 023023 (2016)
12. Low, G.H., Chuang, I.L.: Optimal Hamiltonian simulation by quantum signal processing. Phys. Rev. Lett. **118**, 010501 (2017)
13. Daskin, A., Grama, A., Kollias, G., Kais, S.: Universal programmable quantum circuit schemes to emulate an operator. J. Chem. Phys. **137**, 234112 (2012)
14. Daskin, A., Kais, S.: Direct application of the phase estimation algorithm to find the eigenvalues of the Hamiltonians. Chem. Phys. **514**, 87–94 (2018)
15. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: International Symposium on Handheld and Ubiquitous Computing, pp. 304–307. Springer (1999)
16. Daskin, A., Kais, S.: A generalized circuit for the hamiltonian dynamics through the truncated series. Quantum Inf. Process. **17**, 328 (2018)
17. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum random access memory. Phys. Rev. Lett. **100**, 160501 (2008)
18. Giovannetti, V., Lloyd, S., Maccone, L.: Architectures for a quantum random access memory. Phys. Rev. A **78**, 052310 (2008)
19. Arunachalam, S., Gheorghiu, V., Jochym-O'Connor, T., Mosca, M., Srinivasan, P.V.: On the robustness of bucket brigade quantum ram. New J. Phys. **17**, 123010 (2015)
20. O'malley, P.J.J., Babbush, R., Kivlichan, I.D., Romero, J., McClean, J.R., Barends, R., Kelly, J., Roushan, P., Tranter, A., Ding, N., et al.: Scalable quantum simulation of molecular energies. Phys. Rev. X **6**, 031007 (2016)
21. Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J.M., Gambetta, J.M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature **549**, 242 (2017)
22. Buhrman, H., Cleve, R., Watrous, J., De Wolf, R.: Quantum fingerprinting. Phys. Rev. Lett. **87**, 167902 (2001)
23. McClean, J.R., Kivlichan, I.D., Sung, K.J., Steiger, D.S., Cao, Y., Dai, C., Fried, E.S., Gidney, C., Gimby, B., Gokhale, P., et al.: Openfermion: the electronic structure package for quantum computers. arXiv preprint arXiv:1710.07629 (2017)