UNIVERSIDAD NACIONAL DE COLOMBIA

Ground State Energies of H_2 Using Variational Quantum Circuits

Sergio Andrés Cotrino Sandoval

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Física
Bogotá, Colombia
2024

Ground State Energies of H_2 Using Variational Quantum Circuits

Sergio Andrés Cotrino Sandoval

A thesis submitted in fulfillment of the requirements for the degree of:
Magíster en Ciencias - Física

Director:
Ph.D. Carlos Leonardo Viviescas

Line of Research:
Quantum Computing
Research Group:
Caos y Complejidad

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Física
Bogotá, Colombia
2023

A mi madre

Acknowledgements

I would like to express my deepest gratitude to my advisor, Carlos Viviescas, for his precise and helpful guidance through this research work. With his concise and insightful advice, I was able to identify both potential pitfalls and opportunities alike, while bringing the writing process to a good end. I'm also thankful to Arcesio Castañeda Medina, Senior computational physicist at Fraunhofer ITWM , for the fruitful discussion regarding the execution and analysis of quantum circuits. Additionally, I want to give thanks to Andrés Reyes, professor of Chemistry at Universidad Nacional de Colombia, for his advice regarding molecular Hamiltonians and useful perspectives regarding the analysis of exotic molecules.

Nunca dejaré de agradecer el apoyo incondicional de mi madre. Su fortaleza no deja de sorprenderme, su paciencia me impresiona. Ella ha sido la piedra firme en la vida de muchas personas, ha servido de muchas formas para que familia y amigos crezcan. Para el mundo, la Tia Alicia; para mi, mi mami; siempre presente, siempre atenta, siempre fuerte, siempre dispuesta a ayudar.

A Miguel por su apoyo oportuno a un mensaje de distancia; a Kristian por su soporte constante alrededor música y películas; a Mao por siempre preguntar cómo vamos; a Julian por el vínculo perpetuo; a Cristian, Mariana y Sebastian por estar y compartir la carga.

A mi familia, y a todo el preguntó ¿Cómo va la tesis?

Abstract

Title: Ground State Energies of H_2 Using Variational Quantum Circuits

Considering the current limitations on size and reliability of Noisy Intermediate Quantum Scale devices, Variational Quantum Circuits offer a way to get useful results from quantum computation. On top of that, Machine Learning methods using quantum data offer a way to process the information, but also use it to learn and extract useful information. Meta-Variational Quantum Eigensolver (meta-VQE) was used to learn the ground energy profile of a molecule using a set of training points. By training an ansatz circuit using a non-linear Gaussian encoding of each circuit parameter and setting the interatomic distance as a free parameter, it was possible to find a good approximation of the ground energy of the system for any interatomic distance within a certain region. This method also has the advantage to produce good starting parameters to train using standard VQE, and obtain even better results (opt-meta-VQE). Meta-VQE was implemented in an analytic noise-free simulation and a 10000 shots-based simulation using the software framework for quantum computing PennyLane. In the analytic simulation, it was possible to accurately describe the potential energy surface of an H_2 molecule within chemical accuracy, using a hardware inspired ansatz and the ADAM optimizer. With the 10000 shots-based simulation, the method is capable to approximate the energy profile, but in general its performance is not as good as the analytical approach due to the variability on the samples obtained. Meta-VQE provides a novel way to extract and produce information by learning using quantum data from variational circuits.

Keywords: quantum computing, quantum chemistry, quantum circuits, Variational Quantum Eigensolver, PennyLane, quantum machine learning.

Resumen

Título: Energías de Estado Fundamental de H_2 Usando Circuitos Cuánticos Variacionales

Teniendo en cuenta las limitaciones actuales de tamaño y confiabilidad de los dispositivos de escala cuántica intermedia ruidosa, los circuitos cuánticos variacionales ofrecen una forma de obtener resultados útiles de la computación cuántica. Además de eso, los métodos de aprendizaje automático que utilizan datos cuánticos ofrecen una forma de procesar la información, pero también de usarla para aprender y extraer información útil. Se usó el metodo de meta-autosolucionador cuántico variacional (meta-VQE, por sus siglas en inglés) para aprender el perfil de energía fundamental de una molécula usando un conjunto de puntos de entrenamiento. Al entrenar un circuito usando una codificación gaussiana no lineal de cada parámetro del circuito y estableciendo la distancia interatómica como un parámetro libre, fue posible encontrar una buena aproximación de la energía mínima del sistema para cualquier distancia interatómica dentro de una región determinada. Este método también tiene la ventaja de producir buenos parámetros de partida para entrenar usando VQE estándar y obtener resultados aún mejores (opt-meta-VQE). Meta-VQE se implementó en una simulación analítica sin ruido y una simulación basada en 10000 muestras utilizando el software para computación cuántica PennyLane. En la simulación analítica, fue posible describir con precisión la superficie de energía potencial de una molécula H_2 con precisión química, utilizando un ansatz inspirado en hardware y el optimizador ADAM. Con la simulación basada en 10000 muestras, el método es capaz de aproximar el perfil de energía, pero en general no funciona tan bien como el enfoque analítico debido a la variabilidad de las muestras obtenidas. Meta-VQE proporciona una forma novedosa de extraer y producir información mediante el aprendizaje utilizando datos cuánticos de circuitos variacionales.

Palabras clave: computación cuántica, química cuántica, circuitos cuánticos, PennyLane, autosolucionador cuántico variacional, aprendizaje automático cuántico

Content

List of Figures

List of Tables

1 Introduction

Quantum theory revolutionised the XX century, and most of the technology coming from that boom shapes the current world. This makes even more interesting that, right now, there is a major momentum in both academia and industry pursuing the next breakthrough in quantum-based technology:*quantum advantage*, a demonstration that a programmable quantum processor can solve problems and verify the solutions, in a way that no classical device can solve in a feasible amount of time. The idea is usually attributed to Feynman [29], and it has received strong support through the last few decades such as the DiVincenzo Criteria [24], the development of quantum algorithms [68, 50, 10], and indirectly the widespread of the Quantum Information theory brought by the textbook of Nielsen and Chuang [53] at the beginning of the century. At the moment, multiple initiatives with private and public funding create a potential market of US$104 billion [1], even with the current lack of commercial practical uses. IBM, one of the big players in the field, even published a report in 2021 naming the present time as *The Quantum Decade*, in which proposed a promising path from quantum awareness to quantum advantage for the next immediate years.

Although real quantum processor currently suffer from noise and scale problems, in what Preskill named the Noise-Intermediate State Quantum (NISQ) era [61], this is not impediment for big and small players in the field chasing practical results of these limited devices. The current noisy qubits can provide useful results when using error mitigation and clever approaches for circuit execution (take as reference these state-of-the-art works from Google [51] and IBM [36]). Hybrid devices, combining the edge of the quantum device with the power and reliability of a classical processor are also powerful candidates to the realization of near-term quantum applications [47]. Besides this, there are also initiatives looking for better, more-qubits handling physical processors based on multiple physical principles such as superconducting architectures [38], photonics [5], trapped ions [60], and several more. In addition to all of that, there are multiple Software Development Kits (SDKs), most of them open-sourced, that allow end-users to simulate quantum circuits, or even execute processes in real ones. All the resources mentioned create the perfect scenario in which sustained research and experimentation from different parties with varied backgrounds can reach a breakthrough.

The problems to solve are of high interest for academia and business alike. Big business players have an eye on the possibilities of quantum computing [63, 12], impulsing leaders in different markets to be ready for the breakthrough promised by quantum computing in upcoming years. There are multiple areas that can, and hopefully will, be impacted by this new paradigm: simulations procedures for quantum chemistry and material science, energy and drug discovery; processing linear systems to enhance machine learning, which in turn can be applied to such varied markets as streaming services or stock prices; optimization procedures that allow to take the optimal decision in order to attain a specific objective; even communications through the use of quantum networks and quantum cryptography.

Along with the advent of quantum era computing, there is another hot topic that cannot be ignored: Machine Learning (ML) and Artificial Intelligence (AI). The influence of AI goes from generative algorithms offered by the major computing companies to produce image and text, to affect movie productions in and out of the screen, and even sparking multiple concerns regarding the regulation of content produced by such features. Again, there are a myriad of big and small players in the field offering Artificial Intelligence services, from which ChatGPT language model developed by OpenAI might be the best known. In the area of academic research, machine learning has been a powerful tool for interpretation, manipulation, and approximation of data into relevant insight [39]. What is even more interesting is the use of ML as inspiration to create new possible theories and experimental setups [30, 49].

Both fields join neatly into Quantum Machine Learning (QML) [27, 67]. The union of both areas is wide, applying the processing of quantum data on classical devices, or even classical data in quantum devices; also using quantum data to learn information regarding complex quantum systems and exploit possible results from noisy qubits by training and optimisation of Variational Quantum Circuits (VQC). From all the possibilities, VQC poses as the most promising approach to make quantum processors useful in the NISQ era, offering a way to train circuits and obtain information from quantum systems [14]. It is even common to draw parallels between quantum networks and VQCs considering their flexibility and fine-tuning capabilities.

Having into account these properties, Quantum Machine Learning oriented to computational chemistry has a great potential to overcome the practical challenges in simulating quantum systems on classical computers and NISQ devices [13, 46]. The exponential growth of the dimension of the wavefunction makes manipulation and storage impractical in classical devices. And, at the same time, state-of-the-art quantum processors don't have the accuracy, along with the error correction features, required for useful analysis of complex chemical systems. QML has the chance to deal with these issues by using hybrid architectures, where the properties of a quantum processor are exploited by the use of variational algorithms, and the computational heavy-lifting is done in a classical device. To find the ground state of a molecular system, Variational Quantum Eigensolver (VQE) [73] is a useful approach to study specific configurations. On top of that, the meta-VQE method [16] represents one of the first attempts to actually learn the energy profile of the device, and not just solve an optimization problem. This is the main goal of this work: to study in detail the learning capabilities of the meta-VQE method over a simple molecule.

In this work, the results and performance of variational quantum algorithms for studying basal energies of a H_2 molecule are analysed, considering as variables the circuit layout, gradient calculation and optimization method. In figure **1-1**, the main steps of an hyrid Variational Quantum Algorithm are depicted, where it's key to identify that both quantum and classical processors work together to solve the problem. The minimal energy of the molecule, produced by the expectation value of the Hamiltonian describing the molecule configuration, is obtained for multiple interatomic separation distances implementing VQE and meta-VQE methods in Pennylane [7], an open-source software framework for quantum machine learning. The circuit was trained using analytic modelling of the produced trial states of the quantum systems, but also was considered the expectation value calculation produced by the probability distribution function obtained from 10000 runs (or shots) of the quantum circuit. It was possible to train the circuits using VQE to obtain ground energies of the molecule within chemical accuracy compare to the Full Configuration Interaction value, where the best results were obtained using ADAM optimizer and problem-inspired circuits built upon Single and Double excitations or Coupled Cluster theory. Regarding meta-VQE and further optimization from meta-VQE results (opt-meta-VQE), there was no advantage proven against standard VQE, which might be caused by a very simple definition of the meta-VQE aggregated cost function or some required fine tuning of the optimizer due to the change in nature of the cost function. In the analytic approach using meta-VQE, it was possible to learn the energy profile of the molecule and achieve chemical accuracy for most points using, again, ADAM and problem-inspired circuits. When the shots-based approach was used, ADAM and problem-inspired circuits produce the smallest error but the performance is significantly impacted.

The present work lays the fundamentals for more sophisticated projects involving larger molecules, calculation of another molecular properties (such as excited state energies, dipole moments, electronic charge distributions, to name a few), and the possibility of analysing exotic molecules composed by antimatter particles [17]. In this last regard, there are some works aimed to analyse positronium hydride [56, 57], and there is a lot more to be done. In particular, QML is well suited to accelerate the identification of energy stability in molecules composed by antimatter particles.

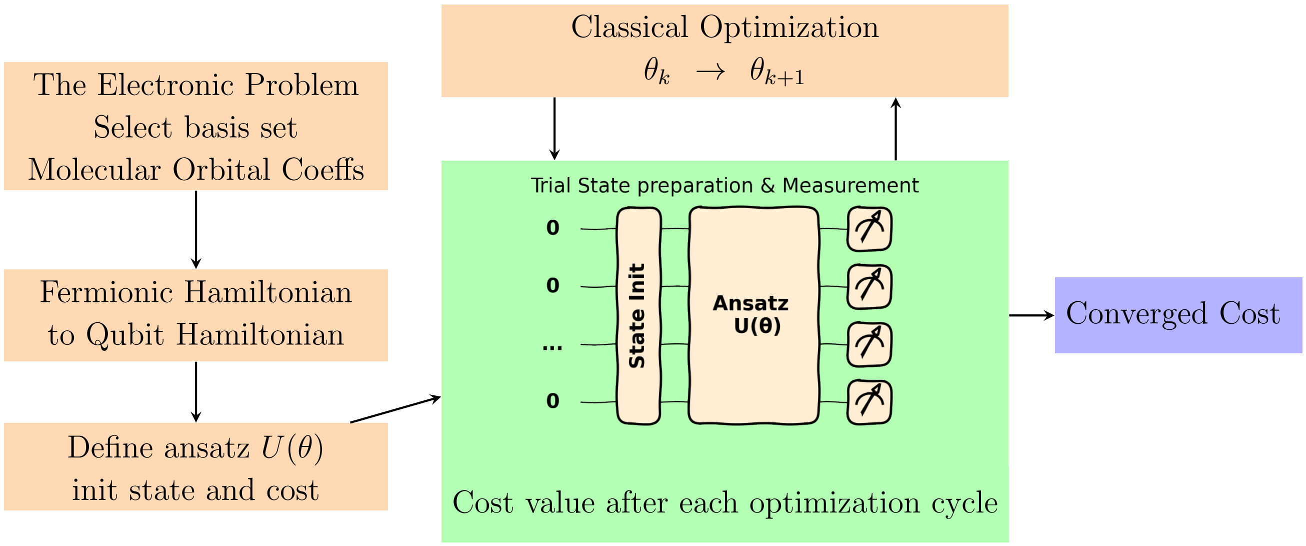


Fig. 1-1: Diagram of Variational Quantum Algorithm in an hybrid architecture, used to get the ground energy of a molecular system. The yellow boxes represent calculations and executions in a classical device; the green box represent sections where quantum processors are required. Initially, the electronic problem of finding the wavefunction for a molecule is approached using tools from quantum chemistry: expressing the hamiltonian of the system using fermionic operators. After that, the Hamiltonian is transformed to qubit operators, and then a parametrized circuit ansatz, initial state and cost function are defined. Then, the quantum processor is trained to minimize the cost function by finding the optimal values for the parameters, until it converges.

2 Quantum Mechanics and Quantum computing

Quantum mechanics deals with the behaviour of particles so small that humans are unable to directly visualise them at all. That makes more impressive the titanic achievements of theoretical and practical work based in a theory built in the mathematical abstractions to understand electrons and their motion. Having this in mind, it is tempting to think that the theory is able to predict the behaviour of any chemical compound. However, as the first attempts to describe molecular systems in the 1920's were made, it was clear that it was not the case. The complexity of the wavefunction of a quantum system, growing exponentially with the number of particles, limits classical computing to efficiently simulate some interesting quantum systems.

That is why quantum chemistry poses as one of the most compelling targets for quantum computing [52]. The production of theoretical methods using quantum devices have been in pair with the growth and robustness of quantum processors [13]. However, to gain some insight regarding why quantum computing might hold an advantage in quantum chemistry, it is necessary to review some key concepts of quantum mechanics.

2.1. Single and Multiple Qubits

In classical computers, information is processed and stored in bits. A *bit* is always in one of two different states, which leads naturally to the common binary representation of 0 or 1 (and hence its name, **binary digit**). The entire current state of computation is a product of humanity learning to manipulate (efficiently, thanks to clever algorithms) enormous amounts of bits to execute processes. A *qubit* is a quantum analogy to a bit, and just as the classical bit, it has a state. Two possible states for a qubit are the states $|0\rangle$ and $|1\rangle$ in the *Dirac Notation*, which might correspond to the classic 0 and 1 values of a bit. These states form what is commonly called as the *computational basis*. However, a qubit can also be in a state other than $|0\rangle$ and $|1\rangle$; it can be in a *superposition* state or *linear combination* of states

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1. \tag{2-1}$$

The complex numbers α and β are known as the probability *amplitudes* of each corresponding state. When a superposition state is measured in a suitable way, the resulting state can be either $|0\rangle$ with a probability of $|\alpha|^2$, or $|1\rangle$ with a probability of $|\beta|^2$. Naturally, $|\alpha|^2 + |\beta|^2 = 1$, as the probabilities must sum to one. The possibility that a qubit can be in a superposition is the origin of the counter intuitive nature of quantum mechanics, but also of the great potential it has into quantum computing. A qubit can exists in a continuum of states *between* $|0\rangle$ and $|1\rangle$ until it is measured, resulting in either $|0\rangle$ *or* $|1\rangle$.

The states of the computational basis can be represented as column vectors, having the advantage of having unit length and being orthogonal

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \quad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

with corresponding elements of the dual space represented as row vectors

$$\langle 0| := \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \end{bmatrix} , \quad \langle 1| := \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 \end{bmatrix} .$$

In general, a random state $|\psi\rangle$ can be expressed as

$$|\psi\rangle = e^{i\gamma} \left(cos\,(\theta/2)\,|0\rangle + e^{i\phi} sin\,(\theta/2)\,|1\rangle \right) \tag{2-2}$$

where the parameters θ, γ and ϕ are real numbers. The parameter θ determines the probability that the returning state is $|0\rangle$ or $|1\rangle$ after the measurement, while the parameter ϕ is the relative phase of the qubit. The parameter γ is the global phase, which does not have observable effects in the measurement process and can be disregarded for the purposes of this work.

The values of θ and ϕ define a point on the unit three-dimensional sphere called the *Bloch Sphere*. This tool provides a useful means of visualising the state of a single qubit, as well as the effect of an operation on it.

In the case of having two qubits, there are four computational basis states denoted by $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. A pair of qubits can also have a superposition state of these four states, described with the linear combination

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle\,. \tag{2-3}$$

The states of the computational basis for multiple qubits can be represented using the Kronecker product. For example

$$|01\rangle := |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} .$$

Again, as in the case for one qubit, the condition that the probabilities of measuring each basis state sum up to one is condensed by the normalisation condition

$$\sum_{x \in \{0,1\}^{\otimes 2}} |\alpha_0|^2 = 1, \tag{2-4}$$

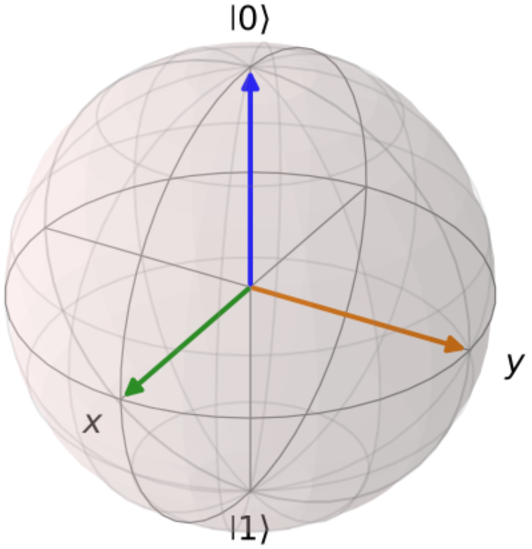


Fig. 2-1: Bloch sphere displaying one qubit states in each of the three axes. The vertical arrow is the state $|0\rangle$, over the x axis is the state $(|0\rangle + |1\rangle)/2$, and the y axis holds the state $(|0\rangle + i\,|1\rangle)/2$. Drawing made using the Python library QuTiP [34].

and so the resulting state is ensured to be one of $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$.

For a multi qubit system, a subset of the qubits can be measured, which can have one out of two values ($|0\rangle$ or $|1\rangle$) and leaves a post-measurement state according to the result. For example, if the first of a two qubit system is measured, that qubit can be $|0\rangle$ with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$, leaving a post-measurement state

$$|\psi_{0x}\rangle = \frac{\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}, \tag{2-5}$$

where the denominator factor ensures the state still satisfies the normalisation condition.

A multi qubit system is *entangled* if it cannot be represented by a product of states for each single qubit. This fairly simple definition contains one of the key components of quantum mechanics: entanglement. The main feature of a product of states is that each subsystem behaves independently of the other. For example, the following product state

$$|\psi\rangle = \{\alpha_0\,|0\rangle + \alpha_1\,|1\rangle\} \otimes \{\beta_0\,|0\rangle + \beta_1\,|1\rangle\} \tag{2-6}$$

$$= \alpha_0\beta_0\,|00\rangle + \alpha_0\beta 1\,|01\rangle + \alpha_1\beta 0\,|10\rangle + \alpha_1\beta_1\,|11\rangle\,, \tag{2-7}$$

where, good enough, the product state is automatically normalised given that the original states were normalized as well. Making some operation on a subset (e.g. the first qubit) of the product state gives exactly the same result as if it was made on the original, isolated state.

An entangled state is something else entirely. As it was expressed in equation 2-5, measuring one subset of the general two-qubit state can affect the result of the measurement in the other subset. Notice how the post-measurement state shows different probabilities for the possible results of measuring the second qubit if the first measurement give as a result the state $|1\rangle$

$$|\psi_{1x}\rangle = \frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}, \tag{2-8}$$

and there is a *correlation* between the results of the measurements in both subsets.

An important two qubit state showing this feature is the *Bell State*

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{2-9}$$

which has the property that, upon measuring the first qubit, the result of measuring the second one is completely determined. That is, the measurement outcomes are completely correlated.

For a system of n qubits, the computational basis states are of the form $|x_1 x_2 ... x_n\rangle$, and a quantum state of such a system is specified by 2^n complex coefficients, or amplitudes. For $n = 500$, that number is larger than the estimated number of atoms in the Universe. And here lies the huge potential of computational power acquired by using physical quantum systems to perform calculations. The only requirement is to learn how Nature keeps tabs of everything and makes such enormous calculations in the first place.

2.2. Single and Multiple Qubit Gates

Just as in classical computing, it is possible to implement logic gates on a single qubit, as well as in an array of multiple qubits. However, the superposition of computational states that can be created while working with qubits give rise to a very interesting feature. While the only non-trivial logic gate for a simple bit is the NOT gate (that interchanges the states 0 and 1 such that $0 \rightarrow 1$ and $1 \rightarrow 0$), the gate U acting on a qubit only needs to keep the state normalized. For example, starting with the state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $|\alpha|^2 + |\beta|^2 = 1$, after the gate has acted the resulting state takes the form

$$U|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle,$$

where $|\alpha'|^2 + |\beta'|^2 = 1$. This is accomplished by requiring that the gate U is *unitary*, that is $U^\dagger U = I$, where $U^\dagger$ is the *adjoint* of U (obtained by transposing and then complex conjugating U). Logical gates and operators over a system of qubits can be described as square matrices of size $2^n \times 2^n$, where n is the number of qubits in which the gate acts.

An unitary operator on a single qubit can be understood essentially as a state rotation over the surface of the Bloch sphere. This means there is an infinite number of unitary rotations that can act over a qubit. Considering this, an arbitrary rotation can be described by a succession of rotations over each orthogonal axis of the Bloch sphere. This is done using the *Pauli Matrices*

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{2-10}$$

$$\mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{2-11}$$

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2-12}$$

which are Hermitian (self-adjoint matrices) with real eigenvalues ±1, unitary, and equal to the identity matrix when squared ($\mathbf{X}^2 = \mathbf{Y}^2 = \mathbf{Z}^2 = \mathbf{I}$). The anticommutation relation of the Pauli matrices (useful later when representing Hamiltonians of fermionic systems using qubits) is

$$\{\sigma_i, \sigma_j\} = 2\delta_{ij}\mathbf{I}, \tag{2-13}$$

where $\{\sigma_i, \sigma_j\}$ is defined as $\sigma_i\sigma_j + \sigma_j\sigma_i$ and δ_{ij} is the *Kronecker delta*. The commutation relation of the Pauli matrices also comes in handy, and reads

$$[\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k, \tag{2-14}$$

where ϵ_{ijk} is the Levi-Civita symbol. Rotation matrices around the 3 orthogonal axis are obtained by getting the exponent of each Pauli Matrix

$$R_{\mathbf{X}}(\theta) = e^{-i\theta\mathbf{X}/2}, \tag{2-15}$$

$$R_{\mathbf{Y}}(\theta) = e^{-i\theta\mathbf{Y}/2}, \tag{2-16}$$

$$R_{\mathbf{Z}}(\theta) = e^{-i\theta\mathbf{Z}/2}. \tag{2-17}$$

where the rotations are parametrized by an angle, which is useful when searching for an arbitrary operation. Specifically, an arbitrary 2 × 2 unitary matrix may be decomposed as

$$U = e^{i\alpha} R_{\mathbf{Z}}(\beta) R_{\mathbf{Y}}(\gamma) R_{\mathbf{Z}}(\delta) \tag{2-18}$$

$$= e^{i\alpha} \begin{pmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{pmatrix} \begin{pmatrix} cos\frac{\gamma}{2} & -sin\frac{\gamma}{2} \\ sin\frac{\gamma}{2} & cos\frac{\gamma}{2} \end{pmatrix} \begin{pmatrix} e^{-i\delta/2} & 0 \\ 0 & e^{i\delta/2} \end{pmatrix} \tag{2-19}$$

and thus it is possible to make an arbitrary gate by knowing the optimal values of the coefficients α, β, γ and δ.

Another useful concept while representing Hamiltonians of fermionic systems is the *Pauli string*. A Pauli string is a tensor product of N Pauli matrices $\hat{P}_a \in \{I, X, Y, Z\}^{\otimes N}$. These kinds of strings form observables suitable for measurement on a quantum device, as each qubit is affected by a simple gate.

Logical gates on a system of multiple qubits offer the possibility to act on a qubit regarding the value of the other. The usual gate to show this is the *controlled*-NOT or CNOT gate. This gate has two input qubits (*control* and *target* qubit, respectively). If the control qubit is in the state $|0\rangle$, the target qubit is unchanged. However, if the control qubit is $|1\rangle$, the target qubit is flipped. The CNOT gate is represented by the following matrix

$$U_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{2-20}$$

Single qubit gates and controlled-NOT gates are prototypes for all other gates in the way they are *universal*: Any multiple logic gate may be composed from CNOT and single qubit gates (see [53], section 4.5). And even more interesting, any hermitian matrix (of dimension equal to a power of 2) can be described as a linear combination of products of Pauli matrices, where the coefficients are always real.

2.3. Circuit Representation

To dive into the formalism of quantum computing, it is necessary to understand the ubiquitous circuit representation, which is an abstract representation of the collection of quantum gates interconnected by quantum wires. Each wire corresponds to the state of a qubit in a specific stage of the circuit. The actual structure of the quantum circuit and the interconnection inside is fully dictated by the unitary transformation carried out by the circuit. This is how, for different problems, different sequences of gates are applied to specific wires. The selection of ordered gates over wires is called an *ansatz*. In the figure **2-2** there is an example on how the circuits and gates can be easily visualized in a diagram, commonly used while discussing quantum computing.

Gates displayed in figure **2-2** can be seen as the building block of more complex, larger circuits with different capabilities and uses. While working with quantum circuits in the NISQ era, it is desired to use the simplest configurations to reduce the impact of the State preparation and measurement (SPAM) errors. However, the circuits should also be capable of expressing the searched quantum state (or even a good approximation of it) to perform the desired calculation. Otherwise, it would be like looking for something that is not even in the searching area. In [69], Sim discusses this feature in great detail.

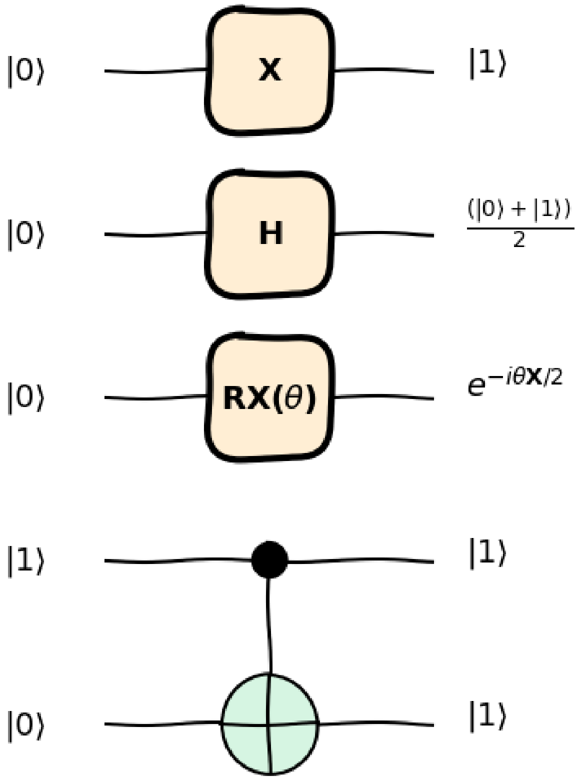


Fig. 2-2: An example of gates acting over wires. The first three are 1-qubit gates. Over the last two wires acts a 2-qubit gate, a CNOT. The black dot represents that the qubit over that wire is the control qubit. At the other end of the gate, where the cross is located, is the wire with the target qubit. Schematic figure made using Pennylane

3 Molecular Hamiltonians for Quantum Computing

The goal of computational quantum chemistry is to unveil the quantum effects that determine structure and properties of molecules. Questions that are commonly investigated through computational methods are *Molecular geometries*, *chemical reactivity*, *electronic states* and even *spectroscopy*.

Analysis of molecular systems using “first principles” are based on the Schödinger equation, which kickstarted the development of the quantum theory at the beginning of the XX century (in fact, work on this field started shortly after the presentation of Schödinger famous theory, with the introduction *self-consistent field method* proposed by D. R. Hartree). Solving the Schödinger equation (eq. **??**) for a molecule gives the molecule's energy and wavefunction, used to calculate the electron distribution, and from that, everything to be known about the molecule.

$$H|\Psi\rangle = E|\Psi\rangle. \tag{3-1}$$

The Schröedinger equation gives impressive results for atoms with just one electron, but that's it. Analysing atoms or small molecules (not mentioning the long and complex molecular systems such as molecular chains or macromolecules) is a task out of the reach for this powerful equation. Thus approximations are required.

3.1. The Electronic Problem

The main interest of Quantum Chemistry is (typically) find approximate solutions of the non-relativistic time-independent Schrödinger equation

$$H(\vec{R},\vec{P},\vec{r},\vec{p})|\Psi(\vec{R},\vec{P},\vec{r},\vec{p})\rangle = E(\vec{R},\vec{P})|\Psi(\vec{R},\vec{P},\vec{r},\vec{p})\rangle,$$

where H is the Hamiltonian operator for a system of nucleus and electrons, $|\Psi\rangle$ is the quantum state of the complete system, E is the allowed energy of the system; the parameters $\vec{r}$, $\vec{p} = \partial/\partial\vec{r}$ are the electronic collective coordinates and momentum, and $\vec{R}$, $\vec{P} = \partial/\partial\vec{R}$ are the nuclear collective coordinates and momentum. The equation **??** shows how the quantum system is described by the position and momenta of every electron and nucleus that conform the molecule, but the energy is just a scalar value determined by the collective coordinates of the nuclei alone. In other words, detailed information of the electrons is not required to calculate the allowed energies of the molecular system.

In atomic units, the Hamiltonian from N electrons and M nuclei is

$$H = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_{i,A} \frac{Z_A}{r_{iA}} + \sum_{i>j} \frac{1}{r_{ij}} + \sum_{B>A} \frac{Z_A Z_B}{R_{AB}} - \sum_A \frac{1}{2M_A}\nabla_A^2. \tag{3-2}$$

where the distance between the ith electron and Ath nucleus is $r_{iA} = \left|\vec{r_i} - \vec{R_A}\right|$; the distance between the ith and jth electron is $r_{ij} = |\vec{r_i} - \vec{r_j}|$, and the distance between the Ath nucleus and the B nucleus is $R_{AB} = \left|\vec{R_A} - \vec{R_B}\right|$. The term M_A is the ratio of the mass of the nucleus A to the mass of an electron, and Z_A is the atomic number of the nucleus A. The equation 3-2 represents the kinetic energy of the electrons and the nuclei, along with the Coulomb repulsion between each pair of electrons and nuclei, and the attraction between electrons and nuclei.

The Hamiltonian expressed above is written using atomic units. In SI units, the Schrödinger equation with the kinetic and potential energy of a simple system $|\phi\rangle$ reads

$$\left[-\frac{\hbar^2}{2m_e}\nabla^2 - \frac{e^2}{4\pi\epsilon_0 r}\right]|\phi\rangle = E\,|\phi\rangle\,, \tag{3-3}$$

where $\hbar$ is Plank's constant divided by 2π, m_e is the mass of the electron and $-e$ is the charge of the electron. In order to write this equation into dimensionless form a substitution $x,\ y,\ z \to \lambda x',\ \lambda y',\ \lambda z'$ is applied, and the result is

$$\left[-\frac{\hbar^2}{2m_e\lambda^2}\nabla'^2 - \frac{e^2}{4\pi\epsilon_0\lambda r'}\right]|\phi'\rangle = E\,|\phi'\rangle\,. \tag{3-4}$$

The constants in front of the kinetic and potential energy operators can then be factored choosing a suitable value for λ,

$$\frac{\hbar^2}{m_e\lambda^2} = \frac{e^2}{4\pi\epsilon_0\lambda} = E_a, \tag{3-5}$$

where E_a is the atomic unit of energy called the *Hartree* (Ha). Solving Eq. 3-5 for λ

$$\lambda = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} = a_0 \tag{3-6}$$

.

At the end, λ is just the Bohr radius a_0 and the Schrödinger equation in atomic units takes the form

$$E_a\left[-\frac{1}{2}\nabla'^2 - \frac{1}{r'}\right]|\phi'\rangle = E\,|\phi'\rangle\,,$$

$$\left(-\frac{1}{2}\nabla'^2 - \frac{1}{r'}\right)|\phi'\rangle = E'\,|\phi'\rangle\,,$$

where $E' = E/E_a$. The solution of this equation for the ground state of the Hydrogen atom is $E' = -0,5$ atomic units $\equiv -0,5$ Hartrees.

3.2. Variational Principle

Fortunately, the impossibility to get exact solutions for complex systems didn't stop researchers in their attempt to understand them. Several approximation methods can be used to solve the Schröedinger equation to almost any desired accuracy. Two of the most widely used approximation procedures are the *Variational method* [75] and the *Perturbation Theory* [20, 42]. When there is a good unperturbed Hamiltonian that can be approximated to the problem at hand, perturbation theory can be more efficient than the variational method. As the Variational Method is a key component of the variational quantum algorithms used in this work, we will discuss the variational principle in some detail.

As the ground state wavefunction Ψ_0 and energy E_0 satisfy the Schrödinger equation

$$H\left|\Psi_0\right\rangle = E_0\left|\Psi_0\right\rangle, \tag{3-7}$$

it is possible to calculate an upper bound to E_0 by using a trial function ϕ. This trial function ϕ may depend on some arbitrary parameters α, β, γ, ..., called *variational parameters*. The energy also depends on these variational parameters, and then

$$E_\phi(\alpha, \beta, \gamma, ...) \geq E_0. \tag{3-8}$$

Having selected a trial function suited for the system to be analysed and considering the compromise between accuracy and efficiency, it is possible to minimize E_ϕ with respect to each of the variational parameters and thus approach the exact ground state energy E_0.

Consider that for a Hamiltonian H, there exists a set of orthonormal eigenfunctions $\{\Psi_n\}$ such that $H\Psi_n = E_n\Psi_n$, and the function of the ground-state Ψ_0 returns the lowest energy E_0. Then the trial function

$$\phi = \sum_n c_n\Psi_n, \tag{3-9}$$

where c_n are constants defined as

$$c_n = \int \Psi_n^*\phi d\tau, \tag{3-10}$$

and the energy of the state ϕ is

$$\begin{aligned} E_\phi =& \frac{\int \phi^* H\phi d\tau}{\int \phi^*\phi d\tau} \\ =& \frac{\int \sum_{n,m} c_n^* c_m \Psi_n^* H \Psi_m d\tau}{\int \sum_{n,m} c_n^* c_m \Psi_n^* \Psi_m d\tau} \\ =& \frac{\sum_n c_n^* c_n E_n}{\sum_n c_n^* c_n}. \end{aligned}$$

Finally, subtracting E_0 from the left side of the above equation and $E_0 \sum_n c_n^* c_n / \sum_n c_n^* c_n$ from the right hand side

$$E_\phi - E_0 = \frac{\sum_n c_n^* c_n (E_n - E_0)}{\sum_n c_n^* c_n} \tag{3-11}$$

which is a value equal or greater than zero, as for all E_n is held that $E_n > E_0$ except when $n = 0$, and clearly all the modulus squared $c_n^* c_n > 0$.

3.3. The Born-Oppenheimer Approximation

The Born-Oppenheimer approximation is central to quantum chemistry, allowing to get results that can be verified by experience and so, bypassing the problems set by the complexity of the system. In order to do this, the following assumptions are made:

- The nuclear motion is so much slower than the electronic motion, so that the nuclei can be considered to be fixed,
- The electronic motion is then moving in a field of fixed nuclei,
- The nuclear motion (e.g. vibrations, rotations) feels an effective potential set from the behaviour of the electrons.

These assumptions lead to a separation of the electronic motion and the nuclear motion. This is very powerful, as the resulting wavefunction is parametrized by the nuclear positions but does not take into account their velocities. Within this approximation, the kinetic energy of the nuclei can be neglected inside the Hamiltonian, and the repulsion between the nuclei can be considered to be constant.The remaining terms of the expression form what its called the *electronic Hamiltonian*, that describes the motion of N electrons in the field of M point charges

$$H_{elec}(\vec{R}, \vec{r}, \vec{p}) = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_{i,A} \frac{Z_A}{r_{iA}} + \sum_{i>j} \frac{1}{r_{ij}}, \tag{3-12}$$

which is solved by an electronic wavefunction $\Psi_{elec}(\vec{R}, \vec{r}, \vec{p})$ with an eigenvalue $E_{elec} = E_{elec}(\vec{R})$,

$$H_{elec}\Psi_{elec} = E_{elec}\Psi_{elec}. \tag{3-13}$$

The electronic wavefunction Ψ_{elec} *explicitly* depends on the electronic coordinates but depends *parametrically* on the nuclear coordinates, as does the electronic energy E_{elec}. The collection of all the possible allowed energies resulting from the possible nuclear configurations $\vec{R}$ defines a *potential energy surface* $V(\vec{R})$. The corresponding approximation to the total wavefunction of 3-2 is a product between the electronic part and the nuclear part

$$\Psi(\vec{r}, \vec{R}) = \Psi_{elec}(\vec{r_i}\vec{R_A})\Psi_{nucl}(\{\vec{R_A}\}). \tag{3-14}$$

Among other applications, the energy analysis of a chemical system can be successfully used to find the equilibrium geometry of a molecule, searching the lowest-lying state of a molecule in specific sectors of the Hilbert space, and even explore chemical reactions through the analysis of the potential energy surface (PES) of the molecule. Potential energy surfaces describe the energy for different positions of its atoms. The usefulness of the concept originates from the Bohr-Oppenheimer approximation. The fact that the electrons, being much lighter than protons and neutrons, adjust their location almost instantaneously based on the new positions of the nuclei. Then, the energy of the molecule $E(\vec{R})$ is a function of the nuclear coordinates $\vec{R}$. The potential energy surface is precisely this function $E(\vec{R})$, relating energies to different geometries of the molecule.

3.4. Hartree-Fock Approach

Based on the very powerful approximations made, it is possible to approach the electronic problem in different ways. It is safe to say that the Hartree-Fock (HF) method is the simplest, but at the same time the most used while considering electronic systems with multiple particles. The key of the Hartree-Fock method is considering the electron wave function in terms of single electron wave functions. The choosing of this trial function is of great importance, as it needs to be sufficiently accurate to give good energy results but not too complex so that the calculations take excessively long. The trade-off between accuracy and efficiency in quantum chemistry is an issue that becomes more critical to take into account as larger molecules are analysed.

3.4.1. The Hartree-Fock Equation

The Variational principle can be used to define the set of functions used to represent each orbital of the molecule. These optimal functions are given by the solution of the Hartree-Fock equations, a set of equations derived by Vladimir Fock and John Slater in 1930. For an atom or molecule with an even $2n$ number of electrons paired off in n orbitals, the HF equations are

$$f(\vec{r})\chi_i(\vec{r}) = \epsilon_i \chi_i(\vec{r}) \quad \text{for } i = 1, 2, ..., n, ... \tag{3-15}$$

where $\{\chi_k\}$ denote the eigenfuntions of the Fock operator $f(\vec{r})$. The corresponding eigenvalues are called *orbital energies*.

Defining the set of functions considered as a solution for these coupled one-electron equations is the first thing to do. As a starting point consider the electronic structure of atoms with a single electron. The electron can be found in a region of space around the nucleus, and the shape of that region is defined by the energy and angular momentum of that electron. That region of space is called an *spatial orbital*. The spatial orbitals $\phi_i(\vec{r})$ describe the location and wave-like behaviour of an electron in an atom. Specifically, the probability of finding an electron in the small volume element $d\vec{r}$ surrounding $\vec{r}$ is $|\phi_i(\vec{r})|^2$.

Now, as the problem at hand is the electronic structure of a molecular system, the wavefunction of the system would be composed of *molecular orbitals*. Clearly, atomic orbitals are not enough to describe an electron around multiple nuclei and interacting with other electrons. Having this in mind, one can define a trial molecular orbital $\phi_i(\vec{r})$ and optimize it in multiple ways, giving as a result a solution of the Schröedinger equation with a good enough approximation of the true energy of the system. In that line, a trial molecular orbitals can be defined as a linear combination of atomic orbitals (LCAO), where coefficients of the linear combination and parameters of the orbital can then be optimized to fit the molecular system. If the set of spatial orbitals $\{\phi(\vec{r}\}\}$ were complete, then any arbitrary function $g(\vec{r})$ could be expanded as

$$g(\vec{r}) = \sum_{i=1}^{\infty} c_i \phi_i(\vec{r}), \tag{3-16}$$

where the c_i are constant coefficients. In general, the set of orbitals would need to be infinite to be complete. However, in practice such a set is not necessary as a finite set $\{\phi_i | i = 1, 2, ..., K\}$ of K orbitals might be enough to describe an "exact" result within its subspace. In this subspace, a molecular orbital is expressed as

$$\psi_i(\vec{r}) = \sum_{i=1}^{K} c_\nu^i \phi_\nu(\vec{r}, \alpha). \tag{3-17}$$

To completely describe an electron, it is necessary to introduce its spin. As a half-integer spin particle 1/2, this last feature of the electron can be described by two orthonormal functions (α, β). The wave function that describes both its spatial distribution and its spin is a *spin orbital*, $\chi(\vec{x})$ where $\vec{x}$ indicate both space and spin coordinates. From each spatial orbital, $\psi_i(\vec{r})$, one can form two different spin orbitals

$$\chi_i(\mathbf{x}) = \begin{cases} \psi_i(\vec{r})\alpha \\ \text{or} \\ \psi_i(\vec{r})\beta \end{cases}$$

and now the set used to expand molecular orbitals contains $2n$ spin orbitals.

Having selected a basis set to describe the molecular orbitals, the matrix form of the Hartree-Fock equations is obtained, an then the problem reduces to process of eigenvalues calculation. The equation 3-15 is rewritten as

$$F \sum_{i=1}^{K} c_\nu^i \phi_\nu(\vec{r}) = \epsilon_i \sum_{i=1}^{K} c_\nu^i \phi_\nu(\vec{r}),$$

where values c_ν^i are the expansion coefficients of the ith molecular orbital of the matrix $\mathbf{C}$. To extract the equation for a specific basis function, integrate in the left-hand side by the complex conjugate function $\phi_\mu(\vec{r})^*$

$$\sum_{i=1}^{K} c_\mu^i \int d\vec{r}\phi_\mu(\vec{r})^* f(\vec{r})\phi_\nu(\vec{r}) = \epsilon_i \sum_{i=1}^{K} c_\nu^i \int d\vec{r}\phi_\mu(\vec{r})^* \phi_\nu(\vec{r})$$
$$\sum_{i=1}^{K} f_\mu^i c_\mu^i = \epsilon_i \sum_{i=1}^{K} c_\mu^i s_\mu^i,$$

where s_μ^i coefficients form an *overlaping* matrix of basis functions, and f_μ^i describe the elements of the Fock matrix. Finally, the set of coupled equations takes the form

$$\mathbf{FC} = \mathbf{SC}\epsilon, \tag{3-18}$$

which are commonly known as the Hall-Roothaan equations. The Hartree-Fock method is also called *self-consistent field* method, as the mean field affecting each individual electron depends on the molecular orbitals (specifically, their expansion coefficients), which depends on the same coefficients. So, the solutions for the set of equations is to be to be found in an iterative manner.

3.4.2. The Antisymmetry Principle and Slater Determinants

Having the molecular orbitals of the systems, then the complete electronic wavefuntion of N electrons can be approximated as a product of individual orbitals, in what is commonly known as *Hartree Product*

$$\Psi^{HP}(\vec{x_1}, \vec{x_2}, ..., \vec{x_N}) = \chi_i(\vec{x_1})\chi_j(\vec{x_2})...\chi_k(\vec{x_N}). \tag{3-19}$$

The wavefunction described by a Hartree Product is an independent-electron wavefunction since the join probability of finding each electron in a given volume of space is equal to the product of probabilities for each electron

$$|\Psi^{HP}(\vec{x_1}, ..., \vec{x_N})|^2 d\vec{x_1} \cdots d\vec{x_N} = |\chi_i(\vec{x_1})|^2 d\vec{x_1}\, |\chi_j(\vec{x_2})|^2 d\vec{x_2} \cdots |\chi_k(\vec{x_N})|^2 d\vec{x_N}. \tag{3-20}$$

There is still another deficiency in the Hartree product: it does not takes account of the indistinguishability of electrons, e.g. specifically identifies electron-one as occupying spin orbital χ_i, electron-two as occupying χ_j, an so on. The *antisymmetry principle* requires that the system does not distinguish between identical electrons and requires that the electronic wavefunction be antisymmetric.

A satisfactory expression can be found by writing it as a determinant. For any N electron system, the Slater determinant is defined as

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \cdots & \chi_N(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \cdots & \chi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{x}_N) & \chi_2(\mathbf{x}_N) & \cdots & \chi_N(\mathbf{x}_N) \end{vmatrix}$$
$$\equiv |\chi_1, \chi_2, \cdots, \chi_N\rangle,$$

where the last expression is a shorthand for Slater determinants (and a ket in Dirac notation, which is not a coincidence). The determinant ensures that two particles cannot be in the same *place* with the same *quantum state*.

3.4.3. Slater and Gaussian Type Orbitals

The selection of a basis set is key in the efficiency and accuracy of the results for a chemical calculation. The common approach used is to approximate the orbitals of each atom using a function from the basis set, and then a molecular orbital is created as a linear combination of atomic orbitals. In order to do this, a function is designed to describe an electron for each subshell of the atom ($1s$, $2p_x$, $2p_y$, $2p_z$ and so on). There are two generally basis functions used in electronic structure calculations: *Slater Type Orbitals* (STO) and *Gaussian Type Orbitals* (GTO). Other basis sets can be created by a linear combination of the mentioned functions, resulting in *contracted basis functions* that are less flexible but reduce the computational cost significantly.

Slater type orbital functions are solutions to the Schrödinger equation of hydrogen-like atoms (atom or ion with a single valence electron) and decay exponentially far away from the nucleus. The radial part of STOs is described as

$$R(r) = Nr^{n-1}e^{-\zeta r}$$

where n is the principal quantum number, N is a normalizing constant, r is the distance of the electron from the atomic nucleus and ζ is a constant related to the effective charge of the nucleus. As STOs do not have any radial nodes, these are introduced by making linear combinations of STOs functions. And by finely tuning the value of ζ for each basis function used, the behaviour of the molecular orbital function regarding the effective field can be defined.

Calculating integrals with STOs is computationally difficult, so they are commonly replaced with the more calculation-friendly Gaussian Type Orbitals. The radial part of GTOs take the form

$$R(r) = Ne^{-\zeta r^2},$$

where, again, N is a normalizing constant and ζ is related to the effective charge. The main issues with GTOs are that they don't represent accurately the extremes of the atom (the nucleus and the "tail" of the wavefunction) as the radial r^2 dependence of the GTOs falls off to rapidly and has zero slope at the nucleus. The loss of accuracy caused by using Gaussian functions is faced by using fixed linear combinations of Gaussians with parameters predetermined by atomic calculations or some other prescription. Besides, the relatively easy calculations required using the well known Gaussian functions makes them really easy to work with.

A *minimal basis set* contains one basis function for each type of atomic orbital that is occupied in the ground state of each atom in the molecule. For example, methane (CH_4) uses $1 \times 5 + 4 \times 1 = 9$ basis functions, in which each electron of the system is described by a different function linked to an atomic orbital. In fact, it's possible to construct a basis set for any element in the second period of the periodic table using five functions with different radial exponents to represent the $1s$, $2s$, $2p_x$, $2p_y$, $2p_z$ orbitals of the electronic configuration of these elements. A well-known minimal contracted basis function set is the STO-nG in which each atomic orbital is described by n GTOs. Although minimal basis sets are easy to use, they are not accurate enough for most purposes. Instead, further basis sets can be built using the minimal sets as building blocks by, for example, augmenting the number of basis sets using the N-Zeta strategy (zeta referring to the variable exponent represented by ζ). There is an extensive basis set library online called Basis Set Exchange [62] with a vast amount of available basis sets for atomic orbitals of each element. In figure **3-1**, it is possible to see how energy calculations using different basis sets vary in accuracy.

3.5. Occupation Basis and Fermionic Description of the Molecule

The use of molecular orbitals to describe the electronic system of the molecules comes handy in multiple ways. In the last section, a rudimentary wavefunction Ψ^{HP} was created using a product of molecular orbitals, which was then improved using a Slater determinant to take into account the antisymmetry principle, and from that correction the exchange correlation arises. Using a basis set of K spatial orbitals $\{\phi_i\}$ leads to a set of $2K$ spin orbitals $\{\chi_a\}$. Thus, for a system with N electrons, there should be N occupied spin orbitals $\{\chi_a\}$ and a complementary set of $2K - N$ unoccupied or virtual spin orbitals. The Hartree-Fock ground state Ψ_0 has the N lowest energy spin orbitals occupied, which is just one of the many different states that can be formed from the $2K > N$ spin orbitals. Every other possible state consist of one or multiple filled exited orbitals, with the total number of possibilities being the number of combinations of $2K$ objects taken N at the time

$$\binom{2K}{N} = \frac{2K!}{N!(2K-N)!}, \tag{3-21}$$

and this number also happens to be the total number of Slater determinants that can be built with N particles distributed among $2K$ spin orbitals. The dimensionality of the problems

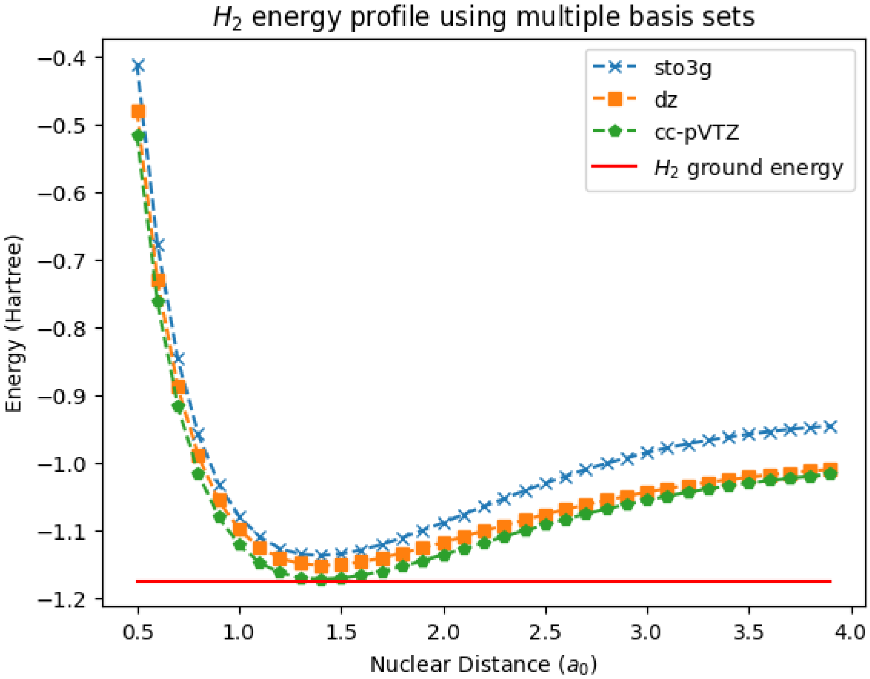


Fig. 3-1: H_2 energy profile using multiple basis sets and Full Configuration Interaction. Larger basis sets give better results at the expense of more computation load. The basis used point to the correct internuclear distance of $1{,}4a_0$, and the larger triple Zeta basis "cc-pVTZ" shows a minimal energy of $-1{,}172$ Hartrees. This values is just above the experimental value of $-1{,}174$ [33].

explodes when more electrons and their corresponding spatial orbitals are considered. The *Full Configuration Interaction* (FCI) method considers a trial wavefunction of the system as a linear combination of *all* the possible Slater determinants. This approach usually provides numerically exact solutions, at the cost of being computational expensive. As the molecular systems in this work are rather small, FCI are used to benchmark the obtained results.

Taking aside the intimidating growing pace of the problem, trial functions produced by a Slater determinant are sophisticated representations of distributed particles and holes. From this perspective, it is straightforward to look for a basis that represents occupied or empty states. This is where the occupation basis appears, describing a fermionic system as n single-particle states (spin orbitals) that can be either filled or empty

$$|f_1 \cdots f_N\rangle, \text{ where } f_j \in \{0, 1\}. \tag{3-22}$$

A system of N fermionic modes, as well as any interaction within, can be expressed using a set of fermionic *creation* and *annihilation* operators. For each fermionic annihilation operator $a_i | i = 1, 2, ..., N$, its adjoint $a_i^\dagger$ is the creation operator. These fermionic operators satisfy the canonical anticommutation relations $\{a_i, a_j\} = 0$, $\{a_i^\dagger, a_j^\dagger\} = 0$, and $\{a_i, a_j^\dagger\} = \delta_{ij}$ where

$\{A, B\} := AB + BA$. The anticommutation relations listed before ensure that the states in the occupation basis satisfy the antisymmetric principle.

The fermionic operators $a_i^\dagger$, a_i commute with each other, and have eigenvalues 0 and 1, when using the occupation basis. From this frame nataurally arises a normalized vector $|0\rangle$ called the *vacuum state*. This state has eigenvalue 0 for the annihilation operator,

$$a_i|0\rangle = 0 = \langle 0|a_i^\dagger, \tag{3-23}$$

and, when operated by the creation operator $a_i^\dagger$, fills the state i

$$a_i^\dagger|0\rangle = |f_i\rangle, \quad a_i|f_i\rangle = |0\rangle. \tag{3-24}$$

Keeping in mind the antisymmetric state and the Pauli exclusion principle, one cannot create or annihilate a fermion in the same mode twice

$$a_i^2 = 0 = a_i^{\dagger\,2}. \tag{3-25}$$

The set of 2^n vectors considering all the possible combinations of filled or empty single-particle states

$$|f_0 f_1 \cdots f_n\rangle := \left(a_0^\dagger\right)^{f_0} \cdots \left(a_0^\dagger\right)^{f_0} |0\rangle\,, \quad f_i \in \{0,1\}, \tag{3-26}$$

are orthonormal. We can assume they form a basis for the entire vector space.

Finally, the action on an arbitrary state of the creation or annihilation takes the form

$$\begin{aligned}
&a_j^\dagger|f_0 \cdots f_{j-1}\,0\,f_{j+1} \cdots f_n\rangle = (-1)^{\sum_{s=0}^{j-1} f_s}|f_0 \cdots f_{j-1}\,1\,f_{j+1} \cdots f_n\rangle \\
&a_j^\dagger|f_0 \cdots f_{j-1}\,1\,f_{j+1} f_n\rangle = 0 \\
&a_j|f_0 \cdots f_{j-1}\,1\,f_{j+1} f_n\rangle = (-1)^{\sum_{s=0}^{j-1} f_s}|f_0 \cdots f_{j-1}\,0\,f_{j+1} \cdots f_n\rangle \\
&a_j|f_0 \cdots f_{j-1}\,0\,f_{j+1} \cdots f_n\rangle = 0
\end{aligned}$$

The factor $(-1)^{\sum_{s=0}^{j-1} f_s}$ maintains the antisymmetry of the system. The sum accounts for how many swaps need the $a_j^\dagger$ or a_j operator to act against the vacuum state. Here is an example of an annihilation operator acting on a predefined state

$$\begin{aligned}
a_1|f_1 0 f_3 f_4\rangle =& a_1 a_4^\dagger a_3^\dagger a_1^\dagger|0\rangle = (-1)^1 a_4^\dagger a_1 a_3^\dagger a_1^\dagger|0\rangle = (-1)^2 a_4^\dagger a_3^\dagger a_1 a_1^\dagger|0\rangle \\
=& a_4^\dagger a_3^\dagger(1 - a_1^\dagger a_1)|0\rangle = a_4^\dagger a_3^\dagger|0\rangle \\
=& |00 f_3 f_4\rangle
\end{aligned}$$

3.6. Electronic Hamiltonian using Fermionic Operators

In the previous section it has been shown that the occupation basis is a powerful and flexible tool that allows us to work with molecular systems. Also, fermionic operators $a_i^\dagger$, a_i

were introduced as a way to build and manipulate such states. The next step to develop an entire workframe of many-electron systems is to express the many particle operators that compose the Hamiltonian in creation and annihilation operators.

To write the electronic Hamiltonian in terms of fermionic operators, first take into account that all the terms in the equation describe:

- kinetic and potential energy of each electron in the field of the nuclei (*one electron* terms), or
- total Coulomb repulsion between each pair of electrons (*two electron* terms)

which allows to separate the electronic Hamiltonian into one-electron and two-electron operators

$$\begin{aligned} H_{elec} &= -\sum_i \frac{1}{2}\nabla_i^2 - \sum_{i,A}\frac{Z_A}{r_{iA}} + \sum_{i>j}\frac{1}{r_{ij}} \\ &= \mathcal{O}(1) + \mathcal{O}(2), \end{aligned}$$

$$\mathcal{O}(1) = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_{i,A}\frac{Z_A}{r_{iA}} = \sum_n h(\mathbf{r}_n) \tag{3-27}$$

$$\mathcal{O}(2) = \sum_{i>j}\frac{1}{r_{ij}}. \tag{3-28}$$

It is interesting to notice that the part $\mathcal{O}(1)$ is just the sum of the Hamiltonian $h(\mathbf{r}_n)$ that obtains the total energy of each electron in the field of all the nuclei.

To express the one electron operator, A, consider the existence of orthonormal spin orbitals $\{\chi_i\}$, $i \in \{1, ..., 2K\}$. Each orbital has a linked pair of creation and annihilation operators $\{a_i^\dagger, a_i\}$ that fill or clear out the electron in said orbitals. If the operator A acts over the state $|f_i\rangle$

$$A\,|f_i\rangle = A a_i^\dagger\,|0\rangle \tag{3-29}$$

$$= b_i^\dagger\,|0\rangle = \left(\sum_k^{2K} c_{ik} a_k^\dagger\right)|0\rangle \tag{3-30}$$

where the last line shows how the effect of the operator A is a linear combination of the well known fermionic operators previously introduced. To determine the expansion coefficients c_{ik} one just needs to operate on the left side by another state $\langle f_j|$

$$\begin{aligned} \langle f_j|\,A\,|f_i\rangle &= \langle 0|\,a_j\left(\sum_k^{2K} c_{ik} a_k^\dagger\right)|0\rangle = \langle 0|\left(\sum_k^{2K} c_{ik} a_j a_k^\dagger\right)|0\rangle \\ &= \langle 0|\left(\sum_k^{2K} c_{ik}\left[\delta_{jk} - a_j^\dagger a_k\right]\right)|0\rangle = c_{ij}\,\langle 0|0\rangle = c_{ij}, \end{aligned}$$

and the anticommutation relation of the operators is used in the last line, along with the fact that any annihilation operator acting on the vacuum state $|0\rangle$ returns 0. Combining equations 3-29 and **??** leads to

$$Aa_i^\dagger = \sum_k^{2K} c_{ik} a_k^\dagger, \tag{3-31}$$

which are operators that can only act on the vacuum state. To change this, one just needs to add an annihilation operator to the left

$$Aa_i^\dagger a_i = \sum_k^{2K} c_{ik} a_k^\dagger a_i. \tag{3-32}$$

Finally, summing over all one electron states (all orbitals) one gets

$$A \sum_i^{2K} a_i^\dagger a_i = \sum_{i,k}^{2K} c_{ik} a_k^\dagger a_i \tag{3-33}$$

where on the left-hand side the operator $a_i^\dagger a_i$ returns 1 for all i as the system is composed of electrons and there can be just one electron in every spin orbital. Finally, the one electron operator can be expressed as

$$A = \sum_{i,k}^{2K} c_{ik} a_k^\dagger a_i \tag{3-34}$$

$$c_{ik} = \langle \chi_i | A | \chi_k \rangle . \tag{3-35}$$

The expression of two electron operators follows the same line.

After all, the electronic Hamiltonian expressed in fermionic operators takes the form

$$H_{elec} = \sum_{i,j} \langle i|h|j\rangle a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} \langle ij|kl\rangle a_i^\dagger a_j^\dagger a_l a_k, \tag{3-36}$$

where the sums run over the set of used spin orbitals $\{\chi(\mathbf{x})\}$.

$$h_{ij} = \langle i|h|j\rangle = \int d\mathbf{x}_1 \, \chi_i^\dagger(\mathbf{x}_1) h(\mathbf{r}_1) \chi_j(\bar{x}_1)$$

$$h_{ijkl} = \langle ij|kl\rangle = \int d\mathbf{x}_1 \, d\mathbf{x}_2 \, \chi_i^\dagger(\mathbf{x}_1) \chi_j^\dagger(\mathbf{x}_2) \, r_{12}^{-1} \, \phi_k(\mathbf{x}_1) \chi_l(\mathbf{x}_2).$$

Notice how the factor 1/2 appears in front of the two electron parts, as the sum in the last equation considers each pair of interacting electrons (spin orbitals) twice. The molecular orbitals then have to be written in the molecular orbital basis as

$$h_{ij} = \sum_{\mu\nu} \mathbf{C}_{i\mu} h_{\mu\nu} \mathbf{C}_{\nu j}$$

$$h_{ijkl} = \sum_{\mu\nu\rho\sigma} \mathbf{C}_{i\mu} \mathbf{C}_{j\nu} h_{\mu\nu\rho\sigma} \mathbf{C}_{\rho k} \mathbf{C}_{\sigma l}$$

where **C** is the molecular orbital expansion coefficient matrix.

This fermionic Hamiltonian has several advantages that can be exploited while working on computational chemistry. The domain of the operator is given by the domain of creation and annihilation operators acting on the occupation basis, which offers a very pictorial representation as the filled or void orbitals invite to use a binary basis. Also, remember that this fermionic representation comes from the introduction of the finite basis of one electron functions $\{\chi_i\}$, $i \in \{1, ..., 2K\}$ which is very important from the point of view of practical quantum chemistry, and again, the equilibrium between accuracy and efficiency of the calculations. As this finite set cannot express exactly the wavefunctions resulting from the action of the original Hamiltonian, this (reduced) Hamiltonian can be interpreted as the projection of the exact one to a finite subspace spanned by the basis molecular orbitals. One final thing is that, in this representation, the Hamiltonian is clearly independent by the number of electrons on the systems. However, it explicitly depends on the list of one and two electron integrals, and thus on the size of the basis set of used orbitals.

3.7. Mapping Fermions to Qubits with Transformations

To simulate a molecular system on a quantum computer, it is required to choose a representation of the creation and annihilation operators on the space of the qubits. In other words, a set of qubit operators are required which satisfy the anticommutation relations. As qubit operators are written in terms of the Pauli matrices **X**, **Y** and **Z**, there are the building blocks of the qubits operators to be created.

Good enough, the occupation number basis suggest a direct identification with the computational basis used in quantum computing

$$|f_{n-1} \cdots f_0\rangle \;\rightarrow\; |q_{n-1}\rangle \cdots \otimes |q_1\rangle \otimes |q_0\rangle, \quad f_j = q_j \in \{0, 1\}, \tag{3-37}$$

where there are, again, $n = 2 * K$ qubits which span the whole subspace after the transformation (one for each spin orbital). A wave function written in this new basis takes the form

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle (\text{mod } 2) \tag{3-38}$$

3.7.1. The Jordan-Wigner Transform

There are several transformations that can be applied in order to use the fermionic Hamiltonian in a qubit system. The *Jordan-Wigner Transform* is the most basic transformation to be applied, as it uses the well-known ladder operators for angular momentum that can be applied in a qubit system

$$Q^+ = |1\rangle\langle 0| = \frac{1}{2}(\mathbf{X} - \mathrm{i}\mathbf{Y}), \qquad Q^- = |0\rangle\langle 1| = \frac{1}{2}(\mathbf{X} + \mathrm{i}\mathbf{Y}), \tag{3-39}$$

as can be seen from the ket-bra expression in each equation, that disclose the matrix representation of each operator.

With the qubit operators, it is straightforward to fully represent the fermionic operators in terms of basic quantum gates

$$a_j^\dagger \equiv \mathbf{1}^{\otimes n-j-1} \otimes Q^+ \otimes (\mathbf{Z})^{\otimes j} \tag{3-40}$$

$$a_j \equiv \mathbf{1}^{\otimes n-j-1} \otimes Q^- \otimes (\mathbf{Z})^{\otimes j} \tag{3-41}$$

where the $\mathbf{Z}$ operators at the right with eigenvalues $\{+1, -1\}$ keep track of the parity of the system (which is related to the fat that the number of occupied states is even or odd), adding a (-1) phase value whenever a qubit in the way is occupied. The identity operator $\mathbf{1}^{\otimes n-j-1}$ to the right shows that no qubit after the zeroes j is affected in any way.

The introduced operators Q_j^+, Q_j^- obey the fermionic anticommutation relations

$$\{Q_i^+, Q_j^+\} = 0 \tag{3-42}$$

$$\{Q_i^-, Q_j^-\} = 0 \tag{3-43}$$

$$\{Q_i^+, Q_j^-\} = \delta_{ij}\mathbf{1} \tag{3-44}$$

and, as they also anticommute with $\mathbf{Z}$, the qubit representation on equation 3-40 also comply with the fermionic anticommutation relations.

The Jordan-Wigner transformation is associated with a particular mapping of bitstrings $e : \{0,1\}^N \to \{0,1\}^N$ that is the identity map, i.e. $e(x) = x$. This can be easily seen by noting that the fermionic basis vector $|f_0 \cdots f_{N-1}\rangle$ is represented by the computational basis vector $|q_0 \cdots q_{N-1}\rangle$, where $f_p = q_p$ for all p. The mapping can be written as

$$|x\rangle \to |e(x)\rangle \tag{3-45}$$

where the vector on the left is fermionic and the vector on the right is qubit. The mapping e is called an *encoder*.

The operators obtained from the Jordan-Wigner transformation have high weight, which means that the operator acting on the qubit $|p\rangle$ also acts on qubit or state $|0\rangle, ..., |p-1\rangle$. This can be a disadvantage as it requires to operate on a high number of qubits, which implies a high number of gates to simulate attached to a more expensive measurement. In the worst case, an operator required to act on the last mode is mapped to a qubit operator that will act on all modes, being this computationally expensive.

3.7.2. The Parity Transform

The Jordan-Wigner Transform uses the string of Pauli $\mathbf{Z}$'s to introduce the phase factor $(-1)^{\sum_{s=0}^{p-1} q_s}$ when acting on a computational basis state. Calculating this factor requires reading p states and operating a Pauli $\mathbf{Z}$ string of length p. So, is there some improvement to be gained by using this transformation? Consider the *Parity transform* defined by the encoding

$$e(x)_p = \sum_{q=0}^{p} x_q (\text{mod } 2), \tag{3-46}$$

in which the qubit p does not store the occupation of the state p, but the *parity* of orbitals with index less than or equal to p. This encoding has the great advantage of computing the parity sum $\sum_{s=0}^{p-1} q_s$ by reading just one qubit.

However, it is no longer possible to represent the creation or annihilation of a particle in a state p by simply operating with $Q^{\pm}$ on the qubit p. Whether the operator Q^{+} or Q^{-} acts on the qubit p depends on the qubit $p-1$. If qubit $p-1$ is in the state $|0\rangle$, then qubit p will accurately reflect the occupation of state j, and then the operators can be directly translated. However, if the qubit $p-1$ is in the state $|1\rangle$, then qubit p will have inverted parity compared to the occupation of state j, and so the operators need to be inverted (Q^{-} to act as $a^{\dagger}$ and *vice versa*). Finally, the operators equivalent to $Q^{\pm}$ in the parity bases are therefore a two-qubit operator acting on qubits p and $p-1$

$$\mathbf{P}_p^{\pm} \equiv Q^{\pm} \otimes |0\rangle \langle 0|_{p-1} - Q_p^{\mp} \otimes |1\rangle \langle 1|_{p-1} = \frac{1}{2} \left(\mathbf{X}_p \otimes \mathbf{Z}_{p-1} \mp i \mathbf{Y}_p \right). \tag{3-47}$$

This encoding solves the issue of calculating the parity up to a state p, but comes with the burden of updating all the parity data of every state of index greater than p when a creation or annihilation operator acts on p. This is done with a Pauli $\mathbf{X}$ string operating in qubits $j+1, ..., N-1$. This is why the Parity transform does not offer any advantages over the Jordan-Wigner transform. And also comes the downside of losing the nice parallel with the occupation encoding of states and the computational basis of quantum computing. The fermionic operators operating over states encoded in the parity bases take the form

$$a_j^{\dagger} \equiv \frac{1}{2} \left(\mathbf{X}^{\otimes n-j-1} \otimes \mathbf{X}_j \otimes \mathbf{Z}_{j-1} - i \mathbf{X}^{\otimes n-j-1} \otimes \mathbf{Y}_j \right) \tag{3-48}$$

$$a_j \equiv \frac{1}{2} \left(\mathbf{X}^{\otimes n-j-1} \otimes \mathbf{X}_j \otimes \mathbf{Z}_{j-1} + i \mathbf{X}^{\otimes n-j-1} \otimes \mathbf{Y}_j \right). \tag{3-49}$$

As in the Jordan-Wigner transform, the fermionic operators expressed in Pauli matrices also comply with the fermionic anticommutation relations. While checking this, it is helpful to remember that all Pauli operators anticommute if acting on different modes.

3.7.3. The Bravyi-Kitaev Transform

To simulate fermionic operators with qubits two things are needed: the occupation of a given mode, and the parity of the orbitals with index less than the target mode. The precious two approaches store this information, but are not optimal. In the Jordan-Wigner transform the mode is stored locally (i.e. the occupation data can be read just by reading the value of the state in its index) but the parity information is non-local (i.e. multiple states are read to get the parity data).

A clever encoding that is capable of balancing these two requirements is then needed. It was pointed out by Bravyi and Kitaev that the encoder which achieves this is implemented by a classical data structure called a *Binary Indexed Tree* (also known as Fenwick tree). This is a data structure that can efficiently update elements and calculate prefix sums. It can be seen as a balanced transformation between Jordan-Wigner and parity. However, this encoding also loses the relation of the states with the computational basis for quantum computing.

Using this encoding, the ladder operators $Q^{\mp}$ are still being used along with Pauli Operators $\mathbf{X}$ and $\mathbf{Z}$, which again ensures that the creation and annihilation operators for this encoding comply with the anticommutation relations.

3.8. Example: qubit Hamiltonian for H_2

As a summary and example, here are the steps to follow to calculate the qubit Hamiltonian of a molecule.

1. Use the Born-Oppenheimer approximation to reduce the electronic problem.
2. Select a basis set of functions used to express the molecular orbitals of the system.
3. Calculate the one- and two-electron integrals.
4. Write the system in the Hall-Roothaan equations, and use the Hartree-Fock (self-consistent) method to find the molecular orbital coefficients.
5. Write the one-and two-electron integrals in the molecular orbital basis using the molecular orbital coefficients.
6. Express the Hamiltonian in fermionic operators.
7. Transform the Hamiltonian to a qubit basis by using a transformation of choice.

Below is the qubit Hamiltonian of an H_2 molecule. The Hamiltonian is encoded using the Jordan-Wigner transform, and the Basis Set STO-3G was used to describe the $s1$ atomic orbitals employed to create the molecular orbital. The interatomic distance of the system was set in $1,0$ angstroms. More specifically, the Gaussian functions of one atom were centered on the origin, while the functions for the other molecule used the coordinates $(0,0,1)$. The $s1$ atomic orbital of Hydrogen in the STO-3G basis set takes the form

$$\psi_{\mathrm{STO-3G}}(s) = c_1\phi_1 + c_2\phi_2 + c_3\phi_3, \tag{3-50}$$

where

$$\begin{aligned}\phi_1 &= \left(\frac{2\alpha_1}{\pi}\right)^{3/4} e^{-\alpha_1 r^2}\\ \phi_2 &= \left(\frac{2\alpha_2}{\pi}\right)^{3/4} e^{-\alpha_2 r^2}\\ \phi_3 &= \left(\frac{2\alpha_3}{\pi}\right)^{3/4} e^{-\alpha_3 r^2}.\end{aligned}$$

The Hamiltonian below requires 4 qubits (two molecular orbitals with two spins each) and requires 15 terms. Notice that all terms are associated with a Pauli String, with each gate acting on a specific qubit.

$$\begin{aligned}H_{STO-3G} =&(-0{,}34724) * \mathbf{Z2}\\ &+ (-0{,}34724) * \mathbf{Z3}\\ &+ (0{,}208133) * \mathbf{Z0}\\ &+ (0{,}208133) * \mathbf{Z1}\\ &+ (0{,}298178) * \mathbf{I0}\\ &+ (0{,}132902) * \mathbf{Z0} \otimes \mathbf{Z2}\\ &+ (0{,}132902) * \mathbf{Z1} \otimes \mathbf{Z3}\\ &+ (0{,}175463) * \mathbf{Z0} \otimes \mathbf{Z3}\\ &+ (0{,}175463) * \mathbf{Z1} \otimes \mathbf{Z2}\\ &+ (0{,}178609) * \mathbf{Z0} \otimes \mathbf{Z1}\\ &+ (0{,}184709) * \mathbf{Z2} \otimes \mathbf{Z3}\\ &+ (-0{,}04256) * \mathbf{Y0} \otimes \mathbf{Y1} \otimes \mathbf{X2} \otimes \mathbf{X3}\\ &+ (-0{,}04256) * \mathbf{X0} \otimes \mathbf{X1} \otimes \mathbf{Y2} \otimes \mathbf{Y3}\\ &+ (0{,}042560) * \mathbf{Y0} \otimes \mathbf{X1} \otimes \mathbf{X2} \otimes \mathbf{Y3}\\ &+ (0{,}042560) * \mathbf{X0} \otimes \mathbf{Y1} \otimes \mathbf{Y2} \otimes \mathbf{X3}.\end{aligned}$$

4 Variational Quantum Circuits

As the limitations of the current physical devices used to perform quantum computation are being tackled one by one with the effort of a booming worldwide community, there is an equally large community trying to get results from these machines right now. The main used approach consists in using the variational principle to reshape problems of interest, and then, using an hybrid framework of classical and quantum devices where the advantages of both systems are exploited. So, by implementing some subroutines on classical hardware, the requirements of quantum resources are reduced. Then, quantum resources are focussed on the main feature of interest (often, the representation of a quantum system). A *Variational Quantum Circuit* (VQC) [14] is then used to find the best values of an array of parameters acting on a sequence of gates and produce the state of interest.

For quantum chemistry, the *Variational Quantum Eigensolver* (VQE) and several proposed variants have been used for searching the ground state of the electronic Hamiltonians of molecules. In the same sense, other problems have been studied under this framework with promising results.

In this chapter, the hybrid framework of quantum and classical devices is presented. Then, Variational Quantum Circuits are presented as a way to approximate the ground state of a molecular Hamiltonian, being VQE a prototype of the process. Finally, some implementations using several software development kit (SDK).

4.1. Definition of Variational Quantum Circuits

A Variational Quantum Circuit is a collection of quantum gates interconnected by quantum wires that receives an array of parameters as variables. The values of these variables determine the outcome of the circuit. So, by selecting a suitable sequence of gates (commonly Pauli rotation plus entangler gates such as CNOT) and then getting the optimal values of the variational parameters, the product of the circuit can be a good approximation of the state of interest. A VQC consists of three ingredients:

1. Preparation of a fixed initial state ψ_{init} (e.g. the vacuum state, or a quantum state with embedded data x of the system of interest)

2. A quantum circuit $U(\theta)$, parameterized by a set of free parameters θ.

3. Measurement of the output that defines a scalar cost C for a given task. Typically, that measurement is the expected value of an observable B.

Each one of the ingredients required to create a VQA are highly customizable depending on the problem at hand. This versatility translates into a wide variety of possibilities for each element (initial state, circuit, and cost function), usually related to the system of interest. The first step to developing a VQC is to define a cost function C which encodes the solution to the problem. Then a parametrized circuit is proposed along with the initial state. This ansatz is then trained in an hybrid quantum-classical loop to solve the optimization task

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, C(\theta). \tag{4-1}$$

The big appeal of VQC lies in the fact that the quantum behaviour is kept by using a quantum device to prepare the state and measure the observables of interest, while the optimization of the parameters (usually a computing expensive task) is outsourced to classical devices. In addition to that, the sequence of gates used by the circuit can be modelled to tackle a variety of problems, but also can be kept as short as desirable to reduce the issues presented by the hardware in the NISQ era. With this in mind, the ever present equilibrium between accuracy and efficiency of computational problems is managed by the complexity of the circuit and the way it is trained. Taking into account their flexibility, VQCs have been developed for a wide range of applications including quantum chemistry simulation of dynamic quantum systems, solving linear systems of equations, extension of machine learning problems and even optimization.

One of the main challenges while using a VQC is to choose an effective circuit that represents well the space in which the solution state is searched, but at the same time keeping a low circuit depth and number of parameters [69]. It is required that the circuit can create states all over the Hilbert space being researched, ensuring that the objective state is within reach (*Expressibility*). In the same line, it is desirable to use a circuit that can generate highly entangled states so it can efficiently represent the solution space for the task at hand (*Entangling Capability*). These metrics are usually tuned by changing the used rotation gates and its combinations (**RX**, **RY**, **RZ** gates), the two-qubit entangler gates (**CNOT**, **CZ**, or event a controlled rotation such as **CRX**), and the number of circuit layers. So, again, it is necessary to balance the performance of the circuit against the number of parameters and circuit depth. One thing to highlight is that keep adding gates and layers of gates do not always improves the expresability of the circuit. This implies that there is a threshold beyond which the expressibility value does not improve.

4.2. A Word on Quantum Advantage

The main reason quantum computation attracts that much attention is the possibility to achieve practical quantum advantage with universal quantum simulators. As interesting as physical quantum computers are, it's key to find and validate situations in which efficient classical learning and calculation is impossible, while quantum learning can be efficient. In [23], the authors present some details about the perspectives of quantum advantage on quantum simulations. This reference also describes an interesting view on the requirements of practical advantage (section IV). Not only the quantum simulator needs to return reliable solutions for relevant problems beyond the classical reach, but also it is required to be verifiable. With such hard problems, there might be some of them that can be verified using powerful classical devices. However, the harder ones might need more calculations by a quantum device in order to be confirmed.

At this point it would be possible to use a device with quantum advantage to validate the results of another one. However, there is a more practical way of Hamiltonian learning. In this way, it might be possible to learn the Hamiltonian from a quantum device by making some appropriate measurements on the quantum state resulting from the simulation. Then, comparing the results from the simulated and reconstructed state, it might be possible to evaluate the results of said simulation. It is also helpful that the number of terms in the Hamiltonians that describe many-body physics (usually the target of complex simulations) is relatively small.

On a more quantitative side, it is also important to define some base metrics to keep track of the scale of quantum devices. The following concise parameters to evaluate current and future quantum processors are presented in [74]:

- **Scale:** (Number of qubits) How much information can be encoded in a quantum device? The number of qubits grows steadily due to extended research on materials, fabrication technologies and physical ways in which the qubits are encoded (trapped-ions, cold atoms, photons, etc.). However the true challenge is to make scaling possible while maintaining the coherence between qubits in the device.

- **Quality:** (Quantum Volume [22]) How faithfully a circuit can be implemented in a quantum computer? Quality of a device is a holistic metric due to its dependence on multiple factors, such as qubit connectivity, coherence, gate fidelity and measurement fidelity.

- **Speed:** (Circuit Layer Operations per Second or CLOPS) How fast is the quantum device? In [74], CLOPS are proposed as a counterpart to the classical metric of computer performance floating point operations per second, or FLOPS. It is proposed to capture the real-world use by considering the performance of the run-time environment that invokes the circuits. In the reference, there is also a detailed description on how to calculate that number using parametrized circuits as a good representation of systems with non-persistent quantum data across multiple invocations.

The discussion regarding quantum advantage and quantum supremacy has been disputed, to say the least, in recent years. Since the announcement of quantum supremacy from the Sycamore processor using 53 qubits [8], there have been research groups that challenged those claims by showing that similar (if not better) results have been achieved using classical devices and different, more optimal algorithms. One of the most recent episodes regarding this was a paper claiming evidence for quantum computers simulating an Ising model beyond quantum capabilities by combining error mitigations techniques [36]. Shortly after the publication, the community responded, referring to an earlier publication claiming to do the same thing using the same class of parametrized quantum circuits [48]. As the authors of this last work state in their abstract, ”Understanding the limits of classical computing in simulating quantum systems is an important component of addressing this question“. So, let’s keep an eye on this interesting discussion while working on quantum computing.

4.3. Variational Quantum Eigensolver

Variational Quantum Eigensolver (VQE) appears as one of the most simple implementations of VQCs. However, it is a friendly entrypoint for the work on quantum computing in the NISQ era. VQEs were first proposed in 2014 [58], but there are currently several reviews on the intricacies of this algorithm [73, 28], as each an every one of the components of the circuit play an important role in the efficiency and accuracy of the results. VQE is grounded in the variational principle, which optimizes an upper bound for the lowest possible expectation value of an observable with respect to a trial quantum state. Providing a hamiltonian H, a a trial wavefunction $|\psi\rangle$, the ground state energy associated with this hamiltonian, E_0 is bounded by

$$E_0 \leq \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle}. \tag{4-2}$$

The goal of VQE is to find the parametrization of $|\psi\rangle$ such that the expectation value of the Hamiltonian is minimized. Then, $|\psi\rangle$ can be expressed as the application of a generic parametrized unitary operator $U(\theta)$ over an initial state of N qubits, which is the heart of the variational circuits method. The qubit register is usually initialized in the vacuum or empty state, namely $|0\rangle^{\otimes N}$, but any other low-depth initialization can be done depending on the system. The optimization problem then takes the form

$$E_{VQE} = \underset{\theta}{\operatorname{argmin}} \langle 0| U(\theta)^{\dagger} H U(\theta) |0\rangle . \tag{4-3}$$

To perform this procedure on a quantum computer, it is required that both the parametrized unitary $U(\theta)$ and the Hamiltonian H are described using the building blocks used by such devices: qubit operators. Therefore, H is defined and mapped into qubit operators giving as a result a linear combination of Pauli strings $P_a \in \{I, X, Y, Z\}^{\otimes N}$, with N the number of qubits required to represent the Hamiltonian,

$$H = \sum_{a}^{\mathcal{P}} \omega_a P_a, \tag{4-4}$$

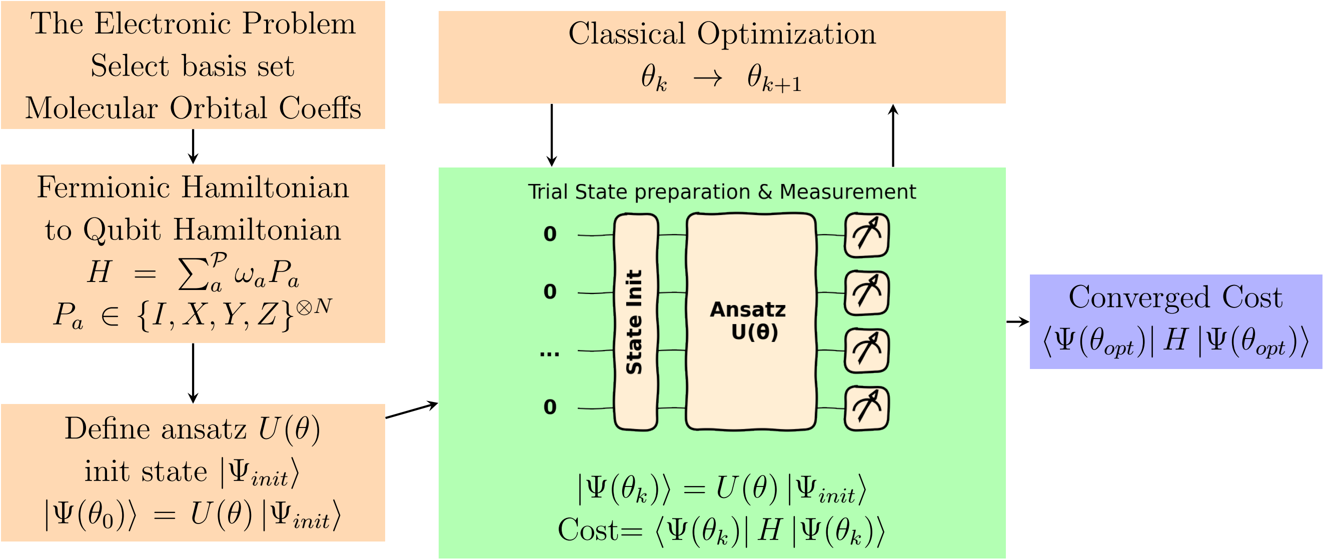

The Electronic Problem
Select basis set
Molecular Orbital Coeffs
Fermionic Hamiltonian
to Qubit Hamiltonian
$H = \sum_a^{\mathcal{P}} \omega_a P_a$
$P_a \in \{I, X, Y, Z\}^{\otimes N}$
Define ansatz $U(\theta)$
init state $|\Psi_{init}\rangle$
$|\Psi(\theta_0)\rangle = U(\theta)|\Psi_{init}\rangle$
Classical Optimization
$\theta_k \rightarrow \theta_{k+1}$
Trial State preparation & Measurement
0
0
...
0
State Init
Ansatz
U(θ)
$|\Psi(\theta_k)\rangle = U(\theta)|\Psi_{init}\rangle$
Cost= $\langle\Psi(\theta_k)| H |\Psi(\theta_k)\rangle$
Converged Cost
$\langle\Psi(\theta_{opt})| H |\Psi(\theta_{opt})\rangle$

Fig. 4-1: Detailed diagram of Variational Quantum Algorithm in an hybrid architecture, used to get the ground energy of a molecular system (based on figure **1-1**). The yellow boxes represent calculations and executions in a classical device; the green box represent sections where quantum processors are required. After the molecular system is analysed and the Hamiltonian is written as a linear combination of Pauli strings, the parametrized circuit is used to calculate its value. Then, training the circuit is traduced to find the optimal circuit parameters that minimize the energy.

with ω_a the coefficients and $\mathcal{P}$ the set of Pauli strings in the Hamiltonian. The optimization problem now reads

$$E_{VQE} = \operatorname*{argmin}_{\theta} \sum_{a}^{\mathcal{P}} \omega_a \langle 0| U(\theta)^{\dagger} P_a U(\theta) |0\rangle = \operatorname*{argmin}_{\theta} \sum_{a}^{\mathcal{P}} \omega_a \langle H_a \rangle, \tag{4-5}$$

which is just the sum of the expectation values of each individual term in the encoded Hamiltonian. Obtaining these expectation values $\langle H_a \rangle$ is the main job of the quantum processor, and is the place where the quantum advantage is expected to appear. The components of the VQE method are depicted in figure **4-1**, and the steps beyond the Hamiltonian caluculation are discussed in more detail in the next subsections.

4.3.1. Cost Function

Choosing a suitable way to encode the optimization problem into a scalar cost function is crucial. In a machine learning approach, the cost defines a hyper-surface and the task at hand is to navigate through this landscape to find the global minima. In that sense, the change of the calculated cost in each iteration gives information about the hyper-surface itself. Usually, the optimization procedure ends when the cost value change between iterations is lower than a predefined threshold, signalling that a local or global minimum has been reached.

The cost function C uses as argument the variational parameters θ, but can also use a set of non-adaptable parameters $x = (x_1, x_2, ...)$ such that

$$C(x, \theta) = \langle 0| U(x, \theta)^\dagger H U(x, \theta) |0\rangle . \tag{4-6}$$

Following [14] regarding *Variational Quantum Algorithms*, Cerezo *et al.* list some desirable criteria for the cost function. Some of the required conditions are that the cost function should be:

- Faithful: The parameters that minimize the function can be used to solve the problem.
- Efficiently solvable in a quantum device: No just that the solution can be found using a quantum processor, but also that the problem is not efficiently computable in a classical device, implying that no quantum advantage can be achieved using VQE.
- Meaningful: Smaller cost values indicate higher quality of the solution.
- Trainable: It is possible to optimize the parameters θ in the VQC.

In the VQE framework, the used cost function is the encoded expected value Hamiltonian $\langle H\rangle$, which simplifies enough the problem: let's find the minimal energy of the system by finding the circuit that returns the lowest $\langle H\rangle$. The way the hamiltonian is created and encoded is key in VQE, as it defines the number N of qubits required to represent the electronic quantum state. It also specifies the number of Pauli strings and the Pauli weight (i.e. the number of tensor factors in the Pauli string that are not equal to the identity) which also affect the circuit length.

As the cost function is the expected value of an operator, a single run of the circuit is not enough to obtain its value. The fact that quantum computers are inherently probabilistic machines, accompanied by noise and errors, require multiple executions of the circuit in order to obtain a probability distribution that allows to calculate the desired expectation value. This is why it is necessary to define how many times (i.e. the number of *shots*) an algorithm is going to run to get the probability distribution of the results. There is no rule to define an optimal number of shots,and just like all the other buttons and knobs of machine learning, it can be set by trial and error by increasing the number until the results are statistically accurate but also reducing the number until the execution time is optimal. Some common shots values are 1024 or 1000, but usually SDKs for quantum computing allow setting different numbers. Additionally, there are several shot optimization procedures focusing on reducing the number of shots while keeping the accuracy of the results [40, 4, 59].

4.3.2. State Encoding and Initial State

A qubit, or quantum bit, has two distinct states usually represented as $|0\rangle$ and $|1\rangle$. The fact that they can exist in superposition states and be entangled gives rise to exciting possibilities. One of the new interesting things is the way information is encoded in such states. The way this is done is by using a *Quantum Embedding* [43, 66]. This is especially important when considering the embedding of classical data into quantum states. For example, consider M items with N features

$$\mathcal{D} = x^{(1)}, ..., x^{(m)}, ..., x^{(M)},$$

where $x^{(m)}$ is an N dimensional vector from $m = 1, ..., M$. To embed this data in a quantum system, an embedding technique is required.

The *Basis embedding* links each input with a computational basis state of the qubit system. In this case, classical data takes the form of binary strings. Having the classical data in the form $x^{(m)} = (b_1 ..., b_N)$, with $b_i \in \{0, 1\}$ for $i = 1, ..., N$, each input item $x^{(m)}$ can be directly mapped to the quantum state $\left|x^{(m)}\right\rangle$. The entire dataset thus takes the form

$$\mathcal{D} = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \left|x^{(m)}\right\rangle .$$

Another common embedding is the *Amplitude Embedding*, which encodes the data into the amplitude state. A normalized classical N-dimension vector $x^{(m)}$ is represented by the amplitudes of an n-qubit quantum state

$$|\psi_x\rangle = \sum_{i=1}^{N} x_i \, |i\rangle ,$$

where $N = 2^n$, x_i is the i-th element of $x^{(m)}$, and $|i\rangle$ is the i-th computational basis state.

Finally, the embedding method used for VQE is the *occupation embedding* or occupation number representation. As the Hamiltonian of the electronic system is composed of n electrons occupying spin orbitals, this embedding readily displays which orbitals have excitations (i.e. electrons), and which are empty. Even better, when working with fermions, the occupation p_i of each orbital can only be 1 or 0 due to the Pauli exclusion principle, which allows to work directly with the computational basis (using the Jordan-Wigner transformation from fermionic to qubit Hamitonians, or some other similar transform). For $n = 2$ electrons in $K = 4$ spin orbitals, a valid occupation state is $|\psi\rangle = |1100\rangle$.

However, another valid state is a superposition over all the states of the computational basis

$$|\psi\rangle = c_1 \, |1100\rangle + c_2 \, |1010\rangle + c_3 \, |1001\rangle + c_4 \, |0110\rangle + c_5 \, |0101\rangle + c_6 \, |0011\rangle .$$

Perhaps neither of the states displayed above correspond with the state that returns the lowest expected value of the Hamiltonian. However, it is useful to start the training process with a state that conveys some information of the studied system; in this case, the number of electrons in the molecule. For VQE, it is usual to use the first few gates of the circuit to initialize the qubit register. It is customary to use the Hartree-Fock state as the initial state, where the excitations fill the lower spin orbital levels. In the circuit, this is seen as a single Pauli X gate acting on the wires with the excitation.

4.3.3. Circuit Selection and Construction

A very important aspect of a VQC is its circuit layout, usually called ansatz. The form of the ansatz dictates what the parameters θ are and how many there are, which defines how they can be efficiently trained to get an accurate response. The specific structure of an ansatz generally depends on the task at hand, using general information of the system. However, there are some general and problem-agnostic architectures that are used when no prior information is available, or when an even shorter, shallow circuit is desired for the calculation. A circuit $U(x,\theta)$ can be expressed as the product of L sequentially applied unitary operators

$$U(x,\theta) = U_L(\theta_L)\cdots U_2(\theta_2)U_1(\theta_1)U_0(X), \tag{4-7}$$

in which each layer $U_l(\theta_l)$ can be composed by unparametrized unitary operators along with gates function of θ_l, and the initial gate $U_0(X)$ performs the state initialization using some predefined metaparameter x. Remember that all unitary operators act over some set of wires (a single wire, some of them, or even all).

The ansatz discussed in this work are *fixed structure ansatz*, which are set at the beginning of the optimization process and remain unchanged afterwards. There is another important type, called *adaptive structure ansatz*, referring to the circuits constructed iteratively as part of the optimization process [32, 55, 54, 25]. In the table **4-1** three key metrics to compare ansatzë are presented:

- *Depth:* Number of sequential operations required for the gate implementation. This impacts the overall runtime of the circuit and its resilience to noise.
- *Number of parameters:* Has influence over the overall implementation and the complexity of the optimization process.
- *Number of entangling gates:* In general is the main source of noise resulting from the execution of a quantum circuit.

Finally, before talking about some of the multiple available ansatz, it is important to remember that these circuits eventually run in a physical quantum device. The quantum circuits discussed are abstract objects whose qubits are virtual representations of actual qubits used in computations. In the same way, the quantum gates of these circuits may not be available in a quantum device, or they may require multiple (sometimes dozens of simple gates). Notice that for a physical implementation one needs to deal with several levels of

Table 4-1: Summary of circuit depth, parameters and entangling gates across several ansatz. The number of entangling gates does not consider possible extra gates required for real devices without full connectivity. The number of layers is denoted by L, the number of qubits (i.e. wires in the circuit) is n.

Method	**Depth**	**Parameters**	**Entangling gates**	**Notes**
Two local Circuit	$\mathcal{O}(L)$	$\mathcal{O}(L(N-1))$	$\mathcal{O}\left(\frac{L(N-1)}{2}\right)$	L is an arbitrary number of layers.
Single Givens Rotation	$\mathcal{O}(3\tilde{k})$	$\mathcal{O}(\tilde{k})$	$\mathcal{O}(3\tilde{k})$	$\tilde{k}$ as the number of single excitations used
Double Givens Rotation	$\mathcal{O}(20\tilde{k})$	$\mathcal{O}(\tilde{k})$	$\mathcal{O}(14\tilde{k})$	$\tilde{k}$ as the number of double excitations used
k-UpCCGSD	$\mathcal{O}(70kN)$	$\mathcal{O}\left(2k\tilde{q}\right)$	$\mathcal{O}\left(16k\tilde{q}\right)$	k as the number times UpCCGSD unitary is repeated. $\tilde{q}$ is the number of generalized singles and paired doubles excitation terms.

circuit encoding. The goal is to reduce the input circuit into a series of gates and an optimal layout that can be understood by the physical device, so the device can execute the gates and move the qubits accordingly. This is done through *Transpilation*, where the input circuit is rewritten to match the topology of a specific device.

Simplified Two Design

The hardware efficient ansatz [35] is a generic name used for ansatzes aimed at reducing the circuit depth required to implement $U(\theta)$ when using a quantum hardware. Specifically, these kinds of circuits are tailored to create the unitaries using the quantum gates offered by the quantum device on which the experiment may run. All the proposed hardware-efficient ansatz circuits consist of repeating blocks on successive single parametrized rotation gates, followed by ladders of entangling gates. Both rotating and entangling gates can vary depending on the native gate set of the device.

One of the main advantages of this kind of circuit is its versatility, as it can accommodate encoding symmetries and ensure correlated qubits are together to ensure depth reductions, making the transpilation easier. However, these circuits have some important limitations. The first one arises directly from the fact that the circuit has no prior knowledge and/or information of the system under study. This means that it must span a very large portion of the Hilbert space of states to guarantee it can get an accurate enough representation of the ground state. In this regard, problem-inspired ansatz start with an advantage, spanning a portion of the space where the solution is expected to be. There are several works about the training of these circuits over a cost landscape such as [41, 19]

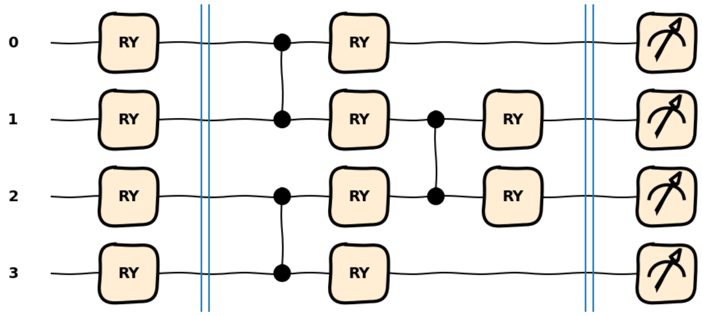


Fig. 4-2: Layout for Simplified Two Design ansatz. The entangling gates are implemented by pairs, which comes in handy for the partially connected real devices in the NISQ era.

Two-Local Circuit (figure **4-2**) is one of the most widely used hardware-efficient ansatz, consisting of alternating rotation layers and entanglement layers proposed by Cerezo et al. in [15]. The Simplified Two Design ansatz used in this work uses layers consisting of a simplified 2-design architecture of Pauli-Y rotations and controlled-Z entanglers. It is interesting as it exhibits important properties to study barren plateaus in quantum optimization landscapes, while keeping a simple connectivity and a low gate number.

Changing the number of layers has a great impact on the accuracy of the circuit. For hardware efficient ansatz, increasing the number of used layers becomes mandatory as just one layer might not have access to the searched state into the Hilbert space. In the figure **4-3** it's clear how one layer of the circuit is insufficient for the calculation. When two layers are used, the performance increases significantly for points after the minimum value, even better for some points compared with circuits using more layers. With three and four layers the errors are within the chemical accuracy. It's worth noticing how there is no important improvement between three and four layers, which might be caused by a saturation of the circuits.

Combinations of Single and Double Excitations

Using the occupation encoding and the Jordan-Wigner transformation provides a convenient way to describe the occupation of the molecular orbitals. And having a fixed number of excitations (i.e. the number of electrons), superposition over all basis states are valid states describing electrons on a molecule. For $n = 2$ electrons in $k = 4$ spin orbitals, a valid

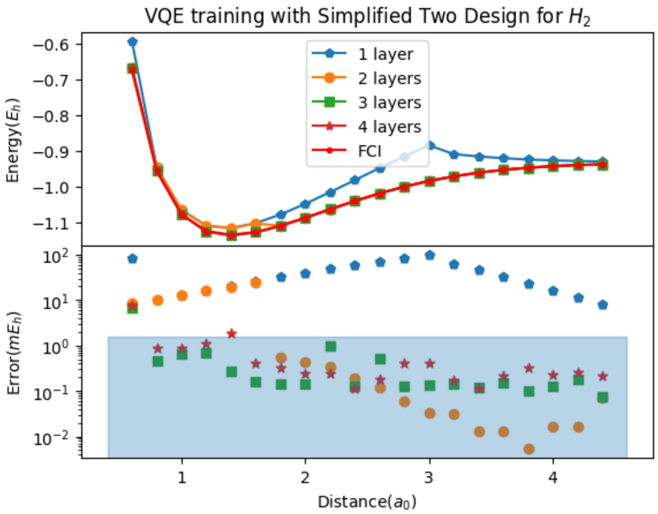


Fig. 4-3: Potential Energy Surface plots for 1 to 4 layers of Simplified Two Design ansatz of the H_2 molecule. For this plot, STO-3G basis was used along with DEMON ADAM optimizer. At the bottom, the error of training using each circuit in a log scale. Notice that using 3 and 4 layers produces results inside the chemical accuracy.

occupation state is $|\psi\rangle = |1100\rangle$. However, another valid state is a superposition over all the states of the computational basis

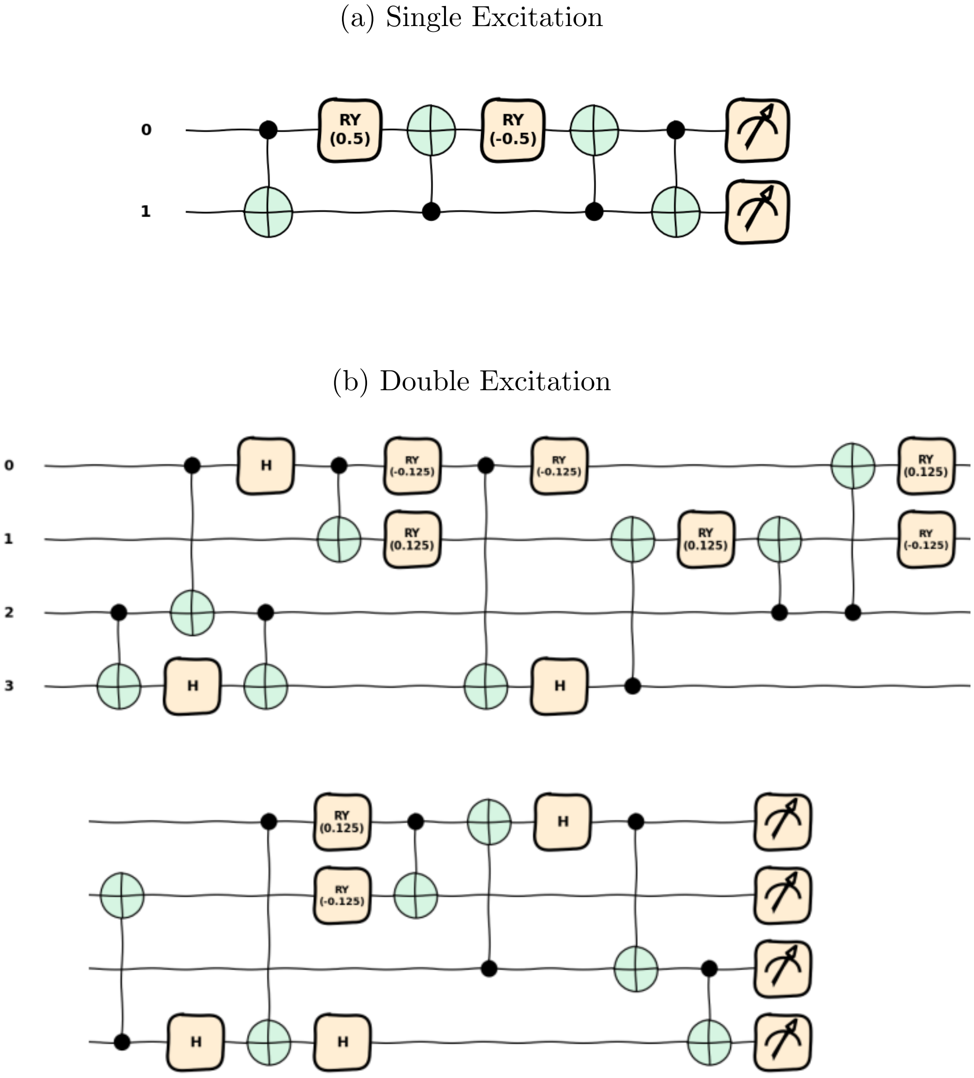
$$|\psi\rangle = c_1 |1100\rangle + c_2 |1010\rangle + c_3 |1001\rangle + c_4 |0110\rangle + c_5 |0101\rangle + c_6 |0011\rangle , \qquad (4\text{-}8)$$

where all basis states show the same number of excitations: 2. Any transformation capable of conserve the number of particles is called a *particle-conserving unitary operation*. For VQCs, these kind of gates are not strictly necessary, as it is possible to produce wavefunctions with the wrong number of excitations but the correct energy of the system. However, particle-conserving unitary operators ensure the resulting wavefunction produced is a valid state that can be used in some other procedures.

The simplest particle-conserving unitaries are the *Givens Rotations*, which describe an arbitrary U(2) rotation in the subspace spanned by two states, parametrized by an angle ϕ. The rotation can be also understood as an excitation, as the transformation from one basis state to the other is interpreted as “exciting” a particle. For instance, a single excitation gate is expressed by the two-qubit gate

$$G(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\phi/2) & -sin(\phi/2) & 0 \\ 0 & sin(\phi/2) & cos(\phi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (4\text{-}9)$$

and the circuit representation is depicted in figure **4-4** (a). The double excitation circuit (figure **4-4** (b)) shares the same principle of controlled rotations, and its use in the ansatz can improve greatly the performance of the training.


(a) Single Excitation

(b) Double Excitation

Fig. 4-4: Circuit representation for a single and double excitation gates. In (a), the excitation goes from the qubit 0 to the qubit 1. The middle four gates constitute a controlled **Y** rotation, where each angle rotation is half the excitation parameter (in this case, $\theta = 1$). In (b), the excitations goes from the qubits 0 and 1 to the qubits 2 and 3. The circuit is clearly larger and more intricate that the single excitation circuit, but share the same principle of using controlled **Y** rotations.

An ansatz can be created using combinations of single and double excitations from the basis occupation states of the system. Although these kinds of gates are not universal [3], it is show in [6] that controlled single-excitation gates in the form of Givens rotations are universal. There are also some methods that allow to select only the excitations that are found to be important for the given molecule [32].

Unitary Coupled Clustered Ansatz

Given the current effervescence of the research in quantum computing, it is possible to find a problem-inspired ansatz for almost every problem of interest for the community. For quantum chemistry problems, the Unitary Coupled (UCC) ansatz is a circuit widely used where the goal is to obtain the ground state energy of a fermionic molecular Hamiltonian H.

For quantum chemistry problems, the Unitary Coupled (UCC) ansatz [9, 2] is a circuit widely used where the goal is to obtain the ground state energy of a fermionic molecular Hamiltonian H. The UCC ansatz propose a candidate for such ground state based on exciting some reference state $|\psi_0\rangle$ as $e^{T(\vec{\theta})-T(\vec{\theta})^\dagger}|\psi_0\rangle$. Here, $T=\sum_k T_k$ is the cluster operator and T_k are excitation operators. There T_k are actual excitations of the state, in contrast with the Givens rotations. For UCCSD (SD stands for singles and doubles) the summation on the cluster operator is truncated to contain single excitations $T_1=\sum_{pq}\theta_{pq}a_p^\dagger a_q$, and double excitations $T_2=\sum_{pqrs}\theta_{pqrs}a_p^\dagger a_q^\dagger a_r a_s$. To implement this ansatz the transformation from fermionic to spin operators (usually the Jordan-Wigner transformation) is used.

In this work, the k-Unitary Pair Coupled-Cluster Generalized Singles and Doubles (k-UpCCGSD) ansatz [70, 2] is used. The ansatz uses k repetitions (layers) of generalized singles and pair coupled-cluster doubles excitation operators. Here, "generalized"means that the single and double excitation terms do not distinguish between occupied and unoccupied orbitals. Also,"pair coupled-cluster"means that the double excitations contain only those two-particle excitations that move a pair of electrons from one spatial orbital to another.

4.3.4. Gradients

In order to train a Variational Quantum Circuit in the NISQ era, a classical device is required. And the used methods are based on training procedures that have been getting continually better through the last decades with the widespread use of Machine Learning. This procedures can be divided in two big sections

- Gradient-based Strategies.
- Gradient-free Strategies.

Each strategy comes with its own advantages and challenges. As a significant amount of optimization methods use the gradient (or higher-order derivatives) of the objective functions at a given parameter value, some ways to obtain these values are presented here. Interestingly enough, Schuld et al. [65] show how gradients of the cost function can be evaluated using the same architecture of the original circuits, or a very similar one.

A common approach to evaluate gradients is to use stochastic approximation methods. *Finite Difference Stochastic Approximation (FDSA)* is one of the simplest methods, which uses two cost points with a small shift of the parameter of interest, and uses the difference as the approximated gradient

$$\frac{\partial f}{\partial \theta_i} = \frac{f(\theta + c_t e_i) - f(\theta - c_t e_i)}{2c_t}, \tag{4-10}$$

where c_t is the unit vector with a 1 in the i-th dimension. There are several strategies of the finite difference method that define how the small shift is done. A symmetrical shift around the starting point specified in eq. 4-10 is very common. Besides that, the "forward" and "backward" methods use the value of the function at a starting point θ_0 and make a shift forward $(\theta_0 + h)$ or backward $(\theta_0 - h)$. This method calculates the gradient using *Number of params + 1* runs of the circuit, as the value in the starting point is reused.

An improved stochastic method is the *Simultaneous Perturbation Stochastic Approximation (SPSA)* [72], where the perturbation is done over all parameters and not just one as in FDSA. For this method the gradient takes the form

$$\frac{\partial f}{\partial \theta_i} = \frac{f(\theta + c_t \Delta) - f(\theta - c_t \Delta)}{2c_t \Delta_i}, \tag{4-11}$$

where Δ is a random perturbation vector and Δ_i is the displacement over the i-th dimension. Notice that, once the perturbation vector Δ is defined, the numerator in 4-11 is unchanged, which reduces the number of required cost calculations in half.

A different approach to evaluate the gradients is the *Parameter-Shift rule* [65, 21], that has the great advantage to be analytic and exact. An objective function expressed as an expected value of an observable (eq. 4-6) uses an unitary transformation $U(x, \theta)$ carried out by the circuit, which can be broken into a product of unitaries, as expressed in eq. 4-7. As each of these gates are unitary and must have the form $U_l(\theta_l) = exp(-ia\theta_l G_l)$ where G_l is a Hermitian operator which generates the gate, a is a real constant, and θ_l is the gate parameter.

Without loss of generality, consider the cost function

$$f(\theta) = \langle\psi| U(\theta)^\dagger H U(\theta) |0\rangle, \tag{4-12}$$

where the unitary operator of the circuit takes the form

$$U(\theta) = e^{-ia\theta G},$$

the operator G acting as the gate generator. The parameter-shift rule then states that if the generator G has only two eigenvalues (e_0 and e_1), then the derivative of the expectation value $f(\theta)$ is proportional to the difference in expectation of two circuits with shifted parameters

$$\frac{\partial}{\partial \theta} f(\theta) = r \left[f\left(\theta + \frac{\pi}{4r}\right) - f\left(\theta - \frac{\pi}{4r}\right) \right], \tag{4-13}$$

where the shift constant $r = \frac{a}{2}(e_0 - e_1)$. The parameter shift rule applies to several cases, being one of the most important the 1-qubit Pauli rotation gates with $a = 1/2$ and $r = 1/2$. As the parametrized gates of the Simplified Two Design ansatz use these kinds of gates, two circuit runs per parameter are needed to get a parameter shift gradient. For more complex gates, such as controlled rotation or excitation preserving gates (e.g. single or double excitation gates), it is required a *four-term* parameter shift rule [3]. This is the case for single and double excitation gates (each one using a single parameter), and also for every parameter of the k-UpCCGSD ansatz.

4.3.5. Optimization Methods

As for any variational approach, the success of employing a Variational Quantum Circuit depends on the efficiency and reliability of the optimization method used. For the algorithm to be useful, it is required that it can learn a good enough approximation of the solution within an acceptable number of learning steps. However, compared to conventional numerical optimization problems, optimizing the expectation value of a variational quantum ansatz faces additional challenges. The biggest issue currently is the sampling noise and gate noise on NISQ devices, which disturb the landscape of the cost function. The precision of the found solution can also be an issue because it is limited by the sample shot number when calculating the expectation value. In this section several optimization methods are presented, with a detailed description of the algorithms. The used values in the execution of each method are presented in the requirements of the algorithm. The values used are the suggested ones by the literature [73].

The most common approaches to optimization fall under the category of *Gradient Descent*, where the gradient indicates the direction to take for each iteration step. Into this category, the simplest of all is *Simple Gradient Descent* (detailed in algorithm 1), where a parameter θ is updated by taking a step toward the opposite direction of the gradient. For this method, the used gradient can be calculated in any available way.

In the same line, the *Simultaneous Perturbation Stochastic Approximation method* (SPSA) [72, 71] updates the parameters in each step according to the calculated gradient, and the gradient is approximated through a simultaneous perturbation of the input parameters. The SPSA method (detailed in algorithm 2) involves evaluating the cost function twice in each iteration step. Notice how this method uses several nonnegative coefficients a, c, A, α, and γ, which is critical to the performance of the optimization algorithm. In [71], the creator of the algorithm gives some guidelines on how to pick these coefficients. In particular, in high-noise settings, he recommends selecting a smaller a and larger c than in a low-noise setting. In that reference, he also suggests using $\alpha = 0{,}602$ and $\gamma = 0{,}101$ as the lowest allowable values.

One more gradient-based algorithm for training a circuit is the *Adaptative Moment Estimation method* (ADAM) [37], presented as computationally efficient and optimal when working with very noisy and/or sparse gradients. In each training step, ADAM (detailed in

Algorithm 1 Simple Gradient Descent

```
    Hamiltonian operator $H$ encoding the cost function $\langle H(\theta)\rangle$, parametrized quantum circuit $U(\theta)$ with $D$ parameters to train.
Require:
    $\eta$: Step size (0,1)
    $\epsilon$: Limit of convergence ($1 \times 10^{-5}$)
Initialize $\theta_d \in (-\pi, \pi]$ for $d = 1, ..., D$ heuristically or at random
repeat
    for $d = 1, ..., D$ do
        Get gradient $g_d^{(t)} = \frac{\partial \langle H(\theta^{(t)})\rangle}{\partial \theta_d}$
        $\theta_d^{(t-1)} \leftarrow \theta_d^{(t)} - \eta g_d^{(t)}$
    end for
until $|\langle H(\theta^{(t+1)})\rangle - \langle H(\theta^{(t)})\rangle| < \epsilon$ or max iterations
```

Algorithm 2 Simultaneous Perturbation Stochastic Approximation (SPSA)

```
    Hamiltonian operator $H$ encoding the cost function $\langle H(\theta)\rangle$, parametrized quantum circuit $U(\theta)$ with $D$ parameters to train
Require:
    $\alpha$: Scaling Exponent (0,602)
    $\gamma$: Scaling exponent (0,101)
    $c$: Hyperparameter related to the expected noise (0,2)
    $A$: Scaling parameter ($0{,}1 * maxiter$)
    $a$: Scaling parameter ($0{,}05 * [A+1]^{\alpha}$)
    $\epsilon$: Limit of convergence ($1 \times 10^{-5}$)
Initialize $\theta_d \in (-\pi, \pi]$ for $d = 1, ..., D$ heuristically or at random
repeat
    in step $t$
    $a^{(t)} \leftarrow \frac{a}{(t+A)^{(\alpha)}}$
    $c^{(t)} \leftarrow \frac{c}{t^{\gamma}}$
    Random perturbation vector $\Delta^{(t)} = [\Delta_1^{(t)}, ..., \Delta_D^{(t)}]$
    Gradient vector $g^{(t)}(\theta^{(t)}) = \langle H(\theta^{(t)} + c^{(t)}\Delta^{(t)})\rangle - \langle H(\theta^{(t)} - c^{(t)}\Delta^{(t)})\rangle / 2c^{(t)}\Delta^{(t)}$
    $\theta^{(t+1)} = \theta^{(t)} - a^{(t)} g^{(t)}(\theta^{(t)})$
until $|\langle H(\theta^{(t+1)})\rangle - \langle H(\theta^{(t)})\rangle| < \epsilon$ or max iterations
```

algorithm 3) introduces the effect of past cost changes on the current direction of movement in the cost landscape, while also adapting the used learning rate. Considering the effect of past costs provides a kind of 'momentum' in the cost landscape that effectively filters out high-frequency variations [64]. This is done by computing a weighted moving average of the gradient

$$m^{(t+1)} = \beta m^{(t)} + (1-\beta) g^{(t)}, \tag{4-14}$$

where $m^{(t)}$ is the momentum used to update the parameter, and β controls the moving average, by gauging the relevance of the previous momentum. Regarding the adapting learning rate (key in the *RMSProp* optimization method [31]), the moment is divided by the weighted moving average of the root mean square of the gradient value, which improves the direction of the step to take and reduces the significance of its magnitude.

The training results from ADAM are generally accurate results. Its adaptative learning rate and momentum-based approach let a circuit learn faster and converge more quickly towards an optimal set of parameters that minimize the cost function. There are several optimization methods with a direct relation to ADAM and worth considering such as the already mentioned RMSProp, ADAMAX ([37], section 7.1), *AdaGrad* [26] , *AdamW* [44], and *Decaying Momentum (DEMON) ADAM* [18]. There are several works comparing optimizers for Quantum Machine learning such as [76], where several optimizers are used along with SPSA-based gradients, and [11] where gradient-free optimization methods are compared with SPSA.

DEMON ADAM is one of the most promising improvements over ADAM, and can be applied to any gradient descent algorithm with momentum parameters. The researchers that introduce DEMON mention that it is less sensitive to parameter tuning while also performing very well training neural networks. By decaying the momentum parameter, the total contribution of a gradient to all future updates is decayed. This is done with the advantage that no hyperparameter tuning is required, as the momentum parameter usually decays to 0 or a small value. To apply DEMON, the parameter β_1 is a function of the training step t

$$\beta_1 \rightarrow \beta_t = \beta_{init} \frac{\left(1 - \frac{t}{T}\right)}{(1-\beta_{init}) - \beta_{init}\left(1 - \frac{t}{T}\right)}, \tag{4-15}$$

where β_{init} is the value set for the momentum parameter at the beginning of the training and total T steps.

As a final word for this topic, it's important to understand that optimization problems in multidimensional spaces are a challenging task. And more so for the optimization landscape of variational hybrid algorithms, whose cost function shows many local minima and barren plateaus. The number of variables, as well as the difficulty to evaluate the cost function or its gradients are key in selecting an useful optimizer. If the model is very cheap (easy cost and gradient calculation) it is possible that any of the available optimizers will provide a reasonable solution within a reasonable time. But in general, gradient-based optimization methods are the most efficient way to explore the cost landscape; even better when the optimizers are sophisticated enough to ensure a quick convergence.

Algorithm 3 Adaptative Moment Estimation (ADAM)

```
    Hamiltonian operator $H$ encoding the cost function $\langle H(\theta)\rangle$, parametrized quantum circuit $U(\theta)$ with $D$ parameters to train
Require:
    $\eta$: Step size (0,01)
    $\beta_1 \in [0,1)$: Update rate for the first momentum (0,9)
    $\beta_2 \in [0,1)$: Update rate for the second momentum (0,99)
    $\epsilon_0$: Offset for numerical stability ($1 \times 10^{-8}$)
    $\epsilon$: Limit of convergence ($1 \times 10^{-5}$)
    Initialize $\theta_d \in (-\pi, \pi]$ for $d = 1, ..., D$ heuristically or at random
    $m^{(0)} \leftarrow 0$ Initialize first moment
    $v^{(0)} \leftarrow 0$ Initialize second moment
    repeat
    in step $t$
        for $d = 1, ..., D$ do
            Get gradient $g_d^{(t)} = \frac{\partial \langle H(\theta^{(t)})\rangle}{\partial \theta_d^{(t)}}$
            $m^{(t+1)} \leftarrow \beta_1 m^{(t)} + (1 - \beta_1) g_d^{(t)}$
            $v^{(t+1)} \leftarrow \beta_2 v^{(t)} + (1 - \beta_1)(g_d^{(t)})^2$
            $(m^{(t+1)}) \leftarrow m^{(t+1)}/(1 - \beta_1)$
            $v^{(t+1)} \leftarrow v^{(t+1)}/(1 - \beta_2)$
            $\theta_d^{(t+1)} \leftarrow \theta_d^{(t)} - \eta m^{(t+1)}/(\sqrt{v^{(t+1)}} + \epsilon_0)$
        end for
    until $|\langle H(\theta^{(t+1)})\rangle - \langle H(\theta^{(t)})\rangle| < \epsilon$ or max iterations
```

4.4. VQE with multiple configurations for H_2

Training results in VQE can vary greatly regarding on the configuration of ansatz, gradient method and optimizers used. And keep in mind that each possible selection in one of the pieces mentioned can have multiple variants, with equal (if not more) importance, that allow a very fine tuning of the training procedure. In this word, simulations of VQE performed using PennyLane are presented. For this, two big approaches are presented: *analytical* execution, where the results of the circuits and expectation values are computed exactly from the quantum state; and *shots-based* execution, the way expected in a real quantum processor, where the circuit is evaluated (or "sampled") a number of times to estimate statistical quantities. As the cost function used in VQE is based on expectation values, the number of shots affects greatly the accuracy of the results.

4.4.1. Simulation details

Here you can find the details of the simulations done to calculate the ground energy for a H_2 molecule using VQE.

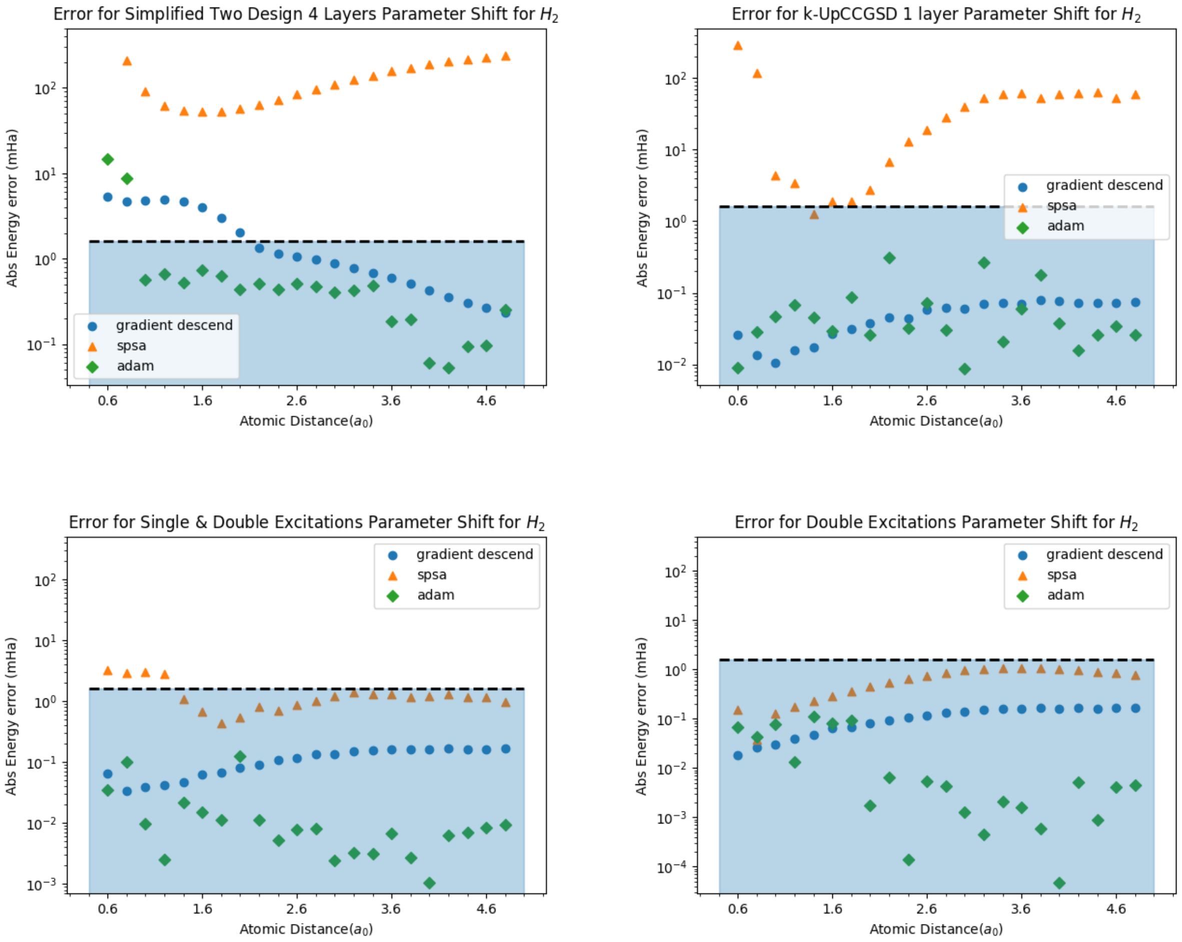
- The electronic wavefunction is described using LCAO of lowest energy states S_1, and the chemical basis set STO-3G.
- The integrals required to get the molecule hamiltonian are calculated using the PennyLane wrapper of PySCF.
- 150 steps max in each training method are used.
- FCI (Full Configuration Interaction) is used as comparison value for error calculation, using chemical accuracy (Error $< 1{,}6mHa$) as baseline

4.4.2. Analytic Approach

The analytic results are, as expected, very accurate. For all the ansatz tested (given there were enough layers used in the case of hardware efficient ansatz) it was possible to train the circuits and obtain energy results within chemical accuracy (shaded region). In the figure **4-5** is clear how the use of different optimizers affects the results, specifically the absolute difference between the FCI energies and the ones obtained. For an analytics calculation, the differences caused by using finite differences or parameters shift rule are very small. Both gradient descend and ADAM give great optimization results, but ADAM behaves better in general. This is convenient ad ADAM uses the same circuit resources per run to get the gradient of the circuits (regardless of the method) as gradient descend, but performs better. On the other hand, SPSA behaves good enough with few parameters to train, as in the excitations ansatz, but shows significant energy differences for circuits with many parameters, such as the Simplified Two Design Ansatz.

4.4.3. Shots-based approach using 10000 shots

When considering the shots based approach, the training results are somewhat more interesting. In this scenario, using parameter shift rule to calculate the gradients do make an impact in the accuracy of the procedure. Figure **4-6** displays the absolute error of multiple running configurations, obtained from the average of 5 complete runs of each configuration, using 10000 shots. The results in the left column are obtained using finite differences, and the ones in the right use parameters shift rule. SPSA outperforms the other optimizers when they use finite differences. It is important to remember that SPSA uses an stochastic approximation to obtain the gradients, so it is expected to vary between runs and configurations. ADAM always outperforms gradients descend, which again give good word regarding the optimization procedure, and achieves chemical accuracy (or gets very close) in all considered ansatz.



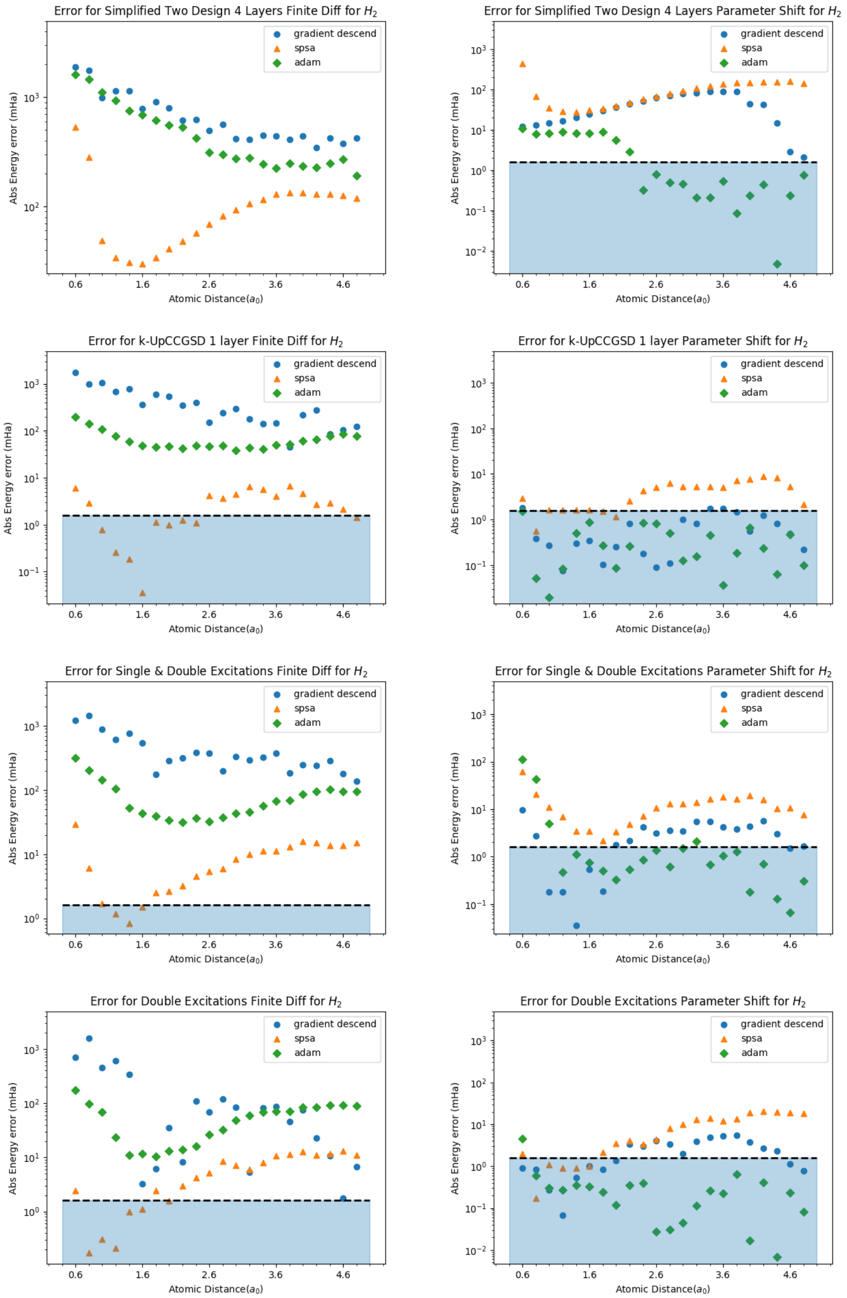
Error for Simplified Two Design 4 Layers Parameter Shift for H_2

Error for k-UpCCGSD 1 layer Parameter Shift for H_2

Error for Single & Double Excitations Parameter Shift for H_2

Error for Double Excitations Parameter Shift for H_2

Fig. 4-5: Error in log-scale for VQE analytic training using different optimizers. For all ansatz it was possible to train the circuit to reach error within chemical accuracy (shaded region). Notice how the errors in the bottom row using Excitations ansatz perform particularly well. ADAM outperforms the other optimizers, with gradient descend close enough and SPSA showing larger error for ansatz with many parameters.



Error for Simplified Two Design 4 Layers Finite Diff for H_2

Error for Simplified Two Design 4 Layers Parameter Shift for H_2

Error for k-UpCCGSD 1 layer Finite Diff for H_2

Error for k-UpCCGSD 1 layer Parameter Shift for H_2

Error for Single & Double Excitations Finite Diff for H_2

Error for Single & Double Excitations Parameter Shift for H_2

Error for Double Excitations Finite Diff for H_2

Error for Double Excitations Parameter Shift for H_2

Fig. 4-6: Error in log-scale for VQE shots-based training using different optimizers. The errors presented are the average of 5 complete runs of each configuration, using 10000 shots. Left column results use finite difference, and right column uses parameters shift rule. SPSA outperforms the other optimizers when they use finite difference, and ADAM provides good enough accuracy when using parameters shift.

5 The Meta-Variational Quantum Eigensolver

In the best scenario, with an initial state preparation and a physically inspired circuit, a VQE execution only returns a specific ground-state energy, associated with the punctual geometry and other parameters used to calculate the Hamiltonian of the system. Parameters such as nuclear coordinates, external field strengths or model-specific parameters define the Hamiltonian, the cost function, and the resulting energy of the calculation. However, finding the ground energy for a specific configuration can be insufficient when the goal is to find the configuration that leads to specific properties of the system. In these situations, the potential energy surfaces are of great utility, but they come with the computational cost of running many instances of the VQE to scan over the parameters. That is the reason why some works have explored the possibility of reducing this problem by making the circuit 'learn' the Hamiltonian.

The *Meta-Variational Quantum Eigensolver* (meta-VQE) [16] encodes the circuit parameters using the Hamiltonian meta-parameters as free variables. Then, the meta-VQE is trained using a small set of Hamiltonian meta-parameters by setting a cost function that is a combination of all expected values. Finally, the meta-VQE has learned the optimal values used to encode the meta-parameters, and then other Hamiltonian meta-parameters can be used to obtain good estimations of the ground-state energy without further optimizations. This concept is portrayed in figure **5-1**, with the left panel showing the results of a standard VQE procedure, and the right panel depicting multiple training points (Hamiltonian meta-parameters) used to learn the optimal values that encode the circuit parameters. If the estimation is not precise enough, the resulting parameters of meta-VQE can be used as a starting point to running VQE (opt-meta-VQE) for all the Hamiltonian meta-parameters.

5.1. General Description

Given an n qubits parametrized Hamiltonian of the form $H = H(\vec{\lambda})$ where $\vec{\lambda} = \{\lambda_1, ..., \lambda_q\}$ are q different meta-parameters, M sets of $\vec{\lambda}$ are selected as the training set. Notice how the set of meta-parameters is closely related to the problem's complexity: it might be just the distance R between the two nuclei of a single molecule, or a group of variables encoding a complex geometry. Then, the unitary VQC $U(\vec{\theta})$ is expresed using an encoding unitary operator $\mathcal{S}$ which arguments include the Hamiltonian meta-parameters $\vec{\lambda}_i$ and extra variational

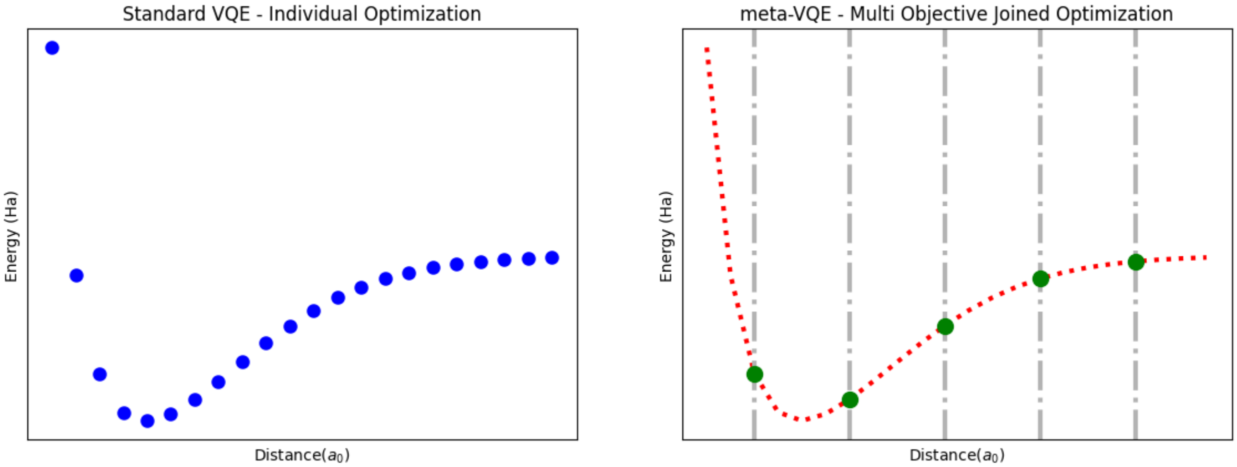


Fig. 5-1: Layout for meta-VQE. Using a set of training points (interatomic distances for a molecular system, represented in the plot as vertical lines) it is possible to learn the energy profile of the system and obtain energy values for other points.

parameters $\vec{\Phi}$. This means that all the variables $\vec{\theta}$ that rule the effect of the circuit over the inital state are now encoded as

$$\vec{\theta} = \vec{\theta}(\vec{\lambda}_i, \vec{\Phi}). \tag{5-1}$$

The final state of the circuit takes the form

$$|\psi_i\rangle = \mathcal{S}(\vec{\lambda}_i, \vec{\Phi})\, |\psi_0\rangle \,, \tag{5-2}$$

where $|\psi_0\rangle$ is the initial state of the circuit. The meta-VQE is optimized over a set of training points $\vec{\lambda}_i$ by minimizing a cost function that depends on a mixture of the energies $H(\vec{\lambda}_i)$ calculated over all training points. The simplest cost function is the average of all expected values

$$C(\vec{\lambda}, \vec{\Phi}) = \sum_{i=1}^{M} \frac{1}{M} \left\langle \psi_i(\vec{\lambda}_i, \vec{\Phi}) \right| H(\vec{\lambda}_i) \left| \psi_i(\vec{\lambda}_i, \vec{\Phi}) \right\rangle , \tag{5-3}$$

but other more sophisticated aggregations of cost functions can be used (e.g. aggregations inspired on the problem at hand). Once the circuit is trained, the expected value of the Hamiltonian can be calculated for other values of $\vec{\lambda}$ using the trained parameters $\vec{\theta}_{opt} = \vec{\theta}(\vec{\lambda}, \vec{\Phi}_{opt})$. Notice how this procedure can be described as an multi-objective optimization (MOO) [45] where a collection of functions are minimized over the same set of designed variables.

For a simple molecule, the meta-parameter to use is usually the nuclei separation R. The encoding of the circuit parameters $\vec{\theta}$ done by the unitary $\mathcal{S}$ used for meta-VQE can be done using a linear function

$$\theta_i = w_i\, R + \phi_i, \tag{5-4}$$

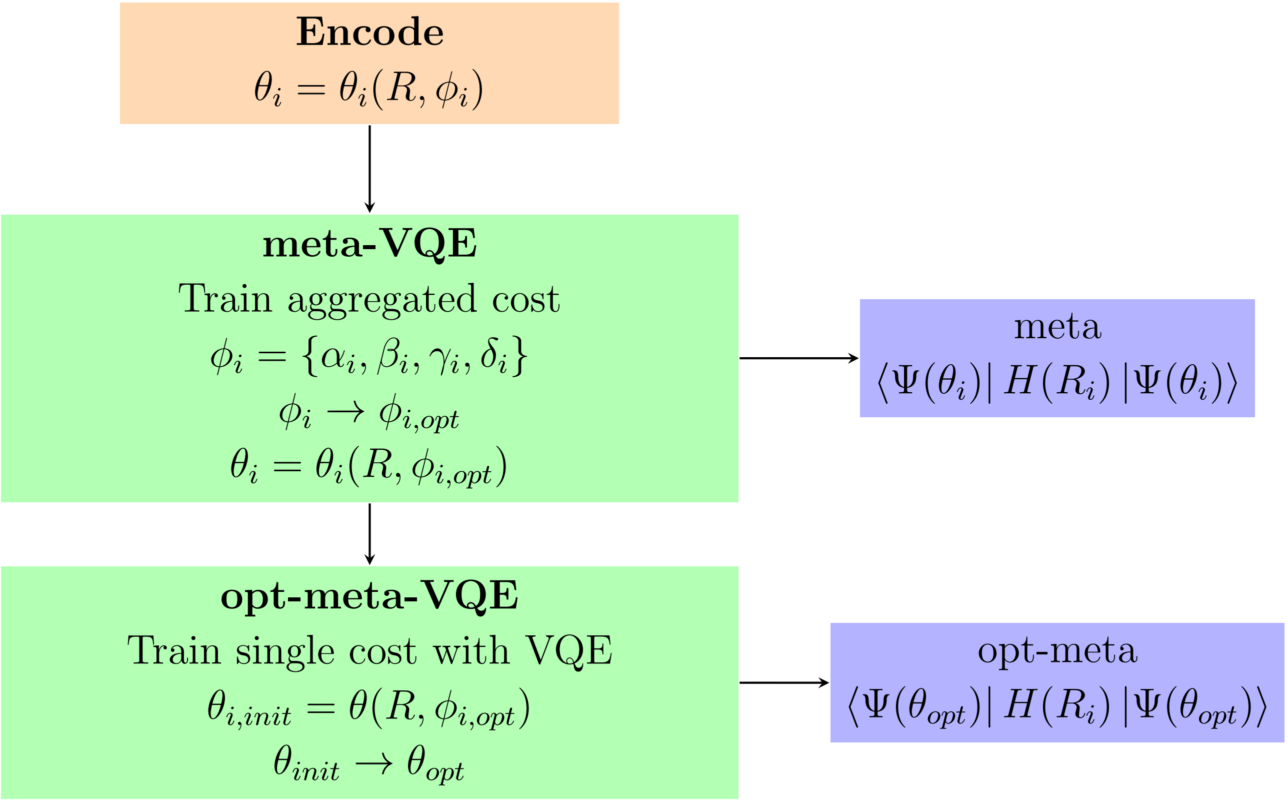

Encode
$\theta_i = \theta_i(R, \phi_i)$
meta-VQE
Train aggregated cost
$\phi_i = \{\alpha_i, \beta_i, \gamma_i, \delta_i\}$
$\phi_i \rightarrow \phi_{i,opt}$
$\theta_i = \theta_i(R, \phi_{i,opt})$
meta
$\langle\Psi(\theta_i)| H(R_i) |\Psi(\theta_i)\rangle$
opt-meta-VQE
Train single cost with VQE
$\theta_{i,init} = \theta(R, \phi_{i,opt})$
$\theta_{init} \rightarrow \theta_{opt}$
opt-meta
$\langle\Psi(\theta_{opt})| H(R_i) |\Psi(\theta_{opt})\rangle$

Fig. 5-2: Diagram of meta-VQE and opt-meta-VQE. After the encoding with the interatomic distance as a free parameter is defined, the encoding parameters are optimized by training the aggregated cost function for a set of interatomic distances. Then, with the encoding and its optimized parameters, optimal circuit parameters are obtained for any interatomic distance. The process can be understood as a statistical regresion where the encoding parameters are optimized to minimize the aggregated cost function.

with individual variational parameters w_i and ϕ_i, or more complex non-linear encoding such as a Gaussian function

$$\theta_i = \alpha_i e^{\beta_i(\gamma_i - R)} + \delta_i, \tag{5-5}$$

with individual variational parameters α_i, β_i, γ_i and δ_i. Gaussian functions are fairly good encoders to use for approximations, and have the additional advantage of capturing the asymptotic behaviour of molecular distances over the constant offset δ. Figure **5-2** depicts the procedure of meta-VQE and opt-meta-VQE, where encoding parameters are optimized to obtain the optimal values of the circuit variables.

5.2. Meta-VQE results with multiple configurations for H_2

As an example for meta-VQE, a H_2 molecule is analysed using different ansatz. The Hamiltonian was calculated using a STO-3G basis and a Jordan-Wigner encoding. The metaparameter of the Hamiltonian is the interatomic distance of the atoms. Remember that the position of the atoms, affected by the change in the interatomic distance, also affect the one- and two-electron integrals, which makes it necessary to calculate a Hamiltonian for each separation value considered. For training points, an equally spaced set of interatomic distances were selected: 1.0 a_0, 1.8 a_0, 2.6 a_0, 3.4 a_0, 4.2 a_0. There is no restriction or rule of thumb to select these points. However, it is recommended that they surround the inflection points of the physical quantity under analysis.

The ansatz used will be the same as in the previous chapter: Simplified Two Design, Combinations of Singles and Double excitations, Just Double excitations and k-UpCCGSD. Some metrics for circuit comparison can be found in table **5-1**. As with other calculations, the following were also made using PennyLane considering an analytic and a shots-based approach with 10000 shots. The circuit's parameters were encoded with the Gaussian function in equation 5-5, and the initial values of each variable in the encoding were initialized at random.

Table 5-1: Cost estimates for circuits used in meta-VQE and opt-meta-VQE. Number of parameters, number of two-qubit operators (entangling gates), and circuit depth. Notice that for H_2 with 4 qubits, result of using the STO-3G basis, there is two single and one double excitations.

Ansatz	Depth	Parameters	Entangling Gates
Simplified Two Design - 4 layers	17	28	12
2 Single and 1 Double Excitation	32	3	22
1 Double Excitation	20	1	14
k-UpCCGSD - 1 layer	92	6	72

For meta-VQE, there is a single aggregated cost function that is a combination of the costs produced by each training point. The aggregation used was the *average* of the expected value of the Hamiltonian at each point. The initial values of the Gaussian encoding parameters, can be set at random, or in such a way that the initial values produced are the same as the linear encoding ($\alpha, \delta = 0$, $\beta, \gamma = 1$). After meta-VQE optimization procedure, the results are the optimal parameters used to encode the interatomic distance. For example, using the Gaussian encoding in the k-UpCCGSD ansatz with 1 layer, there are 6 parameters for the ansatz and $6 \times 4 = 24$ parameters to train in meta-VQE. After the optimization, these 24 values are plugged in in the equation 5-5 with an inter atomic distance R as a free variable. This would produce 6 new variables that depend explicitly on R, and finally these values are used in the original circuit to calculate the ground energy for any distance within a certain range.

5.2.1. Simulation details

Here you can find the details of the simulations done to calculate the ground energy for a H_2 molecule using meta-VQE.

- The electronic wavefunction is described using LCAO of lowest energy states S_1, and the chemical basis set STO-3G.
- The integrals required to get the molecule hamiltonian are calculated using the PennyLane wrapper of PySCF.
- 150 steps max in each training method are used.
- For aggregated cost function used in meta-VQE, 5 train points are used: $\{R_i\} \in \{$ 1.0 a_0, 1.8 a_0, 2.6 a_0, 3.4 a_0, 4.2 $a_0\}$.
- FCI (Full Configuration Interaction) is used as comparison value for error calculation, using chemical accuracy (Error $<$ 1,6mHa) as baseline

5.2.2. Analytic Approach

The figure **5-4** depicts the evolution of this aggregated cost function for the ansatz, where the max number of repetitions were set at 150. All the plots use finite difference, as the selection of gradient method affects very slightly the results (just like standard VQE). An important thing to see is that all ansatz portrayed are able to learn, reducing the cost function of meta-VQE. The Hardware Efficient Ansatz is the one that starts with the highest cost, as expected due to not considering any information regarding the chemical problem at hand. However, all ansatz reach a joined close or lower to −1,0 Ha. Again, ADAM optimizer outperforms the other optimizers, with Gradient descent being a close competitor in Double Excitation and k-UpCCGSD circuits. SPSA performance is poor in all but one circuit (k-UpCCGSD).

After the learning process, and with good enough results, the calculations can be further improved by training all the test points using VQE, using the optimal circuit parameters for each atomic distance produced by the optimal encoding parameters in a ´´warm start". The best results of meta-VQE are obtained using a problem-inspired ansatz and ADAM optimizer (figure **5-5**), even being able to obtain the ground energy within chemical accuracy for most of the test points. For analytical calculations, there is not noticeable difference in the results when finite differences or parameters shift rule are used. Regarding the other circuits, good results are obtained in general for the problem inspired ansatz, with errors of order 10^2 Ha or lower. A detailed comparison of the results can be seen in the table **5-2**, which contains the root mean squared error (RMSE) of the energies obtained by variational methods with the FCI energies.

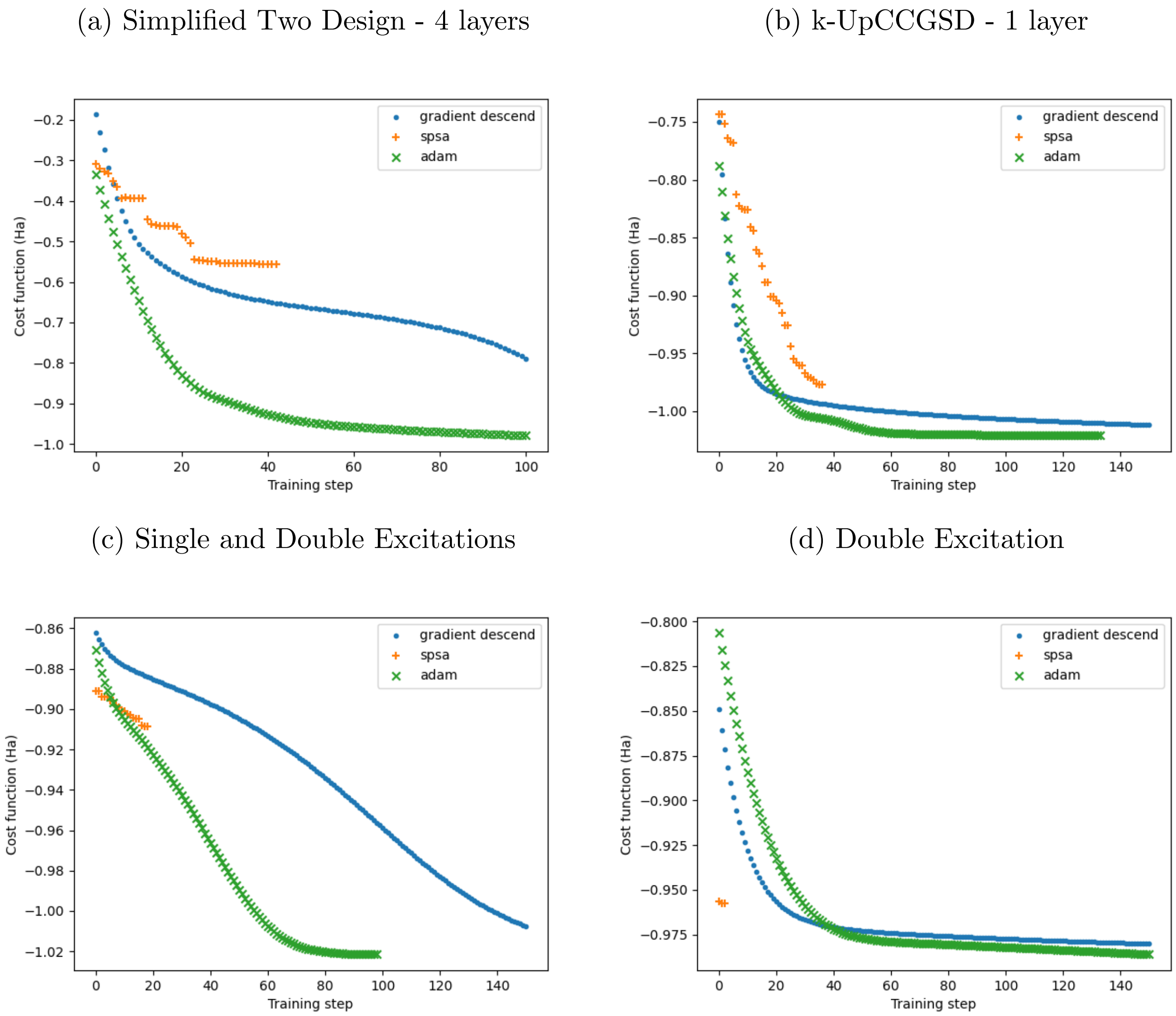
In principle, meta-VQE pursues the reduction of computational cost when analysing the potential energy surface of a molecule. To prove if there is an actual reduction compared to VQE, it is necessary to compare the number of runs of the quantum circuit. For this, it is not considered the (unlikely) possibility that the Hamiltonian might have a variable number of terms for different intermolecular distances, which might affect the time required to obtain the expectation value. For each optimizer step and each parameter to train in VQE, it is required to run the circuit twice (considering the gradient strategy is a simple finite difference approximation or simple parameter shift rule). In meta-VQE this value is multiplied by the number of training points used, but also by the number of variables used to encode the ansatz. For the Gaussian encoding of equation 5-5 that uses 4 variables, using 5 training point would require $2 \times 4 \times 5 = 40$ runs per ansatz parameter and optimizer step. This is an overhead of 20. So, in principle, meta-VQE becomes cheaper if more than 20 points are required.

5.2.3. Shots-based approach using 10000 shots

Getting the statistics of the systems based on a defined number of shots changes completely the results of the training. The RMSE errors are shown in table **5-3**. For this approach, it is noticeable how the variational algorithm struggles to minimize the meta-VQE cost function. It is also clear how the gradient method used makes a difference in the training of cost function. Parameter-shift rule clearly improves the performance in all VQE, meta-VQE and opt-meta-VQE. Also, it is interesting to notice how SPSA outperform the training of the other gradient based methods when finite differences are used. This may be explained by the stochastic property of SPSA, which helps to reduce the effect of the variability of the samples obtained.

Table 5-2: RMSE error of VQE, meta-VQE and opt-meta-vqe for multiple ansatz and gradient methods using analytical calculation, in mHa. The term “fd” reefers to finite difference, and “ps” to parameters shift rule. The selection of the differentiation method does not make any significant difference in the calculated errors. ADAM optimizer gives better results for meta-VQE, although Gradient descend performance is also good. About the ansatz, problem-inspired circuits produce the smallest error.

		Gradient		SPSA	ADAM	
		fd	ps		fd	ps
Simplified Two Design - 4 layers	vqe-previous	2.668	2.668	229.307	3.683	3.683
	meta	254.312	254.311	626.152	58.264	58.264
	opt-meta	34.785	34.785	52.414	19.477	19.477
Single and Double Excitations	vqe-previous	0.123	0.123	1.574	0.036	0.036
	meta	29.925	29.925	146.567	5.015	5.015
	opt-meta	0.163	0.163	2.120	0.171	0.171
Double Excitations	vqe-previous	0.122	0.122	0.734	0.043	0.043
	meta	67.518	67.518	117.082	57.753	57.753
	opt-meta	0.113	0.113	0.928	0.080	0.080
k-UpCCGSD - 1 layer	vqe-previous	0.056	0.056	33.185	0.701	0.104
	meta	19.840	19.840	68.734	7.076	7.076
	opt-meta	0.055	0.055	1.519	0.144	0.546



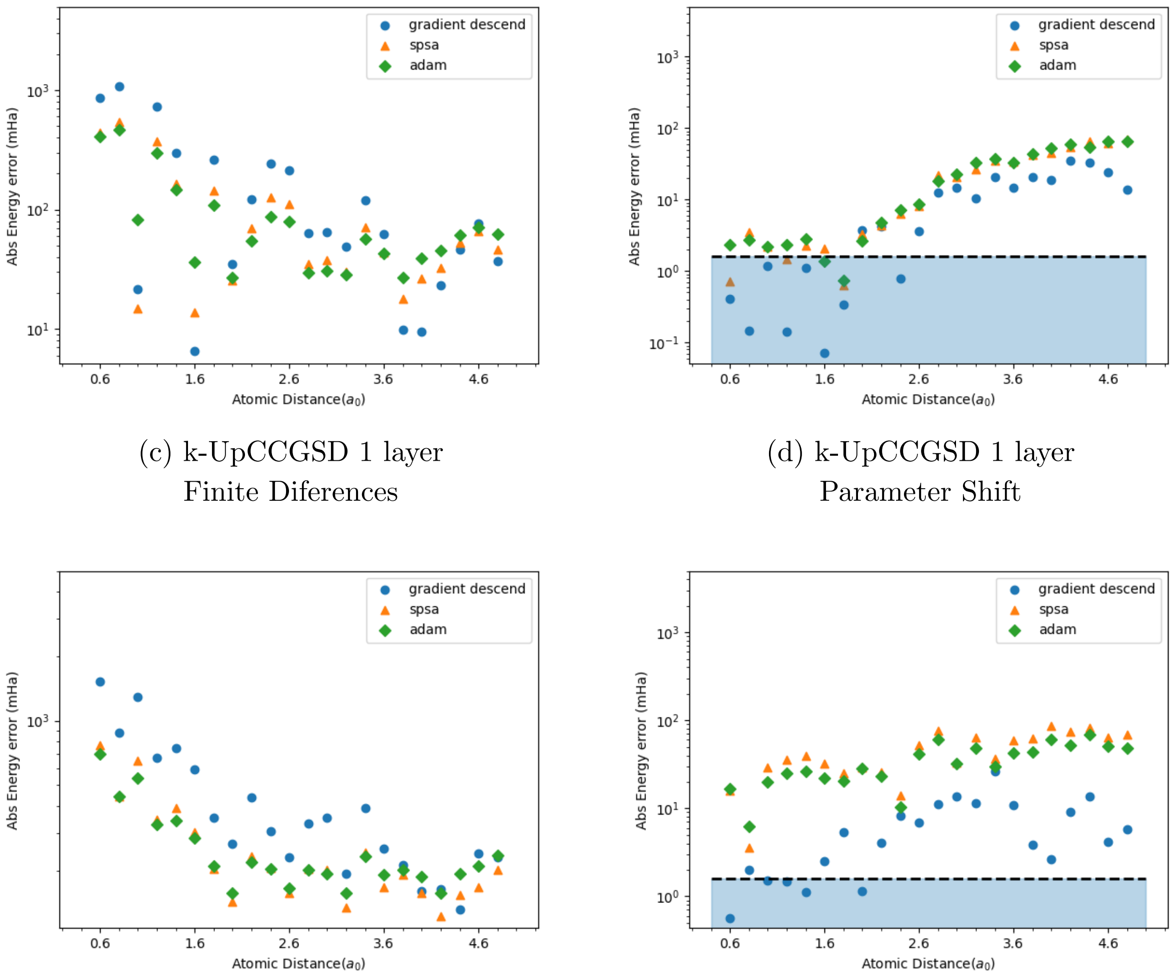
(a) Simplified Two Design - 4 layers

(b) k-UpCCGSD - 1 layer

(c) Single and Double Excitations

(d) Double Excitation

Fig. 5-3: Meta-VQE cost function for training step using parameter shift rule. The average of all expected values of the Hamiltonian at each test point was optimized for each circuit. Both Hardware-Efficient and problem-inspired ansatz can learn the energy profile (given enough layers).



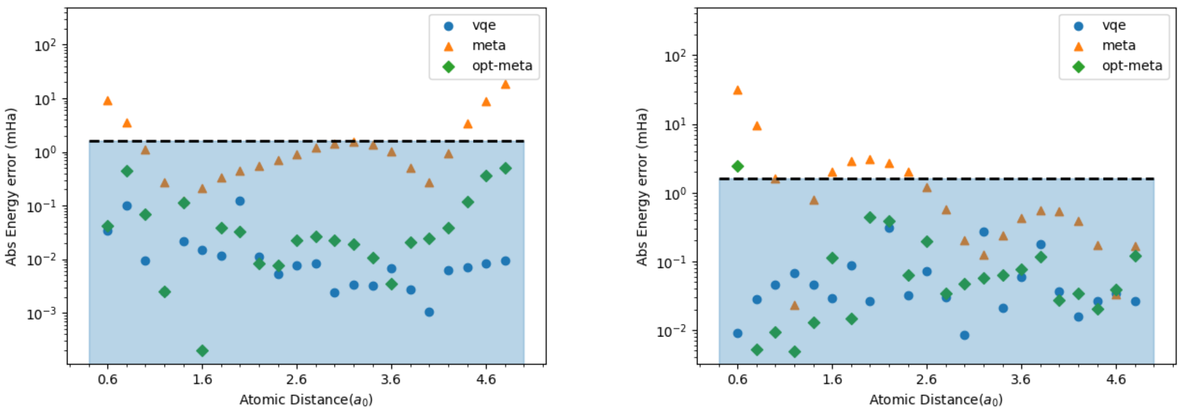
(a) Double Excitation
Finite Diferences

(b) Double Excitation
Parameter Shift

(c) k-UpCCGSD 1 layer
Finite Diferences

(d) k-UpCCGSD 1 layer
Parameter Shift

Fig. 5-4: Meta-VQE trained energies for different optimizers using k-UpCCGSD and Double Excitation ansatz.



(a) Single and Double Excitations

(b) k-UpCCGSD - 1 layer

Fig. 5-5: Error in log-scale for VQE, meta-VQE and opt-meta-VQE using analytical calculations, parameter shift rule and ADAM optimizer. The results of meta-VQE are within chemical accuracy for most of the distances considered. A further training of the circuit parameters coming from meta-VQE show error in the order of Standard VQE.

Table 5-3: RMSE error of VQE, meta-VQE and opt-meta-vqe for multiple ansatz and gradient methods using shots-based approach, in mHa. The results are the average of 5 complete runs of each configuration, using 10000 shots. The term “fd” reefers to finite difference, and “ps” to parameters shift rule. With this approach, the differentiation method does affect the results, with parameter shift rule being the method with the bests results. Also, when using shots, SPSA appears to perform better than other optimizers while using finite differences.

		Gradient		SPSA	ADAM	
		fd	ps		fd	ps
Simplified Two Design - 4 layers	vqe	854.324	54.734	181.795	673.237	5.695
	meta	863.937	846.799	632.235	816.570	833.424
	opt-meta	812.076	168.150	371.659	792.859	98.876
Single and Double Excitations	vqe	595.586	4.873	12.187	109.237	26.615
	meta	610.868	282.837	180.910	201.429	268.976
	opt-meta	654.683	57.206	96.391	185.635	94.520
Double Excitations	vqe	602.135	3.436	8.297	87.357	1.355
	meta	172.344	73.782	159.681	197.089	69.279
	opt-meta	554.103	18.614	33.831	139.945	38.397
k-UpCCGSD - 1 layer	vqe	634.205	1.361	4.222	82.542	0.990
	meta	596.225	354.612	147.007	323.514	438.432
	opt-meta	636.187	10.966	103.582	281.916	24.317

6 Conclusions

This work presents a detailed description of Variational Quantum Circuits used to obtain the ground energy of a molecular system. Starting with a small description of quantum mechanics, followed by a discussion on how to write a Molecular Hamiltonian in a qubit basis to be used by a quantum processor, and ending with a comprehensive discussion on how to build a Variational Quantum Circuit to implement the Variational Quantum Eigensolver method and meta-Variational Quantum Eigensolver.

In particular, it was shown how VQE is able to find the ground state and obtain the potential energy surface of an H_2 molecule, executing analytics and shots-based simulations. For both approaches, results within chemical accuracy are obtained using ADAM optimizer, problem-inspired ansatz composed by Single and Double excitations and Coupled Cluster methods, and parameter-shift rule as a gradient method for training. Regarding meta-VQE and further optimization from meta-VQE results (opt-meta-VQE), there was no advantage proven against standard VQE, which might be caused by a very simple definition of the meta-VQE joined cost function or some required fine tuning of the optimizer due to the change in nature of the cost function. Some things to highlight is that the differentiation methods do not play an important role when analytical simulations are considered. Energies and errors are almost the same, regardless of using finite differences or parameter-shift rule. This is not the case for the shots-based approach, where parameter-shift rule clearly has a better performance. Also, SPSA appeared to perform as well or better than the other optimizers when the shots based approach was used. This is important as the real use of quantum devices is shots based.

Possible perspectives of this work include the consideration of refined meta-VQE costs functions to achieve good performance in the learning process, while also reducing the computational cost against the standard VQE training. Also, it would be interesting to see how the training performance of meta-VQE using ADAM and SPSA is impacted by changing the number of shots. Finally, it is necessary to address the impact the error has in the presented training procedures. This could the achieved considering error models from real qiskit devices, or even perform the training in a real quantum processor using Qiskit Runtime.

This work has laid down the grounds for multiple, more sophisticated projects involving larger molecules, calculation of other molecular properties (such as excited state energies, dipole moments, charge distributions, to name a few), and the possibility of analysing exotic molecules composed by antimatter particles. In particular, QML is well suited to accelerate the identification of energy stability in molecules composed by antimatter particles.

References

[1] The quantum state of affairs. *Nature Physics*, 19(5):605–605, May 2023.

[2] A. Anand, P. Schleich, S. Alperin-Lea, P. W. Jensen, S. Sim, M. Díaz-Tinoco, J. S. Kottmann, M. Degroote, A. F. Izmaylov, and A. Aspuru-Guzik. A quantum computing view on unitary coupled cluster theory. *Chemical Society Reviews*, 2022.

[3] G.-L. R. Anselmetti, D. Wierichs, C. Gogolin, and R. M. Parrish. Local, expressive, quantum-number-preserving vqe ansätze for fermionic systems. *New Journal of Physics*, 23(11):113010, 2021.

[4] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles. Operator sampling for shot-frugal optimization in variational algorithms. *arXiv preprint arXiv:2004.06252*, 2020.

[5] J. M. Arrazola, V. Bergholm, K. Brádler, T. R. Bromley, M. J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L. G. Helt, et al. Quantum circuits with many photons on a programmable nanophotonic chip. *Nature*, 591(7848):54–60, 2021.

[6] J. M. Arrazola, O. Di Matteo, N. Quesada, S. Jahangiri, A. Delgado, and N. Killoran. Universal quantum circuits for quantum chemistry. *Quantum*, 6:742, 2022.

[7] J. M. Arrazola, S. Jahangiri, A. Delgado, J. Ceroni, J. Izaac, A. Száva, U. Azad, R. A. Lang, Z. Niu, O. Di Matteo, et al. Differentiable quantum computational chemistry with pennylane. *arXiv preprint arXiv:2111.09967*, 2021.

[8] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[9] R. J. Bartlett and M. Musiał. Coupled-cluster theory in quantum chemistry. *Reviews of Modern Physics*, 79(1):291, 2007.

[10] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[11] X. Bonet-Monroig, H. Wang, D. Vermetten, B. Senjean, C. Moussa, T. Bäck, V. Dunjko, and T. E. O'Brien. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A*, 107(3):032407, 2023.

[12] S. Buchholz, D. Golden, and C. Brown. A business leader's guide to quantum technology — www2.deloitte.com. `https://www2.deloitte.com/us/en/insights/topics/innovation/quantum-computing-business-applications.html`, 2021. [Accessed 21-Jul-2023].

[13] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19):10856–10915, 2019.

[14] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[15] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature communications*, 12(1):1791, 2021.

[16] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik. Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation. *PRX Quantum*, 2(2):020329, 2021.

[17] J. Charry, M. T. d. N. Varella, and A. Reyes. Binding matter with antimatter: the covalent positron bond. *Angewandte Chemie International Edition*, 57(29):8859–8864, 2018.

[18] J. Chen, C. Wolfe, Z. Li, and A. Kyrillidis. Demon: improved neural network training with momentum decay. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3958–3962. IEEE, 2022.

[19] B. Choy and D. J. Wales. Molecular energy landscapes of hardware-efficient ansätze in quantum computing. *Journal of Chemical Theory and Computation*, 19(4):1197–1206, 2023.

[20] D. Cremer. Møller–plesset perturbation theory: from small molecule methods to methods for thousands of atoms. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(4):509–530, 2011.

[21] G. E. Crooks. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition. *arXiv preprint arXiv:1905.13311*, 2019.

[22] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019.

[23] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller. Practical quantum advantage in quantum simulation. *Nature*, 607(7920):667–676, 2022.

[24] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik: Progress of Physics*, 48(9-11):771–783, 2000.

[25] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao. Quantum circuit architecture search for variational quantum algorithms. *npj Quantum Information*, 8(1):62, 2022.

[26] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[27] V. Dunjko and H. J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

[28] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev. Vqe method: a short survey and recent developments. *Materials Theory*, 6(1):1–21, 2022.

[29] R. P. Feynman et al. Simulating physics with computers. *Int. j. Theor. phys*, 21(6/7), 1982.

[30] P. Friederich, M. Krenn, I. Tamblyn, and A. Aspuru-Guzik. Scientific intuition inspired by machine learning-generated hypotheses. *Machine Learning: Science and Technology*, 2(2):025027, 2021.

[31] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[32] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1):3007, 2019.

[33] G. Herzberg. The dissociation energy of the hydrogen molecule. *Journal of molecular spectroscopy*, 33(1):147–168, 1970.

[34] J. R. Johansson, P. D. Nation, and F. Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[35] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.

[36] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023.

[37] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11:369–395, 2020.

[39] M. Krenn, J. Landgraf, T. Foesel, and F. Marquardt. Artificial intelligence and machine learning for quantum technologies. *Physical Review A*, 107(1):010101, 2023.

[40] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum*, 4:263, 2020.

[41] L. Leone, S. F. Oliviero, L. Cincio, and M. Cerezo. On the practical usefulness of the hardware efficient ansatz. *arXiv preprint arXiv:2211.01477*, 2022.

[42] J. Li, B. A. Jones, and S. Kais. Toward perturbation theory methods on a quantum computer. *Science Advances*, 9(19):eadg4576, 2023.

[43] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.

[44] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[45] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26:369–395, 2004.

[46] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, 2020.

[47] A. McCaskey, E. Dumitrescu, D. Liakh, and T. Humble. Hybrid programming for near-term quantum computing systems. In *2018 IEEE international conference on rebooting computing (ICRC)*, pages 1–12. IEEE, 2018.

[48] M. Medvidović and G. Carleo. Classical variational simulation of the quantum approximate optimization algorithm. *npj Quantum Information*, 7(1):101, 2021.

[49] A. A. Melnikov, H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel. Active learning machine learns to create new quantum experiments. *Proceedings of the National Academy of Sciences*, 115(6):1221–1226, 2018.

[50] A. Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8, 2016.

[51] A. Morvan, B. Villalonga, X. Mi, S. Mandrà, A. Bengtsson, P. Klimov, Z. Chen, S. Hong, C. Erickson, I. Drozdov, et al. Phase transition in random circuit sampling. *arXiv e-prints*, pages arXiv–2304, 2023.

[52] M. Motta and J. E. Rice. Emerging quantum computing algorithms for quantum chemistry. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(3):e1580, 2022.

[53] M. A. Nielsen and I. Chuang. *Quantum computation and quantum information.* American Association of Physics Teachers, 2002.

[54] M. Ostaszewski, E. Grant, and M. Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, 2021.

[55] M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko. Reinforcement learning for optimization of variational quantum circuit architectures. *Advances in Neural Information Processing Systems*, 34:18182–18194, 2021.

[56] F. Pavošević and S. Hammes-Schiffer. Multicomponent equation-of-motion coupled cluster singles and doubles: Theory and calculation of excitation energies for positronium hydride. *The Journal of Chemical Physics*, 150(16), 2019.

[57] F. Pavosevic and S. Hammes-Schiffer. Multicomponent unitary coupled cluster and equation-of-motion for quantum computation. *Journal of Chemical Theory and Computation*, 17(6):3252–3258, 2021.

[58] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

[59] K. Phalak and S. Ghosh. Shot optimization in quantum machine learning architectures to accelerate training. *IEEE Access*, 2023.

[60] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, et al. Compact ion-trap quantum computing demonstrator. *PRX Quantum*, 2(2):020343, 2021.

[61] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[62] B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus. New basis set exchange: An open, up-to-date resource for the molecular sciences community. *Journal of chemical information and modeling*, 59(11):4814–4820, 2019.

[63] J. Ruane, A. McAfee, and W. D. Oliver. Quantum Computing for Business Leaders — hbr.org. `https://hbr.org/2022/01/quantum-computing-for-business-leaders`, 2022. [Accessed 21-Jul-2023].

[64] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[65] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[66] M. Schuld and N. Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[67] M. Schuld and F. Petruccione. *Machine learning with quantum computers.* Springer, 2021.

[68] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[69] S. Sim, P. D. Johnson, and A. Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[70] I. O. Sokolov, P. K. Barkoutsos, P. J. Ollitrault, D. Greenberg, J. Rice, M. Pistoia, and I. Tavernelli. Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents? *The Journal of chemical physics*, 152(12):124107, 2020.

[71] J. C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on aerospace and electronic systems*, 34(3):817–823, 1998.

[72] J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest*, 19(4):482–492, 1998.

[73] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[74] A. Wack, H. Paik, A. Javadi-Abhari, P. Jurcevic, I. Faro, J. M. Gambetta, and B. R. Johnson. Quality, speed, and scale: three key attributes to measure the performance of near-term quantum computers. *arXiv preprint arXiv:2110.14108*, 2021.

[75] D. Wecker, M. B. Hastings, and M. Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.

[76] M. Wiedmann, M. Hölle, M. Periyasamy, N. Meyer, C. Ufrecht, D. D. Scherer, A. Plinge, and C. Mutschler. An empirical comparison of optimizers for quantum machine learning with spsa-based gradients. *arXiv preprint arXiv:2305.00224*, 2023.