Contents lists available at ScienceDirect

Chemical Physics

journal homepage: www.elsevier.com/locate/chemphys

Direct application of the phase estimation algorithm to find the eigenvalues of the Hamiltonians

Ammar Daskin [a], Sabre Kais [b,*]

[a] Department of Computer Engineering, Istanbul Medeniyet University, Kadikoy, Istanbul, Turkey
[b] Department of Chemistry, Department of Physics and Birck Nanotechnology Center, Purdue University, West Lafayette, IN, USA

ARTICLE INFO

Article history:
Available online xxxx

ABSTRACT

The eigenvalue of a Hamiltonian, $\mathcal{H}$, can be estimated through the phase estimation algorithm given the matrix exponential of the Hamiltonian, $exp(-i\mathcal{H})$. The difficulty of this exponentiation impedes the applications of the phase estimation algorithm particularly when $\mathcal{H}$ is composed of non-commuting terms. In this paper, we present a method to use the Hamiltonian matrix directly in the phase estimation algorithm by using an ancilla based framework: In this framework, we also show how to find the power of the Hamiltonian matrix-which is necessary in the phase estimation algorithm-through the successive applications. This may eliminate the necessity of matrix exponential for the phase estimation algorithm and therefore provide an efficient way to estimate the eigenvalues of particular Hamiltonians. The classical and quantum algorithmic complexities of the framework are analyzed for the Hamiltonians which can be written as a sum of simple unitary matrices and shown that a Hamiltonian of order 2^n written as a sum of L number of simple terms can be used in the phase estimation algorithm with $(n+1+logL)$ number of qubits and $O(2^a nL)$ number of quantum operations, where a is the number of iterations in the phase estimation. In addition, we use the Hamiltonian of the hydrogen molecule as an example system and present the simulation results for finding its ground state energy.

1. Introduction

With the recent efforts such as digitalizing adiabatic quantum computers [1], a digital quantum simulation of the real-time dynamic of a lattice gauge theory [2], a simulation of the Hubbard model [3], and the race to build universal quantum computers among the big companies [4], quantum computers become closer to be used in unsolved real-world applications.

As Feynman suggested [5], one of the breakthroughs of quantum computers is expected to be in the simulation of quantum systems, a known difficult problem for classical computers because of the exponential growth of the system size with the number of qubits (e.g., see Ref. [6]). In the standard formalism of quantum mechanics, the Hamiltonian of a quantum system is considered as a Hermitian matrix. Since the exponential of a skew-Hermitian matrix is a unitary matrix, the time evolution, $exp(-i\hbar\mathcal{H})$, of a quantum system represented by the Hamiltonian $\mathcal{H}$ describes a unitary transformation. In general, a computation or an algorithm should be represented by a unitary matrix in order to be implemented on a quantum computer. If an algorithm can be formalized in terms of unitary matrices, then a possible implementation may yield a computational efficiency over the classical algorithms: e.g. Shor's integer factoring algorithm [7]. Many quantum algorithms use the exponential $exp(-i\hbar\mathcal{H})$ to map the description of the original problem $\mathcal{H}$ to a unitary matrix. An exact mapping through matrix exponential generally requires eigendecomposition of $\mathcal{H}$: In spite of the existence of various numerical methods, the eigendecomposition in the cases where $\mathcal{H}$ is a large-dense matrix is still a big computational challenge for classical computers. Because of this computational difficulty, it is common to use some order of Trotter-Suzuki approximation [8] by writing $\mathcal{H}$ as a sum of terms whose exponentials are known or easy to compute. In general case, the order of the approximation affects the correctness and the high order approximations dramatically increases the computational complexity [9]. Therefore, the complexity of this exponentiation should be also embodied in the consideration of the computational complexity of a quantum algorithm (see Ref. [10] for a relevant discussion of the topic).

It is known that using additional qubits eases the implementation difficulty of a quantum circuit. A quantum operator (or a dynamic of a system) can be implemented inside a larger system, where some reduced part of the system represents the action of the original system. This idea is used in different contexts such

[*] Corresponding author.
E-mail address: kais@purdue.edu (S. Kais).

https://doi.org/10.1016/j.chemphys.2018.01.002
0301-0104/© 2018 Elsevier B.V. All rights reserved.

as designing circuits from matrix elements [11,12], and generating exponential of a matrix written as a sum of unitary matrices through Taylor series [13]. Moreover, in Ref. [14], to use a matrix on a quantum computer directly, we considered converting a non-unitary matrix, $\mathcal{H}$, to a unitary by using its square root. In a more general fashion, Ref. [15] described a qubitization approach to complete a non-unitary matrix to a unitary matrix whose multiple applications generate Chebyshev polynomials of $\mathcal{H}$.

In this paper, we follow a different approach: for a Hamiltonian $\mathcal{H}$ of order 2^n with real eigenvalues; we first put the matrix in the form $\widetilde{\mathcal{H}} = (\mathbf{I} - \mathbf{i}\mathcal{H}/\kappa)$, where $\mathbf{I}$ is an identity matrix and κ is a coefficient. For a sufficiently large κ; the value of the angle, the phase, seen in the polar form of the eigenvalue of $\widetilde{\mathcal{H}}$ becomes approximately equal to λ_j/κ, where λ_j is an eigenvalue of $\mathcal{H}$. Using this fact and assuming $\mathcal{H}$ is a sum of some simple terms which can be easily mapped to quantum circuits; we present a phase estimation framework for finding λ_j/κ. In this framework, we use $\widetilde{\mathcal{H}}$ directly and show that necessary powers of $\widetilde{\mathcal{H}}$ can be estimated through successive applications after the application of the oblivious amplitude amplification. This provides a way to estimate the eigenvalues of $\mathcal{H}$ by using $(n + 1 + logL)$ qubits and $O(2^a nL)$ number of quantum operations. Here, a is the number of iterations in the phase estimation and L is the number of terms in the sum yielding $\mathcal{H}$. We also show that the powers of $\widetilde{\mathcal{H}}$ can be more accurately estimated by using additional qubits and a permutation operator in the phase estimation algorithm, which requires also the same number of quantum gates and $O(a + n + logL)$ number of qubits. As an example system, we use the Hamiltonian matrix of the hydrogen molecule and estimated its ground state energy within the presented framework.

In the following sections, after first describing the proposed framework, we will present the complexity analysis. Then, in Section 5, the simulation results for the hydrogen molecule are presented and discussed. And in the final section the paper is concluded with a summary.

2. Notes on notations

Throughout this paper, we use the following notations:

- While the Hamiltonian is represented by $\mathcal{H}$, bold capital letters are used for other matrices.
- n is used to represent the number of qubits in the system.
- $N = 2^n$ represents the size of the Hamiltonian operator (matrix) on an n-qubit system size.
- When the Hamiltonian is a sum of unitary matrices, L is used to represent the number of unitaries in the sum. L is considered to be smaller than N or at least be bounded by $O(N)$.
- $l = logL$ represents the number of qubits necessary for the ancilla system. Here, $logL$ means $log_2 L$. Therefore, the dimension of any operator acting on the ancilla is represented by L.
- a represents the number of iterations used in the phase estimation algorithm.
- k is an index used in different contexts to represent a kth step.
- M is equal to L^K represents the number of unitary matrices in the Taylor series which is truncated from the Kth term.

3. Proposed eigenvalue estimation

Phase estimation algorithm (PEA) estimates the phase ϕ for a given operator with the eigenvalue $e^{i\phi}$. For a Hamiltonian $\mathcal{H} \in C^{\otimes n}$, we propose to consider the following matrix in the phase estimation algorithm:

$$\widetilde{\mathcal{H}} = \left(\mathbf{I} - \frac{i\mathcal{H}}{\kappa}\right) \tag{1}$$

where $\mathbf{I}$ is an identity matrix, and κ is a coefficient. The eigenvalues of this matrix are in the form $\widetilde{\lambda}_j = (1 - i\lambda_j/\kappa)$, where λ_j is the jth eigenvalue of $\mathcal{H}$. When κ is sufficiently large; in the polar form of $\widetilde{\lambda}_j$, the value of the angle becomes λ_j/κ since $sin(\lambda_j/\kappa) \approx \lambda_j/\kappa$:

$$\widetilde{\lambda}_j = 1 - i\frac{\lambda_j}{\kappa} = \left|1 + i\frac{\lambda_j}{\kappa}\right| e^{i\frac{\lambda_j}{\kappa}}. \tag{2}$$

Also note that when κ is large, $\widetilde{\mathcal{H}}^\kappa$ gives an approximation to $e^{i\mathcal{H}}$. To be able to use the matrix in Eq. (1) directly in the phase estimation, we will initially assume that we have an efficient mechanism to produce the following unitary matrix:

$$\mathbf{U}^{(\mathbf{1})} = \begin{pmatrix} \widetilde{\mathcal{H}} & \bullet \\ \bullet & \bullet \end{pmatrix}, \tag{3}$$

where the first part of the matrix is equal to $\widetilde{\mathcal{H}}$ with/without some normalization, and a $\bullet$ indicates another part of the matrix. As also mentioned in the introduction, these types of matrices are used in various contexts: either to design quantum circuits from matrix elements [11,12] or to be able to use it with the oblivious amplitude amplification[14], or to estimate unitary dynamic of a Hamiltonian through truncated Taylor series [13] and its generalized form [15]. In the next section we will discuss how the above matrix can be generated by using an ancilla register. Now let us consider the input state $|\mathbf{0}\rangle|\varphi\rangle$, where $|\varphi\rangle$ represents an abstract quantum state on the system register and $|\mathbf{0}\rangle$ is the first vector in the standard basis and the state on the ancilla register. The application of $\mathbf{U}^{(\mathbf{1})}$ to this input yields the following state [13]:

$$\mathbf{U}^{(1)}|\mathbf{0}\rangle|\varphi\rangle = \sqrt{p_0}|\mathbf{0}\rangle\widetilde{\mathcal{H}}|\varphi\rangle + \sqrt{1 - p_0}|\mathbf{1}\rangle|\varphi\rangle. \tag{4}$$

The application of $\widetilde{\mathcal{H}}$ to the input $|\varphi\rangle$ on the system register can be obtained from the above output with the probability given by $p_0\|\widetilde{\mathcal{H}}\||\varphi\rangle\|$, where $||.||$ represents a vector norm. For a unitary $\widetilde{\mathcal{H}}$, the probability simply becomes p_0. The desired output can be singled out through the application of a projector $\mathbf{P} = |\mathbf{0}\rangle\langle\mathbf{0}|$ to the ancilla register.

3.1. Estimation of $\widetilde{\mathcal{H}}^{2^j}$

For the phase estimation, we need to be able to efficiently generate the set of operators $\{\mathbf{U}^{(2^0)} \ldots \mathbf{U}^{(2^a)}\}$ which consist of $\widetilde{\mathcal{H}}^{2^0} \ldots \widetilde{\mathcal{H}}^{2^a}$. This can be done through the successive applications of $\mathbf{U}^{(1)}$ after the application of the oblivious amplitude amplification described in Section 3.3: The amplitude amplification process to maximize the probability of $|\mathbf{0}\rangle$ in the ancilla turns $\mathbf{U}^{(\mathbf{1})}$ into the following form:

$$\widetilde{\mathbf{U}}^{(1)} \approx \begin{pmatrix} \widetilde{\mathcal{H}} & 0 \\ 0 & \bullet \end{pmatrix}, \tag{5}$$

k number of applications of the matrix $\widetilde{\mathbf{U}}^{(1)}$ can be used to obtain kth power of $\widetilde{\mathcal{H}}$:

$$\underbrace{\widetilde{\mathbf{U}}^{(1)} \ldots \widetilde{\mathbf{U}}^{(1)}}_{k \text{ times}} \approx \begin{pmatrix} \widetilde{\mathcal{H}}^k & 0 \\ 0 & \bullet \end{pmatrix}. \tag{6}$$

A more accurate estimation for the kth power of $\widetilde{\mathcal{H}}$ in $\mathbf{U}^{(k)}$ can be obtained from $\mathbf{U}^{(k/2)}$ by using an additional qubit in the ancilla and a permutation matrix: For instance, if $\widetilde{\mathcal{H}}^2$ is desired, we first add one more qubit to the ancilla. Then the matrix representation of the circuit becomes

Please cite this article in press as: A. Daskin, S. Kais, Chem. Phys. (2018), https://doi.org/10.1016/j.chemphys.2018.01.002

$$\mathbf{I}^{\otimes\mathbf{1}} \otimes \mathbf{U}^{(1)} = \begin{pmatrix} \widetilde{\mathcal{H}} & \bullet & \mathbf{0} & \mathbf{0} \\ \bullet & \bullet & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathcal{H}} & \bullet \\ \mathbf{0} & \mathbf{0} & \bullet & \bullet \end{pmatrix}. \tag{7}$$

Using a permutation matrix, $\mathbf{\Pi}$, similar to the one below, we can obtain the square from a successive application of this matrix as shown below:

$$\left(\mathbf{I}^{\otimes\mathbf{1}} \otimes \mathbf{U}^{(1)}\right) \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \left(\mathbf{I}^{\otimes\mathbf{1}} \otimes \mathbf{U}^{(1)}\right) = \begin{pmatrix} \widetilde{\mathcal{H}}^2 & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}, \tag{8}$$

where the dimension of the sub-matrices are assumed to conform to the multiplications. In more general sense, $\mathbf{U}^{(\mathbf{k})} = (\mathbf{I}^{\otimes logk} \otimes \mathbf{U}^{(\mathbf{k/2})})\mathbf{\Pi}(\mathbf{I}^{\otimes logk} \otimes \mathbf{U}^{(\mathbf{k/2})})$. Here, $\mathbf{\Pi}$ can be implemented through some swap operations on the ancilla register.

Consequently, we can use the set of operators $\{\mathbf{U}^{(2^0)} \dots \mathbf{U}^{(2^a)}\}$ generated by successive applications in the phase estimation algorithm and estimate the value of $\frac{\lambda_j}{\kappa}$ and hence the eigenvalue of the Hamiltonian with an accuracy affected by a. Because of the additional qubits, it may be easier to implement this method through the iterative version of the phase estimation algorithm as depicted in Fig. 1.

3.2. Steps inside PEA

The phase estimation algorithm at the $x = (a - k - 1)$th iteration is composed of three registers: vis., the phase with one-qubit, the ancilla with $x + logL$ qubits, and the system register with n qubits. The initial input to the algorithm is given by the following:

$$|\psi_0\rangle = |0\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle, \tag{9}$$

where $\left|\boldsymbol{\varphi}_j\right\rangle$ is an estimation to the jth eigenvector of $\mathcal{H}$. In PEA, we first apply the Hadamard gate to the phase qubit which simply puts this qubit into the superposition and yields the state:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle + |1\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle\right). \tag{10}$$

Then, $\mathbf{U}^{(\mathbf{2^k})}$ controlled by the phase qubit is applied to the remaining qubits:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle + |1\rangle\mathbf{U}^{(\mathbf{2^k})}|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle\right). \tag{11}$$

With the help of Eq. (4), $|\psi_2\rangle$ can be rewritten in the following form:

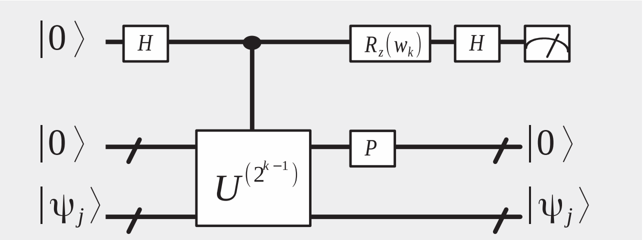

Fig. 1. The kth iteration of the phase estimation algorithm: In the circuit, $|\psi_j\rangle$ is an approximate eigenvector of $\mathcal{H}$ and $w_k = -2\pi(0.0\phi_k\phi_{k-1}\dots\phi_a)$, where $\phi_k, \phi_{k-1}, \dots, \phi_a$ represents the previously measured bit values.

$$\begin{aligned}|\psi_2\rangle = &\frac{1}{\sqrt{2}}\left(|0\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle\right) \\ &+ \frac{|1\rangle}{\sqrt{2}}\left(|\mathbf{0}\rangle\hat{p}e^{i2\pi\phi_j 2^k}\left|\boldsymbol{\varphi}_j\right\rangle + \sqrt{1-\hat{p}^2}|\Phi\rangle\right),\end{aligned} \tag{12}$$

where $\hat{p} = \sqrt{p_0}|\widetilde{\lambda}_j|^{2^k}$. The application of the operator $\mathbf{P}$ to the ancilla register causes the state to collapse into the following unnormalized state:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle + \hat{p}|1\rangle|\mathbf{0}\rangle e^{i2\pi\phi_j 2^k}\left|\boldsymbol{\varphi}_j\right\rangle\right). \tag{13}$$

In the iterative phase estimation, in each iteration we get the estimate kth bit value of the phase. This is achieved with the help of a R_z gate whose rotation angle is determined from the previous iterations (please refer to Fig. 1 for the value of the angle.). This gate converts the term $e^{i2\pi\phi 2^k}$ into the form $e^{i2\pi(0.x_k)}$, where x_k represents a binary bit: i.e., simply:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + \hat{p}|1\rangle e^{i2\pi(0.x_k)}\right)|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle, \tag{14}$$

In the final step, the Hadamard gate is applied again to the phase qubit:

$$\begin{aligned}\left|\psi_{final}\right\rangle = &\frac{1}{2}\left(1 + \hat{p}e^{i2\pi(0.x_k)}\right)|0\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle \\ &+ \frac{1}{2}\left(1 - \hat{p}e^{i2\pi(0.x_k)}\right)|1\rangle|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle\end{aligned} \tag{15}$$

Measurement on the phase qubit in this unnormalized state yields the bit value of x_k:

- When $x_k = 0$, the probability of $|0\rangle$ becomes greater in the final state:
$$\frac{1}{2}((1+\hat{p})|0\rangle + (1-\hat{p})|1\rangle)|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle \tag{16}$$
- In the case $x_k = 1$, the probability of $|1\rangle$ becomes greater:
$$\frac{1}{2}((1-\hat{p})|0\rangle + (1+\hat{p})|1\rangle)|\mathbf{0}\rangle\left|\boldsymbol{\varphi}_j\right\rangle \tag{17}$$
Here the probability difference between measuring 1 and 0 is determined by $\frac{2\hat{p}}{\mu}$, where μ is a normalization constant. Before the application of the projector $\mathbf{P}, \mu$ can also be eliminated by applying the oblivious amplitude amplification to $\mathbf{U}^{(1)}$ as described below.

3.3. Application of the oblivious amplitude amplification

The amplitude amplification [16–18] is based on the Grover's search algorithm [19] where one applies a sequence of the operators to increase the magnitude of the amplitudes of some desired states: Consider the following output state:

$$\mathbf{A}|\mathbf{0}\rangle = \sum_{x \in X_{good}} \alpha_x|\mathbf{x}\rangle|\mathbf{\Phi}\rangle + \sum_{x \in X_{bad}} \alpha_x|\mathbf{x}\rangle|\mathbf{\Phi}\rangle, \tag{18}$$

where $\mathbf{A}$ is a quantum algorithm, X_{good} and X_{bad} are the sets of good (desired) states and bad (undesired) states of the first register, x represents a standard basis vector, and $|\mathbf{\Phi}\rangle$ represents the states of the qubits in the second register. The probability of the good states in this output can be increased by the application of the iteration operator:

$$\mathbf{Q} = \mathbf{A}\mathbf{U}_\mathbf{0}^\perp\mathbf{A}^\dagger\mathbf{U}_\mathbf{f}. \tag{19}$$

Here the operator $\mathbf{U_f}$ marks (multiply by -1) the amplitudes of the desired states and $\mathbf{U}_\mathbf{0}^\perp = 2|\mathbf{0}\rangle\langle\mathbf{0}| - \mathbf{I}$.

A version of the amplitude amplification called oblivious amplitude amplification [13,20,21] can be used to increase the probability of the part $\left(\sqrt{p_0}|\mathbf{0}\rangle\widetilde{\mathcal{H}}|\varphi\rangle\right)$ in Eq. (3). In the oblivious amplitude amplification algorithm, $\mathbf{U_f}$ and $\mathbf{U}_\mathbf{0}^\perp$ only operates on the ancilla and the probability change does not affect the state in the second register. In our case, we use the following iteration operator:

$$\mathbf{Q} = \mathbf{U}^{(1)}\left(\mathbf{U}_\mathbf{0}^\perp \otimes \mathbf{I}^{\otimes n}\right)\mathbf{U}^{(1)^\dagger}\left(\mathbf{U}_\mathbf{0}^\perp \otimes \mathbf{I}^{\otimes n}\right), \qquad (20)$$

where $\mathbf{U}_\mathbf{0}^\perp$ acts on the ancilla register. The main difference from Eq. (19) is that Eq. (20) does not depend on the input to the system register and the marking and the amplifying operators, $\mathbf{U}_\mathbf{0}^\perp$, are applied only on the ancilla register.

The oblivious amplitude amplification algorithm best works when $\widetilde{\mathcal{H}}$ is a unitary process. However, recently it is shown that the algorithm also works when $\widetilde{\mathcal{H}}$ is close to a unitary matrix [13]. In this case, the error in the output of one step of the oblivious amplification is bounded by the distance which in our case measured as: $\|\widetilde{\mathcal{H}}\widetilde{\mathcal{H}}^\dagger - \mathbf{I}\|$. Since $\widetilde{\mathcal{H}}\widetilde{\mathcal{H}}^\dagger = \mathbf{I} + \mathcal{H}^2/\kappa^2$, this distance becomes:

$$\|\widetilde{\mathcal{H}}\widetilde{\mathcal{H}}^\dagger - \mathbf{I}\| = \left\|\widetilde{\mathcal{H}}^\mathbf{2}/\kappa^\mathbf{2}\right\|. \qquad (21)$$

Thus, the error in the output of the oblivious amplitude amplification is bounded by $O(\|\widetilde{\mathcal{H}}^2/\kappa^2\|)$. For a large κ, $\|\mathcal{H}^2/\kappa^2\|$ and so the order of the error can be expected to be very small. In addition, since $\widetilde{\mathcal{H}}^\kappa$ gives an approximation to $e^{i\mathcal{H}}$, in the iterations of PEA, $\widetilde{\mathcal{H}}^{2^j}$ for some j can be expected to be nearly a unitary matrix. Therefore, in the case of $\widetilde{\mathcal{H}}$ with a large κ, it is possible to use oblivious amplitude amplification. To observe this, the simulation results in Section 5 presented with and without the oblivious amplitude amplification.

3.4. Simulating sums of unitary matrices

Any Hamiltonian can be decomposed into a linear combination of unitary matrices:

$$\mathcal{H} = \sum_{l=1}^{L}\alpha_l\mathbf{H_l}, \qquad (22)$$

where α_l is some complex coefficient, $\mathbf{H}$ $_{bfl}$ represents a unitary matrix and we assume $L \leqslant N$.

Berry et al. [13] have showed that the time evolution of $\mathcal{H}$ can be simulated as follows: The time is considered to be divided into r-segments so that the time evolution at segment r, $\mathbf{U_r} = e^{-i\mathcal{H}t/r}$, is approximated through the following Taylor expansion:

$$\mathbf{U_r} \approx \sum_{k=0}^{K}\frac{1}{k!}\left(\frac{-i\mathcal{H}t}{r}\right)^k. \qquad (23)$$

Since $\mathcal{H}^k = \sum_{l1,\dots,l_k=1}^{L}\alpha_{l_1}\dots\alpha_{l_k}\mathbf{H}_{\mathbf{l_1}}\dots\mathbf{H}_{\mathbf{l_k}}$, this expansion is rewritten as:

$$\mathbf{U_r} \approx \sum_{k=0}^{K}\frac{(-it/r)^k}{k!}\sum_{l1,\dots,l_k=1}^{L}\alpha_{l_1}\mathbf{H}_{\mathbf{l_1}}\dots\alpha_{l_k}\mathbf{H}_{\mathbf{l_k}} = \sum_{j=0}^{M}\beta_j\mathbf{V_j}, \qquad (24)$$

where $\mathbf{V_j}$ is in the form of $(-i)^k\mathbf{H}_{\mathbf{l_1}}\dots\mathbf{H}_{\mathbf{l_k}}$, β_j is a complex coefficient, and M is the number of terms in the summation and equal to L^K. Assuming a mechanism to implement each $\mathbf{V_j}$ exist, then $\mathbf{V_j}$s are combined in a quantum circuit $select(\mathbf{V})$ by using $m = logM$ ancilla qubits in a way that for each state $|\mathbf{j}\rangle$ on the ancilla, a $\mathbf{V_j}$ is applied to the main qubits. Here, $|\mathbf{j}\rangle$ represents the jth vector in the standard basis. For an arbitrary state $|\psi\rangle$, this can be shown as:

$$select(\mathbf{V})|j\rangle|\psi\rangle = |j\rangle\mathbf{V_j}|\psi\rangle. \qquad (25)$$

The state on the ancilla can be put into a superposition state where each basis biased by the coefficients β_js:

$$\mathbf{B}|\mathbf{0}\rangle = \frac{1}{\sqrt{p_0}}\sum_{j=0}^{M}\sqrt{\beta_j}|j\rangle; \qquad (26)$$

where $p_0 = \sum_{j=0}^{M}\beta_j$ and β_js are assumed to be positive. The time evolution $\mathbf{U_r}$ can be obtained by combining the operator $select(\mathbf{V})$ with $\mathbf{B}$ in the following way:

$$\mathbf{U} = (\mathbf{B}^\dagger \otimes \mathbf{I})select(\mathbf{V})(\mathbf{B} \otimes \mathbf{I}) = \begin{pmatrix}\mathbf{U_r} & \bullet \\ \bullet & \bullet\end{pmatrix}, \qquad (27)$$

As similarly explained in the previous sections, this matrix can be used to simulate action of $\mathbf{U_r}$ on some arbitrary state $|\psi\rangle$ with the probability determined by p_0.

When $\mathcal{H}$ given in the form of Eq. (22), the matrix described in Eq. (1) can also be represented as a sum of unitary matrices:

$$\widetilde{\mathcal{H}} = \left(\mathbf{I} - \frac{i\mathcal{H}}{\kappa}\right) = \frac{1}{\kappa}\sum_{l=0}^{L} - i\alpha_l\mathbf{H_l} = \sum_{l=0}^{L}\beta_l\mathbf{V_l}, \qquad (28)$$

where $\mathbf{H_0} = i\kappa\mathbf{I}$. Using the same methodology given in Eq. (27), we can form the matrix $\mathbf{U}^{(1)}$ in Eq. (3) easily.

Also note that since $\widetilde{\mathcal{H}}$ is a sum of unitary matrices, its powers are also similar to those of $\mathcal{H}$. For instance, its kth power reads as follows:

$$\widetilde{\mathcal{H}}^k = \frac{1}{\kappa^k}\sum_{l1,\dots,l_k=0}^{L}(-i)^k\alpha_{l1}\mathbf{H}_{\mathbf{l_1}}\dots\alpha_{lk}\mathbf{H}_{\mathbf{l_k}}. \qquad (29)$$

Therefore,

$$\mathbf{U}^{(k)} = \begin{pmatrix}\widetilde{\mathcal{H}}^k & \bullet \\ \bullet & \bullet\end{pmatrix}. \qquad (30)$$

However, in terms of complexity, the required number of qubits is $klogL$ which grows linearly with the power. In the case of using this matrix inside PEA, the required number of qubits grows exponentially with the precision: That means if we need $\mathbf{U}^{2^j}$ at an iteration of PEA, we need to use $2^j logL$ qubits to represent L^{2^j} number of terms in the matrix $select(\mathbf{V})$. Therefore, when the available number of qubits is limited, the power should be taken through successive applications of Eq. (5) or Eq. (8). In the case of Eq. (8), only one additional qubit is necessary to find the square of $\widetilde{\mathcal{H}}$ from the previous iteration of the phase estimation algorithm.

3.5. Simulating sums of rank one matrices

Any matrix can also be written as a sum of rank one matrices. Consider the matrices in the following form:

$$\mathcal{H} = \sum_j\mathbf{H_j} = \sum_j|\mathbf{x_j}\rangle\langle\mathbf{x_j}|, \qquad (31)$$

where $\mathbf{H_j} = |\mathbf{x_j}\rangle\langle\mathbf{x_j}|$ is a rank one Hermitian matrix constructed by the normalized vector $|\mathbf{x_j}\rangle$. Such sums are frequently seen in the form $\mathbf{XX^T}$ in machine learning problems and statistical analysis of data points-where a column of the matrix $\mathbf{X}$ represents a data point. We can rewrite Eq. (31) as follows:

$$\mathcal{H} = -\frac{1}{2}\sum_{j=1}^{L}[(\mathbf{I} - 2|\mathbf{x_j}\rangle\langle\mathbf{x_j}|) - \mathbf{I}] = -\left(\sum_{j=1}^{L}\frac{\mathbf{R_j}}{2}\right) + \frac{L}{2}\mathbf{I} = \sum_{j=0}^{L}\alpha_j\mathbf{R_j}, \qquad (32)$$

Please cite this article in press as: A. Daskin, S. Kais, Chem. Phys. (2018), https://doi.org/10.1016/j.chemphys.2018.01.002

where α_js are coefficients and $\mathbf{R_0} = \mathbf{I}$. The above equation describes $\mathcal{H}$ in terms of a sum of $(L+1)$ number of idempotent-unitary matrices: i.e., $\mathbf{R_j}^2 = \mathbf{I}$. As a result, $\mathcal{H}$ can be simulated in the phase estimation after being turned into the form of Eq.(1). In the next section, we will analyze the complexity of the method in the case the Hamiltonian is given as a sum of unitary matrices.

4. Computational complexity

There are two aspects of the computational complexity of a quantum algorithm: vis., classical and quantum complexities. The classical complexity covers the preprocessing time of circuits implementing the quantum algorithm, i.e. $e^{i\mathcal{H}}$, and post-processing of the output for obtaining a desired solution. In this paper, we have mainly focused on avoiding the necessity of $e^{i\mathcal{H}}$ in the estimation of the eigenvalue of $\mathcal{H}$ on quantum computers. When the matrix $\mathcal{H}$ can be mapped to quantum circuits easily (e.g., it is a sum of "simple" unitary matrices); then the classical complexity is bounded by the number of matrix elements: i.e., $O(N^2)$, assuming all the matrix elements are needed to be processed and stored.

On the other hand, quantum complexity of a quantum algorithm is determined by the number of qubits and the number of one-and two-qubit quantum gates involved in the circuit implementing the quantum algorithm (Here, we only consider algorithmic complexity.). In this perspective, the complexity analysis of the method is given below.

4.1. Number of qubits

When the powers of $\widetilde{\mathcal{H}}$ are obtained through Eq. (5), the required number of qubits is $(n + 1 + logL)$. However, when obtained through Eq. (8), it becomes $(a + n + logL)$, where a is the number of iterations in the phase estimation.

Here, note that with the current quantum computer technology, the physical implementation of many entangling qubits is a difficult task. Although 50-qubit quantum computers are being currently built by the companies such as IBM [22] and Google [4], using extra $(a + logL)$ or $(1 + logL)$ qubits will increase the simulation burden and hinder the applicability of the algorithm on the current quantum computers.

4.2. Circuit Implementation of B

For a normalized L-dimensional column vector $|\mathbf{x_j}\rangle \in C^{\otimes l}$ and the identity matrix $\mathbf{I}, \mathbf{R_j} = \mathbf{I} - 2|\mathbf{x_j}\rangle\langle\mathbf{x_j}|$ is an Householder transformation describing a reflection operator around the vector $|\mathbf{x_j}\rangle$. On quantum computers, a Householder transformation [23] can be implemented by using $O(2^l)$ total number of two- and one-qubit quantum gates [24–26] (In Ref.[24], an implementation with the same complexity is also presented for a general version of the Householder transformation: i.e., $\mathbf{I} - (e^{i\varphi} - 1)|\mathbf{x_j}\rangle\langle\mathbf{x_j}|$).

Note that a Householder matrix, $\mathbf{R}$, can also be implemented through $(2L-3)$ number of plane (Givens) rotations[23] as follows: It is known that a Givens rotation $\mathbf{G_{j-1}}$ can be used to zero out jth entry in the column of a matrix. The Givens rotation $\mathbf{G_{L-1}}$ applied to $\mathbf{R}$ not only nullifies the last element in the first column of $\mathbf{R}$ but all the entries on the Lth row up to the first sub-diagonal entry as well. Furthermore, the matrix product $(\mathbf{G_{L-1}}\mathbf{R}\mathbf{G_{L-1}^T})$ yields a matrix where all entries but the diagonal entry on the last row and column are zeros and the diagonal element is one. Because of this property, $\mathbf{R}$ can be shown as a product of $(2L-3)$ number of plane rotations: That is, $\mathbf{R} = \mathbf{G_{L-1}^T} \ldots \mathbf{G3^T G_2^T G_1^T G_2 G_3} \ldots \mathbf{G_{L-1}}$. As a result, the circuit for $\mathbf{R}$ can be formed via plane rotations in this decomposition. Since a rotation matrix can be implemented through a multi controlled quantum gate [27]; when considered together, the rotations in the product form a uniformly controlled gate-network. These networks are well-studied in Ref. [28] where it is shown that a uniformly controlled network acting on l qubits can be simplified into $O(2^l)$ number of one- and two-qubit gates. Also note that the sparsity of the vector $|\mathbf{x_j}\rangle$ directly affects the number of plane rotations in the decomposition and hence the number of gates required for the implementation.

4.3. Circuit implementation of *select*(V)

The circuit for $select(\mathbf{V})$ is determined by the number of terms which is given by L and the number of gates required to implement each term. If each term in the sum can be implemented through simple $O(n)$ quantum gates on different qubits, then the combination of the all terms form $O(n)$ different gray-coded networks controlled by l number of qubits in the ancilla. The decomposition of this networks will require $O(nL)$ quantum gates.

Therefore, the total complexity to implement $\mathbf{U^{(1)}} = (\mathbf{B}^\dagger \otimes \mathbf{I})select(\mathbf{V})(\mathbf{B} \otimes \mathbf{I})$ is bounded by $O(nL + 2L) = O(nL)$ with the assumption that each term in the sum can be implemented with $O(n)$ quantum gates. In an iteration of the phase estimation, finding the power $\mathbf{U^{(2^k)}}$ requires 2^k number of successive applications of $\mathbf{U^{(1)}}$. Therefore, the total required quantum gates for an iteration of the algorithm becomes $O(2^k nL)$, where $1 \leqslant k \leqslant a$ with a being the total number of iterations.

5. Example application to the Hamiltonian of H_2

The concept of the second quantization is used in quantum chemistry to simplify the formalism of fermionic many particle systems. It represents the interacting systems of electrons and nuclei through the creation and annihilation operators. The molecular electronic Hamiltonian in electronic structure problem is expressed in the second quantization form as follows [29–31]:

$$\mathcal{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_s a_r, \tag{33}$$

where h_{pq} is the one-electron integrals including the electronic kinetic energy and the electron nuclear attraction terms. h_{pqrs} represents the set of two-electron integrals with the electron-electron interactions. a_j and $a_j^\dagger$ are the lowering and raising operators. This type of Hamiltonians can be represented in terms Pauli matrices by using the following Jordan-Wigner transforms:

$$a_j \to \sigma_-^j \left(\prod_{k=1}^{j-1} \sigma_z^k \right), \quad \text{and} \quad a_j^\dagger \to \sigma_+^j \left(\prod_{k=1}^{j-1} \sigma_z^k \right), \tag{34}$$

where

$$\sigma_+ = \frac{\sigma_x - i\sigma_y}{2}, \quad \text{and} \quad \sigma_z = \frac{\sigma_x + i\sigma_y}{2}. \tag{35}$$

Whitfield et al. [29,30] and Seeley et al. [31] thoroughly studied this mapping and used the Hamiltonian for the hydrogen molecule as an example system: Using a minimal number of basis, only four spin orbitals indexed from 0 to 3 are involved in the above sum. They found the values of the one- and two-electron integrals by using a restricted Hartree-Fock calculation at an internuclear separation of 7.414×10^{-11} m. Because of the overlap in the integral values, the Hamiltonian is reduced to the following form:

Please cite this article in press as: A. Daskin, S. Kais, Chem. Phys. (2018), https://doi.org/10.1016/j.chemphys.2018.01.002

$$\begin{aligned}\mathcal{H} = {} & h_{00}a_0^\dagger a_0 + h_{11}a_1^\dagger a_1 + h_{22}a_2^\dagger a_2 + h_{33}a_3^\dagger a_3, + h_{0110}a_0^\dagger a_1^\dagger a_1 a_0 \\ & + h_{2332}a_2^\dagger a_3^\dagger a_3 a_2 + h_{0330}a_0^\dagger a_3^\dagger a_3 a_0 + h_{1221}a_1^\dagger a_2^\dagger a_2 a_1 \\ & + (h_{0220} - h_{0202})a_0^\dagger a_2^\dagger a_2 a_0 + (h_{1331} - h_{1313})a_1^\dagger a_3^\dagger a_3 a_1 \\ & + (h_{0132})(a_0^\dagger a_1^\dagger a_3 a_2 + a_2^\dagger a_3^\dagger a_1 a_0) + (h_{0312})(a_0^\dagger a_3^\dagger a_1 a_2 \\ & + a_2^\dagger a_1^\dagger a_3 a_0). \end{aligned} \tag{36}$$

Using the mappings in Eq. (34), the Hamiltonian is rewritten into the terms of Pauli matrices with the values of the coefficients given in Table 1 (taken from Eq. (80) of Ref. [31]):

$$\begin{aligned}\mathcal{H} = {} & \beta_1 I + \beta_2\sigma_z^0 + \beta_3\sigma_z^1 + \beta_4\sigma_z^2 + \beta_5\sigma_z^3 + \beta_6\sigma_z^1\sigma_z^0 + \beta_7\sigma_z^2\sigma_z^0 \\ & + \beta_8\sigma_z^2\sigma_z^1 + \beta_9\sigma_z^3\sigma_z^0 + \beta_{10}\sigma_z^3\sigma_z^1 + \beta_{11}\sigma_z^3\sigma_z^2 \\ & + \beta_{12}\sigma_x^3\sigma_x^2\sigma_y^1\sigma_y^0 + \beta_{13}\sigma_x^3\sigma_y^2\sigma_y^1\sigma_x^0 + \beta_{14}\sigma_y^3\sigma_x^2\sigma_x^1\sigma_y^0 \\ & + \beta_{15}\sigma_y^3\sigma_y^2\sigma_x^1\sigma_x^0 \end{aligned} \tag{37}$$

The simulation of this Hamiltonian within the phase estimation algorithm is done after generating a circuit equvailent of $e^{i\mathcal{H}t}$ through the Trotter-Suzuki approximation (In the simulations generally the ground state of the Hamiltonian is estimated. We will also estimate the ground state. However, the excited states can be also obtained within the same framework.). As mentioned before, the complexity of the generated circuit is determined by the order of the approximation which affects the accuracy of the obtained ground state energy of the Hamiltonian from the phase estimation algorithm.

In our case, we first convert the Hamiltonian into the form of Eq. (1) by using $\kappa = 10 \times ||\mathcal{H}||_1$, which guaranties that $|\lambda_j/\kappa| \leqslant 0.1 \approx sin(0.1)$. This results in total 16 terms for *select*($\mathbf{V}$), where κ is taken as the coefficient for the additional identity and the negative signs and i are shifted to the terms inside *select*($\mathbf{V}$) so that all β_js become positive real numbers (This is because $\sqrt{\beta_j}$s are involved in the construction of $\mathbf{B}$).

Table 1
The values of the coefficients in Eq. (37).

Coefficient	Value	Coefficient	Value
β_1	−0.8126	β_9	0.1659
β_2	0.1712	β_{10}	0.1205
β_3	0.1712	β_{11}	0.1743
β_4	−0.2228	β_{12}	-0.0453
β_5	−0.2228	β_{13}	0.0453
β_6	0.1686	β_{14}	0.0453
β_7	0.1205	β_{15}	−0.0453
β_8	0.1659	κ	20.117

In the circuit, an ancilla of 4 qubits is required to control each term separately and construct *select*($\mathbf{V}$). As also mentioned in the previous section, in Ref. [32], a circuit decomposition technique is described for circuits (called gray-coded networks) where each quantum gate on a qubit is controlled by 2^l distinct states of some other l qubits. It is shown that the network can be decomposed into 2^l number of *CNOT* and 2^l number of single gates by using some form of the Hadamard transformation. The circuit for *select*($\mathbf{V}$) consists of 4 such networks (one network for per qubit in the system) each controlled by *logL* qubits in the ancilla. Since each decomposed network will require L number of *CNOTs* and L number of single quantum gates, the circuit complexity of *select*($\mathbf{V}$) is bounded by $O(nL)$, where n is the number of qubits in the system. As a result, the circuit for H_2 will require 64 *CNOT* gates. Combining this with the circuit of $\mathbf{B}$ given in the previous subsection makes the whole circuit complexity $\approx$ 100 gates. Note that in the phase estimation, since we have also a phase qubit added to the control, the required number of gates for *select*($\mathbf{V}$) will be $\approx$ 128 gates.

The simulation is done by running the iterative phase estimation in three different ways:

- In the first case, the powers are estimated through the successive application of $\widetilde{\mathbf{U}}^{(1)}$ described in Eq. (5). In addition, the amplitude amplification operator $\mathbf{Q}$ given in Eq. (20) is applied six times to obtain a maximum probability. The phase estimation algorithm is then run for 25 iterations. The resulting unnormalized probabilities for the phase qubit are shown in Fig. 2. A possible measurement on the phase qubit yields bit values that produces 0.014603 as the value of the phase. This gives $e^{-i2\pi 0.014603} = (0.995794 - 0.091624i)$ as an eigenvalue. Converting this into $(1 + i\lambda/\kappa)$ form, we compute the value of λ: i.e., the eigenvalue is divided by the real part,i.e. $(1.0 - 0.0920114i)$. Then, the estimated ground state energy is found from $-\kappa \times 0.0920114$ as -1.8510403. In comparison to the classically computed eigenvalue -1.851046 through the eigen-decomposition, this displays an estimation error around 10^{-6} caused by the error in the estimation of the powers of the matrices through Eq. (5)the error in the approximation $sin(\lambda_j/\kappa) \approx \lambda_j/\kappa$. the error is around 10^{-6}.
- In the other two cases, the power of the matrices is computed through Eq. (8) by adding one more qubit to the ancilla in each iteration. Here, because of the computational difficulty-the

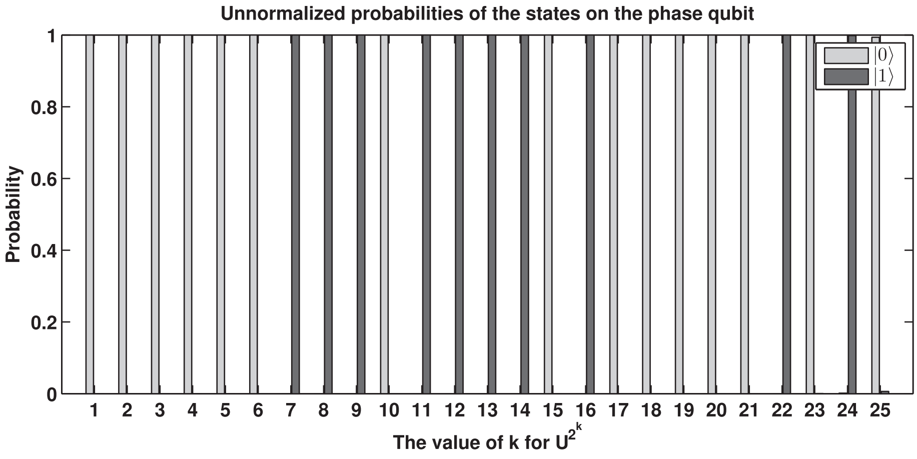

Fig. 2. The probabilities on the phase qubit with the amplitude amplification applied only to $\mathbf{U}^{(1)}$: i.e. $\mathbf{Q}^6\mathbf{U}^{(1)}$. The power of the matrix is estimated by direct application: for square it is $\mathbf{Q}^6\mathbf{U}^{(1)}\mathbf{Q}^6\mathbf{U}^{(1)}$. Also note that the probabilities are not normalized after the application of $\mathbf{P}$.

Please cite this article in press as: A. Daskin, S. Kais, Chem. Phys. (2018), https://doi.org/10.1016/j.chemphys.2018.01.002

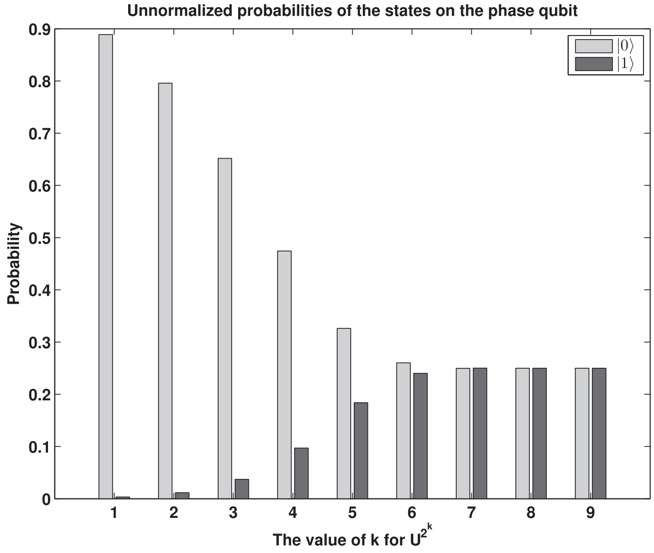

Fig. 3. The probabilities on the phase qubit without the amplitude amplification applied to $\mathbf{U}^{(1)}$. Note that the probabilities are not normalized after the application of **P**.

system size for 9 iteration requires 17 qubits, we present the output of the phase qubit for only 9 iterations of the phase estimation without and with the amplitude amplification applied to $\mathbf{U}^{(1)}$ respectively in Fig. 3 and Fig. 4. In comparison to classically computed value, the simulation results produce the correct eigenvalue if the previous bit values are provided; otherwise, it generates an estimate of the eigenvalue, as expected, with an error bounded by $O(\kappa 2^{-9})$. As seen in Fig. 3, without the amplitude amplification, depending on the value of k in $\mathbf{U}^{2^k}$, the probability difference between 1 and 0 on the phase qubit diminishes and the total success probability approaches to 0.5 (the probability of the part of the ancilla system simulating $\widetilde{\mathcal{H}}$). As shown in Fig. 3 with the same κ value, the total probability can be maximized by the application of the amplitude amplification only to $\mathbf{U}^{(1)}$. The probability difference now survives for 9 iterations. This can be further improved by using oblivious amplitude amplification algorithm in the subsequent iterations when the probability difference starts to drop and changing the value of κ. Note that a larger κ will require more iterations to obtain a higher precision. Also note that bit values and the probabilities are affected by the number of iterations because the iterative phase estimation algorithm uses a rotation gate whose angle determined by the bit values measured in the previous iterations.

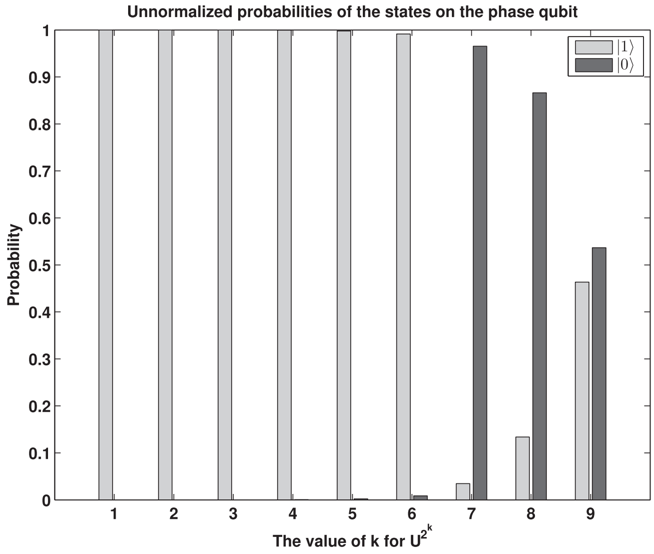

Fig. 4. The probabilities on the phase qubit with the amplitude amplification applied only to $\mathbf{U}^{(1)}$. Note that the probabilities are not normalized after the application of **P**.

6. Conclusion and future direction

In this paper, we have described a framework to estimate the eigenvalues of Hamiltonians in the phase estimation algorithm without using the time evolution operator, the exponential, of the Hamiltonian. We have showed how to find the powers of the matrices necessary in the phase estimation algorithm by successive applications. We have analyzed the circuit implementation of the whole framework in terms of classical and quantum complexities and showed that the framework provides an efficient way to estimate the eigenvalues of Hamiltonians which are written in terms of sums of simple unitary matrices (here, “simple” means a unitary matrix which can be implemented through a few number of one- and two-qubit quantum gates.). The circuits for the sum of unitary and the sum of rank one matrices have been analyzed. In addition, we have used the Hamiltonian of hydrogen molecule as an example system and showed how to estimate its ground state energy through the described method. We believe this framework can be used efficiently for many eigenvalue related problems such as the finding ground state energy of quantum systems. However, there is still a need for further research to be able simulate a Hamiltonian which is not given as a sum of simple unitaries.

References

[1] Rami Barends, Alireza Shabani, Lucas Lamata, Julian Kelly, Antonio Mezzacapo, Urtzi Las Heras, Ryan Babbush, A.G. Fowler, Brooks Campbell, Yu Chen, et al., Digitized adiabatic quantum computing with a superconducting circuit, Nature 534 (2016) 222–226.

[2] Esteban A. Martinez, Christine A. Muschik, Philipp Schindler, Daniel Nigg, Alexander Erhard, Markus Heyl, Philipp Hauke, Marcello Dalmonte, Thomas Monz, Peter Zoller, et al., Real-time dynamics of lattice gauge theories with a few-qubit quantum computer, Nature 534 (2016) 516–519.

[3] J. Salfi, J.A. Mol, R. Rahman, G. Klimeck, M.Y. Simmons, L.C.L. Hollenberg, S. Rogge, Quantum simulation of the hubbard model with dopant atoms in silicon, Nat. Commun. 7 (2016).

[4] D. Castelvecchi, Quantum computers ready to leap out of the lab in 2017, Nature 541 (2017) 9.

[5] Richard P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. 21 (1982) 467–488.

[6] Matthias Troyer, Uwe-Jens Wiese, Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations, Phys. Rev. Lett. 94 (2005) 170201.

[7] Peter W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in: 35th Annual Symposium on Foundations of Computer Science, 1994 Proceedings, IEEE, 1994, pp. 124–134.

[8] Masuo Suzuki, Generalized trotter’s formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, Commun. Math. Phys. 51 (1976) 183–190.

[9] David Poulin, Matthew B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doberty, Matthias Troyer, The trotter step size required for accurate quantum simulation of quantum chemistry, Quantum Info. Comput. 15 (2015) 361–384.

[10] Scott Aaronson, Read the fine print, Nat. Phys. 11 (2015) 291–293.

[11] Anmer Daskin, Ananth Grama, Giorgos Kollias, Sabre Kais, Universal programmable quantum circuit schemes to emulate an operator, J. Chem. Phys. 137 (2012) 234112.

[12] Anmer Daskin, Ananth Grama, Sabre Kais, A universal quantum circuit scheme for finding complex eigenvalues, Quantum Inf. Process. 13 (2014) 333–353, https://doi.org/10.1007/s11128-016-1452-3.

[13] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, Rolando D. Somma, Simulating hamiltonian dynamics with a truncated taylor series, Phys. Rev. Lett. 114 (2015), https://doi.org/10.1103/PhysRevLett.114.090502, 090502.

[14] Ammar Daskin, Sabre Kais, An ancilla-based quantum simulation framework for non-unitary matrices, Quantum Inf. Process. 16 (2017) 33.

[15] Guang Hao Low, Isaac L. Chuang, Hamiltonian simulation by qubitization, arXiv preprint arXiv: 1610.06546, 2016.

[16] Michele Mosca et al., Quantum searching, counting and amplitude amplification by eigenvector analysis, in: MFCS98 workshop on Randomized Algorithms, 1998, pp. 90–100.

[17] Gilles Brassard, Peter Hoyer, Michele Mosca, Alain Tapp, Quantum amplitude amplification and estimation, Contemporary Math. 305 (2002) 53–74.

[18] Phillip Kaye, Michele Mosca, Raymond Laflamme, An Introduction to Quantum Computing, Oxford Univ. Press, UK, 2006.

[19] Lov K. Grover, Quantum computers can search rapidly by using almost any transformation, Phys. Rev. Lett. 80 (1998) 4329.

[20] Adam Paetznick, Krysta M. Svore, Repeat-until-success: non-deterministic decomposition of single-qubit unitaries, Quantum Inf. Comput. 14 (2014) 1277–1301.

[21] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, Rolando D. Somma, Exponential improvement in precision for simulating sparse hamiltonians, in: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, ACM 2014, pp. 283–292.

[22] Shi-Jie Wei, Tao Xin, Gui-Lu Long, Efficient universal quantum channel simulation in ibm's cloud quantum computer, arXiv preprint arXiv: 1706. 08080, 2017.

[23] Gene H. Golub, Charles F. Van Loan, Matrix Computations, vol. 3, JHU Press, 2012.

[24] Peter A. Ivanov, E.S. Kyoseva, N.V. Vitanov, Engineering of arbitrary u (n) transformations by quantum householder reflections, Phys. Rev. A 74 (2006), 022323.

[25] Stephen S. Bullock, Dianne P. Leary, Gavin K. Brennen, Asymptotically optimal quantum circuits for d-level systems, Phys. Rev. Lett. 94 (2005) 230502.

[26] Jesús Urías, Diego A. Quiñones, Householder methods for quantum circuit design, Can. J. Phys. 94 (2015) 150–157.

[27] George Cybenko, Reducing quantum computations to elementary unitary operations, Comput. Sci. Eng. 3 (2001) 27–32.

[28] Juha J. Vartiainen, ikko Möttönen, Martti M. Salomaa, Efficient decomposition of quantum gates, Phys. Rev. Lett. 92 (2004) 177902.

[29] Benjamin P. Lanyon, James D. Whitfield, Geoff G. Gillett, Michael E. Goggin, Marcelo P. Almeida, Ivan Kassal, Jacob D. Biamonte, Masoud Mohseni, Ben J. Powell, Marco Barbieri, et al., Towards quantum chemistry on a quantum computer, Nat. Chem. 2 (2010) 106–111.

[30] James D. Whitfield, Jacob Biamonte, Alán Aspuru-Guzik, Simulation of electronic structure hamiltonians using quantum computers, Mol. Phys. 109 (2011) 735–750.

[31] Jacob T. Seeley, Martin J. Richard, Peter J. Love, The bravyi-kitaev transformation for quantum computation of electronic structure, J. Chem. Phys. 137 (2012) 224109, https://doi.org/10.1063/1.4768229.

[32] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, Martti M. Salomaa, Quantum circuits for general multiqubit gates, Phys. Rev. Lett. 93 (2004) 130502, https://doi.org/10.1103/PhysRevLett.93.130502.

Please cite this article in press as: A. Daskin, S. Kais, Chem. Phys. (2018), https://doi.org/10.1016/j.chemphys.2018.01.002