Jacob Williams
April 28, 2020

FINAL PROJECT: QUANTUM SIMULATION OF A HEISENBERG CHAIN PHYSICS 690, SPRING 2020

Note: All matrices are assumed to be in the computational basis; for two-qubit gates, the order of rows and columns is $|00\rangle\langle 00|, |01\rangle\langle 01|, |10\rangle\langle 10|, |11\rangle\langle 11|$.

1. Introduction

1.1. **Problem description.** Consider the Heisenberg model with open boundary conditions, which has the Hamiltonian

$$\mathcal{H} = \sum_{i=1}^{L-1} \mathbf{S}_i \cdot \mathbf{S}_{i+1} = \sum_{i=1}^{L-1} \left[\sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z\right],$$

where, taking $|0\rangle = |\uparrow\rangle$ and $|1\rangle = |\downarrow\rangle$,

$$\sigma_i^x = I \otimes \cdots \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \cdots \otimes I, \quad \sigma_i^y = I \otimes \cdots \otimes \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \otimes \cdots \otimes I, \quad \sigma_i^z = I \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \cdots \otimes I;$$

the only nontrivial action is on lattice site i.

Then we may write

$$\mathcal{H} = \sum_{i=1}^{L-1} \left[I \otimes \cdots \otimes (\sigma^x \otimes \sigma^x + \sigma^y \otimes \sigma^y + \sigma_z \otimes \sigma^z)_{i,i+1} \otimes \cdots \otimes I\right] \coloneqq \sum_{i=1}^{L-1} I \otimes \cdots \otimes h_k \otimes \cdots \otimes I,$$

where again the nontrivial action h_k is on lattice sites i and $i+1$. Next, let the subscripts on the matrices denote the sites of nontrivial action, and observe that

$$\sigma_i^x \otimes \sigma_{i+1}^x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}_{i,i+1} \quad ; \quad \sigma_i^y \otimes \sigma_{i+1}^y = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}_{i,i+1} \quad ; \quad \sigma_i^z \otimes \sigma_{i+1}^z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{i,i+1},$$

so that

$$\mathcal{H} = \sum_{i=1}^{L-1} I \otimes \cdots \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{i,i+1} \otimes \cdots \otimes I.$$

Unitarily diagonalizing, we have that

$$h_k = V^\dagger \Lambda V = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Next, to evolve the system in time, we require the time evolution operator

$$U(t) = e^{-i\mathcal{H}t}.$$

This is difficult to obtain, so we will use a first-order Trotter approximation, yielding

$$U(t) = e^{-i\sum_k h_k t} \approx \prod_k e^{-ih_k t}.$$

Furthermore, we must use discrete timesteps Δt, so, letting $t = n\Delta t$ we obtain that

$$U(\Delta t)\left(e^{-ih_1 \Delta t} \dots e^{-ih_{L-1}\Delta t}\right)^n = U(t) + \mathcal{O}(\Delta t^2).$$

Thus, we seek the operators $e^{-ih_k\Delta t}$, for each k. Since we have diagonalized the h_k, these are not hard to find:

$$e^{-ih_k\Delta t} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e^{3i\Delta t} & 0 & 0 & 0 \\ 0 & e^{-i\Delta t} & 0 & 0 \\ 0 & 0 & e^{-i\Delta t} & 0 \\ 0 & 0 & 0 & e^{-i\Delta t} \end{pmatrix} \begin{pmatrix} 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} e^{-i\Delta t} & 0 & 0 & 0 \\ 0 & \frac{1}{2}\left(e^{-i\Delta t} + e^{3i\Delta t}\right) & \frac{1}{2}\left(e^{-i\Delta t} - e^{3i\Delta t}\right) & 0 \\ 0 & \frac{1}{2}\left(e^{-i\Delta t} - e^{3i\Delta t}\right) & \frac{1}{2}\left(e^{-i\Delta t} + e^{3i\Delta t}\right) & 0 \\ 0 & 0 & 0 & e^{-i\Delta t} \end{pmatrix}.$$

1.2. **Quantum circuits for unitary time evolution.** We describe our quantum operations this using the language of quantum circuit diagrams, where a qubit is denoted by a horizontal line, with unitary gates (denoted by squares with inset letters) acting on them from left to right. In particular,

(A) One-qubit unitary U. (B) Two-qubit unitary U_2. (C) The CNOT gate. (D) Controlled-U.

FIGURE 1. Quantum circuit diagrams.

Note that a controlled unitary operation CU is defined by

$$CU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U;$$

that is, U is executed only if the *control qubit* is in the state $|1\rangle$, and the identity operator is executed if the control qubit is in the state $|0\rangle$.

In particular, the CNOT (equivalently, CX) gate, is given by

$$CX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

As described in [1], single-qubit gates and CNOT are *universal*; that is, any unitary operation acting on an arbitrary number of qubits can be constructed by composing (possibly exponentially many) single-qubit gates and CNOTs. Since we only require one- and two-qubit gates, it will not be too difficult; Vatan and Williams found in [2] that any two-qubit gate can be simulated with no more than three CNOT and fifteen single-qubit gates (and that Pauli operator exponentials like $U(\Delta t)$ can be simulated even more efficiently).

1.3. **Implementing observables.** An observable of a system is a Hermitian operator A, which necessarily has a spectral decomposition

$$A = \sum_i \lambda_i \Pi_i, \qquad \lambda_i \in \mathbb{R} \ \forall \ i,$$

where Π_i is the projector onto the eigenspace λ_i. Measuring an observable on a quantum system projects the state of the system onto one of the eigenspaces of the observable and records the proper eigenvalue. The probability that measuring A on $|\psi\rangle$ gives the eigenvalue λ_i is $P_i = \langle\psi|\Pi_i|\psi\rangle$.

An important measurement in quantum computation is called the *measurement in the computational basis*, which returns (say) 0 if $|\psi\rangle = |0\rangle$ and 1 if $|\psi\rangle = |1\rangle$; that is,

$$\lambda_0 = 0, \quad \Pi_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}; \quad \lambda_1 = 1, \quad \Pi_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

In the language of quantum circuits, measuring $|\psi\rangle$ in the computational basis is denoted as

$|\psi\rangle$  .

For our spin system, measurement in the computational basis is equivalent to measuring spin, for (without loss of generality) we may let $|\downarrow\rangle = |0\rangle$ and $|\uparrow\rangle = |1\rangle$. In particular, it is well known that the ground state of the Heisenberg chain is $|\downarrow \cdots \downarrow\rangle = |0 \cdots 0\rangle$, which has energy zero.

2. Implementation details

IBM Q Experience is a cloud quantum computing service publicly available to academics, with one 16-qubit quantum processor and several 5-qubit processors, as well as a classical simulator called `QASM_simulator` of up to 32 qubits. The Q Experience quantum computers run on an architecture of superconducting qubits, usually with quasi-two-dimensional nearest-neighbor connectivity (the precise nature of which depends on the specific processor in question).

Q Experience processors are programmed with the Qiskit Python interface [3], or with a drag-and-drop circuit builder. The primitive gate set for the Q Experience processors includes the identity operator; arbitrary single-qubit unitary operations U, with

$$U = U_3(\theta,\phi,\lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix};$$

as well as by the more limited (but likely more reliable)

$$U_1(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$$

and

$$U_2(\phi,\lambda) = U_3(\frac{\pi}{2},\phi,\lambda) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{pmatrix}.$$

The CNOT gate is the only primitive two-qubit gate.

However, a few common quantum operators such as the Hadamard gate

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

the Pauli rotations $\{R_x(\theta), R_y(\theta), R_z(\theta)\}$ (defined below), and the controlled-Z gate, are included as software builtins. They are converted automatically by Qiskit's "transpiler" to the primitive gates. In addition, Q Experience processors can in some cases automatically condense two or more consecutive single-qubit operations into a single gate, shortening the effective depth of the quantum circuit; in the analysis following, we will not consider this effect when discussing our circuit.

2.1. **Implementing one-qubit gates.**

We may readily observe that the Hadamard gate $H = U_2(0,\pi)$.

Next, we define the Pauli rotation gates. Following section 4.2 of [1], for any $\theta \in [0, 2\pi]$ we have

$$R_z(\theta) = e^{-i\theta Z/2} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} = e^{-i\theta/2}\,U_1(\theta);$$

thus,

$$R_z(\tfrac{\pi}{2}) = e^{-i\pi/4}\,U_1(\tfrac{\pi}{2}); \quad R_z(-\tfrac{\pi}{2}) = e^{i\pi/4}\,U_1(-\tfrac{\pi}{2}); \quad R_z(-2\Delta t - \tfrac{\pi}{2}) = e^{-i(\Delta t+\frac{\pi}{4})}\,U_1(-2\Delta t - \tfrac{\pi}{2}).$$

Likewise,

$$R_y(\theta) = e^{-i\theta Y/2} = \begin{pmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{pmatrix} = U_3(\theta,0,0),$$

so naturally our gates

$$R_y(\tfrac{\pi}{2} + 2\Delta t) = U_3(\tfrac{\pi}{2} + 2\Delta t, 0, 0); \quad R_y(-2\Delta t - \tfrac{\pi}{2}) = U_3(-2\Delta t - \tfrac{\pi}{2}, 0, 0).$$

The particular Pauli rotations mentioned are necessary for implementing our Trotterized operator $U(\Delta t) = e^{-ih_k\Delta t}$, as will be seen in the next section.

2.2. **Noise in the quantum computer.** Typical processors in the IBM Q Experience have error rates of 1–2% for CNOT gates, and of around $6 \cdot 10^{-4}$ for single-qubit gates. This yields a high probability of error after a fairly short circuit; for a quick estimate, we may assume the errors are independent and even neglect the single-qubit errors, obtaining for a circuit with n CNOT gates (at a 2% error rate) that

$$P(\text{error}) = 1 - P(\text{no error}) \approx 1 - 0.98^n.$$

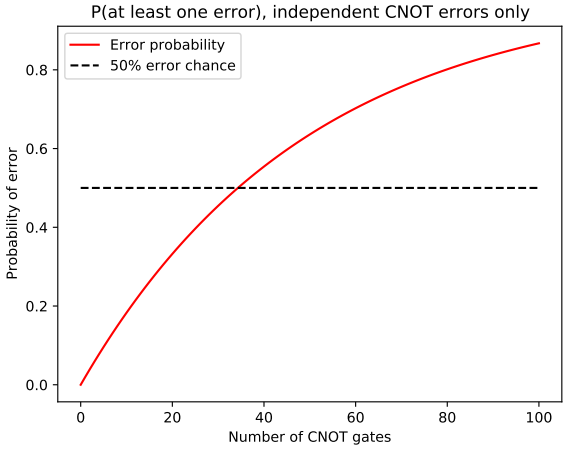


FIGURE 2. The CNOT error.

We see in Figure 2 that we have a better than even chance of error with 40 CNOT gates; considering the single-qubit errors reduces the manageable depth of our circuit even further. Thus, without implementing error correction, which is beyond the scope of this project and likely requires more than the five qubits available on most of the Q Experience processors, we will limit our circuit to fewer than 30 CNOT gates or so.

As mentioned in the previous paragraph, most public-facing IBM quantum processors have five qubits (one exception has sixteen, but for a Hamiltonian simulation limited to 30 CNOT gates it is too large to use effectively). Furthermore, most of them are not arranged linearly, rather having a shape like a T, which limits the Ising model to a maximum of four spins. We will use all four available spins for our simulation.

As each implementation of $U(\Delta t)$ requires three CNOT gates, and three applications of $U(\Delta t)$ are required to move four spins forward by one unit of Δt, we will limit ourselves to three time evolution operations; that is, $t_{max} = 3\Delta t$. For our first computation suite, we take $t = \frac{1}{2}$.

2.3. **Implementing time evolution.** Following the method and notation of [2], our two-qubit Trotterized time evolution operator is $N(-\Delta t, -\Delta t, -\Delta t)$. We can implement it using with three CNOT gates as

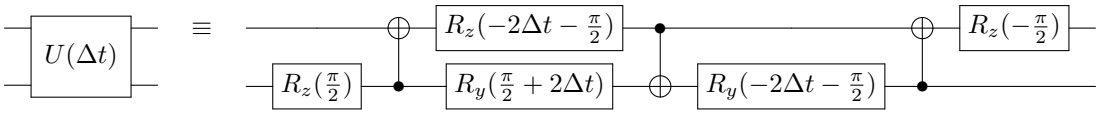
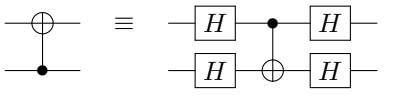


FIGURE 3. The quantum circuit for $U(\Delta t)$.

Note first that we implement the inverted CNOT gate, which gives the second and sixth gates in Figure 3, as



,

where H is the Hadamard gate. Additionally, this construction neglects a global phase of $e^{i\pi/4}$. However, we expect only to implement local observables, so the global phase can be safely ignored; even if it could not, we could keep track of it by hand to retain full information about our system.

As seen in Figure 4, to approximately evolve our system in time by Δt, we implement $U(\Delta t)$ on each pair of qubits in a cascading sequence.

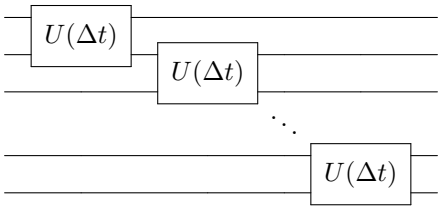


FIGURE 4. The Trotterized circuit for one increment of Δt.

Finally, our full time evolution is obtained by repeating the entire circuit of Figure 4 n times (here, thrice).

3. Running the computation

We implement the Trotterized time evolution $(U(\frac{1}{2}))^3$ approximating the evolution to time $\frac{3}{2}$ in increments $\Delta t = \frac{1}{2}$ on the IBM quantum computer `ibmq_rome` and on the simulator `qasm_simulator`, which simulates the evolution in the absence of any noise. We chose our particular processor for its fairly low CNOT error rate — its maximum two-qubit error is only 1.072%. Running the computation on a more error-prone processor, `ibmqx2`, gave abysmal results; nearly 75% of the computations evolving the ground state failed to preserve it. Both the real computation and the simulation are averages of 1024 runs (called shots in Qiskit).

Additionally, we simulated the *exact* unitary time evolution, without Trotterization, in Mathematica, to observe the error introduced by the Trotter decomposition.

We implemented $U(\Delta t)$ on the following initial states:

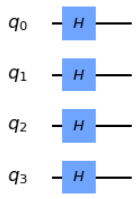
(i) $|0000\rangle$, the ground state of the system;

(ii) $|\Psi\rangle = \frac{1}{2}\sum_{x\in\{0,1\}^4} |x\rangle$, the uniform superposition over all basis states;

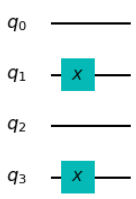
(iii) $|0101\rangle$ and $|1010\rangle$, two singlet states with $S^z_{tot} = 0$;

(iv) The highly entangled "cluster state" $|\Phi_C\rangle = \frac{1}{2}(|0+0+\rangle + |+0-1\rangle + |-1-0\rangle + |-1+1\rangle)$, prepared according to the procedure given in [4]. Note that $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and that $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

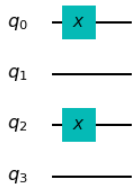
The circuits for initialization are given in Figure 5. Note that Qiskit automatically initializes qubits in the state $|0000\rangle$, so that required no additional processing.



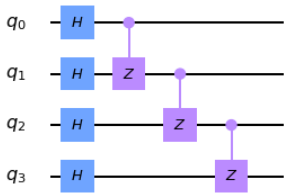
(A) The circuit to initialize $|\Psi\rangle$.



(B) The circuit to initialize $|0101\rangle$.



(C) The circuit to initialize $|1010\rangle$.

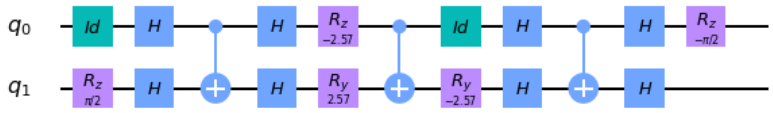


(D) The circuit to initialize $|\Phi_C\rangle$.

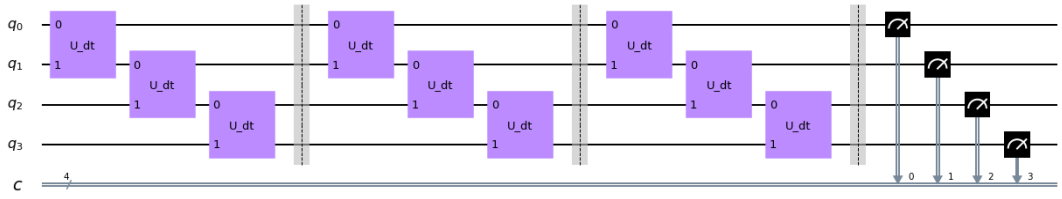
Figure 5. The initialization circuits

In Figure 3, we give the Qiskit circuit representation of $U(\Delta t)$ and of our time evolution circuit acting on the prescribed initial state.

Finally, in Figure 7, we give histograms of the outputs. On the left are the Q Experience histograms, which compare the real quantum computer to the classical simulator; on the right are the Mathematica histograms giving the exact outputs without Trotter error. The code used to generate them is included in separate files with this report, but we regret that the download feature for the Jupyter notebooks on IBM's Q Experience was not working, so that code is included only as a series of screenshots.

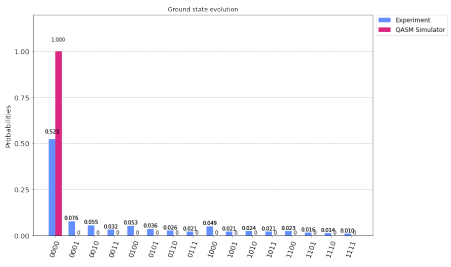


(A) The circuit implementing $U(\Delta t)$.

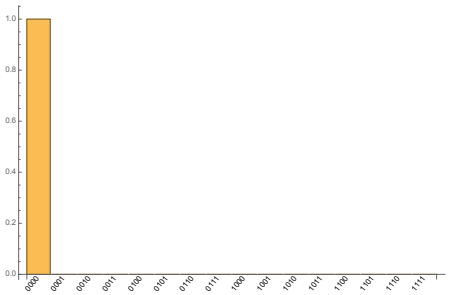


(B) The circuit implementing time evolution on $|0000\rangle$.

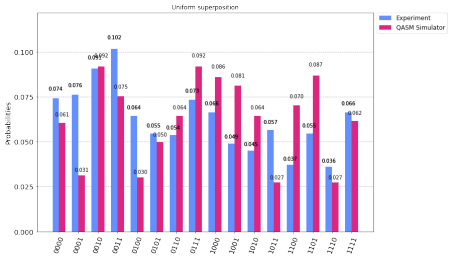
FIGURE 6. The circuit diagrams from Qiskit.



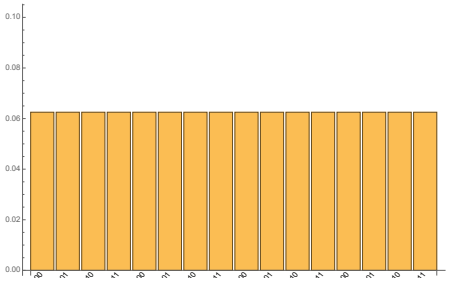
(A) The quantum output from $|0000\rangle$.



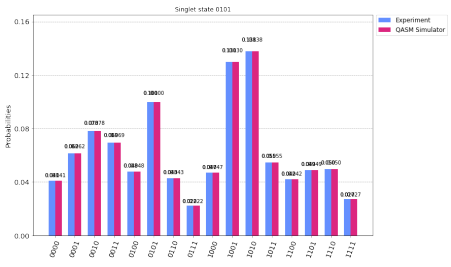
(B) The exact output from $|0000\rangle$.



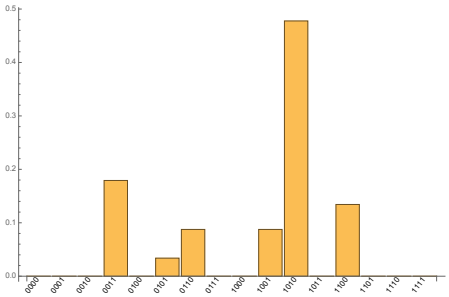
(C) The quantum output from $|\Psi\rangle$.



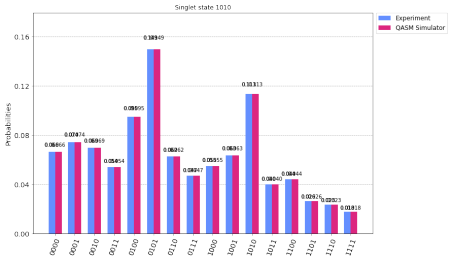
(D) The exact output from $|\Psi\rangle$.



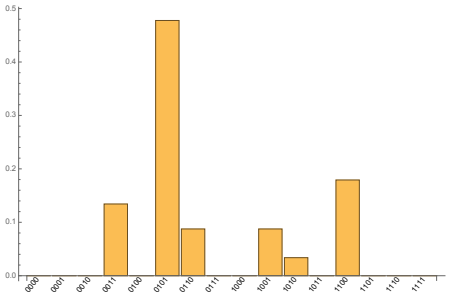
(E) The quantum output from $|0101\rangle$.



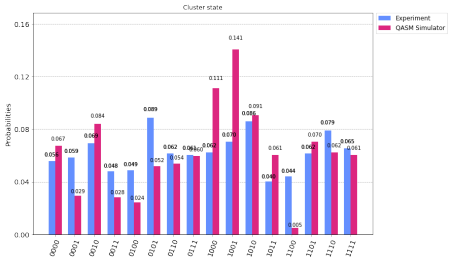
(F) The exact output from $|0101\rangle$.



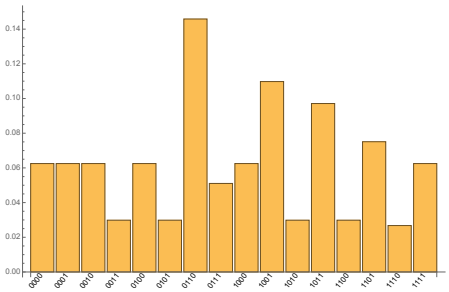
(G) The quantum output from $|1010\rangle$.



(H) The exact output from $|1010\rangle$.



(I) The quantum output from $|\Phi_C\rangle$.



(J) The exact output from $|\Phi_C\rangle$.

FIGURE 7. The output for $U(\Delta t = \frac{1}{2})^3$.

Observing Figure 7, we may notice several things. The evolution of the ground state $|0000\rangle$ is expected to preserve the state; both the exact output and the simulated output replicate this behavior, but the real quantum computation only maintains this 52.3% of the time, even with our comparatively short circuit.

When considering the uniform superposition state $|\Psi\rangle$, we may observe that while the simulator and the true quantum computer do not agree very closely, neither comes close to replicating the exact output, which preserves $|\Psi\rangle$ (up to a global phase). Thus, Trotterization yields substantial error in this case.

The output from the simulator and from `ibmq_rome` agree almost exactly when time evolving $|0101\rangle$ and $|1010\rangle$ in time, and they also both preserve the dominance of the respective input state in the distribution of the output. However, neither accurately predicts the second-most-likely output of the exact evolution, nor do they maintain the expected zero probability of the outputs not included in the exact result.

Finally, Trotter error is significant once more in the cluster state; the simulator predicts the most probable outcome to be $|1001\rangle$, while the exact outcome is most probably $|0110\rangle$. However, the noise in the quantum processor is also significant, with the real quantum computer predicting $|0101\rangle$ and $|1010\rangle$ to be the most likely outcomes (and with more total variation overall, with only one of the sixteen outcomes having an empirical probability of occuring below 0.04, while this is attained for five outcomes in the exact output).

3.1. **A run over a shorter time.** We additionally conducted a run of the circuit, identical except that $\Delta t = \frac{1}{100}$. This was designed to mostly remove the Trotter error, allowing the quantum noise to shine through. The results of this are seen in Figure 8.

The noise is obviously apparent in the output for the ground state; the probability of error in `ibmq_rome` is still around $\frac{1}{2}$. The Trotter error is still somewhat visible, as $|\Psi\rangle$ is still not an eigenstate, although some of the variation is explainable due to the finite sample size (of 1024). The noise in the real computation increases this variance as well. Nevertheless, the simulation has all probabilities between approximately 0.047 and 0.079, not too far off from the exact 0.0625.

Comparing the singlet-state outputs from Figure 7 and Figure 8, an interesting note is made. In the exact solution when $t_{final} = 0.03$, very little time has passed, so the output state is measured to be equal to the input with high probability; on the other hand, after 1.5 time units, it is more likely that $|0101\rangle \mapsto |1010\rangle$ and vice versa. When Trotterization is performed, however, both time steps give a larger weight to the reversed singlet than to the initial state. Indeed, when $\Delta t = 0.01$, neither the experimental nor the simulated Trotterized value have a significant probability of finding $|0101\rangle$ or $|1010\rangle$ in their respective initial states at all! This indicates that, even with such a small time step, the first-order Trotterization scheme is too "harsh" of an evolution; a higher-order decomposition might be required to smooth out the evolution of the Heisenberg chain.

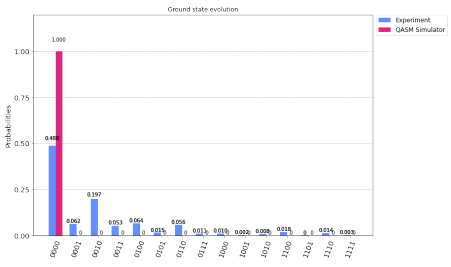
Finally, the cluster state output is close to uniform in the exact description, with some asymmetry beginning to show in the probabilities; the noise in both the simulated and experimental quantum result overwhelms this subtle distinction and makes it impossible to discuss the result in comparison.

3.2. **Error analysis.** Consider our independent error model of Section 2.2. Given the maximum CNOT error rate of 0.0172 in `ibmq_rome`, let us compare our observed error rate over our two runs with the model. We consider the ground state (which is preserved in the absence of noise) to produce this model.

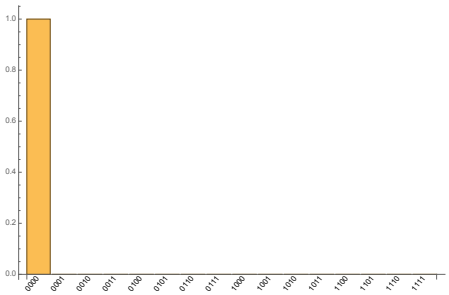
Over 1024 shots, we correctly measured $|0000\rangle$ 52.3% of the time, indicating that 536 of the experiments completed with no errors. The second experiment ran correctly 48.8% of the time, indicative of precisely 500 correctly terminating experiments. On average, then, 518 experiments ran without error, for an error rate of $1 - 0.5059 = 0.4941$.

Now, our circuit has a maximum number of 27 CNOT gates acting on any one qubit: three from each implementation of $U(\Delta t)$, which acts a total of nine times on the two central qubits. Our independent-CNOT-error model thus gives a predicted error rate of $1 - (1 - 0.0172)^{27} = 1 - 0.6260 = 0.3740$, a rather lower error rate than was observed. While our confidence in the actual error rate of our circuit is not very

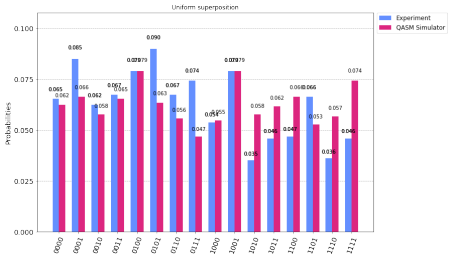
high given only two trials, the discrepancy is large enough that the highly naïve error model we propose is probably not sufficient to characterize the error in a circuit of even modest depth. More sophisticated treatments of the error will consider correlations between errors, the single-qubit error rates, the decoherence time of the whole circuit (by which errors become more likely as more gates are applied in sequence), and even error in measurement.



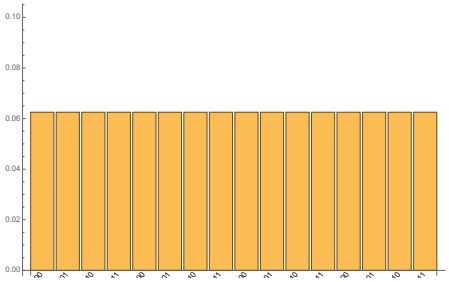
(A) The quantum output from $|0000\rangle$.



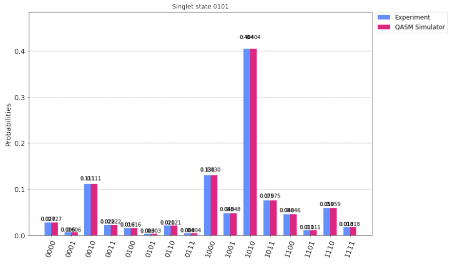
(B) The exact output from $|0000\rangle$.



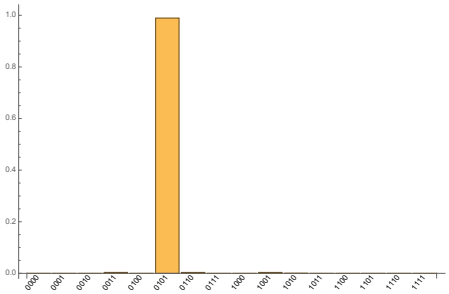
(C) The quantum output from $|\Psi\rangle$.



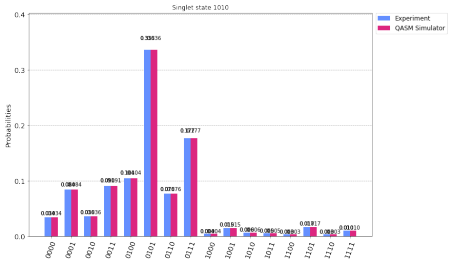
(D) The exact output from $|\Psi\rangle$.



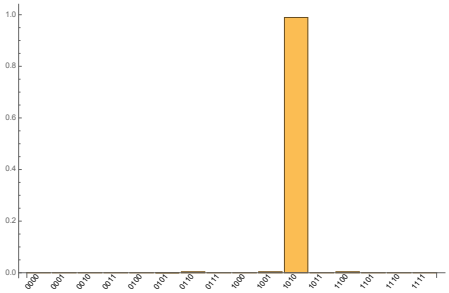
(E) The quantum output from $|0101\rangle$.



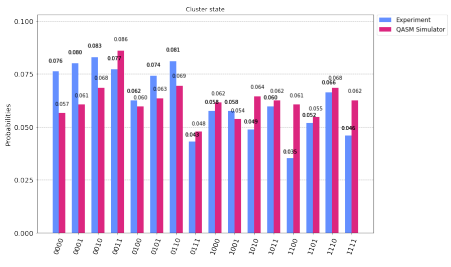
(F) The exact output from $|0101\rangle$.



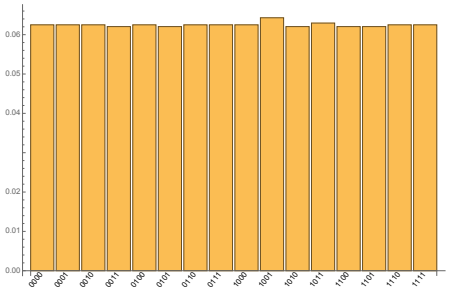
(G) The quantum output from $|1010\rangle$.



(H) The exact output from $|1010\rangle$.



(I) The quantum output from $|\Phi_C\rangle$.



(J) The exact output from $|\Phi_C\rangle$.

FIGURE 8. The output for $U(\Delta t = \frac{1}{100})^3$.

4. Conclusions

The most important conclusion we may draw from this study is the power of noise to destroy quantum computational results and the consequent importance of quantum error correction. Even in this circuit, for which no more than sixty-nine gates act upon any qubit (eleven gates in $U(\Delta t)$, which acts six times in total on the two central qubits, and three more if initializing $|\Phi_C\rangle$), the noise qualitatively changes the outcome of the circuit relative to the noiseless simulator.

Even the Trotter error refers back to the noise; it is $\mathcal{O}(\Delta t^2)$, indicating that our $\Delta t = \frac{1}{2}$ is too large. However, the fact that errors are so ubiquitous in the quantum computation means that we cannot run more applications of $U(\Delta t)$ with a smaller time step to decrease the overall error: while the Trotter error decreases, the error from decoherence will increase apace, rendering our results just as useless. Likewise, the fact that the eigenstates of the exact Hamiltonian are not always conserved under Trotterization introduces error even for small values of Δt, requiring better approximations to e^{-iHt} to remedy.

Thus we see that, even for this simple system, quantum computers are not yet ready to take on their classical counterparts in simulating many-body quantum dynamics.

4.1. **Further direction.**

It is worth mentioning, however, that the algorithm described above is quite flexible. Following the method of [2], $U(\Delta t)$ could be modified to apply to the general XYZ spin chain, which has Hamiltonian

$$H_{XYZ} = \sum_{i=1}^{L-1} \left[\alpha\sigma_i^x\sigma_{i+1}^x + \beta\sigma_i^y\sigma_{i+1}^y + \gamma\sigma_i^z\sigma_{i+1}^z\right],$$

by simply altering the parameters in the Pauli rotation gates. Depending on the connectivity of the quantum circuit, periodic boundary conditions could also be implemented, although in the case of the IBM Q Experience processors would not readily accept this modification.

Additionally, while we showed the effect of preparing different input states, we could also measure various observables. Local observables would be relatively straightforward to implement; we would need to rotate each qubit into the observable's eigenbasis, then measure. For global observables, the quantum phase estimation algorithm, which is described in [1], is utilized; however, it requires an ancillary register of k qubits if the given eigenvalue is to be simulated with a precision of 2^{-k}. This is not particularly difficult to envision, but the five qubits available on most IBM processors are not sufficient. The gate depth of this circuit is also larger than the Q Experience processors could effectively handle.

References

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. ISBN: 978-1-107-00217-3.

[2] Farrokh Vatan and Colin Williams. "Optimal quantum circuits for general two-qubit gates". *Phys. Rev. A* 69 (2004), 032315. DOI: 10.1103/PhysRevA.69.032315. arXiv: quant-ph/0308006.

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

[4] Philippe Jorand and Simon Perdrix. "Unifying quantum computation with projective measurements only and one-way quantum computation". *18th International Conference on Photoelectronics and Night Vision Devices and Quantum Informatics 2004* (Moscow, Russian Federation). Ed. by Yuri I. Ozhigov. SPIE, 2005, 44–51. ISBN: 0277-786X/05/$15. DOI: 10.1117/12.620302.