Experimental Implementation of the Randomized Adaptive Quantum Algorithm

Prakriti Biswas,[1] Ariq Haqq,[1] and Christian Arenz[1]

[1]*Quantum Control and Quantum Information Lab, Arizona State University.*

(Dated: October 21, 2023)

In this document, we discuss the experimental results of implementing the randomized adaptive quantum algorithm on different problem Hamiltonians (ZZ and $XX + YY$) and compare the experimental results with their simulated results. The first three sections explore the results of the algorithm using random unitaries prepared by sampling from fully Haar random unitaries, unitary 2-design, and the Clifford group. Section D and E, the random unitaries are selected from a pool of Pauli operators. The last section, explores the generalization of the algorithm for density operators.

I. RANDOMIZED ADAPTIVE QUANTUM ALGORITHM

Quantum technology relies on methods to create quantum states, and preparing the ground state is essential. This can also have applications in solving combinatorial optimization problems. Variational Quantum Algorithms (VQAs)[3] were introduced to prepare the ground state of a problem Hamiltonian H_p, achieved by minimizing a cost function $J(\underline{\theta})$ that considers the expectation value of the problem Hamiltonian with respect to the prepared state $|\psi(\underline{\theta})\rangle$, mathematically formulated as, $J(\underline{\theta}) = \langle\psi(\underline{\theta})| H_p |\psi(\underline{\theta})\rangle$. Adaptive quantum algorithms address issues faced by VQAs by dynamically expanding the quantum circuit using a sequence of unitaries according to,

$$U_{k+1} = e^{-i\theta_k H_k} U_k,\ U_0 = \mathbb{1} \tag{1}$$

This helps in finding solutions to specific problems effectively. For ground state problems, we aim to adaptively decrease $J_k = \langle\psi_k| H_p |\psi_k\rangle$, in each adaptive step $k = 0, 1, 2\ldots$ by creating the states $|\psi_k\rangle = U_k |\psi_0\rangle$, starting from a fixed initial state $|\psi_0\rangle$, so that $J_{k+1} \leq J_k$. To achieve this goal, in each adaptive step k, we move into the negative direction of the gradient of J with respect to θ_k, by choosing θ_k as follows[4],

$$\theta_k = -\gamma \frac{dJ(0)}{d\theta_k} = -\gamma \langle \text{grad}J[U_k], iH_k\rangle \tag{2}$$

where,

$$\text{grad}J[U_k] = [|\psi_k\rangle \langle\psi_k| , H_p] \tag{3}$$

Picking a sufficiently small learning rate γ,

$$\gamma \leq \frac{1}{4||H_p||_2}$$

ensures that the change in the cost function in successive iterations $\Delta J_k = J_k - J_{k+1}$ is lower bounded by,

$$\Delta J_k \geq \frac{1}{8||H_p||_2} \left(\frac{dJ(0)}{d\theta_k}\right)^2 \tag{4}$$

In the randomized adaptive algorithm[2], randomization is introduced through uniformly randomizing the creation of H_k, which allows for favorable convergence behavior.

H_k are created by conjugating a traceless Hermitian operator H by a random unitary transformation V_k, e.g., as,

$$H_k = V_k^\dagger (X \otimes \mathbb{1}) V_k \tag{5}$$

Convergence is achieved in either one of two cases:

- The gradient $\frac{dJ(0)}{d\theta_k}$ vanishes when iH_k is orthogonal to grad$J[U_k]$. Since H_k is created at random, the probability of this scenario is 0.
- The gradient vanishes if grad$J[U_k] = 0$. The set of critical points where grad$J[U_k]$ vanishes consists of global optima and saddle points only.

A. Theoretical vs. Experimental Results with Full Haar Random Unitaries

In this section, the random H_k is created by conjugating a traceless Hermitian operator H (The Pauli X operator in this case) by a Haar random unitary transformation V_k in each adaptive step k as shown in Eq.(5). The results are shown in figures 1 and 2.

B. Theoretical vs. Experimental Results with Unitaries sampled using 2-designs

In this section, H_k is created according to Eq.(5). The random unitary transformation V_k is prepared by sampling from a unitary 2-design. The unitary 2-designs are approximately implemented by alternately repeating random unitaries diagonal in the Pauli-Z basis and Pauli-X basis. The results are shown in figures 3 and 4

C. Theoretical vs. Experimental Results with Unitaries sampled from the Clifford group

In this variation of the algorithm, instead of taking the sampling our random unitary (V_k) from the Haar mea-

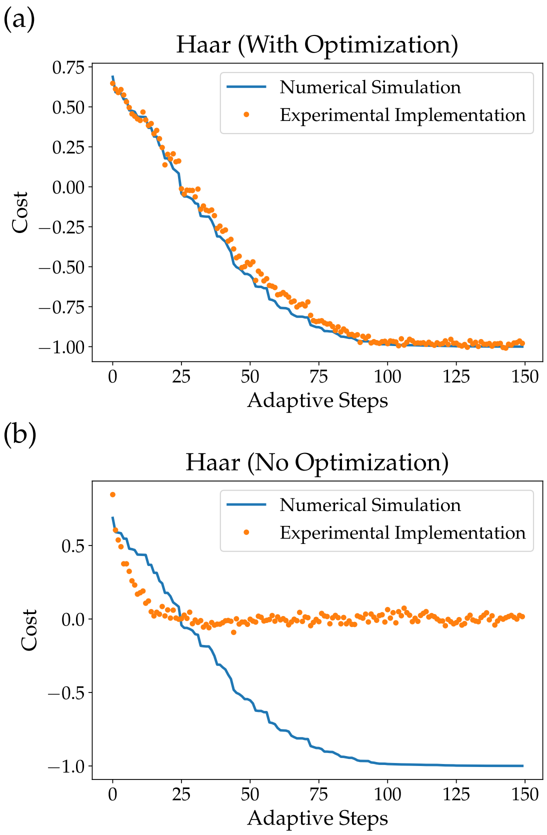


FIG. 1. This figure shows the cost over the number of adaptive steps for the ZZ problem where the random unitary V_k is sampled via the Haar measure. Plot (a) shows the the experimental cost overlaid on the theoretical cost with heavy optimization enabled on Qiskit's transpiler. Plot (b) shows the experimental cost overlaid on theoretical cost with no optimization enabled. The experimental results are generated using IBM's quantum device "IBM Mumbai".

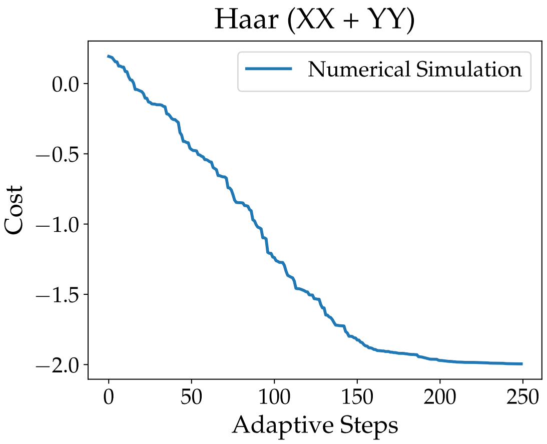


FIG. 2. This figure shows the cost over the number of adaptive steps for the XX+YY problem where the random unitary V_k is sampled via the Haar measure. The results were numerically simulated using Qiskit.

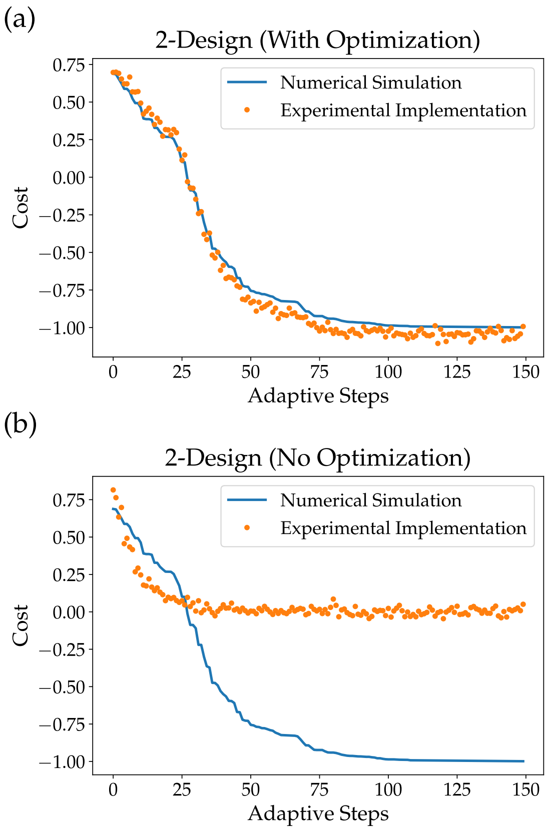


FIG. 3. This figure shows the cost over the number of adaptive steps for the ZZ problem where the random unitary V_k is sampled from the unitary 2-design. Plot (a) shows the the experimental cost overlaid on the theoretical cost with heavy optimization enabled on Qiskit's transpiler. Plot (b) shows the experimental cost overlaid on theoretical cost with no optimization enabled. The experimental results are generated using IBM's quantum device "IBM Mumbai".

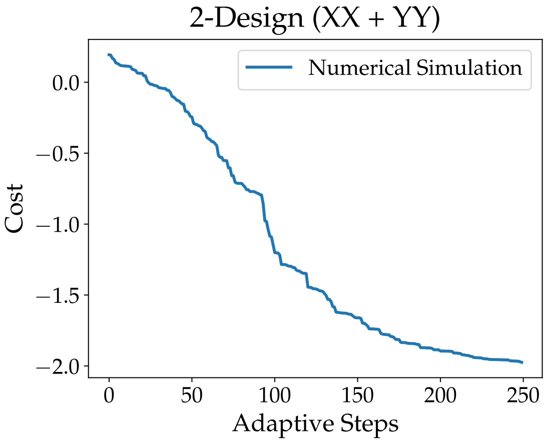


FIG. 4. This figure shows the cost over the number of adaptive steps for the XX+YY problem where the random unitary V_k is sampled from the unitary 2-design. These results were numerically simulated using Qiskit

sure, we instead sample it from the the Clifford group. The Clifford group is a 3-design unitary and we use it to approximate our random unitary. Taking the adjoint is considered to be efficient.

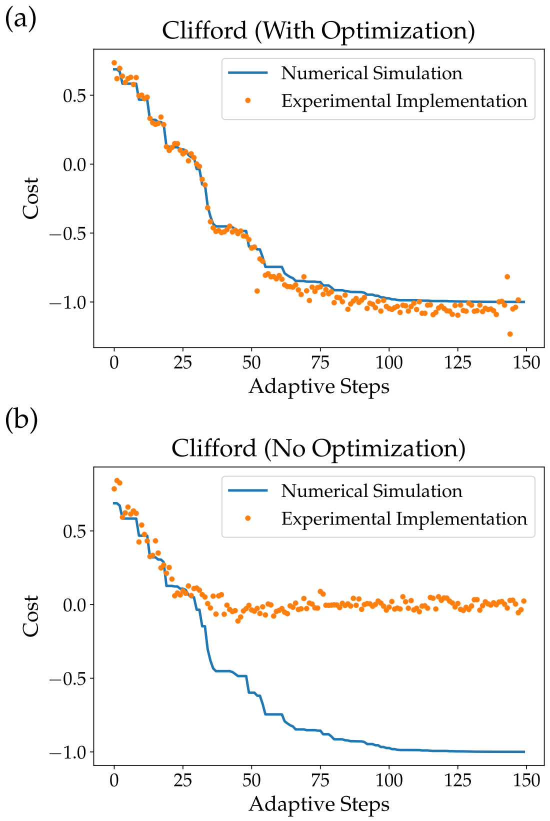


FIG. 5. This figure shows the cost over number of adaptive steps for the ZZ problem where the random unitary V_k is sampled from the Clifford group. Plot (a) shows the the experimental cost overlaid on the theoretical cost with heavy optimization enabled on Qiskit's transpiler. Plot (b) shows the experimental cost overlaid on theoretical cost with no optimization enabled. The experimental results are generated using IBM's quantum device "IBM Mumbai".

D. Theoretical vs. Experimental Results Pooled Pauli

In this variation of the random adaptive algorithm, we employ a different strategy in creating $e^{-i\theta_k H_k}$. Instead of constructing H_k by taking $H_k = V_k^\dagger(X \otimes \mathbb{1})V_k$, we instead sample from the set of Pauli strings, excluding the the identity matrix. For example, for 2 qubits H_k can be $X \otimes Y$ or $\mathbb{1} \otimes X$ for a given iteration of the algorithm. We then utilize quantum simulation in order to simulate $e^{-i\theta_k H_k}$ and append to our quantum circuit. We similarly measure the gradient and set θ_k with the same procedure outlined before.

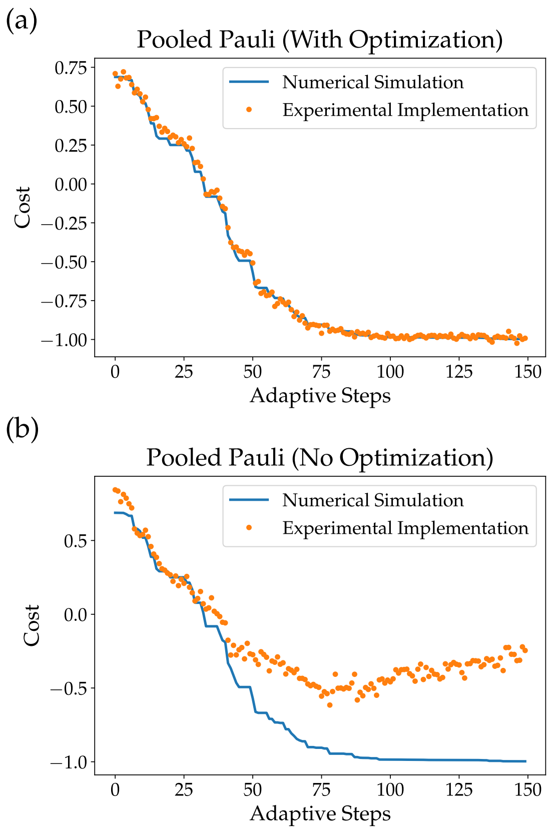


FIG. 6. This figure shows the cost over number of adaptive steps for the ZZ problem where H_k is sampled from the set of 2-qubit Pauli strings. Plot (a) shows the the experimental cost overlaid on the theoretical cost with heavy optimization enabled on Qiskit's transpiler. Plot (b) shows the experimental cost overlaid on theoretical cost with no optimization enabled. The experimental results are generated using IBM's quantum device "IBM Mumbai".

E. Theoretical vs. Experimental Results Pooled Pauli (Pre-selection)

In the following variation, we instead of constructing $H_k = V_k^\dagger(X \otimes \mathbb{1})V_k$, we instead sample a Pauli string n times, without replacement excluding the identity matrix. For example, for a 2 qubit system and $n = 2$, for a given iteration we can sample $X \otimes Y$ and $\mathbb{1} \otimes Z$. We then simulate $e^{-i\theta_k H_k}$ (since θ_k is small we use one repetition). We then measure the gradient, where we set a Pauli string to H_k. Then we select the Pauli string that had the largest gradient in magnitude to add to the quantum circuit.

F. Randomized Adaptive Quantum State Diagonalization

Here we generalize the random adaptive algorithm for density operators. In this case we are specifically, ad-

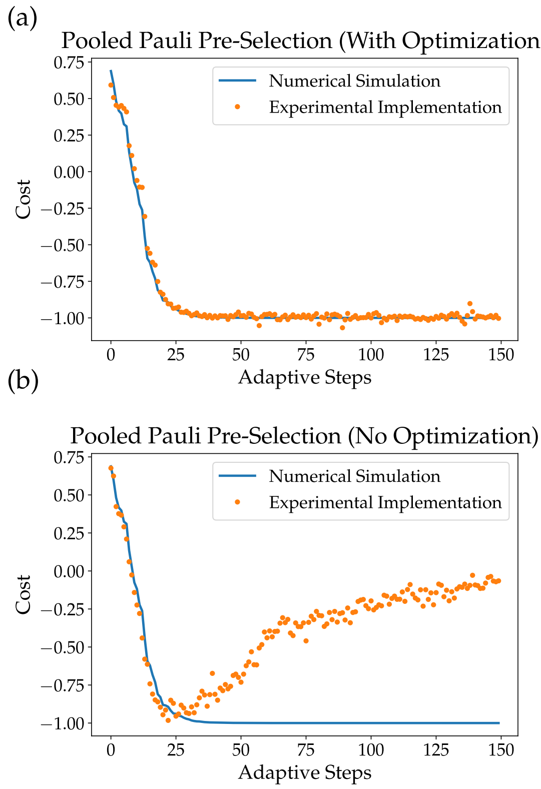


FIG. 7. This figure shows the cost over number of adaptive steps for the ZZ problem where $n = 5$ Pauli strings are sampled from the set of 2-qubit Pauli strings. Plot (a) shows the the experimental cost overlaid on the theoretical cost with heavy optimization enabled on Qiskit's transpiler. Plot (b) shows the experimental cost overlaid on theoretical cost with no optimization enabled. The experimental results are generated using IBM's quantum device "IBM Mumbai".

justing the variational quantum state eigensolver (VQSE) outlined in the paper by Cerezo, Sharma, Arrasmith, and Coles [3]. The goal of VQSE is to find the m largest eigenvalues and eigenvectors of a given density matrix ρ. Note that this is one of the major steps of quantum principal component analysis. The cost function outlined (VQSE) is given by the following $C(\underline{\theta}) = Tr[HV(\underline{\theta})\rho V^{\dagger}(\underline{\theta})]$. Here H is a n-qubit Hamiltonian that is diagonal in that standard basis and $V(\underline{\theta})$ is the quantum circuit that we train on. At each iteration, H is constructed by $\mathbb{1} - \sum_{i=1}^{m} q_i \left|z_i\right\rangle \left\langle z_i\right|$. Where $\{z_i\}_{i=1}^{m}$ is the set of bitstrings ordered by their corresponding m-largest eigenvalues. q_i is taken to be real and positive where $q_i > 0$ and $q_i > q_{i+1}$. The output of VQSE is another density matrix ρ, where the m largest eigenvalues are encoded by measuring in the standard basis and reading off the m largest frequencies of those measurements.

We modify the variation algorithm, by generalizing our corresponding cost function: $J_k = Tr[\rho_k H_p]$. We also modify how we calculate the objective input: $\theta_k = -\gamma i Tr[\rho_k[H_k, H_p]]$. H_k and V_k are created the same way as the normal adaptive algorithm.

The variational quantum algorithm of the eigensolver outlined in the Cerezo paper is not guaranteed to be able to reach the optimal solution for an arbitrary density matrix, however as shown numerically the randomized adaptive approach is able to achieve this.

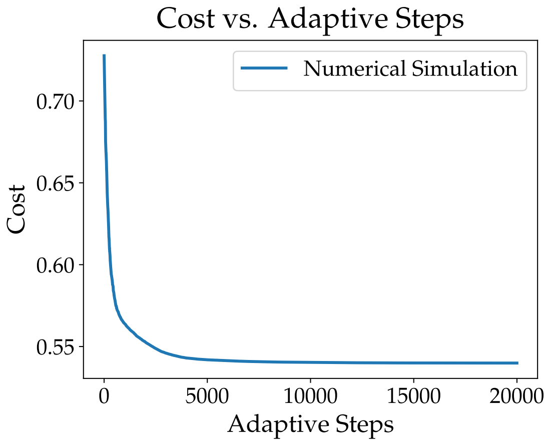
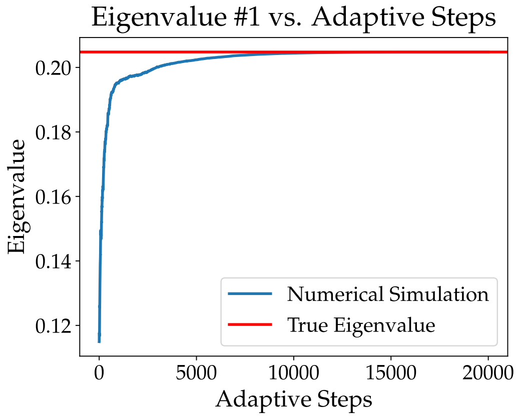


FIG. 8. This figure shows the theoretical cost over adaptive steps of the randomized adaptive quantum state eigensolver. m was taken to be 6 and ρ was a 4 qubit matrix with rank 16. These results were numerically simulated using Qiskit.



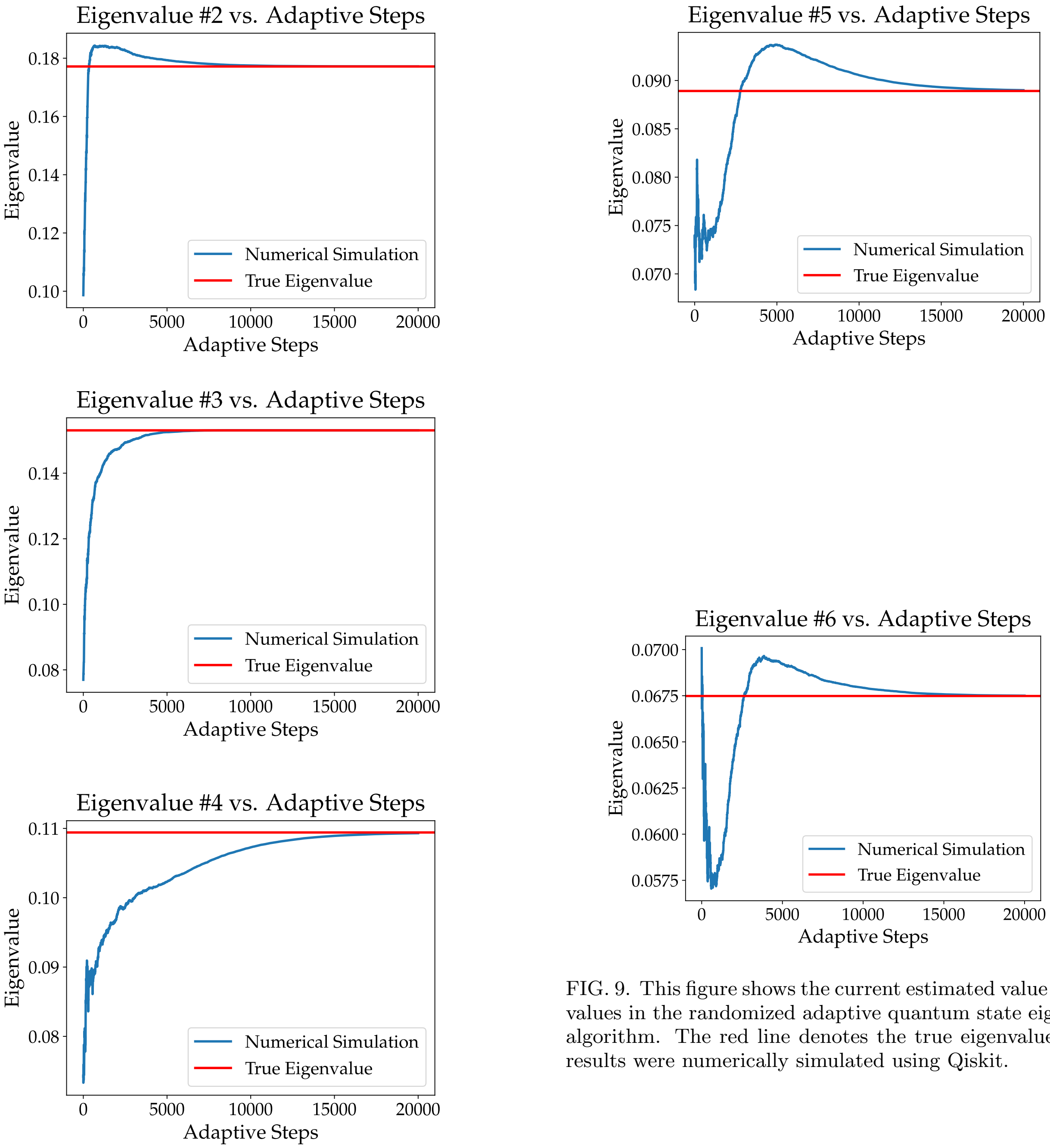


FIG. 9. This figure shows the current estimated value of eigenvalues in the randomized adaptive quantum state eigensolver algorithm. The red line denotes the true eigenvalue. These results were numerically simulated using Qiskit.

[1] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, Nature Communications **10**, 10.1038/s41467-019-10988-2 (2019).

[2] A. B. Magann, S. E. Economou, and C. Arenz, Randomized adaptive quantum state preparation (2023), arXiv:2301.04201 [quant-ph].

[3] M. Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles, Variational quantum state eigensolver, npj Quantum Information **8**, 10.1038/s41534-022-00611-6 (2022).

[4] T. SCHULTE-HERBRÜGGEN, S. J. GLASER, G. DIRR, and U. HELMKE, GRADIENT FLOWS FOR OPTIMIZATION IN QUANTUM INFORMATION AND QUANTUM DYNAMICS: FOUNDATIONS AND APPLICATIONS, Reviews in Mathematical Physics **22**, 597 (2010).