

Yau's Physics Award 2020

**Constructing General Hamiltonian Ground States on a Quantum Computer
Using the Projected Cooling Sensor Algorithm**

Author: Kenneth Choi

School: Ridgefield High School

Ridgefield, Connecticut, USA

Mentor: Dean Lee

Facility for Rare Isotope Beams and Department of Physics and Astronomy

Michigan State University

East Lansing, Michigan, USA

(Dated: August 2020)

Constructing General Hamiltonian Ground States on a Quantum Computer Using the Projected Cooling Sensor Algorithm

Author: Kenneth Choi

Ridgefield High School, Ridgefield, CT 06877, USA

Mentor: Dean Lee

Facility for Rare Isotope Beams and Department of Physics and Astronomy,

Michigan State University, MI 48824, USA

(Dated: August 2020)

In practice, with accurate algorithms, quantum computers are able to find the properties of complex many-body systems that classical methods cannot examine. However, many quantum algorithms that attempt to reconstruct ground state wave functions have low fidelity and are not robust against noise. To this end, we introduce the projected cooling sensor algorithm, which accurately reconstructs the ground state of any general Hamiltonian, to solve the quantum ground state preparation problem. For low-dimension Hamiltonians, the projected cooling sensor algorithm reconstructs the ground state with a relative error of 0.0001 or less. For high-dimension Hamiltonians, multiple iterations of the projected cooling sensor algorithm exponentially decrease the error of the reconstructed ground state. We find that on a quantum computer, the reconstructed ground state has nearly 100% overlap with the exact ground state. The projected cooling sensor algorithm can be applied to a wide range of general many-body systems, including nuclei, bulk materials, superconductors, and Ising models. When simulated on a quantum computer, the projected cooling sensor algorithm has the potential to achieve quantum supremacy over classical computations for any quantum Hamiltonian.

Keywords: Quantum Many-Body System, Hamiltonian, Ground State Preparation, Quantum Supremacy

CONTENTS

I. Introduction	2
II. Methods	5
A. The Projected Cooling Algorithm and its Limitations	5
B. The Projected Cooling Sensor Algorithm	7
C. Optimizing the Projected Cooling Sensor Algorithm	8
D. Projected Cooling Sensor on a Quantum Computer	10
III. Results	10
IV. Discussion	10
V. Future Work	14
VI. Conclusion	15
Acknowledgments	15
VII. Appendixes	15
A. Delaying Wave Reflection	15
B. Code	17
References	21

I. INTRODUCTION

In recent years, physics and computer science have unified at the exciting intersection of quantum computing. Quantum computing makes use of subatomic properties and would, in theory, revolutionize modern electronics and materials. Today, physicists and computer scientists have successfully created algorithms and built small quantum computers that can solve problems faster than any classical computer. In their seminal work, Arute *et al.* [1] introduced a quan-

tum processor that is able to perform algorithms exponentially faster than a state-of-the-art classical computer. This processor is the first ever to quantum supremacy. However, there is still a need for quantum computers that can describe large-scale systems and perform even more complex computations, such as finding the properties of many-body systems.

While a classical bit can store only one of two definite states (0 or 1), a quantum qubit can be in a superposition of the two states, allowing for an infinite number of distinct qubit values. In the computational setting, quantum superposition allows qubits to hold exponentially more information than classical bits, leading to exponentially faster quantum algorithms. Developing quantum computing devices, however, is complicated by environmental noise that interferes with quantum processes. Due to short decoherence times, gate errors, and readout errors, all quantum computing algorithms currently have difficulty addressing real problems of interest on currently available devices. Although we simulate a quantum algorithm using classical computing, we keep in mind the physical constraints associated with quantum computers.

We direct our focus on the quantum many-body problem, which arises when more than two microscopic particles interact in a system. While it is feasible to measure the static and dynamic properties of one- or two-particle systems, most systems become too complicated to predict when many particles are introduced. The quantum Hamiltonian, which is the Hermitian operator that describes a quantum system, can become too difficult for classical algorithms to de-

scribe in many-body systems. We specifically develop an algorithm to solve the *quantum ground state preparation problem*, which occurs when we try to reconstruct the ground state of the Hamiltonian of a quantum many-body system.

The quantum ground state preparation problem is one of the most significant subsets of the quantum many-body problem. Ground state reconstruction has numerous applications in physics. Kitaev *et al.* [2] showed that all quantum circuits can be modeled using only ground states. Peng *et al.* [3] applied quantum adiabatic evolution, a ground state reconstruction algorithm, to Shor’s algorithm, which is a factoring algorithm that has the potential to break 2048-bit RSA cryptography if physically implemented.

Classical analytical methods of finding the ground state of many-body systems such as Schrödinger’s equation are computationally tedious and do not give accurate reconstructions. To this end, several algorithms that reconstruct the ground state of a Hamiltonian have emerged. Most well-known are quantum phase estimation [4, 5] and quantum adiabatic evolution [6]. Quantum phase estimation estimates the phase, or eigenvalues, of a unitary operator and is often used in conjunction with other quantum algorithms.

The quantum adiabatic evolution algorithm is largely considered the standard ground state reconstruction algorithm. The algorithm initializes a system whose ground state is the solution, then slowly interpolates a simple Hamiltonian to the target system Hamiltonian. Quantum adiabatic evolution has been regarded as the most accurate quantum algorithm for recon-

structing a ground state, and it has been successfully applied to specific quantum problems. For example, Farhi *et al.* [7] applied quantum adiabatic evolution to small random examples of an NP-complete problem. However, Childs *et al.* [8] showed that the accuracy of the quantum adiabatic evolution algorithm suffers when there are extremely small gaps between the instantaneous ground state and the rest of the spectrum, and the algorithm time increases proportional to the inverse of the square of the gap. In addition, increasing the number of gaps increases the number of quantum gates required to complete the algorithm. This decreases the algorithm’s overall fidelity, or the percentage of physical information that retains coherence. Thus, while it provides accurate results, the quantum adiabatic evolution algorithm may not outperform classical computers in practical use.

Lee *et al.* [9] recently introduced the projected cooling algorithm, which also aims to solve the quantum ground state preparation problem. The projected cooling algorithm utilizes a lattice Hamiltonian that acts on a one-dimensional chain of qubits and conserves particle number. The lattice Hamiltonian has a potential centered at the origin with a finite width. When the ground state of the Hamiltonian is the only bound state, all the excited states of the Hamiltonian are driven away from the origin as time evolution occurs. When the algorithm terminates, only the bound state remains near the origin. Figure 1 shows a physical representation of the projected cooling algorithm, where over time, the excited states leave the potential well while the bound state remains.

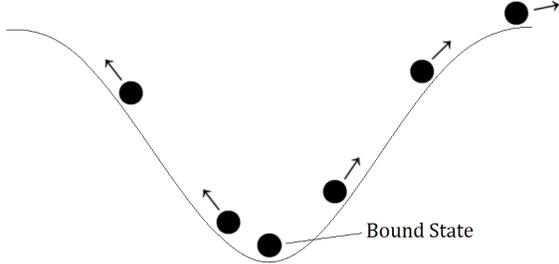


FIG. 1: Potential well in projected cooling.

The bound state of the Hamiltonian remains centered at the origin while the states with kinetic energy are driven away from the origin.

Figure 2 shows that in the same amount of time, the projected cooling algorithm more accurately reconstructs the ground state of the Hamiltonian than does the standard adiabatic evolution algorithm. Each algorithm in Figure 2 is applied to the two linked one-dimensional lattice chains n_1 and n_2 .

Although it provides accurate results, the projected cooling algorithm has several limitations. The projected cooling algorithm is most accurate in a system with large volume because it takes more time for the reflected wave to return back to the interior region near the origin. However, real world systems have finite volume; thus, to increase the volume is an unreliable method to decrease noise interference.

In preliminary experiments, we find that delaying wave reflection is impractical because the entropy of the returning wave cannot be scaled down. See Appendix A for additional information on wave delay in the projected cooling algorithm.

The projected cooling algorithm also uses two-qubit gates, which decrease the system’s fidelity. For every step of time evolution that occurs in the projected cooling algorithm, a two-

qubit gate is applied. While single-qubit gates have achieved greater than 99.9% fidelity using isotopically enriched silicon, Huang *et al.* [10] showed that two-qubit gates have yet to achieve fidelities greater than 98%. Thus, a large number of two-qubit gates cannot be implemented without substantial error. To decrease the noise, the number of two-qubit gates in the algorithm must be limited. This problem is universal in almost every quantum ground state reconstruction algorithm.

We focus on the main problem that limits the projected cooling algorithm’s applicability: the algorithm provides accurate results only for a Hamiltonian with a localized ground state. For the ground state to be localized, its wave function must be limited in spatial size. To generalize the projected cooling algorithm, we develop the *projected cooling sensor algorithm*, which accurately calculates the ground state of any general Hamiltonian H_{obj} by coupling H_{obj} to a circular lattice reservoir and performing projected cooling on the reservoir. We find that the projected cooling sensor algorithm is able to accurately reconstruct the ground state for a large volume and sufficient time period. In addition, we find that running the projected cooling algorithm multiple times, taking each run’s initial state to be equal to the previous run’s final state, provides the same accuracy as running the projected cooling sensor algorithm once for a long time period. The algorithm is able to achieve exponential convergence to the ground state. Finally, it acts as a quantum sensor that detects the negative energies of an object Hamiltonian, and the algorithm can be applied to many-body

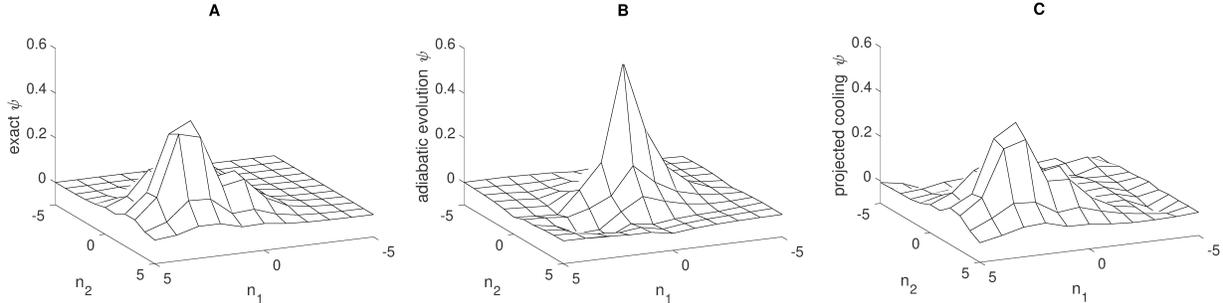


FIG. 2: Comparison between adiabatic evolution and projected cooling. Figure from Lee *et al.* [9]. Graph A shows the exact ground state wave function. Graph B shows the reconstructed ground state after 40 time steps of the adiabatic evolution algorithm. Graph C shows the reconstructed ground state after 40 time steps of the projected cooling algorithm. Each lattice has 25 lattice sites, with the interior region spanning 10 consecutive sites.

systems whose ground states are localized or uniform throughout space.

II. METHODS

We describe the projected cooling algorithm and discuss its limitations in solving the quantum ground state preparation problem. Then, we introduce the projected cooling sensor algorithm, which couples any Hamiltonian to a lattice reservoir and performs projected cooling on the reservoir to reconstruct the ground state of the Hamiltonian. We optimize the projected cooling sensor algorithm for higher-dimension Hamiltonians and describe its implementation as a quantum circuit.

A. The Projected Cooling Algorithm and its Limitations

In the projected cooling algorithm, we begin with a one-dimensional circular lattice of $2L + 1$ qubits with sites at $n = -L, \dots, L$. We define the vacuum as the tensor product state where

all qubits are $|0\rangle$, which means that if there are no particles on the lattice, then all qubits are in the state $|0\rangle$. If a qubit n is in the state $|1\rangle$, then we are certain that there is a particle at site n .

We define a lattice Hamiltonian H that has a translationally-invariant kinetic energy and conserves particle number. By definition, $H = K + V$, where K is the kinetic energy term and V is the potential energy term. We define $\langle [n'] | H | [n] \rangle = K_{n',n} + V_n \delta_{n',n}$, where $K_{n',n} = \delta_{n',n} - \frac{1}{2} \delta_{n',n+1} - \frac{1}{2} \delta_{n',n-1}$ and V_n is the single-particle potential energy at site n . In the standard projected cooling model, $V_n = -\delta_{0,n}$, which is an attractive Kronecker delta function at $n = 0$. Using the attractive function at the origin causes the ground state to stay bound to the initial site as time evolution of the system occurs.

We define ρ to be the compact region over qubits $n = -R, \dots, R$, where $R \ll L$, and we define the projection operator P to project $[[n]]$ onto the subspace where ρ contains all particle excitations. Thus, $P[[n]] = 0$ for $|n| > R$, and $P[[n]] = |[n]\rangle$ for $|n| \leq R$.

By definition, $H|\psi\rangle = E|\psi\rangle$, where the $|\psi\rangle$

vectors are the eigenstates of H and the E values are the energy eigenvalues that correspond to each eigenstate. Let $|\psi_0\rangle$ be the ground state of H . $|\psi_0\rangle$ must be a localized bound state and the only bound state of H ; i.e., $|\psi_0\rangle$ must correspond to the lowest energy eigenvalue, E_{min} , of the Hamiltonian. We define $U(t) = e^{-iHt}$ as the time evolution operator.

We use dimensionless units for all quantities and set $\hbar = 1$. As $L \rightarrow \infty$, the projected time evolution operator $PU(t)P$ has a stable fixed point that is proportional to $P|\psi_0\rangle$. Therefore, as we perform time evolution on $P|\psi_I\rangle$, where $|\psi_I\rangle$ is the initial state, all excited states are driven out of ρ , and only the ground state $|\psi_0\rangle$ remains. We note that we assume $|\psi_I\rangle$ is not orthogonal to $|\psi_0\rangle$ so that the wave function at the origin does not converge to zero.

Let $O(t)$ to be the normalized overlap between the reconstructed ground state and the exact ground state over ρ . Figure 3 tracks $O(t)$ over time for the projected cooling algorithm, where each curve begins at a random initial state. As t increases, $O(t)$ approaches 1, which demonstrates the fixed-point behavior of the ground state. Throughout the algorithm, only the ground state remains bound.

Figure 4 shows an example of the projected cooling algorithm applied to the initial state $|\psi_I\rangle = [1, 0, \dots, 0]^T$ for lattice size $L = 100$ and potential $V_0 = -1$.

For a Hamiltonian with more than one bound state, fixed-point behavior does not occur as the lattice size increases. To solve this, we operate a time-dependent Hamiltonian $H(t)$ on the initial state. We also multiply the kinetic energy

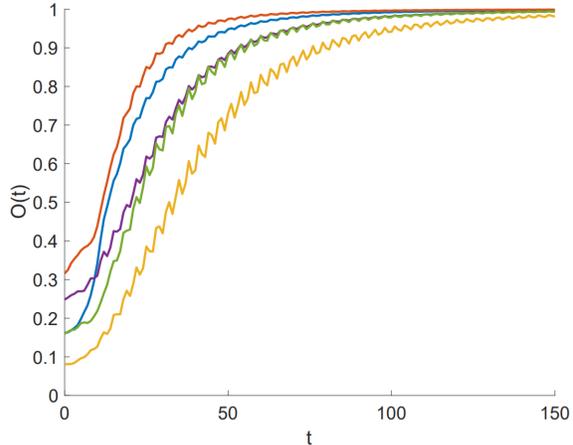


FIG. 3: Stable fixed-point behavior in projected cooling. Figure from Lee *et al.* [9]. As time t increases, the normalized overlap $O(t)$ between the reconstructed ground state and the exact ground state over ρ approaches 1, which demonstrates the fixed-point behavior of the ground state.

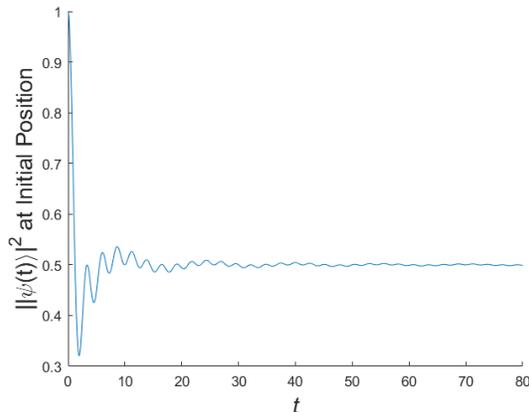


FIG. 4: Convergence to the ground state in projected cooling. $L = 100$, $V_0 = -1$, and $|\psi_I\rangle = [1, 0, \dots, 0]^T$. As time evolution occurs, all excited states are driven away from the localized region, leaving only the bound state $|\psi_0\rangle$.

operator by a factor greater than 1 to prevent sinusoidal oscillations in the expectation values of operators that do not commute with H . This causes only the ground state to remain bound. As time evolution occurs, the time-evolved state

$|\psi(t)\rangle$ converges to the ground state $|\psi_0\rangle$ of $H(t)$. Once $|\psi(t)\rangle$ and $|\psi_0\rangle$ have a sufficient overlap, we can use adiabatic evolution on the system to evolve $|\psi(t)\rangle$ to $|\psi_0\rangle$.

The projected cooling algorithm is most applicable in nuclear physics, namely self-bound systems such as atomic nuclei. In atoms and molecules, electrons are localized in orbitals, so projected cooling can be used to find the ground state of the system. However, the projected cooling algorithm's main limitation is its inability to find the ground state of a general Hamiltonian that does not conserve particle number and/or whose ground state is uniform throughout space. For example, projected cooling cannot be applied to find the ground state of electrons in a conductor because the electrons are spread throughout the conductor. In addition, gas and liquid particles have uniform ground states and are not applicable to the projected cooling algorithm.

Thus, we introduce the projected cooling sensor algorithm, which generalizes projected cooling to find the ground state of any Hamiltonian, including those of Ising ferromagnets and conductors. The projected cooling sensor algorithm gives a generalized solution to the quantum ground state preparation problem.

B. The Projected Cooling Sensor Algorithm

The projected cooling sensor algorithm consists of two connected parts: the Hamiltonian of interest and the one-dimensional projected cooling reservoir. We define a $D \times D$ general Hamiltonian H_{obj} whose ground state is not necessarily localized. H_{obj} is linked to a circular one-

dimensional lattice, called the reservoir, which contains one particle. The reservoir in the projected cooling sensor algorithm is analogous to the lattice in the projected cooling algorithm.

The original projected cooling algorithm includes only the lattice as part of its system. This makes finding the uniform ground state of a general Hamiltonian impossible because the Hamiltonian cannot be scaled while maintaining projected cooling. Instead of finding the ground state of the lattice Hamiltonian, the projected cooling sensor algorithm finds the ground state of H_{obj} coupled to the reservoir. Thus, H_{obj} can be scaled without disrupting projected cooling in the lattice reservoir.

As in the projected cooling algorithm, we use dimensionless units for all quantities and set $\hbar = 1$ in our discussion of the projected cooling sensor algorithm. Let the reservoir Hamiltonian be represented by H_{res} , and let the corresponding reservoir eigenstates be represented by $|E_{\text{res}}\rangle$. We define

$$P_{0,1} = \frac{1}{2} (|n=0\rangle + |n=1\rangle) (\langle n=0| + \langle n=1|) \quad (1)$$

as the operator that projects onto the $n=0$ and $n=1$ lattice sites. The projection operator includes the $n=1$ site in addition to $n=0$ to prevent the formation of localized states above the energy spectrum at momentum $k = \pi$. After H_{obj} is coupled to the reservoir, the reservoir potential V_{res} is equal to $H_{\text{obj}} \otimes P_{0,1}$. Figure 5 shows a representation of the coupling.

By definition, the Hamiltonian is the sum of the kinetic and potential energies. Therefore, the

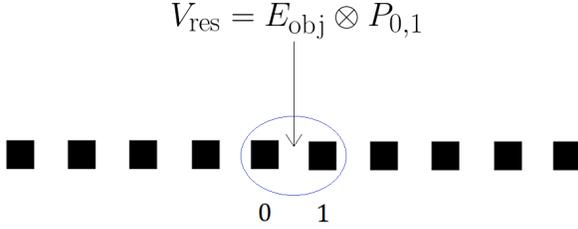


FIG. 5: Coupling representation in projected cooling sensor algorithm. H_{obj} is coupled to the reservoir so that the potential on the reservoir V_{res} is equal to the tensor product $E_{\text{obj}} \otimes P_{0,1}$.

total system Hamiltonian has the form

$$H_{\text{tot}} = K_{\text{res}} + H_{\text{obj}} \otimes P_{0,1}, \quad (2)$$

where K_{res} is the kinetic energy of the reservoir. The eigenstates of the system are equal to

$$|E_{\text{tot}}\rangle = |E_{\text{obj}}\rangle \otimes |E_{\text{res}}\rangle_{V_{\text{res}}=E_{\text{obj}} \otimes P_{0,1}}. \quad (3)$$

When $E_{\text{obj}} < 0$, we find that the reservoir ground state, $|E_{\text{res}}^0\rangle_{V_{\text{res}}=E_{\text{obj}} \otimes P_{0,1}}$, is localized. Thus, the projected cooling sensor algorithm acts as a negative energy sensor: the excited states on the lattice are driven away from the origin while the bound state remains when a negative potential is detected.

For the case where H_{obj} has one ground state, we shift H_{obj} by a constant so that only the ground state has a negative energy. However, when running the projected cooling sensor algorithm, localized states can form above the energy spectrum if E_{obj} is too large and positive. Thus, we rescale H_{obj} so that $E_{\text{max}} \leq 1$, where E_{max} is the greatest energy eigenvalue of H_{obj} . Shifting and rescaling H_{obj} do not change its eigenstates. Figure 6 depicts the energy spectrum of H_{obj} after we perform shifting and scaling.

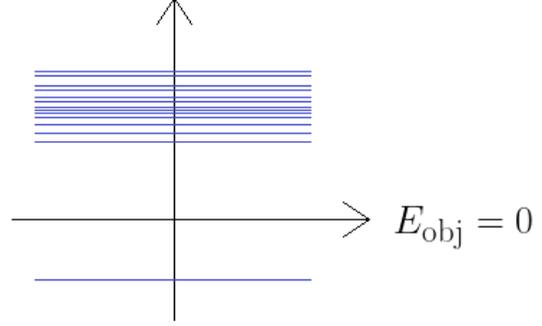


FIG. 6: Energy spectrum in projected cooling sensor. We shift H_{obj} so that its lowest energy eigenvalue is negative. We also scale H_{obj} so that all the other energy eigenvalues are positive.

To run the projected cooling sensor algorithm, we initialize the reservoir with a random state at $n = 0$ and $n = 1$. To achieve a faster rate of convergence to the exact ground state, variational methods can first be applied to obtain an initial state corresponding to an energy numerically close to the ground state energy.

C. Optimizing the Projected Cooling Sensor Algorithm

Let O represent the measured overlap between the exact ground state $|E_{\text{min}}\rangle$ and the resultant ground state $|v_{\text{PCS}}\rangle$ from the projected cooling sensor algorithm. Then,

$$O = \frac{\langle E_{\text{min}} | v_{\text{PCS}} \rangle^2}{\| |E_{\text{min}} \rangle \| \| |v_{\text{PCS}} \rangle \|}. \quad (4)$$

If the projected cooling sensor algorithm perfectly reconstructs the ground state, then $O = 1$. We also define the log error of the resultant ground state as $\ln(1 - O)$.

As time evolution occurs, the continuum state probability at the initial lattice sites is propor-

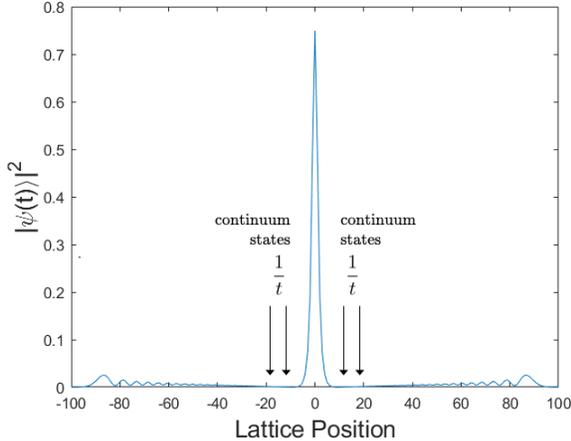


FIG. 7: Continuum state decay in the projected cooling sensor algorithm. The continuum states leave the initial sites at a rate proportional to $\frac{1}{t}$. As $t \rightarrow \infty$, the continuum states are driven away at a lesser rate.

tional to $\frac{1}{t}$. As $t \rightarrow \infty$, the rate at which the excited states leave the initial sites decreases. Figure 7 shows graphically that the propagation of the continuum states leaving the origin decreases in amplitude over time. Thus, for high-dimension Hamiltonians that require long run times, we instead run the projected cooling sensor algorithm multiple times for short time periods, using the final state of each previous run as the initial state of the next run. We find that this causes an exponential increase in the convergence to the exact ground state. For a random 30×30 Hermitian matrix as H_{obj} , we calculate $O = 0.2022$ after the first iteration, $O = 0.6386$ after the fifth iteration, and $O = 0.8609$ after the tenth iteration.

We also find that scaling H_{obj} to have multiple negative energies then running the algorithm multiple times causes the projected cooling sensor algorithm to converge to the ground state quicker than the case where H_{obj} has only one

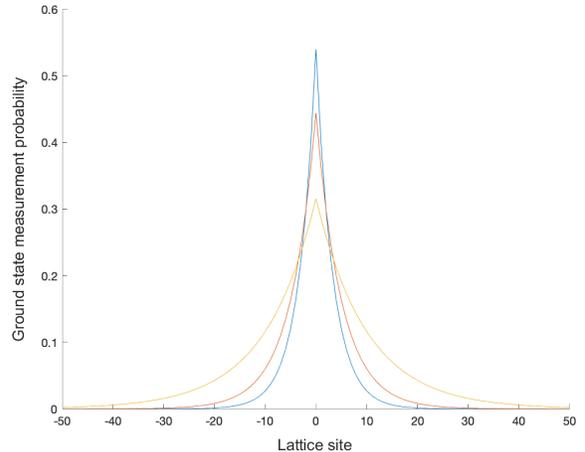


FIG. 8: Multiple bound states in the projected cooling sensor algorithm. Using multiple bound states in the projected cooling sensor algorithm amplifies the measurement probability of the ground state in the interior region.

negative energy. Multiple negative energy states are analogous to a booster seat; they increase the potential strength at the initial sites, which, in turn, amplifies the ground state of H_{obj} . This is especially useful when simulating the projected cooling sensor algorithm on a quantum computer because the ground state probability must be nontrivial to measure the particle at the initial sites. Figure 8 shows multiple bound states in the reservoir. Although including multiple bound states increases the convergence rate, we use only one bound state in our simulations. The case where $H_{r\text{mobj}}$ has multiple bound states should be addressed in future works.

Setting $m = 1$ and $E_{\text{max}} = 0.5$, we perform the projected cooling sensor algorithm once for σ_z and once for an arbitrary 2×2 H_{obj} . Then, we run the projected cooling sensor algorithm for 30 iterations for a 30×30 arbitrary H_{obj} , a 50×50 arbitrary H_{obj} , and a 100×100 arbitrary H_{obj} .

We find that the projected cooling sensor algorithm accurately reconstructs the ground state of any general Hamiltonian. We also find that running the projected cooling sensor algorithm for multiple iterations decreases the ground state reconstruction error, $1 - O$, exponentially.

D. Projected Cooling Sensor on a Quantum Computer

Although we are limited to performing classical simulations of the projected cooling sensor algorithm, we also examine the implementation of the algorithm on a quantum computer. We define the unitary gate

$$u_{0,1} = e^{i\frac{\pi}{8}(\sigma_x\sigma_y - \sigma_y\sigma_x)} \quad (5)$$

at sites $n = 0$ and $n = 1$ to rotate the qubits at both sites to $n = 0$. Then, we couple H_{obj} to $n = 0$. We finally apply the conjugate transpose gate $u_{0,1}^\dagger$ to rotate the qubits back to both sites $n = 0$ and $n = 1$. This process removes the localized states above the energy spectrum because the eigenstate at momentum $k = \pi$ cancels out with the projection operator. Figure 9 depicts the projected cooling sensor algorithm repeated twice on a quantum circuit.

III. RESULTS

We perform the projected cooling sensor algorithm on object Hamiltonians of varying dimensions. We set $m = 1$ and $E_{\text{max}} = 0.5$. Figures 10 and 11 show the convergence of $\ln(|v_{PCS}|^2)$ for each state vs. time for one run of the projected cooling sensor algorithm for σ_z and an

arbitrary 2×2 H_{obj} . For both simulations, we find that $O = 1.0000$. Thus, the projected cooling sensor algorithm accurately reconstructs the ground state of any 2×2 H_{obj} .

Figures 13, 14, and 15 show the log error, $\ln(1 - O)$, of the resultant ground state for multiple iterations of the projected cooling sensor algorithm, where $L = 100$, for an arbitrary 30×30 H_{obj} , an arbitrary 50×50 H_{obj} , and an arbitrary 100×100 H_{obj} , respectively. We find that the log error of the reconstructed ground state for any arbitrary $D \times D$ H_{obj} decreases linearly every iteration. The log errors decrease at average rates of -0.1203 , -0.1259 , and -0.0710 for an arbitrary 30×30 H_{obj} , an arbitrary 50×50 H_{obj} , and an arbitrary 100×100 H_{obj} , respectively. Thus, the error decreases exponentially.

Figure 16 shows the log error of the resultant ground state for multiple iterations of the projected cooling sensor algorithm for an arbitrary 100×100 H_{obj} , where $L = 200$. Again, the error decreases exponentially.

IV. DISCUSSION

We successfully generalize the projected cooling algorithm to any Hamiltonian in the projected cooling sensor algorithm. The projected cooling sensor algorithm provides a solution to the quantum ground state preparation problem. For low-dimensional arbitrary Hamiltonians, Figures 10 and 11 show that the projected cooling sensor algorithm can reconstruct the ground state of any general 2×2 H_{obj} , including σ_z , in a short time period. In addition, because an arbitrary 2×2 H_{obj} is a small-scale system,

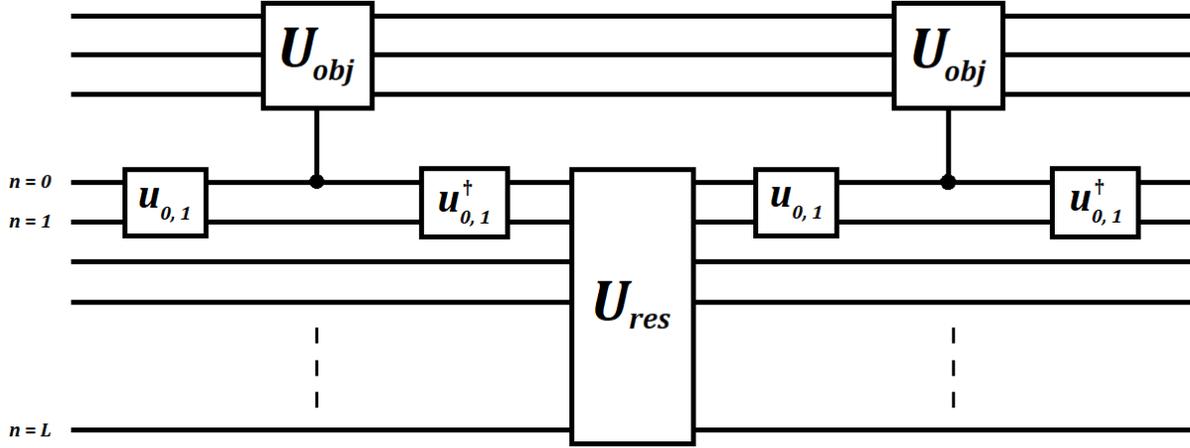


FIG. 9: Projected cooling sensor as a quantum circuit. The projected cooling sensor algorithm is repeated twice on a quantum circuit. Gates $u_{0,1}$ and $u_{0,1}^\dagger$ remove localized states that form above the energy spectrum.

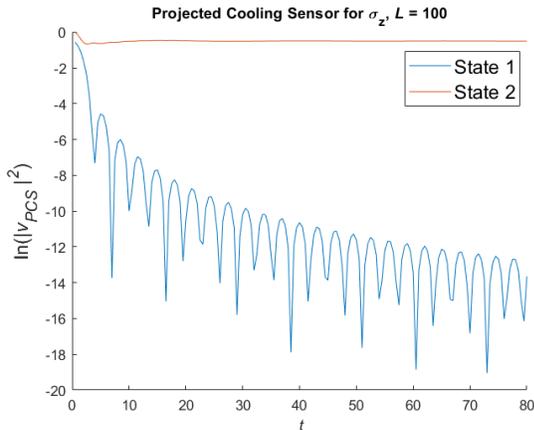


FIG. 10: Projected cooling sensor for Pauli-Z. We run one iteration of the projected cooling sensor algorithm for σ_z . $L = 400$. $O = 1.0000$. State 2 (red) corresponds to the spin-down state, which is the ground state of σ_z .

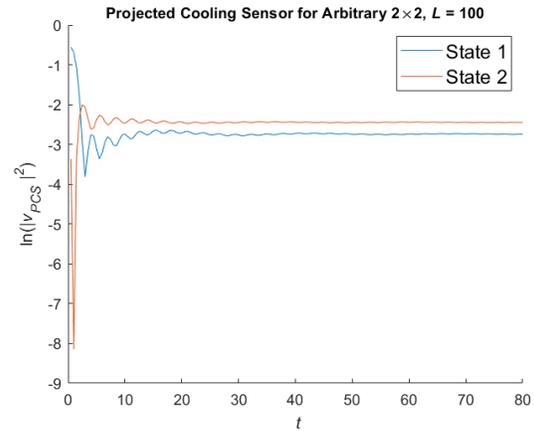


FIG. 11: Projected cooling sensor for arbitrary 2×2 Hamiltonian. We run one iteration of the projected cooling sensor algorithm for an arbitrary 2×2 H_{obj} . $O = 1.0000$. The ground state is a linear combination of State 1 (blue) and State 2 (red).

the projected cooling sensor algorithm requires only 3 iterations to reconstruct the ground state of the Hamiltonian with a relative error of 0.0001 or less. Thus, multiple iterations of the projected cooling sensor algorithm for an arbitrary 2×2 H_{obj} are unnecessary. This reduces the need to apply a large number of two-qubit gates or run

the algorithm for a long time period.

For higher-dimensional arbitrary Hamiltonians, we find in Figure 12 that running the projected cooling sensor algorithm only once for an arbitrary 30×30 Hamiltonian yields inaccuracy in ground state reconstruction. We obtain a final

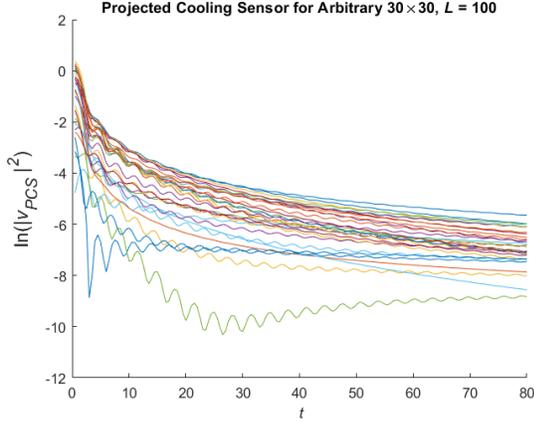


FIG. 12: Projected cooling sensor for arbitrary 30×30 Hamiltonian. We run one iteration of the projected cooling sensor algorithm for an arbitrary 30×30 H_{obj} . $O = 0.5617$. The ground state is a linear combination of the 30 states shown on the graph.

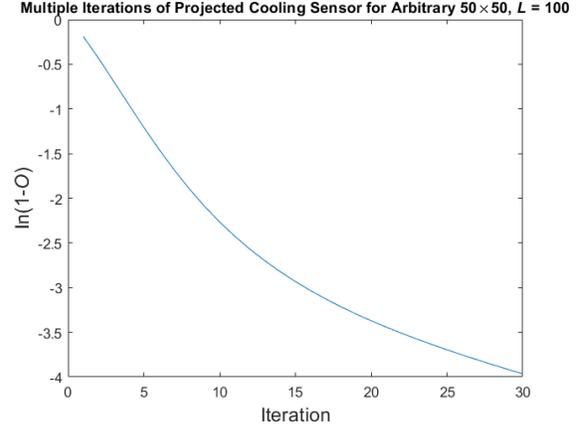


FIG. 14: Multiple iterations of projected cooling sensor for arbitrary 50×50 Hamiltonian. We show the log error of the ground state calculated by 30 iterations of the projected cooling sensor algorithm for an arbitrary 50×50 H_{obj} . $L = 100$.

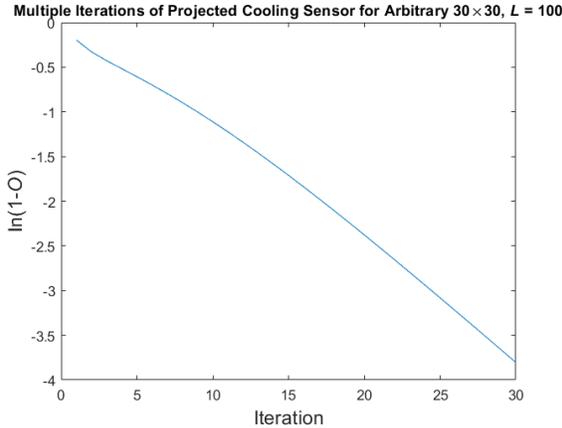


FIG. 13: Multiple iterations of projected cooling sensor for arbitrary 30×30 Hamiltonian. We show the log error of the ground state after 30 iterations of the projected cooling sensor algorithm for an arbitrary 30×30 H_{obj} . $L = 100$.

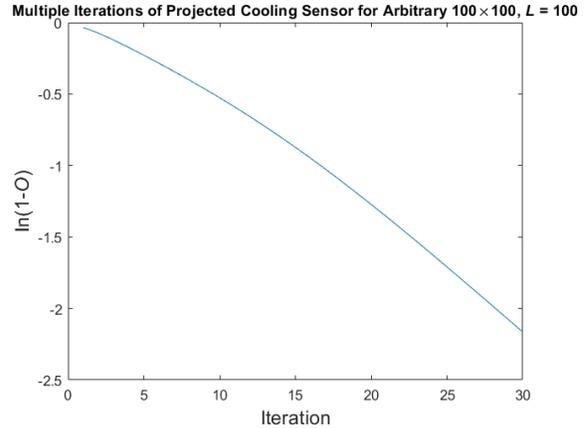


FIG. 15: Multiple iterations of projected cooling sensor for arbitrary 100×100 Hamiltonian. We show the log error of the ground state calculated by 30 iterations of the projected cooling sensor algorithm for an arbitrary 100×100 H_{obj} . $L = 100$.

overlap of $O = 0.5617$. We solve this problem by running the algorithm multiple times in a row. Figures 13, 14, and 15 show that running the projected cooling algorithm for multiple iterations decreases the calculated ground state error ex-

ponentially. In Figure 16, we find that doubling the lattice size L from 100 to 200 for an arbitrary 100×100 H_{obj} increases the rate of convergence to the ground state. However, it is important to note that an increase in the lattice size leads to

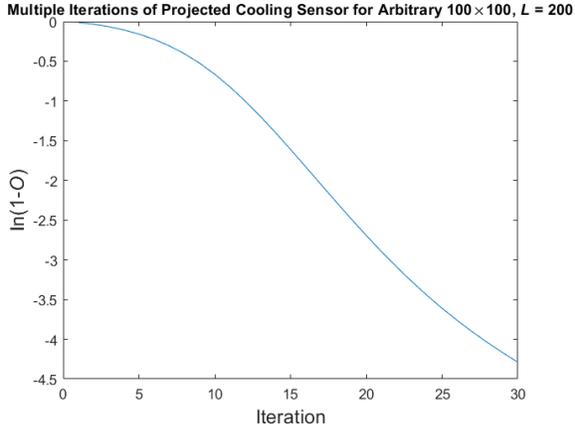


FIG. 16: Multiple iterations of projected cooling sensor for arbitrary 100×100 Hamiltonian. We show the log error of the ground state calculated by 30 iterations of the projected cooling sensor algorithm for an arbitrary 100×100 H_{obj} . $L = 200$.

an increase in the time complexity of the simulation. In addition, on a quantum computer, it is necessary to limit the number of two-qubit gates in the algorithm to maintain the system coherence; thus, running the projected cooling algorithm many times in a row is not ideal.

The projected cooling sensor algorithm is the first ground state reconstruction algorithm to accurately improve an initial wave function systematically. We can start with any random initial state and converge to the ground state without pre-optimization. However, large-scale systems that do not start with a reasonable initial state either take too long to converge or never converge. In practice, classical or quantum variational methods, such as the methods introduced by Peruzzo *et al.* [11] and Dumitrescu *et al.* [12], can be utilized to calculate an initial state guess instead of starting with a random initial state.

The projected cooling sensor algorithm’s exponential convergence to the ground state is sim-

ilar to the Euclidean time projection e^{-Ht} , which is widely used in simulating quantum systems with classical computers. However, in quantum computing, the results of the projected cooling sensor algorithm are probabilistic. Even though we begin with multiple bound states to amplify the potential magnitude at the initial sites, we must still repeat the algorithms for many iterations to achieve the desired nontrivial quantum measurement.

The biggest issue with the projected cooling sensor algorithm lies in the probabilistic nature of its quantum simulation. When we calculate the ground state of a Hamiltonian on a quantum computer, we must perform a measurement on the entire system. Before we start the algorithm, all eigenstates are at the origin, so measuring the system yields a 100% probability of finding a particle at the origin. However, after one iteration of the projected cooling algorithm, most of the excited states leave the origin. When we measure the system again, the probability of measuring the particle at the origin is instead equal to the squared norm of the final ground state, $||v_{PCS}||^2$, which is less than 100%. Thus, for every iteration of the projected cooling sensor algorithm, the ground state is more accurately reconstructed at the origin, but the probability of measuring the ground state on a quantum computer decreases. We find that after 30 iterations of any arbitrary H_{obj} , the measurement probability decreases to approximately 1%. Future research should focus on developing methods to optimize the projected cooling sensor algorithm so that there is a balance between the reconstruction accuracy and the measurement probability. Future algorithms

should improve the accuracy of the projected cooling sensor algorithm so that there is no need for multiple iterations. This would optimally increase the measurement probability to at least 3%.

The potential applications of the projected cooling sensor algorithm are very wide. The ground state preparation problem is prevalent in any many-body system. Because the projected cooling sensor algorithm is able to find the ground state of any arbitrary many-body system, it can be applied to finding the ground state of electrons in superconductors, examining the properties of electron orbitals such as spin, calculating the ground state of an atomic nucleus, finding the ground state of an Ising spin model, and other ground state reconstruction situations. From computer science to physical science, the projected cooling sensor algorithm is successfully able to determine the ground state properties of any general system.

V. FUTURE WORK

In the future, we plan to employ multiple bound states in the projected cooling sensor algorithm. This would amplify the ground state measurement probability at the initial sites, which would increase the algorithm's success rate when run on a quantum computer. We plan to choose the shift value for H_{obj} such that the bottom quarter of the spectrum has negative energy. In preliminary experiments, we achieve ten times faster convergence.

We also consider applying the Trotter-Suzuki approximation to the reservoir Hamiltonian to

increase the fidelity of the two-qubit gates. Introduced by Trotter [13], the Trotter-Suzuki approximation decomposes a time-dependent Hamiltonian into multiple parts. Letting H_A and H_B be the Trotterized components of the Hamiltonian, where $H = H_A + H_B$, the time evolution operator of the projected cooling sensor algorithm would be equal to $U(t) = e^{-iH_A t} e^{-iH_B t} e^{-iH_{obj} t}$. Using the Trotter-Suzuki approximation to improve the fidelity of two-qubit gates would help to maintain the system coherence when the projected cooling sensor algorithm is run multiple times on a quantum computer. In addition, exponentiating the overall Hamiltonian can be costly on a quantum computer. By Trotterizing the reservoir Hamiltonian, the resultant computational complexity is less than that if the Hamiltonian itself were exponentiated.

We consider several open questions. With its exponential convergence to the ground state, the projected cooling sensor algorithm has the potential to achieve quantum supremacy over classical calculations for some general quantum Hamiltonian. We plan to optimize the algorithm more and investigate whether it can achieve quantum supremacy.

We also ask whether we can achieve polynomial computational scaling with the size of any quantum many-body system. This would yield a remarkable advancement in quantum computing by solving NP problems in polynomial time. Whether polynomial computational scaling with the system is possible is less clear than whether the projected cooling sensor algorithm can achieve quantum supremacy, but we plan to

investigate the time complexity and optimize the algorithm further.

VI. CONCLUSION

We solve the quantum ground state preparation problem by developing the projected cooling sensor algorithm, which accurately reconstructs the ground state of any general Hamiltonian. The projected cooling sensor algorithm is the first method to accurately improve a random starting wave function systematically to reconstruct a ground state. The projected cooling sensor algorithm generalizes the projected cooling algorithm, which is already faster and more accurate than the standard adiabatic evolution algorithm, to quantum many-body systems that may not necessarily have localized ground states. Furthermore, when the projected cooling sensor algorithm is run for multiple iterations, the error in ground state reconstruction decreases at an exponential rate. The projected cooling sensor algorithm can be applied to any general many-body system in physical and computer science such as a superconductor or an atomic nucleus, and it has the potential to achieve quantum supremacy over classical computations for any general quantum Hamiltonian. Future research will investigate the potential for the projected cooling sensor algorithm to achieve quantum supremacy and will optimize the algorithm to yield a higher ground state measurement probability.

ACKNOWLEDGMENTS

I completed original research independently under my mentor the past summer. I would like to extend my deepest gratitude to my mentor, Dr. Dean Lee, for proposing the research topic and guiding me through the research process for the past two months. I would also like to thank Dr. John Rickert for teaching me the fundamentals of scientific writing and providing me invaluable feedback throughout the research period. My thanks also go to Dr. Jenny Sendova, Boris Zbarsky, Emma Tan, and Laia Xiao Planas for revising and providing feedback on my paper. I would also like to thank my parents for providing me support and encouragement. I also thank the Center for Excellence in Education as part of the 2020 Research Science Institute for organizing the research period and providing me essential resources and technology. Finally, I would like to thank the U.S. Department of Defense for making my research possible.

VII. APPENDIXES

Appendix A: Delaying Wave Reflection

For wave delay to be worth employing in the projected cooling algorithm, it is not enough to simply delay the return of the wave packet. We must also scale the wave packet down so that it does not return to the interior region after leaving. In preliminary experiments, we find that delaying wave reflection back into the lattice is possible by modifying the construction of the Hamil-

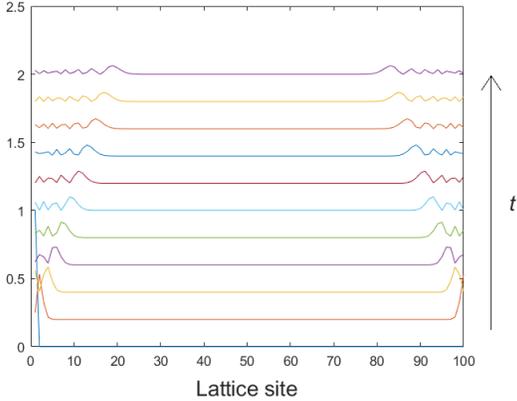


FIG. 17: Time series for wave packet. The time series is shown for a wave packet leaving the origin. $L = 100$, $m(0) = 0.5$, $\epsilon = 0.1$.

tonian. We define a new lattice Hamiltonian

$$H_{ij}^{\text{new}} = H_{ij} * \frac{1}{(m(0) + \epsilon|x|)}, \quad (\text{A1})$$

where H_{ij} is the original lattice Hamiltonian, $m(0)$ is the mass at the origin, ϵ is some arbitrarily small value, and x is the distance from the point in between i and j to the origin. Scaling the Hamiltonian in this way effectively increases the mass as the wave reaches the outer peripheries of the volume, which decreases the group velocity of the wave packet.

We find the group velocity for each set of parameters by calculating the rate the edge of the wave packet expands through its time series, as in Figure 17. We find that as $m(0)$ and ϵ increase, the rate of edge expansion decreases. Thus, we find that increasing the mass of the particles as they are driven away from the interior region decreases the group velocity of the particles as they approach the opposite side of the lattice.

In theory, this method will work if we are also able to decrease the magnitude of the wave

packet as we slow it down. However, we find that scaling down the wave reflection is ultimately impossible due to the high entropy of the wave. Instead of slowing down and completely stopping, the delayed wave eventually reenters the interior region with as much energy as it would have had if it were not delayed. This makes delaying wave reflection virtually useless.

Appendix B: Code

Projected Cooling Sensor Algorithm

```
1 function [first_initial_overlap,overlaps] = projected_cooling_sensor(L,D, ...
2     maxeig,nrepeat,H_obj)
3 %% PROJECTED_COOLING_SENSOR performs the projected cooling sensor algorithm
4 %% multiple times on a DxD Hamiltonian over a lattice of size L. The outputs
5 %% are (1) the initial overlap with the randomized state and (2) the overlaps
6 %% after each iteration of the algorithm.
7     % L is the reservoir size.
8     % D is the dimension of the Hamiltonian.
9     % maxeig is the maximum energy eigenvalue.
10    % nrepeat is the number of iterations the algorithm is performed.
11    % H_obj is the Hamiltonian of interest. Set H_obj = 0 for arbitrary
12    % DxD Hamiltonians.
13
14 %% Set the run time proportional to L, and break each time step dt = 0.5 into
15 %% 10 subdivisions.
16 dt = 0.5;
17 Lt = floor(L/dt*0.8);
18 subdivisions = 10;
19
20 %% Define the reservoir Hamiltonian H_res.
21 r = [0:L-1];
22 H_res = - sparse(mod(r+1,L)+1,r+1,0.5) - sparse(mod(r-1,L)+1,r+1,0.5);
23
24 %% Create arbitrary Hamiltonian if H_obj is not previously defined.
25 if H_obj == 0
26     H_obj = rand(D,D) + i*rand(D,D);
27     H_obj = 0.5*(H_obj + H_obj');
28 end
29
30 %% Shift and scale H_obj so that the maximum energy of H_obj_new is maxeig.
31 list = sort(eig(H_obj));
32 H_object_new = H_obj - (list(1)+list(2))/2*eye(D);
33 scale = max(eig(H_object_new));
34 H_object_new = H_object_new/scale*maxeig;
35
36 %% Define system Hamiltonian H as the tensor product between H_res and H_obj_new.
37 H = sparse(D*L,D*L);
38 for ii = 0:D-1
39     H(ii*L+[0:L-1]+1,ii*L+[0:L-1]+1) = H_res;
40 end
41 for jj = 0:D-1
```

```

42     for ii = 0:D-1
43         H(ii*L+1, jj*L+1) = H(ii*L+1, jj*L+1) + H_object_new(ii+1, jj+1)/2.0;
44         H(ii*L+2, jj*L+1) = H(ii*L+2, jj*L+1) + H_object_new(ii+1, jj+1)/2.0;
45         H(ii*L+1, jj*L+2) = H(ii*L+1, jj*L+2) + H_object_new(ii+1, jj+1)/2.0;
46         H(ii*L+2, jj*L+2) = H(ii*L+2, jj*L+2) + H_object_new(ii+1, jj+1)/2.0;
47     end
48 end
49
50 %% Calculate the exact ground state to check for overlap later.
51 [vv, dd] = eig(H_obj);
52 [~, ord] = sort(diag(dd));
53 index = find(abs(vv(:, ord(1))) == max(abs(vv(:, ord(1))))) ;
54 vv_exact = vv(:, ord(1))/vv(index, ord(1));
55
56 %% Define the random values of the initial state.
57 vobj_init = zeros(D,1);
58 for ii = 0:D-1
59     vobj_init(ii+1) = (rand-0.5) + i*(rand-0.5);
60 end
61 v_init = zeros(D*L,1);
62
63 %% Perform the projected cooling sensor algorithm nrepeat times.
64 for ntrial = 1:nrepeat
65
66     %Define initial state v_init using random values from vobj_init.
67     for ii = 0:D-1
68         v_init(ii*L+1) = vobj_init(ii+1,1);
69         v_init(ii*L+2) = vobj_init(ii+1,1);
70     end
71
72     %Perform time evolution for first time step dt.
73     psil(:,1) = exponentiate(v_init, H, -i*dt, subdivisions);
74
75     %Perform time evolution for remaining time steps.
76     for nt = 1:Lt
77         psil(:,nt+1) = exponentiate(psil(:,nt), H, -i*dt, subdivisions);
78     end
79
80     %Graph the convergence of the state(s) vs. time.
81     figure(ntrial)
82     hold on
83     for ii = 0:D-1
84         plot(dt*[1:Lt], log(abs(psil(ii*L+1,1:Lt)+psil(ii*L+2,1:Lt)).^2))
85     end
86     xlabel('\it t', 'FontSize', 14)

```

```

87     ylabel('ln(|\it|v_{PCS}|^2)','FontSize',14)
88     hold off
89
90     %Find the reconstructed ground state and display with trial number.
91     v_PC = zeros(D,1);
92     for ii = 0:D-1
93         v_PC(ii+1,1) = psil(ii*L+1,Lt) + psil(ii*L+2,Lt);
94     end
95     index = find(abs(v_PC) == max(abs(v_PC)));
96     v_PC = v_PC/v_PC(index);
97     disp(ntrial)
98     disp([v_PC vv_exact])
99
100    %Calculate initial and final overlap values and display for each trial.
101    initial_overlap(ntrial,1) = (abs(vv_exact'*vobj_init))^2 ...
102        /((vv_exact'*vv_exact)*(vobj_init'*vobj_init));
103    final_overlap(ntrial,1) = (abs(v_PC'*vv_exact))^2 ...
104        /((vv_exact'*vv_exact)*(v_PC'*v_PC));
105    disp(initial_overlap(ntrial,1))
106    disp(final_overlap(ntrial,1))
107
108    %Set initial state as reconstructed state for next iteration.
109    vobj_init = v_PC;
110 end
111
112 %% Define initial overlap for algorithm and final overlaps for each ntrial.
113 first_initial_overlap = initial_overlap(1);
114 overlaps = final_overlap;
115
116 %% Graph log error of overlap vs. iteration.
117 figure(nrepeat+1)
118 plot(log(1-final_overlap))
119 xlabel('Iteration','FontSize',14)
120 ylabel('ln(1-|\it 0))','FontSize',14)
121 end

```

Time Evolution

```

1 function v_evolved = exponentiate(v_in,H_in,dt_in,subdivisions)
2 %% EXPONENTIATE performs time evolution on an input state using a Taylor
3 %% approximation of the input Hamiltonian.
4     % v_in is the input state.
5     % H_in is the input Hamiltonian.
6     % dt is the time step for which time evolution occurs. dt_in is defined
7     % as dt multiplied by a factor of -i when using EXPONENTIATE.

```

```

8     % subdivisions is the number of times the Taylor approximation is
9     % used in the time step.
10
11  %% Split the time step based on the number of subdivisions.
12  ddt = dt_in/subdivisions;
13
14  %% Define v_evolved to be equal to the time evolution of v_in for the first
15  %% time subdivision.
16  v_evolved = v_in + ddt*H_in*v_in + ddt^2/2*H_in*(H_in*v_in) ...
17      + ddt^3/6*H_in*(H_in*(H_in*v_in));
18
19  %% Continue evolving v_evolved for each time subdivision in dt.
20  for nt = 1:subdivisions-1
21      v_evolved = v_evolved + ddt*H_in*v_evolved + ddt^2/2*H_in ...
22          *(H_in*v_evolved) + ddt^3/6*H_in*(H_in*(H_in*v_evolved));
23  end

```

Link to code: <https://github.com/kenctrl/Projected-Cooling-Sensor>

REFERENCES

- [1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature* **574**, 505 (2019).
- [2] A. Y. Kitaev, A. Shen, M. N. Vyalyi, and M. N. Vyalyi, *Classical and quantum computation*, 47 (American Mathematical Soc., 2002).
- [3] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter, and J. Du, Quantum adiabatic algorithm for factorization and its experimental implementation, *Physical review letters* **101**, 220405 (2008).
- [4] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem (1995), arXiv:quant-ph/9511026 [quant-ph].
- [5] D. S. Abrams and S. Lloyd, Simulation of many-body fermi systems on a universal quantum computer, *Physical Review Letters* **79**, 2586–2589 (1997).
- [6] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution (2000), arXiv:quant-ph/0001106 [quant-ph].
- [7] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem, *Science* **292**, 472–475 (2001).
- [8] A. M. Childs, E. Farhi, and J. Preskill, Robustness of adiabatic quantum computation, *Physical Review A* **65**, 012322 (2001).
- [9] D. Lee, J. Bonitati, G. Given, C. Hicks, N. Li, B.-N. Lu, A. Rai, A. Sarkar, and J. Watkins, Projected cooling algorithm for quantum computation, *Physics Letters B* , 135536 (2020).
- [10] W. Huang, C. Yang, K. Chan, T. Tanttu, B. Hensen, R. Leon, M. Fogarty, J. Hwang, F. Hudson, K. M. Itoh, *et al.*, Fidelity benchmarks for two-qubit gates in silicon, *Nature* **569**, 532 (2019).
- [11] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, A variational eigenvalue solver on a photonic quantum processor, *Nature Communications* **5**, 10.1038/ncomms5213 (2014).
- [12] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, Cloud quantum computing of an atomic nucleus, *Physical review letters* **120**, 210501 (2018).
- [13] H. F. Trotter, On the product of semi-groups of operators, *Proceedings of the American Mathematical Society* **10**, 545 (1959).

本参赛团队声明所提交的论文是在指导老师指导下进行的研究工作和取得的研究成果。尽本团队所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果。若有不实之处，本人愿意承担一切相关责任。

参赛队员：Kenneth Choi

指导老师：Dean Lee

年 2020 月 9 日 13