Physics Project I
Report: Adapt-VQE

Gaia Stella Bolognini, #342188

Supervisors: Professor Giuseppe Carleo, Patrick Hümbeli

Academic year 2021/2022

Contents

1 Introduction

The aim of the project was to study different optimization techniques to improve the Variational Quantum Eigensolver (VQE), whose aim is to find the ground state of a physical Hamiltonian. Starting from a trial variational circuit that is an assumption for the target wavefunction, the parameters of the state are updated in order to minimize the energy of the system. The accuracy and the correctness of the algorithm is highly dependent on the choice of the pre-selected ansatz state; this problem was addressed through the use of the Adaptive Derivative-Assemble Pseudo-Trotter ansatz Variational Quantum eigensolver (Adapt-VQE). After having studied some molecular problems (LiH and H_2), we focused on the Transverse Field Ising Model where this approach was applied in different contexts. In order to perform the optimization, the standard Gradient Descent and the Quantum Natural Gradient were used and their effect compared. Finally, we tried to use the Hessian of the loss function in order to improve the Adapt-VQE algorithm with some information regarding the local curvature of the Hamiltonian but in the end this method was not effective. In all the trials, the ground state of the Hamiltonian was approached; we show that the success of the optimization is influenced by the size of the operator pool (i.e. the amount of operators that are allowed to build the ansatz) and that on average taking combinations of different operators can lead to a faster convergence. Nevertheless, the circuit depth cannot be reduced by taking combination of different operators; the standard Adapt-VQE leads to shallower circuits that are better suited for NISQ devices.
The codes and the algorithms used were written with the Python library Pennylane [2] and were run on quantum simulators. They are available on GitHub [1].

2 Theoretical introduction: the VQE and the Adapt-VQE algorithm

The **VQE** [11] is a quantum-classical computational approach to find the ground state of a Hamiltonian; it has been proposed as an alternative to fully quantum algorithms, which require quantum hardwares that may be not accessible in the near term future. It bases itself on the so-called **Variational principle** in quantum mechanics:

$$\frac{\left\langle \psi(\vec{\theta}) \right| \hat{H} \left| \psi(\vec{\theta}) \right\rangle}{\left\langle \psi(\vec{\theta}) \middle| \psi(\vec{\theta}) \right\rangle} \geq E_0 \tag{1}$$

where the state $|\psi(\theta)\rangle$ is a trail wave function, defined as a variational form, dependent on some parameters $\vec{\theta}$, and E_0 is the ground state of the hamiltonian $\hat{H}$. It is possible to minimize the energy of the system by optimizing the parameters of the trial wavefunction to approach the ground state. This is performed through the use of a classical optimizer; the algorithm is known to be hybrid, as only the preparation of the state and the evaluation of the energy require quantum computation.

2.1 The Hamiltonian

As we consider the Born-Hoppenheimer approximation, in quantum chemestry we mainly focus on the electronic **Hamiltonian** of the system. This is usually written in second quantization, taking into account possible one- and two- electron excitations. In order to evaluate the energy on a quantum computer it is necessary to map this Hamiltonian into one for distinguishable qubits, using the **Jordan-Wigner** [16] or **Bravyi-Kitaev** [6] transformations. The resulting Hamiltonian will be in the form:

$$H_{el} = \sum_j \alpha_j P_j = \sum_j \alpha_j \prod_i \sigma_i^j \tag{2}$$

where P_j are Pauli matrices strings, products of σ_i, and α_j are real scalar coefficients that depend on the intial second-quantized Hamiltonian [7]. In addition to molecular optimization problems, the VQE can be used to find the ground state of a general Hamiltonian, written always in terms of combinations and products of Pauli matrices.

2.2 Initial ansatz

The accuracy of the algorithm is highly sensitive to the choice of the **initial ansatz state** and its explicit form: the exact energy is obtained if the ansatz is capable of representing any state in the subspace that contains the ground state, using for example some simmetry knowledge [14]. The ansatz generally allows one to obtain only approximations to the ground state and the quality of the VQE simulation is as good as the ansatz.

Different approaches have been studied to find the best trial state, that would imply to obtain a solution with high accuracy and few parameters. Three of them are the most relevant ones [7]: the chemestry inspired ansatz, where every term of the ansatz describes a certain electron configuration, the hardware efficient ansatz, where only limited number of gates that can be efficiently implemented on a quantum hardware are selected, and the the Hamiltonian variational ansatz [15], where exponentialization of different terms present in the Hamiltonian contribute to the trial state.

One of the most used approaches is the **Unitary Coupled Cluster Theory** [4] with singles and doubles (UCCSD), which is a chemestry-inspired ansatz. In this approach, trial states are generated by applying to a reference state, usually chosen as the Hartree-Fock state, an operator written in terms of exponential of a sum of single and double excitations:

$$\left|\psi(\vec{\theta})\right\rangle = U(\vec{\theta})\left|\phi\right\rangle = e^{T(\vec{\theta})-T(\vec{\theta})^{\dagger}}\left|\phi\right\rangle \tag{3}$$

where $T = \sum_i T_i$ is a sum of fermionic excitation operators.
Each of these operators is still expressed through the use of the second quantization method (annihilation and creation operators) generally up to second order, as higher orders may become too expensive. In order to decompose the $U(\vec{\theta})$ into operators that can be implemented on a quantum computer, the Trotter Decomposition is used, leading to an ordered product of individual exponentials that is in general just an approximation since individual excitations do not commute. Then, these fermionic operators are mapped into Pauli matrices, native for a quantum computer, through the Jordan-Wigner [16] or Bravyi-Kitaev transformations [6].

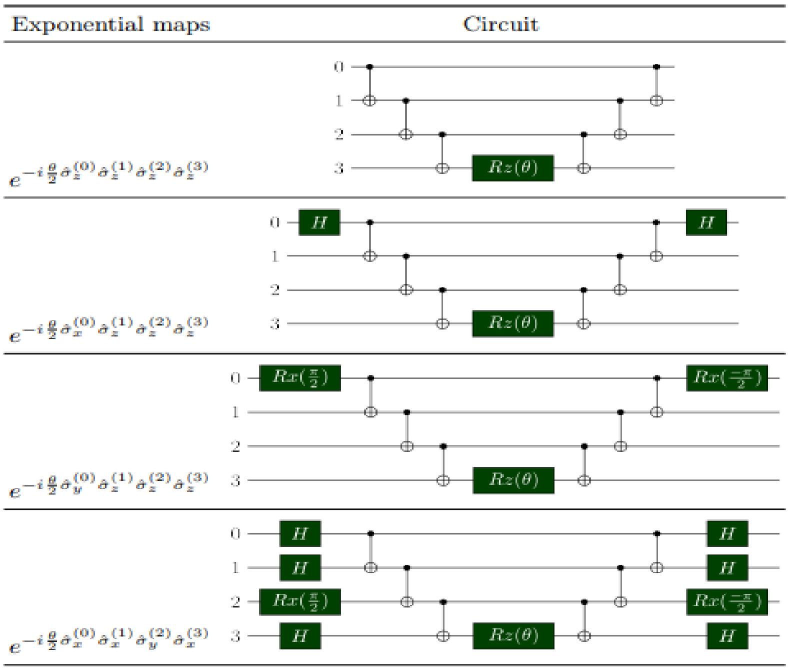
Exponential maps
Circuit
$e^{-i\frac{\theta}{2}\hat{\sigma}_z^{(0)}\hat{\sigma}_z^{(1)}\hat{\sigma}_z^{(2)}\hat{\sigma}_z^{(3)}}$
0
1
2
3
$Rz(\theta)$
$e^{-i\frac{\theta}{2}\hat{\sigma}_x^{(0)}\hat{\sigma}_z^{(1)}\hat{\sigma}_z^{(2)}\hat{\sigma}_z^{(3)}}$
H
$Rz(\theta)$
H
$e^{-i\frac{\theta}{2}\hat{\sigma}_y^{(0)}\hat{\sigma}_z^{(1)}\hat{\sigma}_z^{(2)}\hat{\sigma}_z^{(3)}}$
$Rx(\frac{\pi}{2})$
$Rz(\theta)$
$Rx(\frac{-\pi}{2})$
$e^{-i\frac{\theta}{2}\hat{\sigma}_x^{(0)}\hat{\sigma}_x^{(1)}\hat{\sigma}_y^{(2)}\hat{\sigma}_x^{(3)}}$
H
H
$Rx(\frac{\pi}{2})$
H
$Rz(\theta)$
H
H
$Rx(\frac{-\pi}{2})$
H

Figure 1: Quantum circuits corresponding to the exponential maps of different Pauli strings [4].

Figure 1 shows the quantum circuit corresponding to some exponential map $e^{-i\frac{\theta}{2}\hat{P}}$, where $\hat{P}$ is a product of Pauli matrices. This is practically implemented by creating an entangling CNOT ladder between all the Pauli operators $\hat{\sigma}_i$; in the middle of this ladder a $R_z(\theta)$ gate is added followed by another disentangling CNOT ladder, in reverse order as before. The first circuit shows how to implement strings in the form $\hat{\sigma}_z$; if the Pauli string contains also $\hat{\sigma}_x$ and $\hat{\sigma}_y$ it is necessary to perform a basis transformation before the CNOT ladder and the inverse after the circuit. This is concretely done by adding an Hadamard gate and a $R_x(\pi/2)$ respectively for $\hat{\sigma}_x$ and $\hat{\sigma}_y$.

2.3 Adapt-VQE

Even with multiple improvements, UCCS anstaz remains an approximate approach that works best for systems that are not strongly correlated and it does not allow to obtain an arbitrary accuracy. In order to address these problems, the **Adaptive Derivative-Assemble Pseudo-Trotter ansatz Variational Quantum eigensolver** (Adapt-VQE) [8] was proposed. The key idea of this method is to construct an ansatz by adding one fermionic operator at-a-time; this would recover the ground state with an arbitrary accuracy using the least number of fermionic operators and variational parameters. The following steps are taken into account:

1. Construction of the fermionic Hamiltonian through a classical hardware and its transformation into a qubit representation (Jordan-Wigner [16] or Bravyi-Kitaev [6] transformation).

2. Creation of the Operator Pool: collection of parametrized operators that will be used to construct the ansatz.

3. Initialization to an appropriate reference state, that for molecular problems it is usually represented by the Hartree Fock state.

4. Preparing the trial state with the current ansatz.

5. Evaluation of the gradient with respect to each operator in the pool. This can be done in different ways: in the original paper [8] they propose to use the commutation relation between the operator itself and the Hamiltonian of the system.

6. If the norm of the gradient vector is smaller than some threshold ϵ exit.

7. Identification of the operator with the largest gradient, to be added at the left end of the ansatz (in the operator representation), with a new variational parameter. The operator pool is not drained, so the same operator may be chosen different times.

8. Perform the VQE algorithm to find the best set of parameters for this ansatz; all the parameters are updated at each step.

9. Go to step 4 and repeat.

This method has some advantages compared to the UCCSD for performing simulations on NISQ hardwares as on average it uses a smaller number of operators that would allow to perform simulations for larger molecules. Moreover, the Adapt-VQE adds operators that give the highest contribution to the energy descent and avoids the problem of having to optimize parameters with zero-contribution. An important thing to mention, which will be relevant later on in the discussion, is that the algorithm requires two cycles: we will called the outer one the Adapt-VQE cycle and the inner one the VQE cycle. In the first one the algorithm chooses the operator with the maximum gradient and creates a trial state; inside this cycle, the VQE algorithm is performed to update the parameters and to find the ground state of this ansatz wavefunction, updating the energy.

The **operator pool** is the set of operators available to create the ansatz state; in principle this pool can be constructed from any set of operators, but the most starightforward way is to use the ones defined by the UCCT, mapped to spin operators as shown in Figure 1. Potentially higher order operators than single and double excitations could be added. Other approaches may be taken into account: in our case the operators were defined starting from the Hamiltonian of the system and we will discuss this in more detail in the following section. Another approach has been suggested in [14] where, instead of fermionic operators, individual Pauli strings form the operator pool. A minimal complete pool for three qubits and a recursive formula to create the pool for n-qubits is defined; this would generally allow to obtain configurations with a smaller number of CNOT gates but this ansatz may require more gradient calculations on a quantum computer as it is built with a larger number of parameters.

Different procedures may be considered to compute the **gradient of a quantum circuit** [12]. The expectation value of a quantum observable is a deterministic quantity that varies smoothly with the gate parameters: it is possible to define the gradient of a quantum circuit via the derivatives of expectations. Through the **Parameter Shift Rule**, only two more evaluations of the expectation value are necessary. The overall unitary $U(\theta)$ that is applied to the variational circuit can be decomposed into a sequence of single parameter gates, which can be differentiated using the product rule; we can assume that this single operator is in the form $e^{-i\theta G}$. The derivative of a function f, that considers the evaluation of an expectation value on this parametrized circuit, can be defined as:

$$\partial_\theta f = r[f(\theta + s) - f(\theta - s)] \tag{4}$$

where r is the eigenvalue of the generator of the gate G and $s = \pi/(4r)$. If the parameter appears in more than a single gate in the circuit, the derivative is obtained using the product rule by shifting the parameter in each gate separately and summing the results. Taking into account operators in the form R_x, R_y, R_z, we have that the correspondent eigenvalue is 1/2, leading to the following gradient formula:

$$\partial_\theta f = \frac{1}{2}[f(\theta + \frac{\pi}{2}) - f(\theta - \frac{\pi}{2})]. \tag{5}$$

An evaluation of the function is made only two more times, just shifting the parameter in the corresponding gate.

In the original Adapt-VQE paper [8] they suggest to use the **commutator** of the Hamiltonian with each operator in the pool; indeed, imposing to evaluate the gradient at $\theta = 0$ and using a shift equal to $\pi/2$, it is possible to reconstruct the same result, as proven in the appendix.

Different **classical optimizers** may be taken into account in order to update the parameters in the VQE algorithm. In this project two optimizer were used and their effect compared: the **Gradient Descent** and the **Quantum Natural Gradient**. In the gradient descent approach, given the hyperparameter η known as the learning rate, the parameters (x) at step t are updated through the gradient of the loss function $\nabla f(x)$:

$$x^{(t+1)} = x^{(t)} - \eta \nabla f(x(t)) \tag{6}$$

The Quantum Natural Gradient [13] is a generalization of Natural Gradient for variational quantum circuits. The problem with the previous approach is that we are updating the parameters with respect to the Euclidean Geometry but the parametrization can distort distances within the optimization landscape and the cost function might vary at a different rate with respect to each parameter. By changing the coordinate-system, performing a re-parametrization of the cost function, we might find a parameter space where variations in the loss are similar accross different parameters and so we have a more informative stepsize, leading to faster convergence. A different approach is to consider the optimization process as a probability distribution of possible output values given an input; in this context we can perform the gradient descent in the distribution space, dimensionless and invariant with respect to the parametrization.

Each optimization step will always choose the optimum step size for each parameter, regardless the parametrization [3]. The classical gradient descent is modified as follow:

$$x^{(t+1)} = x^{(t)} - \eta F^{-1} \nabla f(x(t)) \tag{7}$$

where F is the Fisher Information matrix, that acts as a metric tensor, transforming the steepest descent in the euclidean parameter space to the steepest descent in the distribution space. In a similar case, it has been shown that the standard Euclidean geometry is sub-optimal for optimization of quantum variational algorithms [9]; it is possible to define the Fubini-Study metric tensor g_{ij} which can be used to construct a quantum analog to natural gradient descent. In the end, what is obtained is an optimizer with adaptive learning rate and the following equation is applied to update the parameters.

$$x^{(t+1)} = x^{(t)} - \eta g^{\dagger}(x_t)\nabla f(x(t)). \tag{8}$$

3 Set up of the simulations

3.1 Definition of the problem: the TFIM

In this project, we focused on the **Transverse Field Ising Model** (TFIM) as this can be studied for an arbitrary number of qubits, even a small one. This is the quantum analog of the classical Ising model; it features a lattice with nearest neighbour interactions determined by the allignement or antiallignement of spin projections along the z axis plus an interaction with an external field:

$$H = J\sum_{ij} Z_i Z_j + g\sum_{j} X_j. \tag{9}$$

J represents the spin coupling and g the strengthen of the transverse field (in our case both were fixed to 1). We considered 4 qubits (or 4 spins) and the ground state of this specific Hamiltonian resulted in an energy of -5.226 (computed as the minimum eigenvalue of the corresponding matrix).

3.2 The initial state

The initial state, where to apply the operators to create the ansatz, was at first defined as an equal superposition of all the possible states, so applying an Hadamard gate at each of the qubits. It resulted that this configuration had some limitations; indeed the ground state of the system was correctly found but the first step of the Adapt-VQE gave zero value for many of the gradients; only some operators where chosen. After having done different trials regarding the operator pool and the defintion of the gradient to check where the problem could have originated, we changed the intial state considering a random U rotation applied to each qubit $|0\rangle$. Besides, this allowed to check that the optimization obtained was not dependent on the initial state chosen, giving a more general approach.

3.3 The definition of the operator pool

The operator pool was defined taking into account operators that would involve one, two or three qubits at time; they are reported in table 1. We started from operators that were present in the Hamiltonian, taking the exponent of them $e^{-i\frac{\theta}{2}H_i}$; the same procedure as shown in figure 1 was adopted to practically define these operators, considering CNOT ladders with a parametrized R_z rotation in the middle and performing a change of basis when required. Then, operators from powers of the Hamiltonian (H^n) were taken into account, all at second order and one of them at third order. Using these operators, it was not always possible to reach the desidered state; the convergence was quite difficult and the process stucked at some value of energy that was not exactly the ground state of the system. Moreover, for the first initial state that we had defined, the equal superposition of all the qubits, the gradient obtained was zero for many operators at the first step, not allowing a proper operator choice. For these reasons, more operators (even the ones that were not directly present in the powers of the Hamiltonian), were added, applied to all the possible combinations of qubits. An important remark is that, as the Pauli matrices in a single operator apply to different qubits, they commute and so the order in which we place the Pauli matrices does not influence the result we obtain (for example applying the exponentialization of Z_iY_j is the same as applying Y_jZ_i).

	Operators in the complete operator pool
First power H	X_i, Z_iZ_j
Second power H	X_iX_j, Z_iY_j, $Z_iZ_jX_m$, Z_mZ_n, $ZZZZ$
Third power H	$Z_iX_jY_m$
Not present in H	XY, ZX, YY

Table 1: Operators present in the operator pool: $i-j$ represents nearest neighbour qubits, $m-n$ represents not-nearest neighbours qubits. In the case $i-j-m$, $i-j$ are nearest neighbours and m is any other wire different from $i-j$. Where there's no index, it means that any kind of combination among different qubits is taken into account.

In this way the complete operator pool was quite big, taking into account 65 possible operators, but this allowed to reach efficiently the ground state. It was possible to sample from this complete pool a set of operators, defining a smaller-dimension operator pool (in this case this passage was done randomly). A comparison between the results that were obtained by choosing different dimensional pools was performed and it is shown in figure 3.

3.4 The gradient of the circuit

As defined previously, the gradient of a quantum circuit can be evaluated in different ways. In this case, both the approaches of the parameter shift rule and the commutator relation were used and it was checked that the same results were obtained. Later, the gradient of the quantum circuit was defined starting from the built-in function in Pennylane, qml.grad(), which allows to compute gradients of hybrid cost functions.

4 Simulations

Once prepared the system, different simulations in different contexts were performed. We considered the case where only one operator at a time is chosen to perform the Adapt-VQE algorithm with different operator pool dimensions; then, we took combinations of two operators at each step. Finally, we compared the results obtaned using two different optimization algorithms, the Gradient Descent and the Quantum Natural Gradient. For each of these simulations, we performed an average over different trials, 13 or 14, changing the random initialization of the state.
In the case where the dimension of the operator pool is reduced from the intial dimension of 65, the operators were chosen randomly but in such a way to have always the same for the different trials; for all the different initializations the available operators are the identical. Furthermore, the reduced operator pools are always sampled from the bigger ones; for example, the 40 operators pool is sampled from the complete 65 operators and the 30 operators is sampled from the 40 operators. Therefore, the 30 operators one will not contain any operator that is not present in the 40 operator pool.
In all the trials we were able to approach the ground state of the system up to a certain accuracy but some trials allowed to reach it with less iterations (i.e. cycles in the algorithm) or with a steepest descent. One of the most critical metrics when considering the cost of simulations on NISQ devices is the circuit depth, because of short coherence times and large error rates typical of these devices; we took into account this parameter to compare the different trials.

4.1 Changing the dimension of the operator pool

Firstly, we considered only one operator at a time in the choice of the operator that maximizes the gradient and we compared the cases where we took different dimensions of the operator pool. Figure 2 shows the energy loss on average with the corresponding variance band comparing the following cases: dimension of the operator pool 65 vs 15, 20 and 30. On average bigger dimension of the operator pool allows faster convergence: more operators are available to reach the ground state in an efficient way.

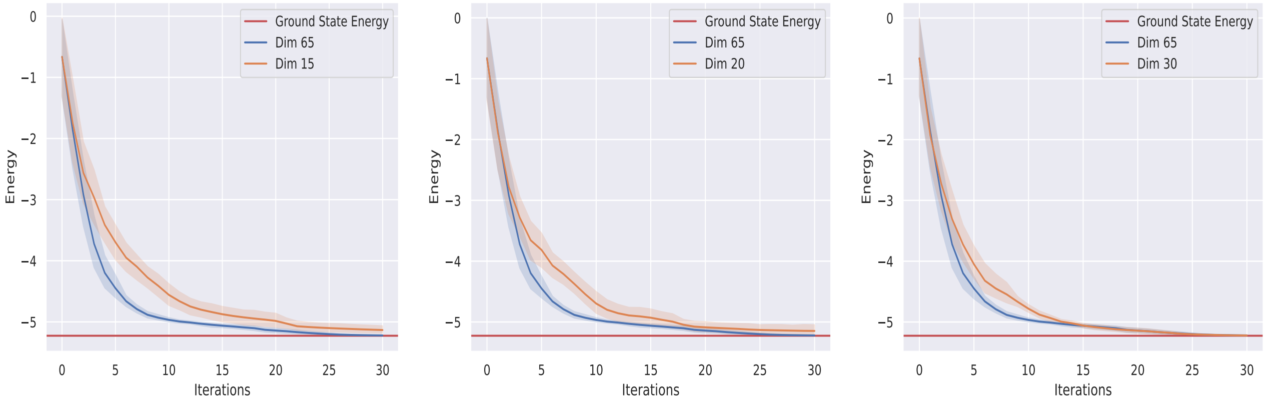

Figure 2: Energy decrease on average and corresponding variance band, comparing the case with 65 operators in the operator pool and 15, 20 and 30 respectively.

In Figure 3 the average trend of the loss function is shown, without the variance band, for the cases 65, 40, 30, 20 and 15 operators (left side). Starting from the same value of energy, the complete operator pool performs a faster optimization in the central part and then it levels off approaching the minimum of the loss function and it performs the same as in the intermediate cases (40-30). In the reduced dimension cases (20-15), the decrease is definitely less steep and the final energy found is a bit higher that in the big dimension case, as reported in table 2. These differences may not be relevant for NISQ devices, as they would be covered by errors, but they can be important for future quantum computation applications in chemestry.

Dimension	**65**	**40**	**30**	**20**	**15**
$\langle E_0 \rangle$	-5.221	-5.221	-5.222	-5.15	-5.13
σ	0.004	0.003	0.003	0.11	0.10

Table 2: Exact E_0= -5.226

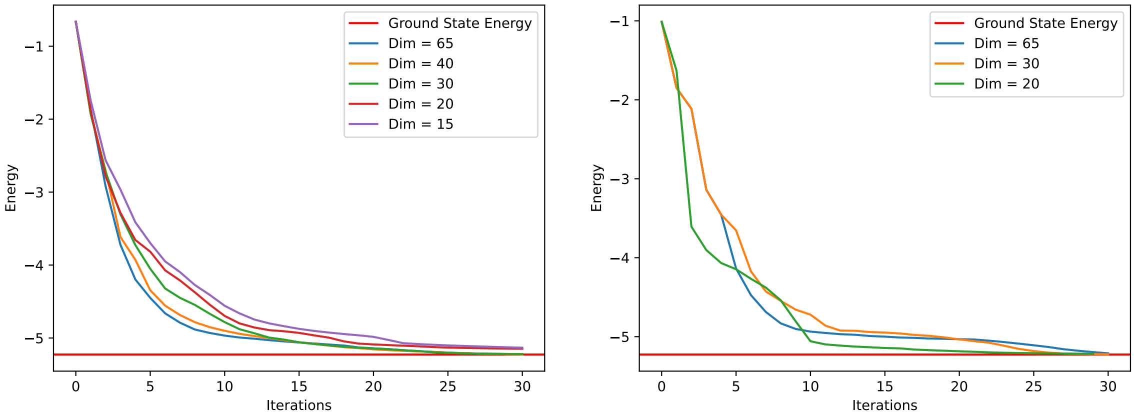

Figure 3: Right side: mean values of the energy decrease for different dimension of the operator pool. Left side: specific trial where it is possible to see an unexpected behaviour for the trend.

In the right side of Figure 3 is shown an interesting trial, with a specific initialization. The trend is not what expected and obtained on average: the energy decreases faster for the reduced-dimension operator pool and it allows a faster convergence and a more accurate final energy. As we stated before, all the operators that are present in the 20 operator pool are also present in the higher-dimension ones and so the operator chosen as the best one in the 20-case should lead to a decrease in the energy equal (in the best case) or lower to the one in the higher dimension. This example uncovers one of the limits of this method: the gradient of the operator allows to obtain a description of the landscape that is only local. It may be that, updating the parameters according to the learning rate, we end up in a situation where the choice of a different operator would have been more efficient as we are no more in the local space we were considering before. The operator with the largest gradient is not guaranteed to have the greatest impact on the energy, so using gradients is not the optimal approach. Different methods may be used, as proposed in [5] where it is identified the operator that reduces the energy by the most significant margin.

In table 3 we report different values of the circuit depth for the different trials. The circuit depth is computed up to a convergence to the ground state energy equal to 0.006 (for 65, 40 and 30 operators) or up to 30 steps in the Adapt-VQE cycles in the lower dimensional case, where the ground state was not found efficiently. We observe that for the three higher dimensional cases the circuit depth is comparable and it is smaller than the one in the lower dimensional cases; this was expected since in these cases more steps are necessary to approach the ground state on average, leading to a bigger number of operators used.

Dimension	**65**	**40**	**30**	**20**	**15**
Circuit depth, mean value	121 ± 22	122 ± 15	119 ± 22	132 ± 16	149 ± 18

Table 3

In figure 4 we report the error of the energy, $E - E_0$ where E_0 is the exact ground state, versus the depth of the circuit for the complete-dimension pool as the energy was optimized throughout the algorithm. We notice how the circuit depth quickly grows in the final steps of the optimization, when approaching the minimum of the energy. A different value of the learning rate could be used or a more deeep analysis could be performed in these last steps to allow a faster convergence, resulting in less operators to be added at the trial wavefunction.

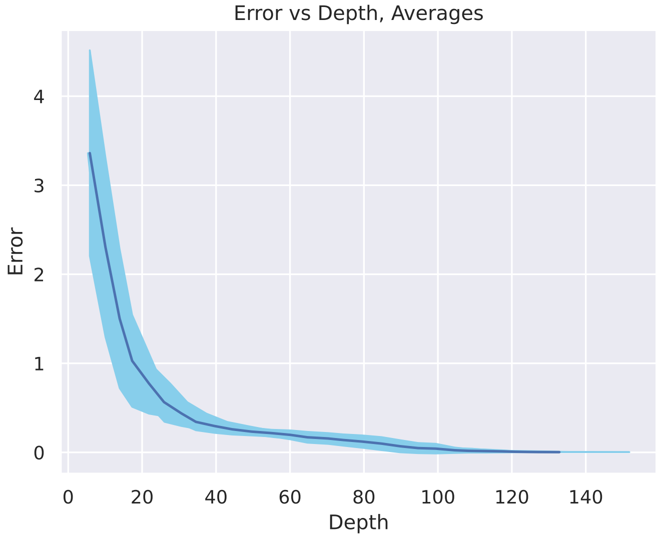

Figure 4: Average errors as a function of the circuit depth

4.2 Combination of operators

Aftewards, we allowed the algorithm to take two operators at a time of the ones present in the operator pool, combining them. To perform a correct comparison with the case where only one operator at a time was chosen, we also allowed to take the single operators. In the end, the algorithm can choose one single operator or combinations of two operators among these. The dimension of the new operator pool would have been quite big (as all the possible combinations are given by the binomial factor of the number of single operators over 2), requiring an higher computational cost in the evaluation of the gradient; for this reason we decided to sample from the complete operator pool, taking only 20 operators from the original one. In this way, the algorithm chooses among 20 single operators and 190 possible combinations.

The choice of the operator or of the combination of them is done always considering the maximum gradient or the norm of the gradient vector for the combinations, where the parameter space has dimension 2. In this way, the algorithm always chose combinations and never single operators; the optimization was quite slow and sometimes got stuck to a specific point. In order to avoid this behaviour, half of the norm of the gradient for combinations was taken into account, considering the contribution given by a single parameter. This actually allowed to obtain a more equal comparison and a more balanced choice among the operators; the ground state of the system was efficiently found.

In figure 5, we show the difference in the energy optimization when only single operators were available versus when combinations were also considered. We observe that the combination of operators allowed to reach the ground state in a most efficient way and that the convergence is reached with a smaller number of steps. In this case however we should consider that at each step we add two operators and this leads to an higher circuit depth, as reported in table 4.

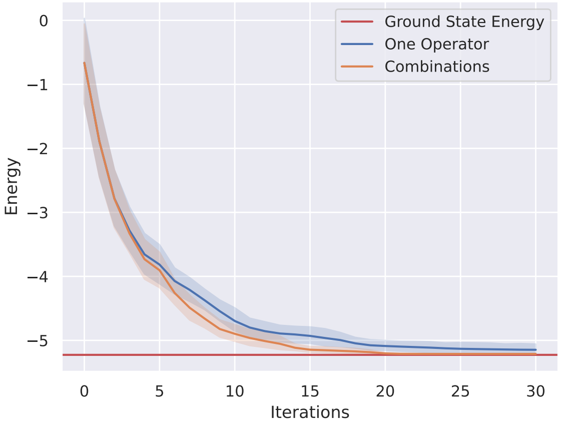

Figure 5: Mean values for the energy convergence taking one operator at a time vs double-combinations of them

	Single operator	Combinations
E_0	-5.15 ± 0.11	-5.212 ± 0.014
Circuit depth	132 ± 16	162 ± 35

Table 4: Exact E_0= -5.226

4.3 Gradient Descent vs Quantum Natural Gradient

Then, we repeated some of the previous trials considering the Quantum Natural Gradient in order to update the parameters; we compare these results with the previous ones, where the gradient descent was used.

The optimization of the parameters enters in the inner VQE-cycle. In the case of the previous trials, the VQE algorithm was left free to update the parameters until a certain convergence for the value of the ground state. We can expect that in these cases the results obtained using the two different optimizations should be approximately the same. Indeed, we are not monitoring how the parameters (and the energy) are updated in a single VQE cycle but how the optimized energy changes adding operators to the ansatz state; we are monitoring the outer Adapt-VQE cycle.

This is exactly what is obtained in figure 6, where we show the optimization of the energy for both the cases where one single operator (for the complete operator pool of 65 operators) and combinations of them were chosen. We can make a remark about the choice of the learning rate: indeed in the case of the QNG it was necessary to use a smaller learning rate than in the case of the GD (0.2 vs 0.3) and it was also sometimes necessary to update this value in order to have a correct convergence in the Adapt-VQE cycle (after 10 iterations it was lowered to 0.1).

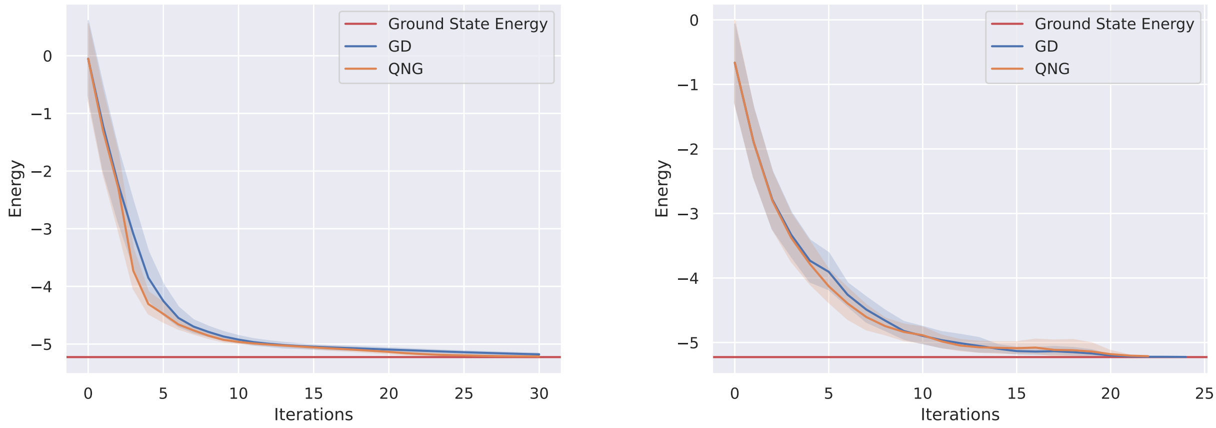

Figure 6: Optimization process comparing the quantum natural gradient and the gradient descent. Right: one operator at a time. Left: combinations of operators. We can see that the results obtained are similar as we are allowing the VQE cycle to update the parameters up to a certain threshold.

In the case of combinations of operators, the QNG performed on average a smaller number of iterations for the outer Adapt-VQE cycle: when approaching the ground state the optimization was sometimes more efficient. This led to a configuration where the total number of operators that were used was smaller and we obtained shallower circuits with a corresponding smaller depth, as reported in table 5. Moreover, in the case where only one operator at a time was taken, the QNG allowed to obtain a smaller value of energy than in the case of the GD(-5.178 vs -5.215).

	GD 1 op	QNG 1 op	GD combinations	GD combinations
Circuit depth, mean value	139	129	162	140
Circuit depth, std dev	19	18	35	21

Table 5

In order to compare the optimization performed by Quantum Natural Gradient and the Gradient Descent, we repeat the experiment shown in left side of Figure 6 but limiting the number of steps in the VQE optimization routine, 10 in our case. The energy update in the case of the GD is on average less efficient: at any iteration the update of the parameters is less important, leading to a final optimized energy higher than in the case of the Quantum Natural Gradient, as reported in table 6. This difference is found from the very beginning where the decrease in energy is definitely steeper for the QNG, as shown if figure 7.

	GD	QNG
E_0	-5.12 ± 0.03	-5.20 ± 0.04

Table 6: Exact E_0= -5.226

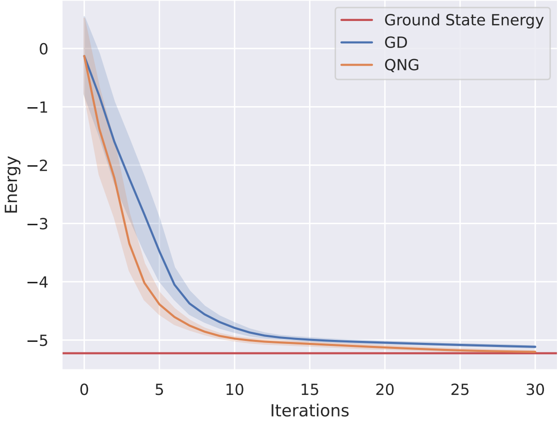

Figure 7: Comparison between the GD and QNG when taking only one operator at a time and fixing the number of steps allowed in the single VQE cycle.

We can conclude that the Quantum Natural Gradient is on average a better choice in the single VQE cycle and that can sometimes lead to a final configuration with smaller circuit depth.

4.4 Hessian of the loss function

We also tried to apply the Hessian of the loss function in order to characterize and to better understand the loss landscape [10]. The Hessian of a function is a methematical object, defined through a square matrix, whose spectrum gives information regarding the local curvature of the function itself, allowing to study the loss landscape and the convergence in the case of Neural Networks. In the particular case of variational quanutm circuits, it can be useful when approaching a flat region, in order to escape from it.

In our case, we considered the spectrum of the Hessian of the loss function when picking combinations of operators at every step; this was done to find the steepest direction and to update the parameters in the most efficient way. Indeed, the signs of the eigenvalues of the Hessian are related to the increase (positive sign) or to the decrease (negative sign) of the loss function and the direction of the increase/decrease is given by the corresponding eigenvector. The maximum increase or decrease direction is given by the eigenvetor corresponding to the maximum eigenvalue; this should lead to a faster convergence as this brings the algorithm to move exactly in the optimal direction. Our original idea was then to choose the combination of operators with the biggest maximum eigenvalue to obtain the best optimization of the parameters.

This method in our case could not apply from the very beginning as we were dealing with points in the loss function that were not close to its minimum and so the information obtained from the Hessian was related to the local change of the gradient but not to the local change of the function itself. Using this approach, the combination with the highest eigenvalue didn't correspond necessarly to the one with the biggest gradient; as the optimization of the parameters was then performed using the gradient of the function, the choice of the operator was not optimal and led to a wrong convergence. Choosing the combination without the biggest gradient, the optimization of the parameters would remain always close to the local region and in the following steps the same combination was chosen. The optimization led to a point in the parameter space close to the one we were before, leading the loss function to become stuck somewhere.

5 Conclusions

We addressed a specific problem in the VQE algortihm, the definition of the initial ansatz, through the Adapt-VQE algorithm to find the ground state of a physical Hamiltonian and we performed different trials to study the optimization process changing some features in our system. In all the cases, the ground state of the system was found, but some trials performed computationally better than others.

- In the case where only one operator at a time is taken into account, we showed that the dimension of the operator pool can be relevant in the optimization process, giving more freedom to choose the operators. A smaller dimension of the operator pool would require more steps on average, while for intermediate and bigger dimension of the operator pool the situation is quite similar.
- Combinations of different operators, when the dimension of the operator pool is reduced, may lead to a better optimized energy, finding the ground state in a more efficient way. The drawback is that this would lead to a bigger circuit depth on average. For NISQ devices, this would be quite important and the differences obtained for the energy of the system may be covered by noise; for future developments this difference may be relevant for applications in quantum chemestry.
- Using the Quantum Natural Gradient is more efficient when using a limited number of steps in the single energy optimization, so in the single VQE cycle. Moreover, using this optimizer can lead to a faster convergence and to configurations with smaller circuit depth on average.
- The use of the Hessian of the loss function may not be effective from the very beginning in this particular case as we are not close to the minimum.

5.1 Further developing

In order to test the generality of the algorithm, we checked that this procedure worked also for a different number of spins (or qubits) increasing the dimension of the system. In order to test the generality of this approach, a problem different from the TFIM model may be used.

The problem of the definition of the operator pool may be addressed to understand what is the minimal number of operators that should be defined to include all the possibilities to solve the problem.

The case where more than two operators are chosen for any combination may be studied, to check if this leads to a condition where smaller circuits and less iterations can be used.

As we briefly discussed before, the maximum gradient of the circuit may not be an optimal metric to choose which operator to add at the trial state; different approaches may be considered.

The Hessian of the function, even if it didn't work from the very beginning, may be used to study the optimization at its end, when close to the ground state of the system. This may lead to a faster convergence in this region a to a smaller final circuit depth.

6 Appendix

1. Commutator relation to compute the gradient of a quantum circuit. We suppose to evaluate the gradient for $\theta = 0$ and shifting the parameter by $\pi/2$, as required for the parameter shift rule in the case where the parametrized gate is a R_z rotation.

$$E^n = \langle \psi^n | \, e^{iA_n\theta/2} H e^{-iA_n\theta/2} \, |\psi^n\rangle \tag{10}$$

Applying the parameter shift rule:

$$\nabla E^n = \langle \psi^n | \, e^{iA_n\pi/4} H e^{-iA_n\pi/4} \, |\psi^n\rangle - \langle \psi^n | \, e^{-iA_n\pi/4} H e^{iA_n\pi/4} \, |\psi^n\rangle \tag{11}$$

Expanding the exponential:

$$e^{\pm iA_n\pi/4} = cos(\pi/4)\mathbb{1} \pm isin(\pi/4)A_n = \frac{1}{\sqrt{2}}(\mathbb{1} + iA_n) \tag{12}$$

Computing $e^{iA_n\pi/4} H e^{-iA_n\pi/4}$ and the difference, we obtain:

$$\begin{aligned} \nabla E^n &= \langle \psi^n | \, i(A_n H - H A_n) \, |\psi^n\rangle \\ &= i \, \langle \psi^n | \, [A_n, H] \, |\psi^n\rangle \end{aligned} \tag{13}$$

References

[1] Github: Adapt-vqe. `https://github.com/GaiaBolognini/AdaptVQE_TPIV`.

[2] Pennylane. `https://pennylane.ai/`.

[3] Pennylane tutorial on quantum natural gradient, 2021.

[4] Abhinav Anand, Philipp Schleich, Sumner Alperin-Lea, Phillip W. K. Jensen, Sukin Sim, Manuel Díaz-Tinoco, Jakob S. Kottmann, Matthias Degroote, Artur F. Izmaylov, and Alán Aspuru-Guzik. A quantum computing view on unitary coupled cluster theory, 2021.

[5] V. Armaos, A. Dimitrios, Deligiannis Paraskevas, Lianos Konstantinos, and S. Yordan. Efficient parabolic optimisation algorithm for adaptive vqe implementations, 2021.

[6] Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, May 2002.

[7] Dmitry A. Fedorov, Bo Peng, Niranjan Govind, and Yuri Alexeev. Vqe method: A short survey and recent developments, 2021.

[8] H.R. Grimsley, S.E. Economou, E. Barnes, et al. An adaptive viariational algorithm for exact molecular simulation on quantum computer. *Nat Commun*, 10, 2019.

[9] Aram W. Harrow and John C. Napp. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Physical Review Letters*, 126(14), Apr 2021.

[10] Patrick Huembeli and Alexandre Dauphin. Characterizing the loss landscape of variational quantum circuits. *Quantum Science and Technology*, 6(2):025011, Feb 2021.

[11] A. Peruzzo, J. McClean, P. Shadbolt, et al. A variational eigenvalue solver on a photonic quantum processor. *Nat Commun*, 5, 2014.

[12] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99:032331, Mar 2019.

[13] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, May 2020.

[14] Ho Lun Tang, V.O. Shkolnikov, George S. Barron, Harper R. Grimsley, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. Qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2), Apr 2021.

[15] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Phys. Rev. A*, 92:042303, Oct 2015.

[16] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik. Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, Mar 2011.