

PHYSICS FINAL DEGREE PROJECT

Adiabatic preparation of molecular ground states

Raquel García Bellés

Supervised by Dr. Artur García Saez and Dr. Pol Forn Díaz Academic tutor: Dr. John Calsamiglia Costa





- July 2021 -

Abstract

Quantum chemistry simulation is just one of the many areas in which quantum computers could outperform classical ones. Aspuru-Guzik et. al. proposed in [4] a quantum algorithm to efficiently calculate the ground state energy of molecules. This algorithm required the ability to prepare the molecular ground state in a quantum register, and for this end they also proposed the adiabatic state preparation (ASP) algorithm. This algorithm is based on the adiabatic theorem, which guarantees that a system prepared in the ground state will remain in the instantaneous ground state when let evolve in time under a varying Hamiltonian, as long as this variation is sufficiently slow. In this project, we use existing Python packages to perform classical simulations of ASP for the ground states of dihydrogen (H_2) and lithium hydride (LiH). We also pay attention to the role played by entanglement in this process.

Acknowledgements

I want to express my gratitude towards my supervisors Artur García and Pol Forn for their advice and their prompt responses. I also want to thank Miquel Saucedo for his support through all these years. And last but not least, I want to thank my family for always letting me follow my dreams, even if that means moving to the other side of the world. In particular, I want to dedicate this work to my father, who recently passed away.

Contents

1	Intr	oduction	3		
2	Theoretical Background				
	2.1	Basics of quantum chemistry	4		
		2.1.1 Mapping fermions to qubits	6		
	2.2	Adiabatic Quantum Computing	8		
	2.3	Entanglement entropy	9		
3	Met	hodology	11		
4	Results				
	4.1	Preliminary study	13		
	4.2	Adiabatic state preparation	14		
	4.3	Improving the evolution time	16		
	4.4	The role of entanglement	19		
5	Conclusion				
A	ppen	dices	25		
A	Complementary results				
	A.1	Occupation number representation	26		
	A.2	Adiabatic evolutions	27		
	A.3	Entanglement results of the lithium hydride	30		
В	Cod	le	31		

Chapter 1

Introduction

In 1982 R. Feynman published the groundbreaking paper Simulating Physics with Computers [11], where he proposed using quantum computers to simulate quantum systems. Since then, research in this area has been very fruitful, and the applications in which quantum computers could outperform classical ones range, among others, from integer factorization [22] to quantum chemistry. The latter is the one that concerns us in this work.

In [4], Aspuru-Guzik et. al. presented a quantum algorithm to efficiently calculate the ground state energy of molecules. In order to obtain a high probability of success, this algorithm requires the preparation of a state in a quantum register with a large square overlap with the ground state of the molecule. The authors proposed that this could be achieved by an adiabatic state preparation (ASP) algorithm relying on the adiabatic theorem [7, 9]. This theorem states that a system will remain in its instantaneous ground state if the Hamiltonian that governs its evolution is changed slowly enough over time. Another requirement is the existence of a positive energy gap between the instantaneous ground state and the rest of the eigenspectrum throughout the evolution. Thus, ASP involves starting with a Hamiltonian whose ground state is easy to prepare, and then time evolving the system while slowly varying the Hamiltonian into the one describing the desired molecule.

In this project we perform a qualitative and numerical study of the adiabatic state preparation of the ground states of dihydrogen (H_2) and lithium hydride (LiH). We will discuss how the evolution time is affected by the minimum energy gap, and different ways in which this time can be reduced. We will also look at the entanglement of the system along the evolution, and we will establish relations between the system entanglement and other magnitudes, such as the probability of success and the position of the minimum energy gap. Previous works [25, 15] studying ASP relied on advanced computational chemistry techniques and programming. In our code we bridge two existing Python packages for the classical simulation of quantum systems, namely OpenFermion [18] and QuTiP [14], which allows us to reproduce and extend the ASP study also for other molecules in a simple way.

Chapter 2

Theoretical Background

2.1 Basics of quantum chemistry

In quantum chemistry, the electronic problem consists in finding approximate solutions to the non-relativistic Schrödinger equation $H|\phi\rangle = E|\phi\rangle$ [24]. Here H is a Hamiltonian that describes a system of nuclei and electrons, usually a molecule. If we denote the position of the Ath nucleus and the *i*th electron by $\mathbf{R}_{\mathbf{A}}$ and $\mathbf{r}_{\mathbf{i}}$ respectively, and we let $r_{iA} = |\mathbf{r}_{\mathbf{i}} - \mathbf{R}_{\mathbf{A}}|$, $r_{ij} = |\mathbf{r}_{\mathbf{i}} - \mathbf{r}_{\mathbf{j}}|$ and $R_{AB} = |\mathbf{R}_{\mathbf{A}} - \mathbf{R}_{\mathbf{B}}|$ be the distances between any pair electron-nucleus, electron-electron and nucleus-nucleus, respectively, then, in atomic units the Hamiltonian for N electrons and M nuclei is

$$H = -\sum_{i=1}^{N} \frac{1}{2} \nabla_i^2 - \sum_{A=1}^{M} \frac{1}{2M_A} \nabla_A^2 - \sum_{i=1}^{N} \sum_{A=1}^{M} \frac{Z_A}{r_{iA}} + \sum_{i=1}^{N} \sum_{j>i}^{N} \frac{1}{r_{ij}} + \sum_{A=1}^{M} \sum_{B>A}^{M} \frac{Z_A Z_B}{R_{AB}}.$$
 (2.1)

 M_A is the ratio of the mass of the Ath nucleus to the mass of an electron, Z_A is the atomic number of nucleus A, and ∇_i^2 and ∇_A^2 are Laplacian operators that involve derivation with respect to the coordinates of the *i*th electron and the Ath nucleus. The first and the second term in (2.1) represent the kinetic energy of the electrons and the nuclei, respectively. The third, fourth and fifth terms are, respectively, the Coulomb interactions between pairs of electron-nucleus, electron-electron and nucleus-nucleus.

In general, this is a complicated problem to solve. We know how to find an exact solution to the Schrödinger equation for the Hydrogen atom Hamiltonian, where N = M = 1. However, even this simple case requires sophisticated mathematical tricks. In a more general situation, for an arbitrary number of nuclei and electrons, we can only hope to find an approximate solution after making some simplifications. One of this simplifications that is key to quantum chemistry is the Born-Oppenheimer Approximation [24]. Since nuclei are much heavier than electrons, its kinetic energy will be much smaller. Thus, we can also think that nuclei are fixed as compared to the electrons. Hence in the Born-Oppenheimer approximation we eliminate the second kinetic term in equation (2.1) and we treat the $\mathbf{R}_{\mathbf{A}}$'s as constant parameters. The last term in equation (2.1) is also treated as a constant and only needs to be taken into account when calculating the total energy. After these simplifications, what remains is called the electronic Hamiltonian and it writes

$$H_{elec} = -\sum_{i=1}^{N} \frac{1}{2} \nabla_i^2 - \sum_{i=1}^{N} \sum_{A=1}^{M} \frac{Z_A}{r_{iA}} + \sum_{i=1}^{N} \sum_{j>i}^{N} \frac{1}{r_{ij}}.$$
(2.2)

This Hamiltonian describes the motion of N electrons in the field of M point charges. Notice that the solutions to the Schrödinger equation $H_{elec}|\phi_{elec}\rangle = E_{elec}|\phi_{elec}\rangle$ will depend on the set of nuclei coordinates used, i.e., $|\phi_{elec}\rangle = |\phi_{elec}(\{\mathbf{r_i}\}; \{\mathbf{R_A}\})\rangle$, and also the energy will be $E_{elec} = E_{elec}(\{\mathbf{R_A}\})$. The total energy is calculated adding the constant term we dropped before. Once the electronic problem is solved, a new Hamiltonian for the nuclei can be defined taking into account the mean field of the electrons. However, throughout this work we will concentrate only on the electronic problem with the Hamiltonian (2.2).

The electronic Hamiltonian (2.2) only involves the spatial coordinates of the electrons, but to fully describe an electron it is also necessary to specify its spin. Therefore, we will introduce two orthonormal spin functions, $\alpha(\omega)$ and $\beta(\omega)$, corresponding to spin up and down, respectively. ω is an unspecified spin variable. Thus in this formalism each electron is described using four coordinates, namely $\mathbf{x}_i = {\mathbf{r}_i, \omega_i}$. Then, the wave function of an N-electron system will be $\Phi(\mathbf{x}_1,...,\mathbf{x}_N)$. This function must satisfy the Schrödinger equation for the electronic Hamiltonian and, because we consider the electrons as indistinguishable fermions, it must be antisymmetric with respect to the exchange of any two electrons. This means that $\Phi(\mathbf{x_1}, ..., \mathbf{x_i}, ..., \mathbf{x_j}, ..., \mathbf{x_N}) =$ $-\Phi(\mathbf{x_1},...,\mathbf{x_i},...,\mathbf{x_i},...,\mathbf{x_N})$. To achieve this, we shall first consider single electron wave functions, these are called orbitals. A spatial orbital $\psi_i(\mathbf{r})$ is a function that describes the spatial position of an electron, so that $|\psi_i(\mathbf{r})|^2 d\mathbf{r}$ is the probability of finding the electron in a volume $d\mathbf{r}$ surrounding point **r**. The spin orbitals take into account the possible spin of the electron, and they are defined as $\chi_{2i-1}(\mathbf{x}) = \psi_i(\mathbf{r})\alpha(\omega)$ and $\chi_{2i}(\mathbf{x}) = \psi_i(\mathbf{r})\beta(\omega)$. If we dropped the last term in (2.2), then we would be left with a Hamiltonian describing non-interacting electrons. Hence, using a separation of variables we could find a wave function of the form $\Psi^{HP}(\mathbf{x_1},...,\mathbf{x_N}) = \chi_i(\mathbf{x_1})\chi_i(\mathbf{x_2})\cdots\chi_k(\mathbf{x_N})$. Ψ^{HP} is called a Hartree product, the problem is that it distinguishes between electrons and is not yet antisymmetric. However, we can transform it into a valid wave function if we consider the determinant

$$\Psi(\mathbf{x_1},...,\mathbf{x_N}) = \frac{1}{(N!)^{1/2}} \begin{vmatrix} \chi_i(\mathbf{x_1}) & \chi_j(\mathbf{x_1}) & \cdots & \chi_k(\mathbf{x_1}) \\ \chi_i(\mathbf{x_2}) & \chi_j(\mathbf{x_2}) & \cdots & \chi_k(\mathbf{x_2}) \\ \vdots & \vdots & \vdots \\ \chi_i(\mathbf{x_N}) & \chi_j(\mathbf{x_N}) & \cdots & \chi_k(\mathbf{x_N}) \end{vmatrix}.$$
(2.3)

 $\Psi(\mathbf{x_1}, ..., \mathbf{x_N})$ is called a Slater determinant, and it has N electrons occupying N orbitals without specifying which electron is in which orbital. As a short-hand notation we will denote $\Psi(\mathbf{x_1}, ..., \mathbf{x_N}) = |\chi_i \chi_j \cdots \chi_k\rangle$. The antisymmetry $|\chi_i \cdots \chi_p \cdots \chi_q \cdots \chi_k\rangle = -|\chi_i \cdots \chi_q \cdots \chi_p \cdots \chi_k\rangle$ is clear from the properties of the determinant. Then, the full interacting N-electron wave function is expanded as a sum of Slater determinants $\Phi = \sum_k c_k \Psi_k$. The procedure in which an infinite set of N-electron Slater determinants is used to approximate the wave function is called configuration interaction (CI). For actual computations it is not possible to use an infinite set, hence given a set $\{\chi_i\}$ of 2K single electron spin orbitals, the wave function can be expanded in terms of the $\binom{2K}{N}$ possible Slater determinants, this is known as full configuration interaction (FCI). Some other approximations try to use much less Slater determinants, for instance, the Hartree-Fock approximation approximates the N-electron wave function by a single Slater determinant.

The electronic Hamiltonian (2.2) is often written in second quantization form. We can define creation $\{a_i^{\dagger}\}$ and annihilation $\{a_i\}$ fermionic operators satisfying the anticommutation relations

 $\{a_i, a_j\} = 0, \{a_i^{\dagger}, a_j^{\dagger}\} = 0 \text{ and } \{a_i, a_j^{\dagger}\} = \delta_{ij}, \text{ where } \{A, B\} \equiv AB + BA.$ If we define a normalized vacuum state $|vac\rangle$ that represents a system containing no electrons, then any Slater determinant can be defined by applying a succession of creation operators to this state. For instance, $|\chi_i\chi_j\cdots\chi_k\rangle = a_i^{\dagger}a_j^{\dagger}\cdots a_k^{\dagger}|vac\rangle$. All the properties of Slater determinants are immediately derived from the previous anticommutation relations. They also imply that it is not possible to create two electrons in the same orbital, as expected from the Pauli exclusion principle. Then, the electronic Hamiltonian (2.2) is written in terms of the creation and annihilation operators as

$$H = \sum_{ij} h_{ij} a_i^{\dagger} a_j + \frac{1}{2} \sum_{ijkl} h_{ijkl} a_i^{\dagger} a_j^{\dagger} a_k a_l, \qquad (2.4)$$

where

$$h_{ij} = \int d\mathbf{x}_1 \chi_i^*(\mathbf{x}_1) \left(-\frac{1}{2} \nabla_1^2 - \sum_A \frac{Z_A}{r_{1A}} \right) \chi_j(\mathbf{x}_1), \qquad (2.5)$$

and

$$h_{ijkl} = \int d\mathbf{x_1} d\mathbf{x_2} \chi_i^*(\mathbf{x_1}) \chi_j^*(\mathbf{x_2}) r_{12}^{-1} \chi_k(\mathbf{x_1}) \chi_l(\mathbf{x_2}).$$
(2.6)

In (2.4) we are summing over all possible spin orbitals. The one (2.5) and two-body (2.6) integrals depend on the basis set used to represent the spin orbitals.

2.1.1 Mapping fermions to qubits

In order to simulate the electronic Hamiltonian (2.4) with a quantum computer, it must first be mapped to operate on qubits. Qubits are the quantum equivalent of classical bits, and instead of being in either a 0 or a 1 state they can be in any superposition $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Here α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$, and $\{|0\rangle, |1\rangle\}$ are two orthonormal states that form a computational basis. They can be represented mathematically by two orthonormal vectors, namely

$$|0\rangle = \begin{pmatrix} 1\\0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0\\1 \end{pmatrix}. \tag{2.7}$$

The operators that act on qubits must preserve the unit norm, that is why unitary operators U which satisfy $UU^{\dagger} = \mathbb{I}$ are considered. The ones that concern us are the Pauli operators

$$\sigma_0 = \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$
(2.8)

On the other hand, a system of n qubits can exist in any superposition of 2^n basis states, where each of these basis states can be obtained from a tensor product of $|0\rangle$'s and $|1\rangle$'s. We will write them as

$$|z_0, z_1 \cdots, z_{n-1}\rangle = |z_0\rangle \otimes |z_1\rangle \otimes \cdots \otimes |z_{n-1}\rangle, \qquad (2.9)$$

with $z_i \in \{0, 1\}$ for all *i*. The previous Pauli operators can also be generalized to act on the *p*th qubit:

$$\sigma_p^{\{x,y,z\}} = \mathbb{I}^0 \otimes \cdots \otimes \sigma_{\{x,y,z\}}^p \otimes \cdots \otimes \mathbb{I}^{n-1}.$$
(2.10)

To map fermions to qubits, we need to specify how to map the fermionic basis states, in our case the Slater determinants, to the qubit basis states, and how to map the fermionic operators a_i and a_i^{\dagger} to qubit operators. There is not a unique way of doing this, and actually, these mappings are not trivial because whereas electrons are indistinguishable fermions, qubits are distinguishable.

The notation of Slater determinants $|\chi_i\chi_j...\chi_k\rangle$ means that the spin orbitals $\chi_i, \chi_j, ..., \chi_k$ are occupied by an electron, and that the $\{\chi_l\}$ that do not appear are not occupied. We can change to an equivalent notation that specifies all possible 2K spin orbitals, namely $|n_0, n_1, ..., n_{2K-1}\rangle$. The $\{n_i\}$ are the occupation numbers of the orbitals $\{\chi_i\}$, and for indistinguishable fermions they can either take the value 0 if the orbital is empty, and 1 if it is occupied. This notation is called occupation number representation, and it suggests a very straightforward mapping to a system of 2K qubits. Namely, we can simply consider $|z_0, z_1, ..., z_{2k-1}\rangle$ where $z_i = n_i$. Now $|z_0, z_1, ..., z_{2k-1}\rangle$ is a basis vector in a Hilbert space of 2K qubits. The only difficulty is to define a set of qubit operators that inherits the same properties of fermionic operators acting on Slater determinants. Note that $a_p|n_0, n_1, ..., 0_p, ..., n_{2K-1}\rangle = 0$ and $a_p|n_0, n_1, ..., 1_p, ..., n_{2K-1}\rangle = (-1)\sum_{i=0}^{p-1} n_i|n_0, n_1, ..., 0_p, ..., n_{2K-1}\rangle$, thus we see that an appropriate definition for an equivalent qubit operator is

$$a_p \to q_p^{JW} := \frac{1}{2} (\sigma_p^x + i\sigma_p^y) \sigma_0^z \cdots \sigma_{p-1}^z.$$
 (2.11)

This mapping is called the Jordan-Wigner transform. Note that the qubit operator not only acts on the *p*th qubit, but also on the qubits from 0 to p - 1. This can be inconvenient for large 2Kbecause simulating these operators with a quantum circuit might require a huge amount of gates.

The function of the σ^z product in (2.11) is to multiply by 1 or -1 depending on the parity of $\sum_{i=0}^{p-1} n_i$. We see that storing these parities could actually be useful. Hence, we can think of defining the qubit basis states as $|z_0, z_1, ..., z_{2k-1}\rangle$ where $z_i = \sum_{j=0}^{i} n_j \mod (2)$. It is clear that if we know all the z_i , then we can recover the occupation numbers n_i and vice versa. After applying a_p to the fermionic state we would need to update the parity of the qubits from p to 2K - 1 in the qubit state, thus the corresponding qubit operator is

$$a_p \to q_p^{par} := \frac{1}{2} (\sigma_p^x \sigma_{p-1}^z + i \sigma_p^y) \sigma_p^x \cdots \sigma_{2K-1}^x.$$
 (2.12)

This mapping is called the parity transform. It represents no advantage with respect to the Jordan-Wigner transform because the corresponding qubit operators also act on several qubits. In fact, both the Jordan-Wigner and the parity transform qubit operators act on $\mathcal{O}(2K)$ qubits. We could think of a transformation that was something in between the two, with some minimal sets of qubits encoding the occupation numbers and the parities. Then, the corresponding qubit operators would be

$$a_p \to q_p^{BK} := \frac{1}{2} (\sigma_{U(p)}^x \sigma_p^x \sigma_{P(p)}^z + i \sigma_{U(p)}^x \sigma_p^y \sigma_{P(p)}^z), \qquad (2.13)$$

where $\sigma_{U(p)}^x = \prod_{j \in U(p)} \sigma_j^x$ and $\sigma_{P(p)}^z = \prod_{j \in P(p)} \sigma_j^z$. U(p) denotes the set of qubits that need to be updated upon flipping the occupation number n_p , and P(p) the set of qubits that encode the sum $\sum_{i=0}^p n_i$. The Bravyi-Kitaev is an optimal transformation in the sense that its U(p) and P(p) sets only involve $\mathcal{O}(\log 2K)$ qubits.

2.2 Adiabatic Quantum Computing

The adiabatic theorem of quantum mechanics states that a system will remain in its instantaneous eigenstate (e.g. the ground state) if a perturbation acts on the system sufficiently slowly. Another condition that is necessary is the existence of a non-vanishing energy gap between that eigenstate and the rest of the eigenspectrum throughout the evolution.

This theorem has found interesting applications, one of them is adiabatic quantum computing (AQC). AQC works by preparing a simple initial eigenstate of a Hamiltonian H_B , and then letting it evolve slowly for a certain time τ under a time dependent Hamiltonian $H_{\tau}(t)$ that goes from $H_B = H_{\tau}(0)$ to $H_P = H_{\tau}(\tau)$. Here, H_P is the Hamiltonian of the system of interest, whose corresponding eigenstate encodes the solution to the computational problem. If the evolution has been done "correctly", then the evolved state should have a large overlap with the eigenstate of H_P .

To be more precise, we can define the adiabatic transition parameter $s \equiv t/\tau \in [0, 1]$ and consider $H(s) = H_{\tau}(\tau s)$. This way τ controls the rate of change of the Hamiltonian. We will consider interpolating Hamiltonians of the type

$$H(s) = A(s)H_B + B(s)H_P \tag{2.14}$$

where A(s) is a monotonically decreasing function from 1 to 0 and B(s) is a monotonically increasing function from 0 to 1. The quantum system will evolve according to the time dependent Schrödinger equation

$$\frac{\hbar}{\tau}\frac{d}{ds}|\psi_{\tau}(s)\rangle = -iH(s)|\psi_{\tau}(s)\rangle.$$
(2.15)

Following the notation of [10], we denote by $|\ell; s\rangle$ the instantaneous eigenstates of the system, so that $H(s)|\ell; s\rangle = E_{\ell}(s)|\ell; s\rangle$. The $\{E_{\ell}(s)\}$ are the instantaneous eigenvalues, and we will suppose that they follow the ordering $E_0(s) \leq E_1(s) \leq \cdots \leq E_{N-1}(s)$ where N is the dimension of the Hilbert state. Suppose that the initial state is prepared in the ground state of $H_B = H(0)$, i.e., $|\psi_{\tau}(0)\rangle = |\ell = 0, s = 0\rangle$. Then if $E_1(s) - E_0(s) > 0$ for all $s \in [0, 1]$, according to the adiabatic theorem:

$$\lim_{\tau \to \infty} |\langle \ell = 0; s = 1 | \psi_{\tau}(1) \rangle| = 1,$$
(2.16)

where $|\psi_{\tau}(1)\rangle$ is the final state of the evolution. However, in practice making $\tau = \infty$ would not be of much use. A better study of the adiabatic theorem tells us that by taking

$$au \gg \frac{\varepsilon}{(\Delta E_{\min})^2}, ag{2.17}$$

where

$$\varepsilon = \max_{s \in [0,1]} |\langle \ell = 1, s | \frac{dH(s)}{ds} | \ell = 0, s \rangle|, \quad \text{and} \quad \Delta E_{\min} = \min_{s \in [0,1]} \Delta E = \min_{s \in [0,1]} E_1(s) - E_0(s), \quad (2.18)$$

we could make $|\langle \ell = 0; s = 1 | \psi_{\tau}(1) \rangle|$ as close to one as we wanted. ΔE_{\min} is called the minimum gap, and as we have seen, it determines the speed at which the transition from H_B to H_P can be made.

Adiabatic state preparation is just one of the applications of AQC among many others, such as

solving certain types of classical optimization problems [9, 6].

2.3 Entanglement entropy

Entanglement is a property that differentiates quantum from classical systems. It appears when the state of one of the parts of the system cannot be described independently from the state of the others. For instance, in a system of two qubits A and B the state $|\Psi^{AB}\rangle = (|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B)/\sqrt{2}$ is entangled because it cannot be written as $|\phi_1\rangle_A \otimes |\phi_2\rangle_B$. The latter is called a product state.

A measure of the entanglement of a system is given by the entanglement entropy, but before we explain how to obtain it, we first need to define the concepts of density matrix and partial trace. Given a pure state $|\psi\rangle$, its density matrix is defined as $\rho = |\psi\rangle\langle\psi|$. Sometimes we might not now for sure if the system was prepared in a pure state $|\psi\rangle$, but rather we might now that with probability p_1 it is in state $|\psi_1\rangle$, with probability p_2 it is in $|\psi_2\rangle$, etc. This situation is denoted by $\{p_i, |\psi_i\rangle\}$, and in this case we say that we have an ensemble of pure states, or a mixed state. The corresponding density matrix is defined by $\rho = \sum_{i=1} p_i |\psi_i\rangle\langle\psi_i|$. For example, the density matrix of the entangled two-qubit pure state we discussed before is

$$\rho^{AB} = \frac{|0\rangle\langle 0|_A \otimes |1\rangle\langle 1|_B + |0\rangle\langle 1|_A \otimes |1\rangle\langle 0|_B + |1\rangle\langle 0|_A \otimes |0\rangle\langle 1|_B + |1\rangle\langle 1|_A \otimes |0\rangle\langle 0|_B}{2}$$

In linear algebra the trace is a linear mapping, and for a square matrix it is defined as the sum of the elements in its diagonal. In the notation we use for density matrices, it can be shown that the trace is given by $\text{Tr}(|\psi\rangle\langle\psi|) = \langle\psi|\psi\rangle$. If we have a density matrix corresponding to a state in the composite Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$, then it is possible to define a partial trace by taking the trace over only one of the parts of the system. For example, $\text{Tr}_B(|\phi_1\rangle\langle\phi_1|_A \otimes |\phi_2\rangle\langle\phi_2|_B) = |\phi_1\rangle\langle\phi_1|_A \operatorname{since}\langle\phi_2|\phi_2\rangle = 1$. In the two entangled qubit example, taking the partial trace with respect qubit *B* gives

$$\rho^{A} = \operatorname{Tr}_{B}(\rho^{AB}) = \frac{|0\rangle\langle 0|_{A}\langle 1|1\rangle + |0\rangle\langle 1|_{A}\langle 1|0\rangle + |1\rangle\langle 0|_{A}\langle 0|1\rangle + |1\rangle\langle 1|_{A}\langle 0|0\rangle}{2}$$
$$= \frac{|0\rangle\langle 0|_{A} + |1\rangle\langle 1|_{A}}{2}.$$

Notice that in both cases we obtain a reduced density matrix. However, if the system is entangled then the reduced density matrix corresponds to a mixed state, this is a general fact. Now for a system that can be divided in two parts, the entanglement entropy is defined as the Von Neumann entropy of the reduced density matrix with respect to one of the parts, namely

$$S(\rho^{A}) = -\operatorname{Tr}(\rho^{A} \log_{2} \rho^{A}) = -\operatorname{Tr}(\rho^{B} \log_{2} \rho^{B}) = S(\rho^{B}).$$
(2.19)

The result is independent of which part is chosen, either A or B, provided that the original density matrix ρ^{AB} came from a pure state. The best way of computing the Von Neumann entropy is to calculate the eigenvalues of the density matrix, which can always be done because density matrices are hermitian and positive semi-definite. Then, $S(\rho) = -\sum_k \lambda_k \log_2 \lambda_k$, where λ_k are the eigenvalues of ρ . It can be shown that the Von Neumann entropy of a pure state density matrix is zero. On the other hand, for a maximally mixed state the Von Neumann entropy is $\log_2(N)$, where N is the dimension of the Hilbert space. For the entangled two-qubit example we have that $S(\rho^A) = \log_2(2) = 1$. This is the maximum value that the entanglement entropy could take, thus we deduce that the original $|\Psi^{AB}\rangle$ was maximally entangled.

Chapter 3

Methodology

Obtaining the electronic Hamiltonian in second quantization form and mapping it to qubits.

We study the adiabatic preparation of the ground state of the H_2 and the LiH molecules. As explained in the theory section, the electronic Hamiltonian depends on the fixed positions of the nuclei. Since both H_2 and LiH are diatomic molecules, we only need to specify the distance r between the two nuclei that form the molecule. Thus, we consider the electronic Hamiltonian (2.4)for several inter-atomic distances ranging from $0.2r_0$ to $4r_0$ for H_2 , and from $0.2r_0$ to $2.1r_0$ for LiH. Here r_0 denotes the inter-atomic distance at the equilibrium geometry of the molecules, which is $r_0 = 0.74$ Å for H_2 , and $r_0 = 1.45$ Å for LiH. Once we have fixed the inter-atomic distance, we compute the coefficients h_{ij} (2.5) and h_{ijkl} (2.6) with the PySCF module of the OpenFermion package [18]. These one- and two-body integrals depend on the basis set used to express the spin orbitals, in this case we use a STO-3G basis, which approximates spin orbitals of the Slater type as the sum of three Gaussian functions. The dihydrogen molecule is optimally described by four spin orbitals, whereas the lithium hydride would in principle require 12 spin orbitals in the minimal basis. However, to ease our calculations we can specify an active space, namely we can assume that the lowest energy spatial orbital of LiH is fully occupied and the higher energy ones are not accessible. After this simplification, we only need to consider four spin orbitals for the lithium hydride. Once we have obtained the electronic Hamiltonian in second quantization form (2.4), we also use OpenFermion to compute the Jordan-Wigner and the Bravyi-Kitaev transforms. Notice that for both H_2 and LiH, their electronic Hamiltonians are mapped into a four qubit Hilbert space. Since for the rest of the calculations we use the QuTiP package [14], we have written a Python function that transforms OpenFermion qubit Hamiltonians to QuTiP quantum objects. See also the code appendix **B**.

Writing the time-dependent Hamiltonian and solving the Schrödinger equation.

Using the same notation as in the theory section, we identify the final Hamiltonian H_P of the adiabatic evolution with the electronic Hamiltonian of either H_2 or LiH once mapped to qubits. As initial Hamiltonian we use

$$H_B = \frac{1}{2} \sum_{i=0}^{3} (1 - \sigma_i^x).$$
(3.1)

Note that the ground state of H_B is $|+,+,+,+\rangle^1$, which in the computational basis becomes an

 $^{^{1}|+\}rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ denotes the eigenstate of σ_x with eigenvalue 1.

homogeneous superposition of all the basis states. Thus this choice of initial Hamiltonian describes a general situation in which no information about the true ground state of H_P was available. Then, we consider adiabatic Hamiltonians of the form

$$H(s) = (1 - s^{\alpha})H_B + s^{\alpha}H_P, \qquad (3.2)$$

where $s = t/\tau \in [0, 1]$, and τ is the total time of the evolution. Unless otherwise specified, we set $\alpha = 1$ so that the interpolation path between H_B and H_P is linear. At some point we also investigate non-linear paths by considering different $0 < \alpha < 1$ exponents.

To study the adiabatic state preparation of the ground states of H_2 and LiH, we evolve the nondegenerate ground state of the initial Hamiltonian H_B under the time dependent Hamiltonian (3.2) during a certain time τ . We try different evolution times ranging from $1 \hbar E_h^{-1}$ to 30000 $\hbar E_h^{-1}$. Note that $1 \hbar E_h^{-1} \approx 10^{-17}$ s is the atomic unit for time, and also $1 E_h \approx 10^{-18}$ J is the atomic unit for energy. To obtain the instantaneous states along the evolution we solve the time dependent Schrödinger equation (2.15) using the QuTiP mesolve function. We also use QuTiP to calculate the eigenvalues and eigenstates of all the Hamiltonians considered, and the overlaps between different states. Finally, just to clarify the notation:

- $|\ell = 0; s\rangle$ denotes the instantaneous ground state of H(s). In general, $|\ell = i; s\rangle$ denotes the *i*th eigenstate of H(s) with energy $E_i(s)$.
- $|\psi_{\tau}(s)\rangle$ denotes the instantaneous evolved state.
- $|\psi_{target}\rangle$ denotes the ground state of the final Hamiltonian H_p . In fact, $|\psi_{target}\rangle = |\ell = 0; 1\rangle$.

The final probability of success P for a certain τ is quantified by the square overlap $|\langle \psi_{\tau}(1)|\psi_{target}\rangle|^2$. We consider that an evolution has been successful if $P \ge 0.99$, and we refer to the minimum time required for a successful evolution as the critical time.

Measuring the entanglement.

We use the entanglement entropy as a measure of the entanglement of a state. To this end, we need to divide the four qubit system in two parts, which can be achieved in multiple ways. To reduce this ambiguity we follow the same approach as [12]. We consider any pair of qubits as subsystem A and the other two as subsystem B. There are in total $\binom{4}{2}$ possible two-qubit partitions, namely $\Gamma = \{[0,1], [0,2], [0,3], [1,2], [1,3], [2,3]\}$. However, since we are only concerned with pure states, we have that $S(\rho^{[0,1]}) = S(\rho^{[2,3]}), S(\rho^{[0,2]}) = S(\rho^{[1,3]})$ and $S(\rho^{[0,3]}) = S(\rho^{[1,2]})$, so in fact we only need to consider $\Gamma = \{[0,1], [0,2], [0,3]\}$. Thus our measure of entanglement will be given by the entanglement entropy maximized over all two-qubit partitions, i.e., $S_2^{max} = \max_{A \in \Gamma} S(\rho^A)$. The density matrices, the partial traces and the Von Neumann entropies are calculated using QuTiP's functions.

Chapter 4

Results

4.1 Preliminary study

First of all, we have verified that the mapping from fermions to qubits and the transformation from OpenFermion qubit Hamiltonians to QuTiP quantum objects has been correctly implemented, by verifying that the ground state energies of the H_P Hamiltonians for several inter-atomic distances coincide with the FCI energies provided by OpenFermion. These FCI energies are shown in the bottom picture of figure 4.1 together with the Hartree-Fock energies. Remember that the Hartree-Fock method consists in approximating the electronic ground state by a single Slater determinant, whereas in the FCI all the possible Slater determinants obtained from a certain set of spin orbitals are considered. The difference between the Hartree-Fock and the exact energy is called correlation energy, and it is an important quantity in quantum chemistry because it indicates when the Hartree-Fock approximation works well. For the two molecules, we see that at small inter-atomic distances the Hartree-Fock method offers a very good approximation to the exact energy, whereas the correlation energy tends to increase for larger bond lengths. In fact, the entanglement entropy has been proposed as a measure for this correlation energy [13]. To verify that our calculations are reasonable, we have also checked that the entanglement entropy of the ground state of H_2 and LiH increases together with the correlation energy, see top picture of figure 4.1. This result is not surprising, if the ground state is well approximated by a single Slater determinant, then there is no entanglement because Slater determinants in the occupation number representation can be written as product states, whereas if a superposition of Slater determinants is required, then the entanglement is likely to increase. On the other hand, we have also calculated the energy difference between the ground and the first excited state of the H_P Hamiltonians for several inter-atomic distances. The results can be seen in the middle picture of figure 4.1. This energy difference tends to decrease for larger r/r_0 in both molecules, particularly for H_2 , thus we can already expect that it will be more costly to achieve an adiabatic evolution for the larger bond lengths.

Last but not least, to end with the preliminary study we have investigated the evolution of the energies of H(s) for $r/r_0 = 1$, see figure 4.2. As expected, for s = 0 we have obtained the energies of H_B . Notice that the ground state is non-degenerate, and that during the evolution it maintains a non-vanishing energy gap with the rest of the eigenspectrum. For s = 1 we get the molecular energies.



Figure 4.1: Top: Maximized entanglement entropy of the ground state over all two-qubit partitions for different bond lengths. Middle: exact energy difference between the ground and the first excited state. Bottom: comparison between the ground state FCI and Hartree-Fock energies.



Figure 4.2: Evolution of the energies $E_0(s)$, $E_1(s)$, $E_2(s)$, \cdots , $E_{15}(s)$ of the Hamiltonian H(s) for H_2 (left) and LiH (right). To obtain the final Hamiltonian we have set $r/r_0 = 1$ and used the Jordan-Wigner transform. The bottom line corresponds to the ground state energy.

4.2 Adiabatic state preparation

The energy difference between the ground and the first excited state plays an important role in the efficiency of adiabatic quantum computing. The minimum value of this energy difference during the adiabatic evolution is called the minimum gap, and it is known to be the main limiting factor, see eq. (2.17). In figure 4.3 we show the energy gap for different inter-atomic distances and for both the Jordan-Wigner (solid line) and the Bravyi-Kitaev (dashed line) transforms. There are a couple of observations that can be made. First, notice that the minimum gap can happen for any value of the adiabatic transition parameter. For example, for small bond lengths of the dihydrogen molecule the minimum gap occurs in the middle of the evolution, whereas for larger r/r_0 the minimum gap tends to happen at the end. Secondly, performing either the Jordan-Wigner or the Bravyi-Kitaev transform does not modify the eigenvalues of the electronic Hamiltonian, however,

as we can see in the figure, choosing one or the other can affect the energies of H(s) for $s \in (0, 1)$ resulting in a substantial change in the magnitude of the minimum gap, as it is the case for H_2 . We will discuss more about this fact in the next section.



Figure 4.3: Energy difference between the ground and the first excited state of H(s). Different bond-lengths are represented with different colors. We use a solid line if the H_P Hamiltonian was obtained via the Jordan-Wigner mapping, and dashed line if it was via the Bravyi-Kitaev.

Then, we have fixed the inter-atomic distance to be the one at the equilibrium geometry of the molecules, and we have evolved the ground state of H_B under the H(s) Hamiltonian for different evolution times. The square overlap between the evolved state and the target state as a function of the adiabatic transition parameter can be seen in figure 4.4. For s = 0 this corresponds to the square overlap of $|+,+,+,+\rangle$ and the corresponding electronic ground state, which is quite small because as we wrote before, our choice of H_B made no assumption about the true ground state of H_P . If the evolution is performed sufficiently slowly, then the evolved state remains in the instantaneous ground state of H(s), see figure 4.5, so the square overlap with the target state increases as the adiabatic evolution parameter goes to one. On the other hand, if the evolution is done too rapidly, then the evolved state may acquire enough energy to get excited into the higher energy eigenstates of H(s). This is more likely to happen in the regions near the minimum gap, since the energy barrier is smaller there. In fact, in figure 4.5 we see that for unsuccessful evolutions $(\tau = 5 \ \hbar E_h^{-1})$ the overlap of $|\psi_{\tau}(s)\rangle$ with $|\ell = 0; s\rangle$ decays at the same time that the overlap with $|\ell = 1; s\rangle$ increases, and this process gets accelerated near the minimum gap. However, notice that once passed the minimum gap, also the overlap between $|\psi_{\tau}(s)\rangle$ and $|\ell=1;s\rangle$ decays. This is because from $|\ell = 1; s\rangle$ the evolved state may also have access to the other higher energy eigenstates of H(s) provided that the corresponding energy gaps are sufficiently small.



Figure 4.4: Square overlap between the evolved and the target states for different times τ in atomic units $(\hbar E_h^{-1})$. The inter-atomic distance has been fixed at $r/r_0 = 1$. We used both mappings to obtain H_P , Jordan-Wigner (solid line) and Bravyi-Kitaev (dashed line).



Figure 4.5: Square overlap of the evolved state with the two lowest energy eigenstates of H(s) for successful and unsuccessful evolutions. The inter-atomic distance is $r/r_0 = 1$ and we show only the results for the Jordan-Wigner mapping. The dotted and dashed line indicates the energy gap throughout the evolution.

4.3 Improving the evolution time

If the time required for a successful adiabatic evolution is too large, then the adiabatic state preparation may become a bottleneck in the performance of quantum chemistry simulations with quantum computers. This is why there have been several proposals to speed up the preparation process. For example, [25, 4] suggested using different initial Hamiltonians that do require some knowledge about the true ground state of H_P , such as its Hartree-Fock energy. [25] also showed for methylene, that having an initial state with a large overlap with the target state could significantly reduce the critical time. In that sense, our choice of initial Hamiltonian H_B , despite being the most general, is far from optimal. See figure 4.6, where for the H_2 molecule, we present the overlap of the evolved state with the states of the computational basis throughout the evolution. We have used the Jordan-Wigner mapping since in this case the qubit basis states coincide with



Figure 4.6: Results for the H_2 molecule. Overlap of the evolved state with the qubit basis states along the evolution. The H_P Hamiltonian has been obtained via the Jordan-Wigner mapping so the qubit basis states coincide with the occupation number representation. The inter-atomic distance has been fixed at $r/r_0 = 1$. The red line indicates the adiabatic parameter for which the minimum gap occurs (a) unsuccessful evolution, (b) successful evolution.

the occupation number representation and it is easier to interpret which are the occupied orbitals. We do know that the dihydrogen molecule has two electrons available, in fact, for $r/r_0 = 1$ the electronic ground state consists mostly of $|1, 1, 0, 0\rangle$ (the Hartree-Fock ground state) with a small contribution from the doubly excited state $|0, 0, 1, 1\rangle$, and this contribution grows for larger interatomic distances, see appendix A. Hence, it is unnecessary to consider the basis states with a total occupation number different from two, such as $|0, 0, 0, 0\rangle$, $|1, 0, 0, 0\rangle$, $|1, 1, 1, 0\rangle$, etc., and finding an appropriate initial Hamiltonian H_B which excluded these states could ease the adiabatic evolution.

On the other hand, if the position of the minimum gap is known, then it is possible to reduce the total time of the evolution by decreasing the transition speed in the regions near the minimum gap and increasing it everywhere else. This means finding other interpolating functions A(s) and B(s) which are not necessarily linear in s. Since we have observed that for the LiH molecule the minimum gap occurs near the end of the evolution, we have studied how the non-linear interpolation of (3.2) with $0 < \alpha < 1$ affects the critical time. Notice that this choice makes the evolution faster at the beginning and slower at the end. The results are shown in figure 4.7 (a). We have observed a meaningful improvement in the critical times required for $\alpha = 0.5$ and $\alpha = 0.2$ with respect to the linear interpolation, $\alpha = 1$. However, we must mention that for $\alpha \leq 0.1$ we have found critical times above $40000 \ \hbar E_h^{-1}$. One possible reason for this is that the evolution rate becomes infinitely fast for s approaching zero, in fact dH(s)/ds diverges. Another reason could be that the system spends more time in a region with a smaller energy gap, see figure 4.7 (b). In the same line of finding other interpolating paths, it has also been proposed adding a catalyst Hamiltonian that appeared only for $s \in (0, 1)$ to aid in the evolution [8].

Overall, the magnitude of the minimum gap is the most limiting factor in the speed of a successful adiabatic evolution. In figure (4.3) we have shown that the size of the minimum gap could be affected by the choice of either the Jordan-Wigner or the Bravyi-Kitaev mappings, and this was



Figure 4.7: Results for the LiH molecule. The H_P Hamiltonian has been obtained via the Jordan-Wigner mapping.(a) Minimum time required for a successful evolution (final overlap between the evolved and the target state of at least 0.99). The different lines correspond to different exponents α . (b) Results shown correspond to the equilibrium geometry. Top: energy gap for different exponents. Bottom: The non-linear interpolation makes the evolution faster at the beginning, when the gap is larger, and slower at the end.



Figure 4.8: Results for the H_2 molecule. The red points represent the Bravyi-Kitaev final Hamiltonian, and the blue ones the Jordan-Wigner. (a) Minimum times required for a successful evolution. The inset represents the ratio between these two times. (b) Critical time as a function of the minimum gap. The points have been obtained from different inter-atomic distances. The solid lines represent the fit of the data points to a function of the type $y = a/x^2 + b$. For Jordan-Wigner we found $a = (1.59 \pm 0.05) \ \hbar E_h, \ b = (1.5 \pm 0.7) \times 10^2 \ \hbar E_h^{-1}$, and for Bravyi-Kitaev $a = (3.21 \pm 0.05) \ \hbar E_h, \ b = (1.6 \pm 0.7) \times 10^2 \ \hbar E_h^{-1}$.

particularly dramatic for the H_2 molecule. In figure 4.8 (a) we see how for several inter-atomic distances the Jordan-Wigner transform offers successful evolution times from two to ten times faster than the Bravyi-Kitaev transform. In figure 4.8 (b) we have represented for both mappings the critical times as a function of the size of the minimum gap. By fitting these data points to a function of the type $y = a/x^2 + b$, we have verified the scaling of the critical times as $\mathcal{O}((\Delta E_{min})^{-2})$, which is the result expected from the theory, see eq. (2.17). On the other hand, the values of the parameter a are different for the Jordan-Wigner and the Bravyi-Kitaev transforms, although they have the same order of magnitude. Hence, for the H_2 molecule, we attribute the different performance of the two mappings to the different minimum gaps that they provide (figure 4.3 (a)), rather than to some intrinsic difference. However, when increasing the number of spin orbitals considered, and therefore, the number of qubits, the Bravyi-Kitaev transform is supposed to be advantageous in actual simulations with a quantum computer due to the lower number of qubit operations that it requires.

4.4 The role of entanglement



Figure 4.9: Results for the H_2 molecule at the equilibrium geometry $(r/r_0 = 1)$. The solid lines represent the maximized entanglement entropy of the evolved state, and the dashed grey line the maximized entanglement entropy of the instantaneous ground state. The dashed and dotted line is the energy gap along the evolution. Notice that the maximum of the entanglement entropy of $|\ell = 0; s\rangle$ happens near the minimum gap.

In figure 4.9 we show the maximized entanglement entropy of the evolved state for different evolution times. There are a couple points to consider. First, it can be seen that as the time of the evolution increases, the entanglement entropy of the evolved state tends to that of the instantaneous ground state $|\ell = 0; s\rangle$. This fact is not surprising, because in successful evolutions we expect the evolved state to remain in the instantaneous ground state of H(s). On the other hand, shorter transition times, and therefore, non-adiabatic evolutions, give rise to final states with large entanglement entropy. This phenomenon has also been observed in AQC applied to solve classical optimization problems [12]. In fact, the authors of [12] were able to find an upper bound for the final entanglement entropy in terms of the final probability of success:

$$S_l^{end} \le -P\log_2(P) - (1-P)\log_2(1-P) + (1-P)\log_2(2^l - 1), \tag{4.1}$$

valid for P > 1/e and if the target state is not entangled. Here, l is the number of qubits in the subsystem considered, which in our case is 2. However, there is a fundamental difference between AQC for adiabatic state preparation, and for solving optimization problems. Namely, in ASP the final Hamiltonian describes a molecule, which is a quantum system and therefore its ground state might present entanglement. On the other hand, the final Hamiltonians used in optimization problems are just classical cost functions whose terms have been mapped to Pauli operators. Thus if there is a unique solution, its ground state is a product state. As we can see in figure 4.10 (a), this fact makes the bound (4.1) not valid in general for ASP. Actually, its accuracy will be linked to that of the Hartree-Fock approximation. In figure 4.10 (b) we also see the maximized entanglement entropy of the evolved state as a function of the total evolution time. For evolution times close to $0 \hbar E_h^{-1}$, the entanglement is small because the system does not have time to change substantially from the initial state, which was not entangled. However, entanglement does grow rapidly. For larger total evolution times the entanglement of the final state decays slowly until it reaches the entanglement entropy of the target state, as expected in successful evolutions.



Figure 4.10: Results for the H_2 molecule. (a) Entanglement of the final state as a function of the final probability of success. The grey dashed line represents the bound for classical optimization problems (4.1). The electronic ground state for $r/r_0 = 2$ is more entangled than that for $r/r_0 = 1$, thus the exceeding of the bound becomes more obvious. The different points have been obtained from different total evolution times. (b) Top: final probability of success for different evolution times. Bottom: the points represent the final entanglement and the grey lines the entanglement along the evolution.

In figure 4.9 we also see that the maximum entanglement entropy of the instantaneous ground state seems to happen near the minimum gap. We have verified that this is a general fact by representing the adiabatic parameter with maximum entanglement of the instantaneous ground state vs. the adiabatic parameter with minimum gap energy, see figure 4.11. In fact, we have established that both adiabatic parameters are linearly correlated with a slope slightly above 1. Some authors have pointed out that this could be a consequence of systems undergoing quantum phase transitions during adiabatic evolutions near the minimum gap [16, 20]. It has been well studied that the entanglement entropy of the ground state of many quantum spin systems scales at quantum phase transitions [26, 17]. We have also observed that the maximum entanglement entropy of the instantaneous ground state and the minimum gap are inversely correlated, see figure 4.11 (b). This means that entanglement will be larger for smaller minimum gaps.



Figure 4.11: Results for the H_2 molecule. The points of both subfigures were obtained by collecting the results for different inter-atomic distances. (a) Adiabatic parameter with maximum entanglement entropy of the instantaneous ground state vs. adiabatic parameter with minimum gap energy. The data points are linearly correlated and the dashed line represents the linear fit. For the Jordan-Wigner points the results of the linear regression are: $R^2 = 0.999$, slope 1.20 ± 0.02 , intercept -0.22 ± 0.02 ; and for the Bravyi-Kitaev: $R^2 = 0.999$, slope 1.11 ± 0.02 , intercept -0.11 ± 0.02 . (b) Magnitude of the maximum entanglement of the instantaneous ground state vs. magnitude of the minimum gap.

Chapter 5

Conclusion

While developing this project we have learned about the potential of using existing Python packages, namely OpenFermion [18] and QuTiP [14], for the classical simulation of quantum systems. These packages, combined with our own code, have allowed us to perform a study of the adiabatic state preparation (ASP) of the ground states of dihydrogen and lithium hydride, which otherwise would have required expert knowledge of computational chemistry methods.

With regard to the ASP study, among other results, we have observed that the minimum energy gap can appear for any value of the adiabatic transition parameter. In contrast to previous works, we have performed and compared both, the Jordan-Wigner and the Bravyi-Kitaev transforms, rather than focusing only on one fermion to qubit mapping. This has allowed us to see that the magnitude of the minimum gap may also depend on the mapping used to obtain the final Hamiltonian. Guaranteeing a short evolution time for a successful evolution is key in the efficiency of ASP. Thus we have discussed some possible ways in which this time can be reduced. The conclusion we draw is that having a good knowledge of the position and magnitude of the minimum gap is fundamental to design optimal adiabatic evolution strategies. In fact, we have seen that the entanglement entropy can constitute a good indicator of both, the position and the magnitude of the minimum gap, which is backed up by the analogy of adiabatic quantum evolutions with quantum phase transitions.

This work can also be extended in several directions. For instance, we have observed that the upper bound for the final entanglement entropy (4.1) in terms of the final probability of success is not valid for entangled target states. Thus, it could be useful to generalize the bound (4.1) to adiabatic quantum computing (AQC) problems whose target ground state might present entanglement, such as ASP and the electronic ground states considered in this work. Our code also allows extending the ASP study to other molecules without further complications, as long as the number of spin orbitals considered is kept under 11 or 12, which are the dimensions that QuTiP can manage well. However, the calculations might become substantially slow. Finally, some works in AQC also look at the scaling of the gap and other magnitudes with the number of qubits. It could be worth doing similar studies by increasing the number of spin orbitals considered.

Bibliography

- Molecular orbital diagrams. https://chem.libretexts.org/@go/page/294891. Accessed: May 24, 2021.
- [2] What is quantum annealing? https://docs.dwavesys.com/docs/latest/c_gs_2.html# what-is-quantum-annealing. Accessed: May 28, 2021.
- [3] T. Albash and D. A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.
- [4] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.
- [5] R. Babbush, P. J. Love, and A. Aspuru-Guzik. Adiabatic quantum simulation of quantum chemistry. *Scientific reports*, 4(1):1–11, 2014.
- [6] C. Bauckhage, E. Brito, K. Cvejoski, C. Ojeda, R. Sifa, and S. Wrobel. Adiabatic quantum computing for binary clustering. arXiv preprint arXiv:1706.05528, 2017.
- [7] M. Born and V. Fock. Beweis des adiabatensatzes. Zeitschrift f
 ür Physik, 51(3-4):165–180, 1928.
- [8] E. Farhi, J. Goldstone, and S. Gutmann. Quantum adiabatic evolution algorithms with different paths. arXiv preprint quant-ph/0208135, 2002.
- [9] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.
- [10] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106, 2000.
- [11] R. P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21(6/7), 1982.
- [12] P. Hauke, L. Bonnes, M. Heyl, and W. Lechner. Probing entanglement in adiabatic quantum optimization with trapped ions. *Frontiers in Physics*, 3:21, 2015.
- [13] Z. Huang and S. Kais. Entanglement as measure of electron-electron correlation in quantum chemistry calculations. *Chemical physics letters*, 413(1-3):1–5, 2005.
- [14] J. R. Johansson, P. D. Nation, and F. Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

- [15] V. Kremenetski, C. Mejuto-Zaera, S. J. Cotton, and N. M. Tubman. Simulation of adiabatic quantum computing for molecular ground states. arXiv preprint arXiv:2103.12059, 2021.
- [16] J. I. Latorre and R. Orús. Adiabatic quantum computation and quantum phase transitions. *Physical Review A*, 69(6), Jun 2004.
- [17] J. I. Latorre, E. Rico, and G. Vidal. Ground state entanglement in quantum spin chains. arXiv preprint quant-ph/0304098, 2003.
- [18] J. R. McClean, N. C. Rubin, K. J. Sung, I. D. Kivlichan, X. Bonet-Monroig, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, et al. Openfermion: the electronic structure package for quantum computers. *Quantum Science and Technology*, 5(3):034014, 2020.
- [19] M. A. Nielsen and I. Chuang. Quantum computation and quantum information. American Association of Physics Teachers, 2002.
- [20] J. Rodríguez-Laguna and S. N. Santalla. Building an adiabatic quantum computer simulation in the classroom. American Journal of Physics, 86(5):360–367, 2018.
- [21] G. Schaller and R. Schützhold. The role of symmetries in adiabatic quantum algorithms. 2009.
- [22] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science, pages 124–134. Ieee, 1994.
- [23] M. Streif, F. Neukart, and M. Leib. Solving quantum chemistry problems with a d-wave quantum annealer. In *International Workshop on Quantum Technology and Optimization Problems*, pages 111–122. Springer, 2019.
- [24] A. Szabo and N. S. Ostlund. Modern quantum chemistry: introduction to advanced electronic structure theory. Courier Corporation, 2012.
- [25] L. Veis and J. Pittner. Adiabatic state preparation study of methylene. The Journal of chemical physics, 140(21):214111, 2014.
- [26] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev. Entanglement in quantum critical phenomena. *Physical review letters*, 90(22):227902, 2003.

Appendices

Appendix A

Complementary results

A.1 Occupation number representation

The following tables present the square overlap of the true ground states of H_2 and LiH with the basis states in the occupation number representation for different inter-atomic distances.

Basis	$r/r_0 = 1$	$r/r_{0} = 2$	$r/r_0 = 3$
$ 0,0,0,0\rangle$	0	0	0
0,0,0,1 angle	0	0	0
0,0,1,0 angle	0	0	0
0,0,1,1 angle	0.01267	0.12058	0.34870
0,1,0,0 angle	0	0	0
0,1,0,1 angle	0	0	0
0,1,1,0 angle	0	0	0
0,1,1,1 angle	0	0	0
1,0,0,0 angle	0	0	0
1,0,0,1 angle	0	0	0
1,0,1,0 angle	0	0	0
1,0,1,1 angle	0	0	0
1,1,0,0 angle	0.9873	0.87942	0.65130
1,1,0,1 angle	0	0	0
1,1,1,0 angle	0	0	0
$ 1,1,1,1\rangle$	0	0	0

Table A.1: Square overlap between the ground state of H_2 and the basis functions for different inter-atomic distances.

Basis	$r/r_0 = 1$	$r/r_0 = 2$
0,0,0,0 angle	0	0
0,0,0,1 angle	0	0
0,0,1,0 angle	0	0
0,0,1,1 angle	0.00031	0.03852
0,1,0,0 angle	0	0
0,1,0,1 angle	0	0
0,1,1,0 angle	3×10^{-5}	0.02162
0,1,1,1 angle	0	0
1,0,0,0 angle	0	0
1,0,0,1 angle	3×10^{-5}	0.02162
1,0,1,0 angle	0	0
1,0,1,1 angle	0	0
1,1,0,0 angle	0.99964	0.91823
1,1,0,1 angle	0	0
1,1,1,0 angle	0	0
$ 1,1,1,1\rangle$	0	0
$ 1, 1, 1, 0\rangle$ $ 1, 1, 1, 1\rangle$	0 0	0 0

Table A.2: Square overlap between the ground state of LiH and the basis functions for different inter-atomic distances.

A.2 Adiabatic evolutions

The figures presented in this section show the square overlap of the evolved states with the computational basis states along the evolutions. Figure A.1 shows the results for the dihydrogen molecule and for different inter-atomic distances, and figure A.2 shows the analogous results for the lithium hydride molecule. The times τ used are enough to provide successful adiabatic evolutions.



Figure A.1: Results for H_2 . Overlap of the evolved state in a successful adiabatic evolution with the qubit basis vectors of the Jordan-Wigner mapping, equivalents to the occupation number representation. The red line represents the adiabatic transition parameter for which the minimum energy gap occurs. For large r/r_0 we observe a strong transition just before this gap.



Figure A.2: Results for LiH. Overlap of the evolved state in a successful adiabatic evolution with the qubit basis vectors of the Jordan-Wigner mapping, equivalents to the occupation number representation. The red line represents the adiabatic transition parameter for which the minimum energy gap occurs. For large r/r_0 we observe a strong transition just before this gap.

A.3 Entanglement results of the lithium hydride

In the Results chapter we have preferred to concentrate on the results for the entanglement of the dihydrogen molecule, because the different inter-atomic distances offered very diverse positions and magnitudes of the minimum energy gap, which made the analysis clearer. However, notice that for LiH the minimum gap and the maximum entanglement of the instantaneous ground state also happen around the same moment of the adiabatic evolution, and that the magnitudes of the maximum entanglement and the minimum energy gap are inversely correlated.



Figure A.3: Results for the LiH molecule. The points of both subfigures were obtained by collecting the results for different inter-atomic distances. (a) Adiabatic parameter with maximum entanglement entropy of the instantaneous ground state vs. adiabatic parameter with minimum gap energy. (b) Magnitude of the maximum entanglement of the instantaneous ground state vs. magnitude of the minimum gap.

Appendix B

Code

Generating the electronic Hamiltonians in second quantization form and performing the Jordan-Wigner and Bravyi-Kitaev transforms with OpenFermion.

```
1 # ----- H2 molecule -----
2 import openfermion
3 from openfermion.transforms import get_fermion_operator
4 from openfermion.transforms import jordan_wigner
5 from openfermion.transforms import bravyi_kitaev
6 from openfermion import FermionOperator
7 from openfermion.chem import MolecularData
8 from openfermionpyscf import run_pyscf
11 bond_length = 0.74
12 geometry = [['H', [0,0,0]], ['H', [0,0, bond_length]]]
13 basis = 'sto-3g'
14 multiplicity = 1
15 charge = 0
16 h2_molecule = MolecularData(geometry, basis, multiplicity, charge)
17
18 h2_molecule = run_pyscf(h2_molecule,
                         run_mp2 = True,
19
                         run_cisd = True,
20
                         run_ccsd = True,
21
                          run_fci = True)
23 h2_filename = h2_molecule.filename
24 h2_molecule.save()
25
26 h2_molecule.load()
27 h2_hamiltonian = h2_molecule.get_molecular_hamiltonian()
28 fermion_hamiltonian = get_fermion_operator(h2_hamiltonian)
29 qubit_hamiltonian_JW = jordan_wigner(fermion_hamiltonian)
30 qubit_hamiltonian_BK = bravyi_kitaev(fermion_hamiltonian, h2_hamiltonin.n_qubits)
1 # ----- LiH molecule -----
2 import openfermion
3 from openfermion.transforms import get_fermion_operator
4 from openfermion.transforms import jordan_wigner
5 from openfermion.transforms import bravyi_kitaev
6 from openfermion import FermionOperator
7 from openfermion.chem import MolecularData
```

```
8 from openfermionpyscf import run_pyscf
10
11 bond_length = 1.45
12 basis = 'sto-3g'
13 multiplicity = 1
14 charge = 0
15 description = str(round(bond_length,2))
16 geometry = [('Li', (0., 0., 0.)), ('H', (0., 0., bond_length))]
17 LiH_molecule = MolecularData(
          geometry, basis, multiplicity, description=description)
18
19
20
21 LiH_molecule = run_pyscf(LiH_molecule,
22
                           run_mp2 = True,
                           run_cisd = True,
23
                           run_ccsd = True,
24
                           run_fci = True)
25
26 lih_filename = LiH_molecule.filename
27 LiH_molecule.save()
28
29 LiH_molecule.load()
30 active_space_start = 1
31 active_space_stop = 3
32 lih_hamiltonian = LiH_molecule.get_molecular_hamiltonian(
                       occupied_indices=range(active_space_start),
33
34
                       active_indices=range(active_space_start,active_space_stop))
35
36 fermion_hamiltonian = get_fermion_operator(lih_hamiltonian)
37 qubit_hamiltonian_JW = jordan_wigner(fermion_hamiltonian)
38 qubit_hamiltonian_BK = bravyi_kitaev(fermion_hamiltonian, lih_hamiltonian.n_qubits)
```

Function to transform OpenFermion qubit Hamiltonians to QuTiP quantum objects.

```
1 from qutip import *
2
3 def qubit_hamiltonian_to_qutip(q_hamiltonian, N):
      # q_hamiltonian is the OpenFermion qubit Hamiltonian
4
      \# N is the number of qubits
5
      # Returns: qubit Hamiltonian as qutip quantum object
6
      si = qeye(2)
7
      sx = sigmax()
8
      sy = sigmay()
9
      sz = sigmaz()
      sx_list = []
12
      sy_list = []
13
      sz_list = []
14
15
      for n in range(N):
16
           op_list = []
17
           for m in range(N):
18
               op_list.append(si)
19
20
           si_n = tensor(op_list)
21
           op_list[n] = sx
23
           sx_list.append(tensor(op_list))
24
```

```
op_list[n] = sy
25
           sy_list.append(tensor(op_list))
26
27
           op_list[n] = sz
28
           sz_list.append(tensor(op_list))
29
30
      # read the qubit_hamiltonian from openfermion
31
      l_h = list(q_hamiltonian)
32
      s_h_0 = str(1_h[0])
33
      t1_0 = s_h_0.split()
34
      numero_0 = t1_0[0].split('+')[0].split('(')[1]
35
      numero_0 = float(numero_0)
36
      H = numero_0*si_n
37
38
      for i in range(1,len(l_h)):
39
           s_h = str(l_h[i])
40
           t1 = s_h.split()
41
           numero = t1[0].split('+')[0].split('(')[1]
42
           numero = float(numero)
43
          lletres = [t1[1][1]]
44
          numeros = [int(t1[1][2])]
45
          lletres2 = [j[0] for j in t1[2:]]
46
           numeros2 = [int(j[1]) for j in t1[2:]]
47
          let = lletres+lletres2
48
          num = numeros+numeros2
49
          H_i = numero*si_n
50
51
          for k in range(0,len(let)):
              if let[k] == 'X':
                   H_i = H_i*sx_list[num[k]]
53
              if let[k] == 'Y':
54
                   H_i = H_i*sy_list[num[k]]
55
               if let[k] == 'Z':
56
57
                   H_i = H_i*sz_list[num[k]]
               else :
58
                   H_i = H_i
59
           H = H + H_i
60
61
      return H
62
```

Simulating the time dependent evolution with QuTiP.

```
1 from qutip import *
2 import numpy as np
4
5 # Function to generate the initial Hamiltonian
6 def generate_Hb(N):
      \# N is the number of qubits
      # Returns: initial Hamiltonian consisting of sigma_x local fields.
8
9
      si = qeye(2)
10
      sx = sigmax()
11
12
      sx_list = []
13
14
      for n in range(N):
15
          op_list = []
16
       for m in range(N):
17
```

```
op_list.append(si)
18
19
           si_n = tensor(op_list)
20
           op_list[n] = sx
21
           sx_list.append(tensor(op_list))
22
23
      # construct the hamiltonian
24
      Hb = 0
25
26
      for n in range(N):
27
          Hb += 1/2*(si_n- sx_list[n])
28
29
      return Hb
30
31
32 # Interpolating functions of the adiabatic Hamiltonian
33
34 def Hp_coeff(t, args):
      tau = args['tau']# time scale of the adiabatic evolution
35
      exp = args['exp'] # in case we want a nonlinear interpolation
36
37
      return (t/tau)**exp
38
39 def Hb_coeff(t,args):
      tau = args['tau'] # time scale of the adiabatic evolution
40
      exp = args['exp'] # in case we want a nonlinear interpolation
41
      return 1-(t/tau)**exp
42
43
44 # Performing the adiabatic evolution
45 def adiabatic_evolution(H, N, args, psi0, tlist):
      # H: Hamiltonian in the format H = [[Hb, Hb_coeff],[Hp, Hp_coeff]]
46
      # args: arguments of the time dependent function of the hamiltonian, args = {'
47
      exp': exp, 'tau': tau}
      # psi0: initial state to be evolved
48
49
      # tlist: list of times for which the evolution is calculated
      # returns: list of expected values of sigma_z for each qubit, and list of
50
      evolved states at each time in tlist
51
      c_op_list = [] # we are not using any colapse operators
53
      si = qeye(2)
54
      sz = sigmaz()
55
      sz_list = []
56
57
      for n in range(N):
58
           op_list = []
59
60
           for m in range(N):
               op_list.append(si)
61
           op_list[n] = sz
62
           sz_list.append(tensor(op_list))
63
64
      options = Options(nsteps=100000, store_states = True)
65
66
      result = mesolve(H, psi0, tlist, c_op_list, sz_list, args = args, options =
67
      options)
68
      return result.expect, result.states
69
70
```

```
71
72 # ---- Example of how to run the evolution ----
73 N = 4 # number of qubits
74
75 Hp = qubit_hamiltonian_to_qutip(qubit_hamiltonian_JW, N) # final Hamiltonian
76 Hb = generate_Hb(N)
77
78 tau = 75 # time scale of the adiabatic evolution
79 exp = 1 # linear path
80 args = {'tau': tau, 'exp': exp}
81
ground = Hb.groundstate(sparse=False, tol=0, maxiter=100000)
83 psi0 = ground[1] # initial state is the ground state of Hb
84
85 tlist = np.linspace(0, tau, 2*tau)
86
87 # adiabatic Hamiltonian
88 H = [[Hb, Hb_coeff], [Hp, Hp_coeff]]
89 expect, evolved_state = adiabatic_evolution(H, N, args, psi0, tlist)
```

Functions to calculate the entanglement.

```
1 import numpy as np
2 from qutip import *
5 def entanglement_entropy(state, partition):
6
      # state - state for which we want to calculate the entanglement entropy
      # partition - integer or array of integers from 0 to N-1 that indicate the
7
      subsystem with respect to which we calculate the partial trace
      # Returns: the Von Neumann entropy of the partial trace
8
9
10
     dm = state*state.dag() # density matrix
11
      partial_dm = dm.ptrace(partition) # partial trace
      return entropy_vn(partial_dm, base = 2) # Von Neumann entropy
12
13
14 def max_entanglement(state, partitions):
      # partitions: partitions of the system considered, in our case partitions =
15
      [[0,1],[0,2],[0,3]]
      # returns: maximized entanglement entropy over partitions
16
      entanglements = []
17
      for partition in partitions:
18
          entanglements.append(entanglement_entropy(state,partition))
19
     return np.max(entanglements)
20
```

Miscellaneous.

```
1 from qutip import *
2
3 # generating the basis elements as qutip quantum objects
4 def generate_basis_element(v, N):
5 # N: number of qubits
6 # v: list of length N with zeros and ones
7 # returns: list v converted into a basis state
8
9 lists = []
10 for i in range(0,N):
11 if v[i]==0:
```

```
lists.append(basis(2,0))
12
13
          if v[i]==1:
               lists.append(basis(2,1))
14
      return tensor(lists)
15
16
17 def generate_basis(N):
      # N: number of qubits
18
      \# returns: list with the 2^N computational basis elements, list with labels
19
      ready to be used by matplotlib.
20
      a = [0, 1]
21
      basis = []
22
      basis_label = []
23
      # Here we are assuming N=4 \,
24
25
      for i in a:
          for j in a:
26
               for k in a:
27
                   for t in a:
28
                       basis.append(generate_basis_element([i,j,k,t],N))
29
                       basis_label.append(r'$|%u, %u, %u, %u \rangle$'%(i,j,k,t))
30
      return basis, basis_label
31
32
33 energies = H.eigenenergies(sparse=False, sort='low', eigvals=2, tol=0, maxiter
      =100000) # calculates the two lowest energies of H
34 eigensates = H.eigenstates(sparse=False, sort='low', eigvals=2, tol=0, maxiter
      =100000) # calculates the two lowest energy eigenstates of {\rm H}
35 overlap_square = abs(eigenstates[1][0].overlap(state))**2 # square overlap of state
    with the lowest energy eigenstate in eigenstates
```