

PAPERS | JANUARY 01 2025

Simulating spin dynamics with quantum computers

Jarrett L. Lancaster  ; D. Brysen Allen 



Am. J. Phys. 93, 98–109 (2025)
<https://doi.org/10.1119/5.0112717>



Articles You May Be Interested In

Quantum thermodynamic methods to purify a qubit on a quantum processing unit

AVS Quantum Sci. (June 2022)

Ptychographic estimation of pure multiqubit states in a quantum device

Testing precision and accuracy of weak value measurements in an IBM quantum system

AVS Quantum Sci. (January 2024)



Simulating spin dynamics with quantum computers

Jarrett L. Lancaster^{a,b} and D. Brysen Allen

Department of Physics, High Point University, One University Parkway, High Point, North Carolina 27262

(Received 21 July 2022; accepted 7 September 2024)

IBM quantum computers are used to simulate the dynamics of small systems of interacting quantum spins. For time-independent systems with fewer than three spins, we compute the exact time evolution at arbitrary times and measure spin expectation values and energy. It is demonstrated that even in such small systems, one can observe the connection between conservation laws and symmetries in the model. Larger systems require approximating the time-evolution operator, and we investigate the case of $N = 3$ spins explicitly. While it is shown to be unfeasible to use such devices to probe such larger systems without more advanced algorithms or reliable error correction, we demonstrate that the quantum circuit simulator is an easy-to-use method for studying spin dynamics in systems with $N \sim \mathcal{O}(10)$ spins. The computations presented provide an interesting experimental component to the standard treatment of quantum spin in an undergraduate quantum mechanics course. © 2025 Published under an exclusive license by American Association of Physics Teachers. <https://doi.org/10.1119/5.0112717>

I. INTRODUCTION

The field of quantum computation has experienced tremendous growth in recent years. In 1982, Richard Feynman suggested that (then-hypothetical) quantum computers were the only practical way to simulate quantum mechanical systems with many degrees of freedom.¹ Over the next several decades, quantum computing algorithms were developed that may play a potentially valuable role in an efficient factorization scheme.² Such algorithms present a threat to current cryptographic systems because the security of the widely used Rivest–Shamir–Adleman (RSA)³ cryptographic scheme rests upon the computational difficulty of factoring a product of two large, prime numbers. The so-called noisy, intermediate-scale quantum (NISQ) devices⁴ are widely available from providers such as Rigetti,⁵ IonQ,⁶ and IBM,⁷ though the currently available devices are still far from capable of these advanced types of computations. The question for the foreseeable future appears to be: what kinds of problems are these NISQ devices capable of solving?

While many potential uses for NISQ-era hardware have been proposed,⁸ one under-appreciated use for these devices is as a learning tool for quantum mechanics. Presently available quantum computers are ideally suited for exploring spin physics in small systems,⁹ such as those that would be explored theoretically in a typical undergraduate course on quantum mechanics. Recent papers in this journal have showcased a variety of quantum information applications suitable for undergraduate students. Examples include exploring basic problems future quantum computers can conceivably be used to solve,¹⁰ employing the Grover search algorithm on actual quantum hardware,¹¹ and applying basic error mitigation to improve results on currently available hardware.¹² A particularly interesting physical application of this technology used quantum computers to measure spin correlations.⁹ Given the free availability actual quantum hardware from IBM, the simulation of spin systems provides an incredible opportunity to students. In other words, there is now a no-cost experimental component to complement the fairly abstract theory one finds in a typical, undergraduate course in quantum mechanics.

The main purpose of this paper is to describe several types of spin simulations involving the time evolution of arbitrarily prepared initial states. The Qiskit software development kit

(SDK)¹³ is used to encode time evolution in quantum circuits, which are then run on the cloud-based IBM hardware. Time evolution of a single spin is simulated by mapping the spin to a qubit and applying a single unitary gate corresponding to the exact time-evolution operator. The results for two- and three-spin systems with nontrivial interactions are also presented. The scheme for performing time evolution on two-spin systems is exact, but it requires a more complicated circuit than the single-spin case. We are forced to use approximate methods when performing time evolution in systems with three or more spins. An additional challenge is that errors from currently available devices make accurate results quite difficult to obtain for systems as small as three spins. However, we demonstrate how students can use Qiskit's built-in circuit simulator to explore systems with sizes up to $N = 20$ spins. The circuit simulator is a fairly efficient and easy-to-use tool for exploring the expected output of quantum circuits for modest system sizes. Operationally, the Qiskit simulator consists of a classical computation that gives the expected output of a given quantum circuit subject only to statistical fluctuations. In this work, we use the simulator to verify the expected output of certain circuits. Using the simulator as a computational tool in its own right, we demonstrate evidence of a quantum phase transition that appears in a large spin system as a particular system parameter is varied.

The remainder of this paper is organized as follows: Sec. II contains a review of relevant spin physics and the essential ingredients needed to simulate spin systems on IBM's quantum hardware. Dynamics in interacting systems of two and three spins are shown in Sec. III, and the Lie–Trotter decomposition is employed as a method for performing approximate time evolution in a three-spin system and on a large ($N = 20$) system using Qiskit's circuit simulator. Finally, a brief discussion is included in Sec. IV.

II. SINGLE SPIN DYNAMICS

In this section, we present the basic framework for simulating a spin degree of freedom on a quantum computer. To illustrate the approach in the simplest possible setting, we discuss the interaction of a single spin with a static magnetic

field. However, the methods detailed below can also accommodate time-dependent magnetic fields with only minor modifications.¹⁴

A single spin-1/2 degree of freedom is a two-level system whose quantum state can be expressed as a linear combination of basis kets $|\psi\rangle = \alpha|+\rangle + \beta|-\rangle$, where $|\pm\rangle$ are eigenstates of the operator \hat{S}^z , satisfying

$$\hat{S}^z|\pm\rangle = \pm \frac{\hbar}{2}|\pm\rangle. \quad (1)$$

We follow standard notation for labeling the eigenstates of \hat{S}^z (e.g., Ref. 15), but we caution the reader with experience in quantum information theory that these labels often refer to eigenstates of \hat{S}^x in other sources.¹⁶ The complex amplitudes are constrained by the normalization of the quantum state, $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$. As the global phase of $|\psi\rangle$ is unobservable, a common parameterization is given by $\alpha = \cos\theta/2$, $\beta = e^{i\phi}\sin\theta/2$. To aid in calculations, the basis states can be represented^{*} by $|+\rangle \doteq (1, 0)^T$ and $|-\rangle \doteq (0, 1)^T$ so that the state $|\psi\rangle \rightarrow |\psi(\theta, \phi)\rangle$ can be represented by the two-component vector

$$|\psi(\theta, \phi)\rangle \doteq \begin{pmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{pmatrix}. \quad (2)$$

By employing the explicit matrix representations of the spin operators $\hat{S}^\alpha = (\hbar/2)\hat{\sigma}^\alpha$, where

$$\begin{aligned} \hat{\sigma}^x &\doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, & \hat{\sigma}^y &\doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \\ \hat{\sigma}^z &\doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \end{aligned} \quad (3)$$

one may compute the expectation values of the spin operators $\langle\hat{S}^\alpha\rangle \equiv \langle\psi|\hat{S}^\alpha|\psi\rangle$ for the state in Eq. (2). Simplifying with standard, trigonometric identities, we obtain

$$\begin{aligned} \langle\hat{S}^x\rangle &= \frac{\hbar}{2}\sin\theta\cos\phi, \\ \langle\hat{S}^y\rangle &= \frac{\hbar}{2}\sin\theta\sin\phi, \\ \langle\hat{S}^z\rangle &= \frac{\hbar}{2}\cos\theta. \end{aligned} \quad (4)$$

In this form, it is apparent that the parameters θ and ϕ correspond to angular coordinates that map the possible states to points on an abstract, unit sphere—also known as the Bloch sphere.¹⁷ Time evolution for a single spin is generated by the interaction of the spin with a magnetic field \mathbf{B} via the Hamiltonian

$$\hat{H}_1 = -\mu\mathbf{B} \cdot \hat{\mathbf{S}}, \quad (5)$$

^{*}Following the convention of Sakurai¹⁸ and McIntyre,¹⁵ the symbol \doteq stands for “is represented by” in the chosen basis. While the basis employed in this work is almost universally adopted elsewhere in the literature, the specific matrix representations shown (e.g., Eq. (3)) would be different for another choice of basis.

where the constant μ represents the magnetic moment that couples the spin to an external field. The time-dependent state $|\psi(t)\rangle$ is obtained by solving the Schrödinger equation,

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle, \quad (6)$$

given some initial state $|\psi(0)\rangle$ and appropriate Hamiltonian operator, \hat{H} . The calculation of $|\psi(t)\rangle$ is typically performed by representing \hat{H} as a 2×2 matrix and extracting separate, coupled, first-order equations for the time-dependent amplitudes $\alpha(t)$ and $\beta(t)$, which can be solved using standard methods.¹⁹ With the aim of simulating such dynamics on a quantum computer, it is convenient to take a different approach. One may write the formal solution as

$$|\psi(t)\rangle = \exp\left[-\frac{i}{\hbar}\hat{H}t\right]|\psi(0)\rangle \equiv \hat{U}(t)|\psi(0)\rangle \quad (7)$$

and calculate the time-evolution operator $\hat{U}(t)$ explicitly as a 2×2 matrix. As matrix exponentiation is not always covered in introductory quantum mechanics, we consider the representative example $\hat{H} = \hat{H}_1$ with $\mathbf{B} = B_0\hat{\mathbf{x}}$ in detail. In this case, the time-evolution operator can be written as

$$\hat{U}(t) = \exp\left[\frac{i\omega_0 t}{2}\hat{\sigma}^x\right], \quad (8)$$

where $\omega_0 \equiv \mu B_0$. Using the property that $[\hat{\sigma}^x]^2 = \hat{I}$, one may expand the exponential as a Taylor series, then group even and odd terms separately to obtain

$$\begin{aligned} \hat{U}(t) &= \hat{I} \left(1 - \frac{(\omega_0 t/2)^2}{2!} + \frac{(\omega_0 t/2)^4}{4!} + \dots \right) \\ &\quad + i\hat{\sigma}^x \left(\omega_0 t/2 - \frac{(\omega_0 t/2)^3}{3!} + \frac{(\omega_0 t/2)^5}{5!} + \dots \right) \\ &= \cos \frac{\omega_0 t}{2} \hat{I} + i \sin \frac{\omega_0 t}{2} \hat{\sigma}^x \\ &\doteq \begin{pmatrix} \cos \frac{\omega_0 t}{2} & i \sin \frac{\omega_0 t}{2} \\ i \sin \frac{\omega_0 t}{2} & \cos \frac{\omega_0 t}{2} \end{pmatrix}. \end{aligned} \quad (9)$$

Here we have made use of the Taylor expansions for $\sin(\omega_0 t/2)$ and $\cos(\omega_0 t/2)$ to sum the infinite series explicitly. For the specific initial state $|\psi(0)\rangle = |+\rangle$, one can employ Eqs. (2) and (3) to obtain

$$\begin{aligned} \langle\hat{S}^x(t)\rangle &= 0, \\ \langle\hat{S}^y(t)\rangle &= \frac{\hbar}{2}\sin\omega_0 t, \\ \langle\hat{S}^z(t)\rangle &= \frac{\hbar}{2}\cos\omega_0 t. \end{aligned} \quad (10)$$

One might recognize these final expressions as a description of the steady Larmor precession exhibited by a spin in a static magnetic field. Larmor precession lies at the heart of nuclear magnetic resonance (NMR) and magnetic resonance imaging (MRI) and is discussed in several widely used texts on quantum mechanics.^{15,19} Notably, the precession frequency ω_0 is independent of the initial spin orientation, only depending on field strength B_0 .

Equations (10) are the types of theoretical predictions we wish to test on a quantum computer. In a quantum computer, the fundamental object is a qubit, which can exist in a superposition of two basis states. These abstract basis states are typically labeled $|0\rangle$ and $|1\rangle$. Using the widely adopted “computational basis,” these basis states take the representations, $|0\rangle \doteq (1, 0)^T$, $|1\rangle \doteq (0, 1)^T$. Thus, a quantum spin is normally represented by a qubit with the mapping

$$\begin{aligned} |+\rangle &\rightarrow |0\rangle, \\ |-\rangle &\rightarrow |1\rangle. \end{aligned} \quad (11)$$

Admittedly, the opposite mapping might reasonably appear more natural based on the labels. However, using Eq. (11), the matrix representations of Pauli operators in the \hat{S}^z eigenbasis will be identical to those in the computational basis, greatly simplifying the design of quantum circuits to simulate spin physics.

Users may access quantum hardware through the Qiskit SDK.¹³ The Qiskit textbook¹⁶ and the IBM Quantum Learning resources²⁰ provide detailed information on using Qiskit to construct quantum circuits, so we only sketch the main steps in translating the physical problem of spin dynamics into a quantum circuit. The reader is referred to other excellent resources^{9,16,20} for additional background information. Sample Jupyter notebooks containing programs used in this work are included in the [supplementary material](#).

Universal quantum computers execute circuits of series of gates that modify the states of the qubits. To extract information about the system, a measurement must be performed. Thus, the three basic stages in a quantum circuit appropriate for simulating the spin dynamics discussed above are (1) state initialization, (2) time evolution of the state, and (3) measurement of the spin components. IBM’s quantum hardware automatically initializes each qubit to the state $|0\rangle$. An arbitrary quantum state of the form in Eq. (2) can be generated by applying the following unitary operator:

$$\hat{U}_3(\theta, \phi, \lambda) \doteq \begin{pmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{pmatrix}. \quad (12)$$

This \hat{U}_3 gate is available as a three-parameter gate in Qiskit, `u(theta, phi, lambda)`. The operator in Eq. (12) is actually more general than we require to create the state in Eq. (2). Moreover, the careful reader might observe that $\hat{U}_3(\theta, \lambda, \phi)$ is not Hermitian for arbitrary choices of θ, ϕ, λ . Operators corresponding to physical observables must be Hermitian, but unitarity ($\hat{O}\hat{O}^\dagger = \hat{O}^\dagger\hat{O} = \hat{I}$) is the only restriction on operators that change the state of a quantum system. This operation provides an efficient mechanism for generating a qubit in an arbitrary state $|\psi(\theta, \phi)\rangle = \hat{U}_3(\theta, \phi, 0)|0\rangle$.[†] Conveniently, time evolution is performed by noting that our expression for the time evolution operator $\hat{U}(t)$ in Eq. (8) may also be written in terms of the \hat{U}_3 gate as $\hat{U}(t) = \hat{U}_3(\omega_0 t, +\pi/2, -\pi/2)$.

[†]Note that λ can actually take on *any* value for this operation. When using this gate in a circuit to generate $|\psi(\theta, \phi)\rangle$, a value must be specified for λ . For simplicity, we choose $\lambda = 0$.

When using IBM quantum devices, measurements can only be performed in the computational basis. That is, for a qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, a measurement will only return 0 or 1 corresponding to the measured state being $|0\rangle$ or $|1\rangle$. This constraint suggests that a single measurement operation can only measure \hat{S}^z . Measurement of spin projection along other directions is made possible by rotating the system appropriately before making a measurement in the computational basis.^{9,14} Within Qiskit, one may perform such an operation through the use of rotation gates, $\hat{R}_\alpha(\theta)$, for $\alpha = x, y, z$. With ordinary (classical) three-dimensional vectors in physical space, a rotation is performed by applying a standard rotation matrix. For example, to rotate a three-dimensional vector by an angle θ about the x -axis, the appropriate transformation is given by

$$\mathbf{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}. \quad (13)$$

For quantum mechanical systems, rotations can be more subtle. While we associate spin components with directions in real space, the two-component object representing a spin exists in the abstract, Bloch sphere rather than in real space. Clearly, the three-dimensional rotation matrices cannot be applied to such a state. Rotation of quantum states is not commonly included in a traditional undergraduate course on quantum mechanics, so we quote the needed result and demonstrate its plausibility. Suppose we take a state $|\psi\rangle$ and perform a rotation about the α -axis ($\alpha = x, y, z$) by angle θ , transforming the state to $|\psi'\rangle = \hat{R}_\alpha(\theta)|\psi\rangle$. The appropriate operator, derivable from commutation relations of the spin-1/2 operators, is $\hat{R}_\alpha(\theta) = \exp[-i\hat{\sigma}^\alpha\theta/2]$.^{17,18} The diligent reader can use Eq. (3) and the method of matrix exponentiation discussed above to show

$$\hat{R}_\alpha(\theta) = \hat{I} \cos \frac{\theta}{2} - i\hat{\sigma}^\alpha \sin \frac{\theta}{2}. \quad (14)$$

If we compute the expectation value of, say, \hat{S}^z in a rotated state $|\psi'\rangle = \hat{R}_x(\theta)|\psi\rangle$, we get

$$\begin{aligned} \langle \psi' | \hat{S}^z | \psi' \rangle &= \langle \psi | \hat{R}_x^\dagger(\theta) \hat{S}^z \hat{R}_x(\theta) | \psi \rangle \\ &= \cos \theta \langle \psi | \hat{S}^z | \psi \rangle + \sin \theta \langle \psi | \hat{S}^y | \psi \rangle. \end{aligned} \quad (15)$$

The fastest way to arrive at Eq. (15) is to employ the commutation relation $\hat{\sigma}^z \hat{\sigma}^x - \hat{\sigma}^x \hat{\sigma}^z = 2i\hat{\sigma}^y$, the anticommutation relation $\hat{\sigma}^x \hat{\sigma}^z = -\hat{\sigma}^z \hat{\sigma}^x$, and the fact that $[\hat{\sigma}^z]^2 = \hat{I}$. These properties are straightforward to verify with the explicit matrix representations of the Pauli operators in Eq. (3). Alternatively, it is also instructive to evaluate both sides of Eq. (15) by brute force for an arbitrary state $|\psi\rangle \doteq (\alpha, \beta)^T$. Regardless of the overall strategy, double-angle formulas again prove essential to the simplification.

To interpret Eq. (15), let $\langle \hat{O} \rangle \equiv \langle \psi | \hat{O} | \psi \rangle$ denotes the expectation value of \hat{O} in the original (unrotated) state. Then, consulting Eq. (13), the right-hand side of Eq. (15) represents the z -component of the vector formed by rotating the vector $(\langle \hat{S}^x \rangle, \langle \hat{S}^y \rangle, \langle \hat{S}^z \rangle)^T$ about the x -axis by angle θ . That is, rotating the state about the x -axis before measurement of \hat{S}^z results in an effective spin measurement of the unrotated system about an axis $\hat{\mathbf{n}} = \sin \theta \hat{\mathbf{y}} + \cos \theta \hat{\mathbf{z}}$. In

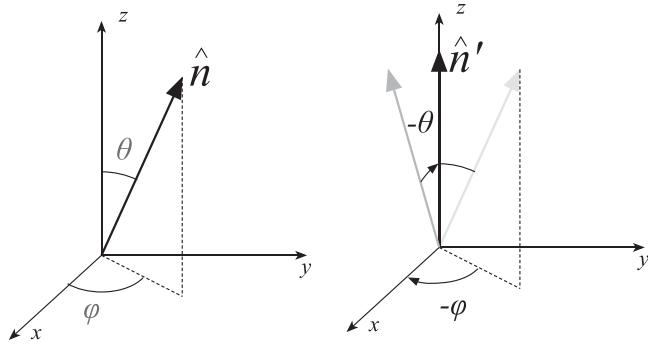


Fig. 1. To measure spin about the axis $\hat{\mathbf{n}}$, one must perform the required rotations to bring $\hat{\mathbf{n}}$ into alignment with the z -axis (the computational basis). Given polar-spherical angles θ , ϕ corresponding to $\hat{\mathbf{n}}$, one finds $\hat{\mathbf{n}}' \equiv \hat{\mathbf{z}} = \mathbf{R}_y(-\theta)\mathbf{R}_z(-\phi)\hat{\mathbf{n}}$.

summary, to measure spin about an arbitrary axis $\hat{\mathbf{n}}$, one can measure \hat{S}^z after performing the rotation on the quantum state that corresponds to the classical rotation that brings $\hat{\mathbf{n}}$ into alignment with the z -axis. The reader is invited to check that $\mathbf{R}_x(\theta)\hat{\mathbf{n}} = \hat{\mathbf{z}}$.

Conveniently, rotation gates are accessible directly as predefined gates in Qiskit (e.g., `rx(theta)`). The motivated reader is encouraged to work out matrix representations for $\hat{R}_x(\theta) = \exp[-i\hat{\sigma}^x\theta/2]$ and verify that these non-Hermitian matrices are special cases of $\hat{U}_3(\theta, \lambda, \phi)$ for particular choices of θ , λ , ϕ . An algorithmic method for performing arbitrary rotations involving two rotations about fixed axes is depicted in Fig. 1. It is also instructive to show explicitly that $\hat{U}_3(\theta, \phi, 0) = e^{i\phi/2}\hat{R}_z(\phi)\hat{R}_y(\theta)$, so that one may view the unitary transformation $|\psi(\theta, \phi)\rangle = \hat{U}_3(\theta, \phi, 0)|+\rangle$ in terms of the successive rotations that map $|+\rangle$ to any point on the Bloch sphere. The extra (global) phase $e^{i\phi/2}$ cannot be measured and has no physical effects. This equivalence is a particular example of the general statement that *any* unitary, single-qubit gate (or sequence of such gates) can be written as $\hat{U}_3(\theta, \phi, \lambda)$ for some choice of parameters θ , ϕ , λ .

Regarding measurements, we will need only to measure spin projections along the $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ directions. With this convention, these required rotations are encapsulated in the statement, $\hat{\mathbf{z}} = \mathbf{R}_y(-\pi/2)\hat{\mathbf{x}} = \mathbf{R}_y(-\pi/2)\mathbf{R}_z(-\pi/2)\hat{\mathbf{y}}$. It is also possible to use the single rotation $\hat{\mathbf{z}} = \mathbf{R}_x(\pi/2)\hat{\mathbf{y}}$, but there are no significant gains in result quality for using different rotation “formats.” In practice, Qiskit performs an automatic, optimization procedure that simplifies a circuit as much as possible before execution on real hardware in order to reduce the effects of noise. We also note that if one were interested in the dynamics *after* such a measurement, it would be necessary to perform the corresponding inverse rotation after the measurement but before implementing additional time evolution.

As IBM quantum devices are freely available, jobs sent from an open-access account are often queued—sometimes for significant times. With efficiency in mind, it is helpful to embed three independent, single-spin circuits into a single three-qubit circuit to measure all three spin components in a single job. The circuit shown in Fig. 2(a) creates three copies of this single spin, performs time evolution generated by a single Hamiltonian on each spin, and then measures one of the three spin components on each qubit. Using Qiskit, the

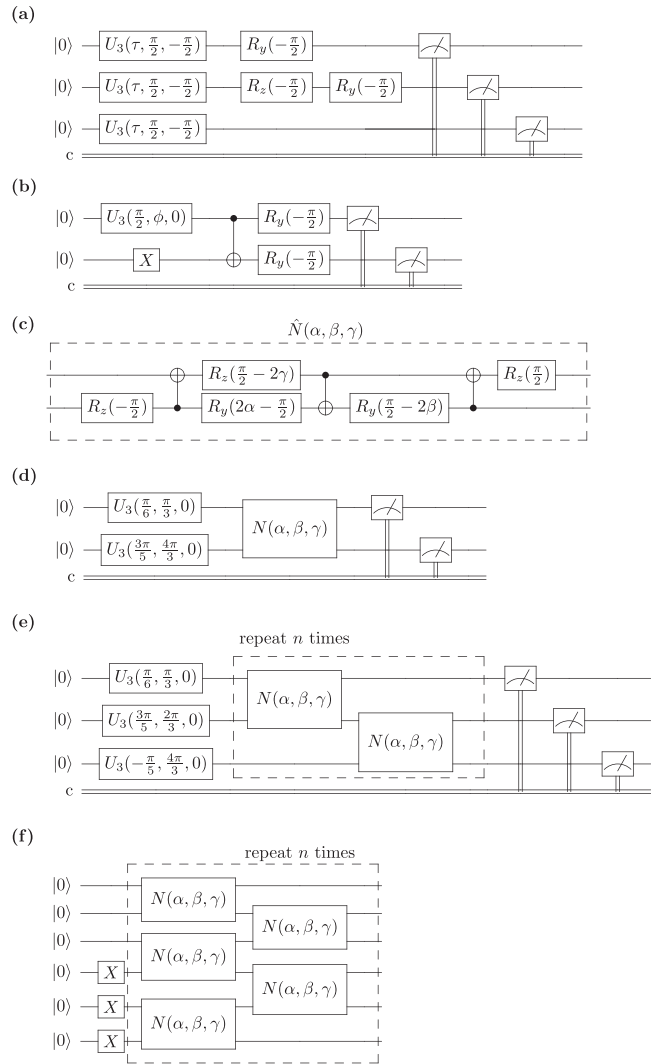


Fig. 2. Quantum circuits used to perform various tasks described in the main text; (a) circuit to perform time evolution and measure three different spin components for three copies of the same single spin; (b) circuit to create the two-spin state $|\psi(\phi)\rangle$ described in Sec. III and measure \hat{S}^x on both spins; (c) circuit representation of $\hat{N}(\alpha, \beta, \gamma)$ defined and used to perform time evolution for two-spin system in Sec. III; (d) circuit to create two-spin state in Eq. (30) and perform time evolution under Hamiltonian \hat{H}_2 (Eq. (26)); (e) circuit for approximate time evolution of a three-spin state with Hamiltonian \hat{H}_N (Eq. (37)); and (f) circuit for initializing a domain wall state for $N = 6$ spins and performing approximate time evolution.

circuit in Fig. 2(a) can be constructed and sent to IBM quantum devices for execution.

When one executes the quantum circuit in Fig. 2(a), the result yields a sequence of three digits, each being either 0 or 1, which is written as a string to the classical register that uses classical bits to store the measurement results. For example, one might obtain the readout ‘100’, indicating that $+\hbar/2$ was obtained for \hat{S}^x and \hat{S}^y , while $-\hbar/2$ was obtained for \hat{S}^z .[‡] The framework of quantum mechanics does

[‡]Qiskit uses a qubit labeling convention, which is opposite to the common convention used for labeling many-particle states in physics. For example, Qiskit labels three-qubit states according to $|\psi\rangle = |q_2q_1q_0\rangle$, where q_0 refers to the top qubit in the circuit diagram and q_2 refers to the bottom qubit in the diagram. We employ the standard notation convention from physics in this work (e.g., labeling three-qubit states by $|q_0q_1q_2\rangle$) and account for the labeling reversal when analyzing counts.

not allow for exact predictions of the outcome of a single such experiment. We are only able to compute statistical quantities such as probabilities and expectation values, which give us information about the averages of many repetitions of an experiment performed on identically prepared systems. Typically, one executes a circuit for some large number of executions, or “shots,” so that experimental averages can be constructed and compared to theoretical predictions (e.g., Eq. (10)). With the release of Qiskit 1.x in 2024, circuit executions on IBM quantum hardware now require the use of Qiskit Runtime²¹ environment. Within this framework, the user selects one of two possible “primitives,” or computational building blocks, for performing an experiment. Both primitives employ additional classical computing resources to optimize a particular job for a given device. The Estimator primitive returns estimates of expectation values for specified operators. This approach can be quite efficient for complicated circuits, but it obscures the actual manner in which these results are obtained. We will focus on the Sampler primitive, which is designed to return a (quasi)-probability distribution of the measurement outcomes. This approach has additional pedagogical value in requiring the user to reconstruct experimental averages from the low-level “counts” that accumulate over successive shots. These counts correspond to the number of times each state was observed during the repeated circuit executions. Examples of using the Estimator to calculate spin expectation values directly are contained in the [supplementary material](#).

When executing a circuit using the Sampler primitive, the user can access the counts in the form of Python dictionary. This dictionary consists of pairs of variables, a bitstring (e.g., ‘101’) labeling each state, and a corresponding integer value representing the counts associated with that state. From this information, we construct spin expectation values as follows: let $n_{q_2q_1q_0}$ be the counts for state $|q_2q_1q_0\rangle$, where $q_{0,1,2} \in \{0,1\}$. Taking $\langle \hat{S}^y \rangle$ as an example, we have eight counts, $n_{000}, n_{001}, \dots, n_{111}$. A weighted average of this spin component is given by

$$\langle \hat{S}^y \rangle = \frac{1}{N_{\text{shots}}} \left[(n_{000} + n_{001} + n_{100} + n_{101}) \left(+\frac{\hbar}{2} \right) + (n_{010} + n_{011} + n_{110} + n_{111}) \left(-\frac{\hbar}{2} \right) \right]. \quad (16)$$

Note that the counts for which the middle qubit is measured to be $|0\rangle$ (corresponding to spin “up”) weight the contribution of $+\hbar/2$, while those in which the middle qubit is in $|1\rangle$ (corresponding to spin “down”) weight the contribution of $-\hbar/2$. Analogous averaging can be performed on the other qubits to obtain experimental values for the expected averages $\langle \hat{S}^x \rangle$, $\langle \hat{S}^z \rangle$. Everything discussed so far, from circuit construction to circuit execution, could be performed using the graphical user interface of the IBM Quantum Composer,²² accessible to anyone with a free IBM Quantum account. In that environment, the results of circuit executions are returned as probability distributions, and it becomes cumbersome to perform any processing of count data for all but the simplest of circuits. Employing the Qiskit SDK within a Jupyter notebook (i.e., a Python environment) allows for straightforward extraction of the counts for computation of expectation values as described above. A more compelling reason for accessing IBM devices from within a Jupyter notebook is that it enables users to bundle much more

complicated jobs into a single “primitive unified bloc” (PUB) to be sent to a quantum computer as a single unit. Since a quantum measurement affects the state of the system, the circuit in Fig. 2(a) can only measure the spin components at a single time t . By making multiple copies of the circuit using a loop over a range of values for time t or by employing built-in Qiskit techniques to rerun a circuit with different parameter values, one can obtain the spin measurements over a range of time samples. The results from running the circuit in Fig. 2(a) on `ibmq_casablanca` (v 1.2.35, a Falcon r4H processor) over a full cycle $0 \leq t \leq 2\pi/\omega_0$ with 75 time samples and 8192 shots per sample are shown in Fig. 3. Error bars shown are statistical estimates given by $\sigma/\hbar \approx 1/2\sqrt{N_{\text{shots}}}$.

The data in Fig. 3 show better agreement with theory when $\langle \hat{S}^z \rangle$ is closer to $+\hbar/2$ than $-\hbar/2$. The simple reason for this uneven signal quality is rooted in the physical realization of the two qubit states, $|0\rangle$ and $|1\rangle$. IBM’s devices employ transmon qubits²³ in which $|0\rangle$ represents the ground state of a small LC circuit (i.e., a quantum harmonic oscillator) with controllable anharmonicity due to a (nonlinear) Josephson junction. This anharmonicity allows for isolation of the ground state and first excited state ($|1\rangle$) as the two states of a physical qubit. As $|1\rangle$ is an excited state, it decays to $|0\rangle$ with characteristic timescale $T_1 \sim 100 \mu\text{s}$, resulting in the so-called “spin-flip” errors when the circuit’s execution time becomes significant compared to T_1 . Circuit execution time is linked to the time required to execute the circuit’s gates rather than the physical time we are simulating. Thus, one expects higher rates of errors when the qubit state is placed in the state $|1\rangle$, which we have mapped to $|-\rangle$.

The process of simulating time evolution on a quantum computer and constructing observables from the raw counts is an instructive process for students of quantum mechanics. Common student difficulties in interpreting expectation values, measurement, and the general mathematical structure of quantum mechanics are well documented.^{24,25} The opportunity to construct operator expectation values from experimental measurements of common textbook-level systems gives students a different perspective of these abstract concepts. Though we have only considered the simplest case of

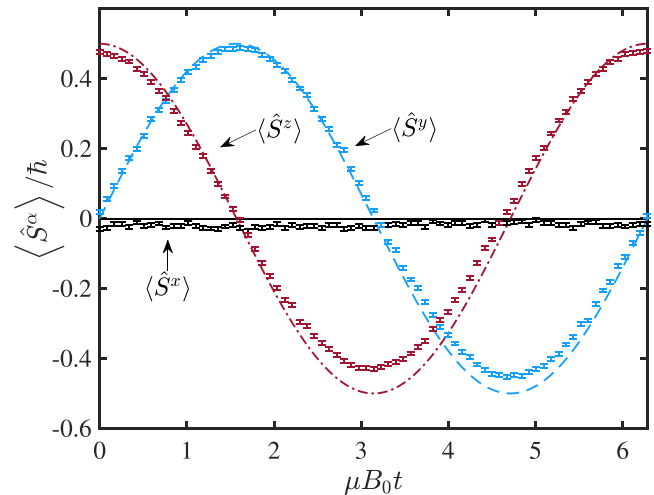


Fig. 3. Theory (lines) and actual results (markers with error bars) for time evolution of spin components with $\mathbf{B} = B_0 \hat{x}$ and $|\psi(0)\rangle = |+\rangle$ using circuit shown in Fig. 2(a) on `ibmq_casablanca` (v 1.2.35, a Falcon r4H processor).

Larmor precession, probing dynamics under time-dependent magnetic fields would require only modest modification to the basic scheme outlined in this section.¹⁴ In the remainder of this paper, we shift focus to investigating the dynamics of multiple, interacting spins.

III. MULTIPLE SPINS

The main focus of this section is to explore the dynamics of interacting spins with quantum computers. Multi-particle quantum states give rise to new phenomena—such as entanglement—which are not present at the single-particle level. Before exploring the dynamics of multiple, interacting spins, we briefly examine some basic aspects of entangled spin states. We demonstrate that quantum computers can be used to distinguish between different types of entangled states by effectively measuring the total spin of the combined state. Additionally, this first exploration provides a simplified setting to introduce two-qubit gates, which will be required for performing time evolution. Subsequent subsections treat dynamics of interacting spins for system sizes $N = 2$ and $N = 3$ separately. Subsection III D outlines how to extend the approach for $N = 3$ to larger systems with a specific focus on obtaining evidence for a phase transition in a particular spin model.

A. Entanglement and spin

The most general single-spin state (Eq. (2)) is represented by a two-component vector. For N spins, a state is represented by a 2^N -component vector with the most general state for $N = 2$ taking the form

$$|\Psi_2\rangle \propto c_{++}|+\rangle \otimes |+\rangle + c_{+-}|+\rangle \otimes |-\rangle + c_{-+}|-\rangle \otimes |+\rangle + c_{--}|-\rangle \otimes |-\rangle, \quad (17)$$

where $c_{\pm\pm}$ are the complex constants. The symbol \otimes denotes the tensor product operation. This construction reflects that a “product” of single-spin operators should act on a product of single-spin kets as $(\hat{A} \otimes \hat{B})(|\psi_1\rangle \otimes |\psi_2\rangle) = (\hat{A}|\psi_1\rangle) \otimes (\hat{B}|\psi_2\rangle)$. In the same way, the notation $| - + + \rangle$ for labeling a state of three spins in Sec. II is actually shorthand for $|-\rangle \otimes |+\rangle \otimes |+\rangle$. Another common convention¹⁹ for writing the same state is $|-\rangle_1|+\rangle_2|+\rangle_3$, where site labels are included explicitly and the tensor product is “hidden.” We will not need an explicit representation of this operation in terms of matrices (e.g., the Kronecker product), so this notation merely provides a careful prescription for keeping track of the particular spins upon which an operator acts. The case of two spins provides simplest setting for probing quantum entanglement and quantum correlations. To this end, let us use the common shorthand (e.g., $|+ - \rangle \equiv |+\rangle \otimes |-\rangle$) and consider the family of states defined by

$$|\psi(\phi)\rangle = \frac{1}{\sqrt{2}}(|+ - \rangle + e^{i\phi}|- + \rangle), \quad (18)$$

for some real phase angle ϕ . This particular family of states is interesting because it interpolates smoothly between two particular, entangled states, such that

$$\begin{aligned} |\Psi^+\rangle &= |\psi(0)\rangle = \frac{1}{\sqrt{2}}(|+ - \rangle + |- + \rangle), \\ |\Psi^-\rangle &= |\psi(\pi)\rangle = \frac{1}{\sqrt{2}}(|+ - \rangle - |- + \rangle). \end{aligned} \quad (19)$$

Saying that $|\Psi^\pm\rangle$ are entangled states is equivalent to the statement that there are no single-spin states $|\varphi_{1,2}\rangle$ for which one may write $|\Psi^\pm\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle$. That is, the state of one spin cannot be fully specified independently of the state of the other spin. Despite their strikingly similar structure, the states $|\Psi^\pm\rangle$ are physically distinct. Specifically, $|\Psi^+\rangle$ is a spin-1 state and $|\Psi^-\rangle$ is a spin-0 state,¹⁹ and a measurement of total spin should distinguish between these two states. This is a particular example of the general statement that, while the overall phase of a state is not measurable, the relative phase of a superposition state (i.e., e^{i0} vs $e^{i\pi}$) is measurable.

Measuring the total spin of a two-spin system is a more complicated process than measuring single-spin components, as described in Sec. II. To represent a total-spin measurement in terms of single-spin properties that can be measured directly, let us write the total, squared-spin operator as

$$\hat{S}_{\text{tot}}^2 = (\hat{S}_1^x + \hat{S}_2^x)^2 + (\hat{S}_1^y + \hat{S}_2^y)^2 + (\hat{S}_1^z + \hat{S}_2^z)^2. \quad (20)$$

Here we make use of the common convention $\hat{S}_1^x = \hat{S}^x \otimes \hat{I}$, $\hat{S}_2^x = \hat{I} \otimes \hat{S}^x$ for brevity. Squaring the first term yields quantities such as $[\hat{S}_1^x]^2$, $[\hat{S}_2^x]^2$, and $\hat{S}_1^x \hat{S}_2^x$. Using the tensor-product multiplication rule,

$$\begin{aligned} [\hat{S}_1^x]^2 &= [\hat{S}_2^x]^2 = \frac{\hbar^2}{4} \hat{I} \otimes \hat{I}, \\ \hat{S}_1^x \hat{S}_2^x &= \hat{S}^x \otimes \hat{S}^x, \end{aligned} \quad (21)$$

where \hat{S}^x without label is the ordinary spin-1/2 operator $(\hbar/2)\hat{\sigma}^x$. Here we have employed the useful property of Eq. (3) that $[\hat{S}^x]^2 = [\hat{S}^y]^2 = [\hat{S}^z]^2 = (\hbar^2/4)\hat{I}$. Collecting the remaining terms and taking an expectation value, we obtain

$$\langle \hat{S}_{\text{tot}}^2 \rangle = \frac{3}{2}\hbar^2 + 2\langle \hat{S}^x \otimes \hat{S}^x \rangle + 2\langle \hat{S}^y \otimes \hat{S}^y \rangle + 2\langle \hat{S}^z \otimes \hat{S}^z \rangle, \quad (22)$$

which is valid for the expectation value taken with respect to any state. The quantities $\langle \hat{S}^\alpha \otimes \hat{S}^\alpha \rangle$ are sometimes called spin correlation functions and have been measured for various entangled, two-spin states in Ref. 9. Equation (22) demonstrates that we can effectively measure the expectation value of total (squared) spin for a two-spin state by measuring these spin correlation functions. For the entangled states $|\Psi^\pm\rangle$, each contribution of $\langle \hat{S}^\alpha \otimes \hat{S}^\alpha \rangle$ yields four terms, which can be evaluated according to the rule $(\langle \varphi_1 | \otimes \langle \varphi_2 |) \hat{S}^\alpha \otimes \hat{S}^\alpha (| \varphi_1 \rangle \otimes | \varphi_2 \rangle) = \langle \varphi_1 | \hat{S}^\alpha | \varphi_1 \rangle \langle \varphi_2 | \hat{S}^\alpha | \varphi_2 \rangle$. It is also known from the general theory of spin angular momentum¹⁹ that eigenstates $|s\rangle$ of the spin operator in Eq. (20) satisfy

$$\hat{S}_{\text{tot}}^2 |s\rangle = \hbar^2 s(s+1) |s\rangle, \quad (23)$$

where $s = 0, 1/2, 1, \dots$ gives the maximal projection of spin on any axis in units of \hbar . What remains is to demonstrate that a quantum computer can effectively measure the appropriate values of s for $|\Psi^+\rangle$ ($s = 1$) and $|\Psi^-\rangle$ ($s = 0$).

The state $|\psi(\phi)\rangle$ can be initialized for arbitrary ϕ by application of three successive gates. First, one applies a unitary gate to the first qubit to place it in a superposition state with a phase difference of ϕ such that

$$\hat{U}_3\left(\frac{\pi}{2}, \phi, 0\right)|00\rangle \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + e^{i\phi}|10\rangle). \quad (24)$$

Next, applying a logical NOT operation $\hat{X} = \hat{U}_3(\pi, 0, \pi)$ to the second qubit to yield the state $(1/\sqrt{2})(|01\rangle + e^{i\phi}|11\rangle)$. Finally, one applies a controlled NOT (CNOT) operation with the first qubit as the control and the second qubit as the target. Specifically, a CNOT gate flips the target qubit *if* the control qubit is in the state $|1\rangle$. In the circuit diagram, the target qubit is denoted by the “+” and connected to the control qubit. The action of this gate on two qubit states can be summarized as

$$\begin{aligned} \text{CNOT}[|0\rangle \otimes |0\rangle] &= |0\rangle \otimes |0\rangle, & \text{CNOT}[|0\rangle \otimes |1\rangle] &= |1\rangle \otimes |1\rangle, \\ \text{CNOT}[|1\rangle \otimes |0\rangle] &= |1\rangle \otimes |0\rangle, & \text{CNOT}[|1\rangle \otimes |1\rangle] &= |0\rangle \otimes |1\rangle, \end{aligned} \quad (25)$$

with the qubits arranged in the form $|\text{target}\rangle \otimes |\text{control}\rangle$.

Two-qubit gates such as CNOT typically lead to errors, which are an order of magnitude larger than those associated with single-qubit gates.²⁶ Consequently, it is desirable to use the minimum number of CNOT gates required for a given calculation, and we should expect larger errors in interacting systems than the single spin systems considered in Sec. II.

A circuit to prepare $|\psi(\phi)\rangle$ and measure $\langle \hat{S}^x \otimes \hat{S}^x \rangle$ is depicted in Fig. 2(b). As in the single-spin case, we cannot measure all three spin components with a single realization of the state. Thus, we repeat the entire initialization, time evolution, and measurement process for the other spin components with 8192 shots per component measurement for each of the 75 samples of $\phi \in [0, \pi]$. Computing the spin correlation functions entails averaging the product of spin components. For example, suppose execution of the circuit in Fig. 2(b) yields n_{00} counts of state $|00\rangle$, ..., n_{11} counts for state $|11\rangle$. The corresponding spin correlation function is given by $\langle \hat{S}^x \otimes \hat{S}^x \rangle = (\hbar^2/4)(n_{00} + n_{11} - n_{01} - n_{10})/N_{\text{shots}}$. The results obtained from the device `ibmq_oslo` are shown in Fig. 4. The measured expectation value $\langle \hat{S}_{\text{tot}}^2 \rangle$ is

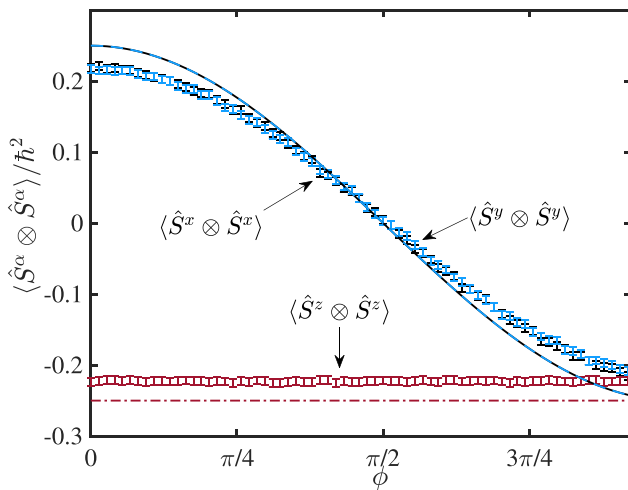


Fig. 4. Measurements of $\langle \hat{S}^\alpha \otimes \hat{S}^\alpha \rangle$ (markers with error bars) for $\alpha = x, y, z$ shown alongside theoretical predictions (lines) for the state $|\psi\rangle = (|+-\rangle + e^{i\phi}|--\rangle)/\sqrt{2}$, with $0 \leq \phi \leq \pi$. The case $\phi = 0$ corresponds to the $s = 1$ state $|\Psi^+\rangle$, whereas $\phi = \pi$ corresponds to the $s = 0$ state $|\Psi^-\rangle$. Data were collected from the device `ibmq_oslo` (v1.0.8, a Falcon r5.11H processor). Results for $\alpha = x, y$ are virtually indistinguishable, in agreement with theory.

$(1.921 \pm 0.003)\hbar^2$ and $(0.189 \pm 0.003)\hbar^2$ for states $|\Psi^+\rangle$ and $|\Psi^-\rangle$, respectively. From Eq. (23), these measurements should correspond $2\hbar^2$ and 0. Due to the higher errors associated with CNOT gates, measured correlations on actual devices are typically smaller in magnitude than theoretical predictions when the expectation value is high and larger when the expectation value is low.⁹ However, these measurements are still roughly consistent with the predicted values, given the significant noise.

B. Dynamics for $N=2$

For the simplest case of $N = 2$ interacting spins, a general two-body interaction Hamiltonian is

$$\hat{H}_2 = -J_x \hat{S}^x \otimes \hat{S}^x - J_y \hat{S}^y \otimes \hat{S}^y - J_z \hat{S}^z \otimes \hat{S}^z, \quad (26)$$

where $J_{x,y,z}$ are the couplings. Additional, “mixed” terms such as $\hat{S}^x \otimes \hat{S}^y$ could also be included, but Eq. (26) is sufficient to give rise to the rich dynamics discussed in this section. The structure of Eq. (26) is motivated by the quantum mechanical exchange interactions,²⁷ which (for $J_{x,y,z} > 0$) predicts a lower interaction energy for aligned magnetic dipoles (i.e., spins). For now, we take the couplings as arbitrarily tunable parameters to explore how to simulate such a system on a quantum computer. Later in this section, we return to the interesting question of how the coupling values affect the dynamics.

To explore dynamics, we must compute the time evolution operator $\hat{U}(t) = \exp[-i\hat{H}t]$. For brevity, we employ units in which $\hbar \rightarrow 1$ for the remainder of this paper. For a Hamiltonian with the basic structure of Eq. (26), we need a circuit representation of the operator $\hat{N}(\alpha, \beta, \gamma) \equiv \exp[i\alpha\hat{\sigma}^x \otimes \hat{\sigma}^x + i\beta\hat{\sigma}^y \otimes \hat{\sigma}^y + i\gamma\hat{\sigma}^z \otimes \hat{\sigma}^z]$ for some constants α, β, γ . Setting $\alpha = J_x/4$, $\beta = J_y/4$, and $\gamma = J_z/4$, the full time evolution operator $\hat{U}(t)$ is given conveniently by exactly $\hat{N}(\alpha, \beta, \gamma)$. To motivate the structure of a circuit representation for $\hat{N}(\alpha, \beta, \gamma)$, let us consider the simpler case of the operator

$$\hat{O}_Z = \exp[i\gamma\hat{\sigma}^z \otimes \hat{\sigma}^z]. \quad (27)$$

Working with four basis states, we wish to obtain a circuit that performs the following operations:

$$\begin{aligned} \hat{O}_Z|++\rangle &= e^{i\gamma}|++\rangle, \\ \hat{O}_Z|+-\rangle &= e^{-i\gamma}|+-\rangle, \\ \hat{O}_Z|-+\rangle &= e^{-i\gamma}|-+\rangle, \\ \hat{O}_Z|--\rangle &= e^{i\gamma}|--\rangle. \end{aligned} \quad (28)$$

Applying the circuit in Fig. 5(a) to all four basis states using the matrix representation in the $\{|0\rangle, |1\rangle\}$ basis (or, equivalently, the $\{|\pm\rangle\}$ basis),

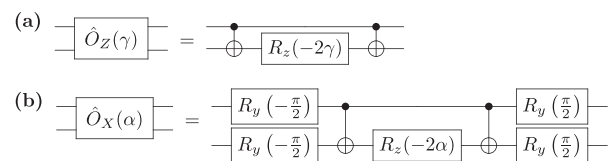


Fig. 5. (a) Circuit representation of operator $\hat{O}_Z(\gamma)$. (b) Circuit representation of $\hat{O}_X(\alpha)$.

$$\hat{R}_z(\phi) = \exp[-i\phi\hat{\sigma}^z/2] = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix}, \quad (29)$$

one recovers the exact transformation defined by Eq. (28). One could show this algebraically or explore experimentally by using the IBM Quantum Composer²² to drag and drop circuit elements and explore the output for all four input basis states individually. The computational basis treats the z direction preferentially, but simple modifications allow us to realize other operators such as $\hat{O}_X(\alpha) = \exp[i\alpha\hat{\sigma}^x \otimes \hat{\sigma}^x]$ and $\hat{O}_Y(\beta) = \exp[i\beta\hat{\sigma}^y \otimes \hat{\sigma}^y]$. One approach is to first rotate the state before performing the same operation and then to perform the opposite rotation to “undo” the initial rotation. For the case of $\hat{O}_X(\alpha)$, we can apply the methods of Sec. II to first rotate each qubit so that the $+x$ direction aligns with $|0\rangle$. Specifically, one can apply $\hat{R}_y(-\pi/2)$ before the CNOT and $\hat{R}_z(-2\alpha)$ operations and then apply $\hat{R}_y(+\pi/2)$, as shown in Fig. 5(b).

Another set of carefully chosen rotations also yields a circuit representation for $\hat{O}_Y(\beta)$. Unfortunately, obtaining the full time-evolution operator $\hat{N}(\alpha, \beta, \gamma)$ is *not* as simple as stitching these three circuits together. In general, $e^{\hat{A}+\hat{B}} \neq e^{\hat{A}}e^{\hat{B}}$ when \hat{A} and \hat{B} do not commute and different spin-component operators at the same site do not commute. When using an approximation scheme (such as the Lie–Trotter decomposition discussed in Sec. III B), one only advances over small time steps, and this level of error might be acceptable. However, in the case of two interacting spins, it is possible²⁶ to construct an exact representation for $\hat{N}(\alpha, \beta, \gamma)$, which is shown in Fig. 2(c). Note that this circuit, still consisting of only CNOT gates and rotations, requires only three CNOT gates instead of the *six* one would need to apply the sequence of circuits for $\hat{O}_X(\alpha)$, $\hat{O}_Y(\beta)$, and $\hat{O}_Z(\gamma)$. Vatan and Williams²⁸ have shown that any two-qubit gate (e.g., $\hat{N}(\alpha, \beta, \gamma)$) can be constructed using no more than three CNOT gates, so this representation turns out to be optimal. We will now proceed to use this circuit to perform time evolution in two-spin systems.

As a specific example, let us take $J_x = 0.5J$, $J_y = -0.45J$, $J_z = 0.25J$ for some arbitrary energy scale J and simulate time evolution on an actual quantum computer. We take the following initial single-qubit states:

$$\begin{aligned} |\psi_1\rangle &= \left(\cos \frac{\pi}{12}\right)|+\rangle + \left(e^{i\pi/3} \sin \frac{\pi}{12}\right)|-\rangle, \\ |\psi_2\rangle &= \left(\cos \frac{3\pi}{10}\right)|+\rangle + \left(e^{i4\pi/3} \sin \frac{3\pi}{10}\right)|-\rangle, \end{aligned} \quad (30)$$

which represent generic superposition states. Both the coupling parameters J_α and these initial spin states $|\psi_{1,2}\rangle$ are chosen to have no obvious spin structure and provide a representative example of generic dynamics. The initial two-spin state is $|\psi(0)\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. Dynamics generated by $\hat{H} = \hat{H}_2$ (Eq. (26)) with $J_x = 0.5J$, $J_y = -0.45J$, $J_z = 0.25J$, is simulated using the circuit shown in Fig. 2(d) on the device `ibmq_bogota` (v1.4.50, a Falcon r4L processor), which measures $\langle \hat{S}_{1,2}^z \rangle$ at a particular value of t . Two additional copies of this circuit with appropriate rotation gates for measuring the orthogonal spin-component expectation values are also used. With 8192 shots per time sample and 100 total time samples, each of the three circuits is executed a total of 100×8192 times to generate the results shown in Fig. 6.

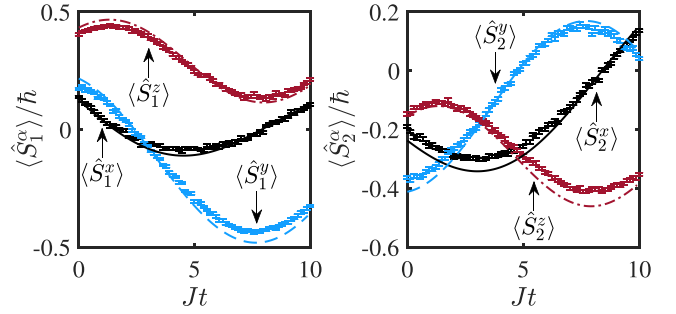


Fig. 6. Time-dependence of spin expectation values (markers with error bars) and theoretical predictions (lines) for an interacting, two-spin system using `ibmq_bogota` to simulate the model in Eq. (26) with the initial state given by Eq. (30).

The theoretical predictions are obtained by calculating $|\psi(t)\rangle = e^{-i\hat{H}_2 t}|\psi(0)\rangle$ via an explicit exponentiation of the Hamiltonian matrix with the function `expm()` in MATLAB.²⁹ A MATLAB script that computes the theoretical predictions is included in the [supplementary material](#).

In addition to computing spin expectation values, we can compute other observables directly from the counts used to produce Fig. 6. In particular, one may use the counts to compute basic spin correlation functions, $\langle \hat{S}^\alpha \otimes \hat{S}^\beta \rangle$ for $\alpha, \beta \in \{x, y, z\}$, as described in Subsection III A. Such correlations give us directly the expected value of \hat{H} , since

$$\langle \hat{H} \rangle = -J_x \langle \hat{S}^x \otimes \hat{S}^x \rangle - J_y \langle \hat{S}^y \otimes \hat{S}^y \rangle - J_z \langle \hat{S}^z \otimes \hat{S}^z \rangle. \quad (31)$$

In Fig. 7, we show the time evolution of $\langle \hat{H} \rangle$ and $\langle \hat{S}_1^z + \hat{S}_2^z \rangle$ for two different sets of parameters in Eq. (26). As the Hamiltonian \hat{H}_2 (Eq. (26)) describes a closed system with no explicit time dependence (e.g., \hat{H}_2 possesses time-translation invariance), $\langle \hat{H} \rangle$ should be constant in time. This statement is an example of Noether’s theorem,¹⁵ which states that conservation laws follow from symmetries. For $J_x = J_y$, rotational invariance of \hat{H}_2 (Eq. (26)) leads to the conservation of the total spin projection along the z -axis. Accordingly, one expects $\langle \hat{S}_1^z + \hat{S}_2^z \rangle$ to remain constant when $J_x = J_y$ but possibly vary when $J_x \neq J_y$. In both cases, $\langle \hat{H} \rangle$ should be constant in time. The initial state for both cases is given by Eq. (30).

As noted previously, the two-qubit CNOT gates used to define the initial state and perform time evolution have significantly higher error rates than single qubit gates. These

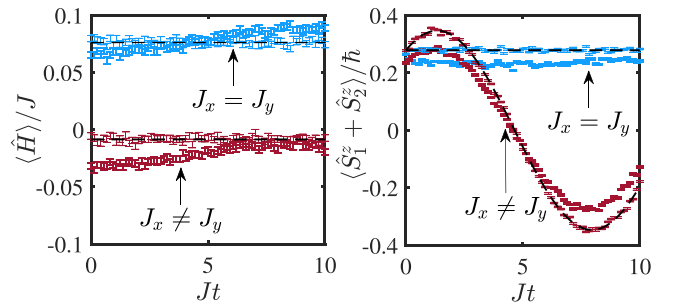


Fig. 7. Expectation value of energy, $\langle \hat{H} \rangle$ (left) and total z -projection of spin, $\langle \hat{S}_1^z + \hat{S}_2^z \rangle$ (right) as functions of time for the initial state in Eq. (30) for two sets of parameters ($J_x = J_y$ and $J_x \neq J_y$). For $J_x = J_y$, both energy and $\langle \hat{S}_1^z + \hat{S}_2^z \rangle$ should be conserved, whereas only energy is conserved for $J_x \neq J_y$. Data (markers with error bars) were collected using `ibmq_lagos` (v1.0.30, a Falcon r5.11H processor). Theoretical predictions (lines) were obtained via exact diagonalization. Circuit simulator results (faint markers) are in good agreement with theory.

errors are compounded when computing averages of products of spin measurements.⁹ Accordingly, we find noticeable discrepancies between the actual results and the theoretical predictions in Fig. 7. For the symmetric case, we take $J_x = J_y = 0.5J$ and $J_z = 0.25J$. In this case, neither $\langle \hat{H} \rangle$ nor $\langle \hat{S}_1^z + \hat{S}_2^z \rangle$ exhibits significant dynamics, with each closely tracking its initial value. Breaking rotational symmetry by setting $J_y = -0.45J \neq J_x$ results in conserved energy, but the total z projection of spin varies significantly over the timescales considered. Qiskit also provides a circuit simulator that produces counts according to exact, theoretical results for a given circuit. The simulator is especially useful for the comparison of experimental results in order to assess the impact of the errors of the physical device. Simulator results (faint markers) are shown to be in good agreement with the theoretical predictions in Fig. 7, indicating that the circuit is performing the desired simulation but that the errors of the device are significant.

Sample Jupyter notebooks for obtaining the results and theoretical predictions in this section are included in the [supplementary material](#). In addition to exploring how parameter choices affect conservation laws, students could also generalize this basic framework to explore other types of systems. For example, in addition to the pure initial states considered in this work, it is possible to compute the dynamics of observables when the initial state is a mixed state, corresponding to thermodynamic equilibrium.³⁰

C. Dynamics for $N=3$

In this subsection, we outline how to investigate larger systems. For $N \geq 3$, it is generally not possible to perform exact time evolution. An exact representation of the time evolution in terms of quantum gates is generally possible only for models that can be diagonalized analytically and are, therefore, “exactly solvable.” Though such models do exist,³¹ the aim of this section is to equip the reader with tactics for the more general situation. The basic strategy, detailed below, uses the Lie–Trotter decomposition³² to break the full time evolution into small-time increments. Other, more efficient approximation schemes, which scale better with increasing circuit depth, are possible,³³ but the basic Lie–Trotter decomposition is chosen as a conceptually simple approach requiring minimal prerequisites to apply. It is also notable that the focus of the 2021 IBM Quantum Open Science Contest³⁴ was to maximize state fidelity using a Lie–Trotter decomposition scheme to perform time evolution in a modest, three-qubit system.

To explore the method, let us consider a time-independent Hamiltonian, which is composed of two parts that do not commute,

$$\hat{H} = \hat{A} + \hat{B}, \quad (32)$$

such that $[\hat{A}, \hat{B}] \neq 0$. The time-evolution operator for this system is given by

$$\hat{U}(t) = \exp[-i\hat{H}t] = \exp[-i\hat{A}t - i\hat{B}t]. \quad (33)$$

For the case of $N = 1, 2$, we computed $\hat{U}(t)$ exactly. In general, such exact computations are not possible. Let us define the operators

$$\hat{U}_A(t) = \exp[-i\hat{A}t], \quad \hat{U}_B(t) = \exp[-i\hat{B}t]. \quad (34)$$

As \hat{A} and \hat{B} do not commute, $\hat{U}(t) \neq \hat{U}_A(t)\hat{U}_B(t)$. However, according to the Lie–Trotter product formula,³² one may write formally

$$\exp[-i(\hat{A} + \hat{B})t] = \lim_{n \rightarrow \infty} \left(\exp[-i\hat{A}t/n] \exp[-i\hat{B}t/n] \right)^n \quad (35)$$

or

$$\hat{U}(t) = \lim_{n \rightarrow \infty} \left(\hat{U}_A\left(\frac{t}{n}\right) \hat{U}_B\left(\frac{t}{n}\right) \right)^n. \quad (36)$$

That is, if we subdivide a time step into n sub-steps, the product of exponentials will converge to the true expression for $\hat{U}(t)$ as $n \rightarrow \infty$. We note that including more Trotter steps requires more circuit gates, leading to an increased circuit depth. Since errors scale with circuit depth, there is a very real tradeoff between using more steps to aid in convergence of the algorithm and using as few steps as possible to minimize accumulation of errors. It has been shown²⁶ that this optimization results in extremely limited timescales for accurate simulation of systems with size $N \sim \mathcal{O}(10)$.

Most IBM devices allow for the application of CNOT gates only between small subsets of nearest-neighbor qubits. Applying CNOT gates to qubits, which are not directly connected, is possible through the application of a sequence of the so-called SWAP gates.¹⁶ In the interest of minimizing circuit depth, we only consider nearest-neighbor interactions in small systems, which can be mapped to physically connected qubits. In this section, we consider the many-body ($N > 2$) generalization of \hat{H}_2 (Eq. (26)) with only nearest-neighbor interactions,

$$\hat{H}_N = \sum_{j=1}^{N-1} \left[-J_x \hat{S}_j^x \hat{S}_{j+1}^x - J_y \hat{S}_j^y \hat{S}_{j+1}^y - J_z \hat{S}_j^z \hat{S}_{j+1}^z \right]. \quad (37)$$

Here, we employ the shorthand $\hat{S}_j^\alpha \hat{S}_{j+1}^\alpha \equiv \hat{I} \otimes \cdots \otimes \hat{I} \otimes \hat{S}_j^\alpha \otimes \hat{S}_{j+1}^\alpha \otimes \hat{I} \otimes \cdots \otimes \hat{I}$. Such nearest-neighbor models are surprisingly useful tools for describing certain interacting, condensed-matter systems. To apply the Trotter decomposition to the time-evolution operator, let us define $\hat{h}_{jk} = -(J_x \hat{S}_j^x \hat{S}_k^x + J_y \hat{S}_j^y \hat{S}_k^y + J_z \hat{S}_j^z \hat{S}_k^z) \delta_{j+1,k}$, so that the time evolution operator can be written as

$$\hat{U}(t) = \exp \left[-it \sum_{j,k} \hat{h}_{jk} \right] \approx \prod_{l=1}^n \prod_{j,k} e^{-i(t/n)\hat{h}_{jk}}. \quad (38)$$

Each factor of $e^{-i(t/n)\hat{h}_{jk}}$ corresponds to the circuit gate $\hat{N}(\alpha, \beta, \gamma)$ shown in Fig. 2(e), which contains three CNOT gates.

We consider three spins, each initialized to the state given in Eq. (2) with $\theta_1 = \pi/6$, $\phi_1 = \pi/3$, $\theta_2 = 3\pi/5$, $\phi_2 = 4\pi/3$, $\theta_3 = -\pi/5$, $\phi_3 = 2\pi/3$. These particular choices carry no particular significance and simply represent a typical state with no obvious structure. Time evolution is performed using the $N = 3$ case of Eq. (37). Students can use the programs in the [supplementary material](#) to explore how the results change with different initial states.

As with the $N = 2$ case, we take $J_x = 0.5J$, $J_y = -0.45J$, $J_z = 0.25J$ and work with dimensionless time Jt . A schematic circuit for computing $\langle \hat{S}_j^z \rangle$ is shown in Fig. 2(e) for

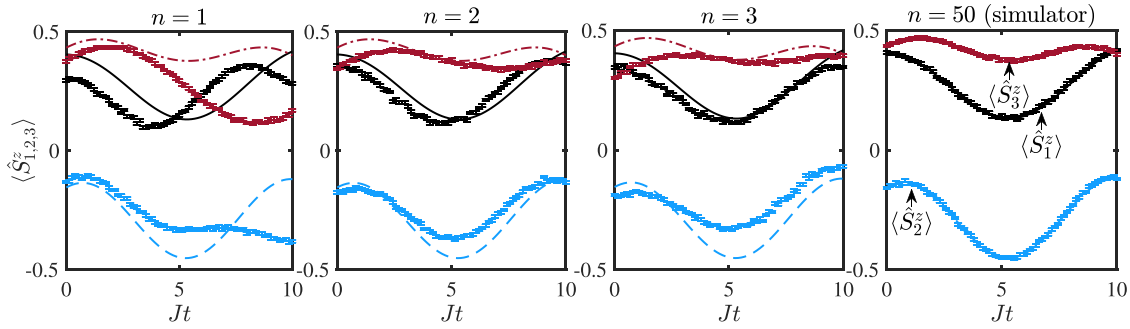


Fig. 8. Approximate time evolution of spin expectation values in three-spin system (Eq. (37)) with a variable number of Trotter steps using `ibm_perth`. The initial state is given in the main text. Results from quantum hardware (markers with error bars) are shown with theory (lines). Results in the right-most panel ($n = 50$) are obtained from the circuit simulator rather than actual quantum hardware.

$j = 1, 2, 3$. The results obtained from `ibm_perth` v1.1.14, a Falcon r5.11H processor, are depicted alongside theoretical predictions in Fig. 8 for $n = 1, 2, 3$ Trotter steps. To produce Fig. 8, the circuit in Fig. 2(e) is executed for 8192 shots for each of the 75 time samples in the range $0 \leq t \leq 10/J$. The challenge of simulating even a three-spin system is evident from the observation that $n = 2$ yields notably better agreement between experiment and theory than $n = 1$, but significant degradation occurs upon increasing the number of Trotter steps to $n = 3$. The noiseless simulator results are unaffected by CNOT errors, as shown by the excellent agreement between simulation and theory when the number of Trotter steps is increased significantly to $n = 50$. We note that the theoretical prediction is obtained from exact diagonalization without Trotter decomposition, so the right-most panel in Fig. 8 demonstrates that the Trotter decomposition itself indeed converges to the exact result for a sufficiently large number of Trotter steps. A MATLAB script to reproduce the theoretical predictions and a Jupyter notebook that executes the circuit in Fig. 2(e) are included in the [supplementary material](#). Due to the significant errors obtained for only three spins, investigation of larger systems is only possible by applying some form of error correction³⁵ or using a more sophisticated algorithm than the Lie–Trotter decomposition.

D. Many-body dynamics: Spin chains

The simple techniques of the preceding sections can now be applied to study complex, condensed-matter systems. We employ the quantum circuit simulator rather than actual hardware due to the significant errors associated with the significant number of Trotter steps required to study larger systems. In this section, we show results for the XXZ spin chain, defined by the Hamiltonian \hat{H}_N with $J_x = J_y \equiv J_{xy}$ and $N \gg 1$. Giamarchi²⁷ describes the XXZ model and its low-energy properties in the thermodynamic limit ($N \rightarrow \infty$) in detail. We will not repeat this full description here. Instead we give a brief summary and then discuss our results using the terminology of Giamarchi. We consider the antiferromagnetic regime in which $J_z < 0$, while $J_{xy} > 0$. In the thermodynamic limit, the XXZ spin chain exhibits a quantum phase transition^{36,37} at the point $|J_z| = J_{xy}$. For $0 < |J_z| < J_{xy}$, transport in the system is “ballistic,”²⁷ with excitations moving freely through the system without scattering. For $|J_z| > J_{xy}$, transport becomes “diffusive,” as excitations interact with each other via scattering processes.³⁸

We note that nonequilibrium *dynamics* in isolated, many-body systems is an active area of research.³⁹ A particular topic of recent interest is “quench dynamics” in which

many-body time evolution is computed for some arbitrary initial state that differs from the many-body ground state.⁴⁰ The circuits described in Secs. III A–III C effectively simulate quantum quenches for systems of size $N = 1, 2, 3$. Here we consider system size $N = 20$. To generate nonequilibrium dynamics, the initial state can be constructed as a “domain wall” spin configuration in which all spins in the left half of the system are in the state $|+\rangle$, while those in the right half of the system are in $|-\rangle$. Succinctly, $|\Psi_0\rangle = |++\cdots+-\cdots--\rangle$. As the system evolves, the domain wall will spread. In the ballistic phase, it is expected that the domain edges will spread with constant speed. For larger $|J_z| > J_{xy}$, the transport should become diffusive.

A circuit to prepare the domain wall state for $N = 6$ spins and perform approximate time evolution is shown in Fig. 2(f). Generalizing to larger systems such as $N = 20$ is accomplished by implementing this pattern on a larger number of qubits. It is striking how the same structure used for $N = 3$ spins (c.f., Fig. 2(e)) generalizes immediately to larger systems with no significant, structural modifications. The results are shown in the top row of Fig. 9 for $N = 20$ spins, 8192 shots, and $n = 100$ Trotter steps for each spin at each of the 20 time samples. For simplicity, we have used n Trotter steps for each time sample. In practice,²⁶ one generally maintains constant Trotter step size Δt and varies the number of Trotter steps according to simulation time t . For theoretical comparison, the bottom row of Fig. 9 depicts predictions obtained using the EXPOKIT package⁴¹ to compute the time-evolution operator from a sparse-matrix representation of Eq. (37) in MATLAB. A Jupyter notebook and a MATLAB script, which, respectively, reproduce the simulation and theoretical predictions, are included in the [supplementary material](#).

In general, excellent agreement is observed between simulation and theory. The transition from ballistic expansion of the domain wall to diffusive behavior in which the wall does not move appreciably at timescales considered is observed clearly. Somewhat crudely, one can think of the transition from ballistic expansion to diffusive spreading as being rooted in the strength of the J_z term, which favors antiferromagnetic alignment in the z direction for $J_z < 0$. For $|J_z| > J_{xy}$, destroying the antiferromagnetic alignment of the two central spins in the domain wall presents an energy penalty, which is not overcome by any order that develops in the xy -plane as the wall spreads.

We note the potentially under-appreciated value of using the simulator to perform such calculations, especially for students. The sparse-matrix-based approach using EXPOKIT is quite efficient, producing each panel of Fig. 9 with 100 time

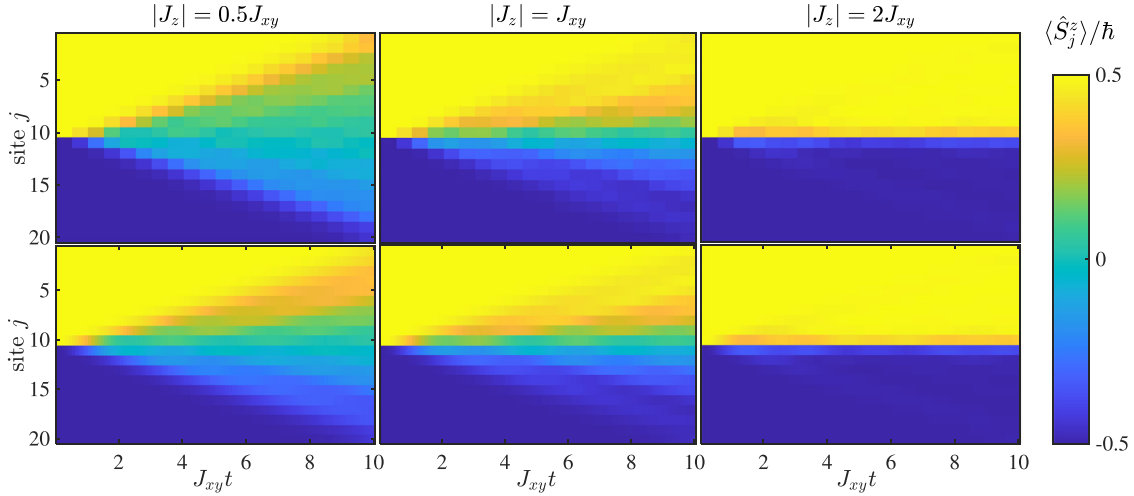


Fig. 9. (Color online) Magnetization dynamics $\langle \hat{S}_j^z \rangle$ are shown for time evolution of a domain wall initial state $|\Psi_0\rangle = |\cdots + \cdots - \cdots\rangle$ using the Hamiltonian \hat{H}_N (Eq. (37)) with $J_x = J_y \equiv J_{xy}$ and various values of $J_z < 0$. For $|J_z| < J_{xy}$, ballistic transport is observed in which the domain wall spreads at a constant speed. A quantum critical point exists at $|J_z| = J_{xy}$, and transport becomes diffusive for $|J_z| > J_{xy}$. The cases considered are $J_z = -0.5J_{xy}$ (left), $J_z = -J_{xy}$ (center), and $J_z = -2J_{xy}$ (right). Data collected from the quantum circuit simulator are shown on the top row, while corresponding theoretical predictions are shown in the bottom row. Quantum circuit simulator results were obtained with $n = 100$ Trotter steps.

samples in approximately 10 min. Each panel on the top row required roughly an hour of computation time on the same personal computer for 20 time samples. However, even with a simple-to-use software package such as EXPOKIT, the user must be able to write a program from scratch to represent the Hamiltonian in Eq. (37) and observables (e.g., \hat{S}_j^z) as sparse matrices. It is notable that the same thermodynamic-limit physics (ballistic vs diffusive transport) can be probed using a circuit scarcely more complex than that used for the system size $N = 3$. The application to $N \gg 3$ requires a many more Trotter steps to converge than the $N = 3$ case. However, this is not a fundamental change in the complexity of the circuit’s working principle, as additional Trotter steps can be implemented with a loop. Thus, the barrier to entry for students and non-experts to use the quantum circuit simulator to investigate many-body physics is arguably lower than that of many traditional computational approaches. Admittedly, the landscape of user-friendly, high-level libraries for probing these types of many-body systems is evolving rapidly. Some recently developed libraries for computing approximate time evolution in many-body systems are quite easy to implement and undeniably much more efficient than quantum circuit simulation. Notably, the iTensor⁴² library allows users to compute approximate dynamics using matrix-product state (MPS) tensor networks for systems with $N \sim \mathcal{O}(100)$. However, we argue that there is significant pedagogical value to developing a fairly robust many-body simulation from scratch.

IV. DISCUSSION

We have presented schemes for simulating small ($N \leq 3$) spin systems on IBM quantum computers and measuring the time-dependent expectation values of various spin components. Systems of size $N = 1, 2$ have been treated exactly, and the results contained only modest errors due to the simple circuits employed. For larger systems, approximate schemes must be employed to perform time evolution. A Lie–Trotter decomposition was used to approximate the time-evolution operator in a three-spin system. Even in a system with as few as three spins, the tradeoff between adding circuit elements to yield

convergence of the algorithm and reducing circuit elements to minimize noise becomes apparent. Finally, Qiskit’s quantum circuit simulator has been shown to be a useful and easy-to-use tool for simulating large ($N = 20$) interacting quantum systems.

The dynamics of small, spin systems are treated frequently in introductory quantum mechanics. The use of free, cloud-based IBM hardware provides a highly accessible opportunity for students to explore these systems experimentally. When comparing experimental results to theoretical predictions, students gain valuable experience in extracting experimental values of physical observables by manipulating the raw counts that are returned from the quantum circuit executions. Students must wrestle with what it means to “measure” a quantum mechanical observable, averaging a large number of independent results, and compute statistical uncertainties. Extensions of the problems presented here could form the basis of student projects in which the basic tools are applied to more complex systems (e.g., assessing effects of next-nearest neighbor interactions in larger systems using the quantum circuit simulator, or investigating time-dependent systems). Using the approach detailed in this work, students are well-equipped to use quantum computers and quantum circuit simulators to probe interesting physics.

In addition to providing a useful and accessible experimental component to the traditional undergraduate treatment of quantum mechanics, these types of simulations give students experience using cutting-edge technology. In this sense, the seemingly abstract content in a quantum mechanics course has direct relevance to an exciting area of industry. Companies such as IBM,⁷ Rigetti,⁵ and IonQ⁶ have devoted considerable resources to constructing functional NISQ devices. Such devices represent a small-but-important step toward the construction of fault-tolerant quantum computers. The current generation of hardware is far from capable of breaking RSA encryption, simulating complex molecules for drug design, or performing any of the other tasks that likely motivate large corporations to invest heavily in the technology. However, this currently available hardware is capable of simulating—albeit *imperfectly*—the type of conceptually

simple quantum physics one finds in undergraduate-level textbooks on quantum mechanics. Thus, students with a working knowledge of basic quantum mechanics are well equipped to explore such technology. It is not yet clear if sufficiently clever error correction and programming will enable NISQ devices to show a legitimate quantum advantage over classical computers or if they simply represent a pivotal step in the evolution of practical quantum computation. Excellent resources^{20,43–45} are available for introducing quantum computing to students with minimal assumptions regarding background. We hope the present work demonstrates how a familiarity with undergraduate-level quantum mechanics can be leveraged to acquire expertise in using this exciting, emerging technology.

SUPPLEMENTARY MATERIAL

Please click on [this link](#) to access the supplementary material for Jupyter notebooks for performing the calculations on quantum computers and quantum circuit simulators described in the main text. The MATLAB scripts to perform theoretical calculations described in the main text are also included. Print readers can see the supplementary material at <https://doi.org/10.60893/figshare.ajp.c.7438423>.

ACKNOWLEDGMENTS

The authors acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team. The authors acknowledge the access to advanced services provided by the IBM Quantum Researchers Program.⁴⁶ Additionally, the authors are grateful for insightful comments and valuable suggestions from the anonymous reviewers.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

^aElectronic mail: jarrett.lancaster@mma.edu, ORCID: 0000-0002-7984-3240.

^bPresent address: Department of Arts & Sciences, Maine Maritime Academy, One Pleasant Street, Castine, Maine 04420, ORCID: 0000-0001-6875-1861.

¹R. P. Feynman, “Simulating physics with computers,” *Int. J. Theor. Phys.* **21**(6-7), 467–488 (1982).

²P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM J. Comput.* **26**(5), 1484–1509 (1997).

³R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM* **21**(2), 120–126 (1978).

⁴J. Preskill, “Quantum computing in the NISQ era and beyond,” *Quantum* **2**, 79 (2018).

⁵See <https://www.rigetti.com> for Rigetti.

⁶See <https://ionq.com> for IonQ.

⁷See <https://quantum-computing.ibm.com> for IBM Quantum.

⁸Y. Alexeev *et al.*, “Quantum computer systems for scientific discovery,” *PRX Quantum* **2**(1), 017001 (2021).

⁹J. Brody and G. Guzman, “Calculating spin correlations with a quantum computer,” *Am. J. Phys.* **89**(1), 35–40 (2021).

¹⁰D. Candela, “Undergraduate computational physics projects on quantum computing,” *Am. J. Phys.* **83**(8), 688–702 (2015).

¹¹B. Kain, “Searching a quantum database with Grover’s search algorithm,” *Am. J. Phys.* **89**(6), 618–626 (2021).

¹²S. Johnstun and J.-F. Van Huele, “Understanding and compensating for noise on IBM quantum computers,” *Am. J. Phys.* **89**(10), 935–942 (2021).

¹³See <https://qiskit.org> for Qiskit.

¹⁴É. M. Alves, F. D. S. Gomes, H. S. Santana, and A. C. Santos, “Simulating single-spin dynamics on an IBM five-qubit chip,” *Rev. Bras. Ensino Fis.* **42**, e20190299 (2020).

¹⁵D. H. McIntyre, C. A. Manogue, and J. Tate, *Quantum Mechanics* (Pearson, New York, 2012), pp. 22–26, 97–105, 210.

¹⁶A. Abbas, S. Andersson, A. Asfaw, A. Corcoles, L. Bello, Y. Ben-Haim *et al.*, see <https://qiskit.org/textbook/ch-demos/piday-code.html> for *Learn Quantum Computation Using Qiskit* (2020).

¹⁷M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge U.P., Cambridge, 2010), pp. 15–16, 174–185.

¹⁸J. J. Sakurai and Jim Napolitano, *Modern Quantum Mechanics*, 2nd ed. (Addison-Wesley, San Francisco, 2011), pp. 20–23, 168–172.

¹⁹D. J. Griffiths and D. V. Schroeter, *Introduction to Quantum Mechanics*, 3rd ed. (Cambridge U.P., Cambridge, 2018), pp. 165–180.

²⁰See <https://learning.quantum.ibm.com>.

²¹See https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/runtime_service for Qiskit Runtime.

²²See <https://learning.quantum.ibm.com/tutorial/explore-gates-and-circuits-with-the-quantum-composer> for IBM Quantum Composer.

²³J. M. Gambetta, J. M. Chow, and M. Steffen, “Building logical qubits in a superconducting quantum computing system,” *npj Quantum Inf.* **3**(1), 2 (2017).

²⁴C. Singh, “Student understanding of quantum mechanics,” *Am. J. Phys.* **69**(8), 885–895 (2001).

²⁵C. Singh, “Student understanding of quantum mechanics at the beginning of graduate instruction,” *Am. J. Phys.* **76**(3), 277–287 (2008).

²⁶A. Smith, M. S. Kim, F. Pollmann, and J. Knolle, “Simulating quantum many-body dynamics on a current digital quantum computer,” *npj Quantum Inf.* **5**(1), 106 (2019).

²⁷T. Giamarchi, *Quantum Physics in One Dimension* (Oxford U.P., Oxford, 2003), pp. 137–153, 160–176.

²⁸F. Vatan and C. Williams, “Optimal quantum circuits for general two-qubit gates,” *Phys. Rev. A* **69**(3), 032315 (2004).

²⁹MATLAB, *MATLAB Version 9.10.0.1669831 (R2021a Update 2)* (The MathWorks Inc., Natick, MA, 2021).

³⁰S. I. Doronin, E. B. Fel’dman, E. I. Kuznetsova, and A. I. Zenchuk, “Simulation of multiple-quantum NMR dynamics of spin dimer on quantum computer,” *Appl. Magn. Reson.* **53**(7-9), 1121–1131 (2022).

³¹A. Cervera-Lierta, “Exact Ising model simulation on a quantum computer,” *Quantum* **2**, 114 (2018).

³²H. F. Trotter, “On the product of semi-groups of operators,” *Proc. Am. Math. Soc.* **10**(4), 545–551 (1959).

³³E. Kökcü, D. Camps, L. Bassman, J. K. Freericks, W. A. de Jong, R. Van Beeumen, and A. F. Kemper, *Phys. Rev. A* **105**, 032420 (2022).

³⁴See <https://research.ibm.com/blog/2022-quantum-open-science-prize>.

³⁵P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, “Scalable mitigation of measurement errors on quantum computers,” *PRX Quantum* **2**(4), 040326 (2021).

³⁶R. Shankar, *Quantum Field Theory and Condensed Matter: An Introduction* (Cambridge U.P., Cambridge, 2017), pp. 334–363.

³⁷S. Sachdev, *Quantum Phase Transitions*, 2nd ed. (Cambridge U.P., Cambridge, 2011), pp. 412–435.

³⁸J. Sirker, “Transport in one-dimensional integrable quantum systems,” in *SciPost Physics Lecture Notes*, Vol. 17 (SciPost Foundation, 2020).

³⁹Anatoli Polkovnikov, Krishnendu Sengupta, Alessandro Silva, and Mukund Vengalattore, “Colloquium: Nonequilibrium dynamics of closed interacting quantum systems,” *Rev. Mod. Phys.* **83**(3), 863–883 (2011).

⁴⁰F. H. L. Essler and M. Fagotti, “Quench dynamics and relaxation in isolated, integrable quantum spin chains,” *J. Stat. Mech.* **2016**(6), 064002 (2016).

⁴¹R. B. Sidje, “EXPokit: A software package for computing matrix exponentials,” *ACM Trans. Math. Software* **24**(1), 130–156 (1998).

⁴²M. Fishman, S. R. White, and E. M. Stoudenmire, “The ITensor software library for tensor network calculations,” in *SciPost Physics Codebases*, Vol. 4 (SciPost Foundation, 2022).

⁴³C. Hughes, J. Isaacson, A. Perry, R. F. Sun, and J. Turner, *Quantum Computing for the Quantum Curious* (Springer, Berlin, 2021).

⁴⁴K. J. Ferris, A. J. Rasmusson, N. T. Bronn, and O. Lanes, “Quantum simulation on noisy superconducting quantum computers,” *arXiv:2209.02795* (2022).

⁴⁵T. Wong, *Introduction to Classical and Quantum Computing* (Rooted Grove, Omaha, NE, 2022).

⁴⁶See <https://quantum-computing.ibm.com/programs/researchers> for IBM Quantum Researchers Program.