Qubit
PHARMACEUTICALS

Quantum Computing for Quantum Chemistry

Axel Courtat[1] and Daniele Loco[1] and Jerome Foret[1]

November 16, 2023

Table of contents

We are here

Why quantum computing for quantum chemistry ?

Qubit PHARMACEUTICALS

Until now you have been introduced to the field of Quantum Chemistry and where it stands in a Drug Discovery pipeline.
Numerical methods are mandatory to solve real quantum chemical problems, in particular when aiming to design drugs.
There is a lot of them, more or less approximating the true nature of molecules, directly influencing the computational cost of the methods.

As an example, if you want to compute the real ground-state energy for the molecule N_2 (i.e. doing a FCI calculation) on a classical computer you will need more than 100GB of memory !
Imagine what could be the cost of FCI calculation for a drug, that can contain hundreds of atom...

Why are such computations so expensive?

Why quantum computing for quantum chemistry ?

The computational cost of all methods scales as the number of orbitals used to describe the system.
First of all, going back to the Second Quantization expression of the Hamiltonian:

$$\mathbf{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pq,rs} a_p^\dagger a_q^\dagger a_r a_s + V_{nuc-nuc}$$

$$h_{pq} = \int_{\mathbb{R}^3} \varphi_p^\dagger(\vec{r}) h(\vec{r}) \varphi_q(\vec{r}) d^3\vec{r}$$

$$V_{pq,rs} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \varphi_p^\dagger(\vec{r}_1) \varphi_q^\dagger(\vec{r}_2) r_{12}^{-1} \varphi_r(\vec{r}_1) \varphi_s(\vec{r}_2) d^3\vec{r}_1 d^3\vec{r}_2$$

We see already that the computational cost of the Hamiltonian is $\mathcal{O}(M^4)$ where M is the number of molecular orbitals.

Why quantum computing for quantum chemistry ?

Qubit PHARMACEUTICALS

As we have briefly seen in the section about the theory of Quantum Chemistry, most of all methods are constructed upon the Hartree-Fock theory which is itself constructed on the Hamiltonian. Thus, the Hartree-Fock computational cost is scaling in the number of orbitals as the Hamiltonian, i.e. $\mathcal{O}(M^4)$.

Why quantum computing for quantum chemistry ?

As we have briefly seen in the section about the theory of Quantum Chemistry, most of all methods are constructed upon the Hartree-Fock theory which is itself constructed on the Hamiltonian. Thus, the Hartree-Fock computational cost is scaling in the number of orbitals as the Hamiltonian, i.e. $\mathcal{O}(M^4)$.
Below we have listed the computational costs for some methods :

Quantum Chemistry method	**Computational cost**
Hartree-Fock (HF)	$\mathcal{O}(M^4)$
Coupled Cluster (CC)	From $\mathcal{O}(M^4)$ to $\mathcal{O}(M^{13})$
Full Configuration Interaction (FCI)	$\mathcal{O}(M!)$

Why quantum computing for quantum chemistry ?

As we have briefly seen in the section about the theory of Quantum Chemistry, most of all methods are constructed upon the Hartree-Fock theory which is itself constructed on the Hamiltonian. Thus, the Hartree-Fock computational cost is scaling in the number of orbitals as the Hamiltonian, i.e. $\mathcal{O}(M^4)$.
Below we have listed the computational costs for some methods :

Quantum Chemistry method	Computational cost
Hartree-Fock (HF)	$\mathcal{O}(M^4)$
Coupled Cluster (CC)	From $\mathcal{O}(M^4)$ to $\mathcal{O}(M^{13})$
Full Configuration Interaction (FCI)	$\mathcal{O}(M!)$

And that is only the orbital-dependent cost, yet we know that we need to choose a basis set to each calculation and a lot of basis have a lot of elements, for correct molecular description...

Why quantum computing for quantum chemistry ?

We clearly see that classical methods can only be approximate methods for the field of Quantum Chemistry. Even with the fast growing of supercomputers performances, a FCI calculation will remain inaccessible without a breakthrough in novel computational architectures : this is where enters Quantum Computing !
The hope of Quantum Computing for enhancing Quantum Chemistry calculations is many-fold :

1. linear/low-order polynomial scaling in the number of orbitals
2. (small) polynomial computational time for all systems
3. allow computations for current unreacheable systems
4. ultimately allowing of FCI-like calculations for (very) large systems

Recall of Quantum Computing

A qubit is a two-state system defined as :

$$|\Psi(t)\rangle = c_0 |0\rangle + c_1 |1\rangle, |c_0|^2 + |c_1|^2 = 1 \quad (1)$$

where we are explicitely working in the **computational basis** $\{|0\rangle, |1\rangle\}$.
In Quantum Chemistry, the states of the computational basis usually correspond to :

- unoccupied atomic/molecular orbital ($|0\rangle$))
- occupied atomic/molecular orbital ($|1\rangle$)

Recall of Quantum Computing

A qubit state evolves under **unitary operations**, i.e. operations preserving the norm. They are generally described as :

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \ a, b, c, d \in \mathbb{C} \tag{2}$$

Belonging to the group $SU(2, \mathbb{C})$, they all possess the properties :

$$U^{-1} = U^{\dagger} \equiv \bar{U}^{T} = \begin{pmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{pmatrix} \tag{3}$$

$$det(U) = 1 \tag{4}$$

Recall of Quantum Computing

A qubit state evolves under **unitary operations**, i.e. operations preserving the norm. They are generally described as :

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix},\ a, b, c, d \in \mathbb{C} \tag{5}$$

Belonging to the group $SU(2, \mathbb{C})$, they all possess the properties :

$$U^{-1} = U^{\dagger} \equiv \bar{U}^{\intercal} = \begin{pmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{pmatrix} \tag{6}$$

$$det(U) = 1 \tag{7}$$

The operator basis of $SU(2, \mathbb{C})$ is made of the so-called **Pauli matrices** :

$$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \sigma_x \equiv X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_y \equiv Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_z \equiv Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{8}$$

Recall of Quantum Computing

Qubit PHARMACEUTICALS

Quantum Chemistry deals with multi-particle systems, thus the studied system will be made of many qubits. To describe multi-qubit system, we use the **tensorial product** :

$$\begin{aligned} \otimes : & E \times F \longrightarrow E \otimes F \\ & (X, Y) \longmapsto Z = X \otimes Y \equiv \sum_{n,m} X_n Y_m e_n \otimes e'_m \end{aligned}$$

Recall of Quantum Computing

Example

Tensorial of two $2x2$ matrices X, Y :

$$X \otimes Y = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \otimes \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \tag{9}$$

$$= \begin{pmatrix} x_{11} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} & x_{12} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \\ x_{21} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} & x_{22} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \end{pmatrix} \tag{10}$$

$$\tag{11}$$

Example

Tensorial of two $2x2$ matrices X, Y :

$$X \otimes Y = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \otimes \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \tag{12}$$

$$= \begin{pmatrix} x_{11} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} & x_{12} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \\ x_{21} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} & x_{22} \cdot \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \end{pmatrix} \tag{13}$$

$$= \begin{pmatrix} x_{11} \cdot y_{11} & x_{11} \cdot y_{12} & x_{12} \cdot y_{11} & x_{12} \cdot y_{12} \\ x_{11} \cdot y_{21} & x_{11} \cdot y_{22} & x_{12} \cdot y_{21} & x_{12} \cdot y_{22} \\ x_{21} \cdot y_{11} & x_{21} \cdot y_{12} & x_{22} \cdot y_{11} & x_{22} \cdot y_{12} \\ x_{21} \cdot y_{21} & x_{21} \cdot y_{22} & x_{22} \cdot y_{21} & x_{22} \cdot y_{22} \end{pmatrix} \tag{14}$$

Recall of Quantum Computing

Any multi-qubit system possesses the unique feature of **entanglement**. If the system is entangled, you cannot completely separate the subsystems.
Mathematically it means that you cannot write an expression with tensorial products to describe the multi-qubit state.

Recall of Quantum Computing

Any multi-qubit system possesses the unique feature of **entanglement**. If the system is entangled, you cannot completely separate the subsystems.
Mathematically it means that you cannot write an expression with tensorial products to describe the multi-qubit state.

Example

For a 2-qubit system :

- A pure (non-entangled) state

$$|\Psi_{pure}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \equiv |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{15}$$

Recall of Quantum Computing

Any multi-qubit system possesses the unique feature of **entanglement**. If the system is entangled, you cannot completely separate the subsystems.
Mathematically it means that you cannot write an expression with tensorial products to describe the multi-qubit state.

Example

For a 2-qubit system :

- A pure (non-entangled) state

$$|\Psi_{pure}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \equiv |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{16}$$

- An entangled state

$$|\Psi_{entangled}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{17}$$

Recall of Quantum Computing

Amongst all the gates existing in Quantum Computing, every quantum circuits that we will use are depending on the following one-qubit and two-qubit gates.

Recall of Quantum Computing

Amongst all the gates existing in Quantum Computing, every quantum circuits that we will use are depending on the following one-qubit and two-qubit gates.

Single qubit gates

$$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Recall of Quantum Computing

Amongst all the gates existing in Quantum Computing, every quantum circuits that we will use are depending on the following one-qubit and two-qubit gates.

Single qubit gates

$$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$R_x(\theta) = exp\left(-i\frac{\theta X}{2}\right) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad R_y(\theta) = exp\left(-i\frac{\theta Y}{2}\right) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_z(\theta) = exp\left(-i\frac{\theta Z}{2}\right) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

Recall of Quantum Computing

Amongst all the gates existing in Quantum Computing, every quantum circuits that we will use are depending on the following one-qubit and two-qubit gates.

Two qubit gates

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Recall of Quantum Computing

Amongst all the gates existing in Quantum Computing, every quantum circuits that we will use are depending on the following one-qubit and two-qubit gates.

Two qubit gates

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad CR_y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -\sin\frac{\theta}{2} \\ 0 & 0 & 1 & 0 \\ 0 & \sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{pmatrix}$$

Variational Quantum Eigensolver (VQE)

The VQE is a **hybrid quantum-classical algorithm**, i.e. it consists of a loop between a quantum computer and a classical one until convergence is achieved (or meeting a stopping criteria).
Using *Variational Principle* from Quantum Mechanics, it aims at minimizing a cost function, defined as the expectation value of a operator with respect to a parametrized wave function :

$$f(\vec{\theta}) = \langle \Psi(\vec{\theta}) | \mathcal{O} | \Psi(\vec{\theta}) \rangle \tag{18}$$

Variational Quantum Eigensolver (VQE)

The VQE is a **hybrid quantum-classical algorithm**, i.e. it consists of a loop between a quantum computer and a classical one until convergence is achieved (or meeting a stopping criteria).
Using *Variational Principle* from Quantum Mechanics, it aims at minimizing a cost function, defined as the expectation value of a operator with respect to a parametrized wave function :

$$f(\vec{\theta}) = \langle \Psi(\vec{\theta}) | \mathcal{O} | \Psi(\vec{\theta}) \rangle \tag{19}$$

The quantum computer is used to :

- quantum state preparation
- quantum circuit ended by measurements

Variational Quantum Eigensolver (VQE)

The VQE is a **hybrid quantum-classical algorithm**, i.e. it consists of a loop between a quantum computer and a classical one until convergence is achieved (or meeting a stopping criteria).
Using *Variational Principle* from Quantum Mechanics, it aims at minimizing a cost function, defined as the expectation value of a operator with respect to a parametrized wave function :

$$f(\vec{\theta}) = \langle \Psi(\vec{\theta}) | \mathcal{O} | \Psi(\vec{\theta}) \rangle \tag{20}$$

The quantum computer is used to :

- quantum state preparation
- quantum circuit ended by measurements

The classical computer is used to :

- Create the cost function from the QPU data
- Minimize the cost function and deliver optimized parameters
- Feed the QPU with optimized parameters for the new input state

Variational Quantum Eigensolver (VQE)

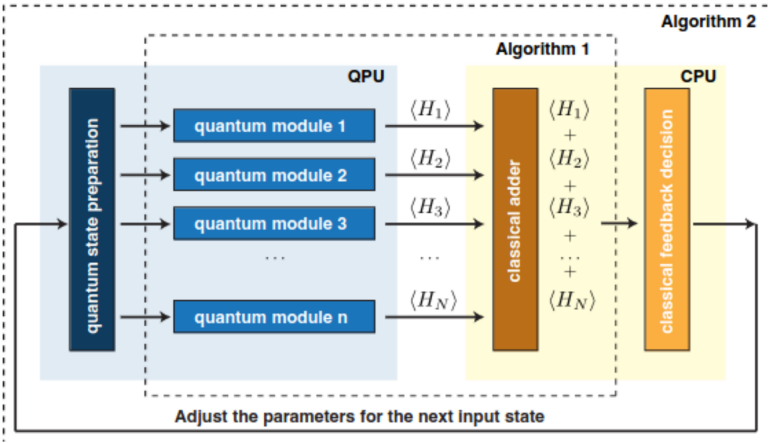
In the framework of Quantum Chemistry, the VQE is primarily aiming at computing the ground-state of molecular systems. More precisely, we want to compute specific features of the ground-state, in particular its *energy*.
To do so, the main operator that will be used in the calculation is the **Hamiltonian** of the molecular system :

$$H(\vec{R}) = \sum_{pq} h_{pq}(\vec{R}) \hat{a}_p^\dagger \hat{a}_q + \sum_{pqrs} h_{pqrs}(\vec{R}) \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s \tag{21}$$

where $\vec{R}$ is a vector representing the positions of the Nuclei in the system.

Variational Quantum Eigensolver (VQE)

The global workflow of the VQE is schematized as below :



Ground-state energy of a molecule

In Quantum Chemistry, the Hamiltonian H is verifying the eigenvalue expression of the **Schrödinger's equation** :

$$H|\Psi_i\rangle = E_i|\Psi_i\rangle \Longrightarrow H = \sum_i E_i|\Psi_i\rangle\langle\Psi_i| \tag{22}$$

where $|\Psi_i\rangle$ are the eigenvectors of H and E_i the associated eigenvalues.

The so-called *ground-state* of a molecule is the state with the minimal energy E_{min}, denoted $|\Psi_{min}\rangle$, i.e. the eigenvector of H with the smallest eigenvalue.

Ground-state energy of a molecule

To compute the expression of E_{min}, we use the Variational Principle of Quantum Mechanics, applied to the expectation value of H :

$$\langle H \rangle_\Psi = \langle \Psi | H | \Psi \rangle = \langle \Psi | \left(\sum_i E_i | \Psi_i \rangle \langle \Psi_i | \right) | \Psi \rangle \tag{23}$$

$$= \sum_i E_i | \langle \Psi | \Psi_i \rangle |^2 \geq \sum_i E_{min} | \langle \Psi | \Psi_i \rangle |^2 \tag{24}$$

$$= E_{min} \langle \Psi | \Psi \rangle \tag{25}$$

Ground-state energy of a molecule

To compute the expression of E_{min}, we use the Variational Principle of Quantum Mechanics, applied to the expectation value of H :

$$\langle H \rangle_\Psi = \langle \Psi | H | \Psi \rangle = \langle \Psi | \left(\sum_i E_i |\Psi_i\rangle \langle \Psi_i| \right) |\Psi\rangle \tag{26}$$

$$= \sum_i E_i |\langle \Psi | \Psi_i \rangle|^2 \geq \sum_i E_{min} |\langle \Psi | \Psi_i \rangle|^2 \tag{27}$$

$$= E_{min} \langle \Psi | \Psi \rangle \tag{28}$$

In definitive, E_{min} is the eigenvalue that minimizes :

$$E_{min} = \min_\Psi \frac{\langle H \rangle_\Psi}{\langle \Psi | \Psi \rangle} \equiv \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \tag{29}$$

Ansatz construction : wave-function connection

From the Variational Principle, we are seeking the ground-state of the molecular system. In practice we have two possibilities :

- find the true ground-state $|\Psi\rangle_{ground}$
- find the best approximation of the ground-state $|\Psi\rangle_{min}$

Here, we define $|\Psi\rangle_{min}$ such that :

$$|\langle\Psi_{min}|\Psi_{ground}\rangle - \langle\Psi_{ground}|\Psi_{ground}\rangle| < \epsilon \tag{30}$$

where ϵ is a well-defined convergence threshold.

To find $|\Psi\rangle_{min}$ variationally, we will use an **ansatz**, a test wave-function that will be additionally parametrized.

Ansatz construction

The ansatz $|\Psi\rangle$ is generally defined as a unitary evolution of a *reference state*, such that :

$$|\Psi\rangle \equiv |\Psi(\vec{\theta})\rangle = U(\vec{\theta})\,|\Psi\rangle_{ref} \tag{31}$$

where $\vec{\theta}$ represents the vector of parameters that will be optimized during the VQE process.

Ansatz construction

The ansatz $|\Psi\rangle$ is generally defined as a unitary evolution of a *reference state*, such that :

$$|\Psi\rangle \equiv |\Psi(\vec{\theta})\rangle = U(\vec{\theta})\,|\Psi\rangle_{ref} \tag{32}$$

where $\vec{\theta}$ represents the vector of parameters that will be optimized during the VQE process.

For the purpose of Quantum Chemistry, usually both the reference state and the unitary evolution operator are chemically-inspired !

Ansatz construction

The ansatz $|\Psi\rangle$ is generally defined as a unitary evolution of a *reference state*, such that :

$$|\Psi\rangle \equiv |\Psi(\vec{\theta})\rangle = U(\vec{\theta})\,|\Psi\rangle_{ref} \tag{33}$$

where $\vec{\theta}$ represents the vector of parameters that will be optimized during the VQE process.

For the purpose of Quantum Chemistry, usually both the reference state and the unitary evolution operator are chemically-inspired !

In everything that will follow, the reference state will be the **Hartree-Fock** state of the molecule, $|\Psi\rangle_{HF}$.

Concerning the evolution operator we require that it is generally defined as :

$$U(\vec{\theta}) = e^{A(\vec{\theta})} \tag{34}$$

in order to assure the unitarity of U, we need $A(\vec{\theta})$ to be Hermitian.

Ansatz construction

We are seeking an operator A that is unitary and chemically-inspired.
From the variety of quantum chemical methods we choose the **Coupled Cluster** method, defined as :

$$T = \sum_n T_n = \sum_{pr} t_p^r a_r^\dagger a_p + \sum_{pqrs} t_{pq}^{rs} a_s^\dagger a_r^\dagger a_q a_p + \dots \tag{35}$$

Ansatz construction

We are seeking an operator A that is unitary and chemically-inspired.
From the variety of quantum chemical methods we choose the **Coupled Cluster** method, defined as :

$$T = \sum_n T_n = \sum_{pr} t_p^r a_r^\dagger a_p + \sum_{pqrs} t_{pq}^{rs} a_s^\dagger a_r^\dagger a_q a_p + \dots \tag{36}$$

More precisely, we are using the specific CCSD method, for Coupled Cluster Single and Double. This method only consider single and double orbital excitations, thus :

$$T_{CCSD} = T_1 + T_2 \begin{cases} T_1 = \sum_{pr} t_p^r a_r^\dagger a_p \\ T_2 = \sum_{pqrs} t_{pq}^{rs} a_s^\dagger a_r^\dagger a_q a_p \end{cases} \tag{37}$$

Ansatz construction

However T_{CCSD} is not hermitian, but we can still construct from it a unitary operator !

$$U_{UCCSD} = e^{T_{CCSD} - T_{CCSD}^{\dagger}} \tag{38}$$

where UCCSD stands for *Unitary Coupled Cluster Single and Double*.

Ansatz construction

However T_{CCSD} is not hermitian, but we can still construct from it a unitary operator !

$$U_{UCCSD} = e^{T_{CCSD} - T_{CCSD}^{\dagger}} \tag{39}$$

where UCCSD stands for *Unitary Coupled Cluster Single and Double*.
The parametrization of this operator is then straightforward :

$$t_p^r \longrightarrow \theta_p^r = t_p^r \cdot \theta \tag{40}$$

$$t_{pq}^{rs} \longrightarrow \theta_{pq}^{rs} = t_{pq}^{rs} \cdot \theta \tag{41}$$

Now we have one parameter for each term in the UCCSD operator.

Fermionic to Qubit mappings

From now we have been working in the second quantization formalism, which cannot be understood by a quantum device. Thus we need to provide a mapping to qubit operators, with the following properties :

Fermionic to Qubit mappings

From now we have been working in the second quantization formalism, which cannot be understood by a quantum device. Thus we need to provide a mapping to qubit operators, with the following properties :

- associate to each orbital operator (creation/annihilation operator) a **Pauli string**
- respect the sum properties
- may use the molecule symmetries

where a Pauli string is a multi-qubit operator acting on each qubit by one Pauli matrix.

Fermionic to Qubit mappings

From now we have been working in the second quantization formalism, which cannot be understood by a quantum device. Thus we need to provide a mapping to qubit operators, with the following properties :

- associate to each orbital operator (creation/annihilation operator) a **Pauli string**
- respect the sum properties
- may use the molecule symmetries

where a Pauli string is a multi-qubit operator acting on each qubit by one Pauli matrix.

Example

For 6 qubits, we can have the following Pauli string :

$$P_6 = \mathbb{I}_0 \otimes X_1 \otimes Y_2 \otimes \mathbb{I}_3 \otimes Z_4 \otimes Z_5 \equiv \mathbb{I}_0 X_1 Y_2 \mathbb{I}_3 Z_4 Z_5 \tag{42}$$

In what follows we will detail one mapping, the **Jordan-Wigner** (the **Bravyi-Kitaev** mapping is explained in the Appendix).

Fermionic to Qubit mappings : Jordan-Wigner

One qubit is associated to a spin-orbital, where $|0\rangle$ corresponds to an unoccupied state while $|1\rangle$ corresponds to an occupied state.
To each orbital operator acting on the jth orbital we associate a Pauli string acting on N-j qubits as :

Fermionic to Qubit mappings : Jordan-Wigner

One qubit is associated to a spin-orbital, where $|0\rangle$ corresponds to an unoccupied state while $|1\rangle$ corresponds to an occupied state.
To each orbital operator acting on the jth orbital we associate a Pauli string acting on N-j qubits as :

$$a_j \longrightarrow \sigma_j^+ \otimes_{k=0}^{j-1} Z_k \tag{43}$$

$$a_j^\dagger \longrightarrow \sigma_j^- \otimes_{k=0}^{j-1} Z_k \tag{44}$$

where we recall that :

$$\sigma^+ = \frac{1}{2}(\sigma_x + i\sigma_y) \equiv \frac{1}{2}(X + iY)$$

$$\sigma^- = \frac{1}{2}(\sigma_x - i\sigma_y) \equiv \frac{1}{2}(X - iY)$$

Fermionic to Qubit mappings : Jordan-Wigner

In the Jordan-Wigner mapping we can easily write the Hamiltonian and the CCSD operators for our molecular system in the qubit basis :

$$\begin{aligned} H &= \sum_{pq} h_{pq} \sigma_p^- \sigma_q^+ \prod_{k=p+1}^{q-1} Z_k + \sum_{pqrs} h_{pqrs} \sigma_p^- \sigma_q^- \sigma_r^+ \sigma_s^+ \prod_{k=p+1}^{q-1} Z_k \prod_{l=r+1}^{s-1} Z_l \\ &= \sum_j h_j P_j \end{aligned} \tag{45}$$

$$T_1(\vec{\theta}) = \sum_{pr} \theta_p^r \sigma_r^- \sigma_p^+ \prod_{k=p+1}^{r-1} Z_k \tag{46}$$

$$T_2(\vec{\theta}) = \sum_{pqrs} \theta_{pq}^{rs} \sigma_s^- \sigma_r^- \sigma_q^+ \sigma_p^+ \prod_{k=p+1}^{r-1} Z_k \prod_{l=q+1}^{s-1} Z_l \tag{47}$$

Hartree-Fock in the qubit space

As we know, the Hartree-Fock state is a good starting point for a quantum algorithm. Thus, let us now indicate how can we systematically define it in the qubit basis.

In Quantum Chemistry by convention, we sort the molecular spin-orbitals left to right from the most energetic orbital to the least energetic one. We shall use the same in the qubit basis.
(Note that this is the inverse as we have previously done in the theoretical construction of Slater determinant.)

H2 in minimal basis with the Jordan-Wigner mapping : HF

Example

In the 2nd quantization formalism :

- atomic orbitals

$$|1s_{A\uparrow}\rangle\,,\ |1s_{A\downarrow}\rangle\,,\ |1s_{B\uparrow}\rangle\,,\ |1s_{B\downarrow}\rangle \tag{48}$$

- molecular orbitals

$$\begin{aligned} |\sigma_{g\uparrow}\rangle &= \frac{1}{\sqrt{N_g}}(|1s_{A\uparrow}\rangle + |1s_{B\uparrow}\rangle) \quad |\sigma_{u\uparrow}\rangle = \frac{1}{\sqrt{N_u}}(|1s_{A\uparrow}\rangle - |1s_{B\uparrow}\rangle \\ |\sigma_{g\downarrow}\rangle &= \frac{1}{\sqrt{N_g}}(|1s_{A\downarrow}\rangle + |1s_{A\downarrow}\rangle) \quad |\sigma_{u\downarrow}\rangle = \frac{1}{\sqrt{N_u}}(|1s_{A\downarrow}\rangle - |1s_{A\downarrow}\rangle) \end{aligned} \tag{49}$$

H2 in minimal basis with the Jordan-Wigner mapping : HF

Example

Then, the final multi-particle state is defined as :

$$|\Psi\rangle = |f_{\sigma_{u\downarrow}}\, f_{\sigma_{u\uparrow}}\, f_{\sigma_{g\downarrow}}\, f_{\sigma_{g\uparrow}}\rangle\,, \quad f_i = \begin{cases} 1, \text{if occupied} \\ 0, \text{if unoccupied} \end{cases} \tag{50}$$

With the JW mapping, we associate one qubit for each spin-orbital above. Thus the multi-qubit state is defined equivalently as above :

$$|\Psi\rangle = |0011\rangle \equiv \mathbb{I}_0\,|0\rangle \otimes \mathbb{I}_1\,|0\rangle \otimes X_2\,|0\rangle \otimes X_3\,|0\rangle \tag{51}$$

Trotterization of the UCCSD operator

Our final goal is to define quantum circuits that are applying the UCCSD unitary operator. However, currently the UCCSD operator is :

$$U_{UCCSD}(\vec{\theta}) = e^{T(\vec{\theta}) - T^{\dagger}(\vec{\theta})} = e^{\sum_j \theta_j P_j - \sum_j \theta_j^* P_j^{\dagger}} \tag{52}$$

where P_j is a Pauli string.

Trotterization of the UCCSD operator

Our final goal is to define quantum circuits that are applying the UCCSD unitary operator. However, currently the UCCSD operator is :

$$U_{UCCSD}(\vec{\theta}) = e^{T(\vec{\theta}) - T^\dagger(\vec{\theta})} = e^{\sum_j \theta_j P_j - \sum_j \theta_j^* P_j^\dagger} \tag{53}$$

where P_j is a Pauli string.

We would like to separate each term involving one Pauli string, from which we can define a circuit. However, we cannot do that directly since, *a priori*, the Pauli strings do not commute with each others.

Trotterization of the UCCSD operator

Our final goal is to define quantum circuits that are applying the UCCSD unitary operator. However, currently the UCCSD operator is :

$$U_{UCCSD}(\vec{\theta}) = e^{T(\vec{\theta}) - T^\dagger(\vec{\theta})} = e^{\sum_j \theta_j P_j - \sum_j \theta_j^* P_j^\dagger} \tag{54}$$

where P_j is a Pauli string.

We would like to separate each term involving one Pauli string, from which we can define a circuit. However, we cannot do that directly since, *a priori*, the Pauli strings do not commute with each others.

In Quantum Mechanics we are granted a powerful tool, called the **Trotterization**.

Trotterization of the UCCSD operator

The **Trotterization** is a process to approximate the exponential of a sum of operators into a product of the same operators, where corrections arise from the non-commutativity between the operators.
It relies on the Lie-Trotter-Suzuki formula :

$$e^{-i\sum_{j=1}^{m}\theta_j T_j} = \left(\prod_{j=1}^{m} e^{-i\frac{\theta_j T_j}{r}}\right)^r + \mathcal{O}\left(\frac{m^2||\theta||^2}{r}\right) \tag{55}$$

Trotterization of the UCCSD operator

The **Trotterization** is a process to approximate the exponential of a sum of operators into a product of the same operators, where corrections arise from the non-commutativity between the operators.
It relies on the Lie-Trotter-Suzuki formula :

$$e^{-i\sum_{j=1}^{m}\theta_j T_j} = \left(\prod_{j=1}^{m} e^{-i\frac{\theta_j T_j}{r}}\right)^r + \mathcal{O}(\frac{m^2||\theta||^2}{r}) \tag{56}$$

For the special case of the UCCSD operator, we usually use the Lie-Trotter-Suzuki formula at order $r = 1$:

$$e^{-i\sum_{j=1}^{m}\theta_j T_j} = \left(\prod_{j=1}^{m} e^{-i\theta_j T_j}\right) + \mathcal{O}(m^2||\theta||^2) \tag{57}$$

Quantum circuit for UCCSD excitations

With the above Trotterization of the UCCSD operator, we have now a product of exponential, where each depends on only one Pauli string and one parameter.
This allow us to construct a quantum circuit for each term systematically. We recall the form of a term :

$$U(\theta) = e^{i\theta^{rs}_{pq} P_{pqrs}} \tag{58}$$

where P_{pqrs} is a Pauli string for either a single or a double UCCSD excitation.

Quantum circuit for UCCSD excitations

From that we can generate a of the associate quantum circuit :

1. Pre-rotations : switch each X and Y Pauli terms in the Z-basis (computational basis) by rotating their current basis.
   For the X basis $\Longrightarrow$ apply a Hadamard gate.
   For the Y basis $\Longrightarrow$ apply a $R_x(\frac{\pi}{2})$ gate

Quantum circuit for UCCSD excitations

From that we can generate a schematic construction of the associate quantum circuit :

1. Pre-rotations : switch each X and Y Pauli terms in the Z-basis (computational basis) by rotating their current basis.
   For the X basis $\Longrightarrow$ apply a Hadamard gate.
   For the Y basis $\Longrightarrow$ apply a $R_x(\frac{\pi}{2})$ gate
2. Entanglement : two adjacent Pauli terms are connected by entanglement with a CNOT gate

Quantum circuit for UCCSD excitations

From that we can generate a schematic construction of the associate quantum circuit :

1. Pre-rotations : switch each X and Y Pauli terms in the Z-basis (computational basis) by rotating their current basis.
   For the X basis $\Longrightarrow$ apply a Hadamard gate.
   For the Y basis $\Longrightarrow$ apply a $R_x(\frac{\pi}{2})$ gate
2. Entanglement : two adjacent Pauli terms are connected by entanglement with a CNOT gate
3. Applied a R_z rotation to the last qubit of the Pauli string where the angle is the variational parameter $R_z(\pm\theta)$

Quantum circuit for UCCSD excitations

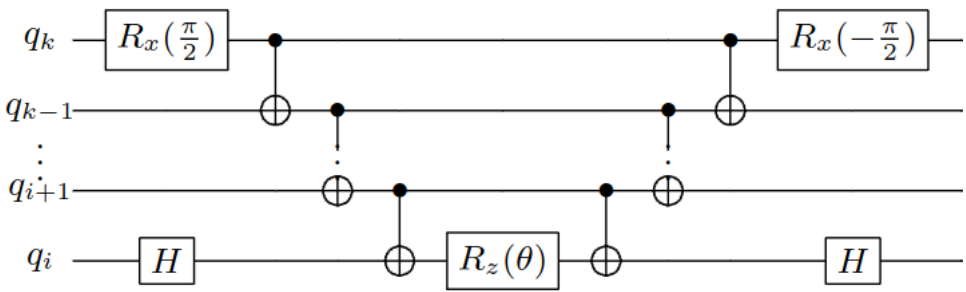
From that we can generate a schematic construction of the associate quantum circuit :

1. Pre-rotations : switch each X and Y Pauli terms in the Z-basis (computational basis) by rotating their current basis.
   For the X basis $\Longrightarrow$ apply a Hadamard gate.
   For the Y basis $\Longrightarrow$ apply a $R_x(\frac{\pi}{2})$ gate
2. Entanglement : two adjacent Pauli terms are connected by entanglement with a CNOT gate
3. Applied a R_z rotation to the last qubit of the Pauli string where the angle is the variational parameter $R_z(\pm\theta)$
4. Inverse Entanglement
5. Post-rotations : switch back each X and Y Pauli terms in their respective basis

Quantum circuit for UCCSD exctations : single

We recall that a single UCCSD excitation is defined as :

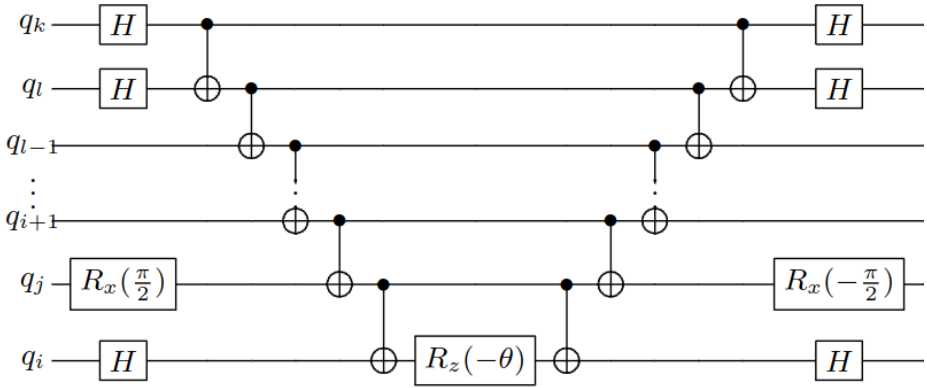
$$U(\theta) = e^{i\theta\left(Y_k X_i \prod_{n=i+1}^{k-1} Z_n\right)} \tag{59}$$



Quantum circuit for UCCSD exctations : double

We recall that a double UCCSD excitation is defined as :

$$U(\theta) = e^{-i\theta\left(X_k X_l Y_j X_i \prod_{n=l+1}^{k-1} Z_n \prod_{m=i+1}^{j-1} Z_m\right)} \tag{60}$$



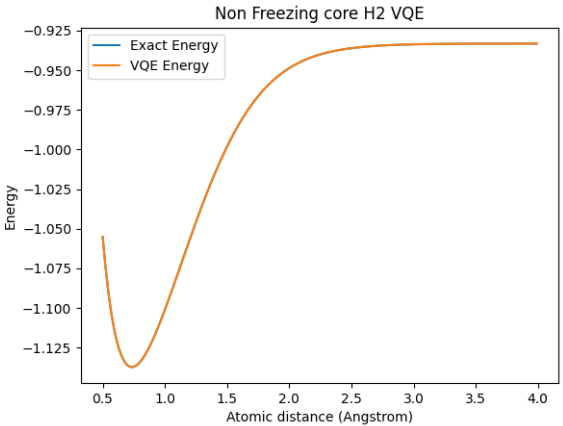
Explicit workflow of the Variational Quantum Eigensolver

Qubit PHARMACEUTICALS

Before going forward, let us sum-up the details about ther VQE workflow :

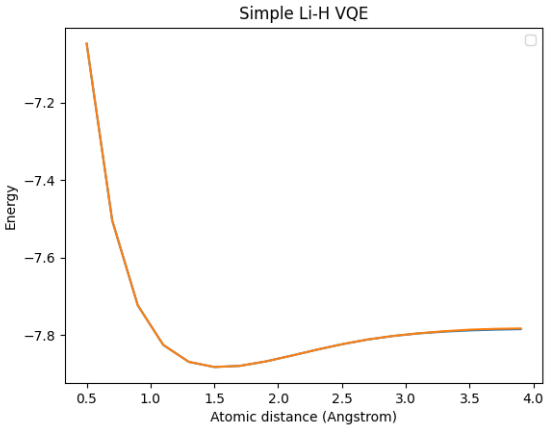
1. Setup the molecular Hamiltonian and the UCCSD operator in the 2nd quantization formalism
2. Map them in the Qubit formalism through the JW or BK mapping
3. Compute the expectation value of each term of the Hamiltonian with respect to the UCCSD ansatz $\{\langle H_i \rangle_{\theta_i}\}_{i \in \{1,\dots,N\}}$
4. Add all measured expectation values to form the total energy of the state $\langle H \rangle_{\vec{\theta}} = \sum_i \langle H_i \rangle_{\theta_i}$
5. Optimized all variational parameters $\vec{\theta^*} = \arg\min_{\vec{\theta}} \langle H \rangle_{\vec{\theta}}$
6. Compute the optimized energy : $E^* = \langle H \rangle_{\vec{\theta^*}}$
7. Back and forth until convergence or stopping criteria met

Some VQE results



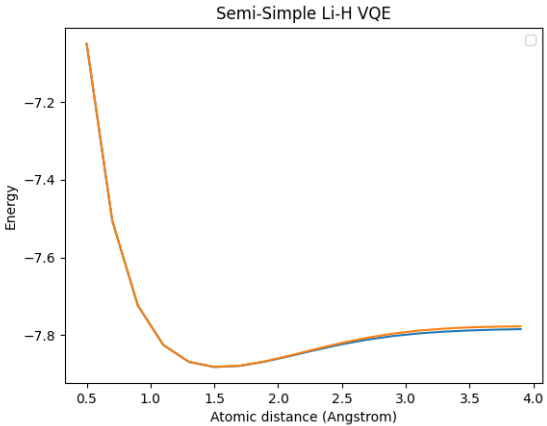
- Non Freezing core $\Longrightarrow$ all electrons are consider
- Use of COBYLA optimizer : gradient-free optimizer using a linear approximation of the cost function around each data point
- Energy difference with FCI below 10^{-10}
- less than one minute of runtime for one energy
- very shallow circuit

Some VQE results



- Non Freezing core $\Longrightarrow$ all electrons are consider
- Use of COBYLA optimizer : gradient-free optimizer
- Energy difference with FCI below 10^{-3} (changing after 3.5 Angstrôm)
- several minutes of runtime for one energy
- more than 1000 operators

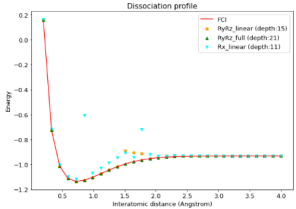
Some VQE results



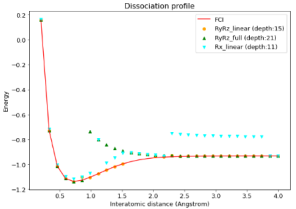
- Freezing core $\Longrightarrow$ core electrons of Li atom not consider
- Use of COBYLA optimizer : gradient-free optimizer
- Energy difference with FCI below 10^{-2} (changing after 2.5 Angström)
- several minutes of runtime for one energy
- cutting operator numbers by two (around 700)

Some VQE results

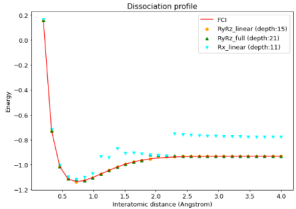
Importance of the optimization method



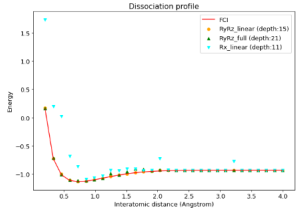
(a) COBYLA



(b) L_BFGS_B



(c) SLSQP



(d) SPSA

Limitations of the VQE

Even though the VQE is a quite successful algorithm for the NISQ era it suffers from several problems :

- It only computes an approximation of the ground-state. Indeed it relies on a quantum chemistry method, here UCCSD, which does not encompass all of the excitations in a molecular system

Limitations of the VQE

Even though the VQE is a quite successful algorithm for the NISQ era it suffers from several problems :

- It only computes an approximation of the ground-state. Indeed it relies on a quantum chemistry method, here UCCSD, which does not encompass all of the excitations in a molecular system
- It computes all the terms of the evolution operator. The number of these terms grows very quickly with more complex systems, leading to not pratical circuits for NISQ devices

Limitations of the VQE

Even though the VQE is a quite successful algorithm for the NISQ era it suffers from several problems :

- It only computes an approximation of the ground-state. Indeed it relies on a quantum chemistry method, here UCCSD, which does not encompass all of the excitations in a molecular system
- It computes all the terms of the evolution operator. The number of these terms grows very quickly with more complex systems, leading to not pratical circuits for NISQ devices
- Limited by the classical optimization since we have one parameter per term in the evolution operator

Improvements of the VQE

So we need to make significant modifications of the VQE algorithm in order to solves these issues :

- Allow new Ansatze without overloading the circuits and consequently the optimization process

Improvements of the VQE

So we need to make significant modifications of the VQE algorithm in order to solves these issues :

- Allow new Ansatze without overloading the circuits and consequently the optimization process
- Reduce the depth of the quantum circuit with keeping a relevant precision in the simulation results

Improvements of the VQE

Qubit PHARMACEUTICALS

So we need to make significant modifications of the VQE algorithm in order to solves these issues :

- Allow new Ansatze without overloading the circuits and consequently the optimization process
- Reduce the depth of the quantum circuit with keeping a relevant precision in the simulation results
- have a workflow that will reduce the computational charge of the QPU and/or of the CPu units

Improvements of the VQE

So we need to make significant modifications of the VQE algorithm in order to solves these issues :

- Allows new Ansatze without overloading the circuits and consequently the optimization process
- Redce the depth of the quantum circuit with keeping a relevant precision in the simulation results
- have a workflow that will reduce the computational charge of the QPU and/or of the CPu units

This is what leads us to the famous modification of the VQE : **ADAPT-VQE** !

We are here

ADAPT-VQE

The **Adaptive Derivative-Assembled Pseudo-Trotter ansatz Variational Quantum Eigensolver** (ADAPT-VQE) algorithm is a modification of the VQE by adding an adaptive feature.

ADAPT-VQE

The **Adaptive Derivative-Assembled Pseudo-Trotter ansatz Variational Quantum Eigensolver** (ADAPT-VQE) algorithm is a modification of the VQE by adding an adaptive feature.
It aims at approximating at best the FCI wave-function with a maximally compact ansatz. To do so, it relies in the use of an *Operator Pool*, a pool containing several operators defined in a specific way.

ADAPT-VQE

The **Adaptive Derivative-Assembled Pseudo-Trotter ansatz Variational Quantum Eigensolver** (ADAPT-VQE) algorithm is a modification of the VQE by adding an adaptive feature.
It aims at approximating at best the FCI wave-function with a maximally compact ansatz. To do so, it relies in the use of an *Operator Pool*, a pool containing several operators defined in a specific way.

At each step of the algorithm, we compute the gradients of the Energy (the expectation value of the Hamiltonian) with respect to the variational parameter of each operator.

ADAPT-VQE

The **Adaptive Derivative-Assembled Pseudo-Trotter ansatz Variational Quantum Eigensolver** (ADAPT-VQE) algorithm is a modification of the VQE by adding an adaptive feature.
It aims at approximating at best the FCI wave-function with a maximally compact ansatz. To do so, it relies in the use of an *Operator Pool*, a pool containing several operators defined in a specific way.

At each step of the algorithm, we compute the gradients of the Energy (the expectation value of the Hamiltonian) with respect to the variational parameter of each operator. Keeping the most impactful operator, i.e. the operator with the most valued gradient, we add it to the ansatz. At the end of step, we perform a VQE process to get the optimized parameters.

ADAPT-VQE

The **Adaptive Derivative-Assembled Pseudo-Trotter ansatz Variational Quantum Eigensolver** (ADAPT-VQE) algorithm is a modification of the VQE by adding an adaptive feature.
It aims at approximating at best the FCI wave-function with a maximally compact ansatz. To do so, it relies in the use of an *Operator Pool*, a pool containing several operators defined in a specific way.

At each step of the algorithm, we compute the gradients of the Energy (the expectation value of the Hamiltonian) with respect to the variational parameter of each operator. Keeping the most impactful operator, i.e. the operator with the most valued gradient, we add it to the ansatz. At the end of step, we perform a VQE process to get the optimized parameters.

The ansatz is thus created **iteratively** until we reach convergence or a stopping criteria.

ADAPT-VQE : computing a gradient on the QPU

How can we compute the gradient of an operator (here the Hamiltonian) with respect to a parameter on a quantum computer ?
We recall that a unitary operator is defined as :

$$U = e^{i\theta A} \tag{61}$$

where A is an hermitian operator.

ADAPT-VQE : computing a gradient on the QPU

How can we compute the gradient of an operator (here the Hamiltonian) with respect to a parameter on a quantum computer ?
We recall that a unitary operator is defined as :

$$U = e^{i\theta A} \tag{62}$$

where A is an hermitian operator.
We know that we can compute expectation values, thus we aim at relating the gradient to an expectation value. Let's do that !

ADAPT-VQE : computing a gradient on the QPU

Qubit PHARMACEUTICALS

Let's do that !

$$\frac{\partial \langle H \rangle}{\partial \theta} = \frac{\partial}{\partial \theta} \langle \Psi(\theta)|H|\Psi(\theta)\rangle = \frac{\partial}{\partial \theta} \langle 0|e^{-i\theta A^\dagger} H e^{i\theta A}|0\rangle \tag{63}$$

$$= -i \langle 0|e^{-i\theta A^\dagger} A^\dagger H e^{i\theta A}|0\rangle + i \langle 0|e^{-i\theta A^\dagger} H A e^{i\theta \dagger}|0\rangle \tag{64}$$

$$= i \langle \Psi(\theta)|(HA - A^\dagger H)|\Psi(\theta)\rangle = \{A^\dagger = A\} \tag{65}$$

$$= i \langle \Psi(\theta)|(HA - AH)|\Psi(\theta)\rangle = i \langle \Psi(\theta)|[H, A]|\Psi(\theta)\rangle \tag{66}$$

$$\equiv \langle [H, A] \rangle_{\Psi(\theta)} \tag{67}$$

ADAPT-VQE : computing a gradient on the QPU

Qubit PHARMACEUTICALS

Let's do that !

$$\frac{\partial \langle H \rangle}{\partial \theta} = \frac{\partial}{\partial \theta} \langle \Psi(\theta)|H|\Psi(\theta)\rangle = \frac{\partial}{\partial \theta} \langle 0|e^{-i\theta A^\dagger} H e^{i\theta A}|0\rangle \tag{68}$$

$$= -i \langle 0|e^{-i\theta A^\dagger} A^\dagger H e^{i\theta A}|0\rangle + i \langle 0|e^{-i\theta A^\dagger} H A e^{i\theta \dagger}|0\rangle \tag{69}$$

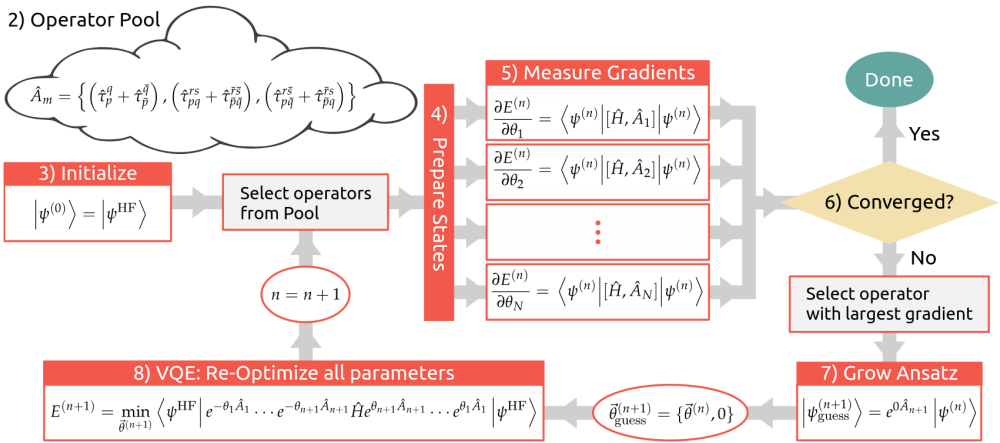
$$= i \langle \Psi(\theta)|(HA - A^\dagger H)|\Psi(\theta)\rangle = \{A^\dagger = A\} \tag{70}$$

$$= i \langle \Psi(\theta)|(HA - AH)|\Psi(\theta)\rangle = i \langle \Psi(\theta)|[H, A]|\Psi(\theta)\rangle \tag{71}$$

$$\equiv \langle [H, A] \rangle_{\Psi(\theta))} \tag{72}$$

Hence, the gradient of an operator with respect to a parameter is simply the expectation value of the commutator between this operator and evolution operator associated with the parameter !
This can be easily computed on a QPU, which is done during the ADAPT-VQE algorithm.

ADAPT-VQE : Workflow



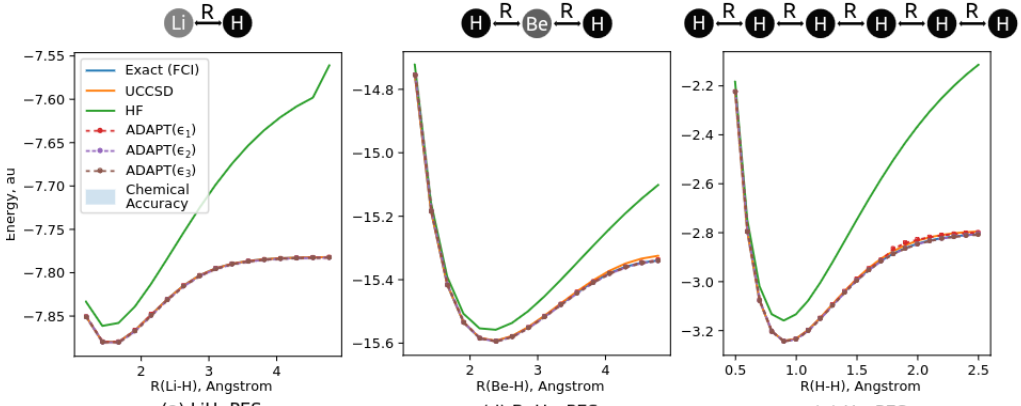
Limitations of ADAPT-VQE

ADAPT-VQE is a successful modification of VQE in terms a shallower circuits leading to :

- way less stress on the QPU
- more CPU usage, but compensated with less parameters to optimize
- faster convergence and smaller running times

Some ADAPT-VQE results

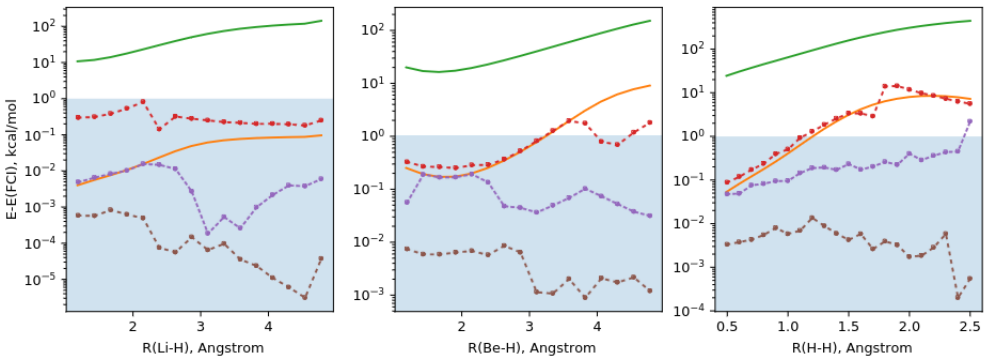
PES curves with ADAPT-VQE



(a) LiH: PES (d) BeH_2: PES (g) H_6: PES

Some ADAPT-VQE results

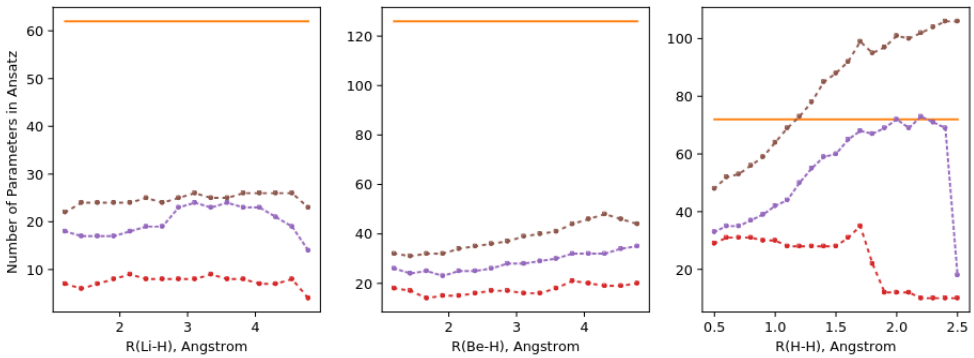
Error curves with ADAPT-VQE



(b) LiH: Error (e) BeH_2: Error (h) H_6: Error

Some ADAPT-VQE results

Number of parameters used in ADAPT-VQE



(c) LiH: Number Parameters

(f) BeH_2: Number of Parameters

(i) H_6: Number of Parameters

Limitations of ADAPT-VQE

ADAPT-VQE is a successful modification of VQE in terms a shallower circuits leading to :

- way less stress on the QPU
- more CPU usage, but compensated with less parameters to optimize
- faster convergence and smaller running times

However, it still suffers from several issues that are quite impactful on its ability to tackle quantum advantage :

Limitations of ADAPT-VQE

ADAPT-VQE is a successful modification of VQE in terms a shallower circuits leading to :

- way less stress on the QPU
- more CPU usage, but compensated with less parameters to optimize
- faster convergence and smaller running times

However, it still suffers from several issues that are quite impactful on its ability to tackle quantum advantage :

- defining the Operator Pool is extremely important and can be a hard during the theoretical setup of the algorithm

Limitations of ADAPT-VQE

ADAPT-VQE is a successful modification of VQE in terms a shallower circuits leading to :

- way less stress on the QPU
- more CPU usage, but compensated with less parameters to optimize
- faster convergence and smaller running times

However, it still suffers from several issues that are quite impactful on its ability to tackle quantum advantage :

- defining the Operator Pool is extremely important and can be a hard during the theoretical setup of the algorithm
- sensible to local minimas where it can stay trapped for multiple steps, known as the famous *Barren Plateaus*

Limitations of ADAPT-VQE

ADAPT-VQE is a successful modification of VQE in terms a shallower circuits leading to :

- way less stress on the QPU
- more CPU usage, but compensated with less parameters to optimize
- faster convergence and smaller running times

However, it still suffers from several issues that are quite impactful on its ability to tackle quantum advantage :

- defining the Operator Pool is extremely important and can be a hard during the theoretical setup of the algorithm
- sensible to local minimas where it can stay trapped for multiple steps, known as the famous *Barren Plateaus*
- still relies on approximating the FCI wave-function (in the Operator Pool's definition)

Limitations of ADAPT-VQE : Barren Plateaus

A **Barren Plateaus** is a plateau appearing in optimization processes, where for several steps in a row the optimization is stuck in a local minima of the landscape.
In particular, they can be dominant for any gradient-descent optimization processes, since by construction a gradient is quite sensible to local extremas.

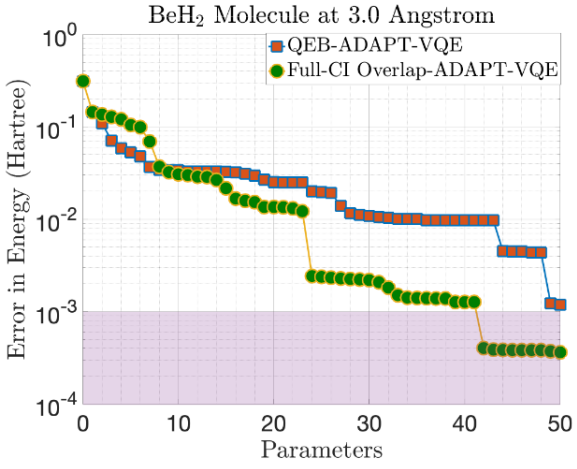
Limitations of ADAPT-VQE : Barren Plateaus

Qubit PHARMACEUTICALS

A **Barren Plateaus** is a plateau appearing in optimization processes, where for several steps in a row the optimization is stuck in a local minima of the landscape.
In particular, they can be dominant for any gradient-descent optimization processes, since by construction a gradient is quite sensible to local extremas.

For variational quantum algorithms, Barren plateaus appear in a different scenarii.
In ADAPT-VQE, they appear several times such that when constructing the ansatz, for several steps the calculated gradients are not updating the energy calculation.

Limitations of ADAPT-VQE : Barren Plateaus

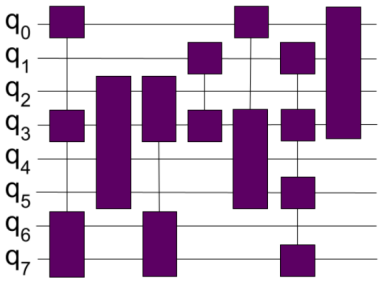


Descendants of ADAPT-VQE

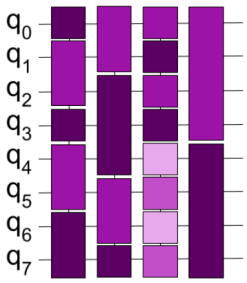
Shortly after the ADAPT-VQE article, several modifications have been made to tackle the more problematic issues of ADAPT.
Amongst them, two modifications are worth noticing :

- **TETRIS** : stands for *Tilling Efficient Trial circuits with Rotations Impletemented Simultaneously*, it modifies how the ansatz is constructed to always use dense circuits
- **QEB** : stands for *Qubit Excitations Based*, it is a truncation of the fermionic operators to only work in the subspace of the active qubits

Descendants of ADAPT-VQE : TETRIS ADAPT-VQE



ADAPT-VQE



TETRIS-ADAPT-VQE

Darkest blocks : operator corresponding to the standard ADAPT-VQE, i.e. with largest gradient

Lighter blocks : operators obtained from beyond the largest gradient

Speed-up in terms of optimization steps and shallower circuits

Descendants of ADAPT-VQE : Qubit Excitations Based

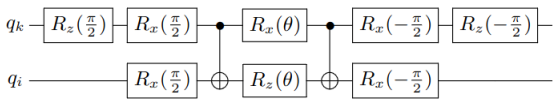
Qubit PHARMACEUTICALS

Qubit Excitations based (QEB) is a method to create a new Operator Pool from the fermionic one.

It relies on reducing the CNOT cost of the fermionic excitations using a so-called *parafermionic* expression : we now work inside the subspace generated by the excitation-active qubits only.

Single QEB

$$\tilde{A}_{ik}(\theta) = exp[i\frac{\theta}{2}(X_i Y_k - Y_i X_k)] \tag{73}$$



Descendants of ADAPT-VQE : Qubit Excitations Based

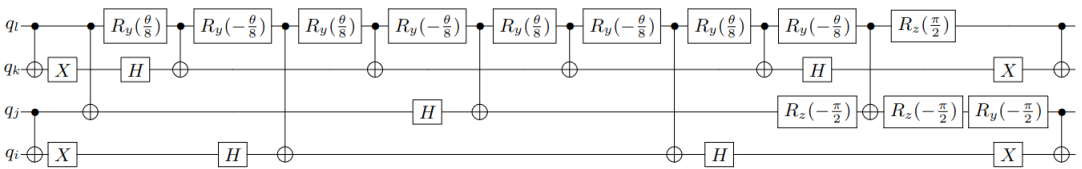
Qubit PHARMACEUTICALS

Qubit Excitations (QEB) based is a method to create a new Operator Pool, from the fermionic one.

It relies on reducing the CNOT cost of the fermionic excitations using a so-called *parafermionic* expression : we now work inside the subspace generated by the excitation-active qubits only.

Double QEB

$$\tilde{A}_{ijkl}(\theta) = exp[i\frac{\theta}{8}(X_iY_jX_kX_l + Y_iX_jX_kX_l + Y_iY_jY_kX_l + Y_iY_jX_kY_l - X_iX_jY_kX_l - X_iX_jX_kY_l - Y_iX_jY_kY_l - X_iY_jY_kY_l)] \quad (74)$$



overlap ADAPT-VQE

overlap ADAPT-VQE is a specific modfication of ADAPT-VQE made by the Research Team of Qubit Pharmaceuticals.
It consists in growing the ansatz not by working directly with the energy of the ground-state but by *maximizing the overlap between the created ansatz and an intermediate known state*.
The energy is then calculated when the overlap process arrives at its end.

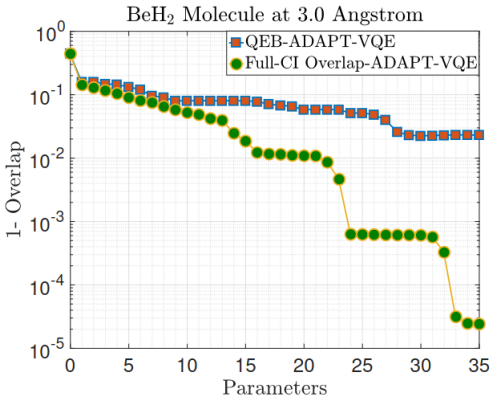
overlap ADAPT-VQE

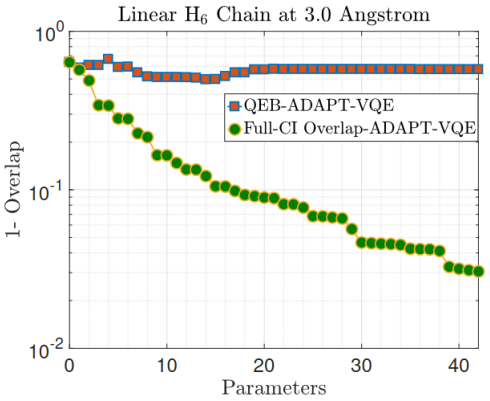
overlap ADAPT-VQE is a specific modfication of ADAPT-VQE made by the Research Team of Qubit Pharmaceuticals.
It consists in growing the ansatz not by working directly with the energy of the ground-state but by *maximizing the overlap between the created ansatz and an intermediate known state*.
The energy is then calculated when the overlap process arrives at its end.

This new method is providing us great advantages :

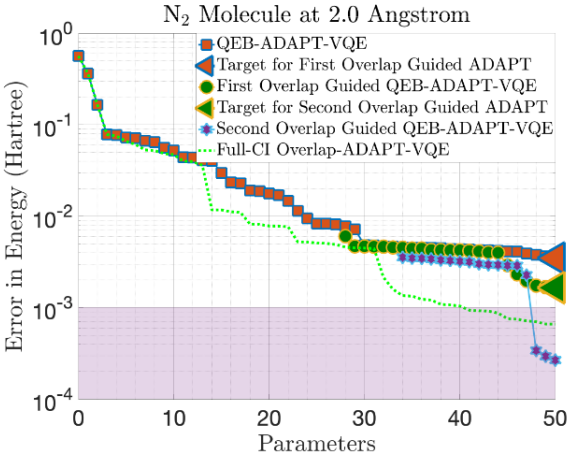
- avoiding, or at least reducing the size of, Barren Plateaus
- faster algorithmic convergence to chemical accuracy simulations
- shallowing more the constructed ansatze
- efficient state preparation since not using FCI target wave-function

overlap ADAPT-VQE





overlap ADAPT-VQE



Outperform every ADAPT-VQE versions in terms of

- convergence runtime
- number of operators/parameters
- resistance on Barren Plateaus

Can be used as an initial-state preparation:

- for another overlap ADAPT-VQE algorithm
- as a quantum state preparation for a QPE algorithm

overlap ADAPT-VQE on a real quantum computer

Qubit PHARMACEUTICALS

Greedy Gradient-free Adaptive Variational Quantum Algorithms on a Noisy Intermediate Scale Quantum Computer

César Feniou[1,2], Baptiste Claudon[2], Muhammad Hassan[3], Axel Courtat[2], Olivier Adjoua[1], Yvon Maday[3,4], and Jean-Philip Piquemal[1,2,*]

[1]Sorbonne Université, Laboratoire de Chimie Théorique (UMR-7616-CNRS), F-75005 Paris, France
[2]Qubit Pharmaceuticals, Advanced Research Department, Paris, France
[3]Sorbonne Université, CNRS, Université Paris Cité, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France
[4]Institut Universitaire de France, 75005, Paris, France
[*]jean-philip.piquemal@sorbonne-universite.fr

We implemented a **"greedy" modification** of both ADAPT-VQE and overlap ADAPT-VQE into a real QPU $\Longrightarrow$ hardware-efficient implementation to deliver real results on a NISQ device.

overlap ADAPT-VQE on a real quantum computer

Qubit PHARMACEUTICALS

Best results where for the **1D transverse Ising model** with open boundary conditions:

$$\mathbf{H} = h \sum_{i=0}^{n-1} X_i + J \sum_{i=0}^{n-1} Z_i Z_{i+1} \tag{75}$$

where h and J are system parameters.
We used 25 qubits (equivalent for 25 spins in a 1D linear chain), implemented on the **Aria** device through Amazon Braket SDK.

overlap ADAPT-VQE on a real quantum computer

Qubit PHARMACEUTICALS

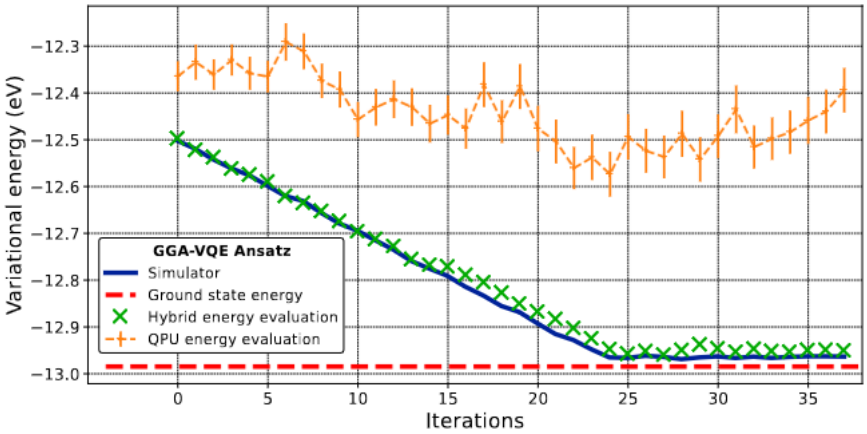
Best results where for the **1D transverse Ising model** with open boundary conditions:

$$\mathbf{H} = h \sum_{i=0}^{n-1} X_i + J \sum_{i=0}^{n-1} Z_i Z_{i+1} \tag{76}$$

where h and J are system parameters.
We used 25 qubits (equivalent for 25 spins in a 1D linear chain), implemented on the **Aria** device through Amazon Braket SDK.
Aria is a ion-trapped quantum computer developed by **IonQ**, with the properties:

- 25 qubits with all-to-all connectivity
- best 1qubit and 2qubit gates error rate ($< 10^{-4}$ and $< 10^{-3}$ respectively)
- longest coherence time T1 and T2 (more than 1s)

overlap ADAPT-VQE on a real quantum computer



correcting QPU results with Hyperion

As seen from above, we have made a so-called **hybrid evaluation** (green points). It consisted of using all QPU datas at each iteration and compute numerically the final results on a supercomputer.

This corresponds between a error-mitigation and a classical error-correction method. It was used primarly to confirm that the algorithm was behaving well on real QPU, and it dit!

correcting QPU results with Hyperion

As seen from above, we have made a so-called **hybrid evaluation** (green points). It consisted of using all QPU datas at each iteration and compute numerically the final results on a supercomputer.
This corresponds between a error-mitigation and a classical error-correction method. It was used primarly to confirm that the algorithm was behaving well on real QPU, and it dit!

For that we used an internal project, the **Hyperion emulator**: highly-parallel CPU/GPU architecture for quantum chemistry

HYPERION

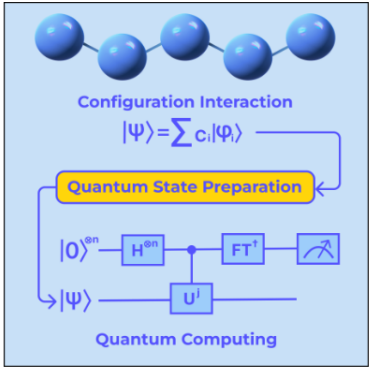
Proprietary Quantum Simulation Platform. Development of next generation algorithms for quantum chemistry & drug discovery.

30+ qubits

Hybrid HPC-QC ready for computations over 30 qubits

overlap ADAPT-VQE as a quantum state preparation

A primordial key feature in any quantum chemistry algorithm on a quantum computer is the **quantum state preparation**: preparing a correct and usable initial state.



overlap ADAPT-VQE algorithm can do that!

- prepare better initial state for VQE algorithm intented to deliver precise results
- as a quantum state preparation part of the Quantum Phase Estimation Algorithm (QPE)

We are here

Bibliography

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] Peruzzo A et al. A variational eigenvalue solver on a phtonic quantum processor. *Nature Communications*, 5:4213, 2014.

[3] Michael A. Nielsen. The fermionic canonical commutation relations and the jordan-wigner transform. 2005.

[4] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, 12 2012.

[5] Economou S.E. Barnes E Mayhall N.J. Grimsley, H.R. An adaptive variational algorithm for exact molecular simulations n a quantum computer. *Nature Communications*, 10:3007.

[6] Yordan S. Yordanov, David R. M. Arvidsson-Shukur, and Crispin H. W. Barnes. Efficient quantum circuits for quantum computational chemistry. *Phys. Rev. A*, 102:062612, Dec 2020.

[7] Panagiotis G Anastasiou, Yanzhu Chen, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. Tetris-adapt-vqe: An adaptive algorithm that yields shallower, denser circuit ans\" atze. *arXiv preprint arXiv:2209.10562*, 2022.

[8] Yordan Yordanov, Vasilis Armaos, Crispin Barnes, and David Shukur. Qubit-excitation-based adaptive variational quantum eigensolver. *Communications Physics*, 4:228, 10 2021.

[9] Feniou C et al. Overlap-adapt-vqe: practical quantum chemistry on quantum computers via overlap-guided compact ansätze. *Communications Physics*, 6:192, 2023.

[10] César Feniou, Baptiste Claudon, Muhammad Hassan, Axel Courtat, Olivier Adjoua, Yvon Maday, and Jean-Philip Piquemal. Greedy Gradient-free Adaptive Variational Quantum Algorithms on a Noisy Intermediate Scale Quantum Computer. 6 2023.

[11] C. Feniou, B. Claudon, J. Zylberman, O. Adjoua, E. Giner, and J. P. Piquemal. Sparse Quantum State Preparation for Strongly Correlated Systems. 11 2023.