TABRIZ UNIVERSITY

DEPARTMENT OF PHYSICS

Variational Quantum Eigensolver: Basics and Implementation

Author: Mehdi Rahmani Zadeh

Supervisor: Mahmoud Mahdian

August 15, 2024

Bachelor's Project

Bachelor of Physics

Contents

English Summary

The Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm designed to find the ground-state energy of quantum systems. It leverages the strengths of both quantum and classical computing to solve problems that are otherwise intractable for classical algorithms. This makes it particularly useful in fields like quantum chemistry and materials science.

The VQE works by parameterizing a trial wavefunction on a quantum computer and iteratively adjusting these parameters using a classical optimizer to minimize the expected energy value. This hybrid method combines the quantum computer's efficiency in representing complex wavefunctions with the classical computer's optimization techniques. The algorithm begins with an initial guess for the wavefunction and improves it through a feedback loop between quantum and classical processors.

This method extends the capabilities of quantum computing for solving complex quantum systems, pushing the limits of what is possible with current quantum technologies.

Preface

This report presents my bachelor's project completed during the fourth year of my physics degree. It was prepared using the master thesis LaTeX template "IT University Copenhagen" by Tony Beltramelli, available on `www.sharelatex.com`. While the methods outlined in this work are theorecital concepts, the focus here is on practical implementations.

A foundational understanding of physics should suffice to follow the content of this report. Appendix A includes most of the Python code used, with clear annotations. While familiarity with Python, Qiskit, and the NumPy package is beneficial, a physics undergraduate should be able to grasp the key points.

Chapter 1

Introduction

The Variational Quantum Eigensolver (VQE) uses the variational principle to determine the lowest energy state of a Hamiltonian, an important idea in quantum chemistry and condensed matter physics. Conventional computing techniques have difficulty in accurately simulating the rapidly increasing electronic wavefunction, but variational algorithms such as VQE present a hopeful solution by doing this efficiently in polynomial time. VQE has shown that it is resistant to quantum hardware noise, making it a promising option for quantum applications in the near future. Although the practical benefits are frequently debated, there is no consensus in the literature on this topic.

VQE operates by putting a trial wavefunction on a quantum computer and using a classical optimizer to continuously change the parameters in order to reduce the energy expectation value. This blend of methods brings together the quantum computer's capability to represent intricate wavefunctions and the classical computer's power in optimization. Beginning with a first estimate, the wavefunction is improved through an iterative process involving quantum and classical processors, allowing VQE to address intricate quantum systems and enhance the capabilities of existing quantum technology.

In recent years, quantum computing has seen fast progress. While still in its early phases, this advancement has resulted in the development of Noisy Intermediate-Scale Quantum (NISQ) devices, which are anticipated to have a small qubit count in the near future. These devices have already surpassed traditional computers on particular problems tailored for quantum abilities. Nevertheless, NISQ algorithms frequently combine both quantum hardware and classical computers in their execution. Because of restrictions in hardware, the quantity of quantum gates needs to be controlled to prevent errors and decoherence. Therefore, algorithms such as Shor's or Grover's are still not practical. However, VQE is considered a promising NISQ algorithm, with potential uses in drug discovery, material science, and chemical engineering.

Qubits obey quantum mechanical principles in a similar manner to electronic wavefunctions. The superposition principle enables quantum computers to encode information efficiently, which would be exponentially expensive for traditional systems, with only a linear increase in the number of qubits needed. The main attraction of quantum computing lies in its ability to simulate and control quantum wavefunctions. Although VQE has been mostly used for studying electronic structures, it has demonstrated promise in various other fields such as high-energy physics, spectroscopy, periodic systems, and predicting photochemical reactions.

1.1 Structure of the VQE:

VQE begins by initializing a qubit register, followed by applying a quantum circuit to model the physics and entanglement of the electronic wavefunction. A quantum circuit consists of a series of quantum operations, known as its depth, which is defined by two components: (1) the structure, given by a sequence of quantum gates called the 'Ansatz', and (2) the parameters that govern the behavior of some of these gates. After applying the circuit, the qubit state represents a trial wavefunction.

The system's Hamiltonian is then measured relative to this wavefunction to estimate the energy. VQE works by adjusting the parameters of the ansatz to minimize the trial energy, which will always be higher than the true ground state energy due to the variational principle. For VQE to remain feasible, the number of quantum operations must be kept low, requiring a compact ansatz. However, a shallower ansatz may cover a smaller range of possible wavefunctions, potentially reducing the accuracy of the energy estimation.

While the ansatz design is key to VQE's potential advantage over traditional methods, other elements of the algorithm also impact its cost and practicality, which are discussed further below.

When studying a quantum system with VQE, the first step is constructing its Hamiltonian, which must be expressible as a sum of individual terms that scale slowly with system size. This is typical for electronic Hamiltonians, where the Coulomb interaction, a sum of two-body terms, scales polynomially. Flexibility in how the Hamiltonian is represented is critical, as it impacts the number of qubits, the ansatz depth, and the number of required measurements. After constructing the Hamiltonian, it must be translated into measurable operators (e.g., spin or Pauli operators), which also influences the ansatz depth and measurement count.

The chosen ansatz must be expressive enough to approximate the ground state wavefunction but not so complex that it leads to deep circuits or inefficient training. Selecting the right optimizer is crucial for improving convergence and reducing the overall cost of the algorithm. Since quantum measurements are stochastic, multiple repetitions are needed to reach a desired precision, with the number of measurements influenced by both precision and the number of operators in the Hamiltonian. Efficient measurement strategies are essential to control the cost of VQE.

To counteract quantum noise, error mitigation techniques can be employed, though they come with additional computational costs. Despite polynomial scaling, the number of required measurements can grow quickly, making the method impractical. Thus, research continues to focus on efficient measurement schemes, compact Hamiltonian representations, and joint measurement strategies for commuting observables to address this challenge.

Another approach to the measurement challenge in VQE lies in its vast potential for parallelization, as originally suggested by Peruzzo et al.[1] Despite this, the community has given little attention to efficient parallelization strategies and potential communication overheads.

A key issue in VQE is the barren plateau problem, where gradients of the parameters vanish exponentially as the number of qubits, ansatz expressibility, or entanglement increase, making optimization difficult for larger systems. While several methods have been proposed to address this, it remains unclear whether these solutions fully resolve the issue in VQE. Similarly, the complexity of the VQE optimization landscape, including local minima and non-convex features, is not well understood. Research by Bittel and Kliesch highlights the importance of understanding how optimizers converge, particularly as system size and entanglement grow.

The resilience of VQE to quantum noise, and how effectively it can be mitigated, is another open question. While variational algorithms can adapt to certain types of noise, the cost of error mitigation methods may outweigh their benefits. Further study is needed to fully understand how quantum noise impacts VQE and whether the errors can be predicted or controlled.

Despite these challenges, VQE remains a promising candidate for early implementation on NISQ devices. However, as hardware advances, so must the theoretical foundations, software efficiency, and robustness of the algorithm to ensure its full potential is realized as early as possible.

Chapter 2

Overview of the VQE

This section provides a technical understanding of VQE, its position relative to classical electronic structure methods and other quantum algorithms, along with best practices and the resources needed to achieve quantum advantage.

2.1 Formal Definition of VQE:

VQE is based on the variational principle, which optimizes an upper bound for the ground state energy of a Hamiltonian $\hat{H}$. Given a trial wavefunction $|\psi\rangle$, the ground state energy E_0 is bounded by:

$$E_0 \leq \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}. \tag{2.1}$$

VQE aims to find a parameterized trial wavefunction that minimizes this energy expectation value. The goal is to approximate the eigenvector of $\hat{H}$ corresponding to the lowest eigenvalue E_0. To perform this on a quantum computer, we use a parameterized unitary operation $U(\theta)$ applied to an initial qubit state, typically . The VQE optimization problem can be written as:

$$E_{\text{VQE}} = \min_{\theta} \langle 0|U^{\dagger}(\theta)HU(\theta)|0\rangle \tag{2.2}$$

This is the cost function of VQE. The Hamiltonian is rewritten as a sum of Pauli strings, $\hat{P}_a \in \{I, X, Y, Z\}^{\otimes N}$, with N the number of qubits used to model the wavefuntion:

$$\hat{H} = \sum_{a}^{p} \omega_a \hat{P}_a \tag{2.3}$$

With ω_a set of weights, and P the number of Pauli strings in the Hamiltonian. Equation 2.2 becomes:

$$E_{\text{VQE}} = \min_{\theta} \sum_{a} \omega_a \langle 0|U^{\dagger}(\theta)P_a U(\theta)|0\rangle \tag{2.4}$$

Each term E_{P_a}represents the expectation value of a Pauli string, computed on the quantum device, while the summation and minimization are handled on a classical computer. This hybrid nature defines the VQE.

2.2 The VQE pipeline

The VQE algorithm, as defined by Equation 2.4, is composed of several components, each involving key choices that influence the design and cost of the algorithm. This sequence of components forms the VQE pipeline, where decisions at any stage impact the entire process. Figure 1 provides a visual summary of the algorithm's iterative loop and its main components.[1] [2] Below is an overview of the key steps:

2.2.1 Hamiltonian Construction and Representation:

The first step in VQE is defining the quantum system to find its ground state. This could range from molecular Hamiltonians for electronic structure to spin lattice models or nuclear systems. The system's geometry, such as atomic distances or lattice arrangement, must be specified. Constructing the Hamiltonian involves identifying the relevant operators and their weights between basis functions, representing the single-particle degrees of freedom. The choice of basis functions significantly affects both the computation size and result accuracy. For electronic structure problems, common basis representations include molecular orbitals, plane-wave functions, or local atomic functions, each describing spatial distributions of particles.

For systems involving electrons, the wavefunction must obey the Pauli exclusion principle, meaning it must be antisymmetric when swapping two electrons. This antisymmetry can be enforced either through the wavefunction's definition (first quantization) or the operators (second quantization). In second quantization, the Hamiltonian is written using fermionic creation $\hat{a}_j^\dagger$ and annihilation $\hat{a}_j$ operators, which add or remove an electron from a specific basis function while preserving antisymmetry between particles.

• **Encoding of Operators:** On quantum computers, qubit registers can only measure observables expressed in a Pauli basis $\hat{P}_a \in \{I, X, Y, Z\}^{\otimes N}$, because qubits function as two-level systems akin to spins. In **first quantization**, operators can be directly translated into Pauli operators without needing to enforce wavefunction antisymmetry. However, in **second quantization**, the Hamiltonian uses fermionic operators, which inherently obey antisymmetry (unlike Pauli operators). Therefore, a fermion-to-spin encoding must map fermionic operators into spin (Pauli) operators while maintaining the necessary antisymmetry.

The efficiency of an encoding depends on:

1. **Pauli weight:** The maximum number of non-identity elements in a spin operator.

2. **Number of qubits** required.
3. **Number of Pauli strings** produced.

The encoding can significantly affect the **gate depth** and the **trainability** of the VQE, especially for ansatzes defined with fermionic operators. For particles like **bosons**, which don't require antisymmetry, the process is simpler, and their Hamiltonians can be used directly.

• **Measurement Strategy and Grouping:** After the encoding, the next step is determining how to efficiently measure the expectation values from the trial wavefunction. To achieve precision ϵ in the measurement of an operator, $O(\frac{1}{\epsilon^2})$repetitions, or **shots**, are typically needed.

The goal is to minimize the number of repetitions. Several strategies can help with this:
1. **Efficient weighting** of the number of measurements per operator.
2. **Grouping operators** into sets that can be jointly measured by leveraging properties of the Lie algebra in which Pauli strings are defined. Commuting operators can be measured simultaneously, reducing the number of required measurements.

For joint measurements, a short quantum circuit is used to **rotate the measurement basis** for each group, enabling simultaneous measurement. Additionally, methods like **inference** from fewer shots, based on the overlap of information between Pauli strings, can further reduce the number of required measurements.

• **Ansatz and State Preparation:** Once the Hamiltonian is ready for measurement, the next step is preparing the trial wavefunction using an ansatz. The ansatz is a parametrized quantum circuit that generates the trial state, which allows the Hamiltonian to be measured. The goal is for the optimized ansatz to approximate the ground state wavefunction of the system.

The effectiveness of an ansatz depends on:
1. **Expressibility:** This measures how well the ansatz can represent a broad range of states in the Hilbert space. A highly expressive ansatz can better approximate the ground state but may also be more complex.
2. **Trainability:** This refers to how easily the ansatz parameters can be optimized. It depends on factors like the number of parameters, their interactions, the optimization landscape, and the presence of barren plateaus (regions where gradients vanish, making optimization difficult).

A well-chosen ansatz balances expressibility and trainability and considers the circuit depth's impact on noise resilience, especially important for near-term quantum computing.

• **Parameter Optimization:** Optimizing the ansatz parameters involves iteratively updating them based on the sampled expectation values of the Hamiltonian. This process requires multiple measurements for each parameter set to determine the update rules.

The choice of optimizer is crucial for several reasons:

1. **Measurement Efficiency:** The optimizer affects how many measurements are needed per optimization step. For example, numerical gradient methods require measuring the Hamiltonian with slightly altered wavefunctions.
2. **Optimization Challenges:** Some optimizers are designed to address specific issues, like the barren plateau problem, where gradients become too small for effective optimization.
3. **Convergence:** The optimizer influences how quickly and effectively convergence is achieved, or whether it can be achieved at all.

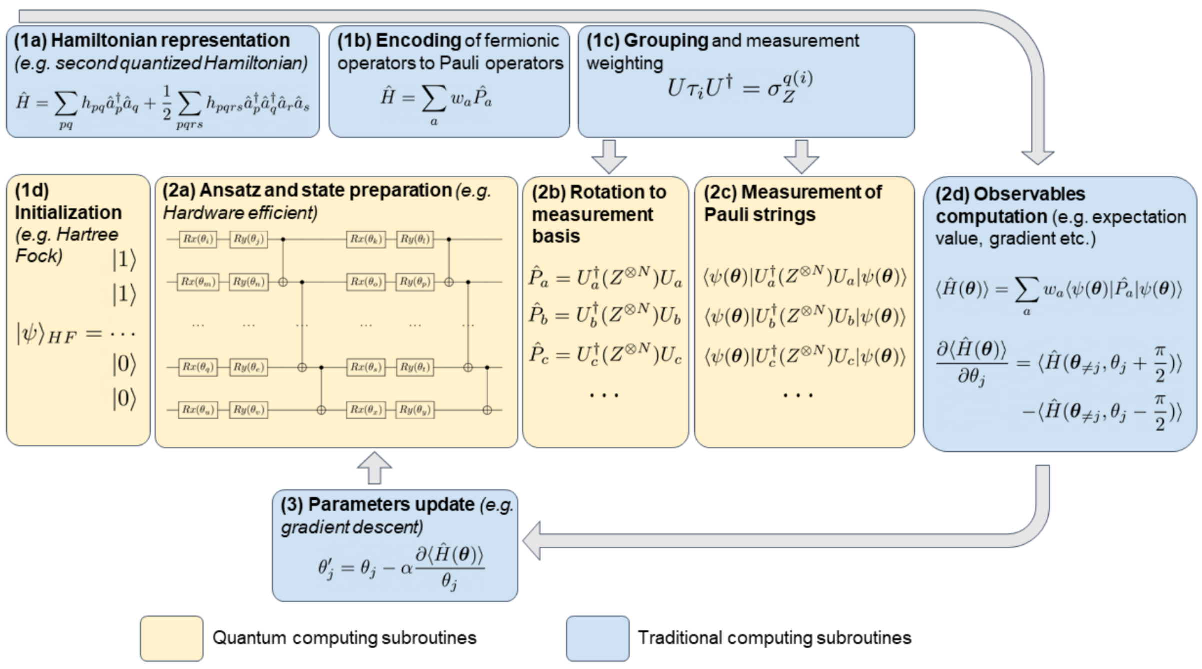

(1a) Hamiltonian representation (e.g. second quantized Hamiltonian)
$\hat{H} = \sum_{pq} h_{pq}\hat{a}_p^\dagger \hat{a}_q + \frac{1}{2}\sum_{pqrs} h_{pqrs}\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s$
(1b) Encoding of fermionic operators to Pauli operators
$\hat{H} = \sum_a w_a \hat{P}_a$
(1c) Grouping and measurement weighting
$U\tau_i U^\dagger = \sigma_Z^{q(i)}$
(1d) Initialization (e.g. Hartree Fock)
$|\psi\rangle_{HF} = $ $|1\rangle$ $|1\rangle$ $\cdots$ $|0\rangle$ $|0\rangle$
(2a) Ansatz and state preparation (e.g. Hardware efficient)
(2b) Rotation to measurement basis
$\hat{P}_a = U_a^\dagger (Z^{\otimes N}) U_a$
$\hat{P}_b = U_b^\dagger (Z^{\otimes N}) U_b$
$\hat{P}_c = U_c^\dagger (Z^{\otimes N}) U_c$
(2c) Measurement of Pauli strings
$\langle \psi(\boldsymbol{\theta}) | U_a^\dagger (Z^{\otimes N}) U_a | \psi(\boldsymbol{\theta}) \rangle$
$\langle \psi(\boldsymbol{\theta}) | U_b^\dagger (Z^{\otimes N}) U_b | \psi(\boldsymbol{\theta}) \rangle$
$\langle \psi(\boldsymbol{\theta}) | U_c^\dagger (Z^{\otimes N}) U_c | \psi(\boldsymbol{\theta}) \rangle$
(2d) Observables computation (e.g. expectation value, gradient etc.)
$\langle \hat{H}(\boldsymbol{\theta}) \rangle = \sum_a w_a \langle \psi(\boldsymbol{\theta}) | \hat{P}_a | \psi(\boldsymbol{\theta}) \rangle$
$\frac{\partial \langle \hat{H}(\boldsymbol{\theta}) \rangle}{\partial \theta_j} = \langle \hat{H}(\boldsymbol{\theta}_{\neq j}, \theta_j + \frac{\pi}{2}) \rangle - \langle \hat{H}(\boldsymbol{\theta}_{\neq j}, \theta_j - \frac{\pi}{2}) \rangle$
(3) Parameters update (e.g. gradient descent)
$\theta_j' = \theta_j - \alpha \frac{\partial \langle \hat{H}(\boldsymbol{\theta}) \rangle}{\theta_j}$
Quantum computing subroutines
Traditional computing subroutines

Figure 2.1: The VQE Pipeline

(1) Preprocessing:

- **Hamiltonian Representation:** Define a set of basis functions to express the Hamiltonian as a quantum observable of the electronic wavefunction.
- **Encoding:** Map the Hamiltonian's fermionic operators to spin operators for measurement on a quantum computer.
- **Grouping and Measurement Strategy:** Group operators for simultaneous measurement and determine the measurement weighting strategy. This may involve modifying the quantum circuit to rotate the measurement basis for each group.
- **State Initialization:** Initialize the qubit register, typically using the Hartree-Fock wavefunction.

(2) The VQE Loop:

- **Ansatz and Trial State Preparation:** Apply the ansatz to the initialized qubit register. Initialize all ansatz parameters before the first iteration.
- **Basis Rotation and Measurement:** Rotate the trial state into the measurement basis, often the Z basis, or a diagonal basis for specific Pauli strings.
- **Observable Computation:** Compute the expectation value, which is then summed and reconstituted on conventional hardware or with machine learning techniques.
- **Parameters Update:** Update the ansatz parameters based on computed observables and the optimization strategy, then begin a new iteration.

(3) Post-processing, Error Mitigation: Apply error mitigation techniques to measurement outputs or directly on the quantum state to reduce the impact of quantum noise.

• **Error Mitigation:** Quantum noise is a major challenge for VQE, especially on NISQ devices without error correction. Error mitigation techniques aim to reduce noise by post-processing measurement data or, in some cases, the trial wavefunction before measurements. These techniques vary in cost and effectiveness, and a balanced approach often combines several methods to achieve optimal results.

It's important to distinguish VQE from other variational quantum algorithms (VQAs). VQE is specifically designed to find the eigenstate of a quantum observable, unlike algorithms such as the Quantum Approximation Optimization Algorithm (QAOA) or the Variational Quantum Linear Solver (VQLS), where Hamiltonian encoding is not a central concern. While all VQAs benefit from efficient measurements, the nature of VQE's observables, which typically scale polynomially with system size, makes grouping and measurement strategies particularly impactful on performance.

Quantum Neural Networks (QNNs) in machine learning offer a related but distinct approach. While both QNNs and VQE are variational, QNNs focus on generalizing patterns from known data rather than solving specific problems like VQE. As a result, QNN pipelines prioritize encoding classical data into quantum states (quantum feature maps), rather than the observable representation or scaling challenges central to VQE.

2.3 Advantage argument, assumptions and limitation of the VQE

Quantum supremacy occurs when quantum algorithms outperform classical counterparts in accuracy or resource efficiency, as shown in recent experiments, although the scale of this advantage has been debated. The term *quantum advantage* is often used interchangeably with quantum supremacy, but here it refers to practical applications of quantum supremacy with measurable impact. Demonstrating quantum advantage requires a clear definition of the resources and accuracy metrics involved, accounting for all overheads, including initialization. Resources may be evaluated in terms of runtime, scalability, memory, financial cost, or energy consumption.

The VQE allows for the probabilistic measurement of observables over parameterized approximate wavefunctions, which cannot be sampled or computed efficiently on classical devices as system sizes grow. This assumes the Hamiltonians studied scale polynomially, as is common in fields like quantum chemistry, condensed matter physics, and nuclear physics. While classical methods can handle some wavefunction classes, the VQE offers access to forms that are otherwise intractable, opening the possibility for quantum advantage. If these wavefunctions provide sufficient accuracy in approximating the ground state, the VQE could surpass classical methods, demonstrating a necessary condition for its practical utility. Under specific theoretical assumptions, this condition appears achievable in certain cases.

There are, however, several constraints in quantum computing that the standard approach does not fully address, prompting two more stringent conditions. First, the VQE must demonstrate equal or greater accuracy than conventional methods, but with a shorter time-to-solution. This accounts for potential hardware limitations that could impose significant overhead, or a large "pre-factor," which is the multiplier applied to scaling rules to estimate runtime. If the VQE exhibits better asymptotic scaling but suffers from a large pre-factor, it would only achieve an advantage for extremely large systems, potentially rendering it impractical for smaller, real-world problems.

The second, more rigorous condition is for the VQE to deliver at least equal accuracy in a shorter computation time for systems of sufficient complexity to model a real-world problem in physics or chemistry. This entails testing on systems where the error from approximating the Hamiltonian is smaller than the VQE's solution. This could involve ensuring that basis sets are adequately refined or that interactions within the larger system are accurately captured. Even if the VQE delivers faster results with improved accuracy, these gains may not translate to practical advantage—particularly if the error margins are still too large to predict physical experiment outcomes due to environmental approximations in the Hamiltonian. [3]

Despite theoretical arguments supporting the VQE's polynomial scaling, several potential limitations could still prevent it from achieving quantum advantage:

- **Measurement Bottleneck**: One significant limitation of the VQE is the large pre-factor associated with the cost of accurately sampling observables. Multiple studies have examined whether VQE can reach a point where it outperforms conventional methods. So far, all have concluded that, given current research and assumptions, VQE does not surpass conventional approaches in the tractable applications considered. The primary bottleneck is the extensive number of measurements required to estimate the Hamiltonian's expectation value. While research into efficient operator sampling is ongoing, achieving a breakthrough may require rethinking how quantum hardware is designed to fully leverage VQE's potential for parallelization.

- **Optimization Complexity**: Another key challenge is that VQE requires solving a complex optimization problem. The overall cost and scaling of VQE depend heavily on the choice of optimizer and the nature of the optimization landscape. While some optimizers can converge in polynomial time for convex cost functions, VQE's optimization landscape is far less favorable. It has been shown that VQE optimization is NP-hard, implying that finding an optimal solution can be computationally intractable in the worst case. However, the crucial question is whether VQE can be heuristically optimized within a polynomial number of iterations to find a sufficiently accurate approximate solution.

- **Convergence and Measurement Challenges**: Even if VQE were to theoretically converge in a manageable number of iterations, this presumes that expectation values and gradients are computed exactly. In quantum computing, however, this assumption fails, particularly as the barren plateau problem may cause the number of measurements required for gradient estimation to scale exponentially in certain cases. Some mitigation strategies, such as identity block initialization and local Hamiltonian encoding, have been proposed, but it remains unclear how effectively these can handle barren plateaus for VQE. [8]

- **Noise Resilience and Error Mitigation**: VQE's resilience to quantum noise is another unresolved issue. While error mitigation has improved the algorithm's accuracy on NISQ devices, it often comes at a significant resource cost. Whether this resource demand will be acceptable in larger-scale applications remains unclear. Recent studies suggest that the resource cost increases exponentially with circuit depth. Although early VQE research suggested an inherent resilience to noise, where optimization adapts to noise levels, this adaptability has only been shown on small-scale quantum systems. Whether it holds in more complex, large-scale experiments involving deeper ansatz remains uncertain. Additionally, while in theory, a linear number of qubits could span the exact state, the VQE's parameterization often sacrifices this exactness. Achieving exact results would likely require an exponential number of variational parameters. Thus, VQE's potential for quantum advantage hinges on whether it can offer superior approximations for quantum many-body systems with more favorable scaling than conventional computational methods.

2.4 VQE and conventional computational chemistry

The first step in applying VQE to ab initio electronic structure problems is to define the basis functions that determine the system's resolution and representation. A common approach is to use molecular orbitals from a prior Hartree-Fock or density functional theory (DFT) calculation to set up the Hamiltonian and compute the Pauli string operator weights. This shows that VQE builds on conventional quantum chemistry techniques. To better understand the challenges of achieving quantum advantage in computational chemistry with VQE, it's useful to briefly consider existing methods in the field. [10]

Most high-accuracy ab initio methods for ground-state energies rely on wavefunction approximations, which aligns with VQE's approach. Although other quantum variables like density matrices or Green's functions are sometimes used, they generally fall short in accuracy for ground-state energies. Methods like DFT, while offering a strong cost-accuracy balance for large systems, are not direct competitors to VQE due to their lack of systematic improvement. The real competition for VQE in the near to medium term comes from high-accuracy wavefunction approaches, which can scale polynomially or even exponentially but can still handle relatively large systems.

Full Configuration Interaction (FCI) serves as the gold standard for accurately representing a quantum state for a given Hamiltonian and basis set. However, due to its computational complexity, it is typically intractable for systems with more than 18 orbitals. Despite this, FCI plays a crucial role in quantum chemistry, offering a numerically exact treatment of correlated physics. The method constructs a wavefunction as a linear combination of all possible electron configurations, ensuring invariance to the specific choice of single-particle orbitals. Typically, FCI uses Hartree-Fock molecular orbitals to enhance convergence, as Hartree-Fock approximates the ground state wavefunction at the mean-field level by providing a variationally optimized single Slater determinant. This basis diagonalizes the Fock matrix, associating single-particle energies with the orbitals. The FCI wavefunction is structured based on particle-hole excitations relative to the Hartree-Fock reference configuration, as:

$$|\psi\rangle_{FCI} = c_0 |\psi\rangle_{HF} + \sum_{ia} c_{ia} \hat{a}_a^\dagger \hat{a}_i |\psi\rangle_{HF} + \sum_{ij\ ab} c_{ijab} \hat{a}_a^\dagger \hat{a}_b \hat{a}_j^\dagger \hat{a}_i |\psi\rangle_{HF} + \dots \tag{2.5}$$

In Full Configuration Interaction (FCI), the first sum in the wavefunction represents 'singly-excited' configurations where an occupied spin-orbital (labeled by indices like i, j) is depopulated, and a virtual spin-orbital (labeled by indices like $a, b, \dots$) is populated. This depopulation and population are achieved while maintaining the antisymmetry of the wavefunction, using fermionic second quantized operators $\hat{a}^\dagger$ and $\hat{a}$.

These excitations start from the reference Hartree-Fock determinant and can extend to double excitations, triple excitations, and so on, all the way up to m-fold excitations, where m is the number of electrons. This spans the entire space of configurations, allowing the minimization of the Ritz functional, which can then be expressed as a diagonalization of the full Hamiltonian in this configuration space. Exact excited states within this defined basis can also be found as successively higher eigenvalues of the Hamiltonian.

Full Configuration Interaction (FCI) serves as the 'ground truth' solution in quantum chemistry for a given Hamiltonian and basis set. However, since FCI quickly becomes computationally intractable for larger systems, much of electronic structure research focuses on approximating the FCI solution. The goal is to reduce the complexity of FCI (ideally to a polynomial scaling with system size) while keeping the loss in accuracy as minimal as possible. Many approximate methods have been developed to strike this balance, allowing systems far larger than those solvable by FCI to achieve results with chemical accuracy. Chemical accuracy is the level of precision needed to compute reaction enthalpies, typically within 1.6 milli-Hartree (mEH) of experimental results. While achieving chemical accuracy for large systems is challenging, methods often focus on getting a qualitatively correct description of chemical properties. Some properties, like spectroscopic ones, require much higher precision. [6]

The considerations for devising an effective parameterization in conventional computational methods are similar to those used when developing ansatz for the VQE. While the functional forms of these ansatz differ due to the need for efficient evaluation on quantum devices, the core principles remain aligned. In the next section, we will examine several parameterizations used on conventional devices and discuss how these ideas have influenced the design of ansatz for VQE in quantum computing. [10]

2.4.1 Efficient approximate wavefunction parameterizations for conventional computation

Efficient approximate wavefunction parameterizations have been developed to tackle the complexity of FCI while maintaining high accuracy. These methods allow for larger system sizes with only minimal accuracy loss. A notable study demonstrated this by comparing nine different approaches on benzene, utilizing 30 electrons and 108 molecular orbitals. The root mean square deviation between the methods was just 1.3 milli-Hartree, indicating near-FCI accuracy. Similar results were seen in a study involving transition metal systems, showcasing the reliability of state-of-the-art wavefunction methodologies.

An effective way to simplify the full configuration interaction (FCI) wavefunction is by truncating the number of excitations from the reference configuration, while maintaining its linear form. A widely used approximation is the configuration interaction with singles and doubles (CISD), which only retains up to double excitations. More advanced methods like Adaptive Sampling CI (ASCI), Semistochastic Heat-Bath CI (SHCI), and Full Configuration Interaction Quantum Monte Carlo further improve accuracy by selectively including configurations based on sparsity in the optimized amplitudes. However, these linear approximations can struggle with scaling properly in larger systems, where energy errors grow with system size. To address this, multi-linear approximations like the matrix product state (MPS) form, optimized within the density matrix renormalization group (DMRG), have been developed for accurate results in molecular and lattice models. Variational Monte Carlo (VMC) offers another alternative, using an approximate ansatz whose parameters are optimized via Monte Carlo integration, though this requires managing stochastic errors. Many of these strategies influence the development of ansatzes for the VQE.

To address the size inconsistency in linear ansatz, the coupled-cluster (CC) method modifies and exponentiates the wavefunction form, ensuring size-consistent results with an excellent accuracy-to-cost ratio. The most popular variant, coupled-cluster with single, double, and perturbative triple excitations (CCSD(T)), is known as the "gold standard" in quantum chemistry when correlations are moderate. In cases with stronger correlations, other coupled-cluster forms have been developed. This approach also inspired the unitary coupled-cluster (UCC) ansatz used in VQE, where a similar structure is adapted for efficient implementation on quantum circuits. Dynamic inclusion of excitations, such as in the ADAPT-VQE ansatz, and systematic improvement in the Efficient Symmetry Preserving ansatz aim to better span the full configuration interaction (FCI) wavefunction. However, even with these enhancements, VQE's ability to fully capture the FCI description may remain exponential in system size, limiting its scalability for larger systems.

In applying conventional approximate wavefunction parameterizations, it's essential to recognize that error sizes vary depending on the system. Through extensive theoretical and numerical analysis, insights have emerged about when specific methods are most accurate. For example, coupled-cluster excels with low-rank excitations, DMRG handles systems with locality well, and selected CI or FCIQMC perform effectively when states exhibit sparsity. This understanding has allowed for the effective application of these methods and has driven improvements in accuracy and scope.

In contrast, the analysis of errors and accuracy in VQE ansatz for quantum simulation is still in its early stages. More research is needed to classify and investigate the approximations in these forms and their impact across different systems.

Established wavefunction methods in conventional computing represent the state of the art for high-accuracy quantum chemistry, especially for ground state energies. These methods provide a benchmark for assessing the success of VQE, as they aim to achieve chemical accuracy with greater efficiency than FCI. VQE faces a challenging target, as conventional methods have benefited from decades of research. Additionally, emerging fields like machine-learning-inspired ansatz are likely to continue pushing the boundaries of accuracy on conventional devices, raising the bar for VQE to demonstrate clear advantages.

2.5 VQE and quantum phase estimation

The Quantum Phase Estimation algorithm (QPE) offers a method for determining a specific eigenvalue of a Hamiltonian from an approximated eigenstate (whether ground or excited). QPE can determine an eigenvalue to a desired level of accuracy, with a probability proportional to how close the approximated eigenstate is to the actual eigenstate. However, it requires quantum circuits with depths that are significantly beyond what is currently achievable in the NISQ era of quantum computing. As part of our discussion on the VQE, we provide a brief outline of QPE and compare the two.

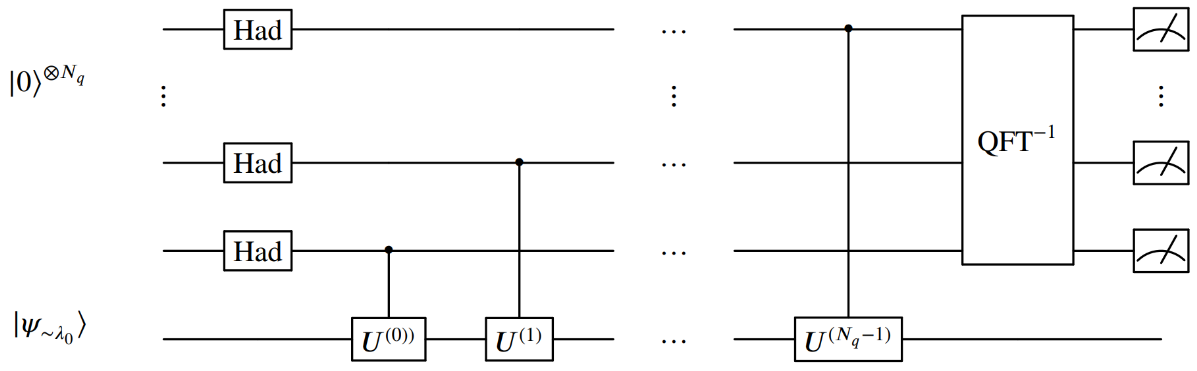


Figure 2: Quantum circuit for Quantum Phase Estimation

A depiction of the quantum circuit used to implement QPE is shown in Fig. 2, and the process can be outlined as follows:

The goal of QPE, similar to the goal of VQE, is essentially to compute an eigenvalue of a Hamiltonian. However, in the case of QPE, the problem described in Equation 2.2 is slightly restructured. For a given Hamiltonian $\hat{H}$, and a given eigenstate $|\lambda_j\rangle$ (usually the ground state: $|\lambda_0\rangle$), the objective is to find an eigenvalue E_j such that:

$$e^{i\hat{H}}|\lambda_j\rangle = e^{iE_j}|\lambda_j\rangle \tag{2.6}$$

The Hamiltonian is exponentiated to create a unitary operator, and without any loss of generality, this can be expressed as: $e^{iE_j} = e^{2\pi i\theta_j}$ where θ_j is the 'phase' that QPE seeks to discover.

However, the only available inputs are the Hamiltonian and an approximation of the ground state $|\psi_0\rangle \sim |\lambda_0\rangle$, which can generally be written in the eigenbasis of the Hamiltonian as:

$$|\psi_0\rangle = \sum_{j=0}^{2^{N_q-1}} \alpha_j|\lambda_j\rangle \tag{2.7}$$

where N_q is the number of qubits used to represent the electronic wavefunction of the Hamiltonian, thus having a total of 2^{N_q} eigenstates.

A register of ancilla qubits is used to map the sought-after eigenvalue, typically as a binary number. The number of ancillas required depends on the desired precision and the specific implementation (more ancillas allow for a longer binary string, which provides higher precision). The ancilla register is initialized in an equally weighted superposition of all possible states in the computational basis (i.e., all possible binary strings). If there are N_a ancilla qubits, there are 2^{N_a} basis elements. Starting from a register in the state $|0\rangle^{\bigotimes N}$, a Hadamard gate (Had) is applied to each qubit. The Hadamard gate operation $Had|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, and $Had|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. After these operations, the state of the ancilla register is

$$|\psi_{anc}\rangle = \frac{1}{(\sqrt{2})^{N_a}} \sum_{x=0}^{2^{N_a}-1} |bin(x)\rangle \tag{2.8}$$

where x represents integers from 0 to $2^{N_a} - 1$ and $bin(x)$ the binary representation of of x. When both the ground state approximation and the ancilla register are considered together we get the total state of the qubit register $|\psi_{total}\rangle = |\psi_{anc}\rangle \bigotimes |\psi_0\rangle$ such that:

$$\begin{aligned} |\psi_{tot}\rangle &= \frac{1}{\sqrt{2}^{N_a}} \sum_{x=0}^{2^{N_a}-1} |bin(x)\rangle \bigotimes \sum_{j=0}^{2^{N_q}-1} \alpha_j |\lambda_j\rangle \\ &= \frac{1}{\sqrt{2}^{N_a}} \sum_{x=0}^{2^{N_a}-1} \sum_{j=0}^{2^{N_q}-1} \alpha_j |bin(x)\rangle \bigotimes |\lambda_j\rangle \end{aligned} \tag{2.9}$$

In the superposition state above, there is no clear relation between an ancilla state $|bin(x)\rangle$ and the j-th eigenstate $|\lambda_j\rangle$. That is, if one were to measure the ancillas resulting in a binary number $bin(x)$, no information could be gained about the state of the wavefunction register which encodes $|\lambda_j\rangle$, and therefore no information can be gained about the associated eigenvalues. In the following, we will apply unitary gates to this superposition state such that there is a clear one to one correspondence between a measured binary number $bin(x)$ and the eigenvector $|\lambda_j\rangle$. Consider the following unitary

$$U^{(k)} = (e^{i\hat{H}})^{2^k} \tag{2.10}$$

with k an arbitrary number for the time being. Following Equation 2.6, if this unitary is applied to eigenstate $|\lambda_j\rangle$, it effectively results in a phase $e^{2\pi i\theta_j 2^k}$. Now suppose that k is the index of the ancilla qubits, i.e. $k \in [0, N_a - 1]$ and that, for each k, U^k is applied to the ground state approximation only if the ancilla qubit of index k is in state $|\mathbf{0}\rangle$. This operation can be performed by mean of a controlled unitary operation, which applies a unitary operation subject to the value of a control qubit. For a superposition instance *bin(x)* of the ancilla register, this means that the unitary U is applied x times in total to the ground state (consider for example the ancilla superposition $|bin(5)\rangle = |101\rangle$, here qubits are indexed from right to left to correspond to binary strings. The unitary is applied for $k = 0$, and for $k = 2$, hence following Equation 2.10 it is applied $5 = 1 \cdot 2^2 + 1 \cdot 2^0$ times). We obtain the state

$$|\psi_{tot}\rangle = \frac{1}{\sqrt{2}^{N_a}} \sum_{x=0}^{2^{N_a}-1} \sum_{j=0}^{2^{N_q}-1} e^{2\pi i \theta_j x} \alpha_j |bin(x)\rangle \bigotimes |\lambda_j\rangle \tag{2.11}$$

The next step reduces the number of superposition instances by applying an inverse quantum Fourier transform (QFT) to the ancilla register. QFT is a transformation from the computation basis to the Fourier basis, mapping a single computational basis element $bin(y)$ to a superposition of all computational basis elements each with different relative phase (due to the periodicity of the phase, each relative phase is a different point on the 2^{π} period, with a total of 2^{N_a} different points)

$$QFT|bin(y)\rangle = \sum_{x=0}^{2^{N_a}-1} \sum_{j=0}^{2^{N_q}-1} e^{2\pi i \theta_j x} \alpha_j |bin(x)\rangle \bigotimes |\lambda_j\rangle \tag{2.12}$$

If we set $y = 2^{N_a}\theta_i$, we can observe that applying the inverse QFT to the ancilla register in Equation 2.11 results in

$$(QFT^{-1} \bigotimes I^{\bigotimes N_q})|\psi_{tot}\rangle = \sum_{j=0}^{2^{N_q}-1} \alpha_j |bin(2^{N_a}\theta_j)\rangle|\lambda_j\rangle \tag{2.13}$$

where for simplicity we have assumed $2^{N_a}\theta_j \in N$. • Measuring the ancilla register in the Z basis returns the binary string $bin(2^{N_a}\theta_i)$ with probability $|\alpha_j|^2$, from which θ_i, and E_i can be recovered easily. The complete qubit register then collapses to the state $|bin(2^{N_a}\theta_j\rangle|\lambda_j\rangle$.

2.5.1 Discussion and comparison

In QPE, since the ground state is measured directly in binary form, the number of ancilla qubits required is directly linked to the precision targeted. Each ancilla qubit provides one bit of information, and thus, the number of ancilla qubits scales as $O(1/log_2(\epsilon))$ in terms of precision. Additionally, the number of controlled unitaries doubles for each added ancilla qubit, scaling as $O(1/\epsilon)$. These controlled unitaries represent the action of the Hamiltonian on a state. [9]

Thus, the central element of QPE is the efficient simulation of the Hamiltonian. Efficient Hamiltonian simulation methods are essential for this process, and relevant techniques are available. If an unlimited pool of qubits is assumed, Babbush et al. demonstrate that by constructing the appropriate Hamiltonian representation (specifically using a plane wave basis in first quantization), one can achieve sub-linear scaling with respect to the number of basis elements for Hamiltonian simulation.

As stated earlier, the success of QPE depends on how close the fidelity of the input state is to the unknown target eigenstate. This means that using a randomized state as input is not feasible, as its expected fidelity to the target eigenstate decreases exponentially with the system size. As a result, QPE becomes exponentially expensive with imperfect input state preparation. Various methods have been suggested to generate a sufficiently accurate approximation of the target eigenstate in a feasible way, many of which are based on conventional quantum chemistry approaches or adiabatic quantum computation.

There have been numerous successful implementations of QPE on quantum devices. However, these have only been executed on small systems, as large-scale implementations necessitate quantum resources that are not currently available. Specifically, large-scale controlled unitaries, essential for QPE, cannot be reliably executed on NISQ devices. This limitation also applies to the inverse Quantum Fourier Transform (QFT). Several numerical studies have been conducted to evaluate the full cost of implementing QPE on relevant systems and estimate the runtime on a fault-tolerant quantum computer. Nitrogen fixation has emerged as a standard benchmark for this algorithm. For example, Reiher et al. estimate that the 54 electrons and 108 spin orbitals of the FeMo-cofactor would require over $O(10^{15})$ T gates, 200 million qubits, and would need to run for over a month to achieve quantitatively accurate results (assuming 100 ns gate times and an error threshold of 10^{-3}. Berry et al. use qubitization—a method that transforms the evolution operator into a quantum walk—to reduce gate requirements to $O(10^{11})$ Toffoli gates, even with an extended active space. Lee et al. further improve upon these results, estimating that they could perform this energy computation with four million physical qubits and less than four days of runtime, maintaining a similar $O(10^{-3})$error threshold. [11]

There have also been numerous resource estimates for condensed matter models, such as for a 100-site version of the Fermi-Hubbard model, which could potentially be solved with approximately 500,000 physical qubits running for a few hours. Additionally, Elfving et al. estimate that, with similar error rates, the chromium dimer (Cr_2) with an active space of 52 spin orbitals and 26 electrons would require $O(10^7)$ physical qubits running for about 110 hours. While the estimated runtimes and hardware requirements remain substantial, research has progressed swiftly, presenting a promising outlook for QPE, especially for specific quantum chemistry tasks.

The VQE balances the depth and number of qubits required by QPE with a higher number of measurements and repetitions of the circuit, as well as constraints from the use of an approximate ansatz for the quantum state. While QPE needs repetitions with circuit depth scaling as $O(1)$ for precision ϵ , VQE requires $O(1/\epsilon^2)$ shots with circuit depth scaling as $O(1)$ in precision. Although many other factors contribute to the overall time scaling of both methods, this highlights the asymptotic efficiency of QPE compared to VQE, assuming access to fault-tolerant quantum computers. However, VQE proves to be more resource-efficient than QPE for NISQ-era devices.

The boundary between NISQ and fault-tolerant quantum computation is not clear-cut, and as Wang et al. point out, neither is the distinction between VQE and QPE. They propose an approach called Accelerated VQE (or α-VQE), which incorporates small-scale QPE calculations as subroutines for VQE. This method introduces a tunable parameter $\alpha \in [0,1]$, allowing the circuit depth to scale as $O(1/\epsilon^{\alpha})$ and the number of samples to scale as $O(1/\epsilon^{2(1-\alpha)})$. With $\alpha = 1$, the scaling matches QPE, and with $\alpha = 0$, the scaling matches VQE. In general, rather than being exclusive alternatives for solving electronic structure problems, VQE and QPE are likely to work best when combined as complementary methods, offering algorithmic flexibility that can adapt as quantum hardware improves. [7]

Chapter 3

Conclusion and outlook

The VQE stands out as one of the most promising near-term applications for quantum computing. Despite solid theoretical foundations, numerous open questions remain regarding its broader applicability. This review outlines the critical research surrounding VQE's various components and their interconnections.

The choice of Hamiltonian significantly influences both the qubit requirements and VQE's ability to yield accurate results. In practice, the second quantization framework using a canonical orbital basis has dominated due to its lower qubit demand, though alternative bases like plane waves and frozen natural orbitals have been explored. Once the Hamiltonian is constructed, choosing an efficient mapping from fermionic to spin operators is crucial for the system under study.

A well-known challenge for VQE is the large number of measurements needed for optimization. Some improvements can be achieved by managing measurement distribution, such as by allocating shots based on operator weights in the Hamiltonian or by truncating insignificant terms. However, more significant scaling reductions require methods that enable simultaneous measurement of Hamiltonian terms. For molecular Hamiltonians, a decomposed interactions method currently offers the most efficient approach, whereas in lattice models, a simple qubit-wise commutativity grouping scheme may prove more cost-effective due to the feasibility of low-weight encodings.

The core of VQE lies in the choice of the ansatz, or parametrized quantum circuit, used to model the trial wavefunction. Fixed-structure ansatzes like k-UpCCGSD and its extensions have shown strong accuracy with linear scaling in qubits, while methods like the Hamiltonian Variational Ansatz are more suited for lattice models. Adaptive ansatz approaches show potential, though their overall cost and scalability remain uncertain. The choice of optimizer is crucial for VQE's convergence, with the quantum natural gradient and RotoSolve showing some success in navigating the complex optimization landscape. Finally, noise mitigation methods, such as symmetry verification and extrapolation, are likely necessary to achieve accurate results, though they add to computational overhead.

There are four main areas for further research in VQE: (1) optimizing measurement strategies to reduce the high number of shots needed for Hamiltonian expectation value estimation; (2) evaluating the costs and benefits of VQE's parallelization potential, which may be necessary for its future relevance; (3) addressing the barren plateau problem, the complexity of the optimization landscape, and the efficacy of optimizers; and (4) exploring the noise resilience of VQE and tractable noise mitigation techniques.

Several algorithms inspired by VQE have also emerged, such as the Full Quantum Eigensolver (FQE) for faster quantum gradient descent optimization, and Projective Quantum Eigensolver, which optimizes energy through projection residuals. Techniques like Quantum Assisted Simulator (QAS) and Quantum Sampling Regression aim to bypass the barren plateau problem and the quantum-classical communication bottleneck. Other innovations, like the Variational Quantum State Eigensolver (VQSE) and Permutation VQE (PermVQE), target specific challenges in eigenstate discovery and correlation minimization. Methods like Filtering VQE and Deep VQE offer more efficient ground state solutions and reduce qubit requirements. Some, like ctrl-VQE, explore performing VQE directly at the pulse level, while variational methods have also been extended to real and imaginary time evolution for discovering ground and excited states.

To evaluate the future potential of VQE as an electronic structure method, it's essential to consider its integration with other algorithms, such as those mentioned earlier or with Quantum Phase Estimation (QPE), to enhance computational efficiency. VQE continues to be a highly promising approach for quantum applications in the near future. Similar to many new technologies, it encounters substantial obstacles that need to be resolved prior to its practical application. Nevertheless, the potential for making significant advancements in areas such as drug discovery, chemical engineering, and material science is quite high.

Appendix A

Python codes and implementations

Here we'll demonstrate variational algorithm examples and how we may apply them in Qiskit; note that this framwork can be applied to any problem, but here we will only call out the framework steps in one example case, run on a real hardware. [12]

To run this on real hardware, we must optimize the quantum circuits for our quantum computer of choice. here, we will simply use the least-busy backend.

```python
from qiskit_ibm_runtime import SamplerV2 as Sampler
from qiskit_ibm_runtime import EstimatorV2 as Estimator
from qiskit_ibm_runtime import Session, Options
from qiskit_ibm_runtime import QiskitRuntimeService

service = QiskitRuntimeService(channel='ibm_quantum')
backend = service.least_busy(operational=True, simulator=
    False)
```

We will use a preset pass manager for transpilation, and we will maximally optimize our circuit using optimization level 3.

```python
from qiskit.transpiler.preset_passmanagers import
    generate_preset_pass_manager

pm = generate_preset_pass_manager(backend=backend,
    optimization_level=3)
isa_ansatz = pm.run(ansatz)
isa_observable = H2_op.apply_layout(layout = isa_ansatz.
    layout)
```

A.1 Custom VQE

Problem definition: Imagine that we want to use a variational algorithm to find the eigenvalue of the following observable:

$$\hat{O}_1 = 2II - 2XX + 3YY - 3ZZ$$

This observable has the following eigenvalues:

$$\begin{cases} \lambda_0 = -6 \\ \lambda_1 = 4 \\ \lambda_2 = 4 \\ \lambda_3 = 6 \end{cases}$$

And eigenstates:

$$\begin{cases} |\phi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\phi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\phi_2\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \\ |\phi_3\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \end{cases}$$

```python
from qiskit.quantum_info import SparsePauliOp

observable_1 = SparsePauliOp.from_list([("II", 2), ("XX", -2)
    , ("YY", 3), ("ZZ", -3)])
```

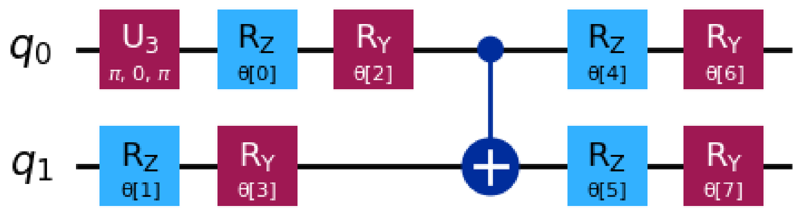
We'll first explore how to construct a VQE instance manually to find the lowest eigenvalue for $\hat{O}_1$. This will incorporate a variety of techniques.

```python
def cost_func_vqe(params, ansatz, hamiltonian, estimator):
    """Return estimate of energy from estimator

    Parameters:params (ndarray): Array of ansatz
    parametersansatz (QuantumCircuit):
    Parameterized ansatz
    circuithamiltonian (SparsePauliOp): Operator
        representation of Hamiltonian
    estimator (Estimator): Estimator primitive instance

    Returns:float: Energy estimate"""
    pub = (ansatz, hamiltonian, params)
```

```
        cost = estimator.run([pub]).result()[0].data.evs

        return cost
```

```
    from qiskit.circuit.library import TwoLocal
    from qiskit import QuantumCircuit

    import numpy as np

    reference_circuit = QuantumCircuit(2)
    reference_circuit.x(0)

    variational_form = TwoLocal(
        2,rotation_blocks=["rz","ry"],
        entanglement_blocks="cx",
        entanglement="linear",
        reps=1,
    )
    raw_ansatz = reference_circuit.compose(variational_form)

    raw_ansatz.decompose().draw('mpl')
```

Output:



We will now start debugging on local simulators.

```
    from qiskit.primitives import StatevectorEstimator as
       Estimator
    from qiskit.primitives import StatevectorSampler as Sampler
    estimator = Estimator()
    sampler = Sampler()
```

We now set an initial set of parameters:

```
    import numpy as np

```

```
x0 = np.ones(raw_ansatz.num_parameters)
print(x0)
```

```
import numpy as np

# Initialize an array of ones
x0 = np.ones(8)
print(x0)
```

Output:

```
[1. 1. 1. 1. 1. 1. 1. 1.]
```

We can minimize this cost function to calculate optimal parameters

```
# SciPy minimizer routine
from scipy.optimize import minimize
import time

start_time = time.time()
result = minimize(cost_func_vqe, x0, args=(raw_ansatz,
    observable_1, estimator), method="COBYLA", options={'
    maxiter':1000, 'disp':True})
end_time = time.time()
execution_time = end_time - start_time
```

Output:

```
Normal return from subroutine COBYLA
NFVALS =   126
F =-6.000000E+00
MAXCV = 0.000000E+00
X = 1.742657E+00   9.451574E-01   1.570823E+00  -3.501199E-05   1.922670E+00
   1.218953E+00   6.220864E-01   6.219692E-01
```

```
result
```

```
message: Optimization terminated successfully.
success: True
status: 1
    fun: -5.999999958375564
      x: [ 1.743e+00  9.452e-01  1.571e+00 -3.501e-05  1.923e+00
           1.219e+00  6.221e-01  6.220e-01]
nfev: 126
maxcv: 0.0
```

Because this toy problem uses only two qubits, we can check this by using NumPy's linear algebra eigensolver.

```
    from numpy.linalg import eigvalsh

    solution_eigenvalue = min(eigvalsh(observable_1.to_matrix()))

    print(f"""Number of iterations:{result.nfev}""")print(f"""
        Time (s): {execution_time}""")

    print(f"Percent error: {abs((result.fun -solution_eigenvalue)
        /solution_eigenvalue):.2e}"
)
```

```
Number of iterations: 126
Time (s): 0.40227270126342773
Percent error: 6.94e-09
```

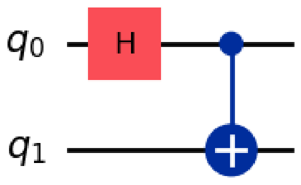
As you can see, the result is extremely close to the ideal.

A.1.1 Experimenting to improve speed and accuracy

Adding reference state: In the previous example we have not used any reference operator U_R. now let us think about how the ideal eigenstate $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ can be obtained. Consider the following circuit.

```
    from qiskit import QuantumCircuit

    ideal_qc = QuantumCircuit(2)
    ideal_qc.h(0)
    ideal_qc.cx(0,1)

    ideal_qc.draw("mpl")
```

Output:



We can check quickly that this circuit gives us the desired state.

```python
from qiskit.quantum_info import Statevector

Statevector(ideal_qc)
```

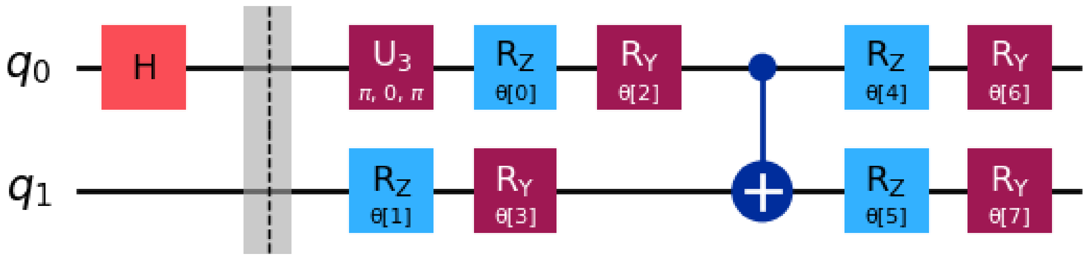
Output:

```
Statevector([0.70710678+    0.j,   0. +0.j,    0. +0.j,0.70710678+0.j],
dims=(2, 2))
```

Now that we have seen how a circuit preparing the solution state looks like, it seems reasonable to use a Hadamard gate as a reference circuit, so that the full ansatz becomes:

```python
reference = QuantumCircuit(2)
reference.h(0)
# Include barrier to separate reference from variational form
reference.barrier()

ref_ansatz = raw_ansatz.decompose().compose(reference, front=
    True)

ref_ansatz.draw("mpl")
```

Output:



For this new circuit, the ideal solution could be reached with all the parameters set to 0, so this confirms that the choice of reference circuit is reasonable.

Now let us compare the number of cost function evaluations, optimizer iterations and time taken with those of the previous attempt.

```python
import time

start_time = time.time()

ref_result = minimize(cost_func_vqe, x0, args=(ref_ansatz,
    observable_1, estimator), method="COBYLA")

end_time = time.time()
execution_time = end_time - start_time
```

Using our optimial parameters to calculate the minimum eigenvalue:

```
    experimental_min_eigenvalue_ref = cost_func_vqe(result.x,
       raw_ansatz, observable_1, estimator)
    print(experimental\_min\_eigenvalue\_ref)
```

Output:

```
-5.999999958375564
```

```
    print("ADDED REFERENCE STATE:")
    print(f"""Number of iterations{result.nfev}""")
    print(f"""Time (s): {execution_time}""")
    print(
    f"Percent error: {abs((experimental_min_eigenvalue_ref -
       solution_eigenvalue)/solution_eigenvalue):.2e}")
```

```
ADDED REFERENCE STATE:
Number of iterations: 126
Time (s): 0.35369086265563965
Percent error: 6.94e-09
```

Changing the Initial Point: Now that we have seen the effect of adding the reference state, we will go into what happens when we choose different initial points $\vec{\theta_0}$. In particular we will use $\vec{\theta_0} = (0, 0, 0, 0, 6, 0, 0, 0)$ and $\vec{\theta_0} = (6, 6, 6, 6, 6, 6, 6, 6)$.

Remember that, as discussed when the reference state was introduced, the ideal solution would be found when all the parameters are 0, so the first initial point should give fewer evaluations.

```
    import time

    start_time = time.time()

    x0 = [0, 0, 0, 0, 6, 0, 0, 0]

    x0_1_result = minimize(cost_func_vqe, x0, args=(raw_ansatz,
       observable_1, estimator), method="COBYLA")

    end_time = time.time()
    execution_time = end_time - start_time
    print("INITIAL POINT 1:")
    print(f"""Number of iterations: {x0_1_result.nfev}""")
    print(f"""Time (s): {execution_time}""")
```

Output:

```
INITIAL POINT 1:
Number of iterations: 120
Time (s): 0.3904449939727783
```

Adjusting inital point to $\vec{\theta_0} = (6, 6, 6, 6, 6, 6, 6, 6, 6)$

```
    import time

    start_time = time.time()

    x0 = 6 * np.ones(raw_ansatz.num_parameters)

    x0_2_result = minimize(cost_func_vqe, x0, args=(raw_ansatz,
        observable_1, estimator), method="COBYLA")

    end_time = time.time()
    execution_time = end_time - start_time
```

```
    print("INITIAL POINT 2:")
    print(f"""Number of iterations: \{x0\_1\_result.nfev\}""")
    print(f"""Time (s): \{execution\_time\}""")
```

Output:

```
INITIAL POINT 2:
Number of iterations: 115
Time (s): 0.41460418701171875
```

By experimenting with different initial points, we might be able to achieve convergence faster and with fewer function evaluations.

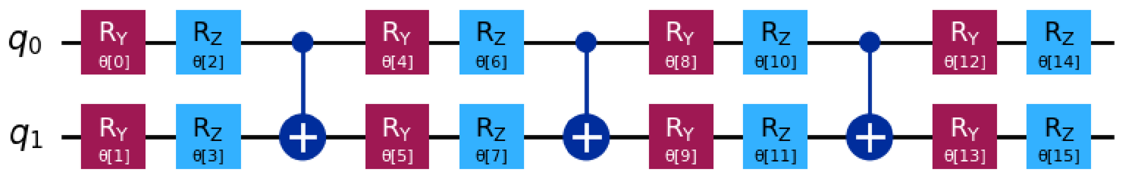
A.2 Quantum Chemistry: Ground State and Excited Energy Solver

Our objective is to minimize the expectation value of the observable representing energy (Hamiltonian $\hat{H}$):

$$\min_{\vec{\theta}}\langle\psi(\vec{\theta})|\hat{H}|\psi(\vec{\theta})\rangle \tag{A.1}$$

```
    from qiskit.quantum_info import SparsePauliOp
    from qiskit.circuit.library import EfficientSU2

    H2_op = SparsePauliOp.from_list(
    [
        ("II", -1.052373245772859),
        ("IZ", 0.39793742484318045),
        ("ZI", -0.39793742484318045),
        ("ZZ", -0.01128010425623538),(
        "XX", 0.18093119978423156),
    ]
)

    chem\_ansatz = EfficientSU2(H2\_op.num\_qubits)

    chem\_ansatz.decompose().draw("mpl")
```

Output:



```
    from qiskit.circuit.library import TwoLocal
    from qiskit import QuantumCircuit

    def cost\_func\_vqe(params, ansatz, hamiltonian, estimator):
    """Return estimate of energy from estimator

        Parameters:
        params (ndarray): Array of ansatz parameters
        ansatz (QuantumCircuit): Parameterized ansatz circuit
        hamiltonian (SparsePauliOp): Operator representation of
           Hamiltonian
        estimator (Estimator): Estimator primitive instance

        Returns:
            float: Energy estimate
        """
        pub = (ansatz, hamiltonian, params)
        cost = estimator.run([pub]).result()[0].data.evs\
        # cost = estimator.run(ansatz, hamiltonian, parameter\
           _values=params).result().values[0]returncost
```

Now we set an initial set of parameters:

```
    import numpy as np

    x0 = np.ones(chem\_ansatz.num\_parameters)
```

We can minimize this cost function to calculate optimal parameters, and we can check our code first by using a local simulator.

```
    from qiskit.primitives import StatevectorEstimator as
       Estimator
    from qiskit.primitives import StatevectorSampler as Sampler
    estimator = Estimator()
    sampler = Sampler()
```

```
    # SciPy minimizer routine
    from scipy.optimize
    import minimize
    import time

    start_time = time.time()

    result = minimize(cost_func_vqe, x0, args=(chem_ansatz, H2_op
       , estimator), method="COBYLA")

    end_time = time.time()
    execution_time = end_time - start_time

    result
```

Output:

```
message: Optimization terminated successfully.
success: True
status: 1
    fun: -1.8572750261786257
    x: [ 7.551e-01  1.303e+00 ...  7.949e-01  1.361e+00]
```

```
nfev: 277
maxcv: 0.0
```

The minimum value of the cost function (−1.857...) is the ground state energy of the H_2 molecule, in units of hartrees.

Excited States We can also leverage VQD to solve for $k = 2$ total states (the ground state and the first excised state).

```
from qiskit.quantum_info
import SparsePauliOpimport numpy as np

k = 2
betas = [33, 33]
#x0 = np.zeros(ansatz.num_parameters)
x0=[ 1.164e+00, -2.438e-01, 9.358e-04, 6.745e-02, 1.990e+00,
   9.810e-02, 6.154e-01, 5.454e-01]
```

We'll add our overlap calculation:

```
from scipy.optimize import minimize

prev_states = []
prev_opt_parameters = []
eigenvalues = []

realbackend = 0

for step in range(1, k + 1):
    ifstep > 1:
        prev_states.append(ansatz.assign_parameters(
           prev_opt_parameters))
    result = minimize(cost_func_vqd, x0, args=(ansatz,
       prev_states, step, betas, estimator, sampler, H2_op,
       realbackend, None), method="COBYLA", options = {'tol':
        0.001, 'maxiter': 2000})
    print(result)

prev_opt_parameters = result.xeigenvalues.append(result.fun)
```

Output:

```
message: Optimization terminated successfully.
success: True
status: 1
```

```
    fun: -1.857267147452959
    x: [ 1.164e+00 -2.429e-01  5.085e-04  6.530e-02  1.990e+00
            9.788e-02  6.153e-01  5.477e-01]
nfev: 54
maxcv: 0.0
message: Optimization terminated successfully.
success: True
status: 1
    fun: -0.8122441045127886
    x: [ 2.921e+00  1.486e+00  1.884e+00 -4.657e-01  2.999e+00
            1.225e+00  6.465e-01  1.460e+00]
nfev: 73
maxcv: 0.0
```

```
eigenvalues
```

```
[-1.857267147452959, -0.8122441045127886]
```

Bibliography

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

[2] McClean, J. R., Romero, J., Babbush, R., Aspuru-Guzik, A. (2016). The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2), 023023.

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

[4] Bittel, L., Kliesch, M. (2021). Training variational quantum algorithms is NP-hard. *Physical Review Letters*, 127(12), 120502.

[5] Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J. S., Menke, T., Mok, W. K., Sim, S., Siopsis, G., Sanz, M., Solano, E., Wehner, S., Whitfield, J. D., Wölk, S., Zeng, B., Aspuru-Guzik, A. (2022). Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1), 015004.

[6] Fedorov, D. A., Peng, B., Govind, N., Alexeev, Y. (2022). VQE and beyond: Quantum algorithms for electronic structure. *Chemical Reviews*, 122(8), 6564-6633.

[7] Szabo, A., Ostlund, N. S. (1989). *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory.* Dover Publications.

[8] Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G. H., Tennyson, J. (2022). The Variational Quantum Eigensolver: a review of methods and best practices. *arXiv preprint arXiv:2111.05176v3.*

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

[10] Bauer, B., Bravyi, S., Motta, M., Chan, G. K.-L. (2020). Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22), 12685-12717.

[11] Rivero, P., Cloët, I. C., Sullivan, Z. (2020). An optimal quantum sampling regression algorithm for variational eigensolving in the low qubit number regime.

[12] IBM Learning Platform. "Variational Algorithm Design." `https://learning.quantum.ibm.com/course/variational-algorithm-design`

[13] Wikipedia. "Quantum computing." Available at: `https://en.wikipedia.org/wiki/Quantum_computing`.

Wikipedia. "Variational quantum eigensolver." Available at:`https://en.wikipedia.org/wiki/Variational_quantum_eigensolver`.