Revista Brasileira de Ensino de Física, vol. 46, e20240129 (2024)
www.scielo.br/rbef
DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0129

Articles

The two-site Heisenberg model studied using a quantum computer: A didactic introduction

O modelo de Heisenberg de dois sítios estudado usando um computador quântico: Uma introdução didática

M.G.J. Oliveira*[1,2], T.V.C. Antão[2,3], N.M.R. Peres[1,4,5,6]

[1]Universidade do Minho, Departamento de Física e Centro de Física, 3710-057, Braga, Portugal.
[2]Universidade do Minho, Escola de Ciências, Laboratório de Instrumentação e Física Experimental de Partículas, 4701-057, Braga, Portugal.
[3]Aalto University, Department of Applied Physics, 02150, Espoo, Finland.
[4]International Iberian Nanotechnology Laboratory, 4715-330, Braga, Portugal.
[5]University of Southern Denmark, POLIMA–Center for Polariton-driven Light–Matter Interactions, DK-5230, Odense M, Denmark.
[6]University of Southern Denmark, Danish Institute for Advanced Study, DK-5230, Odense M, Denmark.

Received on April 16, 2024. Revised on August 17, 2024. Accepted on August 21, 2024.

The two-site Heisenberg model has an extraordinarily simple analytical solution and is traditionally used as a benchmark against numerical methods, such as exact diagonalization and Monte Carlo methods. In the same spirit, we benchmark three quantum algorithms that are implemented in a quantum computer against the analytical solution of this model. In particular, this presentation includes a description of the standard and iterative quantum phase estimation algorithms, as well as the variational quantum eigensolver. These quantum algorithms are introduced in a pedagogical fashion allowing newcomers to the subject, familiar with only the most basic quantum mechanical systems, to easily reproduce the presented results and apply the methods to other problems, thus building a seemingly under-appreciated path towards useful quantum algorithms through the lens of simulating and computing properties of physical quantum systems.
Keywords: Heisenberg Model, Quantum algorithms, Phase estimation, Variational quantum eigensolver.

O modelo de Heisenberg de dois sítios tem uma solução analítica extraordinariamente simples e é tradicionalmente usado como um ponto de referência para métodos numéricos, como a diagonalização exata e métodos de Monte Carlo. Nessa linha de pensamento, comparamos três algoritmos quânticos implementados num computador quântico com a solução analítica deste modelo. Em particular, este trabalho inclui uma descrição dos algoritmos quânticos tradicional e iterativo para a estimação de fase, bem como o *variational quantum eigensolver*. Estes algoritmos quânticos são introduzidos de forma pedagógica, permitindo que leitores familiarizados apenas com os sistemas quânticos mais básicos, reproduzam facilmente os resultados apresentados e apliquem os métodos a outros problemas, construindo assim um caminho, aparentemente subestimado, para algoritmos quânticos úteis através da simulação e cálculo de propriedades de sistemas quânticos.
Palavras-chave: Modelo de Heisenberg, Algoritmos quânticos, Estimativa de fase, Variational quantum eigensolver.

1. Introduction

The second quantum revolution refers to the exploration of quantum superposition and quantum entanglement with sights set on addressing challenging physical and engineering problems rooted in quantum mechanical systems. This revolution has become possible due, for example, to the appearance of intense sources of entangled photons [15–18], which led to extraordinary advances in quantum cryptography [19] and quantum communication [20, 21]. More recently, we have seen spectacular advances in quantum computation [33], with the advent of new age quantum computers.

Quantum computers allow one to treat, in principle, a select but diverse class of problems, including some that are prohibitive to classical computers. Most of those familiar with quantum computation have certainly heard of the possibility of using quantum computers to factor large primes using the famous Shor algorithm [22], built on the idea of the Quantum Fourier transform (QFT) [1]; to search through databases more efficiently using Grover's algorithm [23], or even to help solve the mathematical problem of distinguishing knots [41].

In the spirit of Feynman's original insight into the necessity of utilising quantum computers to simulate

[*] Correspondence email address: mgabijo@gmail.com

intrinsically quantum systems [13], another avenue which showcases the power of quantum computing is the diagonalization of quantum models, which finds several applications in quantum condensed matter physics and quantum chemistry. In this realm, the role of quantum computation is yet to be fully elucidated, as new quantum algorithms are required in order to treat many useful systems. Due to the promise of efficiency, as well as the relative sparseness of already established algorithms, this is an extremely active field of research. Concerning the simulation of physical systems, one may find the direct application of the time evolution operator on a quantum computer [24, 25], or some more sophisticated techniques relying on the Quantum Phase Estimation algorithm (QPEA) [26] or variational quantum algorithms [27].

At the present moment, we are living in the Noisy Intermediate-Scale (NISQ) era [28], where the number of qubits in existing devices is limited, and the devices themselves are severely prone to noise. Therefore, some of the most well studied algorithms in the scope of quantum information theory do not display the expected behaviour when running on real quantum machines. The QPEA, which aims to find the eigenvalues of a unitary operator, requires long and deep circuits, and so, is not functional on current devices. Attending to the requirements of most NISQ era devices, algorithms with smaller quantum circuits, such as the iterative version of QPEA – Iterative Phase Estimation algorithm (IPEA) [29] –, and other hybrid quantum-classical approaches have been widely explored.

In the hybrid quantum-classical paradigm, variational quantum algorithms are a prominent player. In this kind of algorithm, which is, in fact, a particular type of Quantum Machine Learning (QML), some parameterised quantum circuit is evaluated by a quantum computer and its parameters are updated using a classical optimiser. This class of algorithms includes the Variational Quantum Eigensolver (VQE) [3] which aims to find the ground state of a given system. This makes such algorithms very useful in the context of quantum chemistry and simulation of physical systems. Other hybrid quantum-classical algorithms include the Quantum Approximate Optimization Algorithm (QAOA) [30] used to solve optimisation problems [45], and QML [34, 42].

Even though some of the mentioned above quantum algorithms don't behave as desired in NISQ devices, the algorithms here studied, i.e., the QPEA, IPEA, and VQE, seem to find an advantage over their classical counterparts [47–49]. In the specific case of the Heisenberg model, these algorithms have good scalability in the system size, allowing the study of large models unsolvable using classical methods, which is itself a statement of the quantum advantage of these algorithms. Nonetheless, the quantum computations are still limited by the state of the current devices [50, 51], which is perhaps the main obstacle in the study and implementation of large systems.

It is only with a healthy mix of hardware development and algorithm optimisation that one can hope to solve physically relevant problems using quantum computers. In this work, throughout all the following sections, the focus of the discussion will be on some quantum algorithms and their implementation in the context of faulty or noisy hardware. Nevertheless, developments in the hardware of quantum computers have also been on the rise, resulting in the emergence of new kinds of platforms, both larger (with more qubits) and more robust to noise. The major difference between the diverse quantum computing platforms is the type of utilised qubits. The most known and common types of qubits include superconducting junctions [35], neutral atoms [36], photonic platforms [37], trapped ions [38], and quantum dots [39].

This paper is organised following a pedagogical presentation[1] in order overcome students' difficulties with many of the seemingly counter-intuitive aspects of quantum computation [32]. In Section 2, the analytical solution of the two-site Heisenberg model is reviewed. This presentation serves as an introduction to the main system to which we will apply the different algorithms to be explored in later sections. We fix the notation for states and energies and provide the eigenvectors and eigenvalues that later will be compared with the output of the three quantum algorithms developed. In Section 3 we present three algorithms which enable a calculation of the eigenvalues and eigenstates of the two-site Heisenberg model. These are discussed with emphasis on a providing a detailed overview of the algorithms and of their implementation. In Section 4 we discuss our results in broader perspective. Finally, all of the code utilised in this work is available in https://github.com/mgabijo/The-two-site-Heisenberg-model-on-a-quantum-computer.

2. The Two-site Heisenberg Model and its Analytical Solution

It is often the case that in most physics or engineering physics curricula, quantum mechanics is taught from the perspective of "pure" physics, while quantum computation is dealt with using the perspective of computer scientists or information theorists. Indeed, the usage of quantum computers to simulate physical systems is not often emphasised at such early stages of learning. However, it is our belief that such a perspective can successfully be presented quite early in the learning process of quantum mechanics, both for physics and engineering students. Moreover we believe that this would come with benefits both for a thorough understanding of quantum

[1] Attempts to introduce quantum computation through didactic teaching materials have already been proposed [43–46].

DOI: https://doi.org/10.1590/1806-9126-RBEF-2024-0129

algorithms, as well as quantum mechanical and solid state systems.

The choice of a simple and familiar model is often crucial when testing, implementing or developing new algorithms, independently of their quantum or classical nature. For this reason, and with the mixed intent of benchmarking simple results, as well as reviewing an important model for quantum magnetism, we focus on the two-site Heisenberg model.

The Heisenberg model is as old as quantum mechanics itself [10] and has served physicists well as the basis for the study of the quantum nature of magnetic order in many different physical applications.

In this model, quantum objects called atomic spins are paramount. In the simplest magnets, for instance, the atoms forming a crystalline lattice hold a certain amount of electrons. Each of the spin states of these electrons is characterised by an electronic spin quantum number $S = 1/2$, and spin projection along a direction z, labelled by $m = \pm 1/2$.

In magnetic materials, it can be the case that adding up the contribution from each of these electronic spins results in a non-zero value. This occurs, for instance, if electrons in inner shells are all paired up with ones of opposite spin projections, and a single electron with spin $S = 1/2$ is left in a valence shell. Atomic spins of larger value S can also form in case the previously described spin-pairing is not energetically favoured. This is a consequence of the famous Hund's rules of chemistry (see [11] for a lively introduction to the theory of magnetism).

The resulting picture is that atoms carry an overall spin. Due to the statistical properties of electrons shared in chemical bonds between atoms, namely their fermionic statistics, an effective interaction between spins can be derived. This is called the exchange interaction. Ferromagnets or antiferromagnets then appear if it becomes energetically favourable for spins to be respectively aligned or anti-aligned in a macroscopic ordered fashion. The collective alignement of spins results in the magnetization of the material. Although this intuitive picture is fairly clear, Heisenberg models are essentially untractable in any exact fashion in two or three dimensional crystals, with analytical solutions being extremely scarce.

The 1D Heisenberg models, however, does have an exact analytical solution, relying on the so-called Bethe-Ansatz method [12].

Despite this possibility, the Bethe-Ansatz method is not approachable or simple in any way, as exact solutions of models using this method are known to be infamously lengthy and tricky. For this reason, we can reduce the dimensionality of the problem even further, by looking at a "0D" Heisenberg model. The simplest non-trivial such model is the two-site Heisenberg model (see Figure 1).

The solution of the two-site Heisenberg model does not require this sophisticated approach as its exact solution is given by the diagonalization of a 4×4 matrix. Nevertheless, it captures the intuition regarding ferromagnetism or antiferromagnetism, where it becomes clear that aligned or anti-aligned spins are favoured depending on the sign of the exchange interaction.

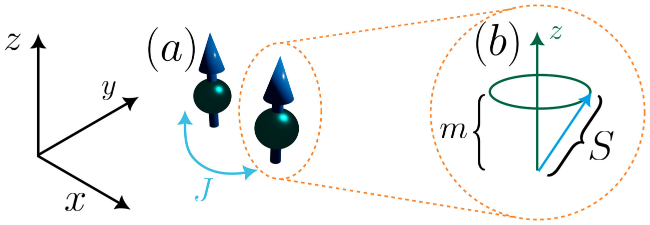

Figure 1: (a) Artistic rendition of the two-site Heisenberg model. The spins are oriented along the z axis, representing a state $|11\rangle$. (b) Semiclassical spin-vector representation of the spin. The eigenvalue S of the total spin is represented by the length of the spin-vector, whereas its projection along z is represented by m, the eigenvalue of $\hat{S}_z$.

Below, we review this simple calculation, providing the results which will be later compared with those of a quantum computer calculation, as well as providing a bridge between the notation of spin $S = 1/2$ states, and the qubit notation, familiar from quantum computation.

As stated before, the quantum mechanical states we are concerned with are the spin states of two "atomic" spins. These states are eigenstates of two distinct operators $\hat{S}$ and $\hat{S}_z$, and are, in fact completely characterized by the eigenvalues of both of these operators, call them S and m respectively.

The operator $\hat{S}$ is called the total spin operator, and its eigenvalues are related to the magnitude of the atomic spin S. For instance, if a single electron with unpaired spin is present in an atom, then the atomic total spin is defined by the eigenvalue as $S = 1/2$, whereas if more electron spins are unpaired, one can have $S = 1, 3/2, 2, \ldots$.

On the other hand, the operator $\hat{S}_z$ is called the spin projection operator along z, and it measures how much the spin "vector"[2] is tilted along this axis (see Figure 1). One can project along other axes, such as x or y in any Cartesian system, and in fact, one has the relation $\hat{S}^2 = \hat{S}_x^2 + \hat{S}_y^2 + \hat{S}_z^2$. The eigenvalues of this squared operator take the form $S(S+1)$.

This relation between $\hat{S}$ and $\hat{S}_z$ imposes certain restrictions on the the eigenvalue m of $\hat{S}_z$. Namely, one can show that $-m \leq S \leq m$. For the simplest example of a single $S = 1/2$ spin, one therefore has a two-level system with state vectors $|S, m\rangle = |1/2, +1/2\rangle, |1/2, -1/2\rangle$.

Since S is really a fixed parameter of most models, one often omits it entirely from the labelling of states, when it is understood its value is. Furthermore, one

[2] The visualization of spins as angular momentum vectors precessing around an axis is a useful intuition although a semiclassical one.

often identifies the positive and negative eigenvalues for $S = 1/2$ with "up" and "down" pointing spins, and simply writes the eigenstates as $|\uparrow\rangle$ and $|\downarrow\rangle$. In this paper, we will use instead a notation stemming from quantum computation. Since $|\uparrow\rangle$ and $|\downarrow\rangle$ form a two-state system, or in other words, a qubit, the basic element of quantum computation. As such, one might as well identify them with $|\uparrow\rangle \equiv |0\rangle$ and $|\downarrow\rangle \equiv |1\rangle$.

With this notation in mind, we can introduce the two-site Heisenberg model *per se*. Let us organise the spin projection operators into a vector $\hat{\mathbf{S}} = (\hat{S}_x, \hat{S}_y, \hat{S}_z)$, and let the spin on each site be additionally labelled by $i = 1, 2$. The Hamiltonian reads

$$H = J\hat{\mathbf{S}}_1 \cdot \hat{\mathbf{S}}_2. \tag{1}$$

The quantity J is the aforementioned exchange integral or interaction. In order to handle this Hamiltonian, it becomes useful to rewrite it in a language where it becomes clear what the action on the eigenstates is. First, it should be noted that for spin $S = 1/2$, one has a direct mapping of the spin operators to the Pauli operators X, Y, Z. Let us, in all discussion that follows, set $\hbar = 1$. This is always possible under a suitable redefinition of the units of the exchange integral and time variables. One has, in this convenient unit system, $\hat{S}_x = X/2, \hat{S}_y = Y/2$ and $\hat{S}_z = Z/2$. Thus, making explicit the tensor product structure of the Hamiltonian, one finds what is sometimes called a sum of Pauli strings

$$H = \frac{J}{4}(X_1 \otimes X_2 + Y_1 \otimes Y_2 + Z_1 \otimes Z_2). \tag{2}$$

As we will discuss later, this sort of decomposition is an essential step in the unfolding of some of the quantum algorithms we will present in this paper. For now, note that in the basis $\{|0\rangle, |1\rangle\}$, a matrix representation of such a Hamiltonian is extremely simple, and computing the necessary tensor products, one finds

$$H = \frac{1}{4}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

Diagonalization of this problem is straightforward, as the Hamiltonian is already conveniently block-diagonalized. One can check that the different eigenstates of this Hamiltonian are

$$|\psi_s\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle), \tag{4}$$

$$|\psi_t^{(1)}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{5}$$

$$|\psi_t^{(2)}\rangle = |00\rangle, \tag{6}$$

$$|\psi_t^{(3)}\rangle = |11\rangle. \tag{7}$$

Furthermore, when checking this, one also realises that all the states $|\psi_t^{(i)}\rangle$, which are symmetric under permutation of the spins, are degenerate in energy. The eigenvalue of the Hamiltonian for such states is $E = J/4$.

On the other hand, the only anti-symmetric eigenstate $|\psi_s\rangle$ has a different energy $E = -3J/4$. Thus, eigenstates are grouped into a spin-singlet and a spin-triplet, each of which has different energy levels.

As expected, the exact diagonalization by hand is simple in all regards, and with matters of notation fixed, what is now left unanswered is the how one can use quantum computational algorithms to reproduce such results. This is the aim of the remainder of the present paper.

3. Quantum Algorithms

The implementation of a quantum algorithm in a quantum computer is not without subtleties, and the role of this paper is as much to introduce some quantum algorithms as it is to address these subtleties. Before describing the quantum algorithms themselves, we shall start by giving an insight into the structure of qubits as the basic units of quantum information.

In a quantum computer all the information is encoded in the states of two-level systems called qubits. In principle a generic qubit state can be written as a superposition of two orthogonal basis states, denoted as $|0\rangle$ and $|1\rangle$. In this basis, the most general qubit state can be represented resorting to two "angle" variables as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{8}$$

where $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi)$.

This general state can be represented as a point in a sphere, often called the Bloch sphere [14]. For example, a state in the equator of the Bloch sphere has $\theta = \pi/2$ and arbitrary φ. Usually, the qubit register of a quantum computer is initialised as a tensor product state where all qubits lie in the $|0\rangle$ state. In all that follows, we shall employ a notation for tensor product states where $|0\rangle^{\otimes N} \equiv \overbrace{|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle}^{N \text{ times}}$. Starting from this initial state, any computation then proceeds via the application of different types of unitary operators. Each of these unitary transformations is commonly referred to as a quantum gate, and a quantum circuit can be defined as a collection of quantum gates interconnected by quantum wires, which represent the flow of the states of each qubit. When a computation is completed, a measurement of the final state of the qubits is performed.

Over the next subsections we introduce three different quantum computational methods for determining the states and energies of a given quantum system. The results of these methods will be compared among themselves in the setting of the two-site Heisenberg model, as well as benchmarked against the exact solution given in Sec. 2.

3.1. Quantum phase estimation

The QPEA has a very suggestive name, directly related to its purpose. Indeed, the objective of the QPEA is estimating the phase θ conferred by a unitary operator U to one of its eigenstates $|\psi\rangle$. Note that, due to the unitarity of U, its eigenenergies can always be written as a complex number in the unit circle, i.e. $e^{2\pi\phi i}$, for $\phi \in [0, 1)$.

Imagine now that $U(t)$ is a unitary operator which describes, at time t, the time-evolution of a quantum system described by a certain Hamiltonian H. In this context, estimating the phase conferred by the time evolution to an eigenstate of H amounts to computing the energy of this state. This, of course, is due to the fact that the rate at which the phase of a stationary state oscillates is proportional to its energy.

The QPEA relies on a well-known sub-routine – the Quantum Fourier Transform (QFT) [1]. The circuit used in the implementation of this algorithm is composed of two different registers of qubits, i.e. two different groups of qubits with different purposes: The counting register, and the state register. The circuit can furthermore be broken down into four different parts, and a schematic representation of these four parts as well as the circuit implementing the QPEA are presented in Figure 2. These parts are the initialisation step, the phase kickback step, the phase estimation step, and the post-processing step.

The first n qubits showcased in Figure 2, belong to the counting register and the size (n) of this register directly impacts the accuracy[3] of the value of the estimated phase.

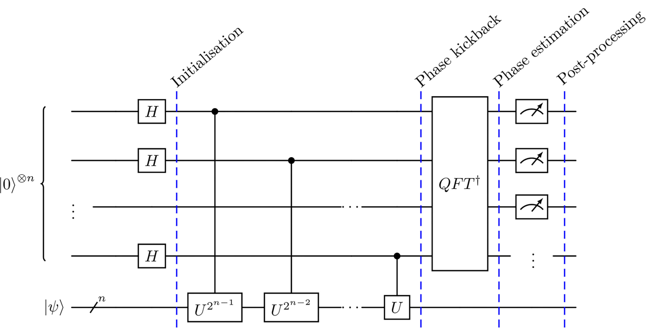

Figure 2: Generic circuit for quantum phase estimation algorithm. The circuit is composed of four major parts: initialisation, phase kickback, phase estimation and post-processing. Where H is the usual Hadamard gate, U is the operator that we want to estimate the phase, the successive application of the controlled gate U to a power of two kickbacks the digits of the phase to the first n qubits (counting qubits) and the $QFT^\dagger$ is the application of the inverse Quantum Fourier transform (Equation 11).

The purpose of the counting register is to host the bits describing a binary decomposition of the phase. Its size thus corresponds to the number of desired bits in the binary decomposition of the phase value.

The remaining qubits form the state register. The size of this register must match the dimensionality of the Hilbert space describing the system we aim to study, e.g. for the two-site Heisenberg model a register of two qubits is necessary.

The circuit which implements the QPEA can now be explained in a detailed step-by-step manner. We detail the aforementioned four steps:

1. **Initialisation** – The state register must be initialised in an eigenstate[4] $|\psi\rangle$ of the operator U. Within the counting register, the core idea is to bring the state of all qubits to the equator of the Bloch sphere in order to encode the information in a phase. Whilst by definition a QFT should be applied to all of the counting qubits, since the initial state of the qubits is $|0\rangle$, the application of the QFT simplifies to the application of an n-bit Hadamard gate to the counting register (see Appendix A), creating a state superposition[5].

2. **Phase kickback** – The goal of this step is to encode each bit of the phase that we want to find in a phase of a counting qubit state. The name phase kickback comes from the fact that the bits of the phase are kicked back to the counting qubits in the form of phases. To implement this step it is necessary to implement a series of gates realising the operator U. This is the operator whose phase-shift imparted on $|\psi\rangle$ we wish to estimate. In the particular case of estimating the energy levels of the two-site Heisenberg model such an operator is simply the time evolution of the Hamiltonian, i.e. $U(t) = e^{-iHt}$.

   If n counting qubits are initially prepared, the k^{th} qubit of the counting register is used as the control for a CU^{2k} gate[6] acting on the state register, with k ranging from 0 to $n - 1$. Since U is an unitary

[3] Note that when talking about simulators or fault-tolerant devices the accuracy of the value found only depends on the number of bits used to its representation. However, NISQ devices are prone to noise, and so, an increase in the number of qubits leads to an increase in the noise, i.e. in practice, there is a trade-off between the number of qubits used and the amount of noise introduced in the calculation.

[4] Although the traditional definition of the algorithm initialises the state register in an eigenstate, some more advanced studies [2] try to find the eigenvalues (phase) through the initialisation in state superposition instead.

[5] For single qubit state the Hadamard gate reads $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Taking, as an example, the qubit state $|\psi\rangle = (1, 0)$, we have $H|\psi\rangle = \frac{1}{\sqrt{2}}(1, 1) = \frac{1}{\sqrt{2}}(1, 0) + \frac{1}{\sqrt{2}}(0, 1)$.

[6] A CG or controlled-G gate is a multi-qubit gate that is only applied in the target qubits if the control qubits are in the state $|1\rangle$. Consider the X gate, defined as $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, if the $CX_{0,1}$ (control-X) is applied on the state $|\psi\rangle = |00\rangle + |10\rangle$, the resulting state is $CX_{0,1}\,|\psi\rangle = |00\rangle + |11\rangle$. Here, an index notation is utilised, where the first index refers to the set of control qubits, whereas the second refers to the set of target qubits. In an arbitrary controlled gate CG_{q_0,q_1}, q_0 is the control qubit and q_1 is the target. Note also that the CX gate is commonly referred to as the CNOT gate.

operator and $|\psi\rangle$ is an eigenstate of U, $U|\psi\rangle = e^{2\pi\phi i}|\psi\rangle$, and so, $U^{2^k}|\psi\rangle = e^{2\pi\phi i 2^k}|\psi\rangle$. Now, recall that the controlled gate U (CU) is only applied in the subspace where the k^{th} counting qubit is in the $|1\rangle$ state. As such, for the k^{th} counting qubit and state register subspace, the effect of CU is

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|\psi\rangle \xrightarrow{CU_{0,1}^{2^k}} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi\phi i 2^k}|1\rangle\right)|\psi\rangle. \tag{9}$$

Bear in mind that since $|\psi\rangle$ is an eigenstate of U, the state register remains unchanged after the application in Equation 10. After applying all the n controlled operations U^{2^k} (with k ranging from 0 to $n-1$), and using the relation in Equation 10, the state of the counting register is

$$\begin{aligned}|\text{count}\rangle_n &= \frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{2\pi\phi i 2^{n-1}}|1\rangle\right)\\ &\quad\otimes\cdots\otimes\left(|0\rangle + e^{2\pi\phi i 2^0}|1\rangle\right)\\ &= \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{2\pi\phi i 2k}|k\rangle. \end{aligned}\tag{10}$$

Note that in the state $|k\rangle$, k should be written in its binary representation, e.g. if $k=2$, the state $|k\rangle$ is $|2\rangle \equiv |10\rangle \equiv |1\rangle \otimes |0\rangle$.

3. **Phase estimation** – After the controlled operations kick back the fractional digits of the phase to the counting register, the inverse Quantum Fourier Transform [1] ($QFT^\dagger$) allows us to obtain them through measurements in the computational basis. The application of the gate labelled by $QFT^\dagger$ results in the state

$$\frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{2\pi\phi i 2k}|k\rangle|\psi\rangle \xrightarrow{QFT^\dagger} |\tilde{\phi}\rangle|\psi\rangle, \tag{11}$$

where $\tilde{\phi}$ is an approximation of ϕ. Bear in mind that our goal with this algorithm is to determine ϕ, but due to the impossibility of using an infinite amount of counting qubits, we most likely will, at most, find a good approximation of ϕ, denoted by $\tilde{\phi}$. Although there is a Qiskit class that directly implements the QFT and its inverse as a quantum circuit, i.e., the class `QFT`, the explicit circuit representation of the QFT is in Appendix B.

4. **Post-processing** – The result of the measurement comes in the form of a probability distribution, i.e. the result consists of a dictionary that maps the possible phase values (between 0 and 2^{n-1}) with the associated probability. Two different post-processing methodologies allow for a mapping of the probability distribution into the desired phase. The first and simplest method only makes use of the most probable phase, picking that alone as the correct result. The second makes use of the concept of circular statistics over every value in the probability distribution (and is for this reason sometimes called circular optimisation): Given a probability distribution $P(x)$ defined for 2^n integers x, one computes the circular mean μ, defined as

$$\mu = \sum_{x=0}^{2^n-1} e^{2\pi i \frac{x}{2^n}} P(x). \tag{12}$$

This results in the weighted average over the possible phases. Such an averaging procedure can often be beneficial since we are dealing with phases. Indeed, such a procedure assures that the modular arithmetic nature of the phases taken into account, with the value of 0 being necessarily identified with the value 2π. In the end, μ is the complex mean value and its argument is the mean estimated phase, which can be used as a good approximation of ϕ. One has

$$\phi = \frac{\arg(\mu)}{2\pi}. \tag{13}$$

3.2. Iterative quantum phase estimation

As stated in Subsection 3.1, the QPEA, has its precision limited by the number of counting qubits. On the other hand, in NISQ devices, the noise increases with the size of the circuit, both in number of qubits and depth of the circuit[7].

Hence, when implementing this algorithm in a real quantum computer, one must keep in mind the fact that adding counting qubits induces more noise in the results, instead of simply increasing the performance. In response to this issue, the iterative quantum phase estimation algorithm, also simply known as iterative phase estimation algorithm, implements the QPEA with a single counting qubit with the accuracy of the algorithm being restricted by the number of iterations rather than the number of counting qubits.

To comprehend this iterative version, recall that the aim of using the quantum phase estimation algorithm is to estimate the phase ϕ conferred by a unitary operator U to one of its eigenstates $|\psi\rangle$. Recall also that the eigenvalues of a unitary operator U can be written as $e^{2\pi\phi i}$. To understand the construction of this algorithm, it is further necessary to realise that ϕ can be decomposed as an expansion in binary, i.e. $\phi = \frac{\phi_1}{2} + \frac{\phi_2}{2^2} + ... + \frac{\phi_n}{2^n}$, or, in another simpler notation as $0.\phi_1\phi_2...\phi_n$. The action of the IPEA over the state register is exactly the same as QPEA, however, ϕ is calculated bit by bit (ϕ_k by ϕ_k, with k ranging from 1 to n). To perform such a calculation in this bit by bit manner, it is necessary to include a rotation factor based

[7] The depth of the circuit can be defined as the length of the longest path from the initialisation to the measurement (including eventual initialisation gates), where the path steps are gates.

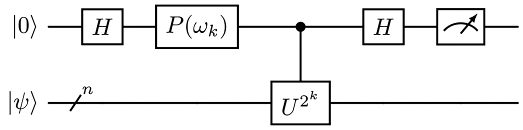

Figure 3: Generic circuit of the k^{th} iteration of IPEA. Where the gate P is the rotation around z gate, defined as $P(\omega) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\omega} \end{pmatrix}$, and the factor ω_k depends on the previous iterations outcomes and is calculated accordingly to Equation 14. As in Figure 2, the controlled operation U^{2^k} kickbacks the k^{th} bit of the phase to the control qubit.

on the previous outcomes, as we shall now describe. The IPEA algorithm is described below, and its k^{th} iteration circuit is presented in Figure 3.

1. Run the $k = 0$ circuit with a null rotation ($\omega_k = 0$) to find the less significant bit, i.e. the rightmost bit.
2. Repeat the following procedure while $k < n$, with n being the number of desired classical phase bits:
   (a) Increment k;
   (b) Calculate the rotation angle ω_k for the kth circuit via
   $$\omega_k = -2\pi \sum_{x=0}^{k-1} \frac{b_x}{2^{k-x+1}}, \tag{14}$$
   where b_x refers to the measurement outcome of the x^{th} circuit, it can be 0 or 1.
   This formula is analogous to the binary decomposition $\phi = \frac{\phi_1}{2} + \frac{\phi_2}{2^2} + ... + \frac{\phi_n}{2^n}$, with the number of parcels increasing in each step, e.g. for $k = 4$ $\omega_4 = -2\pi\left(\frac{b_0}{2^5} + \frac{b_1}{2^4} + \frac{b_2}{2^3} + \frac{b_3}{2^2}\right)$, where b_0 is the least significant bit, this is the binary decomposition ϕ with four bit up to a $-\pi$ factor. From a geometrical point of view, the most significant bit determines a $\frac{\pi}{2}$ angle, the second most significant a $\frac{\pi}{4}$ angle, this means that at each iteration we start with a rotation that traduces the information already retrieved;
   (c) Run the k^{th} circuit with the phase correction ω_k to find b_k.
3. Recover the phase ϕ from its binary decomposition.

Please note that when we say "run the circuit", it is implicit that the circuit is both constructed and measured to retrieve the results.

Finally, note also that in this IPEA approach, since the phase is calculated bit by bit, the value of the phase is simply determined by the most probable value found in the last iteration.

This concludes our generic presentation of the QPEA and IPEA, and we now move on to a different approach to the computation of energy states for a given Hamiltonian, in particular, its ground state.

3.3. Variational quantum eigensolver

The VQE [3] is a hybrid (it requires both a quantum computer (to execute the quantum circuits) and a classical computer (to optimise the parameters of the quantum circuit)) algorithm that also aims to find the ground state of a given system. QML is a computation paradigm briefly defined as the interplay of quantum computing and machine learning ideas. In turn, machine learning algorithms can be defined as algorithms that can learn, and, so able to complete tasks without specific orders. Thus, the VQE is considered a particular type of QML due to the fact that it is a hybrid quantum-classic algorithm that learns the eigenvalues of a target operator.

This algorithm requires as many qubits as the dimension of the physical system, i.e. the dimension of the state register in the previous algorithms, and is made up of four essential components which are described below in detail. A schematic view of the VQE algorithm is presented in Figure 4.

1. **Hamiltonian decomposition** – One of the requirements of this algorithm is to have a decomposition of the Hamiltonian into a sum of Pauli strings. This task is trivial in simple spin problems, as discussed in Section 2, but in many other kinds of physical systems, this can be a hard step, both computationally and analytically. In symbols, the Pauli string decomposition amounts to writing the Hamiltonian as
   $$\hat{H} = \sum_{j=1}^{P_N} w_j \hat{P}_j, \tag{15}$$
   where P_N is the number of Pauli strings in the decomposition, w_j are a set of weights associated with each string $\hat{P}_j = \{I, X, Y, Z\}^{\otimes N}$. N stands for the number of qubits in the register (what we colloquially call the size of the system) and I is the identity matrix.

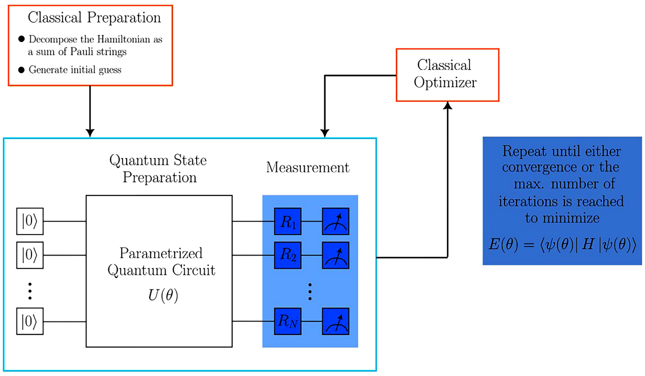
Classical Preparation
● Decompose the Hamiltonian as a sum of Pauli strings
● Generate initial guess
Classical Optimizer
Quantum State Preparation
Measurement
$|0\rangle$
$|0\rangle$
$|0\rangle$
Parametrized Quantum Circuit
$U(\theta)$
R_1
R_2
R_N
Repeat until either convergence or the max. number of iterations is reached to minimize
$E(\theta) = \langle\psi(\theta)| H |\psi(\theta)\rangle$

Figure 4: Structure of the variational quantum eigensolver. Where the parameters θ are randomly initialised and the gates R_i stand for the rotations necessary to measure along an arbitrary direction, with i ranging from 1 to N.

The reason why this decomposition is crucial can be understood after studying the additional components, and especially the measurement part of the algorithm.

2. **Ansatz and initial state preparation** – Before explaining this step, we shall clarify that in the quantum computing context, more precisely in the field of variational circuits, an ansatz circuit is a parametrised circuit used to prepare a trial state to solve the problem. Therefore, an important step of VQE is the choice of an ansatz and the initial state preparation. A good ansatz should be able to span over the parts of the Hilbert space that contain the solution of the problem [3]. For this purpose, there are some well-known ansatzs, such as the hardware efficient ansatz [4] or the unitary coupled cluster ansatz [5]. When the number of utilised qubits is large, this kind of algorithm can exhibit barren plateus [6], i.e. the gradient exponentially vanishes and so the optimisation (training) of the parameters becomes impossible. This means that a careful choice of ansatz is a crucial step, attending to both expressibility (span of possible states within the Hilbert space) and trainability (avoiding barren plateaus) [7].

   In this particular presentation, we opt for an ansatz composed of rotations around the x, y, and z axis[8] and CX gates, usually used in quantum machine learning problems. The circuit that implements this anstaz is in Figure 5.

   The angles of the rotations are the classical trainable parameters and their initial value is randomly selected. The initial state can, for instance, be chosen to be the usual one, i.e. all the qubits starting in the state $|0\rangle$.

3. **Measurement** – In each step of the algorithm, the circuit should be executed and measured as many times as the number of Pauli strings in the decomposition of the Hamiltonian. For each Pauli string, the qubits should be measured along the axis determined for the components of the string, e.g. if we have 3 qubits and the Pauli string $X \otimes Y \otimes Z$, the first qubit should be measured along the x axis, the second along the y axis and the third along the z axis.

   Please note that the default measurement direction is the z direction. Therefore, to measure along this direction, no rotations are necessary, to measure along x is necessary to apply a Hadammard gate since $X = HZH^\dagger = HZH$, or any other equivalent transformation, and to measure along the y direction is necessary to apply an $R_y(-\frac{\pi}{2})$, since $Y = R_y(-\frac{\pi}{2}) Z R_y(-\frac{\pi}{2})^\dagger = R_y(-\frac{\pi}{2}) Z R_y(\frac{\pi}{2})$, or any other equivalent transformation.

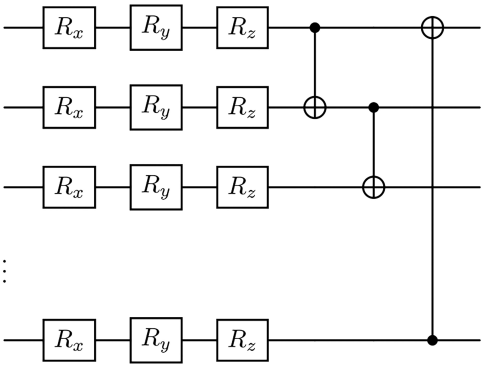

Figure 5: Variational quantum eigensolver circuit Ansatz. R_x, R_y, and R_z are rotations around the x, y, and z axis, respectively and the qubits entangled through the use of CX gates. CX gates are controlled X gates, i.e. a Pauli x gate (X) is applied on the target qubit, if the control qubit is in the state $|1\rangle$.

   After performing all such measures, the expectation value (E) can be obtained via a weighted sum of the measurement of each Pauli string (E_j), i.e.

$$E = \sum_{j=1}^{P_N} w_j E_j, \tag{16}$$

   w_j are a set of weights associated with each $\hat{E}_j$, the exactly ones of Equation 15.

4. **Classical optimisation** – The angles in the ansatz rotations are classically optimised according to the previous measurements to minimise the expectation value (Equation 16). The choice of the classical optimiser defines how the optimisation is done.

3.4. Application to the Heisenberg model

After presenting the generic structure of QPEA, IPEA, and VQE, we are ready to describe and define some details in order to find the eigenvalues of the two-site Heisenberg model. The J factor in Equation 2 is fixed as 4 to avoid unnecessary multiplicative factors, and so the implemented Hamiltonian is simply $H = X_1 \otimes X_2 + Y_1 \otimes Y_2 + Z_1 \otimes Z_2$.

For the first two algorithms, it is necessary to showcase (i) how we can extract the energy from the retrieved phase; (ii) determine the form of the operator U as a quantum circuit in the context of this physical system; (iii) pick how many bits are used to determine the phase. When it comes to the VQE analysis, one would also

[8] The rotations around the x, y and z axis are defined as $R_x(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$, $R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$ and $R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}\cdot} \end{pmatrix}$.

in principle be worried about the necessity of rewriting the Hamiltonian as a sum of Pauli strings. Fortunately, recall that for the two-site Heisenberg model, this was already done in Section 2, as the Hamiltonian is, almost by construction, of this form. Furthermore, one must specify the number of qubits, the maximum number of iterations, the convergence criteria, as well as how the probability distribution is translated into an expectation value.

Starting with point (i) above, for the two-site Heisenberg model, the operator $U(\tau)$ is the time evolution of the Hamiltonian in Equation 2, i.e

$$U = e^{-\frac{i}{\hbar}Ht}. \tag{17}$$

If $|\psi\rangle$ is an eigenstate of H, applying U to $|\psi\rangle$ leads to

$$e^{-iH\tau}\,|\psi\rangle = e^{-iE\tau}\,|\psi\rangle, \tag{18}$$

where $\tau = \frac{t}{\hbar}$ is the evolution parameter utilised in the algorithm application. As already mentioned, the eigenvalues of a unitary operator can always be written as $e^{2\pi\phi i}$, which means that $e^{2\pi\phi(\tau)i} = e^{-iE\tau}$ and $2\pi\phi(\tau)i = -iE\tau$. Note that the phase dependence in τ appears due to the fact that the operator U is, itself, time dependent. With this, the eigenenergies can be determined through

$$E = -2\pi\frac{d\phi(\tau)}{d\tau}. \tag{19}$$

Onto point (ii), in order to implement the time evolution of H, the operator $U(\tau)$ needs to be defined as a quantum circuit (a sequence of quantum gates). Note that the time dependence of $U(\tau)$ is made explicit, since the time parameter must be thought of as an input value to the circuit, and the algorithm should be implemented separately for different values of τ in order to get the energy through Equation 19. In the case of the two-site Heisenberg model, the Pauli strings composing the Hamiltonian of Equation 2 commute with each other, and so the operator can be directly mapped into a quantum circuit. When Pauli strings in the Hamiltonian decomposition do not commute, a Trotter-Suzuki approximation is necessary [9]. Following the procedures in [1, 2], the quantum circuit definition of $U(\tau)$ is represented in Figure 6.

Finally, onto point (iii), we pick an amount of three counting qubits with the aim of hopefully finding the desired energies to a somewhat acceptable degree of precision without introducing significant noise. Furthermore, the size of the state register is simply set to two since we are dealing with a model of two physical spins.

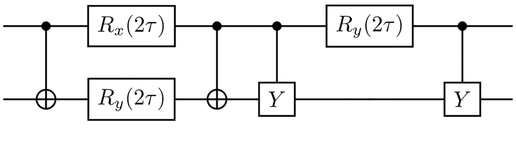

Figure 6: Time evolution of the Hamiltonian in Equation 2 defined as a quantum circuit.

Moving on to the VQE, the number of utilized qubits is also set to two for the exact same reason, i.e. because the problem has a Hilbert space of dimension 2^2. The classical optimiser we made use of – `COBYLA` – is based on the (C)onstrained (O)ptimisation (BY) (L)inear (A)pproximation method [8] (pp. 51–67). We set the maximum number of iterations to 150 and the final desired accuracy is 1×10^{-1}.

In Subsection 3.3, we mention that in each step we need to run the circuit as many times as the number of Pauli's strings in the Hamiltonian decomposition, and the measurement direction is determined by the Pauli matrices in each string. Attending the decomposition in Equation 2, we need to run the circuit three times in each step and measure both qubits along the x, y, and z axes, once in each run.

The resulting output of interest after each run is the probability distribution of the basis state, i.e., for two qubits, the probabilities associated with the states $|00\rangle$, $|11\rangle$, $|01\rangle$ and $|10\rangle$. Consequently, for a measurement along one of the axes, the expectation value (the energy E) (see Appendix C) is given by

$$E = P_{00} + P_{11} - P_{01} - P_{10}, \tag{20}$$

where P_{ij} is the probability of the state $|ij\rangle$. Note that the state $|0\rangle$ has an eigenvalue of 1 and the state $|1\rangle$ of -1.

Now, after presenting all of the theoretical framework, we are primed to implement all three algorithms in the context of finding the energy levels of a physical system, namely, the two-site Heisenberg model.

3.5. Results

The application of the three algorithms presented in the previous subsections to the two-site Heisenberg model was executed both on a real quantum device or a simulator with the noise model of a real device and on a quantum device simulator. We decided to use the 127-qubit `ibm_brisbane` device, which is one of the IBM Quantum Eagle processors, and the `qasm_simulator`. The number of shots in each execution was 20000, which is the maximum number allowed in the open access plan, and, when running on a real device, the chosen optimisation level was 3 (the maximum one).

The probability distributions of the phase values obtained using the QPEA were post-processed with two different methodologies. One where only the most probable value for each time parameter was considered, which results are in Figure 7, and one where the value of the phase was retrieved using circular statistics (Equation 12 and Equation 13), Figure 8.

Using the QPEA, for both the post-processing methods, we could not retrieve the energy values on the

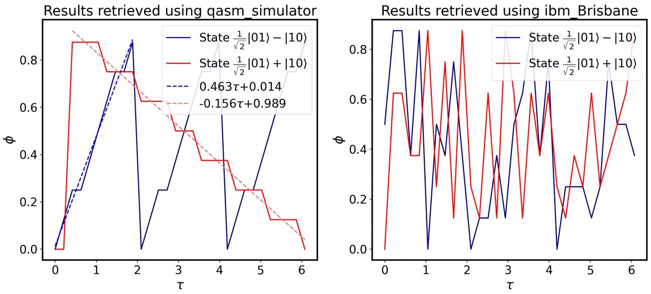

Figure 7: Phase values retrieved with QPEA for different time parameters, using both a real device (right), and a simulator (left). The value of the phase for each time parameter is picked as the most probable one. When the curves show a clear trend, a linear regression is performed. When multiplied by -2π (Equation 19), the slope of the adjusted straight line gives the eigenenergy. Attending to the slope values, the eigenenergies obtained using the simulator are 0.980 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$ and -2.909 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$.

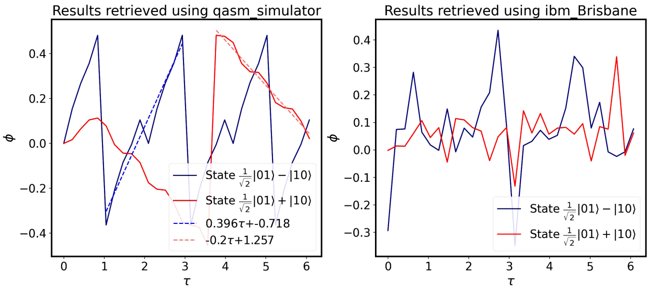

Figure 8: Phase values retrieved with QPEA for different time parameters, using both a real device, figure on the right, and a simulator, figure on the left. The value of the phase for each time parameter is obtained using circular statistics (Equation 12 and Equation 13). When the curves show a clear trend, a linear regression is performed. When multiplied by -2π (Equation 19), the slope of the adjusted straight line gives the eigenenergy. Attending to the slope values, the eigenenergies obtained using the simulator are 1.257 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$ and -2.488 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$.

real device, from the phase variation in times, since the curves do not show a clear trend. Although this behaviour is not the desired one, it is the expected one since the required quantum circuits are deep, and so, the results are severely affected by noise. The right panel of Figure 8 shows that the phase approximately varies around the 0 value, which reflects the fact that the circuits are significantly affected by noise, since the noise mean value should be 0 and when the noise overpasses the signal (the phase value), the result should also have mean 0. Respecting the executions on the simulator, assuming the most probable value for the phase (left panel of Figure 7) the energy value found for the state $\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$ is 0.980 and for the state $\frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$ is -2.909, and, when applying circular statistics the values found are 1.257 and -2.488, respectively. These energy values are calculated from the slope of the regression line through Equation 19, and the linear regression is performed using `Scipy` [40]. As calculated in Section 2, with $J = 4$, the expected energies are -3 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$ and 1 for $\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$. Comparing with the analytical result, assuming only the most probable phase had led to better results than applying the circular mean, which is not surprising. As the simulator is noise-free, and so its output is the ideal one, i.e. the correct phase has clearly the biggest probability, when applying the circular statistics over the ideal, or a good, probability distribution, we are decreasing the performance by adding noncorrect phase values. This method would probably bring advantages for executions on real devices that are not ideal but less noisy than the ones found.

Requiring more quantum circuits but each one less deeper than the previous ones, the IPEA was also executed in the same two devices, which results are in Figure 9.

The results found using the simulator (left panel of Figure 9) are, with no surprise, the same ones found when assuming the most probable phase with the QPEA (left panel of Figure 7) since we are simulating in the ideal scenario in both cases. Executing the IPEA on a real device leads to curves similar to the ones found in simulation, and so a linear regression was also performed to extract the energy values, which are 1.037 and -2.859 for the states $\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$ and $\frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$, respectively. These results, found with a real device, are quite close to the analytical ones, i.e. 1 and -3.

Due to the limitations in the access to IBM devices and the large number of accesses required to execute the VQE algorithm, instead of executing in the real

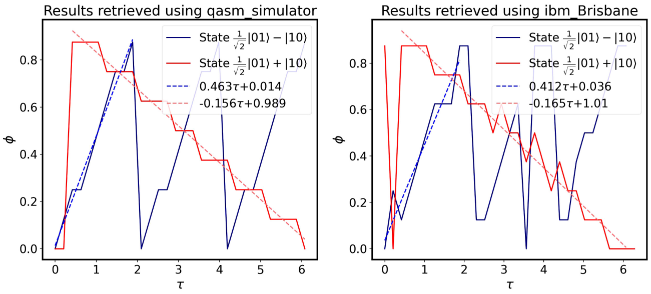

Figure 9: Phase values retrieved with IPEA for different time parameters, using both a real device, figure on the right, and a simulator, figure on the left. The value of the phase for each time parameter is the most probable one. When the curves show a clear trend, a linear regression is performed. When multiplied by -2π (Equation 19), the slope of the adjusted straight line gives the eigenenergy. Attending to the slope values, the eigenenergies obtained using the simulator are 0.980 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$ and -2.909 for the state $\frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$ and using the real device the values are 1.037 and -2.859, respectively.

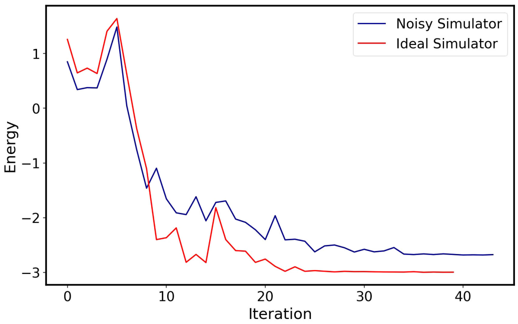

Figure 10: Energy found in each iteration using the `qasm_simulator` with (noisy simulator line) and without (ideal simulator line) the noise model of the `ibm_brisbane`. The minimum energy found with the noisy simulator is -2.679, and with the ideal one is -2.999.

device, we opt to execute in the `qasm_simulator` with the noise model of the `ibm_brisbane` device. The energy found in each iteration when executing the algorithm in both a noisy and an ideal simulators is in Figure 10. The minimum energy found with the noisy simulator is -2.679, and with the ideal one is -2.999, both of these energies are quite close to the analytical one, i.e. -3.

4. Conclusion

In this paper we have presented a pedagogical introduction to the diagonalization or ground state optimization of a quantum system, the spin $S = 1/2$ two-site Heisenberg model, using a quantum computer. We presented three different algorithms, the first two of which, namely the QPEA and the IPEA, allow us to determine the eigenenergy of a given eigenstate. On the other hand, the final algorithm, namely the VQEA allows for an estimate of the minimum eigenvalue of an operator. Considering the Hamiltonian operator for the two-site Heisenberg model, such eigenvalue corresponds to the ground state energy. The values determined using these algorithms, on both real quantum devices and quantum simulators, were bench-marked against the analytical solution.

Using the two-site Heisenberg model as a toy model, we found that all three algorithms have successfully satisfied their purpose when running on ideal simulators. On real devices, however, when using the QPEA, noise severely impacted the results, proving to be dominant over any useful signal. When using the IPEA, the retrieved results were quite close to the desired ones. Due to access limitations to quantum computers, and the large number of accesses required by the VQE, this algorithm was only executed in a simulator with a noise model and not in a real device. The result using this noisy simulator deviated further from the analytical results when compared to the results found with the ideal simulator, but was nonetheless fairly satisfactorily close.

This paper is (a first) and pedagogical introduction, and so some newsworthy and more profound work is yet to be done. One of the following steps, and probably the most direct one, is to implement the same algorithms, namely the IPEA and VQE, for larger models, i.e. for more sites, where the analytical solutions are difficult or do not exist. Other studies that could, and should, be done are to study how the circuit size impacts the noise in the results and how and which error mitigation techniques could lead to more accurate results. Please note that preliminary studies on using the `Qiskit Ignis` native error mitigation techniques showed that the use of these simpler mitigation techniques does not bring significant improvements to the results. In fact, respecting the QPEA the circuits are severely affected by noise, such that even with mitigation techniques the noise surpasses the signal. Therefore, in addition to the introductory character of this article, we chose not to include error mitigation and error treatments. Nonetheless, more complex and profound error mitigation and statistical analysis studies are one of the following steps.

Acknowledgements

The authors acknowledge discussions with Bruno Murta and Pedro Cruz on the early stage of this work. We also acknowledge Nuno Castro, Miguel Romão, Miguel Peixoto and Inês Ochoa for the regular discussions. We thank to Ricardo Ribeiro for kindly providing access to some of the computing resources used. M.G.J.O. and T.V.C.A. acknowledge support from the Portuguese Foundation for Science and Technology (FCT) in the framework of the project CERN/FIS-COM/0004/2021. M.G.J.O. also acknowledges support from the Portuguese Foundation for Science and Technology (FCT) in the framework of the project CERN/FIS-PAR/0032/2021. T.V.C.A. also acknowledges the computational resources provided by the Aalto Science-IT project. N.M.R.P. acknowledges support by the Portuguese Foundation for Science and Technology (FCT) in the framework of the Strategic Funding UIDB/04650/2020, COMPETE 2020, PORTUGAL 2020, FEDER, and through projects PTDC/FIS-MAC/2045/2021 and EXPL/FIS-MAC/0953/2021. N.M.R.P. also acknowledges the Independent Research Fund Denmark (grant no. 2032-00045B) and the Danish National Research Foundation (Project No. DNRF165). We 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.

Supplementary material

The following online material is available for this article:
Appendix A – Quantum Fourier Transform simplification

Appendix B – Quantum Fourier Transform generic quantum circuit
Appendix C – Expectation value from the probability distribution

References

[1] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2011), 10 ed.

[2] P.M.Q. Cruz, G. Catarina, R. Gautier and J. Fernández-Rossier, Quantum Sci. Technol. **5**, 044005 (2020).

[3] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G.H. Booth et al., Phys. Rep. **986**, 1128 (2022).

[4] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J.M. Chow and J.M. Gambetta, Nature **549**, 242 (2017).

[5] J. Romero, R. Babbush, J.R. McClean, C. Hempel, P.J. Love and A. Aspuru-Guzik, Quantum Sci. Technol. **4**, 014008 (2017).

[6] J. Mcclean, S. Boixo, V. Smelyanskiy, R. Babbush and H. Neven, Nat. Commun. **9**, 4812 (2018).

[7] Z. Holmes, K. Sharma, M. Cerezo and P.J. Coles, PRX Quantum **3**, 010313 (2022).

[8] M.J.D. Powell, in: *Advances in Optimization and Numerical Analysis*, edited by S. Gomez and J.P. Hennart (1994).

[9] K.L. Brown, W.J. Munro and V.M. Kendon, Entropy **12**, 2268 (2010).

[10] W. Heisenberg, Zeitschrift für Physik **49**, 619 (1928).

[11] D.C. Mattis, *The Theory of Magnetism Made Simple* (World Scientific, New Jersey, 2006).

[12] H.A. Bethe, Zeitschrift für Physik **71**, 205 (1931).

[13] R.P. Feynman, International Journal of Theoretical Physics **21**, 467 (1982).

[14] R. Vathsan, *Introduction to Quantum Physics and Information Processing* (CRC Press, Boca Raton, 2015), 1 ed.

[15] P.G. Kwiat, K. Mattle, H. Weinfurter, A. Zeilinger, A.V. Sergienko and Y. Shih, Phys. Rev. Lett. **75**, 4337 (1995).

[16] C.S. Wu and I. Shaknov, Phys. Rev. **77**, 136 (1950).

[17] S.J. Freedman and J.F. Clauser, Phys. Rev. Lett. **28**, 938 (1972).

[18] A. Aspect, J. Dalibard and G. Roger, Phys. Rev. Lett. **49**, 1804 (1982).

[19] C.H. Bennett and G. Brassard, Theor. Comput. Sci. **560**, 7 (2014).

[20] C.H. Bennett and S.J. Wiesner, Phys. Rev. Lett. **69**, 2881 (1992).

[21] K. Mattle, H. Weinfurter, P.G. Kwiat and A. Zeilinger, Phys. Rev. Lett. **76**, 4656 (1996).

[22] P.W. Shor, in: *Proceedings 35th Annual Symposium on Foundations of Computer Science* (Washington, 1994).

[23] L.K. Grover, in: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (Philadelphia, 1996).

[24] J. Barata, X. Du, M. Li, W. Qian and C.A. Salgado, Phys. Rev. D **106**, 074013 (2022).

[25] A. Smith, M.S. Kim, F. Pollmann and J. Knolle, Npj Quantum Information **5**, 106 (2019).

[26] A.Y. Kitaev, arXiv:quant-ph/9511026v1 (1995).

[27] M. Cerezo, A. Arrasmith, R. Babbush, S.C. Benjamin, S. Endo, K. Fujii, J.R. McClean, K. Mitarai, X. Yuan, L. Cincio et al., Nat. Rev. Phys **3**, 625 (2021).

[28] J. Preskill, Quantum **2**, 79 (2018).

[29] C.J. O'Loan, J. Phys. A Math. Theor. **43**, 015301 (2009).

[30] L. Zhou, S.T. Wang, S. Choi, H. Pichler and M.D. Lukin, Phys. Rev. X. **10**, 021067 (2020).

[31] A. Perry, R. Sun, C. Hughes, J. Isaacson and J. Turner, arXiv:1905.00282 (2020).

[32] T. Kushimo and B. Thacker, arXiv:2212.03726v3 (2023).

[33] N.S. Yanofsky, arXiv:0708.0261v1 (2007).

[34] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd, Nature **549**, 195 (2017).

[35] H.L. Huang, D. Wu, D. Fan and X. Zhu, Sci. China Inf. Sci. **63**, 180501 (2020).

[36] K. Wintersperger, F. Dommert, T. Ehmer, A. Hoursanov, J. Klepsch, W. Mauerer, G. Reuber, T. Strohm, M. Yin and S. Luber, EPJ Quantum Technol **10**, 32 (2023).

[37] L.S. Madsen, F. Laudenbach, M.F. Askarani, F. Rortais, T. Vincent, J.F.F. Bulmer, F.M. Miatto, L. Neuhaus, L.G. Helt, M. Collins et al., Nature **606**, 75 (2022).

[38] H. Haffner, C.F. Roos and R. Blatt, Phys. Rep. **469**, 155 (2008).

[39] D. Loss and D.P. DiVincenzo, Phys. Rev. A **57**, 120 (1998).

[40] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright et al., Nat. Methods **17**, 261 (2020).

[41] S. Garnerone, A. Marzuoli and M. Rasetti, Laser Physics **16**, 1582 (2006).

[42] M.C. Peixoto, N.F. Castro, M.C. Romão, M.G.J. Oliveira and I. Ochoa, Front. Artif. Intell **6**, 1268852 (2023).

[43] W. Rabelo and M. Costa, Revista Brasileira de Ensino de Física **40**, e4306 (2018).

[44] G.F. de Jesus, M.H.F. Silva, T.G.D. Netto, L.Q. Galvão, F.G.O. Souza and C. Cruz, Revista Brasileira de Ensino de Física **43**, e20210033 (2021).

[45] A. Canabarro, T. Mendonça, R. Nery, G. Moreno, A. Albino, G.F. de Jesus and R. Chaves, Revista Brasileira de Ensino de Física **44**, e20220099 (2022).

[46] A.N. Oliveira, E.V. Oliveira, A.C. Santos and C.J. Villas-Boas, Revista Brasileira de Ensino de Física **44**, e20210333 (2021).

[47] A. Peruzzo, J. McClean, P. Shadbolt, M-H. Yung, X-Q. Zhou, P.J. Love, A. Aspuru-Guzik and J.L. O'Brien, Nat Commun. **5**, 4213 (2014).

[48] D. Wecker, M.B. Hastings and M. Troyer, Phys. Rev. A **92**, 042303, (2015).

[49] D. Jiang, X. Liu, H. Song and H. Xie, in: *IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (Xian, 2021).

[50] H. Mohammadbagherpoor, Y.H. Oh, P. Dreher, A. Singh, X. Yu and A.J. Rindos, in: *IEEE International Conference on Rebooting Computing (ICRC)* (San Mateo, 2019).

[51] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li and J. Tennyson, Physics Reports **986**, 1 (2022).