Statistical Approach to Quantum Phase Estimation

Alexandria J. Moore[*1], Yuchen Wang[2], Zixuan Hu[2], Sabre Kais[†2], and Andrew M. Weiner[‡1]

[1]School of Electrical and Computer Engineering and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, Indiana 47907, USA

[2]Department of Chemistry, Department of Physics and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, Indiana 47907, USA

April 20, 2021

Abstract

We introduce a new statistical and variational approach to the phase estimation algorithm (PEA). Unlike the traditional and iterative PEAs which return only an eigenphase estimate, the proposed method can determine any unknown eigenstate-eigenphase pair from a given unitary matrix utilizing a simplified version of the hardware intended for the Iterative PEA (IPEA). This is achieved by treating the probabilistic output of an IPEA-like circuit as an eigenstate-eigenphase proximity metric, using this metric to estimate the proximity of the input state and input phase to the nearest eigenstate-eigenphase pair and approaching this pair via a variational process on the input state and phase. This method may search over the entire computational space, or can efficiently search for eigenphases (eigenstates) within some specified range (directions), allowing those with some prior knowledge of their system to search for particular solutions. We show the simulation results of the method with the Qiskit package on the IBM Q platform and on a local computer.

1 Introduction

Efficient spectral decomposition of large matrices is a key component to many optimization and machine learning algorithms, with applications ranging from factoring and searching algorithms to computational chemistry [1]. On classical computers, spectral decomposition scales super-linearly with the system dimension [2], making it intractable for large problems. Due to the utility of spectral decomposition and its classical limitations, quantum approaches to spectral decomposition and eigenvalue estimation have been pursued [3]. One significant approach is the quantum phase estimation algorithm (PEA) [4] – a means of determining unknown eigenphases of a unitary matrix – which is a key subroutine in a number of quantum algorithms including Shor's factoring algorithm [5], quantum principal component analysis [6], the generalized Grover's search algorithm [7], and quantum simulations [8, 9, 10].

Near-term quantum systems operate in the noisy intermediate-scale quantum (NISQ) regime [11], facing restrictions on both circuit depth and breadth due to decoherence and gate infidelity. Consequently, interest in the traditional PEA [4] and quantum principal component analysis [6] has been channeled toward developments in the *iterative* PEA (IPEA) [12] – a method which estimates an unknown phase over multiple

---

[*]moore428@purdue.edu

[†]kais@purdue.edu

[‡]amw@purdue.edu

circuit iterations – allowing for significant reduction in both qubit usage (circuit breadth) and controlled-gate operations (circuit depth). The IPEA has been demonstrated on photonic systems [13]. On the other hand, variational quantum algorithms (VQA) – which use a classical computer to control and optimize the parameters applied in a quantum circuit – have been developed for a variety of problems as they leverage the speedup of quantum algorithm with lower-depth circuits [14, 15].

Here, we introduce a quantum-classical hybrid algorithm combining the PEA with the VQA – which we call the *Statistical* PEA (SPEA) – and show preliminary simulation results on the IBM Q platform with the Qiskit package [16] as well as simulations on a local computer. The method is able to determine any unknown eigenstate-eigenphase pair from a unitary matrix by utilizing hardware intended for the IPEA. Further, the SPEA can be applied repeatedly to obtain a full spectral decomposition. The SPEA may be compared to other variational quantum eigensolvers [17, 18, 19], the primary difference being other variational eigensolvers work directly on a (Hermitian) matrix encoded as a quantum state using specially designed quantum circuits. The SPEA assumes access to a gate representation of the unitary exponentiation of the state – or assumes simultaneous availability of several copies of the quantum state to approximate the quantum gate à la [6]. In return, the SPEA requires a polynomially-reduced number of (classical) optimization parameters – as it optimizes for a single eigenstate, rather than diagonalize the entire matrix simultaneously – and directly delivers eigenstate-eigenphase pairs (whereas other approaches may allow on-demand generation of eigenstates, but require tomography if knowledge of the state is needed). The SPEA is also able to search for eigenphases within specified ranges, allowing those with some prior knowledge of their system to search for particular solutions, whether ground state (near minimum eigenphase), principle (near-maximal eigenphase), or any other region of interest.

This paper is organized as follows: Section 2 reviews the traditional and iterative PEA and introduces a statistical metric $\mathcal{C}$ for quantifying the proximity of any given input-state to its closest eigenstate. Section 3 describes the Statistical PEA and discusses the connections between the $\mathcal{C}$ factor and the quality (in terms of proximity) of the derived eigenstate-eigenphase pairs (with the derivation details in Appendix A). Section 3 also outlines the optimization process for obtaining the eigenstate-eigenphase pairs. Simulation results on different platforms are reported and discussed in Section 4; methodology details are provided in Appendix B and C. We conclude with a discussion on the performance of the SPEA and propose future directions and applications of the method in Section 5.

2 Phase Estimation Algorithms

Traditional PEA implementations, diagrammed in Figure 1, take any given unitary $\hat{U}$ and any given eigenstate $|\nu\rangle$ of $\hat{U}$ and return the corresponding eigenphase θ where

$$\hat{U}\,|\nu\rangle = e^{i2\pi\theta}\,|\nu\rangle. \tag{1}$$

The (approximate) eigenphase $\tilde{\theta} \in [0,1)$ (equivalently, $\in [-.5, .5)$) may be directly measured on the control qubits (or qudits, when the control is d_c-dimensional) of the PEA. The target register is typically unmeasured during the process. For an arbitrary target register input $|\Phi\rangle$, the probability of the circuit representing a particular eigenstate $|\nu_k\rangle$ and the associated eigenphase θ_k is $|\langle\nu_k|\Phi\rangle|^2$. If $|\Phi\rangle$ is not itself an eigenstate, the eigenphase retrieved varies each time the PEA circuit is run. The prototypical PEA thus approximates a particular θ in a single trial.

The traditional PEA requires large quantum circuits which are often unreliable in the NISQ regime. To overcome hardware constraints, the iterative PEA (IPEA) was developed. The IPEA significantly reduces circuit depth requirements by approximating a particular θ one qubit (or d_c-level qudit) at a time, starting from the least significant qubit (qudit). The IPEA requires a rotation gate – a linear phase across the control
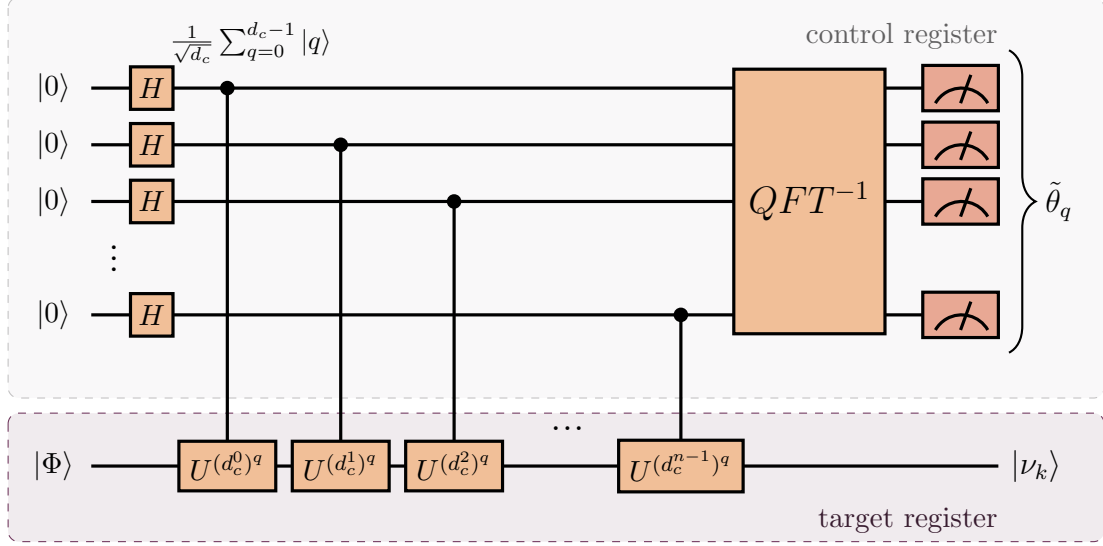
Figure 1: The traditional PEA using n control d_c-dimensional qudits in the control register. The quantum gates are colored in orange and the measurement gates in red. The target register is highlighted in purple and control register in grey. Note that in the $d_c > 2$ case, the H "Hadamard" gate acts as a discrete quantum Fourier transform (QFT) gate. Additionally, in the $d_c > 2$ case the control gates acts as MVCGs, applying the gate to the qth power when the control qudit is in state $|q\rangle$. The circuit will estimate the phase $\tilde{\theta}_k$ of eigenstate $|\nu_k\rangle$ to precision $d_c^{-(n)}$. The estimate of eigenphase θ_k is returned with probability $|\langle\Phi|\nu_k\rangle|^2$.
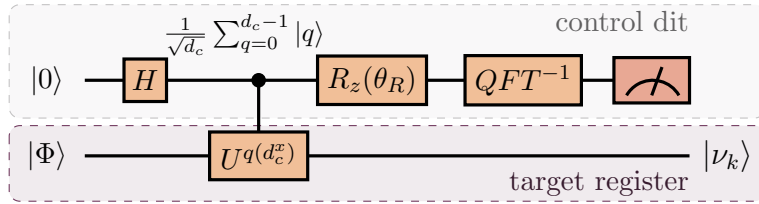

Figure 2: The iterative PEA. See Fig. 1 for color-conventions and notes on the MVCG. To retrieve n dits of the eigenphase (i.e. phase precision $\pm 1/d_c^n$), run the circuit with $x = n - 1$ and $\theta_R = 0$; the measured control state is the nth base-d_c dit of the unknown phase. Proceed to run the circuit for $x = n - 2$ and set θ_R according to the previous control results; the measured control state is the $(n - 1)^{\text{th}}$ dit. Continue the process iteratively until $x = 0$ and the entire phase is recovered. Note that the iterative method is diagrammed for a single qudit control, but may be realized with any number of control qudits, similar to the traditional PEA
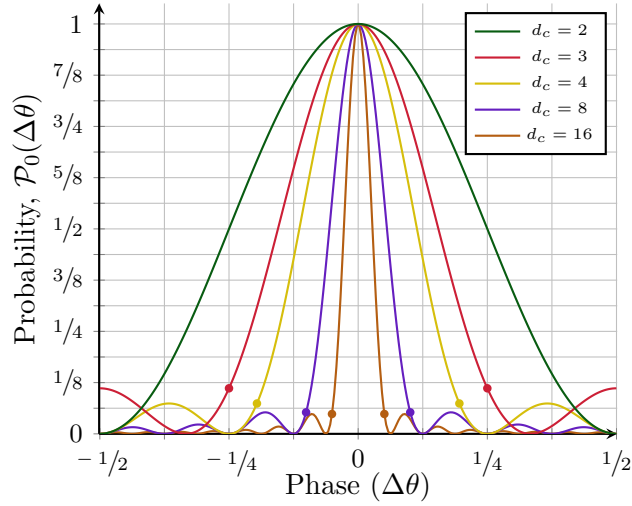
Figure 3: Plot of $\mathcal{P}_0(\Delta\theta)$ (Equation 3) from $\Delta\theta \in [-.5, .5]$ for various d_c. Note that $\mathcal{P}_0$ has infinite domain with period 1. Probability of the (d_c-level) control register of an iterative PEA collapsing to $|0\rangle$ as a function of difference between the eigenphase θ and the applied rotation θ_R, $\Delta\theta \equiv \theta - \theta_R$ for an eigenstate input. Note that when the applied rotation matches the eigenphase ($\Delta\theta = 0$), the control collapses to $|0\rangle$ deterministically. Denote the region around $\Delta\theta = 0$ (from dot to dot) as the central lobe of $\mathcal{P}_0(\Delta\theta)$, and the small lobes with local maxima outside of it as the sidelobes. See that the higher the system's dimensionality, the narrower the probability curve's central lobe and the lower local maxima in the sidelobes. Note that $d_c = 2$ has no sidelobes (the probability is monotonic on either side of the central lobe). Also note $\mathcal{P}(\Delta\theta) = 0$ for $\Delta\theta = d_c^{-1}$ and the width of the central lobe is therefore $\Delta\theta_{\text{FWFM}} = 2d_c^{-1}$.

register – to "subtract" off eigenphase information determined in previous iterations. (I.e. if the quantum circuit's state before $R_z(\theta_R)$ is $\sum_q \alpha_q |q\rangle |\Phi_q\rangle$, then after the rotation gate the quantum circuit's state is $\sum_q \alpha_q e^{-iq2\pi\theta_R} |q\rangle |\Phi_q\rangle$.) The iterative PEA, as the name suggests, requires a number of iterations equal to the number of bits (dits) of precision desired from the eigenphase. Additionally, the input to the target register of an IPEA must either be an eigenstate (and identically prepared each iteration) or the previous iteration's output must propagate forward and serve as the next iteration's input.

An IPEA circuit is diagrammed in Figure 2. In the general case, the control qudit may be high-dimensional (d_c-level). In this case, the Hadamard gates represent a d_c-dimensional quantum Fourier transform gate and the control-$\hat{U}$ gate is a multi-level control gate (MLCG) [20]: when the control state is $|q\rangle$, a $\hat{U}^q$ gate is applied to the target register. Consider the IPEA in its "last" iteration's settings ($x = 0$ in Figure 2). When the target register is an eigenstate $|\Phi\rangle = |\nu\rangle$ and the rotation gate is used to subtract off phase $2\pi\theta_R = 2\pi\theta$, the control dits deterministically collapse to state $|0\rangle$. When either the target input is not an eigenstate and/or θ_R is not the corresponding eigenphase, the control dits will collapse to $|0\rangle$ with non-unity probability.

Indeed, for eigenstate input $|\nu\rangle$ with a d_c-level control, the final state of the control qudit before measurement is

$$|\Psi_C\rangle = \frac{1}{d_c} \sum_{q=0}^{d_c-1} \sum_{n=0}^{d_c-1} e^{2\pi i n(\theta - \theta_R - \frac{q}{d_c})} |q\rangle \tag{2}$$

where θ is the eigenphase of $|\nu\rangle$ and $-\theta_R$ ($\theta_R \in [0, 1)$) is the rotation applied by the rotation gate. The

4

probability of measuring the system in output bin $|0\rangle$ is

$$\mathcal{P}_\theta(-\theta_R) = |\langle 0|\Psi_C\rangle|^2 = \frac{1}{d_c^2}\left|\sum_{n=0}^{d_c-1} e^{2\pi i n(\theta-\theta_R)}\right|^2 \tag{3}$$

$$= \mathcal{P}_0(\theta - \theta_R).$$

$\mathcal{P}_\theta(-\theta_R)$ goes to one as θ_R approaches θ, as shown in Figure 3. In the most general case, where the target register is an arbitrary (non-eigenstate) input state $|\Phi\rangle$ and the rotation gate subtracts off phase $2\pi\theta_R$, the probability that the control qudits will collapse to to $|0\rangle$ is

$$\mathcal{C}(|\Phi\rangle, \theta_R) = \sum_{k=0}^{d_t-1} |\langle\nu_k|\Phi\rangle|^2 \mathcal{P}_{\theta_k}(-\theta_R)$$

$$= \frac{1}{d_c^2}\sum_{k=0}^{d_t-1} |\langle\nu_k|\Phi\rangle|^2 \left|\sum_{n=0}^{d_c-1} e^{2\pi i n(\theta_k-\theta_R)}\right|^2 \tag{4}$$

Where $\hat{U}$ is d_t-by-d_t-dimensional and the target register is d_t-dimensional. Appreciate that $\mathcal{C}(|\Phi\rangle, \theta_R) = 1$ if and only if $|\Phi\rangle$ is an eigenstate and θ_R is its corresponding eigenphase.

3   Statistical Approach to PEAs

The non-deterministic nature of the IPEA (in the non-eigenstate case) disqualifies the circuit from use as an eigenphase estimator in the standard approach. The SPEA instead considers the probabilistic outputs of the IPEA (and PEA) as valuable information which – when coupled with a classical controller as in Figure 4 – allows quantum PEA-like hardware to be used in a variational approach to determine any unknown eigenphase-eigenstate pair. The quantum hardware required is that of a traditional PEA with single-dit precision ($n = 1$) and the rotation gate standard to the IPEA (i.e. an iterative PEA with x set to 0). The classical controller determines $|\Phi\rangle$ and θ_R which are used in the PEA-type circuit. Multiple trials of the quantum circuit are run to approximate the probability $\tilde{\mathcal{C}} \approx \mathcal{C}(|\Phi\rangle, \theta_R)$ (of Equation 4). Note that the PEA-like circuit need only detect two measurement outcomes: $|0\rangle$ and not($|0\rangle$), further reducing hardware requirements compared to typical high-dimensional PEAs. Treating the estimate $-1 \cdot \tilde{\mathcal{C}}(|\Phi\rangle, \theta_R)$ as a cost function in an optimization process (making $\tilde{\mathcal{C}}$ the negative cost function), the classical controller adjusts $|\Phi\rangle$ and θ_R, until the quantum circuit near-deterministically returns $|0\rangle$ as the output state. When $\tilde{\mathcal{C}}(|\Phi\rangle^*, \theta_R^*) \approx 1$, the classical controller has found the (approximate) eigenstate $|\Phi\rangle^*$ and the associated eigenphase θ_R^*.

The quality of the eigenstate $|\Phi\rangle^*$ and eigenphase θ_R^* retrieval can be quantified by $\mathcal{C}^* = \mathcal{C}(|\Phi\rangle^*, \theta_R^*)$. $\mathcal{C}^*$ can both (1) determine the maximum distance from the eigenphase θ_R to the nearest eigenphase θ_k and (2) find the fidelity of $|\Phi\rangle^*$ to actual eigenstate(s). Derivations of both are provided in Appendix A.

The (negative) cost function $\mathcal{C}$ acts as a metric for quality of eigenvalue-eigenstate retrieval as shown in Appendix A; by finding $|\Phi\rangle$ and θ_R which maximize this metric, we arrive at good estimates for an eigenstate ($|\Phi\rangle$) and eigenphase (θ_R) pair. Following is the classical algorithm used to maximize $\mathcal{C}$, which is similar to a gradient search algorithm:

1. The classical controller chooses a $|\Phi\rangle$ at random

2. The classical controller constructs an orthogonal basis $\{|B_m\rangle\}$ including $|\Phi\rangle$

5

instructions θ_R $\tilde{C}(|\Phi\rangle, \theta_R)$

PEA-type Circuit:

$|0\rangle$ — H — $R_z(\theta_R)$ — QFT^{-1} — [measurement]

State Preparation $|\Phi\rangle$ — U^j — $|\nu_k\rangle$
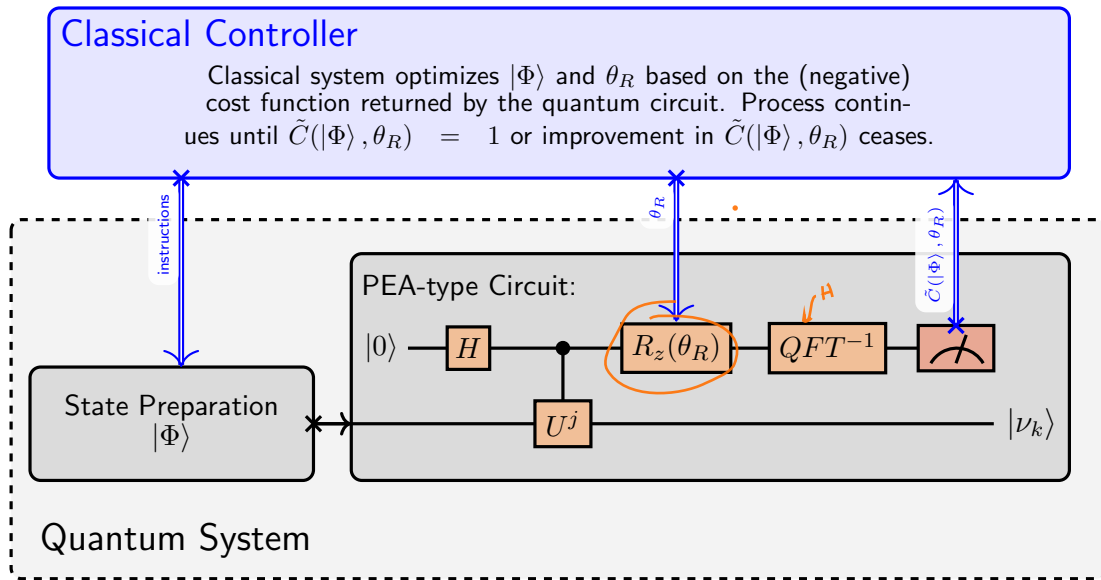
Quantum System

Figure 4: Diagram of variational classical-quantum system. Classical processes are indicated by double blue lines and quantum processes by single black lines. Quantum gates are shown in orange and measurement gates in red. The (potentially high-dimensional) PEA-type circuit is simplified from the typical iterative PEA in that U need not be raised to high orders (d_c^x) corresponding to desired eigenphase precision and the measurement gate need only distinguish between the $|0\rangle$-state and the not($|0\rangle$)-state. The (negative) cost function (estimate) $\tilde{C}$ is returned after a predetermined number of trials of the quantum circuit, approximating a probability.

[handwritten: $\theta_R \, [0,1]$, resolution → 100]

3. • (Standard Method: viable when the quantum circuit can measure output bins $|0\rangle$ and not($|0\rangle$)). The quantum circuit evaluates $\tilde{C}(|\Phi\rangle, \theta_R)$ over a range of θ_R and returns the maximum value $\mathcal{C}^*$.

   • (Alternative Method: viable when the quantum circuit can measure all d_c output bins: $|0\rangle$ through $|d_c - 1\rangle$.)
   [handwritten: which range?]
   The quantum circuit evaluates $\tilde{C}(|\Phi\rangle, 0)$ and uses this result to approximate the eigenphase θ^*. The quantum circuit then evaluates $\tilde{C}(|\Phi\rangle, \theta_R = \theta^*)$ and returns $\mathcal{C}^*$.

4. For all $m = 0$ to $2d_t - 1$, we set a=1 and the following occurs:

   • if $m \geq d_t$ then $z = \sqrt{-1}$. Otherwise $z = 1$. *[handwritten: ?]*

   • the classical controller generates the new state:

   $$|\Phi'\rangle = \frac{|A\rangle}{\sqrt{\langle A|A\rangle}}$$

   $$\text{where } |A\rangle = |\Phi\rangle + z \cdot a \cdot (1 - \mathcal{C}^*)|B_{m \mod d_t}\rangle$$

   [handwritten: why? d_t^{ξ}; would be able to capture if $|\Phi\rangle$ was in span of $|\Phi\rangle$ & B_m]

   (5)

   • $|\Phi'\rangle$ is fed to the quantum circuit, the maximum value returned is $\mathcal{C}'^*$
   • if $\mathcal{C}'^* > \mathcal{C}^*$, then $|\Phi\rangle = |\Phi'\rangle$ and $\mathcal{C}^* = \mathcal{C}'^*$. Otherwise $|\Phi\rangle$ and $\mathcal{C}^*$ are unchanged.

5. If $|\Phi\rangle$ was not updated during step 4, set $a = a/2$ and repeat step 4.

6. If $\mathcal{C}^*$ is greater than the stopping condition or the maximum run-time has been exceeded, the classical controller concludes and returns $|\Phi\rangle$, $\mathcal{C}^*$, and θ_R^*. Otherwise the process continues from step 2.

A few observations on the optimization process may be made. For each iteration, at least $2d_t$ distinct input states are used. For each of these input states a set of $\{\theta_R\}$ is applied (when using the 'standard

approach' in step 3). Initially, the $\{\theta_R\}$ range from 0 to 1 with coarse resolution; as the optimization proceeds, $\{\theta_R\}$ will become fine and include phases from a limited region. Notably, we can choose to run the optimization process limiting $\{\theta_R\}$ to a narrow range of space from the outset. In this way, we may choose to look only for ground state (small θ), principle (large θ), or any other particular solutions to Equation 1. In addition, we may eliminate known eigenstates or directions not of interest by excluding them from $\{|B_m\rangle\}$ (step 2) each iteration. In this fashion, the SPEA may be used to determine a complete (or partial) spectral decomposition of $\hat{U}$. Finally, we note while the hardware conventional to a PEA is utilized, this system is superior to the original PEA, as it determines both the eigenstate and the eigenphase, given no prior knowledge.

4 Statistical PEA Simulation

We test the proposed algorithm on the IBM Q platform and on a local computer. In both cases, a classical computer is used to simulate the $\mathcal{C}$ parameter (of Equation 4) delivered by a quantum circuit. These simulations of a quantum system are ideal; neither the IBM Q nor the local computer simulations include any noise terms. I.e. all quantum gates are assumed to operate with perfect fidelity. The IBM Q trials study the convergence of the optimization algorithm to any single eigenstate on 2- and 4-dimensional systems. The local computer simulations run a *full* spectral decomposition on a 16-dimensional system with various control levels d_c.

Both simulations apply the variational algorithm as defined in Section 3, with one primary difference: the local computer simulations follows the primary method of step 3 whereas the IBM Q simulations follow the alternative method. The IBM Q simulation runs one measurement with $R_z(\theta_R = 0)$ and applies the eigenphase estimation methodology introduced in the Discussion of [20] – under the (inaccurate first, but increasingly accurate) assumption that the input state is an eigenstate – to obtain a phase estimate θ^*. Then, the measurement is run with $R_z(\theta_R = -\theta^*)$ to obtain the metric $\mathcal{C}$ used for the optimization. By contrast, the local computer's simulations follow the primary method, picking a representative sample of input phases to apply to the R_z gate and selecting the largest $\mathcal{C}$ that arises. The local computer's simulations therefore require more runs of the quantum circuit per trial, but only require two control-qudit detectors: one for the $|0\rangle$ state and one for the not($|0\rangle$), whereas the IBM Q methodology needs one detector for each control level ($|0\rangle, |1\rangle, ..., |d_c - 1\rangle$). The alternative approach (or some hybrid approach) is generally preferable if the hardware is available for d_c detectors.

4.1 Qiskit Simulation

On the IBM Q experience platform, we developed our quantum algorithms with Qiskit, the python-based programming package provided by IBM Q which offers all the facilities to design, simulate and execute quantum algorithms on IBM's quantum computers [16]. In this section we present the simulation results of the SPEA on the Qiskit quantum simulator.

Three sets of simulations are run on the IBM Q, one with 2-dimensional operator U_1 and the other two with 4-dimensional operators U_2 and U_3, the matrix forms of which are shown in Appendix B. U_1 and U_2 are operators directly built with the default gates offered by the IBM Q and U_3 is a unitary exponentiation based on the Hamiltonian of the hydrogen molecule generated with Bravyi-Kitaev transformation[21]. The second quantization Hamiltonian of a hydrogen molecule with a bond length 0.74Å is calculated by the $STO-3G$ minimal basis using PySCF [22] and the transformation is done by OpenFermion[23]. We encode the matrix into the "Operator" class provided by Qiskit[16]. In the simulations of each unitary operation U_i where $i = 1, 2, 3$, we start with the input states that are good approximations of one of the operator's eigenstates and then move to input states which are nearly equal-distance from every eigenstate. We quantify

the distance of the input state $|\Phi\rangle$ to its nearest eigenstate $|\nu\rangle$ by taking the absolute inner product $|\langle\Phi|\nu\rangle|$ as reported in Table 1. In each simulation we run the same input state 20 times and set the maximum iteration number to be 50 (to save the resources) and the stopping condition, which is the difference between the $\mathcal{C}$ factor and 1, to be 10^{-4}. The stopping condition is set so that when it is met we will have a reasonably good approximation of the eigenstate. We then calculate the average number of iterations and standard deviation of the number of iterations required to exceed the stopping condition. Most trials reach the stopping condition before exceeding the iteration limit and give a good approximation of one of the eigenstate-eigenphase pair, as indicated by the low mean phase error reported. The results are shown in Table 1.

For each operator U_1, U_2, U_3, input states which are initially close to an eigenstate (input states with a large absolute inner product) have lower required iteration number than those which are initially far from all eigenstates (low absolute inner product). Appreciate that the eigenstate converged to is non-deterministic, as the optimizer itself is non-deterministic due to randomness added by the random orthogonal basis in step 2. In other words, added randomness may converge the input state to an eigenstate other than the closest eigenstate. The mean phase error recorded in Table 1 is calculated by taking the absolute value of the difference between the eigenphase of the converged input state and the true eigenphase of the eigenstate that the input state converged to. As the input state can converge to different eigenstates in the simulation, we report the absolute phase error rather than the error percentage. No correlation between the phase error and the absolute inner product is apparent, indicating the quality of the final eigenphase-eigenstate pair is agnostic to the proximity of the initial input state to any eigenstate. Variations in mean phase error are likely a function of which particular eigenphase-eigenstate pair was converged to. During the simulation of U_3 with an input state of equal weight combination of all the eigenstates – i.e. the hardest input state to converge to an eigenstate – there are few cases that the iteration limit is reached and the simulation did not reach the stopping condition. This can usually be fixed by increasing the iteration limit.

In summary, these results indicate that the SPEA method is capable of delivering high-quality estimates of eigenphase-eigenstate pairs with no prior knowledge of the operator's eigenstates, in the case of both arbitrary (U_1, U_2) and physically relevant (U_3) operators. The quality of the estimates is not influenced by prior system knowledge; however, the resources required to deliver an eigenstate-eigenphase pair may be reduced with prior knowledge.

4.2 Full Spectral Decomposition

The statistical approach differs from some other variational approaches [17] in that it does not diagonalize the input state matrix, but solves for only one eigenphase-eigenstate pair. This allows for significant reduction in the number of parameters (and iterations) needed to perform the optimization. However, as shown below, a complete spectral decomposition is realizable. As a representative case, we consider the 16-by-16 Hamiltonian $\mathcal{H}_{H_2O}$ for the water molecule H_2O with the H-O-H angle at $104.5°$ and the bond length at 1.0 a.u. given in Appendix C [24]. The Hamiltonian is converted to a unitary exponentiation, $U_{H_2O} = e^{i\mathcal{H}_{H_2O}}$, and the matrix's spectral decomposition is simulated with the statistical variational algorithm (SPEA) on a local computer.

The simulation is run for various control levels d_c until 120 successful spectral decompositions are achieved. Each of the 16 eigenphases are retrieved in a random order. The optimization runs until $\mathcal{C}^* \geq C_{goal}$. If the optimizer is unable to reach C_{goal}, the process for that eigenphase will conclude so long as $\mathcal{C}^* \geq C_{req}$. If C_{req} is not met, the entire spectral decomposition is abandoned and the trial is classified as failed. Generally, C_{goal} is achieved for the first 10 eigenvalues and the latter 2 to 6 eigenvalues must settle at a lower value (due to small cumulative errors). Results are recorded in Table 2 and plotted in Figure 5.

To determine the fidelity of the spectral decomposition, the retrieved eigenphase-eigenstate pairs, $(\theta_k, |\nu_k\rangle)$

Operation	Input State	Abs. Inner Product	Iteration Mean	S.D.	Phase Error Mean
U_1	(0.1951, 0.9808)	0.98	6.20	2.82	$1.099 \cdot 10^{-2}$
	(0.3827,0.9239)	0.92	8.15	3.41	$1.005 \cdot 10^{-2}$
	(0.7071,0.7071)	0.71	8.90	3.34	$1.005 \cdot 10^{-2}$
U_2	(0 , 0, 0.7432, 0.6690)	0.99	5.85	8.14	$2.083 \cdot 10^{-2}$
	(0 , 0,0.6690, 0.7432)	0.99	6.7	10.42	$2.168 \cdot 10^{-2}$
	(0,0,1,0)	0.71	17.7	6.06	$1.663 \cdot 10^{-2}$
	(1,0,0,0)	0.71	23.05	11.22	$2.167 \cdot 10^{-2}$
	(0.7071, 0 , 0.7071, 0)	0.50	21.3	10.71	$2.262 \cdot 10^{-2}$
U_3	(-0.1379, 0 , 0 , 0.9904)	0.99	1.15	0.36	$1.885 \cdot 10^{-2}$
	(0 , 0.7807, 0.6247, 0)	0.99	1.1	0.3	$1.508 \cdot 10^{-2}$
	(0,1,0,0)	0.71	4.35	4.17	$1.414 \cdot 10^{-2}$
	(0.7071, 0 ,0 , 0.7071)	0.62	4.15	1.01	$1.570 \cdot 10^{-2}$
	(0.5774, 0.5774, 0 , 0.5774)	0.51	21.5	11.06	$2.199 \cdot 10^{-2}$

Table 1: IBM Q Qiskit QASM simulator results. Three unitary operators are simulated on the IBM Q platform. The SPEA is run 20 times starting from each input state. The distance from the input state to the nearest eigenstate is quantified by the absolute inner product of the two vectors (inner product 1 being identical and smaller values indicting greater difference). The table records the mean and standard deviation (S.D.) of the number of iterations needed to reach the stopping condition $(1 - \tilde{\mathcal{C}} = 10^{-4})$. The average absolute phase error is reported in radians. For each U_i the input states range from good approximations of one of the operator's eigenstates to input states which are nearly equal-distance from every eigenstate. See that the iteration mean tends to increase with decreasing inner product but the ultimate phase error is generally agnostic to the input state difference.

d_c	C_{goal}	C_{req}	Trials	Fails	Fidelity Mean	S.D.	Phase Error Mean	S.D.
2	0.999	0.95	120	77	0.984	$7.29 \cdot 10^{-3}$	$2.84 \cdot 10^{-2}$	$10.7 \cdot 10^{-2}$
2	0.995	0.9	120	13	0.966	$10.68 \cdot 10^{-3}$	$4.34 \cdot 10^{-2}$	$17.3 \cdot 10^{-2}$
3	0.995	0.9	120	17	0.981	$6.62 \cdot 10^{-3}$	$3.12 \cdot 10^{-2}$	$9.14 \cdot 10^{-2}$
4	0.995	0.9	120	32	0.986	$5.32 \cdot 10^{-3}$	$2.40 \cdot 10^{-2}$	$7.36 \cdot 10^{-2}$
5	0.995	0.9	120	30	0.986	$7.10 \cdot 10^{-3}$	$1.86 \cdot 10^{-2}$	$4.98 \cdot 10^{-2}$
6	0.995	0.9	120	26	0.989	$4.80 \cdot 10^{-3}$	$1.53 \cdot 10^{-2}$	$3.80 \cdot 10^{-2}$
7	0.995	0.9	120	60	0.991	$7.05 \cdot 10^{-3}$	$1.37 \cdot 10^{-2}$	$3.65 \cdot 10^{-2}$
8	0.995	0.9	120	132	0.992	$6.13 \cdot 10^{-3}$	$1.20 \cdot 10^{-2}$	$3.54 \cdot 10^{-2}$

Table 2: Statistics for 120 successful trials of complete spectral decomposition of U_{H_2O} matrix. C_{goal} is the $\tilde{C}$ value the optimizer attempts to reach, and generally does reach for at least the first 10 (of 16) eigen-estimates. C_{req} is the $\tilde{C}$ value the optimizer is required to reach for all eigen-estimates, else the trial is abandoned. For successful trials, the mean Fidelity and standard deviation are reported, as well as the mean eigenphase error (in radians). Note that $d_c = 2$ was run on two different C_{goal}, C_{req} levels for comparison.
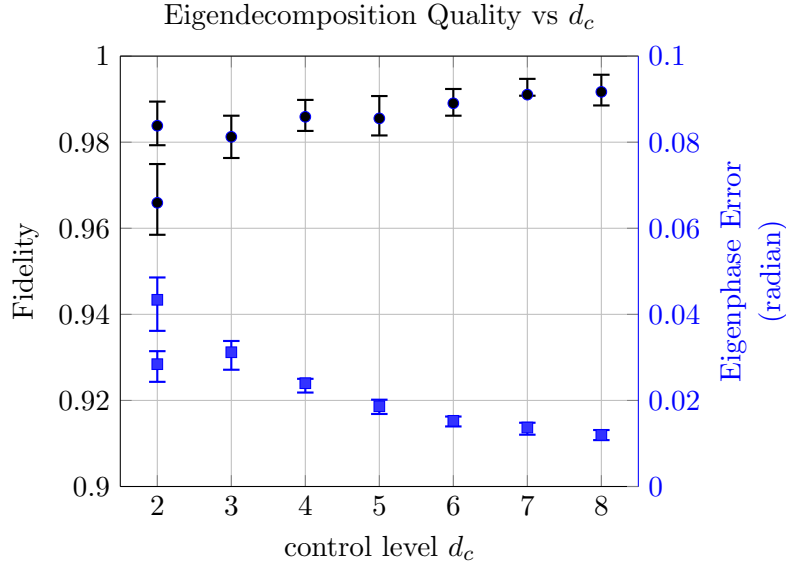
Figure 5: (Left axis; black; circles) mean fidelity achieved for optimization simulation at control level d_c, with error bars from the 25th to 75th percentile. (Right axis; blue; squares) mean phase error (per phase) for optimization simulation at control level d_c, with error bars from the 25th to 75th percentile.

were used to create the matrix,

$$U_{retrieved} = \sum_k e^{i2\pi\theta_k} |\nu_k\rangle \langle \nu_k| . \tag{6}$$

Letting $M = U_{H_2O}^\dagger U_{retrieved}$, the fidelity is defined as

$$\text{fidelity} = \frac{1}{n \cdot (n+1)} \big(Tr(MM^\dagger) + |Tr(M)|^2 \big) \tag{7}$$

following the average fidelity definition of [25] where n is matrix dimension (i.e. $n = 16$). The reported phase error is the average absolute phase error over all 16 phases,

$$\text{phase error} = \frac{\sum_k |2\pi\theta_k - 2\pi\theta_{k,true}|}{n} . \tag{8}$$

Note that the $d_c = 2$ was run for two different sets of C_{goal} and C_{req}. Increasing these values increased the optimization failure rate, but also improved decomposition fidelity and reduced the average eigenphase error. The high failure rate suggests superior results will be achieved by increasing d_c, the number of control levels, over increasing $\mathcal{C}^*$ (analogous to C_{goal}), when possible. This is expected, as increasing d_c leads to a narrower cost function. Overall, these results indicate both the viability of the SPEA for full and partial eigenphase recovery and provides an example of a quantum algorithm which benefits from working with high-dimensional quantum states, i.e. qudits.

5 Conclusion

In this work, we have proposed a novel statistical variational approach (SPEA) to the quantum phase estimation algorithm (PEA). From the probabilistic output of a PEA circuit using non-eigen input states, we have defined a statistical metric $\mathcal{C}$ indicating the proximity of any given input state to the nearest eigenstate and develop an optimization process that can variationally retrieve all the eigenstate-eigenphase

pairs of a given unitary operator. The SPEA takes advantage of the hardware intended for the Iterative PEA and therefore requires no novel quantum hardware development. The main disadvantage of the SPEA is the non-deterministic nature of the measurements requires running the quantum circuit repeatedly for each measurement setting. However, in the near-term era, repeated runs of a quantum circuit per measurement is already the norm, due to noise and imperfect gate fidelity. One of the main advantages of the SPEA compared to the PEA and IPEA is the ability to systematically find both the eigenstates and associated eigenphases, rather than just the eigenphases.

The simulations on the IBM Q platform with Qiskit proves the feasibility of applying the SPEA on standard quantum computation platforms. On the local computer, the full spectral decomposition of the operator generated from the water molecule Hamiltonian demonstrates the viability of the SPEA for applications in quantum chemistry. The ability to retrieve eigenstates and efficiency (in terms of low iterations requirement) of this method shows the improvement to the original PEA methods and offers the clear potential to work with larger physical and chemical systems.

Future work includes improving the optimization process with a more sophisticated algorithm for the classical controller. In addition to improving efficiency and failure rate, this may also improve the accuracy of the eigenphase-eigenstate retrieval as well as the fidelity of the full spectral decomposition. The efficiency and the viability of our methods enable us to simulate more complex systems in quantum chemistry. Future work also includes implementing this method on real computational systems provided by the IBM Q and also on a photonic platform with high-dimensional control qudit capabilities.

Acknowledgements

References

[1] S. Kais et al. *Quantum Information and Computation for Chemistry*. Advances in Chemical Physics. Wiley, 2014. ISBN: 9781118742600. URL: https://books.google.com/books?id=dCXPAgAAQBAJ.

[2] Edo Liberty et al. "Randomized algorithms for the low-rank approximation of matrices". In: *Proceedings of the National Academy of Sciences* 104.51 (2007), pp. 20167–20172. ISSN: 0027-8424. DOI: 10.1073/pnas.0709640104. eprint: https://www.pnas.org/content/104/51/20167.full.pdf. URL: https://www.pnas.org/content/104/51/20167.

[3] Anmer Daskin and Sabre Kais. "Decomposition of unitary matrices for finding quantum circuits: Application to molecular Hamiltonians". In: *The Journal of chemical physics* 134.14 (2011), p. 144112.

[4] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[5] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[6] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis". In: *Nature Physics* 10.9 (2014), pp. 631–633.

[7] Tim Byrnes, Gary Forster, and Louis Tessler. "Generalized Grover's Algorithm for Multiple Phase Inversion States". In: *Phys. Rev. Lett.* 120 (6 Feb. 2018), p. 060501. DOI: 10.1103/PhysRevLett.120.060501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.060501.

[8]   Hefeng Wang et al. "Quantum algorithm for obtaining the energy spectrum of molecular systems". In: *Physical Chemistry Chemical Physics* 10.35 (2008), pp. 5388–5393.

[9]   Alán Aspuru-Guzik et al. "Simulated quantum computation of molecular energies". In: *Science* 309.5741 (2005), pp. 1704–1707.

[10]   Ammar Daskin and Sabre Kais. "Direct application of the phase estimation algorithm to find the eigenvalues of the hamiltonians". In: *Chemical Physics* 514 (2018), pp. 87–94.

[11]   John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79.

[12]   Miroslav Dobšíček et al. "Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark". In: *Physical Review A* 76.3 (2007), p. 030306.

[13]   Stefano Paesani et al. "Experimental Bayesian quantum phase estimation on a silicon photonic chip". In: *Physical review letters* 118.10 (2017), p. 100503.

[14]   Daochen Wang, Oscar Higgott, and Stephen Brierley. "Accelerated variational quantum eigensolver". In: *Physical review letters* 122.14 (2019), p. 140504.

[15]   Alberto Peruzzo et al. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature communications* 5.1 (2014), pp. 1–7.

[16]   Héctor Abraham et al. *Qiskit: An Open-source Framework for Quantum Computing.* 2019. DOI: `10.5281/zenodo.2562110`.

[17]   Ryan LaRose et al. "Variational quantum state diagonalization". In: *npj Quantum Information* 5.1 (2019), pp. 1–10.

[18]   M. Cerezo et al. *Variational Quantum State Eigensolver.* 2020. eprint: `arXiv:2004.01372`.

[19]   Peter JJ O'Malley et al. "Scalable quantum simulation of molecular energies". In: *Physical Review X* 6.3 (2016), p. 031007.

[20]   Hsuan-Hao Lu et al. "Quantum Phase Estimation with Time-Frequency Qudits in a Single Photon". In: *Advanced Quantum Technologies* 3.2 (2020), p. 1900074.

[21]   Jacob T Seeley, Martin J Richard, and Peter J Love. "The Bravyi-Kitaev transformation for quantum computation of electronic structure". In: *The Journal of chemical physics* 137.22 (2012), p. 224109.

[22]   Qiming Sun et al. "PySCF: the Python-based simulations of chemistry framework". In: *WIREs Computational Molecular Science* 8.1 (2018), e1340. DOI: `https://doi.org/10.1002/wcms.1340`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1340`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1340`.

[23]   Jarrod R McClean et al. "OpenFermion: the electronic structure package for quantum computers". In: *Quantum Science and Technology* 5.3 (June 2020), p. 034014. DOI: `10.1088/2058-9565/ab8ebc`. URL: `https://doi.org/10.1088/2058-9565/ab8ebc`.

[24]   Teng Bian et al. "Quantum computing methods for electronic states of the water molecule". In: *Molecular Physics* 117.15-16 (2019), pp. 2069–2082.

[25]   Line Hjortshøj Pedersen, Niels Martin Møller, and Klaus Mølmer. "Fidelity of quantum operations". In: *Physics Letters A* 367.1-2 (2007), pp. 47–51.

[26]   Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `10.1038/s41592-019-0686-2`.

A   Quality of Eigenphase-Eigenstate retrieval

Preliminary note: All phases $\theta \in [0, 1)$. The difference between two phases can be found via the function

$$d(\theta_a, \theta_b) \equiv \frac{\text{angle}(e^{i2\pi(\theta_a - \theta_b)})}{2\pi}. \tag{9}$$

I.e. $d(\theta_a, \theta_b) \in [-.5, .5)$. In short, as the phase wraps around the phase difference also wraps around. E.g. $d(\frac{1}{8}, \frac{3}{4}) = \frac{3}{8}$, not $\frac{-5}{8}$. And $d(\frac{3}{4}, \frac{1}{8}) = \frac{-3}{8}$, not $\frac{5}{8}$. Like usual differences, $d(\theta_a, \theta_b) = -d(\theta_b, \theta_a)$. Appreciate that – for $\mathcal{P}_0$ of Equation 3 – $\mathcal{P}_0(d(\theta_a, \theta_b)) = \mathcal{P}_0(\theta_a - \theta_b) \forall \theta_a, \theta_b$.

Here, we quantify the quality of the eigenstate $|\Phi\rangle^*$ and eigenphase θ_R^* using $\mathcal{C}^* = \mathcal{C}(|\Phi\rangle^*, \theta_R^*)$. When $\mathcal{C}^*$ is greater than the largest (non-global) local maximum of $\mathcal{P}_\theta$ (of Equation 3), then θ_R^* must be within the primary lobe of $\mathcal{P}_\theta$ (examples shown in Figure 3). That is, when

$$\mathcal{C}^* > \max_{\zeta \in [\frac{1}{d_c}, \, 0.5]} \mathcal{P}_0(\zeta) \tag{10}$$

we are within the primary lobe of $\mathcal{P}_\theta$. Let θ_{k^*} be the eigenvalue closest to θ_R and define $\delta^* \equiv d(\theta_{k^*}, \theta_R)$. When Equation 10 is true, then $\mathcal{P}_0(\delta^*) \geq \mathcal{P}_0(\theta_k - \theta_R) \forall k$ and

$$\begin{aligned}
\mathcal{C}^* &= \sum_{k=0}^{d_t-1} |\langle \theta_k | \Phi \rangle|^2 \mathcal{P}_0(\theta_k - \theta_R) \leq \sum_{k=0}^{d_t-1} |\langle \theta_k | \Phi \rangle|^2 \mathcal{P}_0(\delta^*) \\
&= \mathcal{P}_0(\delta^*) \sum_{k=0}^{d_t-1} |\langle \theta_k | \Phi \rangle|^2 = \mathcal{P}_0(\delta^*)
\end{aligned} \tag{11}$$

$$\mathcal{C}^* \leq \mathcal{P}_0(\delta^*)$$

As $\mathcal{P}_0$ is symmetric and monotonic within the primary lobe,

$$|\delta^*| \leq \mathcal{P}_0^{-1}(\mathcal{C}^*). \tag{12}$$

Therefore the estimated eigenphase θ_R^* is within $\pm \mathcal{P}_0^{-1}(\mathcal{C}^*)$ of the nearest eigenphase (whenever Equation 10 is met).

Now to quantify the eigenstate estimate. Define a Δ-eigenstate $|\nu_\Delta\rangle$ as any superposition of eigenstates where the corresponding eigenphases are within $\pm\Delta$ of θ_R^*

$$|\nu_\Delta\rangle = \sum_m \alpha_m |\nu_m\rangle \tag{13}$$
$$\text{where } |d(\theta_m, \theta_R^*)| \leq \Delta \forall m$$

(and where $\sum_m |\alpha_m|^2 = 1$). That is, $|\nu_\Delta\rangle$ is a superposition of eigenstates (indexed $\{m\}_\Delta$) that are *nearly* degenerate: the corresponding eigenphases are all within 2Δ of one another. Proceeding from Equation 4 (whenever Equation 10 holds),

$$\begin{aligned}
\mathcal{C}^* = \mathcal{C}(|\Phi\rangle^*, \theta_R^*) &= \sum_{k=0}^{d_t-1} |\langle \nu_k | \Phi \rangle|^2 \mathcal{P}_0(\theta_k - \theta_R^*) \\
&\leq \sum_{k \in \{m\}_\Delta} |\langle \nu_k | \Phi \rangle|^2 (1) + \sum_{k \notin \{m\}_\Delta} |\langle \nu_k | \Phi \rangle|^2 \mathcal{P}_0(\theta_k - \theta_R^*) \\
&\leq \sum_{k \in \{m\}_\Delta} |\langle \nu_k | \Phi \rangle|^2 + \sum_{k \notin \{m\}_\Delta} |\langle \nu_k | \Phi \rangle|^2 \mathcal{P}_0(\Delta).
\end{aligned} \tag{14}$$

13

Letting $\sum_{k \in \{m\}_\Delta} |\langle \nu_k | \Phi \rangle|^2 = f$,

$$\mathcal{C}^* \leq f + (1-f)\mathcal{P}_0(\Delta)$$
$$\therefore f \geq \frac{\mathcal{C}^* - \mathcal{P}_0(\Delta)}{1 - \mathcal{P}_0(\Delta)} \tag{15}$$

The estimated eigenstate $|\Phi\rangle^*$ matches some Δ-eigenstate (as defined by Equation 13) with fidelity f given by Equation 15 (whenever Equation 10 holds and $|\Delta| \in [0, \frac{1}{d_c}]$).

B   Details for the IBM Q SPEA calculations

On IBM Q we realize an SPEA with a four-dimensional control register by using two qubits (the top two rails) as controls. The target is either two- or four-dimensional, using the bottom one or two rails, respectively.

We list out the three operators in matrix form used in our simulations on the IBM Q accompanied by the eigenstates of each matrix as well as showing how we achieved these matrices with the Qiskit.

In the following we use the rotation-Z gate as defined by Qiskit [16]:

$$RZ(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{16}$$

as well as the phase gate:

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{17}$$

and the Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{18}$$

The first operator U_1 is a single qubit rotation-Z gate with $\theta = \pi/2$,

$$U_1 = RZ(\pi/2) = \begin{pmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \text{ with eigenstates } v_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{19}$$

The second operator is a two qubit operation achieved by a phase gate $P(\theta = \pi/4)$ acting on the first qubit and a rotation-Z gate $RZ(\theta = \pi/2)$ sandwiched between two Hadamard gates H, acting on the second qubit. The matrix form is

$$U_2 = P(\pi/4) \otimes \left(H \cdot RZ(\pi/2) \cdot H \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i & 0 & 0 \\ -i & 1 & 0 & 0 \\ 0 & 0 & e^{i\frac{\pi}{4}} & e^{-i\frac{\pi}{4}} \\ 0 & 0 & e^{-i\frac{\pi}{4}} & e^{i\frac{\pi}{4}} \end{pmatrix} \tag{20}$$

with the eigenstates

$$v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} v_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}, v_5 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, v_6 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} \tag{21}$$

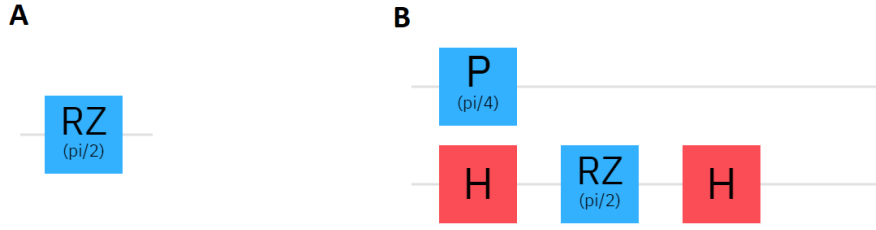The gate representations of the two operators can be found in Figure 6.

Figure 6: Gate representation of the unitary operators applied in the simulation. **(A)** represents operator U_1, a 2-dimensional operator applied to a single qubit. **(B)** represents operator U_2, a 4-dimensional operator applied to two qubits. RZ is the rotation-Z gate, P is the phase gate and H is the Hadamard gate. When a control qubit is present the RZ gate and the P gate become controlled gates. The Hadamard gates will serve as their own inverse and therefore do not need to be implemented as controlled gates.

For the third operators we start with the lowest energy Hamiltonian of the hydrogen molecule generated with Bravyi-Kitaev transformation in $STO-3g$ basis,

$$H = \begin{pmatrix} 0.48704885 & 0 & 0 & 0.18065279 \\ 0 & -0.33769999 & 0.18065279 & 0 \\ 0. & 0.18065279 & -0.33769999 & 0 \\ 0.18065279 & 0 & 0 & -1.11719411 \end{pmatrix}. \tag{22}$$

Then we take the exponential of the Hamiltonian to generate our unitary operators

$$U_3 = e^{iH} = \begin{pmatrix} 0.8686 + 0.4687i & 0 & 0 & 0.0499 + 0.1531i \\ 0 & 0.9282 - 0.3259i & 0.0595 + 0.1695i & 0 \\ 0 & 0.0595 + 0.1695i & 0.9282 - 0.3259i & 0 \\ 0.0499 + 0.1531i & 0 & 0 & 0.4256 - 0.8905i \end{pmatrix}. \tag{23}$$

The exponential of the Hamiltonian is done with the scipy.linalg.expm in the scipy python package [26]. The eigenstates are

$$v_7 = \begin{pmatrix} -0.1105 \\ 0 \\ 0 \\ 0.9939 \end{pmatrix} v_8 = \begin{pmatrix} 0 \\ 0.7071 \\ -0.7071 \\ 0 \end{pmatrix}, v_9 = \begin{pmatrix} 0 \\ 0.7071 \\ 0.7071 \\ 0 \end{pmatrix}, v_{10} = \begin{pmatrix} -0.9939 \\ 0 \\ 0 \\ 0.1105 \end{pmatrix} \tag{24}$$

We design the phase estimation algorithms that work with up to two qubits in the control register and four qubits in total as shown in Figure 7. Due to the restrictions in the qubit numbers, the simulations on the IBM Q are focused on the lower dimensional systems, i.e. one or two qubits in the target register. The rotation RZ gates are applied to each qubit in the control register after the controlled-U_i operations but before inverse Fourier transform. Every time the quantum algorithm is called to generate a new $\mathcal{C}$ factor (following the "alternative method" in step 3 of Section 3), the algorithm runs twice: the first time the RZ gates are set to zero and the phase factor θ_R is calculated statistically; the second time the upper RZ gate applies phase $-\theta_R$ and the lower gate applies phase $-2\theta_R$, together acting as a $-\theta_R$-rotation would on a $d_c = 4$ qudit system. The classical optimization process described in Section 3 is implemented using python. The basis set $\{|B_m\rangle\}$ is generated using the Gram-Schmidt methods with the input vector plus a set of linearly independent vectors obtained from a randomly generated unitary matrix. As a deviation from how the algorithm is described in the main text, the search step a factor is set to $1/2$ in step 4 and is doubled
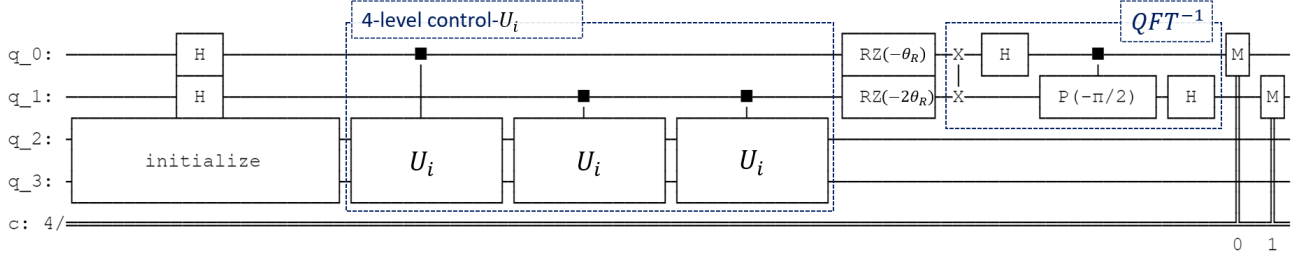
Figure 7: Illustration of the 4-level control ($d_c = 4$) SPEA circuit implemented using Qiskit. H is the Hadamard gate, RZ is the rotation-Z gate, P is the phase gate, and M represents a measurement. The top two rails are the control qubits and realize a 4-dimensional control register. The bottom two rails represents the target register. The initialize block prepares the input state differently throughout the optimization process. U_i represents the unitary the SPEA is running. For $i = 2, 3$ we implement the circuit as diagrammed. For $i = 1$, the unitary is 2-dimensional and we use only a single target rail. The three control gates together realize a multi(4)-level control gate as described in the main text. The two RZ-gates together realize a realize four-level $R_z(-\theta_R)$ as described in the main text. The double line at the bottom represents the classical information retrieved from the measurement gates (a total of 2 bits).

in step 5 (rather than halved) if $\mathcal{C}$-factor is not updated, up to seven times. The optimization concludes when the $\mathcal{C}$ factor meets the stopping condition which means the input state is converged to an eigenstate, or when the maximum iteration time is exceeded.

C   Full H_2O Matrix

The Hamiltonian of the water molecule with the H-O-H angle at $104.5°$ and the bond length at 1.0 a.u. is calculated by $STO-3G$ minimal basis using PySCF [22] and chemistry package provided by the Qiskit[16]. The 16-by-16 Hamiltonian of the water molecule used for the local computer's spectral decomposition simulations is as follows [24]. The exponential of the Hamiltonian is done with the MATLAB expm funtion.

$$\mathcal{H}_{H_2O} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -2.594 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -2.654 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -4.583 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2.594 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -4.427 & 0 & 0 & 0 & 0 & 0.054 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -4.529 & 0 & 0 & 0.054 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -5.696 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.654 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.054 & 0 & 0 & -4.529 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.054 & 0 & 0 & 0 & 0 & -4.428 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5.637 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4.583 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5.696 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5.637 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6.085
\end{pmatrix} \tag{25}$$