



Università degli Studi di Torino
Scuola di Scienze della Natura
Corso di Laurea Magistrale in Fisica dei Sistemi Complessi
a.a. 2021/2022

Tesi di Laurea Magistrale

Quantum computational chemistry: a promising application for quantum computers

Relatore

Prof. Alberto Carlini

Correlatore

Dott. Alberto Acuto

Controrelatore

Prof. Leonardo Castellani

Candidato

Davide Bincoletto

26 October 2022

A chi c'è stato, a chi c'è e a chi ci sarà

To who has been there, who is here and who will be there

Contents

Introduction	4
1 Quantum computational chemistry	6
1.1 Motivation	6
1.2 Classical approach	11
1.2.1 Basis set	11
1.2.2 Wave function-based methods	12
1.2.3 Accuracy and precision	17
1.3 Quantum approach	18
2 Density functional theory	22
2.1 Introduction	22
2.2 First approach	22
2.3 Analysis	23
2.3.1 Hohenberg-Kohn theorems	23
2.3.2 Kohn-Sham self-consistent field method	24
2.3.3 Exchange-correlation functionals	27
2.4 Computational requirements	32
3 Quantum computing	36
3.1 Introduction	36
3.1.1 Qubits	39
3.1.2 Quantum gates	41
3.1.3 Measurement	45
3.2 Important results	45
3.2.1 Set of universal quantum gates	45
3.2.2 Gottesman-Knill theorem	47
3.2.3 Quantum computational complexity	48
3.2.4 Quantum Fourier transform	49
3.3 Quantum devices	50
3.3.1 Optical photons	52
3.3.2 Trapped ions	53
3.3.3 Superconducting circuits	54
4 Quantum computing for computational chemistry: FTQC devices	58
4.1 Fault-tolerant quantum computation devices	58
4.2 Quantum phase estimation	61

4.2.1	State preparation	62
4.2.2	Hamiltonian simulation	63
4.3	Computational requirements	64
4.3.1	Examples	66
5	Quantum computing for computational chemistry: NISQ devices	71
5.1	Noisy intermediate-scale quantum devices	71
5.2	Variational quantum eigensolver	72
5.2.1	Encoding of operators	74
5.2.2	Ansatz and state preparation	77
5.2.3	Efficient grouping and measuring strategies	79
5.2.4	Error mitigation	81
5.3	Computational requirements	84
5.3.1	Example	88
6	Computational advantage	92
6.1	Benchmarking quantum computers	92
6.1.1	Number of qubits	93
6.1.2	Quantum volume	93
6.1.3	Circuit Layer Operations per Second Benchmark	94
6.1.4	Today's quantum computers	95
6.2	Algorithms comparison	96
6.2.1	QPE and VQE	96
6.2.2	Classical and quantum	99
6.3	Targets of simulation	102
6.3.1	Defining advantage	102
6.3.2	Applicability forecast	103
6.4	Next steps	110
6.4.1	Problem fragmentation	110
6.4.2	Materials science	111
Conclusion		113
Acknowledgments		114
Bibliography		115

Introduction

Computer simulation is a fundamental part of most scientific research. It allows to predict the behaviour of a real-world system by creating a mathematical model on a computer and then analyzing the obtained results, without the need to perform a real experiment until one wants to compare the accuracy of the model used. Here we want to analyze the simulation of quantum systems, i.e. physical systems that follow the laws of quantum mechanics, also referred to as 'quantum simulation' or 'computational chemistry'.

From the birth of the theory at the beginning of the twentieth century many models have been proposed, and following the terminology of computer science we also refer to them as methods or algorithms. The main problem these algorithms aim to solve is to find the energy of the ground state of the system, this lets us know the most stable configuration of the system, thus the most probable configuration in which we find it in a real experiment.

This field of research intersects the investigation of new technological devices on which one can perform such simulation. As a matter of fact, in the last forty years there has been a great development of the so-called quantum computers, a type of calculators that exploit the laws of quantum mechanics to perform the computations. Notoriously, physicist Richard Feynman explained this possibility in a 1982 speech [15], where he referred explicitly to "an exact simulation, that the computer will do exactly the same as nature", where the system under study and the calculator, i.e. the system through which we perform the computation, share the same properties and are in a one-to-one correspondence.

This investigation also translates in engineering problems, in effect quantum systems are difficult to manipulate, they need to be isolated and put in specific conditions to work properly, e.g. superconducting circuits need near absolute zero temperatures. For this reason, only in recent years quantum computers with very few components have been produced and are available for use.

Through this thesis we explore the main possibilities to run a quantum simulation. We analyze the methods developed for classical computers and the ones for quantum computers doing a comparison with regard to computational requirements and accuracy of the calculations.

In Chapter 1 we explore the importance of quantum simulation and the scheme to implement one. We then describe the first historical methods used to perform a simulation on classical computers and the birth of the research for methods on quantum computers, also called 'quantum computational chemistry'.

In Chapter 2 we describe density functional theory, the main algorithm used today to perform a quantum simulation on a classical computer.

Then we focus on quantum computers, in Chapter 3 we show the characteristics of this kind of devices and the new potential they bring. We then concentrate on fault-tolerant quantum computing (FTQC) devices, and the methods designed for them, in Chapter 4. These are theoretical universal computers which take full advantage of the potential of quantum computation. After this, in Chapter 5 we characterize noisy intermediate-scale quantum (NISQ) devices and the correspondent methods, these are already available computers with a little number of low quality components, called 'qubits', to perform calculations.

Finally in Chapter 6 we describe the features of the already available quantum computers from a technological perspective, we do a comparison between the different algorithms and present a forecast of the targets of simulation, within chemistry and materials science, that can show an advantage and their computational cost.

Chapter 1

Quantum computational chemistry

1.1 Motivation

Simulating an atomic system is quite a difficult task. In order to understand the full structure and the dynamics of such system, in the non-relativistic approximation, one has to solve the Schrödinger equation

$$i\hbar \frac{\partial |\Psi\rangle}{\partial t} = H|\Psi\rangle, \quad (1.1)$$

which is also referred to as the time-dependent Schrödinger equation (TDSE). If the Hamiltonian H does not depend on time t the equation can be written as

$$H|\Psi\rangle = E|\Psi\rangle, \quad (1.2)$$

which is referred to as the time-independent Schrödinger equation (TISE). This eigenvalue problem leads to find the eigenvectors $|\Psi\rangle$ and the eigenvalues E of the Hamiltonian H , which represent the wave function and the energy levels of the system.

In this approximation we can also describe the time evolution of the wave function with a unitary operator U in the form of

$$U(t_0, t)|\Psi(t_0)\rangle = e^{-\frac{i}{\hbar}H(t-t_0)}|\Psi(t_0)\rangle = |\Psi(t)\rangle. \quad (1.3)$$

This way each eigenvector evolves in time through the action of the operator corresponding to its own eigenvalue.

Unfortunately the TISE can be exactly solved only for the hydrogen atom. The solution leads to the wave function states, also called atomic orbitals, shown in Figure 1.1; they are composed of Laguerre polynomials, for the radial variable, and spherical harmonic functions, for the polar and azimuthal variables. This result can also be synthesized by using three quantum numbers (n, l, m) , called respectively principal, azimuthal and magnetic quantum numbers, to account for the wave functions and the corresponding energy levels. This way the ground state of such atom can be represented by the triplet $(1, 0, 0)$ or by the symbolic writing $1s$.

For bigger systems the complexity grows, the terms in the Hamiltonian operator

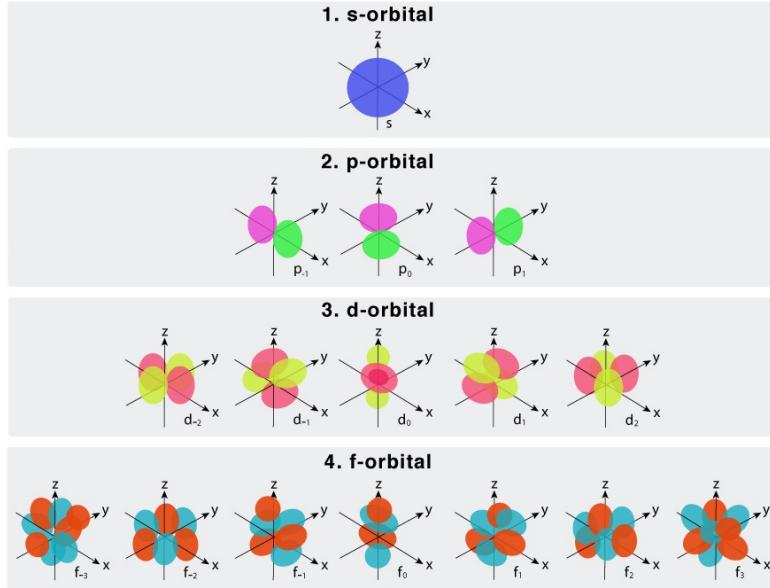


Figure 1.1: Exact atomic orbitals for the hydrogen (H) atom.

adds up and new terms of interaction appears. The latter, especially, grows combinatorially with the number of electrons and nuclei.

The Hamiltonian for a generic multi-electron system can be written as

$$H = - \sum_i \frac{\hbar^2}{2m_e} \nabla_i^2 - \sum_I \frac{\hbar^2}{2M_I} \nabla_I^2 - \sum_{i,I} \frac{e^2}{4\pi\epsilon_0 r_{iI}} Z_I \\ + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{4\pi\epsilon_0 r_{ij}} \frac{1}{r_{ij}} + \frac{1}{2} \sum_{I \neq J} \frac{e^2}{4\pi\epsilon_0} \frac{Z_I Z_J}{r_{IJ}}, \quad (1.4)$$

with i, j and I, J indicating respectively the electrons and the nuclei.

No analytical solution exists for this type of equations. One can use approximations to solve easier versions of these problems, but the amount of calculation needed grows rapidly. This is why, with the advent of digital computers, in the 1950s, a new field of research was born: computational chemistry.

The aim is to simulate chemical compounds and materials that are difficult to study from a theoretical point of view, which, as we have seen, are the vast majority, but also to study those systems that are not available to be studied in laboratory or those that need specific and difficult to obtain experimental conditions. Another important use is to employ simulation as a complement to experiments, both to help explain and interpret certain phenomena, for example why a material show a certain property in a certain condition, and to make predictions, that is, one can compare the result of the simulation with the result of the experiment to show if the predictive model used is correct [45, 13].

From this argument, a simple scheme that one can follow to simulate an atomic system is:

1. Compute the solution for the TDSE or the TISE, with the use of

approximations

A typical approximation is the Born-Oppenheimer approximation, i.e. considering the large difference between the electron mass and the masses of atomic nuclei and correspondingly the time scales of their motion, we can express the total wave function of the system as the product of an electronic wave function and a nuclear wave function, the latter is sometimes called vibrational or rotational.

$$|\Psi_{total}\rangle = |\Psi_{electronic}\rangle |\Psi_{nuclear}\rangle. \quad (1.5)$$

This enables a separation of the Hamiltonian operator into electronic and nuclear terms. In the electronic Hamiltonian the electron–nucleus interactions are simplified, but not removed, i.e. the nuclei are considered stationary and the electrons "feel" the Coulomb interaction. We also use natural units to simplify the numeric constants,

$$H_e = -\frac{1}{2} \sum_i \nabla_i^2 - \sum_{i,I} \frac{Z_I}{r_{iI}} + \frac{1}{2} \sum_{i \neq j} \frac{1}{r_{ij}}, \quad (1.6)$$

thus giving us the electronic TISE,

$$H_e |\Psi_e\rangle = E_e |\Psi_e\rangle. \quad (1.7)$$

By numerically solving the equation (1.7) iteratively, changing the position of the nuclei \mathbf{R} at every step, we obtain the electronic energy E_e as a function of \mathbf{R} , this is the potential energy surface (PES). This is often referred to as the 'electronic structure problem'.

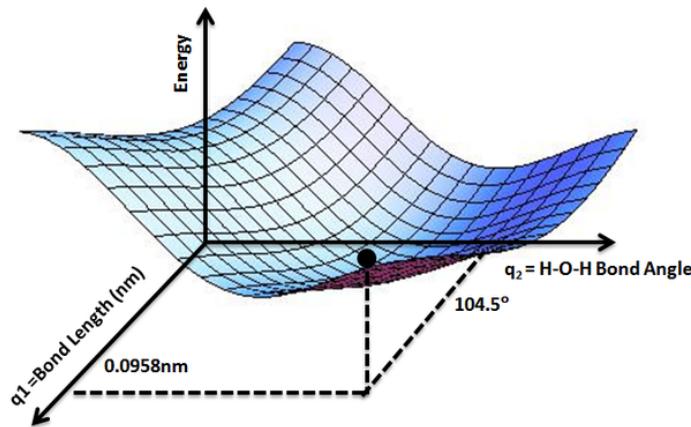


Figure 1.2: Potential energy surface for the water (H_2O) molecule.

In principle we can derive the molecular dynamics by computing the unitary evolution of the system, this is referred to as 'ab initio molecular dynamics' (AIMD), since it relies only on fundamental laws of quantum mechanics. In order to obtain the dynamics we need to solve the nuclear TISE, compute the excited states, also called virtual states, of the compounds and the corresponding energy levels. This is because interactions happen between wave functions

that are not localized in space, thus multiple eigenstates have to be considered at the same time, this is also called superposition.

In this thesis we will not focus on the problem of molecular dynamics and solutions for the TDSE, but only on the equilibrium phase and the TISE, although it is still a wide and open field of research and there are many methods to solve it, both using digital computers and quantum computers (Ehrenfest method, Born-Oppenheimer molecular dynamics, Car-Parrinello method, quantum phase estimation, Zalka method, to name a few [37]).

2. From the eigenvectors and the eigenvalues of the TDSE and the TISE compute the chemical properties of the systems

At the heart of any chemical process there is its mechanism, the elucidation of which requires the identification of all relevant stable intermediates and transition states and the calculation of their properties. The latter allow for experimental verification, whereas energies associated with these structures provide access to thermodynamic stability and kinetic accessibility. In general, a multitude of charge and spin states need to be explicitly calculated in search for the relevant ones that make the whole chemical process viable.

First of all we can study a single compound. From the PES we can find the configurations that lead to the stable structures, i.e. the minima of the multi-dimensional surface, this procedure is also called 'geometry optimization'. We can also find the optical properties, i.e. study the interaction of the compound with electromagnetic fields.

Second, we can model chemical reactions. We can find the reaction rates and the transition states, i.e. reaction intermediates, which cannot be observed in a laboratory experiment, thus reconstructing the entire path of the reaction. Moving on to materials, simulation allow us to study thermodynamic and magnetic properties of complex systems, such as cold atomic gases, spin glasses and the Fermi-Hubbard model.

3. Apply these methods to bigger and more complex systems

With simulation we can study both long chains of reactions, e.g. photosynthesis mechanism, which involve many interactions, and individual systems with a big number of highly-correlated particles, e.g. superconducting materials and catalysts.

Some examples of applications are [7]:

- **The chemistry of enzyme active sites**

These can involve multiple coupled transition metals, which present some of the most complicated quantum chemistry problems in the biological world. Combined theoretical and experimental studies have proven successful in unravelling many structural and electronic features of these systems, however, a detailed understanding of the interplay between spin-coupling and delocalization between metals, which requires quantum correlation to be considered and is needed to interpret aspects of experimental spectroscopy, remains in its infancy;

- **Transition metal nanocatalysts and surface catalysts**

Similarly to enzyme active sites, simulating the action mechanism of synthetic heterogeneous catalysts remains a major challenge. While density functional theory has been widely employed, predictions of even basic quantities such as the adsorption energy of small molecules are unreliable. While not all such problems are expected to be multireference in character, i.e. quantum correlation is fundamental to describe them, even the single-reference modeling of such chemistry, at a level significantly beyond density functional theory, is currently challenging or impossible. In addition, multireference effects are expected to play a role in certain catalysts, such as transition metal oxides, or at intermediate geometries in reaction pathways;

- **Light harvesting and the vision process**

The photochemistry of conjugated organic molecules is the means by which nature interacts with light. The quantum chemical questions revolve around the potential energy surfaces of the ground and excited states, and the influence of the environment on the spectrum. These questions are currently challenging due to the size of the systems involved as well as the varying degree of single- and multireference character in many of the conjugated excited states;

- **Quantum Molecular Spectroscopy**

The theoretical goal is to compute the eigenstates of the nuclear Schrödinger equation. Such spectroscopy is important not only for the fundamental understanding of small molecules and the quantum control of atomic and molecular states, but also to provide insight into the basic chemical processes and species involved, for example, in atmospheric chemistry and in astrochemistry;

- **Chemical Quantum Dynamics**

Classical algorithms for simulation of quantum dynamics generally invoke methods based on classical equations of motion, that scale better to large systems, but are difficult to systematically improve. Proton coupled electron transfer (PCET) is known to be an important mechanism in catalysis and energy storage: electrons are transferred at lower overpotentials when thermodynamically coupled to proton transfer.

Vibrational dynamics is also important for characterization of complex environments;

- **Characterization of high-temperature superconductivity**

There is some overlap in methods and ideas between the materials electronic structure problem and the quantum chemistry one. While the general properties of the superconducting phase itself are relatively well characterized, the mechanism driving superconductivity is not yet fully elucidated. Also, the relation between two nearby regimes, the pseudogap and strange metal phase do not have a theoretical explanation;

- **Materials quantum dynamics**

The main workhorse of mesoscopic quantum physics is electron transport, i.e. the response of the system when it is coupled to electron reservoirs

and a voltage is applied. Going beyond spectral properties, the nonequilibrium real-time dynamics of quantum systems has increasingly come into focus, both because of experiments that can probe quantum dynamics at atomic scales and because of fundamental interest in studying the equilibration of quantum systems, which serves as a bridge between the theories of quantum mechanics and statistical mechanics.

This knowledge can then be used to understand new natural phenomena, for which we still have no theoretical explanation, as well as to search and design new industrially relevant compounds to improve development areas, like pharmaceuticals [3], [8], batteries [23], nitrogen fixation [41], collection of solar energy and so on.

Throughout the thesis we will focus especially on step 1, i.e. retrieving eigenvectors and eigenvalues solutions for the TISE, but it is important to remember the scheme and the goal we have in mind when we want to simulate a quantum system, being it academic research or industrial development.

1.2 Classical approach

In order to simulate a system on a digital computer there are many algorithms one can use and here we will review the most important ones.

We divide them into two groups: the wave function-based, i.e. the ones that aim to find the wave function of the system in order to compute its properties, and the density function-based, i.e. the ones that aim to find the density function. Here we will explain shortly the former and in Chapter 2 we will focus on the latter.

1.2.1 Basis set

First of all, we can carry out a simulation in the first or second quantization scheme, i.e. the antisymmetry of the fermionic wave function is retained in the sign of the wave function itself under transformations or in the algebra of the creation and destruction operators (a^\dagger, a) applied to the wave function, and then we can choose between grid-based and basis set methods. Here we will only cover the basis set methods.

The basis set is a group of functions used to approximate the atomic orbitals (also called spin-orbitals, they are a combination of three spatial coordinates and one spin coordinate) and subsequently the molecular orbitals.

There are many types of functions and many ways to create the sets; the main idea is to have accurate functions, i.e. they take values close to the real orbitals, and small, easily-computable sets. The main examples of functions are: Slater-type orbitals (STO) and Gaussian-type orbitals (GTO), which resemble hydrogen-like wave function states, and plane waves, which resemble the free-particle wave function and are mostly used to model particles moving in lattices.

$$\phi_{a,b,c}^{GTO}(x, y, z) = Nx^a y^b z^c e^{-\zeta r^2} \quad (1.8)$$

$$\phi_{\nu}^{plane \ waves}(r) = \sqrt{\frac{1}{V}} e^{\frac{2\pi i \nu r}{L}} \quad (1.9)$$

The main examples of sets are: minimal set, i.e. one orbital for each electron, split-valence set, i.e. a bigger number of orbitals for the valence electrons, and correlation-consistent polarized valence set, i.e. use also correlated wave functions and add functions to each electron depending on a parameter.

Some common acronyms used to indicate basis sets are:

- STO-3G: it is a minimal basis set which contracts 3 Gaussian functions to approximate the more accurate (but more difficult to compute) Slater type orbitals;
- 6-31G*: it is a split-valence basis set where the core orbital is described by a contraction of 6 Gaussian orbitals, while the valence is described by two orbitals, one made of a contraction of 3 Gaussians, and one a single Gaussian function. The star (*) indicates polarization functions on non-hydrogen atoms;
- cc-PVTZ: it is a Dunning's correlation consistent polarized valence triple- ζ basis set, where ζ indicates the exponent in the Gaussian functions. Together with cc-PVQZ and cc-PV5Z (quadruple and quintuple- ζ) is considered the basis set which leads to the highest accuracy.

1.2.2 Wave function-based methods

Historically the first method comes from the first attempts to solve the TISE, even without the use of computers, this is the Hartree-Fock (HF) method, which dates back to the 1930s. The idea is to simplify the Hamiltonian operator in eq. (1.6) by writing the interaction potential between electrons as a mean-field potential, thus the electrons are independent from one another, they are only subject to an external potential and we can solve the TISE for the system. The mean-field potential v^{HF} can be computed iteratively and the algorithm scheme is:

1. Build a guess function for the orbitals

Either choose functions from the basis set or from the solution of the non-interacting system TISE;

2. Compute the external potential for each electron

That is, integrate over the entire space obtaining the Coulomb operator (J) and the exchange operator (K) and sum over all the orbitals;

3. Solve the TISE diagonalizing the Hamiltonian operator

In this context the Hamiltonian is called 'Fock operator' f . With this process we obtain the eigenvalues and the eigenvectors, i.e. the energy values and the orbitals;

4. With the new guess compute the external potential and repeat until convergence.

For this reason the method is also called self-consistent field (SCF) method. The scheme is depicted in Figure 1.3.

The TISE reduces to the HF equation:

$$f(\mathbf{x}_i)|\chi(\mathbf{x}_i)\rangle = \epsilon_i|\chi(\mathbf{x}_i)\rangle, \quad (1.10)$$

where

$$f(\mathbf{x}_i) = -\frac{1}{2}\nabla_i^2 - \sum_I \frac{Z_I}{r_{ii}} + v^{HF}, \quad (1.11)$$

and $\chi(\mathbf{x}_i)$ represents a spin-orbital, consisting of a spatial and a spin function,

$$|\chi(\mathbf{x}_i)\rangle = |\psi(\mathbf{r}_i)\rangle|\varsigma(s_i)\rangle. \quad (1.12)$$

A formal way to obtain this equation is to use a variational principle, i.e. we build a Lagrange functional and minimize with respect to the orbital functions.

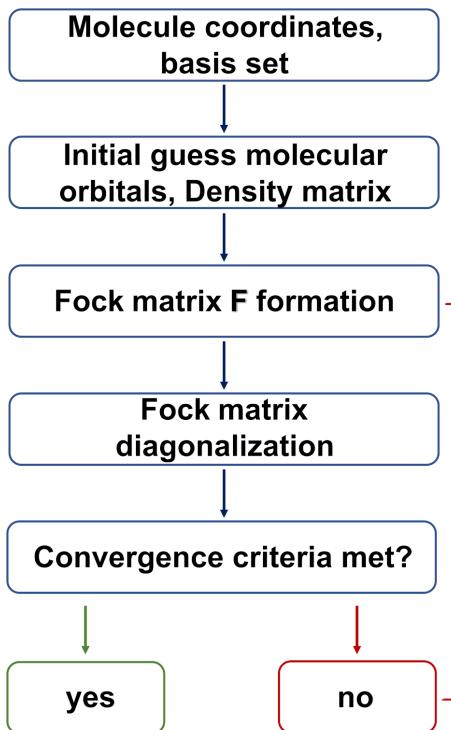


Figure 1.3: Flowchart for the HF SCF method.

To write the algorithm efficiently on a computer we can use a matrix formulation. We consider the spatial part of the orbitals as a combination of functions from the basis set $\{|\phi_\nu\rangle\}$,

$$|\psi(\mathbf{r}_i)\rangle = \sum_{\nu=1}^K C_{\nu i} |\phi_\nu(\mathbf{r}_i)\rangle, \quad (1.13)$$

where $C_{\nu i}$ are real coefficients.

Then we can substitute the formula in the HF equation, multiply by $\langle\phi_\mu|\$, integrate over all space and define two integrals,

$$S_{\mu\nu} = \int d\mathbf{r}_i \langle\phi_\mu(\mathbf{r}_i)|\phi_\nu(\mathbf{r}_i)\rangle, \quad (1.14)$$

called the 'overlap matrix' and

$$F_{\mu\nu} = \int d\mathbf{r}_i \langle \phi_\mu(\mathbf{r}_i) | f(\mathbf{r}_i) | \phi_\nu(\mathbf{r}_i) \rangle, \quad (1.15)$$

called the 'Fock matrix'.

The HF equation can be written as

$$FC = SC\epsilon, \quad (1.16)$$

which is called the 'Roothann equation' [42]. Solving the equation means find the square $K \times K$ matrix C that minimizes the system's total energy.

This result leads us to write the wave function of our system.

Generally, having N electrons and $m > N$ spin-orbitals, we can construct

$\binom{m}{N} = \frac{m!}{N!(m-N)!}$ Slater determinants, where a Slater determinant is the antisymmetric combination of the orbitals

$$|\Psi_{HF}(\mathbf{x}_0, \dots, \mathbf{x}_{N-1})\rangle = \frac{1}{\sqrt{N!}} \begin{vmatrix} |\chi_0(\mathbf{x}_0)\rangle & |\chi_1(\mathbf{x}_0)\rangle & \dots & |\chi_{m-1}(\mathbf{x}_0)\rangle \\ |\chi_0(\mathbf{x}_1)\rangle & |\chi_1(\mathbf{x}_1)\rangle & \dots & |\chi_{m-1}(\mathbf{x}_1)\rangle \\ \vdots & \vdots & \ddots & \vdots \\ |\chi_0(\mathbf{x}_{N-1})\rangle & |\chi_1(\mathbf{x}_{N-1})\rangle & \dots & |\chi_{m-1}(\mathbf{x}_{N-1})\rangle \end{vmatrix}. \quad (1.17)$$

We must combine the spin-orbitals this way in order to be consistent with the Pauli exclusion principle, which states that two fermions can not occupy the same quantum state, i.e. the same orbital. Thus, the probability of that kind of system to be measured must be zero and the wave function must be antisymmetric with respect to particle exchange.

However, the Hartree-Fock approach considers only one Slater determinant, called the Hartree-Fock ground state determinant, this is because in the algorithm we only consider the occupied orbitals in order to compute the potential v^{HF} . The other determinants represent the excited states of the electronic system.

Algorithms like this can be analyzed with respect to their computational cost or requirements. We will define it more precisely in Chapter 4, but for now we refer to it as the resources needed to run an algorithm.

The computational cost of the Hartree-Fock method scales as $O(m^4)$, due to the calculation of the one- and two-electrons integrals for the mean-field potential, where m is the number of basis functions.

This is the lowest cost we will consider in this thesis.

The fact that we consider only a single Slater determinant is fundamental, because the HF method, as a mean-field approach, is a strong approximation, which can not give us the exact solution for the system, especially for many-electrons systems where the interaction potential is not negligible.

First, electrons in this model do not instantaneously interact with each other, but

rather each and every electron interacts with the average field. In reality their movements take into account the positions of all other electrons, in order to avoid being in the same quantum state, i.e. locations in a close proximity. This effect is called 'dynamic correlation' since it is directly related to electron dynamics. This correlation accounts for the short-range structure of the wave function and weak long-range interactions such as London dispersion forces.

Secondly, in certain cases an electronic state can be well described only by a linear combination of more than one Slater determinant, this is obvious since, as we said before, wave functions are not localized and electrons can be found in different locations, i.e. different orbitals. This is called 'static correlation' and is associated with the strong mixing of degenerate or near-degenerate determinants.

More properly we say that the total energy of the system is

$$E_{\text{exact}} = E_{HF} + E_{\text{correlation}}, \quad (1.18)$$

so we still need to take into account the correlation term, which derives from the electrons interaction.

This is why in the decades following the definition of the HF method, scientists came up with new methods to tackle these problems. These go under the name of post-HF methods.

The most intuitive one is the 'configuration interaction' (CI) method. 'Configuration' means that we write the wave function as a linear combination of Slater determinants and 'interaction' means that we mix different electronic states in order to account for electronic correlation. The first term in the expansion is normally the Hartree–Fock determinant and the subsequent ones are the states in which one spin-orbital at a time is swapped with a virtual orbital. If only one spin-orbital differs, we describe this as a single excitation determinant (CIS). If two spin-orbitals differ it is a double excitation determinant (CID) and so on. This is used to limit the number of determinants in the expansion which is called the CI-space. If the expansion includes all possible state functions it is called a 'full configuration interaction' (FCI) method,

$$|\Psi_{FCI}\rangle = (I + \sum_{i,\alpha} C_{i\alpha} a_i^\dagger a_\alpha + \sum_{i>j,\alpha>\beta} C_{ij\alpha\beta} a_i^\dagger a_j^\dagger a_\alpha a_\beta + \dots) |\Psi_{HF}\rangle, \quad (1.19)$$

where a_i^\dagger and a_α are the creation and destruction operators used to swap the spin-orbitals, i.e., in the second quantization scheme, destroy a fermion in the occupied state α and create one in the virtual state i . C are parameters to be optimized according to the Rayleigh-Ritz variational principle, which states that the energy expectation value of a parametrized wave function is greater than or equal to the lowest energy eigenvalue of the Hamiltonian being measured.

The FCI exactly solves the electronic TISE, because it takes into account all the Slater determinants [12]. This also means that the computational cost to execute this algorithm grows combinatorially, because for each new orbital we have to consider all the possible ways the electrons occupy the orbitals. Thus the computational cost of the FCI method is $O(m!)$, which is the highest scaling we can consider.

All the others computational methods aim to find the same results while being less expensive in the scaling of the resources, this leads to a trade-off between accuracy and computational requirements. As a standard to frame these methods we will refer to the 'chemical accuracy', which is defined as an error of 1 kcal/mol, or equivalently 4 kJ/mol or 1 mHa, with respect to the hypothetical exact energy or an error present in an experimental measurement to find thermodynamic properties. This is the target for every simulation to be comparable with real data.

An important post-HF method is the Møller–Plesset (MP) perturbation theory. The idea is to take advantage of Rayleigh–Schrödinger perturbation theory. We write the Hamiltonian as

$$H = H_0 + \lambda V, \quad (1.20)$$

with

$$H_0 = \sum_i f(\mathbf{x}_i). \quad (1.21)$$

We can expand the ground state eigenfunction and eigenvalue as a Taylor series in λ :

$$|\Psi_0\rangle = |\Psi_0^{(0)}\rangle + \lambda|\Psi_0^{(1)}\rangle + \lambda^2|\Psi_0^{(2)}\rangle + \dots, \quad (1.22)$$

$$E_0 = E_0^{(0)} + E_0^{(1)} + E_0^{(2)} + \dots, \quad (1.23)$$

where $|\Psi_0^{(0)}\rangle = |\Psi_{HF}\rangle$ and $E_0^{(0)} = E_{HF}$.

By imposing the normalization of the wave function and by solving the equation (1.22) iteratively for each power of the expansion, we obtain the corrections for the ground state eigenfunction and eigenvalue. For example

$$E_0^{(1)} = \langle \Psi_0^{(0)} | V | \Psi_0^{(0)} \rangle, \quad (1.24)$$

$$E_0^{(2)} = \sum_{j>0} \frac{|\langle \Psi_j^{(0)} | V | \Psi_0^{(0)} \rangle|^2}{E_0^{(0)} - E_j^{(0)}} \quad (1.25)$$

and so on.

One usually truncates the expansion due to its computational cost and the resulting methods are called MP2, MP3, etc., where the number indicate the order of the expansion.

The last method we analyze is the 'coupled cluster' (CC) method. Just like the CI method, the goal is to add terms to the Hartree-Fock wave function to account for correlation.

The expansion is written as

$$|\Psi_{CC}\rangle = \prod_{i,\alpha} (I + C_{i\alpha} a_i^\dagger a_\alpha) \prod_{i>j, \alpha>\beta} (I + C_{ij\alpha\beta} a_i^\dagger a_j^\dagger a_\alpha a_\beta) \dots |\Psi_{HF}\rangle \quad (1.26)$$

and it can be reformulated as

$$|\Psi_{CC}\rangle = e^T |\Psi_{HF}\rangle, \quad (1.27)$$

where

$$T = T_1 + T_2 + \dots = \sum_{i,\alpha} t_{i\alpha} a_i^\dagger a_\alpha + \frac{1}{4} \sum_{i>j,\alpha>\beta} t_{ij\alpha\beta} a_i^\dagger a_j^\dagger a_\alpha a_\beta + \dots \quad (1.28)$$

Again we swap spin-orbitals to build more Slater determinants other than than the HF one. α and β indicate occupied orbitals, i and j indicate virtual orbitals and t indicates excitation amplitudes.

When all of the excitation operators T_i are included, the CC method recovers the FCI wave function and again this calculation is very costly, so the method is normally truncated to a lower excitation level, often single (CCS) and double excitations (CCSD).

Because of its product parametrization, the CC method generates a trial wave function that includes all possible powers of T_i operators, this way it provides faster convergence than the CI method.

The CC method is often referred to as the "gold standard" for chemistry simulation. The CCSD(T) method, where the T inside the brackets means it is calculated only perturbatively, is known to treat dynamic correlation accurately, but is generally inapplicable to molecular problems dominated by non-dynamic correlation [14].

All these methods can be analyzed and compared with respect to the total scaling, i.e. the computational cost. The result we have is [19]

$$\begin{array}{ccccccccc} HF \approx DFT & < & MP2 & < & CISD \approx CCSD & < & MP4 \approx CCSD(T) & << & FCI \\ O(m^{3-4}) & < & O(m^5) & < & O(m^6) & < & O(m^7) & << & O(m!) \end{array} \quad (1.29)$$

where DFT means density functional theory, which is described in Chapter 2.

At first glance we can easily notice that the cost scales up very quickly even though the methods are inaccurate, since they approximate the FCI wave function. This is one of the reasons that makes us look towards a full quantum approach.

1.2.3 Accuracy and precision

The most useful measure of accuracy for applications involving chemical reactions is chemical accuracy, defined before. This is a desirable target because calculations capable of achieving it would rival the accuracy of measurements attainable in a chemical laboratory.

However, it is important to distinguish accuracy (computational error with respect to an experimental measurement) from precision (computational error with respect to a computational reference, for example, a sufficiently accurate result obtained with a large basis set). A simplified representation is shown in Figure 1.4.

Several studies described experiments that reached chemical accuracy, but in reality they reached chemical precision: an error of at most 1 kcal/mol compared to the exact solution typically provided by the combination of the FCI method and a very small basis set that was used as a reference.

This is an important distinction because an answer computed with a 1 kcal/mol precision may be useless for explaining or predicting chemical reactivity if the level

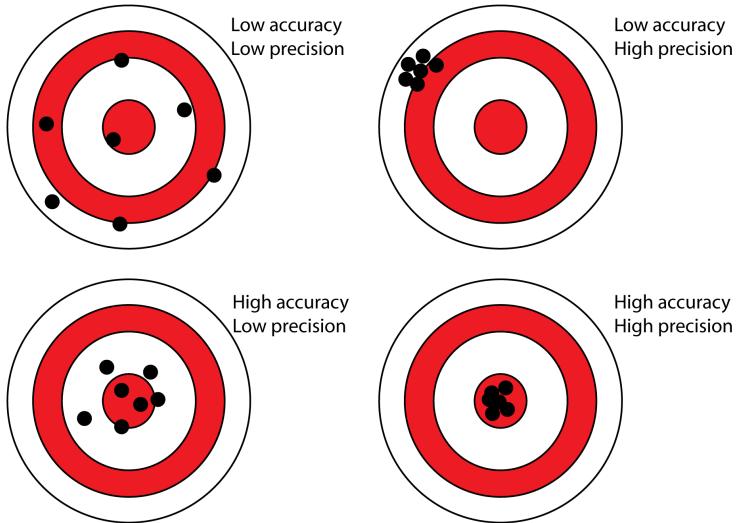


Figure 1.4: Accuracy vs precision.

of theory used as a reference does not allow a similar level of accuracy. Matching an FCI energy value is not necessarily a practically useful achievement, it means attaining a good precision but not a high accuracy.

Configuration interaction (including FCI) energies obtained with Gaussian basis sets converge to the "exact" energy very slowly. Therefore one needs Gaussian basis sets of at least quadruple- ζ quality (where ζ is the number of contracted functions per atomic orbital) to achieve chemical accuracy with pure, non-extrapolated FCI calculations.

1.3 Quantum approach

As mentioned in the introduction, in 1982 Richard Feynman gave a speech titled "Simulating Physics with Computers" at the California Institute of Technology, where he stated [15]:

"And I'm not happy with all the analyzes that go with just the classical theory, because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

This thought was the beginning of a massive research for the 'exact method' and the 'perfect computer' to perform the simulation of an atomic system.

The problem with classical computation is that if one wants to perfectly simulate a quantum system, the calculation must take into account all the possible states in which the wave function can be found, this can be made via encoding and evaluation of the wave function on a discretized spatial grid in the position representation $\{\mathbf{r}\}$ or on a basis set in the orbitals representation. In both cases one can see that the representation, and therefore the memory to store the representation, grows superpolynomially as the system (i.e. the number of electrons and the number of

configurations) grows.

In the grid-based method a wave function can be written as

$$|\Psi\rangle = \sum_{\mathbf{x}_1, \dots, \mathbf{x}_N} \psi(\mathbf{x}_1, \dots, \mathbf{x}_N) \mathcal{A}(|\mathbf{x}_1, \dots, \mathbf{x}_N\rangle) \quad (1.30)$$

where $|\mathbf{x}_i\rangle = |\mathbf{r}_i\rangle|\sigma_i\rangle$ is a spatial and spin coordinate, $|\mathbf{r}_i\rangle = |x_i\rangle|y_i\rangle|z_i\rangle$, $\forall i \in \{1, 2, \dots, N\}$, $x_i, y_i, z_i \in \{0, 1, \dots, P-1\}$, and $\sigma_i \in \{0, 1\}$. In total there are $P^{3N} \times 2^N$ complex amplitudes ψ to store, thus the memory required scales exponentially with the size of the simulated system.

In the basis set method (let's take the second quantization scheme for simplicity) to write down a Slater determinant we only need to indicate which spin-orbitals are occupied by electrons

$$|\Psi\rangle = |n_0, n_1, \dots, n_i, \dots, n_{m-1}\rangle, \quad (1.31)$$

where $|n_i\rangle$ is the occupation number and it indicates the number of particles in spin-orbital $|\chi_i\rangle$, thus $|n_i\rangle = 1$ when $|\chi_i\rangle$ is occupied and $|n_i\rangle = 0$ when $|\chi_i\rangle$ is unoccupied, these are the only two possible values since we are working with fermions that follow Pauli exclusion principle, if we worked with bosons $|n_i\rangle$ would take values from 0 to ∞ . It is implied that the vector of occupation numbers has to be antisymmetrized and normalized when we expand it, e.g.

$$|1, 1, 0, \dots, 0\rangle = \frac{1}{\sqrt{2}}(|\chi_1(\mathbf{x}_1)\rangle|\chi_2(\mathbf{x}_2)\rangle - |\chi_2(\mathbf{x}_1)\rangle|\chi_1(\mathbf{x}_2)\rangle). \quad (1.32)$$

Hence one can see that, as the number of electrons grows, the number of configurations grows combinatorially as $\binom{m}{N}$, as we have seen before for the FCI wave function, and the memory scales in the same way.

One can also consider the most simple and fundamental encoding in the computational basis, let's say that a system, e.g. a particle, can be found in states $|0\rangle$, ground state, or $|1\rangle$, excited state, the generic wave function for this two-state system is

$$|\Psi\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle, \quad (1.33)$$

due to superposition. If we now add a new particle and write down the wave function for the composite system this becomes

$$|\Psi\rangle = \gamma_{00}|00\rangle + \gamma_{01}|01\rangle + \gamma_{10}|10\rangle + \gamma_{11}|11\rangle, \quad (1.34)$$

this is because in quantum mechanics states are vectors of Hilbert spaces and when we describe composite systems we have to take the tensor product of all the basis states of the Hilbert spaces of the individual systems. One can easily see that, adding more and more particles to this simple system, the number of states grows exponentially as 2^N , where N is the number of particles in the composite system, and so grows the memory required to store all of these states.

This argument focuses only on the computational memory to store the wave function, i.e. all the information necessary to describe a quantum system, let alone the

requirements to simulate the evolution of such wave function. In general one has to store the complete Hamiltonian, which, in a discrete finite space, is a matrix with the same size of its eigenvectors, i.e. the states of the wave function. Thus if the wave function needs 2^N complex numbers to be stored, the Hamiltonian needs $2^N \times 2^N$ complex numbers, which is an extraordinary amount of resources. For example, a 40 spin- $\frac{1}{2}$ particles need $2^{40} \approx 10^{12}$ numbers and its Hamiltonian needs a $2^{40} \times 2^{40}$ matrix with $\approx 10^{24}$ entries.

In this context Feynman's idea is simple: if we want to study a physical system (P), is it possible to take a simulator system (S) which resembles system P, elaborate it via the laws of quantum mechanics, extract properties and then translate them back to system P?

If the system S follows the laws of quantum mechanics than the encoding of the wave function and its evolution will translate naturally from one system to the other.

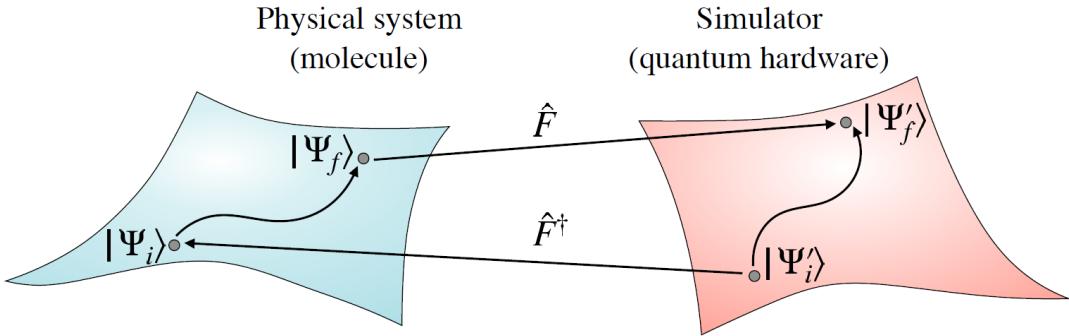


Figure 1.5: Scheme for the simulation of a physical system [33].

The system S is the one we can manipulate, this is what is called a 'universal quantum computer'. In the years next to Feynman's speech many researchers refined this intuition. First of all how is this simulator made, i.e. what kind of hardware one needs? And then what type of manipulations one needs to do, i.e. what kind of software or algorithm?

The history of quantum computers is full of new discoveries year by year, among the most admirable are Shor's algorithm, for finding the prime factors of an integer [43], and Grover's algorithm, for unstructured search [18], but we will only focus on the simulation of atomic systems.

In 1996 Seth Lloyd [27] found out a way to carry on a simulation 'efficiently' (in Chapter 3 we will give a definition of what efficiently means), he used qubits, which are the most simple two-state systems, in analogy to classical bits, to naturally store all the information about the wave function, and demonstrated that if the physical systems evolve according to local interactions, i.e. $H = \sum_i h_i$, their evolution can be written using the Lie-Trotter-Suzuki decomposition, commonly referred to as Trotterization,

$$U = e^{-iHt} = e^{-i\sum_i h_i t} = \left(\prod_i e^{-ih_i t/S} \right)^S + O(t^2/S), \quad (1.35)$$

where each operator can be applied S times to regain the full time evolution. Similarly in 2005 Alan Aspuru-Guzik et al. [5] described an algorithm to provide a solution for eq. (1.7), i.e. find the energy of the ground and the excited states of a system. The algorithm exponentiates the eigenvalues of the Hamiltonian and retrieve them through the quantum Fourier transform (QFT), a version of the Fourier transform designed for quantum computers.

These two algorithms go under the name of quantum phase estimation (QPE) algorithms, since the goal of them is to find the phase of the exponential in the evolution formula, in this case the energy of the system.

Remaining in the field of quantum computation applied to chemical simulations, in 2014 Alberto Peruzzo et al. [36] derived another algorithm called variational quantum eigensolver (VQE). This is a hybrid quantum-classical algorithm since, after the manipulation, the measured values are optimized iteratively through a classical computer, as it happens in machine learning algorithms. This algorithm was meant to be run on a quantum computer with low computing power, the so-called noisy intermediate-scale quantum (NISQ) devices, which we describe in Chapter 5, in stark contrast to previous algorithms which were meant to be run on a quantum computer with high computing power, the so-called fault-tolerant quantum computation (FTQC) devices, which we describe in Chapter 4.

Chapter 2

Density functional theory

2.1 Introduction

Amongst the most used methods in computational chemistry nowadays, especially for research on industrially relevant compounds, are the ones based on density functional theory (DFT), that is the methods in which one computes the electron density ρ , instead of the wave function $|\Psi\rangle$, of the system. The motivation for this is that the wave function of an N electrons system depends on $3N$ spatial coordinates and N spin coordinate, while the electron density is the square of the wave function, integrated over $N - 1$ electron coordinates, thus while the complexity of the wave function increases exponentially with the number of electrons, the electron density has the same number of variables, independent of the system size.

$$\rho(\mathbf{r}) = N \sum_{s_1, \dots, s_N} \int d\mathbf{r}_2 \dots d\mathbf{r}_N |\Psi(\mathbf{r}, s_1, \mathbf{r}_2, s_2, \dots, \mathbf{r}_N, s_N)|^2 \quad (2.1)$$

The electron density can be linked to all the properties of the Hamiltonian and therefore of the atomic system. First of all, by integrating it over all space we can get the total number of electrons N ,

$$N = \int \rho(\mathbf{r}) d\mathbf{r}. \quad (2.2)$$

Secondly, it is possible to get information about the nuclei via the formula

$$\left. \frac{\partial \bar{\rho}(r_A)}{\partial r_A} \right|_{\mathbf{r}_A=0} = -2Z_A \rho(\mathbf{r}_A), \quad (2.3)$$

where Z is the atomic number, r_A is the radial distance and $\bar{\rho}$ is the spherically averaged density.

This is not a simple formalism for finding the energy, but it shows that given a known density, one could form the Hamiltonian operator, solve the TISE and determine the wave function along with the energy eigenvalues.

2.2 First approach

The simplest approach to this calculation is to consider the system to be classical and provide the total energy as a sum of kinetic and potential energy.

The potential energy components are easily determined. The attraction between the density and the nuclei is

$$V_{ne}[\rho(\mathbf{r})] = \sum_I^{nuclei} \int \frac{Z_I}{|\mathbf{r} - \mathbf{r}_I|} \rho(\mathbf{r}) d\mathbf{r}, \quad (2.4)$$

and the self-repulsion of a classical charge distribution is

$$V_{ee}[\rho(\mathbf{r})] = \frac{1}{2} \iint \frac{\rho(\mathbf{r}_i)\rho(\mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|} d\mathbf{r}_i d\mathbf{r}_j. \quad (2.5)$$

For the kinetic energy we can use the result of Thomas and Fermi from 1927 for the jellium model, i.e. a system composed of electrons moving in an infinite volume characterized by a uniformly distributed positive charge, also called uniform electron gas,

$$T_{jellium}[\rho(\mathbf{r})] = \frac{3}{10} (3\pi^2)^{2/3} \int \rho^{5/3}(\mathbf{r}) d\mathbf{r}. \quad (2.6)$$

These functions depends on density ρ which in turn depends on position \mathbf{r} , for this reason T and V are called density functionals.

The Thomas-Fermi equations, together with an assumed variational principle, represented the first effort to define a density functional theory, the energy is computed with no reference to a wave function.

However this formulation is highly inaccurate, due to the large approximation on the interelectronic repulsion, eq. (2.5), which does not take into account the effects associated with correlation and exchange.

A hole function can be introduced to correct for the energetic errors introduced by assuming classical behaviour, this acts like a negative density.

In the following years Slater suggested to consider a functional for the exchange term by approximating the exchange hole as a sphere of constant potential with a radius depending on the magnitude of the density at that position

$$E_x[\rho(\mathbf{r})] = -\frac{9\alpha}{8} \left(\frac{3}{\pi} \right)^{1/3} \int \rho^{4/3}(\mathbf{r}) d\mathbf{r}, \quad (2.7)$$

this is called the Slater exchange. Within Slater's derivation the value of α is 1.

Later on, Bloch and Dirac derived a similar expression with $\alpha = \frac{2}{3}$. The combination of these expressions defines the Thomas-Fermi-Dirac model, although it too remains sufficiently inaccurate for modern use.

Early DFT models found widespread use in the solid state physics community, however had little impact on chemistry due to large errors in molecular calculation and the failure to provide a rigorous foundation.

2.3 Analysis

2.3.1 Hohenberg-Kohn theorems

In 1964 Hohenberg and Kohn proved two theorems that established the legitimacy for DFT as a computational chemistry method.

1. The ground state electron density ρ_0 completely determines the external potential.

This is also called 'the existence theorem', because it proves the existence of a density function that satisfies the properties we want.

The external potential represents the attraction between the electron density and the nuclei. The proof proceeds via *reductio ad absurdum* and takes advantage of the variational principle for the ground state energy from quantum mechanics [13]. If the electron density determines the external potential then it determines the Hamiltonian and therefore the wave function of the system.

2. For any positive density ρ , such that $\int \rho(\mathbf{r}) d\mathbf{r} = N$,

$$\langle \Psi(\rho) | H(\rho) | \Psi(\rho) \rangle = E(\rho) \geq E_0 \quad (2.8)$$

This is also called the 'variational theorem', because it provides a variational principle for the electron density function.

Just as the HF-based methods, generally called molecular orbital (MO) theory, we need a means to optimize the quantity we aim to find. This theorem shows that the density obeys a variational principle. So we can keep choosing different densities and those that provide lower energies are closer to the correct one, but we have no means to improve the candidate density that we choose.

These two theorems make a solid foundation for DFT, since there is a formal way that we can get the energy values of the system, but we still can not avoid solving the TISE, which is the motivation for which we study DFT. So we need further refinement to the method in order to use it.

Note also that the Hamiltonian determines not just the ground state wave function, but all the excited state wave functions as well, so there is an enormous amount of information coded in the density.

2.3.2 Kohn-Sham self-consistent field method

In 1965 Kohn and Sham described a method to find energy values using DFT. We know that the Hamiltonian for a non-interacting system of electrons: can be expressed as a sum of one-electron operators, has eigenfunctions that are Slater determinants of the individual one-electron eigenfunctions and has eigenvalues that are simply the sum of the one-electron eigenvalues. So we take a fictitious system of non-interacting electrons that have, for their overall ground state density, the same density as some real system of interest where electrons do interact. We know we can do that because the density determines the position and the atomic numbers of the nuclei and these quantities are necessarily identical.

The energy functional can be written as

$$E[\rho(\mathbf{r})] = T_{ni}[\rho(\mathbf{r})] + V_{ne}[\rho(\mathbf{r})] + V_{ee}[\rho(\mathbf{r})] + \Delta T[\rho(\mathbf{r})] + \Delta V_{ee}[\rho(\mathbf{r})], \quad (2.9)$$

where ni stands for *non-interacting* and ΔT and ΔV_{ee} are the corrections to the kinetic energy and to the electron-electron repulsion energy, due to non-classical terms

of interactions between particles that arises from quantum mechanical treatment. Those are the terms we have to add to the non-interacting system to regain the real system.

This is a useful formulation because we already know the form of kinetic and potential energy from the classical scheme. However, the cost of this facilitation is that we have to reintroduce orbitals function to give an exact definition of these functionals, in contrast with first approaches to DFT where kinetic and potential energy were completely defined by density but the model used to define them was approximated. This means that the complexity of the method increases from 3 to $3N$ spatial variables.

We can rewrite eq. (2.9) as

$$E[\rho(\mathbf{r})] = \sum_i^N \left(\langle \chi_i | -\frac{1}{2} \nabla_i^2 | \chi_i \rangle - \langle \chi_i | \sum_I^{nuclei} \frac{Z_I}{r_{iI}} | \chi_i \rangle \right) + \sum_i^N \left(\langle \chi_i | \frac{1}{2} \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}_i - \mathbf{r}'|} d\mathbf{r}' | \chi_i \rangle \right) + E_{xc}[\rho(\mathbf{r})], \quad (2.10)$$

where the density for the exact eigenfunction for the non-interacting system, i.e. a Slater-determinantal wave function, is

$$\rho = \sum_i^N \langle \chi_i | \chi_i \rangle. \quad (2.11)$$

The corrections we have introduced have been grouped under $E_{xc}[\rho(\mathbf{r})]$, which is called the 'exchange-correlation energy' or the 'exchange-correlation functional'. This term includes not only the effects of quantum mechanical exchange and correlation, but also the correction for the classical self-interaction energy and for the difference in kinetic energy between the fictitious non-interacting system and the real one.

As we have done before with the HF method we can find the orbitals $|\chi_i\rangle$ that minimize the energy E . To do this we can write the pseudoeigenvalue equations as

$$h_i^{KS} |\chi_i\rangle = \epsilon_i |\chi_i\rangle, \quad (2.12)$$

where

$$h_i^{KS} = -\frac{1}{2} \nabla_i^2 - \sum_I^{nuclei} \frac{Z_I}{r_{iI}} + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}_i - \mathbf{r}'|} d\mathbf{r}' + V_{xc} \quad (2.13)$$

is the Kohn-Sham (KS) one-electron operator, defined in an analogous way to the Fock operator, and

$$V_{xc} = \frac{\delta E_{xc}}{\delta \rho} \quad (2.14)$$

is a functional derivative.

Since the functional E we are minimizing is exact (we know that because we have considered a simple system) the orbitals $|\chi_i\rangle$ must provide the exact density of the

real system. Furthermore, these orbitals are the correct ones to form the Slater-determinantal eigenfunction for the separable non-interacting Hamiltonian defined as the sum of the KS operators,

$$\sum_i^N h_i^{KS} |\chi_1 \chi_2 \dots \chi_N\rangle = \sum_i^N \epsilon_i |\chi_1 \chi_2 \dots \chi_N\rangle. \quad (2.15)$$

It is therefore justified the use of the fictitious non-interacting system we have introduced before [13].

To obtain the KS orbitals we follow the same path developed for MO theory, we express them within a basis set of functions $\{|\phi_\nu\rangle\}$ and we determine the individual orbital coefficients by solution of a secular equation analogous to the one for HF theory except that the elements $F_{\mu\nu}$ are replaced by $K_{\mu\nu}$ defined as

$$K_{\mu\nu} = \langle \phi_\mu | \left(-\frac{1}{2} \nabla_i^2 - \sum_I^{nuclei} \frac{Z_I}{r_{iI}} + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}_i - \mathbf{r}'|} d\mathbf{r}' + V_{xc} \right) | \phi_\nu \rangle. \quad (2.16)$$

Now that we have defined the target of the method we have to define the procedure to obtain it.

We take advantage of the second HK theorem, i.e. the variational principle, and define the procedure as follow:

1. **Choose a basis set**, sometimes orbitals and density use different basis sets;
2. **Choose a molecular geometry;**
3. **Compute overlap and one-electron integrals;**
4. **Guess initial density**, this can be constructed as a matrix;
5. **Construct and solve KS secular equation;**
6. **New orbitals are determined from the solution, from these construct the new density;**
If they are sufficiently similar compute the energy by plugging the final density in eq. (2.9), this is in contrast with HF theory where the energy is evaluated as the expectation value of the Hamiltonian operator acting on the HF Slater determinant, otherwise repeat from 5;
7. **Compare the two densities.**
8. **Optional: optimize molecular geometry**, i.e. determine if the structure corresponds to a stationary point.

Due to the repetition and the similar structure to HF method, this procedure is also called KS self-consistent field (SCF) method.

The scheme is depicted in Figure 2.1.

There is a key difference between HF theory and DFT. HF is a deliberately approximate theory, whose development was in part motivated by an ability to solve the

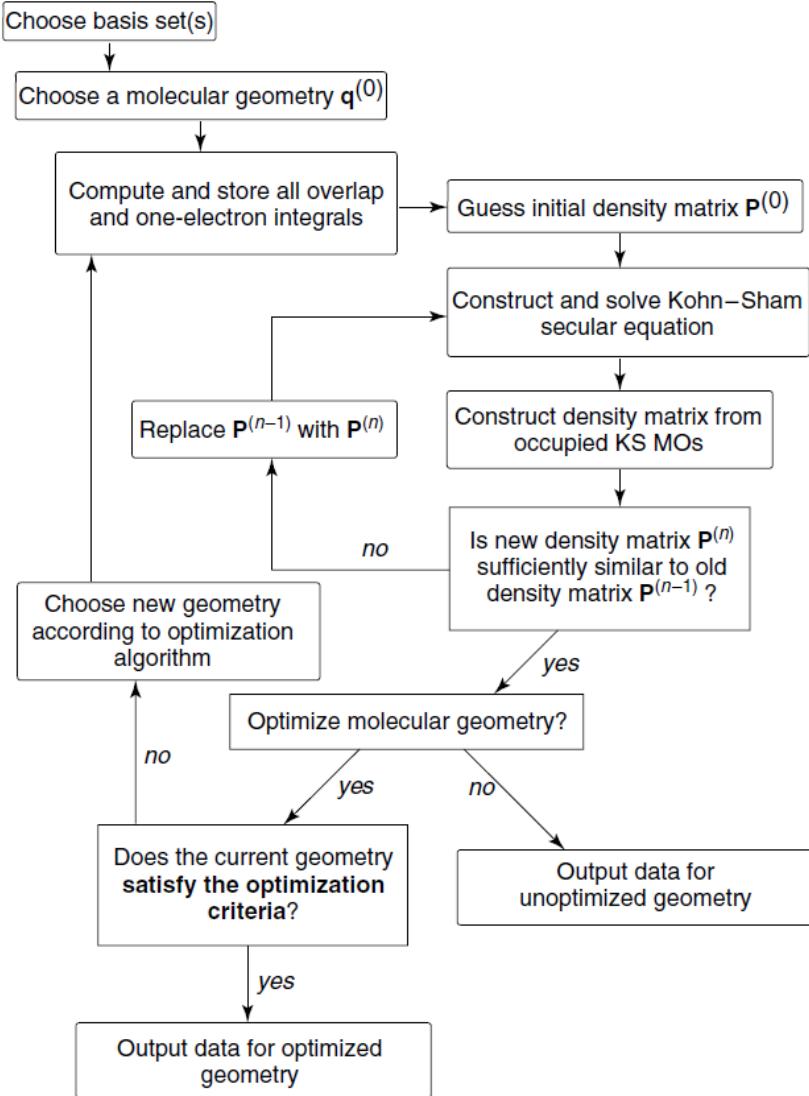


Figure 2.1: Flowchart for the KS SCF method [13].

relevant equations exactly. DFT, instead, contains no approximation, all we need to know is E_{xc} as a function of ρ . However while Hohenberg and Kohn proved that a functional of the density must exist, their proofs provide no guidance whatsoever as to its form. As a result, considerable research effort has gone into finding functions of the density that may be expected to reasonably approximate E_{xc} . Therefore, DFT is an exact theory, but the relevant equations must be solved approximately because a key operator has unknown form.

2.3.3 Exchange-correlation functionals

From the first version of DFT hundreds of different exchange-correlation functionals have been introduced. Here we will show some of the most important ones. First of all, the functional is expressed as an interaction between the electron density

and an energy density, that is dependent on the electron density

$$E_{xc}[\rho(\mathbf{r})] = \int \rho(\mathbf{r}) \epsilon_{xc}[\rho(\mathbf{r})] d\mathbf{r}, \quad (2.17)$$

and the energy density ϵ_{xc} is treated as a sum of individual exchange, ϵ_x , and correlation, ϵ_c , contributions. As an example, the Slater exchange energy density is

$$\epsilon_x[\rho(\mathbf{r})] = -\frac{9\alpha}{8} \left(\frac{3}{\pi}\right)^{1/3} \rho^{1/3}(\mathbf{r}). \quad (2.18)$$

For simplicity we will not consider spin polarization, although it is possible to formulate the functional to account for it.

An important consideration is that, although we do not know the form of the exact functional, we know a number of properties that it should have: [22]

1. The energy functional should be self-interaction-free, i.e. the exchange energy for a one-electron system, such as the hydrogen atom, should exactly cancel the Coulomb energy, and the correlation energy should be zero;
2. When the density becomes constant, the uniform electron gas result should be recovered;
3. The coordinate scaling of the exchange energy should be linear, i.e. multiplying the electron coordinates with a constant factor should result in a similar linear scaling of the exchange energy;
4. No direct scaling law applies for the correlation energy, but scaling the electron coordinates by a factor larger than 1 should increase the magnitude of the correlation and viceversa. In the low density limit, the scaling becomes linear, as for the exchange energy;
5. As the scaling parameter goes to infinity, the correlation energy for a finite system approaches a negative constant;
6. The Lieb–Oxford condition places an upper bound for the exchange–correlation energy relative to the Local density approximation (LDA) exchange energy

$$E_x[\rho(\mathbf{r})] \geq E_{xc}[\rho(\mathbf{r})] \geq 2.273 E_x^{LDA}[\rho(\mathbf{r})]; \quad (2.19)$$

7. The exchange potential should show an asymptotic $-r^{-1}$ behaviour as $r \rightarrow \infty$. Furthermore, the exchange–correlation potential is discontinuous as a function of the number of electrons, by an amount corresponding to the difference between the ionization potential and electron affinity;
8. The correlation potential should show an asymptotic $-\frac{1}{2}\alpha r^{-4}$ behaviour, with α being the polarizability of the $N_{elec} - 1$ system.

The first type of functionals we study is the one derived from the uniform electron gas, where the density has the same value at every position, this type is also called local density approximation (LDA), to express that the value of ϵ_{xc} at some position \mathbf{r} could be computed exclusively from the value of ρ at that position, i.e the local value of ρ .

As discussed before the exchange energy for the uniform electron gas can be computed exactly and is given by eq. (2.18), where α can take different values based on the model considered.

Systems including spin polarization, e.g. open-shell systems, must use spin-polarized formalism, and its greater generality is sometimes distinguished with the expression local spin density approximation (LSDA).

As for the correlation energy density no analytical derivation of this functional has been proven possible and it is sometimes computed via quantum Monte Carlo techniques.

One obvious way to improve the correlation functional is to make it depend not only on the local value of the density, but on the extent to which the density is locally changing, i.e. the gradient of the density. The more common term in modern nomenclature for functionals that depend on both the density and the gradient of the density is ‘gradient corrected’. Including a gradient correction defines the ‘generalized gradient approximation’ (GGA).

Most gradient corrected functionals are constructed with the correction being a term added to the LDA functional, i.e.

$$\epsilon_{x/c}^{GGA}[\rho(\mathbf{r})] = \epsilon_{x/c}^{LDA}[\rho(\mathbf{r})] + \Delta\epsilon_{x/c} \left[\frac{|\nabla\rho(\mathbf{r})|}{\rho^{4/3}(\mathbf{r})} \right]. \quad (2.20)$$

The first widely popular GGA exchange functional was developed by Becke and it is usually abbreviated simply B. It further incorporates a single empirical parameter the value of which was optimized by fitting to the exactly known exchange energies of the six noble gas atoms He through Rn.

Other exchange functionals similar to the Becke example, in one way or another, have appeared, including CAM, FT97, O, PW, mPW, and X, where X is a particular combination of B and PW found to give improved performance over either.

Alternative GGA exchange functionals have been developed based on rational function expansions of the reduced gradient. These functionals, which contain no empirically optimized parameters, include B86, LG, P, PBE, and mPBE.

With respect to correlation functionals, corrections to the correlation energy density following eq. (2.20) include B88, P86, and PW91.

Another popular GGA correlation functional, LYP, does not correct the LDA expression but instead computes the correlation energy in toto. It contains four empirical parameters fit to the helium atom. Of all of the correlation functionals discussed, it is the only one that provides an exact cancellation of the self-interaction error in one-electron systems.

Typically in the literature, a complete specification of the exchange and correlation functionals is accomplished by concatenating the two acronyms in that order. Thus, for instance, a BLYP calculation combines Becke’s GGA exchange with the GGA

correlation functional of Lee, Yang, and Parr.

The next step in functional improvement might be to take account of the second derivative of the density, i.e., the Laplacian. Such functionals are termed meta-GGA (MGGA) functionals as they go beyond simply the gradient correction. However, numerically stable calculations of the Laplacian of the density pose something of a technical challenge, and the somewhat improved performance of MGGA functionals over GGA analogs is balanced by this slight drawback.

An alternative MGGA formalism that is more numerically stable is to include in the exchange-correlation potential a dependence on the kinetic-energy density τ , defined as

$$\tau(\mathbf{r}) = \sum_i^{\text{occupied}} \frac{1}{2} |\nabla \chi_i(\mathbf{r})|^2, \quad (2.21)$$

where the χ_i are the self-consistently determined KS orbitals.

Another important type of functional is defined by the adiabatic connection method (ACM).

Let's imagine we have a parameter λ we can tune to smoothly convert the non-interacting KS reference system to the real, interacting system. Using the Hellmann–Feynman theorem, one can show that the exchange-correlation energy can then be computed as

$$E_{xc} = \int_0^1 \langle \Psi(\lambda) | V_{xc}(\lambda) | \Psi(\lambda) \rangle d\lambda. \quad (2.22)$$

To evaluate this integral, it is helpful to adopt a geometric picture, as shown in Figure 2.2.

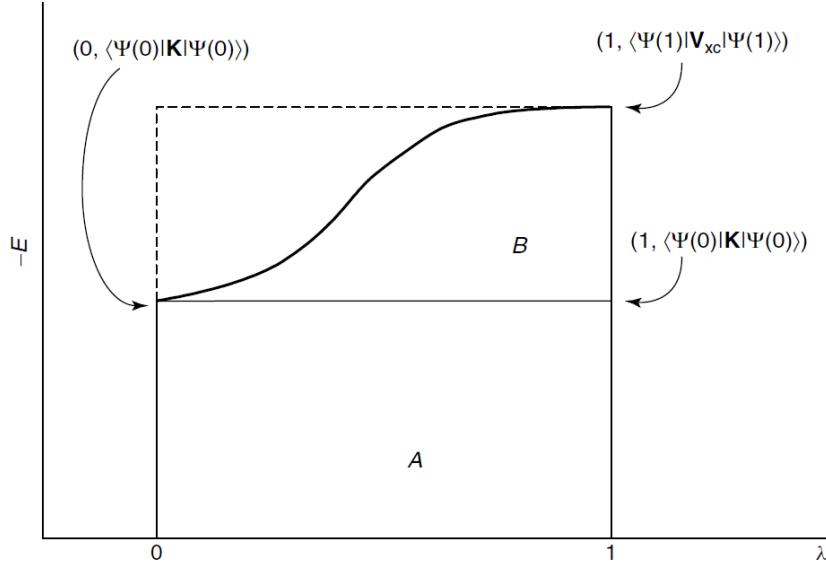


Figure 2.2: Geometrical scheme for the exchange-correlation energy [13].

We seek the area under the curve defined by the expectation value of V_{xc} . While we know very little about V and $|\Psi\rangle$ as functions of λ in general, we can evaluate

the left endpoint of the curve. In the non-interacting limit, the only component of V is exchange (deriving from antisymmetry of the wave function). Moreover, the Slater determinant of KS orbitals is the exact wave function for the non-interacting Hamiltonian operator. Thus, the expectation value is the exact exchange for the non-interacting system, which can be computed just as it is in HF calculations except that the KS orbitals are used. The total area under the expectation value curve thus contains the rectangle having the curve's left endpoint as its upper left corner, which has area E_x^{HF} .

The remaining area is some fraction z of the area corresponding to the rectangle sitting immediately on top of the first; the second rectangle has area $\langle \Psi(1) | V_{xc}(1) | \Psi(1) \rangle - E_x^{HF}$. Unfortunately, not only do we not know z , we also do not know the expectation value of the fully interacting exchange-correlation potential applied to the fully interacting wave function. However, we may regard z as an empirical constant to be optimized. In that case, we may as well approximate the unknown right endpoint, and a convenient approximation is the E_{xc} computed directly by some choice of DFT functional.

We see then that the total area under the expectation value curve can be written as

$$E_{xc} = E_x^{HF} + z(E_{xc}^{DFT} - E_x^{HF}) = (1 - a)E_{xc}^{DFT} + aE_x^{HF}, \quad (2.23)$$

where $a = 1 - z$.

The transition from one system to another one makes the ACM similar to the KS SCF scheme. In the latter case, one does not know the exact kinetic energy as a function of the density, so one employs a scheme where a large portion of it is computed exactly (as the expectation value of the kinetic energy operator over the KS determinant) and worries later about the small remainder. So too, the ACM approach computes a large fraction of the total exchange-correlation energy, and then worries later about the difference between the total and the exact (HF) exchange.

We may also think whether inclusion of additional empirical parameters results in sufficient improvement to make such inclusion worthwhile. Becke was the first to do this, developing the 3-parameter functional expression

$$E_{xc}^{B3PW91} = (1 - a)E_x^{LSDA} + aE_x^{HF} + b\Delta E_x^B + E_c^{LSDA} + c\Delta E_c^{PW91}, \quad (2.24)$$

where a , b and c were optimized to 0.20, 0.72 and 0.81.

A similar model is the B3LYP model, defined as

$$E_{xc}^{B3LYP} = (1 - a)E_x^{LSDA} + aE_x^{HF} + b\Delta E_x^B + (1 - c)E_c^{LSDA} + c\Delta E_c^{LYP}, \quad (2.25)$$

where a , b and c have the same values as in B3PW91. Of all modern functionals, B3LYP has proven the most popular to date [11].

Because they incorporate HF and DFT exchange, ACM methods are also called hybrid methods.

Another possible type of functionals is the double-hybrid, i.e. where we combine LDA-type functionals with HF energy and post-HF energy functionals.

Finally, one of the latest type of functionals is the Minnesota Functionals (Myz). This is a group of highly parameterized approximate exchange-correlation energy functionals. These functionals are based on the meta-GGA approximation, i.e. they include terms that depend on the kinetic energy density τ , and are all based on complicated functional forms parametrized on high-quality benchmark databases, some examples are: M05, M05-2X, M06-L, M06-2X, M08-HX, M11, M12-L, M15.

2.4 Computational requirements

Let's now evaluate the scaling, the advantages and the disadvantages of the DFT method.

In the KS SCF method, if we write the density with an auxiliary basis set as

$$\rho(\mathbf{r}) = \sum_{i=1}^{m'} c_i \Omega_i(\mathbf{r}), \quad (2.26)$$

we can find that the number of Coulomb integrals requiring evaluation in order to compute all the KS matrix elements in eq. (2.16) is formally $O(m^2 m')$, where m is the number of KS atomic orbital basis functions, instead of $O(m^4)$, as for HF theory. As a result, the formal bottleneck in solving the KS SCF equations is matrix diagonalization, which scales as $O(m^3)$, and one frequently sees in the literature reference to this total scaling behavior associated with DFT.

Compared to HF theory, DFT includes electron correlation and does so in a fashion that scales more favorably with respect to system size. However, many electronic structure programs do not employ auxiliary basis sets to represent the density, choosing instead to compute it in the HF-like way as a product of KS-orbital basis functions (the motivation for this choice was primarily historical: existing HF codes could be easily modified to carry out DFT calculations with this choice), in which case formal $O(m^4)$ scaling is not reduced.

Therefore the scaling is, in principle, no worse than $O(m^3)$, but if we use a hybrid method, e.g. with the B3LYP functional, we pass through the calculation of the HF energy thus the scaling of the method rise to $O(m^4)$, as we have seen before.

The local density approximation is very simple and remarkably reliable for the structure, elastic moduli and relative phase stability of many materials, but is less accurate for binding energies and details of the energy surface away from equilibrium geometries, e.g. transition states. The GGA family of functionals improves binding energies to average errors of 20 kcal/mol and relative errors of 3-7% while meta-GGA and hybrid-exchange functionals reduce these errors to 3-5 kcal/mol and 2-3%. This is close to the accuracy required for predictive simulations of thermochemical properties, i.e chemical accuracy (1 kcal/mol). The GGA, meta-GGA and hybrid functionals retain, and somewhat improve on, the LDA's excellent description of bonds lengths with typical errors in the region of 1-2 milli-Angstrom. Using these functionals elastic moduli are reproduced to within 10% and vibrational frequencies to $\sim 40\text{cm}^{-1}$. There is a distinct tendency for functionals that are highly parameterised and fitted to the properties of molecular systems to perform somewhat better

than lightly parameterised functions for molecules, but to perform relatively poorly in simulations on periodic materials [19].

The main limitation of DFT method is that it fundamentally depends of the functional one uses. Every year new functionals are defined and the accuracy the method can reach depends on the functional chosen and even for the specific system one wants to simulate. This means that if one use the simplest functionals there is no guarantee to improve the result by using a more complicated functional. The DFT method has no systematic improvability.

The other limitation we already underlined is that it gives an exact solution, but the equations must be solved approximately because the form of the functional is approximated. This means that the solution will not be exact, thus we can only consider a minimum accuracy target to reach.

Moreover, as there are more well-characterized operators then there are generic property functionals of the density, wave functions clearly have broader utility. As a simple example, consider the total energy of interelectronic repulsion. Even if we had the exact density for some system, we do not know the exact exchange-correlation energy functional, and thus we cannot compute the exact interelectronic repulsion. However, with the exact wave function it is a simple matter of evaluating the expectation value for the interelectronic repulsion operator to determine this energy,

$$E_{ee} = \langle \Psi | \frac{1}{2} \sum_{i \neq j} \frac{1}{r_{ij}} | \Psi \rangle \quad (2.27)$$

Another key example is in the area of dynamics, where transition probabilities depend on matrix elements between different wave functions. Because densities do not have phases as wave functions do, multistate resonance effects, interference effects, etc., are not readily evaluated within a DFT formalism. In practice, however, one can use KS orbitals, whose shapes tend to be remarkably similar to canonical HF molecular orbitals, and they can be quite useful in qualitative analysis of chemical properties, but nevertheless they represent an approximation of the solution to the TISE.

Lastly, dependently on the functional used one can have problems in finding specific properties of the system, e.g. older popular functionals have deficiencies in estimating barrier heights, transition metal bond energies and noncovalent interactions, as shown in Figure 2.3.

DFT is one of the most used methods in computational chemistry, it can reach a good level of accuracy while dealing with low scaling with respect to system size. This is why it is the method we take as a standard to compare the algorithms which runs on quantum computers.

In Figure 2.4 the main algorithms used for quantum simulation on a classical computer are depicted. The algorithms previously described are the ones that achieve a high level of accuracy up to the highest accuracy for the FCI method, but there are also cheaper methods which we did not covered.

The aim of quantum simulation on quantum computers is to find a method with

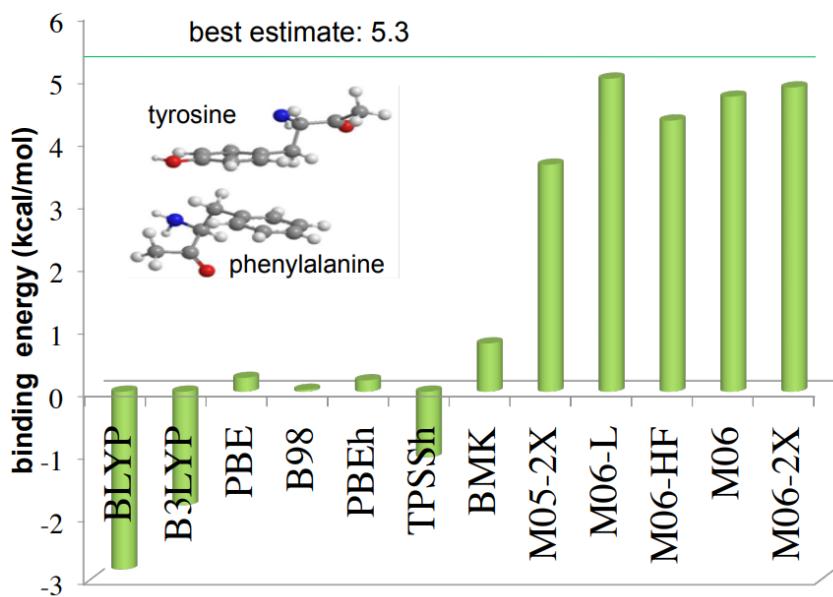


Figure 2.3: Comparison of the results from different functionals.

either higher accuracy or lower computational resources compared to the already existing ones, this is represented by the purple arrow.

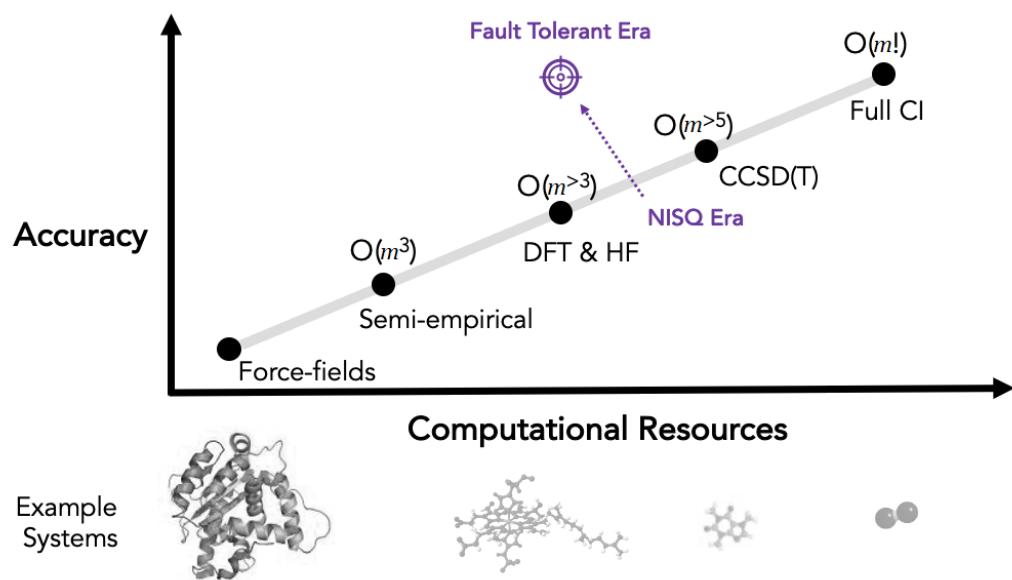


Figure 2.4: Accuracy vs Computational resources of the main algorithms.

Chapter 3

Quantum computing

3.1 Introduction

As mentioned in the introduction, quantum computers are specialized calculators which exploit the laws of quantum mechanics to perform the computations. This process is also called 'quantum computing' or 'quantum computation'.

A definition of quantum computation, together with quantum information, is the study of the information processing tasks that can be accomplished using quantum mechanical systems.

The first historical strand contributing to the development of quantum computation and quantum information is the interest, dating to the 1970s, of obtaining complete control over single quantum systems. This research comprehend, for example, experiments on Bell's inequality formula and entanglement between particles.

This field is very important for scientific research, since it represents a method for probing a new regime of Nature of which we know so little.

The second motivation is the intersection with the field of computer science. Since the standardization of digital computers in 1947, year in which John Bardeen, Walter Brattain, and Will Shockley developed the transistor, classical computer hardware has grown in power at an amazing pace, following the so-called Moore's law, which states that computer power will double constantly roughly once every two years. Nowadays, though conventional approaches to the fabrication of computer technology are beginning to run up against fundamental difficulties of size, quantum effects are beginning to interfere in the functioning of electronic devices as they are made smaller and smaller.

Still, at a more fundamental level are classical and quantum computers different from one another? Notoriously, in 1936 mathematician Alan Turing developed in detail an abstract notion of what we would now call a programmable computer, a model for computation now known as the Turing machine, in his honor. Turing showed that there is a universal Turing machine that can be used to simulate any other Turing machine. Furthermore, he claimed that the universal Turing machine completely captures what it means to perform a task by algorithmic means. That is,

if an algorithm can be performed on any piece of hardware, then there is an equivalent algorithm for a universal Turing machine which performs exactly the same task as the algorithm running on the personal computer. This assertion, known as the Church–Turing thesis in honor of Turing and another pioneer of computer science, Alonzo Church, asserts the equivalence between the physical concept of what class of algorithms can be performed on some physical device with the rigorous mathematical concept of a universal Turing machine. What was noticed in the late 1960s and early 1970s was that it seemed as though the Turing machine model of computation was at least as powerful as any other model of computation, in the sense that a problem which could be solved efficiently in some model of computation could also be solved efficiently in the Turing machine model, by using the Turing machine to simulate the other model of computation. This observation was codified into a strengthened version of the Church–Turing thesis:

Any algorithmic process can be simulated efficiently using a Turing machine.

The key strengthening in the strong Church–Turing thesis is the word efficiently. Roughly speaking, an efficient algorithm is one which runs in time polynomial in the size of the problem solved. In contrast, an inefficient algorithm requires superpolynomial (typically exponential) time. If the strong Church–Turing thesis is correct, then it implies that no matter what type of machine we use to perform our algorithms, that machine can be simulated efficiently using a standard Turing machine. This is an important strengthening, as it implies that for the purposes of analyzing whether a given computational task can be accomplished efficiently, we may restrict ourselves to the analysis of the Turing machine model of computation. The first major challenge to the strong Church–Turing thesis arose in the mid 1970s, when Robert Solovay and Volker Strassen showed that it is possible to test whether an integer is prime or composite using a randomized algorithm. This, together with other such algorithms brought to a modification of the strong Church–Turing thesis:

Any algorithmic process can be simulated efficiently using a probabilistic Turing machine.

This ad hoc modification of the strong Church–Turing thesis rises many doubts. Might it not turn out at some later date that yet another model of computation allows one to efficiently solve problems that are not efficiently soluble within Turing’s model of computation? Is there any way we can find a single model of computation which is guaranteed to be able to efficiently simulate any other model of computation?

Motivated by this question, in 1985 David Deutsch asked whether the laws of physics could be used to derive an even stronger version of the Church–Turing thesis. Because the laws of physics are ultimately quantum mechanical, Deutsch was naturally led to consider computing devices based upon the principles of quantum mechanics. These devices, quantum analogues of the machines defined forty-nine years earlier by Turing, led ultimately to the modern conception of a quantum computer.

It is not clear whether Deutsch’s notion of a universal quantum computer is sufficient to efficiently simulate an arbitrary physical system. Proving or refuting this

conjecture is one of the great open problems of the field of quantum computation and quantum information. It is possible, for example, that some effect of quantum field theory or an even more exotic effect based in string theory, quantum gravity or some other physical theory may take us beyond Deutsch's universal quantum computer, giving us a still more powerful model for computation.

What Deutsch's model of a quantum computer did enable was a challenge to the strong form of the Church–Turing thesis. Deutsch asked whether it is possible for a quantum computer to efficiently solve computational problems which have no efficient solution on a classical computer, even a probabilistic Turing machine. He then constructed a simple example suggesting that, indeed, quantum computers might have computational powers exceeding those of classical computers.

It turns out that while an ordinary computer can be used to simulate a quantum computer, it appears to be impossible to perform the simulation in an efficient fashion. Thus quantum computers offer an essential speed advantage over classical computers. This speed advantage is so significant that many researchers believe that no conceivable amount of progress in classical computation would be able to overcome the gap between the power of a classical computer and the power of a quantum computer.

The remarkable first step taken by Deutsch was in fact improved in the subsequent decade by many people, culminating in the already mentioned Peter Shor's 1994 demonstration that the problem of finding the prime factors of an integer, and the so-called 'discrete logarithm problem', could be solved efficiently on a quantum computer [43] and in 1995 when Lov Grover showed that the problem of conducting a search through some unstructured search space could also be sped up on a quantum computer [18].

More recently, many companies, among which Google, IBM and Microsoft, invested a great deal of money in research and development of this new technology. In 2019 researchers at Google claimed they achieved the so-called 'quantum supremacy' with a 53 qubits quantum processor called Sycamore, a term to describe the huge advantage a process executed on a quantum computer can reach against the same process executed on a classical computer. The process consisted in sampling the probability distribution of all the possible outputs of a pseudorandom quantum circuit, comparing them with the theoretically expected values [4].

On the other side, IBM is among the first companies to have developed and made available quantum computers, even though with little computational power, up to tens of qubits, and plans to build more and more powerful ones in the coming years, as showed in the IBM roadmap for quantum computation (we will focus more on this topic in Chapter 6).

Let's now describe the fundamentals of quantum computing.

Classical computer circuits consist of wires and logic gates. The wires are used to carry information around the circuit, while the logic gates perform manipulations of the information, converting it from one form to another.

In quantum computing there are wires used to carry quantum information, i.e.

qubits, and quantum logic gates, which manipulate the information. They are generally presented in a circuital form, where single lines represent qubits and parallel lines represent classical bits.

A collection of qubits, as well as a collection of bits, is called a register. These are the targets of the manipulation.

Specific symbols representing the gates are placed on the path to indicate the transformations applied to the register.

A quantum algorithm is an algorithm designed for a quantum computer, it can be described mathematically as a sequence of operations on quantum states or pictured in a quantum circuit, which generally end with the measurement of the states of the registers.

As we see there is a one-to-one correspondence between the two paradigms. This is referred to as digital quantum computing, but other type of schemes can be used to carry on the calculation, such as analog quantum computing or adiabatic quantum computing.

3.1.1 Qubits

A qubit, short for quantum bit, is the basic unit of quantum information, analogous to a bit for classical information. It is a two-state quantum system that can be described as a vector in a two dimension Hilbert space \mathcal{H} , whose basis vectors are pure states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, together called the 'computational basis'. This definition derives naturally from the first postulate of quantum mechanics. The state of a qubit is usually expressed as a superposition of these two basis states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad (3.1)$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$, or in a form that exploit the normalization of the state

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \quad (3.2)$$

with $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. This last representation is called the 'Bloch sphere', because θ and ϕ represent the azimuthal and the polar variables of the spherical coordinate system.

A property of qubits, as stated in Chapter 1, is that if we want to process more than one of them, then a vector of the composite system can be expressed as the tensor product of all the basis states of the Hilbert spaces of the individual systems and the dimension of the composite system basis is the product of the dimensions of the individual systems. This also derives naturally from the first postulate of quantum mechanics.

As an example, if $\{|0\rangle, |1\rangle\}$ is a basis for system A and system B, then the composite system AB can be expressed as

$$|\Psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \quad (3.3)$$

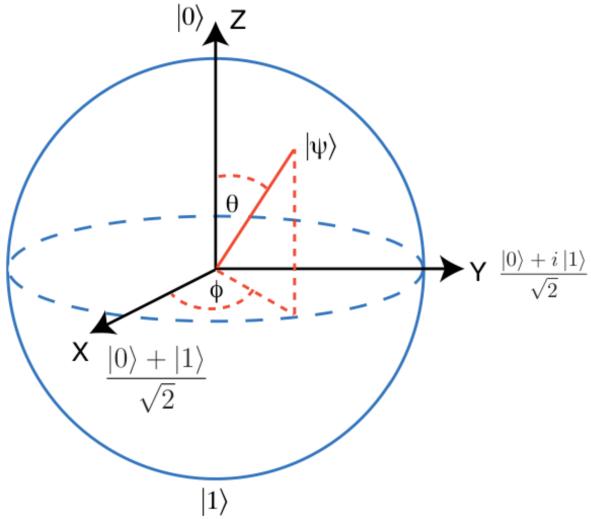


Figure 3.1: Geometrical representation of the Bloch sphere.

where α, β, γ and δ are correctly defined to maintain the normalization of the vector, and the dimension of system AB is

$$\dim(AB) = \dim(A) \cdot \dim(B) = 2 \cdot 2 = 4. \quad (3.4)$$

Again it is easy to see that for an N-qubits system the state is

$$|\Psi\rangle = \frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} |x\rangle, \quad (3.5)$$

where x is a string of binary numbers and the dimension of the system is

$$\dim(S_{1,2,\dots,N}) = \dim(S_1) \cdot \dim(S_2) \cdot \dots \cdot \dim(S_N) = 2 \cdot 2 \cdot \dots \cdot 2 = 2^N. \quad (3.6)$$

Here we described qubits as a coherent superposition of pure states. Coherence is essential for a qubit to be in such a state. If interactions, quantum noise and decoherence take place, it is possible to put the qubit in a mixed state, a statistical combination or 'incoherent mixture' of different pure states and that makes it hard to retrieve the correct computational states we were processing. Mixed states can be described with the so-called density matrix ρ defined as

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|, \quad (3.7)$$

where p_i is the probability that the system is in state $|\psi_i\rangle$. These states are represented by points inside the Bloch sphere. A mixed qubit state has three degrees of freedom: the angles ϕ and θ and the length r of the vector that represents the mixed state. It describes a real system that is evolving within a dynamical environment, which constantly interacts with the system.

This leads us to a more general definition: a 'physical' qubit is a physical device that behaves as a two-state quantum system, used as a component of a computer system.

A 'logical' qubit is a physical (error-corrected) or abstract qubit that performs as specified in a quantum algorithm or quantum circuit subject to unitary transformations and has a long enough coherence time to be usable by quantum logic gates, which we will explore later.

This is an important distinction to keep in mind when one describes quantum algorithms, because, since quantum systems are hard to control, it is also hard to keep them isolated from the environment, thus, theoretically one can work only with logical qubits, but to complete the computation it is necessary to describe also the real system, i.e. the hardware, where the algorithm can be run.

3.1.2 Quantum gates

Quantum gates are derived from the second postulate of quantum mechanics, that is: every physical quantity is described as a Hermitian operator, i.e. a linear and unitary operator, acting in the state space \mathcal{H} . This operator is an observable, meaning that its eigenvectors form a basis for \mathcal{H} .

For this reason quantum gates are invertible and reversible, whereas classic gates are irreversible unless one takes into account trash bits.

Hermitian operators belongs to the unitary group $U(2)$ and can be represented by 2×2 matrices.

There are two types of quantum gates: single-qubit gates and multiple-qubits gates. The most important single-qubit gates are:

- **Pauli gates (X , Y and Z)**

These three gates correspond to the Pauli operators and are represented with the three Pauli matrices, which define the spin observable of spin- $\frac{1}{2}$ particles.

$$X = \sigma_x = NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (3.8)$$

$$Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (3.9)$$

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (3.10)$$

The X gate is the quantum equivalent of the NOT gate for classical computers with respect to the computational basis, which distinguishes the z axis on the Bloch sphere. It is sometimes called 'bit-flip' as it swaps the two pure states,

$$X|\psi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle. \quad (3.11)$$

The eigenvectors of the Z gate are the vectors that form the computational basis,

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle. \quad (3.12)$$

The Z gate is also called 'phase-flip'.

Each gates performs a rotation of an angle π around the corresponding axis on the Bloch sphere.

- **Hadamard gate (H)**

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (3.13)$$

This is the transformation that creates a superposition of pure states,

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \quad (3.14)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle. \quad (3.15)$$

- **Rotation gates (R_x , R_y and R_z)**

The rotation operator gates are the angle rotation matrices in three cartesian axes of SO(3).

$$R_x(\theta) = e^{-iX\frac{\theta}{2}} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad (3.16)$$

$$R_y(\theta) = e^{-iY\frac{\theta}{2}} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad (3.17)$$

$$R_z(\theta) = e^{-iZ\frac{\theta}{2}} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \quad (3.18)$$

A more general form is

$$R_{\hat{n}}(\theta) = e^{-i\frac{\theta}{2}\hat{n}\cdot\vec{\sigma}} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} \hat{n}\cdot\vec{\sigma}, \quad (3.19)$$

where I is the identity operator.

- **Phase shift gates (P)**

$$P(\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}, \quad (3.20)$$

where φ is the phase shift with the period 2π .

The probability of measuring a $|0\rangle$ or $|1\rangle$ is unchanged after applying this gate, however it modifies the phase of the quantum state. This is equivalent to tracing a horizontal circle (a line of latitude), or a rotation along the z -axis on the Bloch sphere by φ radians.

Some common examples are:

$$P(\pi) = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (3.21)$$

$$P\left(\frac{\pi}{2}\right) = S = \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad (3.22)$$

$$P\left(\frac{\pi}{4}\right) = T = \sqrt{S} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{\pi}{4}} \end{pmatrix} \quad (3.23)$$

The gate performs a phase rotation in U(1) along the specified basis state.

Multiple-qubits gates transform states from multiple qubits at once, as long as the number of input qubits is equal to the number of output qubits, to preserve the unity property of quantum gates.

The most important ones are:

- **Controlled gates (CU)**

If U is a gate that operates on a single qubit with matrix representation

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}, \quad (3.24)$$

then the controlled- U gate operates on two qubits in such a way that the first qubit serves as a control and the second is the target of the operator U ,

$$CU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \quad (3.25)$$

An important example is $U = X$, this gate is also called Controlled-*NOT* or *CNOT* gate,

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (3.26)$$

it performs the *NOT* operation on the second qubit only when the first qubit is in state $|1\rangle$, and otherwise leaves it unchanged. For a comparison, it acts like a *XOR* classical gate plus a trash bit to take account of reversibility.

Another example is $U = P$

$$CP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{pmatrix}. \quad (3.27)$$

- **Swap gate**

The swap gate swaps the states of two qubits.

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.28)$$

- **Toffoli gate (TOFFOLI, CCNOT)**

$$TOFFOLI = CCNOT = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.29)$$

If the first two bits are in the state $|1\rangle$ it applies a X gate on the third bit, else it does nothing. The Toffoli gate is an example of a CCU , controlled-controlled unitary gate.

- **n-times-controlled gate ($C^n(U)$)**

This is the generalization of the controlled and the controlled-controlled gates for any unitary gate U .

$$C^n(U) = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ . & . & . & . & . \\ 0 & 0 & \dots & u_{00} & u_{01} \\ 0 & 0 & \dots & u_{10} & u_{11} \end{pmatrix} \quad (3.30)$$

Operator	Gate(s)
Pauli-X (X)	
Pauli-Y (Y)	
Pauli-Z (Z)	
Hadamard (H)	
Phase (S, P)	
$\pi/8$ (T)	
Controlled Not (CNOT, CX)	
Controlled Z (CZ)	
SWAP	
Toffoli (CCNOT, CCX, TOFF)	

Figure 3.2: Summary of the main quantum gates and their representation on a circuit.

3.1.3 Measurement

Measurement, also called observation, is an irreversible process and therefore not a quantum gate, because it assigns the observed quantum state to a single value. It takes a quantum state and projects it to one of the basis vectors with a likelihood equal to the square of the vector's length along that basis vector. This is known as the 'Born rule' or the third postulate of quantum mechanics and appears as a stochastic non-reversible operation as it probabilistically sets the quantum state equal to the basis vector that represents the measured state. At the instant of measurement, the state is said to 'collapse' to the definite single value that was measured. Why and how, or even if the quantum state collapses at measurement, is called the measurement problem.

Measuring a single qubit, whose quantum state is represented by the vector in eq. (3.1), will result in $|0\rangle$ with probability $|\alpha|^2$, and in $|1\rangle$ with probability $|\beta|^2$.

If we take a quantum state $|\Psi\rangle$ that spans a register, i.e. a collection of N qubits, this can be measured to 2^N distinct states, similar to how a register of n classical bits can hold 2^n distinct states. Unlike with the bits of classical computers, quantum states can have non-zero probability amplitudes in multiple measurable values simultaneously. This is called superposition.

As mentioned before, the sum of all probabilities for all outcomes must always be equal to 1.

We can formally define the measurement operation as the application of the operator M_m

$$|\psi\rangle \rightarrow \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}, \quad (3.31)$$

and the probability of finding the eigenvalue a_m is

$$\mathbb{P}(a_m) = |\langle a_m|\psi\rangle|^2. \quad (3.32)$$



Figure 3.3: Representation of the measurement process in a circuit.

Generally the computational basis $\{|0\rangle, |1\rangle\}$ is used to project the states of the qubits, but different basis can be used, such as the basis $\{|+\rangle, |-\rangle\}$.

3.2 Important results

3.2.1 Set of universal quantum gates

A small set of classical gates, e.g. $AND + OR + NOT$ or $NAND + FANOUT$, can be used to compute an arbitrary classical function. We say that such a set of gates

is universal for classical computation.

Similarly in quantum computing, a set of universal quantum gates is any set of gates to which any possible operation on a quantum computer can be reduced, that is, any other unitary operation can be expressed as a finite sequence of gates from the set. Technically, this is impossible with anything less than an uncountable set of gates since the number of possible quantum gates is uncountable, whereas the number of finite sequences from a finite set is countable. To solve this problem, we only require that any quantum operation can be approximated by a sequence of gates from this finite set.

The rotation operators R_x , R_y , R_z , the phase shift P and the $CNOT$ gates form a universal set of quantum gates.

A common universal gate set is the Clifford + T gates set, which is composed of the $CNOT$, H , S and T gates.

The last statement can be demonstrated rigorously by decomposing generic unitary gates into single-qubit gates and $CNOT$ gates and then by approximating single-qubit gates with products of H and T gates. The P gates are needed to make the approximation fault-tolerantly, i.e. during the computation the states are encoded in a way that prevent interactions, quantum noise and decoherence to produce errors and lose information [35].

This knowledge allows us to decompose every circuit in a simpler one that contains only this set of gates. This is useful especially for highly-correlated circuits, i.e. circuits with many multiple-qubits gates, that are complicated to implement due to the difficulty of controlling interactions between individual qubits with high precision.

A simple example is the decomposition of the X and Z gates in a sequence of H and S gates,

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = SS, \quad (3.33)$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = HZH = HSSH. \quad (3.34)$$

Unfortunately, the decomposition of a generic unitary transformation on N qubits through a small set of elementary gates is generally hard, meaning that there is no efficient way to do it [33]. To see this, suppose we have g different types of gates available, and each gate works on at most f input qubits. These numbers, f and g , are fixed by the computing hardware we have available, and may be considered to be constants. Suppose we have a quantum circuit containing m gates, starting from the computational basis state $|0\rangle^{\otimes N}$. For any particular gate in the circuit there are therefore at most $\binom{N}{f}^g = O(N^{fgm})$ possible choices. It follows that at most $O(N^{fgm})$ different states may be computed using m gates, meaning there are unitary operations requiring exponentially many operations.

Thus, to within a polynomial factor the construction for universality is optimal. Unfortunately, it does not address the problem of determining which families of unitary operations can be computed efficiently in the quantum circuits model.

3.2.2 Gottesman-Knill theorem

A very important result in quantum information is the Gottesman-Knill theorem. It provides a connection between quantum and classical computing in terms of efficient computation, which, as we already saw, is a fundamental topic throughout all of computer science.

Generally we say that a quantum computation can be efficiently simulated classically in the 'strong' sense, if it is possible to evaluate the function $\alpha \in \{0, 1\}^S \rightarrow \pi(\alpha)$ up to M digits in $\text{poly}(M, N)$ time on a classical computer, where N is the number of qubits, α is the measurement outcome bit string, S is the subset of qubits measured and $\pi(\alpha)$ is the probability that the outcome α occurs, given by

$$\pi(\alpha) = \langle 0 |^{\otimes N} U^\dagger | \alpha \rangle \langle \alpha |^S U | 0 \rangle^{\otimes N}. \quad (3.35)$$

Furthermore, we say that the quantum computation can be efficiently simulated classically in the 'weak' sense if it is possible to sample once from the probability distribution $\{\pi(\alpha)\}$ in $\text{poly}(N)$ time on a classical computer.

The Gottesman-Knill theorem states:

Every uniform family of Clifford circuits, when applied to the input state $|0\rangle^{\otimes N}$ and when followed by a measurement of an arbitrary subset of qubits in the computational basis, can be efficiently simulated classically in the strong sense.

This can be proved using stabilizer operations, i.e. operations that leave the state unchanged. One can build a generic state by using only Clifford gates, then decompose them in Pauli gates and show that they are stabilizer operations for this generic state. Using this stabilizer description it can then be shown that the measurement outcomes of computational basis measurements can efficiently be computed on a classical computer, hence showing that Clifford circuits can be efficiently simulated classically in the strong sense.

The theorem leads to the conclusion that any quantum circuit which is composed of only H , P and $CNOT$ gates can not provide any speedup with respect to classical computation.

This result exhibits some rather remarkable and sometimes puzzling features, not all of which are fully understood. For example, even though they are efficiently classically simulatable, Clifford circuits can generate a high degree of entanglement. This very feature raises doubts about the often-quoted statement that "entanglement is responsible for the quantum computational speedup". In particular, it highlights that, while the presence of certain types of entanglement in a quantum computation is provably necessary to disallow efficient classical simulation, it is certainly not sufficient [34].

Nevertheless, supplementing Clifford operations with essentially any non-Clifford gate immediately yields the full quantum computational model, as we have shown before.

3.2.3 Quantum computational complexity

There is considerable interest in developing a theory of quantum computational complexity, and relating it to classical computational complexity theory, which will doubtlessly represent an enormously fruitful direction for future researchers. Here we present an introductory discussion.

In classical computation we can define the **PSPACE** as the class of decision problems which can be solved on a Turing machine using space polynomial in the problem size and an arbitrary amount of time.

BQP is an essentially quantum complexity class consisting of those decision problems that can be solved with bounded probability of error using a polynomial size quantum circuit.

One of the most significant results in quantum computational complexity is that **BQP** \subseteq **PSPACE**. It is clear that **BPP** \subseteq **BQP**, where **BPP** is the classical complexity class of decision problems which can be solved with bounded probability of error using polynomial time on a classical Turing machine. Thus we have the chain of inclusions **BPP** \subseteq **BQP** \subseteq **PSPACE**. Proving that **BQP** \neq **BPP**, i.e. quantum computers are more powerful than classical computers, will therefore imply that **BPP** \neq **PSPACE**. However, it is not presently known whether this is true and proving it would represent a major breakthrough in computer science.

To prove that **BQP** \subseteq **PSPACE** we do a computation involving $\text{poly}(n)$ gates and calculate the probability of ending up in a generic state. Given that the individual unitary gates appearing are operations such as H , $CNOT$ and so on, it is clear that the individual probabilities can be calculated to high accuracy using only polynomial space on a classical computer, and thus the total probability can be calculated using polynomial space. Of course, this algorithm is rather slow, since there are exponentially many terms which need to be calculated. However, only polynomially much space is consumed, and thus **BQP** \subseteq **PSPACE** [35].

Therefore, the class of problems solvable on a quantum computer with unlimited time and space resources is no larger than the class of problems solvable on a classical computer. Stated another way, this means that quantum computers do not violate the Church–Turing thesis that any algorithmic process can be simulated efficiently using a Turing machine. Of course, quantum computers may be much more efficient than their classical counterparts, thereby challenging the strong Church–Turing thesis that any algorithmic process can be simulated efficiently using a probabilistic Turing machine.

In the context of quantum simulation for quantum chemistry the simulation of Hamiltonian dynamics resides in the **BQP** class, while ‘hard’ problems can be found in the quantum Merlin-Arthur class (**QMA**). The latter is defined similarly to the **NP** class for classical computation, i.e. it comprises the problems where putative solutions can be verified but not computed in polynomial time on a quantum computer. Producing a putative solution means executing a quantum circuit giving access to a wave function $|\Psi\rangle$ and verifying a putative solution means executing a second quantum circuit to ensure that $|\Psi\rangle$ is actually a solution of the problem of interest. In the context of quantum simulation for quantum chemistry the most important **QMA** problem is the simulation of Hamiltonian eigenstates. However,

there are a number of practical considerations to take into account. For example, the statements of computational complexity theory refer to exact solutions of the problem at hand, and experience indicates that approximate methods can deliver accurate results for certain problems, which calls for a systematic characterization of quantum algorithms for chemistry in both accuracy and computational cost across a variety of chemical problems [33].

3.2.4 Quantum Fourier transform

Moving on to quantum algorithms we show one of the most important quantum routines used by many important algorithms, like Shor's algorithm and quantum phase estimation: the quantum Fourier transform (QFT).

It is the analogous to the discrete Fourier transform. It is a linear transformation with the following action on the basis states:

$$|x\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |y\rangle. \quad (3.36)$$

Equivalently, the action on an arbitrary state may be written as

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle, \quad (3.37)$$

where the amplitudes y_k are the discrete Fourier transform of the amplitudes x_j . Since it is a linear transformation, the QFT can be implemented as the dynamics for a quantum computer.

It is helpful to write the state $|x\rangle$ using the binary representation

$$|x\rangle = x_1 x_2 \dots x_n = x_1 2^{n-1} + x_2 2^{n-2} + \dots + x_n 2^0 \quad (3.38)$$

and to write a binary fraction as

$$\frac{x_l}{2} + \frac{x_{l+1}}{4} + \dots + \frac{x_m}{2^{m-l+1}} = 0.x_l x_{l+1} \dots x_m. \quad (3.39)$$

With a little algebra the QFT can be given the following useful 'product representation':

$$|x_1, \dots, x_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.x_n})(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}) \dots (|0\rangle + e^{2\pi i 0.x_1x_2\dots x_n})}{2^{n/2}}. \quad (3.40)$$

This representation makes it easy to derive an efficient circuit for the quantum Fourier transform. Such a circuit is shown in Figure 3.4. Swap gates at the end of the circuit are used to reverse the order of the qubits. The gate $UROT_k$ denotes the unitary transformation

$$UROT_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{pmatrix}, \quad (3.41)$$

which is a phase gate where φ is the primitive 2^k -th of unity. In the circuit they appear in the controlled form.

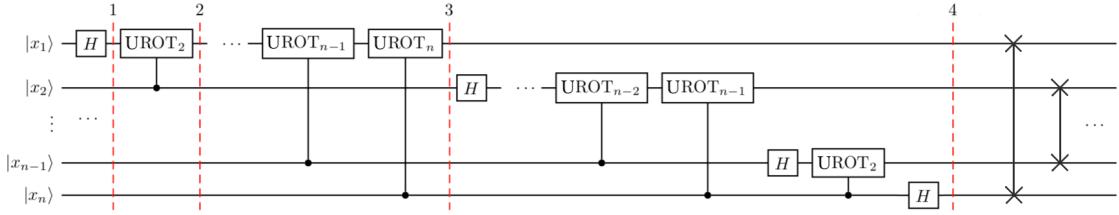


Figure 3.4: Circuit for the quantum Fourier transform.

This construction also proves that the quantum Fourier transform is unitary, since each gate in the circuit is unitary.

How many gates does this circuit use? We start by doing a Hadamard gate and $n - 1$ conditional rotations on the first qubit, a total of n gates. This is followed by a Hadamard gate and $n - 2$ conditional rotations on the second qubit, for a total of $n + (n - 1) + \dots + 1 = n(n + 1)/2$ gates are required, plus the gates involved in the swaps. At most $n/2$ swaps are required, and each swap can be accomplished using three controlled-gates. Therefore, this circuit provides a $O(n^2)$ scaling algorithm for performing the QFT.

In contrast, the best classical algorithms for computing the discrete Fourier transform on $2n$ elements are algorithms such as the fast Fourier transform (FFT), which compute the discrete Fourier transform using $O(n2^n)$ gates. That is, it requires exponentially more operations to compute the Fourier transform on a classical computer than it does to implement the QFT on a quantum computer.

At face value this is an outstanding result, since the Fourier transform is a crucial step in so many real-world data processing applications. Can we use the QFT to speed up the computation of these Fourier transforms? Unfortunately, there is no known way to do this. The problem is that the amplitudes in a quantum computer cannot be directly accessed by measurement. Thus, there is no way of determining the Fourier transformed amplitudes of the original state. Worse still, there is in general no way to efficiently prepare the original state to be Fourier transformed. Thus, finding uses for the quantum Fourier transform turns out to be very subtle [35].

3.3 Quantum devices

Experimental realization of quantum circuits, algorithms, and communication systems has proven extremely challenging. Here we explore some of the guiding principles and systems for physical implementation of quantum information processing devices and systems.

A quantum computer has to be well isolated in order to retain its quantum properties, but at the same time its qubits have to be accessible so that they can be manipulated to perform a computation and to read out the results. A realistic implementation must strike a delicate balance between these constraints, so that the relevant question is not how to build a quantum computer, but rather, how good a

quantum computer can be built.

First of all, we must use systems with a two-state configuration to represent qubits, secondly, we must select a system in which they can be made to evolve as desired. Furthermore, we must be able to prepare qubits in some specified set of initial states and to measure the final output state of the system.

- **Configurations**

For the purpose of computation, the crucial realization is that the set of accessible states should be finite. It is generally desirable to have some aspect of symmetry dictate the finiteness of the state space, in order to minimize decoherence. For example, a spin-1/2 particle lives in a Hilbert space spanned by the $| \uparrow \rangle$ and $| \downarrow \rangle$ states, the spin state cannot be anything outside this two-dimensional space, and thus is a nearly ideal quantum bit when well isolated. A particle in a finite square well would make a mediocre quantum bit, because transitions from the bound states to the continuum of unbound states would be possible. These would lead to decoherence since they could destroy qubit superposition states.

For single qubits the figure of merit is the minimum lifetime of arbitrary superposition states. A good measure, used for spin states and atomic systems, is T_2 , the ‘transverse’ relaxation time of superposition states such as $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$, while T_1 is the ‘longitudinal’ relaxation time of the higher energy $|1\rangle$ state, i.e. the classical state lifetime, which is usually longer than T_2 .

- **Controllability**

To perform quantum computation one must be able to control the Hamiltonian in order to effect an arbitrary selection from a universal family of unitary transformations. Unrecorded imperfections in unitary transforms can lead to decoherence. Similarly, the cumulative effect of systematic errors is decoherence when the information needed to be able to reverse them is lost.

Two important figures of merit for unitary transforms are: the minimum achievable fidelity, \mathcal{F} , and the maximum time, t_{op} , required to perform elementary operations such as single spin rotations or a *CNOT* gate.

- **State preparation**

One of the most important requirements for being able to perform a useful computation, even classically, is to be able to prepare the desired input. Generally a computation starts with N qubits in the state $|00\dots0\rangle$, but they may not stay there for very long due to thermal heating. Simply letting the system equilibrate establishes it in the thermal state, with the density matrix $\rho \approx e^{\mathcal{H}/k_b T} / \mathcal{Z}$.

Two figures of merit are relevant to input state preparation are: the minimum fidelity with which the initial state can be prepared in a given state ρ_{in} and the entropy of ρ_{in} , the latter is important because the input state is generally a pure state, with zero entropy.

- **Measurement**

Let us think of measurement as a process of coupling one or more qubits to a

classical system such that after some interval of time, the state of the qubits is indicated by the state of the classical system. Many difficulties with measurement can be imagined. Furthermore, projective measurements, sometimes called ‘strong’ measurements, are often difficult to implement. They require that the coupling between the quantum and classical systems be large, and switchable. Measurements should not occur when not desired; otherwise they can be a decoherence process. Surprisingly, however, strong measurements are not necessary; weak measurements which are performed continuously and never switched off, are usable for quantum computation. This is made possible by completing the computation in time short compared with the measurement coupling.

A good figure of merit for measurement capability is the signal to noise ratio (SNR).

In the following we describe the main physical systems used to build quantum processor.

In Figure 3.5 are represented the most researched and mature qubit technologies.

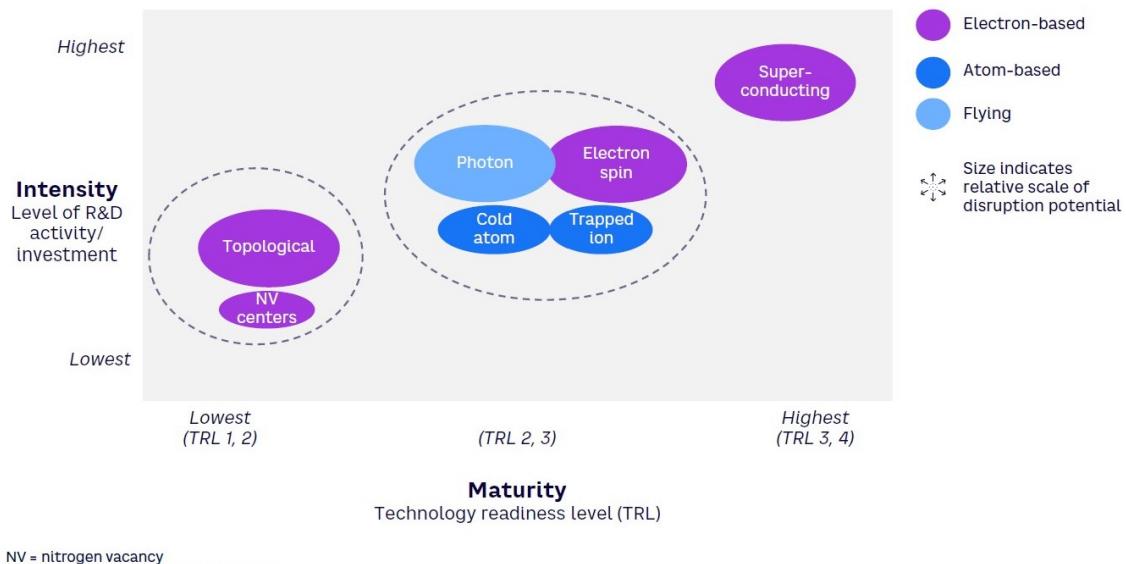


Figure 3.5: Core qubit technologies mapped by maturity, intensity and disruption potential [39].

3.3.1 Optical photons

An attractive physical system for representing a quantum bit is the optical photon. It can be guided along long distances with low loss in optical fibers, delayed efficiently using phase shifters and combined using beamsplitters.

There are different ways to encode information in photonic qubits, for example path or polarization. The latter is defined as

$$|0\rangle = |H\rangle, \quad |1\rangle = |V\rangle, \quad (3.42)$$

where $|H\rangle$ denotes horizontal polarization and $|V\rangle$ denotes vertical polarization. Experimentally, the polarization of light can be manipulated with phase retarders or ‘wave plates’. These uniaxial, birefringent crystals introduce a polarization-dependent phase shift. Combining multiple wave plates facilitates the realization of every possible single-qubit gate on polarization-encoded qubits. Polarizing beam splitters can be used for the analysis of a polarization state. Combined with half-wave plates and quarter-wave plates they facilitate measurements in each possible direction on the Bloch sphere.

Unfortunately, interaction between photons is difficult to achieve. In fact, because a nonlinear index of refraction is usually obtained by using a medium near an optical resonance, there is always some absorption associated with the nonlinearity. The standard process of creating entangled photons works probabilistically and with low efficiency and the quality of multi-photon states generated in a process are intrinsically limited due to noise caused by higher-order emissions.

Secondly, the development of efficient photonic two-qubits gates is of great importance. Although it has been shown that quantum computing is possible with only linear optics and photon detection, in practice these schemes become inefficient due to an enormous amount of required ancilla photons.

Thirdly, current experiments are limited by the low detection efficiencies of avalanche photodiodes which are widely used in photonic quantum computing experiments. Much higher detection efficiencies can be obtained by employing superconducting transition-edge detectors or superconducting nanowire detectors. While poor time resolution of the former impedes their application in pulsed multi-photon experiments, the latter are ideally suited for this task [1].

3.3.2 Trapped ions

Trapped ions are one approach to a large-scale quantum computer in which ions are confined and suspended in free space using electromagnetic fields. To achieve quantum computation with these, ion qubits are stored in the stable electronic states of each ion and quantum information is transferred through collective quantized motion of the ions in a shared trap. The manipulations made to these qubits are performed by lasers, where coupling is induced between qubit states for single qubit operations and between internal qubit states and external motional states for entanglement.

One can trap ions by using an apparatus known as a magneto-optical trap, or MOT for short. Magneto-optical traps rely on the principle of Doppler cooling. This is a mechanism that can be used to trap and slow the motion of atoms to cool a substance by operating on the principle of the Doppler shift experienced by the ion when hit by a laser photon.

To form a qubit one can produce hyperfine qubits, where the two states are two ground state hyperfine levels, or produce optical qubits, in which the two states are the ground state level and the excited level. Hyperfine qubits are quite long-lived, with experimental lifetimes often exceeding 10 minutes, and optical qubits, although shorter-lived, have lifetimes on the order of seconds, which is still long compared

with logic gate operation time, which is on the order of microseconds for both types of qubits.

We can prepare the qubits by optical pumping, in which a laser couples the ion to some excited states that can decay to one state which is not coupled to the laser. Eventually, after being excited enough times, the ion reaches a state in which it has no excited levels to couple to in the presence of that laser, and thus, the ion will remain in this state. After enough time essentially all of the ions will be in this state and therefore the qubits will be prepared. This preparation has extremely high fidelity, exceeding 99.9%.

To make a measurement we apply to the ion a laser that couples only one qubit state. We now have two cases for how the ion will collapse upon measurement: if the ion collapses to the state to which the laser is coupled then the laser excites the ion, resulting in the emission of a photon when the ion decays from the excited state, otherwise, if the ion collapses to the state to which the laser is not coupled then the laser does not excite the ion, and thus, no photons will be released. We can subsequently count the number of photons and determine the state of the qubit and, moreover, we can do so to very high accuracy, again exceeding 99.9%.

Regarding transformations, to implement single-qubit rotations for hyperfine qubits we can use magnetic dipole transitions or stimulated Raman transitions and for optical qubits we can achieve them via electric quadrupole transitions. For *CNOT* gates there are various implementation, for example using *SWAP* gates as well as controlled *Z* gates. Such gates are generally no faster than one microsecond. These gates have a high fidelity, exceeding 99% for each type of qubit.

However, trapped-ion quantum computation is limited by the finite number of qubits that can be stored in each trap. Therefore, to achieve scalability, schemes have been introduced in which there are interconnected ion traps with storage and interaction regions. The qubits can thus be stored in the storage regions until they are needed for computation, at which point they can be shuttled to the interaction regions for manipulation. These qubits can turn corners at T junctions, which allows for a two-dimensional array trap design. Moreover, in the modern era of semiconductor fabrication, techniques have been employed to manufacture the ‘ion trap on a chip’, demonstrating the potential for scalability [9].

3.3.3 Superconducting circuits

It is also possible to build quantum computers from superconducting electronic circuits. This research is conducted by companies such as Google and IBM, e.g. the quantum processor Sycamore, designed by Google, uses 53 superconducting qubits. In a superconductor the basic charge carriers are pairs of electrons, known as Cooper pairs, rather than the single electrons in a normal conductor. The total spin of a Cooper pair is an integer number, thus the Cooper pairs are bosons, i.e. electrons condense into Cooper pairs that form a single superfluid.

Superconducting qubits realize artificial atoms with engineered energy levels using the non-linearity of Josephson junctions and surrounding microwave circuitry. A

Josephson junction consists of a thin layer of aluminium oxide, which is an insulator, sandwiched between two superconducting layers of aluminium, which becomes a superconductor when cooled below 1.2 K. The insulating layer is so thin (a few nanometres) that Cooper pairs can tunnel through it and couple the superconducting wave functions on either side of the barrier.

The quantum dynamics of superconducting qubits follows that of a damped and driven anharmonic oscillator whose anharmonicity is controlled by choice of circuit parameters, e.g. linear capacitance and inductance of the Josephson junction. For experimental design and control practitioners draw from the Jaynes-Cummings model and its variants from cavity quantum electrodynamics (QED). Circuit quantum electrodynamics borrows the application of second quantized Hamiltonians from atomic optics via a standardized procedure for quantizing passive circuit.

The operational modes of superconducting qubits vary by their energy spectra, where non-linearity plays a role in realizing accessible and isolated states. If we consider the lowest two levels of the quantum harmonic oscillator to be the ground and excited states of a qubit, $|g\rangle$ and $|e\rangle$, the energies for the two states are separated by integer multiples of $\hbar\omega$. When we use an anharmonic oscillator the spacing between these two states is larger, further isolating the qubit states from the other states of the oscillator. A qubit built upon this system is also called a transmon.

An anharmonic oscillator-based superconducting qubit inherits its non-linearity from Josephson junctions, where the non-linearity is tunable through fabrication and microwave circuit design. This is because there are several phenomenological models for Josephson junctions.

The behaviour of the electron superfluid is completely determined by a single quantum wave function. The amplitude of this wave function determines the number of Cooper pairs, while the value of the phase is related to the supercurrent and any magnetic field that is present.

The two primary types of superconducting qubit, the charge qubit and the flux qubit, are directly related to two variables: charge qubits are associated with the amplitude, while flux qubits are related to the phase.

For a flux qubit a standard circuit is composed by three Josephson junctions connected by superconducting aluminium leads to form a closed loop and uses an applied magnetic field, perpendicular to the loop, to drive the circuit by controlling the phase. When the magnetic flux through the loop, ϕ , is equal to half the quantum of magnetic flux in a superconductor, the state with a zero phase difference around the loop ($|0\rangle$) has the same energy as the state with a 2π phase difference ($|1\rangle$). One of these states corresponds to a current circulating around the loop in the clockwise direction, while the other state corresponds to a current moving in the opposite direction [30].

Measurements are made with a superconducting quantum interference device (SQUID). This device, which consists of two Josephson junctions in parallel, is the most sensitive magnetic-flux detector known.

If we shine a coherent state light with frequency ω_a , i.e. the transmon frequency gap, on the cavity and phase ϕ at the position of the qubit, then the Hamiltonian

for the artificial atom is

$$H = \hbar\Omega_R(\sigma_x \cos\phi + \sigma_y \sin\phi), \quad (3.43)$$

where Ω_R is called the Rabi frequency, which depends on the average electric field in the cavity and the size of the superconducting qubit. With this Hamiltonian we can implement a universal set of single-qubit gates since $\phi = 0$ implements an X -rotation and $\phi = \frac{\pi}{2}$ applies a Y -rotation.

As for multiple-qubits gates, an advantage of superconducting technology is the variety of options to connect qubits with each other. One of the best ways is via capacitative coupling, where we connect two transmons through a wire and coupling capacitor. We can form a Hamiltonian that allows to implement the two-qubits *iSWAP* gate and if we combine this with X - and Z -rotations we can build a *CNOT* gate, thus enabling universal quantum computation.

Nonetheless, there are still some technological challenges preventing superconducting qubits from scaling further. A glaring issue is that we need to keep the transmons at very low temperatures, which requires the use of large cryogenic devices known as dilution refrigerators. In the future other quantum technologies may bypass the use of low temperatures, but superconducting qubits may not be so lucky, since they are constrained by the laws of physics that allow for superconductivity.

Superconducting circuits are large compared to other quantum systems, so their interaction with the environment is difficult to control. As a consequence, the excited state lasts for about 1 micro-second before decaying back to the ground state. Current single-qubit gates are acceptable since they can be applied in a matter of nanoseconds, thanks to our ability to manufacture very small cavities. Problems arise, however, when we want to perform precise measurements of the qubit [40].

In Figures 3.6 and 3.7 are listed the main advantages and the disadvantages of trapped ions and superconducting qubits.

Trapped Ion	Superconducting Qubit
Long coherence	Compatible with existing fabrication
All-to-all connectivity	Easy coupling
Mature technology	Mature technology
Qubits are all identical	Qubits are easy to reproduce

Figure 3.6: Advantages of trapped ions and superconducting qubits.

Trapped Ion	Superconducting Qubit
Difficult to scale lasers	Limited gate connectivity
Slow gate speed	Short coherence
	Near absolute zero cooling

Figure 3.7: Disadvantages of trapped ions and superconducting qubits.

Chapter 4

Quantum computing for computational chemistry: FTQC devices

4.1 Fault-tolerant quantum computation devices

One of the most powerful applications of quantum error correction is not merely the protection of stored or transmitted quantum information, but the protection of quantum information as it dynamically undergoes computation.

Using logical qubits and auxiliary or 'ancilla' qubits, one can build circuits that work even if there is an error during the process. Specifically, if the probability that an error occurs is below a certain threshold one can make a series of encodings on the physical qubits and the circuit will return the correct result with a mistake probability lower than the threshold.

This process is called 'fault-tolerant' quantum computing and it is generally very delicate to perform since many qubits are needed and the threshold is generally hard to achieve.

The basic idea of fault-tolerant quantum computing is to compute directly on encoded quantum states in such a manner that decoding is never required.

Unfortunately, noise afflicts each of the elements used to build this circuit: state preparation procedures, quantum logic gates, measurement of the output, and even the simple transmission of quantum information along the quantum wires.

To combat the effect of this noise one can replace each qubit in the original circuit with an encoded block of qubits, using an error-correcting code, e.g. bit-flip code, sign-flip code or Shor code, and replace each gate in the original circuit with a procedure that performs an encoded gate on the encoded state. By performing error correction periodically on the encoded state we prevent accumulation of errors in the state.

An important condition is that errors do not propagate throughout the circuit, i.e. from qubit to qubit. For example, one can see that if a *CNOT* gate is applied and the control qubit is affected by an error the resulting state after the gate show that both qubits are affected by errors. Obviously, for an error to propagate there must

be interaction. Encoded gates should therefore be designed very carefully so that a failure anywhere during the procedure for performing the encoded gate can only propagate to a small number of qubits in each block of the encoded data, so that error correction is effective at removing the errors.

We define the fault-tolerance of a procedure to be the property that if only one component in the procedure fails then the failure causes at most one error in each encoded block of the output qubits. For example, the failure of a single component, in a fault-tolerant recovery procedure for quantum error correction, results in the recovery procedure being performed correctly, up to an error on a single qubit of the output. By ‘component’ we mean any of the elementary operations used in the encoded gate, which might include noisy gates, noisy measurements, noisy quantum wires and noisy state preparations.

The fundamental theorem for a fault-tolerant procedure states that:

If P is the probability of an error to occur for every component of the procedure, then the probability of introducing two or more errors in the first encoded qubit is CP^2 .

Here C is a constant that depends on the topology of the procedure, i.e. on the components that can be affected by an error, and it is typically a power of 10 [35]. Thus we have an advantage if

$$P < \frac{1}{C} = P_{\text{threshold}}. \quad (4.1)$$

This is known as the ‘threshold condition’ for quantum computation, since provided it is satisfied we can achieve arbitrary accuracy in our quantum computations.

The error can be further reduced by making a “cascade” of encodings, e.g. using Shor code we take 9 physical qubits and produce 1 logical qubit, using the code again we take 9 logical qubits, i.e. 81 physical qubits, and produce 1 logical qubit. The likelihood of failure at each level becomes

$$P \rightarrow CP^2 \rightarrow C(CP^2)^2 = C^3 P^4 \rightarrow \dots \rightarrow \frac{(CP)^{2^k}}{C} \quad (4.2)$$

Thus, if we concatenate k times, the failure probability for a procedure at the highest level is $(CP)^{2^k}/C$, while the size of the simulating circuit scales as d^k times the size of the original circuit, where d is a constant representing the maximum number of operations used in a fault-tolerant procedure to do an encoded gate and error correction. Suppose that we wish to execute a circuit containing $\text{poly}(n)$ gates, where n specifies the size of some problem, and that we wish to achieve a final accuracy of ϵ in our execution of this algorithm. To do so the performance of each gate in the algorithm must be accurate to $\epsilon/\text{poly}(n)$, so we must concatenate a number of times k such that

$$\frac{(CP)^{2^k}}{C} \leq \frac{\epsilon}{\text{poly}(n)}. \quad (4.3)$$

How large a simulating circuit is required to achieve this level of accuracy? We have

$$d^k = \left(\frac{\log(\text{poly}(n)/C\epsilon)}{\log(1/PC)} \right) = O(\text{poly}(\log(\text{poly}(n)/\epsilon))), \quad (4.4)$$

from which derives the threshold theorem for quantum computation:

A quantum circuit containing $\text{poly}(n)$ gates may be simulated with probability of error at most ϵ using $O(\text{poly}(\log(\text{poly}(n)/\epsilon)))$ gates on hardware whose components fail with probability at most P , provided P is below some constant threshold, $P \leq P_{\text{threshold}}$, and given reasonable assumptions about the noise in the underlying hardware.

Sophisticated calculations for the threshold have typically yielded values in the range $10^{-5} - 10^{-6}$.

We will not investigate fault-tolerance here, but this is the setup on which we describe the computational results of our algorithms in this chapter.

Because of the threshold theorem we know that, provided the noise in individual quantum gates is below a certain constant threshold, it is possible to efficiently perform an arbitrarily large quantum computation.

Nevertheless, as we said, this is a delicate process for a number of reasons. First, the threshold result requires a high degree of parallelism in our circuits. Second, depending on the error correction code used one may need to use many physical qubits with high connectivity, and that is still a big engineering challenge. The research group at Google, in its roadmap, estimated 1000 physical qubits would be necessary to encode a single logical qubit that maintains coherence [28]. Related to this, today's quantum components have a high failure likelihood, namely around $10^{-3} - 10^{-2}$ [21], which is far from the required value of $P_{\text{threshold}}$.

To end this section on the requirements needed for a error-free universal quantum computer we show a more concrete set of conditions for the purpose, i.e. DiVincenzo's criteria, proposed in 2000 by the theoretical physicist David P. DiVincenzo. According to DiVincenzo's criteria constructing a quantum computer requires that the experimental setup meet seven conditions. The first five are necessary for quantum computation:

1. **A scalable physical system with well-characterized qubit;**
2. **The ability to initialize the state of the qubits to a simple fiducial state;**
3. **Long relevant decoherence times;**
4. **A universal set of quantum gates;**
5. **A qubit-specific measurement capability.**

The remaining two are necessary for quantum communication:

1. **The ability to interconvert stationary and flying qubits;**
2. **The ability to faithfully transmit flying qubits between specified locations.**

Here 'stationary' qubits indicate qubits that are able to store quantum information reliably on a timescale of $\sim ms$ and 'flying' qubits are qubits that can be send over macroscopic distances while keeping their encoded information intact.

We analyze technological aspects more deeply in Chapter 6, but this is a big picture of what the research in this field is facing.

4.2 Quantum phase estimation

The first algorithm that we analyze is the quantum phase estimation (QPE) algorithm. This is the most important algorithm to simulate a quantum system on a quantum computer, since it has a very high precision and its form derives directly from Trotter decomposition formula, used for Lloyd's description of unitary evolution [27].

Suppose a unitary operator U has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$, where the value of φ is unknown. The goal of the phase estimation algorithm is to estimate φ . To perform the estimation we assume that we have available black boxes, also known as oracles, capable of preparing the state $|u\rangle$ and performing the controlled- U^{2^j} operation for suitable non-negative integers j .

The QPE procedure uses two registers. The first register, also called the ancilla register, contains ω qubits initially in the state $|0\rangle$. How we choose ω depends on two things: the number of digits of accuracy we want to have in our estimate for φ and how likely we want the phase estimation procedure to be successful. The second register begins in the state $|u\rangle$ and contains as many qubits as is necessary to store this state.

In the case of quantum simulation $|u\rangle$ is the wave function $|\Psi\rangle$ of the system and φ is the energy value we aim to find. It can be the energy of the lowest eigenstate, E_0 , or the energy of the excited states, $E_{i>0}$, of a physical Hamiltonian. This means that the operator U satisfies the condition

$$U|\chi_i\rangle = e^{2\pi i E_i} |\chi_i\rangle, \quad (4.5)$$

where $|\chi_i\rangle$ are the orbital functions of the system, defined by the basis set functions. For example, the Hartree-Fock wave function can be used as a first approximation of the real wave function.

In Chapter 5 we will describe the possible mappings between the fermionic Hamiltonian and the qubit Hamiltonian, here we focus mainly on the core part of the algorithm. For a brief introduction, a mapping is a function that transforms the many-body Hamiltonian into the qubit Hamiltonian, this means that the second quantization operators are transformed into Pauli operators. Thus, the qubit Hamiltonian can be written as a linear combination of Pauli strings,

$$H = \sum_i w_i P_i, \quad (4.6)$$

where each P_i is a Pauli operator and w_i its corresponding coefficient.

A general scheme of the algorithm is:

1. Initialize the qubit register in state $|\Psi\rangle$

This state must have nonzero overlap with the true FCI target wave function of the system. We can expand the wave function $|\Psi\rangle$ in terms of energy eigenstates of the Hamiltonian, i.e. $|\Psi\rangle = \sum_i c_i |\chi_i\rangle$, where c_i are complex coefficients;

2. Apply a Hadamard gate to each qubit of the first register

This places them in a superposition of all the possible states of the computational basis,

$$|0\rangle \sum_i c_i |\chi_i\rangle \xrightarrow{H^{\otimes \omega}} \frac{1}{\sqrt{2^\omega}} \sum_x |x\rangle \sum_i c_i |\chi_i\rangle, \quad (4.7)$$

where x are all possible bit strings that can be constructed from ω bits;

3. Apply the controlled-U gates

The first register is the control and the second register is the target,

$$\frac{1}{\sqrt{2^\omega}} \sum_x |x\rangle \sum_i c_i |\chi_i\rangle \xrightarrow{C-U^{2^j}} \frac{1}{\sqrt{2^\omega}} \sum_i \sum_x e^{2\pi i E_i} c_i |x\rangle |\chi_i\rangle; \quad (4.8)$$

4. Apply the inverse quantum Fourier transform to the first register

This allows us to write the state of the qubits with a basis that brings the information of the energy levels of the system defined in binary numbers,

$$\frac{1}{\sqrt{2^\omega}} \sum_i \sum_x e^{2\pi i E_i} c_i |x\rangle |\chi_i\rangle \xrightarrow{QFT^{-1}} \sum_i c_i |bin(E_i)\rangle |\chi_i\rangle; \quad (4.9)$$

5. Finally, measure the first register in the Z basis

This results in an estimate of the energy eigenvalues as a bit string E_i with probability $|c_i|^2$. This procedure collapses the main register into the corresponding eigenstate $|\chi_i\rangle$,

$$\sum_i c_i |bin(E_i)\rangle |\chi_i\rangle \xrightarrow{Z} bin(E_i), \quad \mathbb{P}(\chi_i) = |c_i|^2. \quad (4.10)$$

A circuital scheme of this algorithm is shown in Figure 4.1.

In the following we analyze the two main components of the QPE method: the state preparation and the implementation of the controlled- U^{2^j} operation.

4.2.1 State preparation

Initializing the qubit register in a state which has a sufficiently large overlap with the target eigenstate, typically the ground state, is a nontrivial problem. This is important because a randomly chosen state would have an exponentially vanishing probability of collapsing to the desired ground state as the system size increases. This highlights the necessity of developing state preparation routines which result in, at worst, a polynomially decreasing overlap with the FCI ground state as the system size increases.

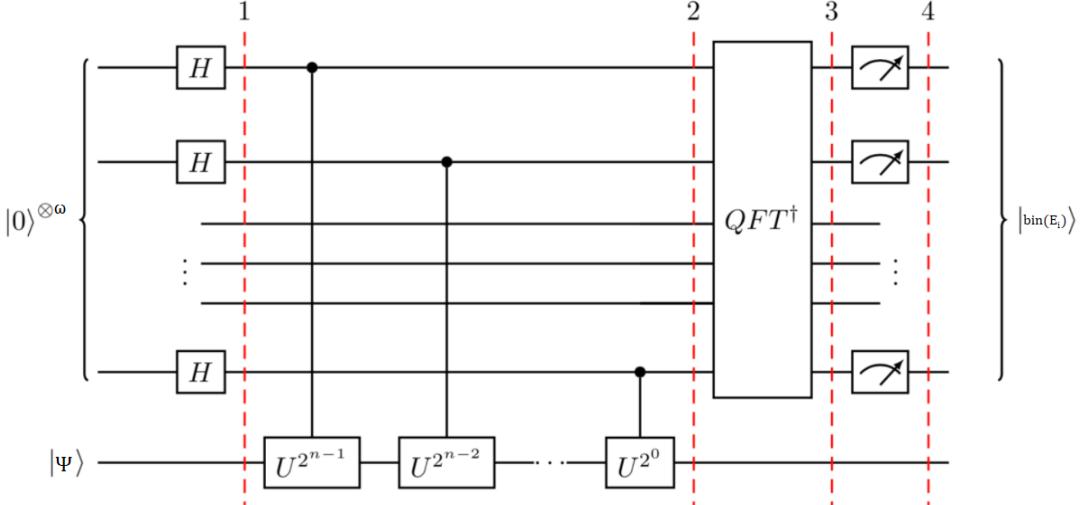


Figure 4.1: Circuital scheme of the QPE algorithm.

Several techniques have been proposed for state preparation. One approach is to prepare reference states obtained from classically tractable calculations. Alternatively, we can use the variational methods discussed in Chapter 5.

Another method is adiabatic state preparation (ASP), an approach inspired by the adiabatic model of quantum computation. To do so, we first start with a simple Hamiltonian H_0 and prepare its ground state. We then time evolve the system under a Hamiltonian that changes slowly from H_0 to H_s , thus preparing a state that is close to the ground state of H_s . The efficiency of ASP depends on the gap between the ground state and the first excited state along the path between H_0 and H_s . For chemical systems, ASP may be achieved by initializing the system in the ground state of the Hartree-Fock Hamiltonian (H_0), and interpolating between the initial and final Hamiltonians using an annealing schedule such as $H(t) = (1 - t/T)H_0 + (t/T)H_s$, where t is the time and T is the maximum desired simulation time [31].

Numerical simulations support this method, it was shown that annealing times for methylene (CH_2) could be reduced by up to 4 orders of magnitude by using an initial state with larger overlap with the true ground state [47]. We note, however, that if an initial state with sufficiently large overlap with the ground state is available, we may be able to forgo ASP entirely and instead carry out phase estimation directly on that initial state. As discussed previously, phase estimation requires only a non-negligible overlap with the target ground state.

4.2.2 Hamiltonian simulation

As discussed, the canonical phase estimation algorithm require implementation of the time evolution operator e^{-iHt} , where H may or may be not time dependent. As described in Chapter 3, the simulation of Hamiltonian dynamics, i.e. the solution of the time-dependent Schrödinger equation (TDSE), is a **BQP**-complete problem, and thus a natural application for a quantum computer [33].

There are several ways to do this, each with its own advantages and disadvantages. The most simple method for time evolution is the Trotter decomposition formula, which we have used here to describe the algorithm. If a time-independent Hamiltonian H can be decomposed as $H = \sum_i h_i$, where h_i are local Hamiltonians, then a first order Lie-Trotter-Suzuki approximation of the time evolution is

$$U = e^{-iHt} = e^{-i\sum_i h_i t} = \left(\prod_i e^{-ih_i t/S} \right)^S + O(t^2/S), \quad (4.11)$$

This approach is also referred to as the 'product formula' method. In practice, to achieve accuracy ϵ in the retrieving of the correct unitary evolution, the number of Trotter steps $S = O(t^2/\epsilon)$ should be large in order to suppress the errors in the approximation. This is effectively a step by step evolution under time evolution operators corresponding to each of the terms in the Hamiltonian. It is also possible to use higher order product formulas, which scale better with respect to the simulation error than the first order method. Randomization procedures, such as randomly ordering the terms in the Trotter sequence or stochastically choosing which terms to include in the Hamiltonian, have been shown to improve the accuracy obtained using product formulas.

Product formulas can also be used to simulate dynamics under a time-dependent Hamiltonian $H(t)$ and it can be shown that the accuracy of such simulations depends on the derivatives of the Hamiltonian [50].

Alternative methods that may realize the time evolution operator more efficiently than Trotterization have been introduced, including quantum-walk-based methods, multiproduct formulas and qubitization. One approach is the 'linear combinations of unitaries' (LCU) query model, which decomposes the Hamiltonian or time evolution operator into a linear combination of unitary operators that are then applied in superposition using oracle circuits. These must be explicitly constructed for a given problem.

As a linear combination of unitaries is itself not necessarily unitary, these approaches may require additional techniques such as amplitude amplification to maintain a high probability of success [33].

4.3 Computational requirements

A complete analysis of the computational requirements for quantum algorithms is hard to achieve, since it highly depends on the device architecture and the target of the simulation, but we can study some common indicators in order to do a comparison between different methods.

The algorithmic parameters that we consider here are:

- **Circuit width (n_q)**
i.e. the number of qubits in the registers;
- **Circuit depth (n_g)**
i.e. the number of layers of gates that must be applied, where a layer of gates is a set of gates that can be applied simultaneously;

- **Repetitions** (n_{rep})

i.e. the number of times the circuit must be performed, but not the measurement, to obtain the result;

- **Total scaling** ($n_t = n_g \cdot n_{rep}$).

In Chapter 6 we will also refer to technological inspired characteristics like connectivity or coherence time.

Generally speaking the depth and the repetitions for the QPE algorithm scale as

$$n_g = O\left(\frac{1}{\epsilon}\right), \quad n_{rep} = O(1), \quad (4.12)$$

where ϵ is the error between the result and the correct value of the phase φ , or equivalently the accuracy of the estimate of the algorithm.

If we consider the simulation of a quantum system ($\varphi = E_i$), then we have to consider: first, the probability overlap η of the initial state with the target state $|\chi_i\rangle$; second, depending on the method used to do the simulation, the coefficients of the resulting qubit Hamiltonian, which can be summarized with the limit value $E_{max} = \sum_i |w_i|$; finally the number of orbitals, m , with which we model the electrons. Thus the circuit depth scales as

$$n_g = O\left(\frac{E_{max}m}{\epsilon\eta}\right). \quad (4.13)$$

As for the width, QPE requires approximately m qubits to store the relevant quantum state,

$$n_q = O(m). \quad (4.14)$$

However, QPE typically also requires additional auxiliary qubits.

First, such qubits are needed to store the bits corresponding to the numerical value of the energy estimate. The number of ancilla qubits ω required is determined by the desired success probability and precision. If the phase E_i can not be written exactly with a ω binary expression, one can prove that to successfully obtain E_i accurate to n bits with probability of success at least $1 - \epsilon$ we choose [35]

$$\omega = n + \left\lceil \log\left(2 + \frac{1}{2\epsilon}\right) \right\rceil, \quad (4.15)$$

however, this can be reduced to just a single qubit using iterative phase estimation. Auxiliary qubits are also required for some methods of implementing the required unitary operators, but this is negligible for the most recent methods.

Secondly, since the circuits used when performing QPE are very deep we expect error correction procedures to be required in order to obtain useful results from the calculations. This introduce an overhead on both the circuit width (spatial overhead) and depth (temporal overhead). We write these overheads as θ_S and θ_T respectively. For the surface code, which is one of the most researched, the overheads are determined by the code distance, d , with $\theta_S \sim d^2$ and $\theta_T \sim d$, and therefore $\theta_S \sim \theta_T^2$. We note that, in order to maintain a constant probability of a logical error occurring, these overheads must increase with increasing logical circuit depth and number of

logical qubits; however, they increase logarithmically and so we ignore this here. In conclusion, the algorithmic parameters scale as [8]

$$n_q^{QPE} = O(m\theta_T^2), \quad n_g^{QPE} = O\left(\frac{E_{max}m\theta_T}{\epsilon\eta}\right), \quad n_{rep}^{QPE} = O\left(\frac{1}{\eta}\right). \quad (4.16)$$

In the next chapter we will see how these parameters, especially the number of measurements, are related to the coefficients in the qubit Hamiltonian. For now we use the result from Lee et al. [25], which states that E_{max} scales between $O(m)$ and $O(m^3)$.

Thus the total scaling of the QPE algorithm is

$$n_t^{QPE} = n_g^{QPE} \cdot n_{rep}^{QPE} = O\left(\frac{E_{max}m\theta_T}{\epsilon\eta^2}\right). \quad (4.17)$$

4.3.1 Examples

Several studies estimated the resource requirements for QPE methods by combining and comparing different components of the algorithm and the general consensus is that the fault-tolerant overhead and the long coherence time needed to execute the algorithm make the method very expensive, but at the same time the most accurate one can use for simulation. We discuss more on this in Chapter 6.

One of the first estimates for the QPE algorithm applied to molecular systems was made by Alan Aspuru-Guzik et al. in 2005 [5]. They showed that the number of qubits required for both the compact and direct mappings scales linearly with the number of basis functions. For the compact mapping the qubit requirement depends also on the ratio of number of electrons to basis functions, which is relatively constant for a given basis set.

Although the higher quality cc-pVTZ basis is more economical per basis function, a molecule in this basis uses substantially more functions than with the 6-31G* basis. This trend and the qubits required for specific molecules and basis sets are represented in Figure 4.2.

In this research they carried out calculations on H₂O and LiH. For the former they used the minimal STO-3G basis set, yielding 196 singlet-spin configurations, and for the latter they used the larger 6-31G basis yielding 1210 such configurations. The algorithm they used is a modified QPE which uses a relatively small number of qubits. The sequence is: a reference energy is estimated, then the Hamiltonian is shifted by that value and an estimate of the deviation is computed, then the reference energy is updated and the procedure is repeated until the desired precision is obtained. With each iteration one additional bit of E is obtained.

After 20 iterations the electronic energy obtained for H₂O matched the Hamiltonian diagonalization energy and so did the LiH calculation, both within chemical accuracy.

They ended up proving that the lengths of the gate sequences involved are bounded from above by a polynomial function of the number of qubits. This a consequence of the decomposition of the unitary evolution operator.

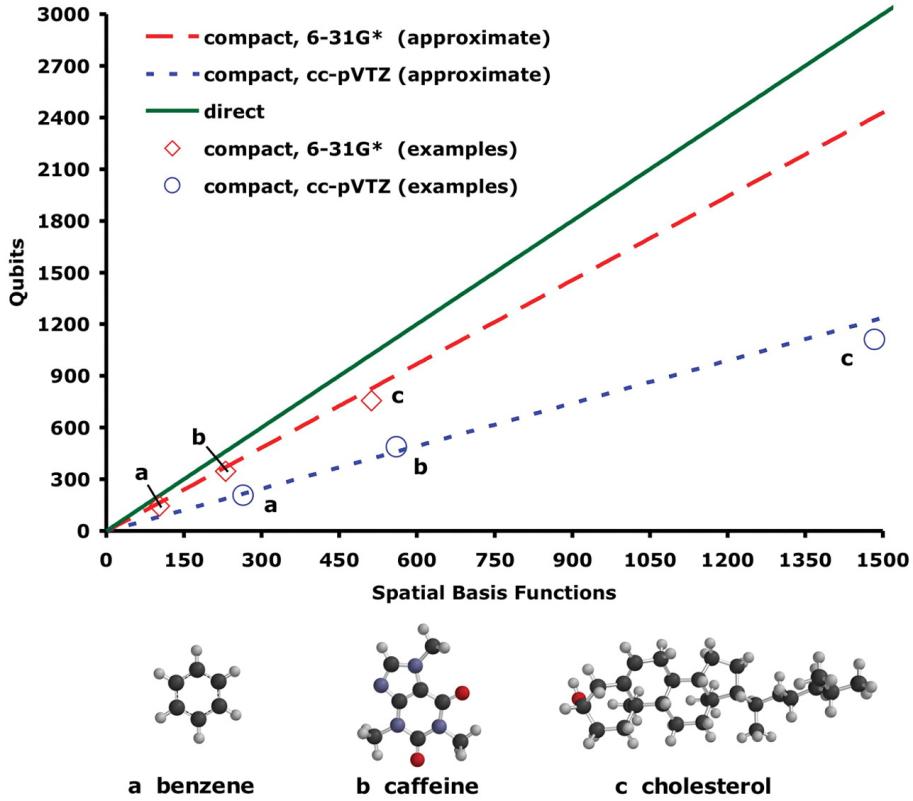


Figure 4.2: Qubit requirements versus basis size [5].

Moreover, analyzing the mappings between fermionic and qubit Hamiltonians we will see that the number of terms grows approximately with the fourth power of the number of orbitals, thus, the most expensive part would be the the number of one- and two-qubits elementary gates used to represent an arbitrary four-qubit unitary operation.

A more recent estimate on the QPE algorithm has been presented by Markus Reiher et al. in 2017 [41]. They considered the chemical process of biological nitrogen fixation by the enzyme nitrogenase, also called MoFe protein. This enzyme accomplishes the remarkable transformation of turning a dinitrogen molecule into two ammonia molecules and one dihydrogen molecule under ambient conditions. Whereas the industrial Haber-Bosch catalyst requires high temperatures and pressures and is therefore energy intensive, the active site of Mo-dependent nitrogenase (the iron molybdenum cofactor or FeMoco) can split the dinitrogen triple bond at room temperature and standard pressure. Mo-dependent nitrogenase consists of two subunits, the Fe protein, a homodimer, and the MoFe protein, an $\alpha_2\beta_2$ tetramer. Its X-ray crystal structure is shown in Figure 4.3.

Despite the importance of this process for fertilizer production that makes nitrogen from air accessible to plants, the mechanism of nitrogen fixation at FeMoco is not yet completely understood.

Authors used standard DFT, with B3LYP functional, for molecular structure op-

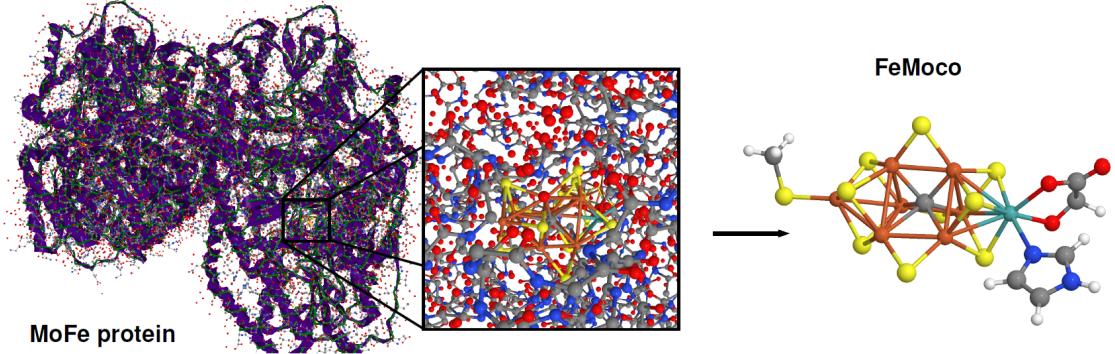


Figure 4.3: Structure of the MoFe protein and the FeMoco active site [41].

timization, thanks to its low computational requirements even for complex many-body systems like this one, and studied the active site with a quantum computer that accesses truly large active orbital spaces, the latter is sometimes called multi-configuration wave function model or complete active space (CAS) method, if all the combinations of the active space orbitals are included. They focused on QPE using the Trotter decomposition formula.

In order to consider error correction in the algorithm they made two approximations. First, the exponentials in the Trotter formula are decomposed into single-qubit rotations and Clifford gates. In the surface code, which they considered for qubit encodings, Clifford gates can be implemented fault tolerantly; the single-qubit rotations, however, require approximation by a discrete set of gates consisting of Clifford operations and at least one non-Clifford operation, usually taken to be the T gate, a rotation by $\pi/8$ about the z-axis. Each T gate requires a procedure called 'magic state distillation', which consumes a host of noisy quantum states to output a single accurate magic state.

Secondly, the magic state is used to teleport a T gate into the computation. The space and time overheads of state distillation render it by far the most costly aspect of quantum error correction, leading to a large multiplicative overhead for each single-qubit rotation. The number of T gates therefore typically dominates the cost when implementing a fault tolerant algorithm.

The resource estimates are based on extrapolations from the costs found from simulations of smaller molecules and focus on two prototypical structures of FeMoco, one with 54 electrons in 54 spatial orbitals (108 spin-orbitals) and the second with 65 electrons and 57 spatial (114 spin-orbitals) orbitals. Moreover, the calculation is conducted within a basis set that is a reasonable match for the target accuracy required on the energy value, i.e. 1 mHa (chemical accuracy) and 0.1 mHa. These errors include the error in phase estimation, the error in the Trotter deoceanation and errors from decomposing the Trotter approximation into H and T gates.

Three implementations of the quantum algorithm are considered. Serial: the single-qubit rotations are constrained to occur serially. Nesting approach: Hamiltonian terms that affect disjoint sets of spin-orbitals are executed in parallel, which sub-

stantially reduces the runtime of the calculations. Programmable ancilla rotations (PAR), rotations are cached in qubits, which are then teleported into the circuit as needed.

The cost in terms of logical qubits is shown in Figure 4.4.

Quantitatively accurate simulation (0.1 mHa)				
Struct. 1	T-Gates	Clifford Gates	Time	Log. Qubits
Serial	1.1×10^{15}	1.7×10^{15}	130 days	111
Nesting	3.5×10^{15}	5.7×10^{15}	15 days	135
PAR	3.1×10^{16}	3.1×10^{16}	110 hours	1982
Struct. 2	T-Gates	Clifford Gates	Time	Log. Qubits
Serial	2.0×10^{15}	3.1×10^{15}	240 days	117
Nesting	6.5×10^{15}	1.0×10^{16}	27 days	142
PAR	6.0×10^{16}	6.0×10^{16}	204 hours	2024
Qualitatively accurate simulation (1 mHa)				
Struct. 1	T-Gates	Clifford Gates	Time	Log. Qubits
Serial	1.0×10^{14}	1.6×10^{14}	12 days	111
Nesting	3.3×10^{14}	5.6×10^{14}	1.4 days	135
PAR	3.0×10^{15}	3.0×10^{15}	11 hours	1982
Struct. 2	T-Gates	Clifford Gates	Time	Log. Qubits
Serial	1.9×10^{14}	3.0×10^{14}	22 days	117
Nesting	6.0×10^{14}	9.9×10^{14}	2.5 days	142
PAR	5.5×10^{15}	5.5×10^{15}	20 hours	2024

Figure 4.4: Simulation time and resources estimate for FeMoco [41].

We can see that the nesting approach gives a reasonably trade-off between the other methods.

To complete the resource estimate the authors added the overheads required to perform the simulation fault tolerantly. The physical error rates in the hardware is assumed to be: 10^{-3} , a near-term standard, 10^{-6} or 10^{-9} , which might be achieved in future. The results are shown in Figure 4.5.

The number of logical qubits in the main quantum processor is in the hundreds, which translates into tens of thousands up to millions of physical qubits. Most of the qubits are used in the T factories, each of which needs fewer physical qubits than the main quantum processor.

The authors also noted that the required quantum computing resources are comparable to that needed for Shor's factoring algorithm for interesting 4096 bit numbers, both in terms of number of gates and also physical qubits. Also, the resources required depend sensitively on gate error rates.

	Serial rotations			PAR rotations			Nested rotations		
Error Rate	10^{-3}	10^{-6}	10^{-9}	10^{-3}	10^{-6}	10^{-9}	10^{-3}	10^{-6}	10^{-9}
Required code distance	35,17	9	5	37,19	9,5	5	37,17	9	5
Quantum processor									
Logical qubits	111			110			109		
Physical qubits per logical qubit	15313	1013	313	17113	1013	313	17113	1013	313
Total physical qubits for processor	1.7×10^6	1.1×10^5	3.5×10^4	1.9×10^6	1.1×10^5	3.4×10^4	1.9×10^6	1.1×10^5	3.4×10^4
Discrete Rotation factories									
Number	0			1872			26		
Physical qubits per factory	—	—	—	17113	1013	313	17113	1013	313
Total physical qubits for rotations	—	—	—	3.2×10^7	1.9×10^6	5.9×10^5	4.5×10^5	2.6×10^4	8.1×10^3
T factories									
Number	202	68	38	166462	41110	29659	5845	1842	1029
Physical qubits per factory	8.7×10^5	1.7×10^4	5.0×10^3	1.1×10^6	7.5×10^4	5.0×10^3	8.7×10^5	1.7×10^4	5.0×10^3
Total physical qubits for T factories	1.8×10^8	1.1×10^6	1.9×10^5	1.8×10^{11}	3.1×10^9	1.5×10^8	5.1×10^9	3.0×10^7	5.2×10^6
Total physical qubits	1.8×10^8	1.2×10^6	2.3×10^5	1.8×10^{11}	3.1×10^9	1.5×10^8	5.1×10^9	3.0×10^7	5.2×10^6

Figure 4.5: Resource estimates including error correction [41].

Chapter 5

Quantum computing for computational chemistry: NISQ devices

5.1 Noisy intermediate-scale quantum devices

The acronym NISQ stands for 'noisy intermediate-scale quantum' devices and it is an expression coined by John Preskill in 2018 [38]. It indicates quantum computers with a number of qubits ranging from 50 to a few hundreds, thus the term 'intermediate-scale'. 'Noisy' emphasizes that the control over these qubits is imperfect. Some of these devices are already available today and they are regarded as a step toward more powerful quantum technologies that will be developed in the future.

50 qubits is a significant milestone, because it is estimated to be beyond what can be simulated by brute force using the most powerful existing digital supercomputers, but, as we already mentioned, quantity is not the only important algorithmic parameter and quality must be considered and improved, i.e. longer coherence time, lower quantum noise and environment interactions and higher accuracy with which to perform quantum gates.

With these noisy devices researchers do not expect to be able to execute a circuit that contains many more than about 1000 gates, that is, 1000 fundamental two-qubit operations, because the noise will overwhelm the signal in a circuit much larger than that. This limitation on the circuit size imposes a ceiling on the computational power of NISQ technology. This condition can likely be improved only by using quantum error correction to scale up to larger circuits, meaning that when we speak of the NISQ era, we describe quantum computers with noisy gates, unprotected by quantum error correction.

A great component of the scientific debate on NISQ devices, and also on quantum computers, is if and when they will be able to solve important problems faster than classical computers and what type of problems. This condition is defined 'quantum speedup' and it considers using classical computers with the best available hardware and running the best algorithm which performs the same task.

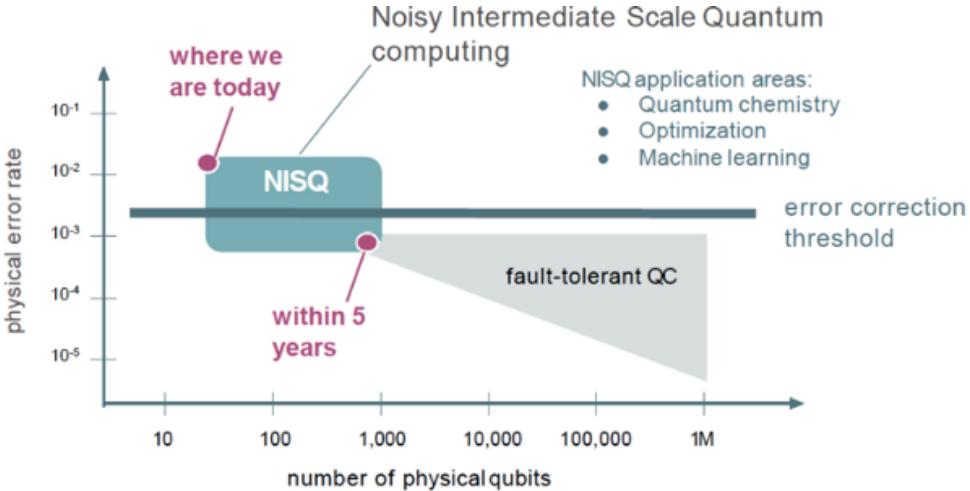


Figure 5.1: NISQ computing: error rate vs number of qubits.

Several research fields are considered better candidates for the first result to be shown using these devices, among these are: machine learning, optimization processes, linear algebra and simulation of quantum systems.

5.2 Variational quantum eigensolver

A highly researched algorithm for near-term quantum hardware is the variational quantum eigensolver (VQE), first proposed and experimentally realized by Alberto Peruzzo et al. [36] and elaborated on by McClean et al. [32].

The VQE aims to find the lowest eigenvalue of a given Hamiltonian, such as that of a chemical system, and is a hybrid quantum-classical algorithm. This means that it uses a quantum computer for state preparation, unitary evolution and measurement subroutine and it uses a classical computer to process the measurement results and update the quantum computer according to a specific rule. This exchanges the long coherence times needed for QPE for a overhead due to measurement repetitions and classical processing.

The VQE relies upon the Rayleigh-Ritz variational principle. This states that, for a parametrized trial wave function $|\Psi(\vec{\theta})\rangle$,

$$\langle\Psi(\vec{\theta})|H|\Psi(\vec{\theta})\rangle \geq E_0, \quad (5.1)$$

where $\vec{\theta}$ is a vector of independent parameters $\vec{\theta} = (\theta_1, \dots, \theta_n)^T$. This implies that we can find the ground state wave function and energy by finding the values of the parameters which minimize the energy expectation value,

$$E_{VQE} = \min_{\vec{\theta}} \langle 0 | U^\dagger(\vec{\theta}) H U(\vec{\theta}) | 0 \rangle. \quad (5.2)$$

As classical computers are unable to efficiently prepare, store and measure the wave function, we use the quantum computer for this subroutine. We then use the classical computer to update the parameters using an optimization algorithm.

The scheme of the algorithm is:

1. Hamiltonian construction and representation

Here we define the system to study. We specify the geometry, find the specific operators and their weights between basis functions spanning the physical problem.

2. Encoding of operators

Qubit registers can only measure observables expressed in a Pauli basis. In second quantization we have fermionic operators, thus we need a transformation of fermionic operators to spin operators which maintain the antisymmetry as given by Pauli exclusion principle. This is also called fermionic-to-spin mapping or encoding.

The key factors of an encoding are their Pauli weight, the number of qubits required and the number of Pauli strings produced.

3. Ansatz and state preparation

We prepare a trial wave function to measure the expectation value. For this we must decide on a structure for the parametrized quantum circuit, denoted as ansatz.

A wide range of ansätze are possible, the key aspects are their expressibility and trainability. The former defines the ability of the ansatz to span a large class of states in the Hilbert space, the latter describes the practical ability of the ansatz to be optimized. These aspects are translated in terms of scaling and complexity of the circuit depth with system size.

4. Measurement strategy and grouping

We determine how measurements are distributed and organized to efficiently extract the required expectation values from the trial wave function. The objective of this component of the algorithm is to make the number of repetitions of the circuit as low as possible.

5. Parameter optimization

Here we update the parameters of the ansatz iteratively until convergence. This requires sampling the expectation value of the Hamiltonian several times for a given parameter set in the ansatz in order to define an update rule for the parameters.

6. Error mitigation

The VQE method is to be used without error correction schemes on NISQ devices. Error mitigation aims to reduce the impact of quantum noise through post-processing of the measurement data.

The sequence of components is shown in Figure 5.2.

In the following we analyze some important components of the VQE method: the encoding of operators, the ansatz and state preparation, the efficient grouping and measuring strategies and the error mitigation.

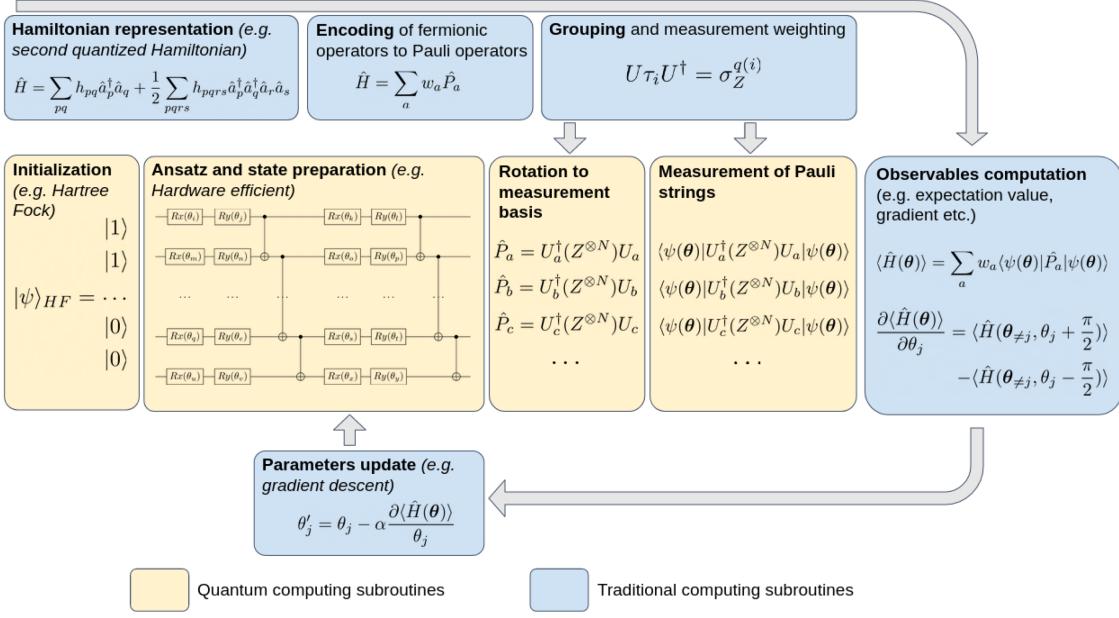


Figure 5.2: Flowchart for the VQE algorithm [46].

5.2.1 Encoding of operators

In the NISQ-era algorithms second quantization formalism has been favored over first quantization for quantum simulation. The fermionic creation and annihilation operators in second quantization formalism obey the anticommutative algebra. Qubits in comparison are spin-1/2 objects and, as such, the only operators that can be directly measured on quantum processors are spin operators, i.e. the Pauli operators X , Y , and Z , which obey a different algebra specified by their Lie bracket and do not naturally obey anticommutative rules.

All transformations between these two sets of operators can be formalized as

$$\mathcal{T} : \mathcal{F}_n \rightarrow (\mathbb{C}^2)^{\otimes N}, \quad (5.3)$$

where a transformation \mathcal{T} maps the space of operators acting on Fock states of n orbitals, \mathcal{F}_n , to the Hilbert space $(\mathbb{C}^2)^{\otimes N}$ of operators acting on spin states of N qubits.

The second quantized fermionic Hamiltonian is mapped to a linear combination of products of single-qubit Pauli operators

$$H = \sum_j w_j P_j = \sum_j w_j \prod_i \sigma_i^j, \quad (5.4)$$

as already presented in Chapter 4, where w_j is the coefficient that correspond to the Pauli string P_j . The mappings described here not only affect the operators measured when computing the expectation value of the Hamiltonian, but also the construction of an ansatz that is initially defined in fermionic terms.

There are three main features relevant to deciding on a specific encoding:

- **The number of qubits required to represent the electronic wave function**

In general, the number of qubits is directly proportional to the number of spin-orbitals or sites considered. However, several techniques have been developed to concentrate the information held in the wave function to as few qubits as possible.

- **The maximum number of qubits on which each produced Pauli string acts**

This is the maximum number of non-identity operators in any Pauli string produced by the mapping and it is referred to as Pauli weight. It is important because low Pauli weight results in lower depth ansätze and lower circuit construction costs, also it may provide resilience against the barren plateaus problem, i.e. a trainability problem that occurs in optimization algorithms when the gradient used to update the parameters $\vec{\theta}$ vanishes in every direction of the multidimensional space.

- **The number of different Pauli strings resulting from the mapping**

In general, we see that this number of strings scales as $O(m^4)$ for molecular Hamiltonian and with the number of edges for lattice models.

Among the most used encodings are:

1. Jordan-Wigner encoding (JW)

This is a map that encodes the electronic wave function in an array of qubits by mapping the occupation number of spin-orbitals directly.

The theoretical foundation of this mapping lies in the 1928 Jordan-Wigner transformation which uses spin-1/2 operators to explicitly describe fermionic ladder operators.

We store the occupation number of a spin-orbital in the $|0\rangle$ or $|1\rangle$ state of a qubit. More formally,

$$|n_0, n_1, \dots, n_i, \dots, n_{m-1}\rangle \rightarrow |q_0, q_1, \dots, q_i, \dots, q_{m-1}\rangle, \quad (5.5)$$

$$q_i = n_i \in \{0, 1\}. \quad (5.6)$$

The fermionic creation and annihilation operators increase or decrease the occupation number of a spin-orbital by 1 and they also introduce a multiplicative phase factor. The qubit mappings of the operators preserve these features and are given by

$$a_i = \frac{X_i + iY_i}{2} \otimes Z_{i-1} \otimes \dots \otimes Z_0, \quad (5.7)$$

$$a_i^\dagger = \frac{X_i - iY_i}{2} \otimes Z_{i-1} \otimes \dots \otimes Z_0. \quad (5.8)$$

The operators $\frac{X_i \pm iY_i}{2}$ change the occupation number of the target spin-orbital, while the string of Z operators recovers the exchange phase factor $(-1)^{\sum_k n_k}$. Working in the JW basis it is easy to see the advantage that quantum computers have over their classical counterparts for chemistry problems, since every

Slater determinant required for the FCI wave function can be written as one of these computational basis states. As such, quantum computers can efficiently store the FCI wave function. However, while the occupation of a spin-orbital is stored locally, the parity is stored nonlocally. From the string of Z operators we can see that a fermionic operator mapped to qubits generally has a weight of $O(m)$ Pauli operators, each acting on a different qubit.

2. Bravyi-Kitaev encoding

The Bravyi-Kitaev (BK) encoding is a midway point between the JW encoding and parity encoding methods because it compromises on the locality of the occupation number and the parity information. The orbitals store partial sums of occupation numbers and the occupation numbers included in each partial sum are defined by the BK matrix β_{ij} .

$$|n_0, n_1, \dots, n_i, \dots, n_{m-1}\rangle \rightarrow |q_0, q_1, \dots, q_i, \dots, q_{m-1}\rangle, \quad (5.9)$$

$$q_i = \sum_j \beta_{ij} n_j \pmod{2}. \quad (5.10)$$

The BK matrix is defined recursively via

$$\beta_1 = 1, \quad (5.11)$$

$$\beta_{2^{x+1}} = \begin{pmatrix} \beta_{2^x} & \mathbf{0} \\ \mathbf{A} & \beta_{2^x} \end{pmatrix}, \quad (5.12)$$

where \mathbf{A} is a $2^x \times 2^x$ matrix of zeroes, with the bottom row filled with ones, and $\mathbf{0}$ is a $2^x \times 2^x$ matrix of zeroes.

Applying the BK mapping to a fermionic operator results in a qubit operator with a Pauli weight of $O(\log_2 m)$.

3. Optimal general encoding based on ternary trees

'Optimal' refers to the fact that for a Hamiltonian for which fermionic modes are fully connected it achieves the minimum average Pauli weight possible. It organizes qubits along with ternary trees and relies on the definition of the second quantized Hamiltonian in terms of Majorana fermions.

Majorana fermions are theorized particles which acts as their own antiparticle. Formally this means that creation and annihilation operators for Majorana fermions are identical $\gamma_i^\dagger = \gamma_i$. These must anticommute if they are of different indices and commute otherwise.

To build this encoding one must first map the qubits to the vertices of a ternary tree (a tree that splits into three edges after each vertex), as shown in Figure 5.3.

For any path p in the tree one can define the following operators:

$$A_p = \bigotimes_{l=0}^{h-1} \sigma_{\alpha(l)}^{\nu(l)}, \quad (5.13)$$

where h is the height of the tree, ν is the qubit index on path p and α corresponds to X , Y or Z , depending on whether the path follows the left, central

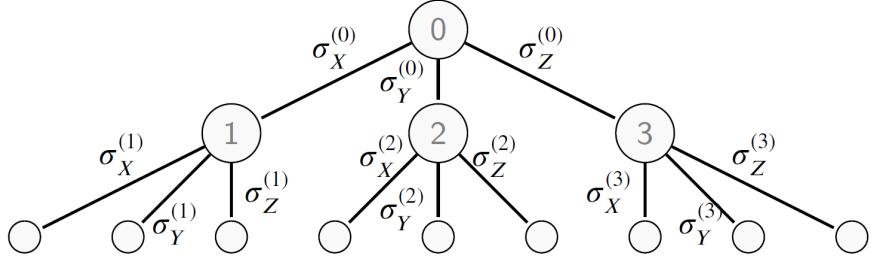


Figure 5.3: Ternary tree structure in the case of a 4 fermionic modes system [46].

or right edge respectively after qubit ν .

These operators clearly obey the same anticommutative relationships as Majorana operators. Given that there are $2n + 1$ distinct path in the ternary tree and that we need $2n$ Majorana operators, we can map each A_p to a single Majorana operator and these operators have a maximum Pauli weight equal to the height of the tree (h), hence $\log_3(2n + 1)$.

To construct the Hamiltonian one needs to first transform the fermionic operators into Majorana operators and rewrite the Hamiltonian and then decide on an allocation of the Pauli operators defined above to the Majorana operators. Similar to the encodings defined previously defined, the number of qubits n_q is equal to the number of fermionic modes m and the scaling of the number of Pauli operators is the same as that of the two body-terms in the second quantized Hamiltonian $O(m^4)$.

5.2.2 Ansatz and state preparation

The parametrized circuits, or ansätze, for the VQE lie between two extremes: hardware efficient and chemically inspired. There has been relatively little work comparing the effectiveness of different ansätze for anything but the smallest chemistry problems.

1. Hardware-efficient ansätze (HEA)

These ansätze are composed of repeated dense blocks of a limited selection of parametrized gates that are easy to implement with the available hardware. They seek to build a flexible trial state using as few gates as possible. As such, they are well suited to the quantum computers currently available, which have short coherence times and constrained gate topologies. However they are unlikely to be suitable for larger systems, as they take into account no details of the chemical system being simulated.

A generic representation of HEA is

$$|\Psi(\vec{\theta})\rangle = \left(\prod_{i=1}^d U_{rot}(\theta_i) U_{ent} \right) U_{rot}(\theta_{d+1}) |\Psi_{in}\rangle, \quad (5.14)$$

where U_{rot} and U_{ent} indicate a sequence of rotational and entangling gates. The main promise of hardware-efficient ansätze is that it can be flexibly tailored to the specific native gate set of the device used, while at the same time

being highly expressive.

The scaling of this method can be controlled by the number of layers L of the circuit built, the depth scales as $n_g = O(L)$ and the number of parameters scales as $n_{rep} = O(mL)$.

2. Chemically inspired ansätze

Chemically inspired ansätze result from adapting classical computational chemistry algorithms to run efficiently on quantum computers. Most notably, the CC method discussed in Chapter 1 can be extended to produce the ‘unitary coupled cluster’ (UCC) ansatz [2].

The UCC method creates a parametrized trial state by considering excitations above the initial reference state and can be written as

$$U(\vec{\theta}) = e^{T - T^\dagger}, \quad (5.15)$$

where $T = \sum_i T_i$, and

$$T_1 = \sum_{i,\alpha} t_{i\alpha} a_i^\dagger a_\alpha, \quad (5.16)$$

$$T_2 = \sum_{i,j,\alpha,\beta} t_{i,j,\alpha,\beta} a_i^\dagger a_j^\dagger a_\alpha a_\beta, \dots, \quad (5.17)$$

i, j indicate the occupied orbitals in the reference state and α, β indicate the orbitals that are initially unoccupied in the reference state.

The UCC method is intractable on classical computers, but can be efficiently implemented on a quantum computer, as showed in the first description of the VQE algorithm by Peruzzo [36].

A clear drawback of UCC-type ansätze is specifically that these are not hardware efficient. All UCC ansätze are designed in a manner that is agnostic to the connectivity of the device. The number of CNOT gates for each of the exponentiated Pauli strings required in the ansatz scales with its Pauli weight when full connectivity is allowed. Overall, this implies a very large prefactor for the depth and number of entangling gates of UCC anäteze.

For strongly-correlated systems, UCCSD may not be able to prepare a state which is suitably close to the ground state due to the limitation on the excitations considered. That is why an important extension of the UCC method can be used: the ‘paired-UCC with generalized singles and doubles’, k-UpCCGSD. This version is based on the paired-coupled cluster method, it only includes two-body terms that move pairs of electrons together between spatial orbitals. ‘Generalized’ means that unlike conventional CC and UCC, which only include excitation operators that correspond to transitions from occupied to virtual orbitals, UCCG is agnostic to the electron configuration.

Finally, k is a constant that indicates how many times the ansatz is repeatead allowing more flexibility for the wave function description.

The key advantage of this method is to allow a linear scaling ansatz, namely with m the number of spin-orbitals, the ansatz depth scales $O(km)$, and a quadratic scaling in the number of parameters $O(km^2)$.

3. Adaptative structure ansätze

Adaptative structure ansätze aim to create a circuit structure tailored to the problem studied. As such they all work similarly: the ansatz is grown iteratively throughout the optimization process by adding new operators at each step based on their contribution to the overall energy estimate.

The most simple is the ADAPT-VQE. Starting from an initial state, usually the Hartree-Fock wave function, and given a pool of operators $\{A_i\}$ the ADAPT-VQE ansatz evolves as follows, with the iteration number as superscripts:

$$|\Psi^0\rangle = |\Psi_{HF}\rangle \quad (5.18)$$

$$|\Psi^1\rangle = e^{\theta_1 A_1} |\Psi_{HF}\rangle \quad (5.19)$$

$$|\Psi^2\rangle = e^{\theta_2 A_2} e^{\theta_1 A_1} |\Psi_{HF}\rangle \quad (5.20)$$

$$\dots \quad (5.21)$$

$$|\Psi^k\rangle = \prod_i e^{\theta_i A_i} |\Psi_{HF}\rangle. \quad (5.22)$$

The operators in the pool are similar to those produced as part of the UCC ansatz, however, rather than specifying a number of possible excitation, one can include any one-, two body- or even higher body operator that is believed to be particularly relevant for the system considered. A good choice are the operators that result in the highest gradient of the energy functional with respect to the operator parameter.

The k-UpCCGSD ansatz and its extensions have been numerically shown to achieve excellent accuracy while offering linear scaling, thereby arguably offering the best trade-offs of cost to accuracy among proposed ansätze. Adaptive ansätze could place themselves as a reliable alternative subject to further studies on their expected computational cost.

5.2.3 Efficient grouping and measuring strategies

One of the key challenges possibly holding back the VQE is the very large amount of samples that are required to accurately compute the relevant values of the algorithm. There are two main aspects to manage for efficiently sampling these expectation values: the number of terms in the Hamiltonian cost functions, computed using the mappings described before, and the number of shots required to sample an expectation value at a certain level of accuracy.

First, let's consider the overall scaling of measurements, i.e. the number of shots. As seen previously, generalized mappings for molecular Hamiltonians result in $P = O(m^4)$ distinct Pauli strings to estimate.

In any sampling experiment the standard error is equal to $\epsilon = \sigma/\sqrt{S}$, where σ is the population standard deviation and S is the experimental sample size, in our case the number of shots. This means that the number of times an experiment needs to be repeated to achieve a given expected error ϵ goes as $O(1/\epsilon^2)$. More

specifically, when measurements are distributed optimally among the different Pauli strings, such that the variance is minimized with respect to a given precision ϵ , the number of measurements required is upper-bounded by

$$S \leq \left(\frac{\sum_i^P w_i}{\epsilon} \right)^2, \quad (5.23)$$

where w_i are the weights of the Pauli strings in the Hamiltonian.

As a result, for a given level of accuracy for each Pauli string measured independently, the overall scaling of the number of shots required for an energy estimation is:

$$S = O\left(\frac{m^4}{\epsilon^2}\right). \quad (5.24)$$

In the context of quantum chemistry, successful computing methods are expected to produce results within a precision of $\epsilon = 1$ mHa to the target. When results obtained numerically are within this level of precision to experimental results the simulation is deemed to reach chemical accuracy.

One should be cautious however not to assume too much of a relationship between this number and the number of shots required to perform VQE. That is because the key bottleneck of VQE optimization is not the estimation of the wave function itself but the estimation of gradients and in particular the difference between these gradients. While polynomial in scaling, it has been pointed out on several occasions that the number of shots required to accurately compute a VQE optimization process rapidly becomes unmanageable.

1. Qubit-wise commutativity (QWC)

Two Pauli strings are said to be QWC if each Pauli operator in the first string commute with the Pauli operator of the second one that has the same index. Generally speaking, that would be any group where any given Pauli operator in any Pauli string has an index such that all the operators of the same index across all the other Pauli strings in the group are either the same Pauli operator or the identity (for example, XI , IZ , and XZ are altogether QWC).

This basis for grouping terms has been widely used and studied. It allows performing joint measurements more efficiently. In particular, Gokhale et al. [16] found that this method reduces the prefactor for the number of Pauli terms to be measured by about three, without however changing its asymptotic scaling.

Another key advantage is that the basis rotation used to conduct the joint measurements only requires a circuit of depth 1. To achieve this, we need to find the unitary which rotates all the Pauli strings in a given QWC group into a basis in which they are all diagonalized. This is a straightforward process as any individual Pauli operator can be rotated in the Z basis with one single-qubit operation as follows:

$$Z = R_y\left(-\frac{\pi}{2}\right) X R_y\left(-\frac{\pi}{2}\right)^\dagger, \quad (5.25)$$

$$Z = R_x\left(\frac{\pi}{2}\right) Y R_x\left(\frac{\pi}{2}\right)^\dagger. \quad (5.26)$$

This method of grouping and joint measurement is therefore relatively cheap to implement and allows for significant savings in the number of shots required to complete a VQE, although without changing the overall scaling.

2. Basis rotation grouping

Basis rotation grouping is based on a tensor decomposition of the two-body operators. It reduces the overall number of joint terms to measure in the Hamiltonian, down to a linear number with system size. This same decomposition has also been used to reduce the total gate depth of the full UCCSD ansatz as well as the Trotter steps. It also provides a large improvement in the noise resilience of mappings with high Pauli weight such as Jordan-Wigner.

The price to pay for this is that the measurement has to take place in a different basis for each term, necessitating an additional $O(m)$ gate depth before measurement to implement this orbital rotation for each grouped term of this decomposed Hamiltonian.

The definition of the 'best possible grouping method' is not straightforward. While it is clear that aiming for the lowest number of groups possible is advantageous, it is not the only metric to take into consideration. In particular, it was shown that grouping terms suffers from covariances arising from the joint measurement. This covariance effects increase the sampling noise and as such the total number of measurements required to achieve a given level of precision, which should be taken into consideration as figure of merit for a grouping strategy.

Another cost to consider is the additional quantum noise resulting from the circuit used to rotate the measurement basis, since further resources may be required to mitigate these additional errors.

5.2.4 Error mitigation

As we mentioned before, when the noise of a quantum computer is below a certain threshold and a sufficient number of qubits are available, quantum error correction schemes can be applied to suppress the noise to arbitrarily small levels, however this brings different types of overheads, including large amounts of extra ancilla qubits, fast decoding and communication between quantum and conventional devices.

As an alternative, a series of techniques for mitigating the effects of noise for quantum algorithms running on NISQ hardware have been developed. These techniques have been shown to achieve a reduction in noise levels on expectation value estimates, without requiring the large resources involved in error correction. Such methods are critical for early implementations of the VQE algorithm in order to achieve the required precision for quantum chemistry computation.

These techniques are effective only when used with low-depth circuits such that the total error rate in the circuit is low. However, the additional resources required are much more modest than those for full error correction. In general, these techniques require only multiplicative overhead in the number of measurements needed if the error rate is sufficiently low.

As we are dealing with errors, it becomes necessary to consider mixed states rather than just pure states, thus we have to use the density matrix formalism of quantum

mechanics.

A general output state where each gate is affected by a noise channel \mathcal{N}_i is

$$\rho = \prod_i \mathcal{N}_i \circ \mathcal{U}_i(|0\rangle\langle 0|), \quad (5.27)$$

and the output without the noise is

$$\rho_0 = \prod_i \mathcal{U}_i(|0\rangle\langle 0|), \quad (5.28)$$

where, for a density matrix ρ it is true that $\mathcal{U}_i(\rho) = U_i \rho U_i^\dagger$ and the result of the measurement of a Hermitian observable O on this state is

$$\bar{O} = \text{Tr}(\rho O). \quad (5.29)$$

The error mitigation methods can approximate the noiseless measurement result \bar{O}_0 from the noisy measurement result \bar{O} when the error rate is sufficiently low. It is important to note that error mitigation schemes are not a scalable solution to the problem of noise in quantum hardware. In order to scale up computations to arbitrarily large-size, fault-tolerant, error-corrected quantum computers are required.

1. Extrapolation

The extrapolation method works by intentionally increasing the dominant error rate ϵ_0 by a factor λ , and inferring the error-free result by extrapolation. The technique is based on Richardson extrapolation, which to first order corresponds to linear extrapolation using two points. We could also take a linear or higher order fit with several data points. For the former case, the estimated value of the observable is given by

$$\bar{O}_0^{\text{est}} = \frac{\lambda \bar{O}(\epsilon_0) - \bar{O}(\lambda \epsilon_0)}{\lambda - 1}. \quad (5.30)$$

While this method can improve the accuracy of calculations, it requires additional measurements in order to keep the variance of the measured observable the same as in the non-extrapolated case.

An alternative is the exponential extrapolation which was introduced as a more appropriate extrapolation technique for large quantum circuits.

2. Probabilistic error cancellation

This method works by effectively realizing the inverse of an error channel, \mathcal{N}^{-1} , such that $\mathcal{N}^{-1}[\mathcal{N}(\rho_0)] = \rho_0$.

Because realizing the inverse channel is in general an unphysical process, we use the scheme depicted in Figure 5.4 to effectively realize the inverse channel by focusing only on measurement results.

For example in a depolarizing error channel the unphysical inverse channel is

$$\rho_0 = \mathcal{D}^{-1}(\rho) = \gamma[p_1\rho - p_2(X\rho X + Y\rho Y + Z\rho Z)]. \quad (5.31)$$

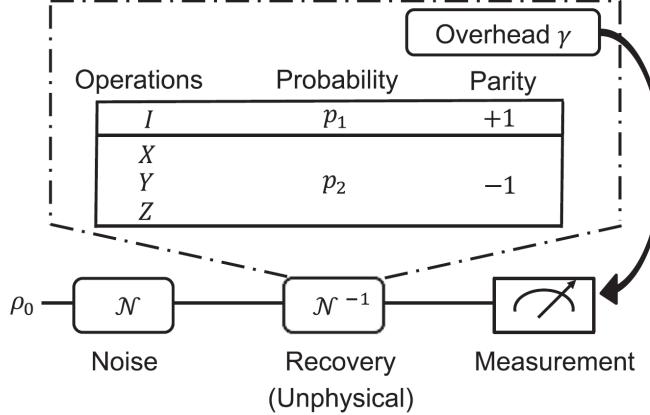


Figure 5.4: Probabilistic error cancellation method for a depolarizing error [46].

However, we can consider and correct its effect on the expectation value \bar{O}_0 ,

$$\bar{O}_0 = \text{Tr}[OD^{-1}(\rho)] \quad (5.32)$$

$$= \gamma[p_1\langle O \rangle_\rho - p_2(\langle O \rangle_{X\rho X} + \langle O \rangle_{Y\rho Y} + \langle O \rangle_{Z\rho Z})] \quad (5.33)$$

$$= \gamma[p_1\langle O \rangle_\rho - p_2(\langle O \rangle_\rho + \langle O \rangle_\rho + \langle O \rangle_\rho)], \quad (5.34)$$

where $\langle O \rangle_\rho = \text{Tr}[O\rho]$. We can therefore measure O , XOX , YOY , and ZOZ and linearly combine the measurement results to effectively realize the inverse channel, thus obtaining the noiseless measurement result \bar{O}_0 .

In practice, it is not possible to exactly measure all of the possible terms resulting from errors if there are many gates in the circuit. Instead, we can consider only the most important terms, which result from a small number of errors occurring. If the error rate is low, then the other terms can be considered negligibly small.

After each single-qubit gate we can apply X , Y or Z operators with probability p_2 or the identity gate with p_1 . We repeat that circuit variant many times to extract the expectation value and multiply the expectation value by $(-1)^{G_p}$, where G_p is the number of additional X , Y or Z gates that were applied in that circuit iteration. We then sum up the values for several circuit variants and multiply by γ to obtain the error mitigated result. This method can also be extended to multiple-qubits gates.

Probabilistic error cancellation requires full knowledge of the noise model associated with each gate. This can be obtained from either process tomography or a combination of process and gate set tomography. The latter approach reduces the effect of errors due to state preparation and measurement.

3. Quantum subspace expansion

The quantum subspace expansion (QSE) can mitigate errors in the VQE, in addition to calculating the excited energy eigenstates. This method is most effective at correcting systematic errors, but it can also suppress some stochastic errors. Suppose that we use the VQE to find an approximate ground state $|\tilde{E}_0\rangle$. Noise may cause this state to deviate from the true ground state $|E_0\rangle$. For example, if $|\tilde{E}_0\rangle = X_1|E_0\rangle$ we can simply apply an X_1 gate to recover the

correct ground state.

However, as we do not know which errors have occurred, we can instead consider an expansion in the subspace $\{|P_i \tilde{E}_0\}$, where P_i are matrices belonging to the Pauli group. Then one can measure the matrix representation of the Hamiltonian in the subspace

$$H_{ij}^{QSE} = \langle \tilde{E}_0 | P_i H P_j | \tilde{E}_0 \rangle. \quad (5.35)$$

As the subspace states are not orthogonal to each other we should also measure the overlap matrix

$$S_{ij}^{QSE} = \langle \tilde{E}_0 | P_i P_j | \tilde{E}_0 \rangle. \quad (5.36)$$

Then, by solving the generalized eigenvalue problem

$$H^{QSE} C = S^{QSE} C E, \quad (5.37)$$

with a matrix of eigenvectors C and diagonal matrix of eigenvalues E , we can get the error mitigated spectrum of the Hamiltonian. A small number of Pauli group operators are typically considered in order to minimize the required number of measurements [31].

5.3 Computational requirements

Analogously to Chapter 4 we give an estimate of the computational requirements for the VQE with one example.

Again the algorithmic parameters that we consider are:

- **circuit width** (n_q)
- **circuit depth** (n_g)
- **repetitions** (n_{rep})
- **total scaling** ($n_t = n_g \cdot n_{rep}$)

The number of qubits needed to represent the relevant quantum states is generally equal to the number of spin-orbital,

$$n_q = O(m). \quad (5.38)$$

This is the case for both the Jordan-Wigner and Bravyi-Kitaev transformations. It is, however, typically possible to reduce this number slightly by conserving symmetries of the chemical system.

The number of times the circuit must be repeated will be the product of two factors: the number of circuit applications or shots, S , required to obtain a single estimate of the Hamiltonian expectation and the number of Hamiltonian expectations, n_h , required in order to optimize the parameters so that

$$n_{rep} = S \cdot n_h. \quad (5.39)$$

To find these we first consider the ansatz.

The choice of the circuit plays a key role in determining the performance of a VQE calculation. The ideal ansatz would:

- **Enable preparation of a state close to the true ground state;**
- **Require as few parameters as possible**, so as to minimise the time required to perform the classical optimization;
- **Use as few quantum computational resources as possible.**

In general the ansatz circuit depth will depend on the accuracy, ϵ , as a deeper circuit will typically allow a state closer to the true ground state to be prepared. However, it is difficult to quantify the relationship between n_g and ϵ . Here, we first consider a fixed ansatz: the UCCSD ansatz.

The key property of the ansatz that affects the VQE calculation time is the number of parameters, n_p . For the UCCSD ansatz, this is

$$n_p = O(n_e^2(m - n_e)^2) \sim O(m^4), \quad (5.40)$$

which is the scaling of the number of $t_{i,j,\alpha,\beta}$ parameters, where n_e indicates the number of electrons considered.

The number of Hamiltonian expectations required in a particular VQE calculation is difficult to know in advance as it will depend on the shape of the ansatz parameter space. Here we assume that this number, n_h , is simply given by

$$n_h = n_p. \quad (5.41)$$

This, for example, could arise if the optimizer need only to look in each parameter direction once, perhaps to verify that a minimum has already been found. Needing any fewer evaluations would imply that it was known before the calculations occurred that some parameters were not needed in the ansatz. Typical calculations will require more evaluations than this.

The number of applications of the ansatz circuit needed depends on the form of the Hamiltonian and the particular quantum state. Measurements of Pauli operators can easily be obtained, thus, assuming that each one of these is obtained separately, the number of times the ansatz circuit must be performed is given by

$$S = \left(\frac{1}{\epsilon} \sum_i |w_i| \sqrt{Var[P_i]} \right)^2, \quad (5.42)$$

where ϵ is the desired error in the expectation estimate, w_i is the coefficient of the Pauli string P_i and

$$Var[P_i] = 1 - \langle \psi(\vec{\theta}) | P_i | \psi(\vec{\theta}) \rangle. \quad (5.43)$$

The maximum value of each variance is 1 and so

$$S \leq \left(\frac{E_{max}}{\epsilon} \right)^2, \quad (5.44)$$

as we saw before, where again $E_{max} = \sum_i |w_i|$.

In the following we take the equality in this expression.

We note that it is possible to reduce this through several different measuring strategies, such as those described before. It is possible to improve upon the assumption

that we measure each Pauli individually by, for example, measuring commuting Pauli operators simultaneously or factorising the two-electron integral tensor. Such methods reduce the overall number of measurements required whilst retaining the scaling in $\frac{1}{\epsilon^2}$. This scaling can also be improved using QPE-inspired methods at the cost of an increased circuit depth. However, such increased depths are unlikely to be possible in the NISQ era, before error correction is available.

For the circuit depth we assume that it is possible to perform n_q parameters per layer of gates and so

$$n_g = O(m^3) \quad (5.45)$$

for the UCCSD ansatz.

In conclusion, the algorithmic parameters scale as [8]

$$n_q^{VQE} = O(m), \quad n_g^{VQE} = O(m^3), \quad n_{rep}^{VQE} = O\left(\frac{m^4 E_{max}^2}{\epsilon^2}\right) \quad (5.46)$$

Likewise QPE we consider E_{max} scaling between $O(m)$ and $O(m^3)$ [25].

Thus the total scaling for the VQE algorithm is

$$n_t^{VQE} = n_g^{VQE} \cdot n_{rep}^{VQE} = O\left(\frac{E_{max} m^7}{\epsilon^2}\right). \quad (5.47)$$

Moreover, J. Tilly et al. [46] presented the sequence of components for the VQE algorithm which offer the most promising scaling without compromising excessively on accuracy for molecular systems and lattice models.

The key distinctive factor separating molecular systems and lattice models is that the former makes no assumption on the range and type of interaction between the fermionic modes, beyond it being a two-body interaction, while the latter usually has a simplified and parameterized form which often only connects fermionic modes following a nearest-neighbor lattice structure and/or features a lower effective rank of interactions.

Here the list of components:

1. Hamiltonian construction

Molecular system: Second quantization - $O(m^4)$ Hamiltonian terms and $O(m)$ number of qubits;

Lattice models: Second quantization - $O(m^4)$ Hamiltonian terms and $O(m)$ number of qubits;

2. Fermion to spin encoding

Molecular system: Ternary tree encoding - $O(n_q^4)$ operators and $O(\log_3(2n_q))$ Pauli weight;

Lattice models: Generalized superfast encoding - $O(md/2)$ qubits, with d the fermionic-interaction graph maximum degree, $O(ND)$ operators, with D the lattice dimension, and $O(\log_2(d))$ Pauli weight;

3. Ansatz

Molecular system: k-UpCCGSD - $O(kn_q)$ circuit depth and $O(kn_q^2)$ parameters;

Lattice models: Hamiltonian variational ansatz (HVA) - $O(k\tilde{C})$ circuit depth and parameters, with \tilde{C} the number of commutative groups in the Hamiltonian;

4. Grouping and measurement strategy

Molecular system: Decomposed interactions - $O(n_q)$ operators to measure and $O(n_q/2)$ additional basis rotation circuit depth;

Lattice models: Qubit-wise commutation - $O(1)$ operators to measure and additional basis rotation circuit depth;

5. Optimizer

Molecular system: Rotosolve - requires sampling three values for each parameter at each step;

Lattice models: Rotosolve - requires sampling three values for each parameter at each step;

6. Error mitigation strategy

Molecular system: Symmetry verification and extrapolation based methods - exponential with respect to the circuit depth;

Lattice models: Symmetry verification and extrapolation based methods - exponential with respect to the circuit depth;

Now we describe the total scaling of such a sequence of components.

The computation of the expectation value of a single operator at a precision ϵ , which would cover the same range as chemical accuracy, requires $O(1/\epsilon^2)$ repetitions of the ansatz. If the k-UpCCGSD ansatz is chosen this scales as $O(km)$ [26], while choosing to use the decomposed interactions requires $O(m)$ different operators to be measured and therefore a gate depth of $O(m)$ for rotation to the joint measurement basis, resulting in a total scaling for a single estimation of the entire Hamiltonian of $O(km^2/\epsilon^2)$.

There are $O(km^2)$ parameters in the k-UpCCGSD ansatz [26], hence this represents the cost scaling of updating each parameter using the Rotosolve optimizer. As this optimizer is not parallelizable one may prefer to use a different method if sufficient sets of qubits are available. Overall, this gives a total scaling for one iteration of the VQE of

$$n_t^{VQE} = O\left(\frac{k^2 m^4}{\epsilon^2}\right) \quad (5.48)$$

without parallelization, and

$$n_t^{VQE} = O(km) \quad (5.49)$$

with full parallelization for the circuit depth.

This example shows how different ansätze can lead to very different scalings for and how research on better components of an algorithm can improve scaling even by several orders of magnitude.

A perfect parallelization would require $O(km^3/\epsilon^2)$ sets of $O(m)$ qubits. Note that while qubits within one set need to be entangled for the course of a single measurement, there is no requirement for entanglement between qubits of different sets of

parallel quantum computers nodes. The sets of qubits can therefore be either all within one quantum computer, or else also distributed across different separated quantum computers.

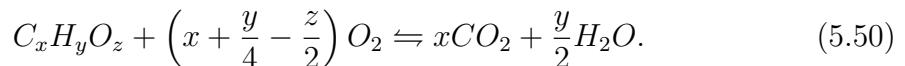
So far we have only considered the scaling of one iteration. It is still an open research question how the number of iterations required to achieve convergence scales with system size for the VQE. This depends on numerous factors, including the ansatz, the optimizer used and the system studied. One important point to note is that convergence tends to be rapid at the beginning of the optimization process, with large gradients that require only a low number of shots to be computed accurately enough to progress. It becomes more challenging close to the optimum, where gradients are smaller, requiring a larger number of shots to continue the optimization. As such the last few iterations of the VQE are likely orders of magnitude more expensive than the rest of the optimization, if the algorithm is implemented efficiently.

5.3.1 Example

Several studies estimated the resource requirements for NISQ approaches. The majority stand on the statement that the VQE method leads to a number of measurements and repetitions needed to simulate industrially-relevant systems that are prohibitively large. However, we discuss more on this in Chapter 6.

One example is the study by J. Gonthier et al. [17] who, in 2020, estimated the number of qubits, number of measurements and total runtime required for calculating combustion energies for small organic molecules to within chemical accuracy with a single VQE energy evaluation. These estimates consider the so-called 'frozen natural orbitals' (FNO) as well as measurement reduction techniques such as Hamiltonian term grouping, application of fermionic marginal constraints and low-rank factorization of the Hamiltonian.

They considered a benchmark set of organic molecules, shown in Figure 5.5, and studied their combustion reactions, whose general formula is:



First, they benchmarked classical chemistry methods reaching the conclusion that the performance of common methods that can routinely be applied to larger systems is insufficient to reach chemical accuracy, although it improves with larger basis set, and to get even better results the very resource-demanding coupled cluster method is needed.

The quantum computation part was executed on the Zapata Computing's Orquestra workflow management platform.

For the scaling in depth only the number of measurements is considered.

The formula to estimate this number, M , is

$$M = \frac{K}{\epsilon^2}, \quad (5.51)$$

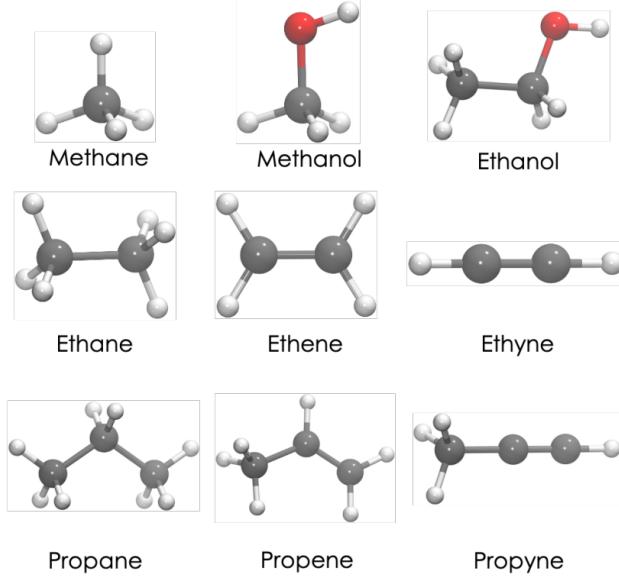


Figure 5.5: Benchmark set of organic molecules [17].

where K is the Hamiltonian variance and is defined as

$$K = \left(\sum_C \sqrt{\sum_{\alpha, \beta \in C} w_\alpha w_\beta \text{Covar}(P_\alpha, P_\beta)} \right)^2, \quad (5.52)$$

which is analogous to eq. (5.42), with P_i the Pauli operators and w_i their coefficients. They evaluated K for QWC grouping and basis rotation approach with two different basis for the Hamiltonian (canonical orbitals or FNOs) and two different estimates for the variances (upper bounds or CISD), giving a total of 4 different settings for each grouping method.

They ran computations for all molecules in the set and also included H_2O and CO_2 , that are necessary for computing combustion energies.

For each molecule they computed different active spaces with an integer number of qubits per active electron up to a total of 80 qubits. Then they fit the results to a power law for each grouping method:

$$K = a(n_q)^b. \quad (5.53)$$

The results are shown in Figure 5.6.

The number of terms in the quantum chemistry Hamiltonian scales as $O(n_q^4)$, as we have seen before. However, the QWC grouping method with optimal measurement allocation approximately scales between $O(n_q^5)$ and $O(n_q^6)$. Thus, the observed scaling for QWC grouping only constitutes a modest improvement over the estimated upper bound of $O(n_q^6)$ for scaling without grouping. Basis rotation grouping offers significantly better scaling. This varies between $O(n_q^{2.3})$ and $O(n_q^{3.6})$, a very significant improvement compared to QWC grouping results. In addition, the effect of this improved scaling is already beneficial at low number of qubits, so that QWC grouping never appears advantageous in the computed data.

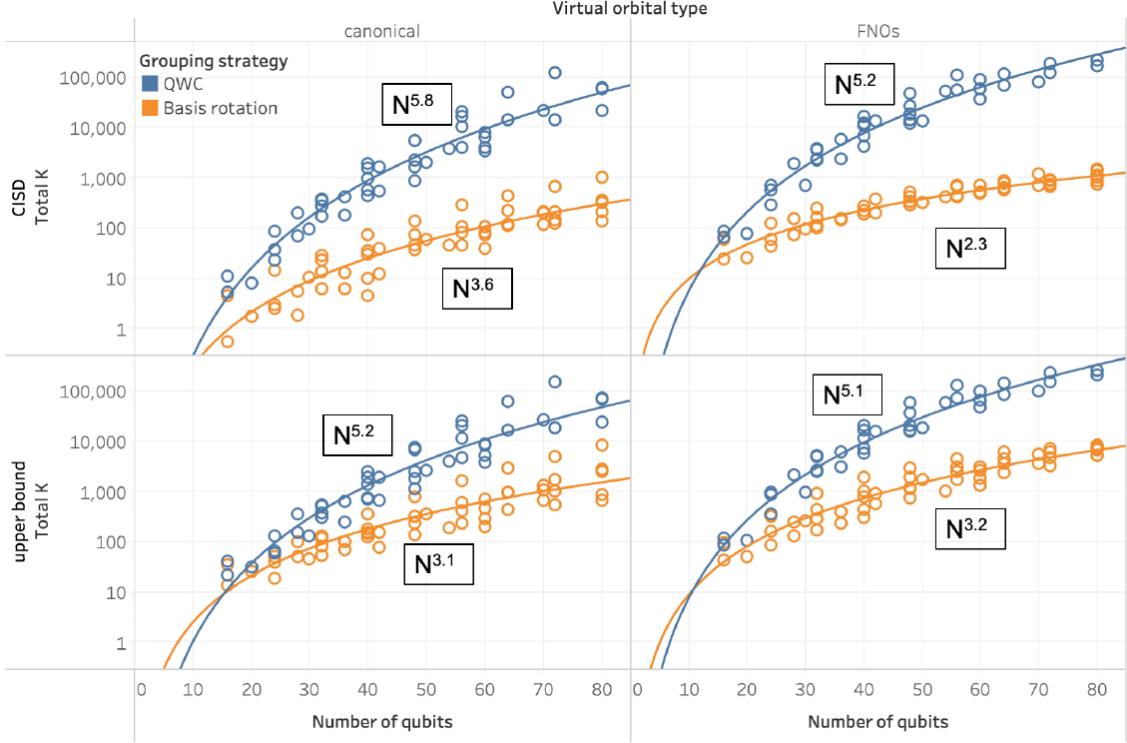


Figure 5.6: Values of K for QWC and basis rotation grouping, with $K = a(n_q)^b$ [17].

For the number of qubits they used $n_q \approx 13n_{el}$, where n_{el} is the number of electrons. This comes from using FNO as a basis set with which to model the atomic orbitals.

Finally, for the runtime they assumed to use a shallow hardware-efficient ansatz, a linear connectivity of the qubit array, in which case a single layer is defined as the circuit of depth 2 that entangles every neighboring pair of qubits, and assumed that the number of layers needed to reach the ground state energy scales linearly with the number of qubits.

Considering the runtime dominated by execution times of two-qubit gates, assumed to be 100 ns (a value on the faster side of current superconducting gate times), the final formula used to obtain runtimes t in seconds is

$$t = 10^{-7}M(5n_q - 3). \quad (5.54)$$

The results for the different molecules are reported in Figure 5.7, with $\epsilon = 0.5$ mHa.

Moreover, they highlighted that this is the time necessary for a single energy evaluation and that running the full VQE algorithm involves optimizing the circuit parameters, which requires at least a few dozen to hundreds of iterations even with excellent optimizers. Hence, the total VQE runtime would be about a month for the smallest molecules in this test set.

Molecule	H ₂ O	CO ₂	CH ₄	CH ₄ O	C ₂ H ₆	C ₂ H ₄	C ₂ H ₂	C ₂ H ₆ O	C ₃ H ₈	C ₃ H ₆	C ₃ H ₄
n _{el}	8	16	8	14	14	12	10	20	20	18	16
n _q	104	208	104	182	182	156	130	260	260	234	208
K · 10 ⁻³	1.9	16	1.6	8.4	8.5	6.6	3.1	24	16	23	18
M · 10 ⁻⁹	3.9	32	3.2	17	17	13	6.2	48	31	46	36
t (days)	2.3	39	1.9	18	18	12	4.6	71	47	62	44

Figure 5.7: Estimated runtimes for a single energy evaluation [17].

Chapter 6

Computational advantage

6.1 Benchmarking quantum computers

As increasingly capable quantum computers are built and deployed, it is important to develop benchmarks that track the performance of these systems. The main goals of these benchmarks are to identify the necessary technology for each specific application and to link the debate on quantum advantage, i.e. where a user can run a quantum program to and a solution faster, cheaper or more accurately than classical computing alone, to the specific hardware that can achieve such calculation.

Unlike classical computer systems, where instructions are executed directly by a CPU, the Quantum Processing Unit (QPU), which is the combination of the control electronics and quantum memory, is supported by a classical runtime system for converting the circuits into a form consumable by the QPU and then retrieving results for further processing. Performance on actual applications depends on the performance of the complete system and, as such, any performance metric must holistically consider all of the components.

The performance of a quantum computer is governed by three key factors [48]:

- **Scale:**

It determines the size of problem that can be encoded and solved. The technological parameter that defines the scale of a quantum computer is the number of qubits;

- **Quality:**

It determines the size of a quantum circuit that can be faithfully executed. The technological parameter for the quality of a quantum computer is the quantum volume (QV);

- **Speed:**

It is related to the number of primitive circuits that the quantum computing system can execute per unit of time. The technological parameter that defines the speed of a quantum computer is the circuit layer operations per second (CLOPS);

6.1.1 Number of qubits

The number of qubits determines the amount of information that can be encoded in a quantum computer for computation, which caps the size of solvable problems. In quantum simulation the number of qubits sets the size of the basis set that represents each electron wave function in a molecule and therefore the size of the molecules that can be simulated. Number of qubits can also be used as a resource to improve the other two metrics of quality and speed. For example, auxiliary qubits can often be used to reduce the depth of circuits and increase their fidelity. Extra qubits can also be used in multiprogramming of QPUs to increase their circuit processing speed. For most quantum computing platforms increasing the number of qubits, i.e. the scalability, relies heavily on the available materials and fabrication technologies developed from the semiconducting industry (superconducting, semiconducting, ion traps and photonics qubit platforms). While all quantum hardware platforms have challenges in scaling, superconducting qubits are making fast progress to scale beyond 100 qubits.

The performance metrics to choose depend on the level of complexity one wants to capture with them. A useful way of thinking about it is a benchmarking pyramid, as shown in Figure 6.1.

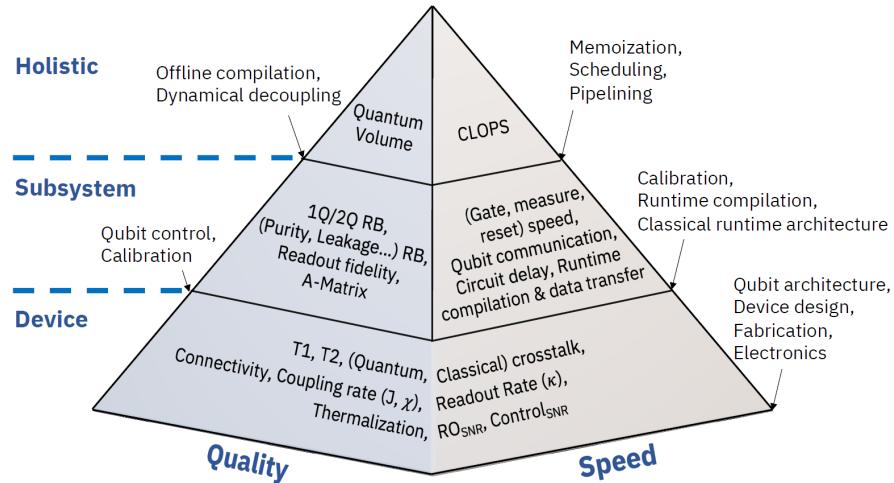


Figure 6.1: Benchmarking pyramid. Higher-level benchmarks capture more complexity but less specificity [48].

6.1.2 Quantum volume

Quantum volume (QV) indicates how faithfully a quantum circuit can be implemented in a quantum computing system. A QV layer is defined as one layer of permutation among qubits and one layer of pair-wise random SU(4) 2-qubit unitary gates. The QV is defined by the width or number of QV layers of the largest random square circuit (with width equal to the number of layers) that a quantum processor can successfully run. Note that, when a QV circuit is compiled to the native gate

set of a particular QPU, the circuit depth of the compiled circuit will typically be much larger than the number of QV layers as the abstract permutations and SU(4) unitaries may be each decomposed into multiple native gates.

QV measurement starts with executing a square circuit of width N and then compares the measurement results from the heavy output states (the states with probabilities higher than the median of probabilities of all output states) with the ideal results from simulation. The largest N -qubits square circuit that can run successfully to produce more than 2/3 of heavy outputs determines the quantum volume on a quantum computing system, given by $2N$.

QV is sensitive to coherence, gate fidelity and measurement fidelity, which are hardware properties of a quantum processor. It is also influenced by connectivity and compilers, which can make circuits efficient to minimize the effect of decoherence.

QV is a holistic metric because it cannot be improved by just improving one aspect of the system, but rather requires all parts of the system to be improved in a synergistic manner. It has been adopted widely by research and industry and has been reported for several ion trap and superconducting quantum computers.

6.1.3 Circuit Layer Operations per Second Benchmark

Circuit layer operations per second (CLOPS) is a measure correlated with how many QV circuits a QPU can execute per unit of time.

A. Wack et al. [48] formally defined as: the number of QV layers executed per second using a set of parameterized QV circuits, where each QV circuit has $D = \log_2$ QV layers. Circuit execution time includes updating parameters to the circuit, submitting the job to the QPU, executing on the QPU and sending results back to be processed. CLOPS is then calculated as the total number of QV layers executed divided by the total execution time.

The CLOPS benchmark is designed to allow the system to leverage all of the quantum resources on a device to run a collection of circuits as fast as possible as well as to stress all parts of the execution pipeline. This includes data transfer of circuits and results, runtime compilation (lowering basis-gate level circuits to control electronics instructions), latencies in loading control electronics, initialization of control electronics, gate times, measurement times, reset time of qubits, delays between circuits, processing results and parameterized updates. The generation of random parameters from a seed constructed from the shots of the previous execution simulates the parameter updates in iterative quantum algorithms.

Including all of these parameters in the benchmark ensures that all aspects of the system are included to generate a meaningful comparison between systems. Physical qubit architectures may effect the repetition time, gate times, reset times and measurement times and can vary significantly across technologies. For example, the repetition rate and the gate rate of superconducting qubits can be orders of magnitude faster than the ones of ion trap qubits which significantly impacts the CLOPS. Similarly, software components such as runtime compilation, orchestration of the control electronics, etc. are all aspects that are necessary in current architectures to run already compiled programs for the QPU and have considerable impact on the overall performance. Finally the CLOPS is also impacted by how efficiently circuits

can be delivered to the system for execution and results returned to the user-space application.

6.1.4 Today's quantum computers

Quantum computers already available come with a series of technological parameters useful to describe and improve them.

Here we take IBM quantum computers [21] as an example to show them.

The most fundamental technological parameters one can refer to are:

- **Number of qubits, QV and CLOPS:** defined before;
- **Connectivity:** how the links between the single qubits in the processor are prepared, e.g. linear or full connectivity. It is also shown by a map of the processor;
- T_1 : the longitudinal relaxation time of the higher energy $|1\rangle$ state into ground state $|0\rangle$, i.e. the classical state lifetime;
- T_2 : the transverse relaxation time of superposition states such as $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$, i.e. the dephasing time.
 T_1 and T_2 are generally referred to as 'decoherence time';
- **Frequency:** the qubits from IBM are superconducting qubits, thus interactions with these qubits are performed via microwave resonators. To address each different qubit separately each qubits resonator has a different frequency. This value is defined as the energy difference between the $|0\rangle$ and $|1\rangle$ states;
- **Readout assignment error:** the measure of how well the measurement performs, more formally, the probability of a measurement returning the wrong value;
- **Prob meas0 prep1:** the probability that a measurement returns 0 immediately after preparing the $|1\rangle$ state. A combination of faulty preparation and faulty measurement. These are also often referred to as SPAM (state-preparation-and-measurement) errors;
- **Prob meas1 prep0:** likewise the previous one, but for the $|0\rangle$ state;
- **Readout length (ns):** the time it takes to perform a measurement (in ns). This is one of the main limiting factors of current quantum hardware as these times are often considerably longer than the T_2 , which makes intermediate measurements, i.e. measurements that are not at the end of a circuit, practically unfeasible;
- **ID error:** the measure of the error induced by having the qubit idle for a typical gate time. The measure used is the one returned by randomized benchmarking, which closely resembles the process fidelity;

- **\sqrt{X} error:** the measure of error induced by applying the \sqrt{X} gate. Average over all different qubits;
- **Single-qubit Pauli X error:** likewise the previous one, but for the X gate;
- **CNOT error:** the error of the only two-qubits gate natively implementable on the devices. Average over all possible CNOT gates;
- **Gate time (ns):** the time it takes to perform a CNOT gate.

As of today IBM has released a 127-qubits processor called `ibm_washington`, but, as we have described before, this fact does not tell the whole story, in fact this processor has a QV of 64 and CLOPS of 850, while a 27-qubits processor called `ibmq_montreal` has a QV of 128 and CLOPS of 2000. Moreover, a 7-qubits processor called `ibm_perth`, built with the same technology, has CLOPS of 2900, one of the highest already available. The highest value of QV, as of today, has been reached by the company Quantinuum with the System Model H1-2 which presents a QV of 4096.

A more precise analysis on the relations between different technological parameters, especially the number of qubits and CLOPS, can be made by breaking down the time to execute the benchmark into different areas [48].

These and many other companies frequently present roadmaps for the technology advancements in the quantum computers they build. Some focus more on scale up rapidly the number of qubits to run complex calculations, some focus on achieving a high quality before the progress in scale takes place, i.e. build logical qubits by implementation of error-correction codes and high QV devices.

Below we show the roadmaps from IBM and Google and a general one representing different companies with different technologies. This is a useful instrument to carry out an analysis and a forecast of the applications that will be available in the future years.

6.2 Algorithms comparison

6.2.1 QPE and VQE

By comparing the formulas in eq. (4.16) and (4.17) and eq. (5.46) and (5.47), or more precisely eq. (5.48), we can see that the QPE algorithm is asymptotically more efficient than the VQE.

It scales linearly, with respect to the number of spatial orbitals, in both width and depth. It allows to achieve a higher precision with a lower computational cost in terms of depth, since it scales as $O(\frac{1}{\epsilon})$, while the VQE scales as $O(\frac{1}{\epsilon^2})$. The number of repetitions is a constant value that depends on the chemical system and this is the best result one can hope to achieve.

This leads to the conclusion that QPE is most likely the best quantum algorithm to simulate large-size molecules, i.e. molecules with hundreds or thousands of atoms. However, as already expressed throughout the thesis, QPE also requires to work with

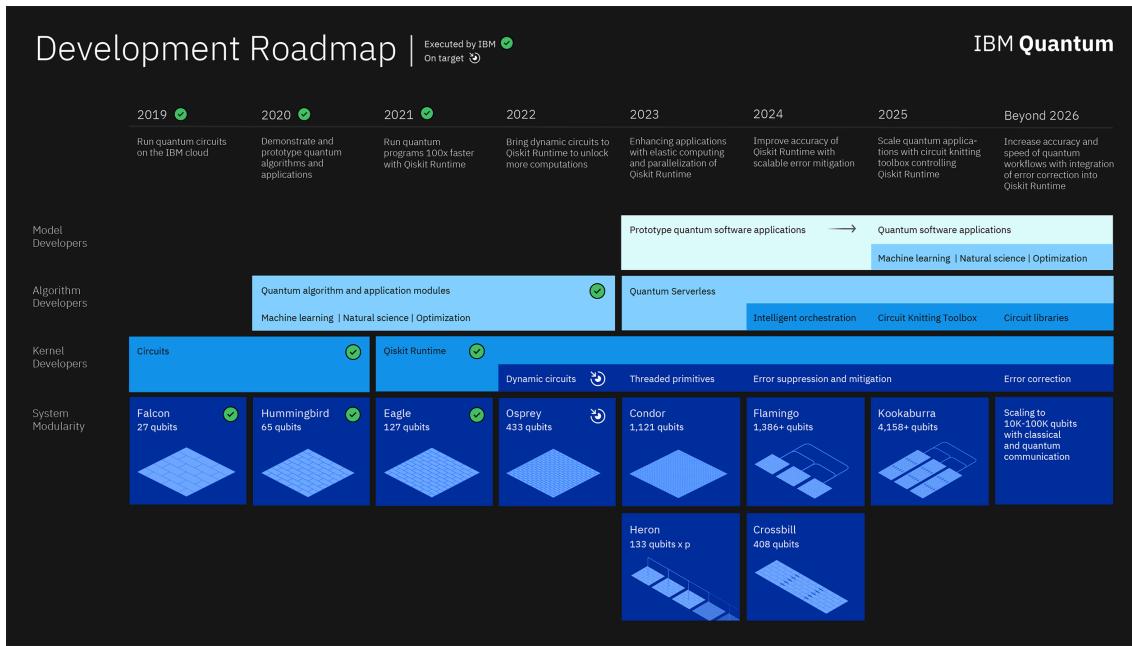


Figure 6.2: IBM quantum roadmap.

logical qubits or equivalently fault-tolerant devices, which translates in long coherence time and number of physical qubits even for simulation of small-size molecules. Furthermore, a good approximation of the target wave function is needed, i.e. the fidelity of the input state to the unknown target eigenstate approaches one, thus using a randomized state as input is not an option as its expected fidelity to the target eigenstate approaches zero exponentially in the system size, resulting in QPE becoming exponentially costly with imperfect input state preparation.

The core component of QPE, however, is efficient Hamiltonian simulation and it is fundamental to study alternative methods to Trotterization that perform better in order to lower the computational requirements, especially in the depth of the circuit, i.e. gate count. In addition to that, researches on scalable quantum hardware, be it superconducting circuits, ion traps or other technologies, will enable increasingly large circuits both for calculation and error-correction schemes.

The VQE, on the other side, trades off the depth and the number of qubits required under QPE with a higher number of measurements and repetitions of the circuit as well as with the constraints of an approximate ansatz for the state. Although this could result in a more efficient simulation for small-size molecules it is not a viable path, in other words, the VQE is not an efficiently scalable algorithm. It is important to note that much lower number of shots would be sufficient to progress the initial part of the optimization and a high number is only required in the last iterations of a VQE close to convergence to reach chemical precision (however this number may need to be much higher in case of barren plateaus).

A solution which needs to be researched more thoroughly is the potential parallelization of the VQE. This is because parallelism of the VQE offers a direct way to

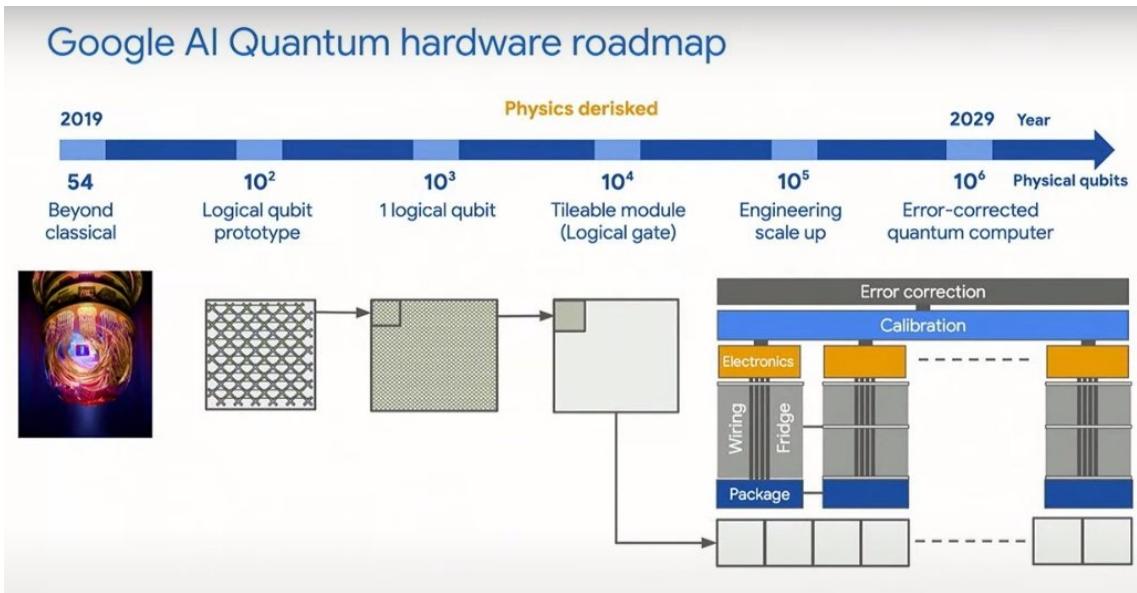


Figure 6.3: Google quantum roadmap.

convert runtime cost into hardware cost by splitting the shots required onto different sets of qubits, which can be arranged in different threads on a single quantum computer or multiple disconnected quantum computers.

However, there are many caveats to the potential of this method. First, parallel quantum computing suffer from communication overheads and secondly, parallelization could result in higher noise levels.

While many other factors affect the overall time scaling of both methods, this point illustrates the asymptotic efficiency of QPE compared to VQE assuming access to fault-tolerant quantum computers, but also the resource efficiency of VQE over QPE for NISQ-era devices.

The frontier between NISQ and FTQC is blurry and as pointed out by Wang et al. [49] so is the frontier between VQE and QPE. They presented an interpolation between the two algorithms labeled accelerated VQE (or α -VQE), which uses smaller scale QPE calculations as subroutines for the VQE. This method introduces a parameter $\alpha \in [0, 1]$ which allows the tuning of the circuit depth $O(1/\epsilon^\alpha)$ and the number of samples $O(1/\epsilon^{2(1-\alpha)})$. One recovers the QPE scaling if $\alpha = 1$ and the VQE scaling if $\alpha = 0$.

As a general perspective, rather than being mutually exclusive methods for solving an electronic structure problem, VQE and QPE are likely to provide the most benefit when combined as complementary approaches, offering algorithmic flexibility that can be adjusted depending on the progress of quantum hardware. This means, for example, that VQE can be used to implement the state preparation task of the QPE algorithm, this would make the overlap between the input state and the target eigenstate more prominent, it raises the probability of calculating the correct expected eigenvalue and correspondingly it lowers the number of repetitions of the entire QPE method, as we saw mathematically from eq. (4.16) and (4.17).

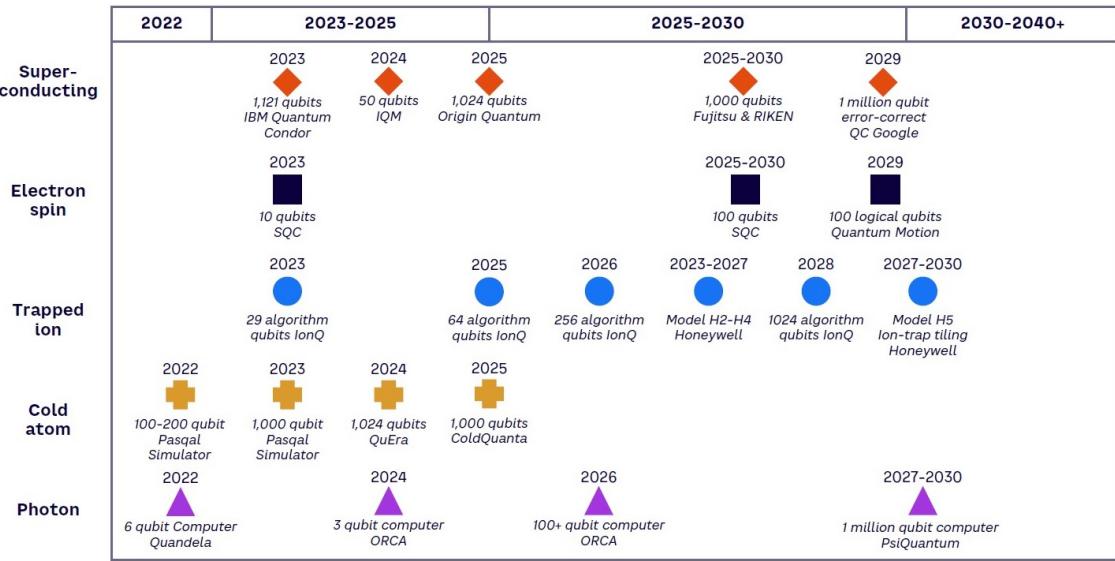


Figure 6.4: Quantum computing prototypes announced on different roadmaps [39].

6.2.2 Classical and quantum

Most conventional approaches for high accuracy calculation of ground state energetic properties of molecular systems rely on wave function approximations, which we presented in Chapter 1. These show a similarity with VQE, in which the focus of the algorithm is to best approximate the wave function by defining the single basis functions, which determines the resolution and the representation of the system.

This means that the most likely competitors for short to medium term applications of VQE are accurate wave function approaches, e.g. CC and CI methods, which, as we have described, scale as high polynomials or even combinatorially, but they are still able to access systems of comparable size.

The largest conventional CI calculations which have been reported in the literature are proof-of-principle studies. They include [14]:

1. A calculation involving a complete active space (CAS), i.e. modelling the electrons in the partially occupied orbitals with all combinations of the Slater determinants with 20 electrons in 20 spatial orbitals (40 spin-orbitals), realized within the framework of the MCSCF method on a chromium trimer, corresponding to approximately $4.2 \cdot 10^9$ single determinants (SDs);
2. A single point CASSCF calculation on a pentacene molecule with 22 electrons in 22 spatial orbitals (44 spin-orbitals), corresponding to approximately $5.0 \cdot 10^{11}$ SDs;
3. A single iteration of the iterative CI algorithm for a chromium tetramer with 24 electrons in 24 spatial orbitals (48 spin-orbitals), $\sim 7.3 \cdot 10^{12}$ SDs.

All these CI calculations utilized the 6-31G* basis set. To put the number of the SDs in chemical context a FCI calculation on the propene molecule in the minimal

STO-3G and the larger, but still very small, 6-31G basis set would correspond to 24 electrons in 21 and 39 spatial orbitals (42 and 78 spin-orbitals), respectively. The frequently used frozen core (FC) approximation would reduce the number of active electrons in propene to 18, but already the next homolog, 2-butene, will have 24 active electrons in the FC approximation and push against the limits of the feasible FCI calculations.

As for the CC method, describes in Chapter 1 that the CCSD(T) method treat dynamic correlation accurately and shows size extensivity (correct scaling with the system size) and size consistency (separability of energies for non-interacting fragments).

Among the largest CC calculations reported in the literature are [14]:

1. A CCSDT(Q)/cc-pVTZ single point energy calculation on benzene in the FC approximation (30 electrons, 264 basis functions) which corresponds to $\sim 3.1 \cdot 10^9$ single, double and triple as well as $\sim 2.2 \cdot 10^{12}$ perturbative quadruple t-amplitudes;
2. The largest conventional CCSD(T) calculation ever performed is that by Yoo and co-workers on a $(\text{H}_2\text{O})_{17}$ cluster in the aug-cc-pVTZ basis set, which corresponds to 128 electrons in the FC approximation and 1564 orbitals;
3. A similar $(\text{H}_2\text{O})_{16}$ calculation had to be executed on 120 000 computer cores and took more than 3 hours;
4. Another notable CCSD(T) effort is that of Gyevi-Nagy and co-workers in which the energy of 2-aminobenzophenone (ABP, $\text{C}_{13}\text{H}_{11}\text{NO}$) was computed in the large def2-QZVPPD basis set. That calculation, utilizing the density-fitting approximation, correlated 90 non-FC electrons among 1569 orbitals and was completed on 224 computer cores in 68 hours.

VQE has an advantage over classical methods since in its circuit it implements an ansatz that can approximate a wide range of wave functions. To put it another way, it analyzes a big portion of the Hilbert space derived by the qubits register, which ultimately corresponds to the number orbitals of the chemical compounds. An example of this result is the UCC ansatz, which was already described by Peruzzo et al. [36] as a modelization that has no efficient implementation on a classical computer but it has one on a quantum computer.

VQE does so with a scaling in computational requirements which competes with the one of high accuracy classical methods, both in width and in depth. We can see that from eq. (1.29) and eq. (5.46) and (5.47) or eq. (5.48) or otherwise from Figure 2.4.

Taking the results from Gonthier et al. [17], described in Chapter 5, focusing on the scaling of b from Figure 5.6 and omitting the prefactor a , the results for the basis rotation grouping technique suggest that VQE has the potential to scale better with system size than methods such as CC.

The expressivity and the better scaling can result in a computational advantage, but we need to consider two restrictions to define this advantage. First, VQE must

demonstrate similar or higher accuracy than any conventional method but with lower computational time-to-solution. This condition takes into account possible limitations due to hardware runtime, potentially resulting in a large prefactor for VQE computations. If the VQE has better asymptotic scaling than conventional methods, but a large prefactor, this means an advantage could only be achieved in the asymptotic regime of very large systems, with runtime possibly too large for VQE to be realistically usable. This would make it difficult to demonstrate quantum advantage for practical moderately sized systems. The second condition is to achieve at least as good accuracy, with faster compute time, for a system of sufficient complexity to accurately model a real problem of physical and chemical relevance. This involves demonstrations on systems where the approximation error in defining the specific Hamiltonian for the original problem is of smaller magnitude than its solution using the VQE. This could be as simple as ensuring that basis sets are sufficiently saturated or that the complexity of the interactions with a wider system were sufficiently resolved. For instance, consider computing the energy of a series of protein-ligand complexes: even if the VQE achieves better accuracy in lower computation time, it is not guaranteed that these accuracy gains lead to a practical advantage.

These methods all aim to simulate small-size molecules with high accuracy. If instead we want to analyze large-size and industrially relevant compounds, especially in the case of complex heterogeneous materials, the great majority of first-principles simulations are conducted using density functional theory, which, as shown in Chapter 2, is in principle exact but in practice requires approximations to enable calculations. Within its various approximations DFT has been extremely successful in predicting numerous properties of solids, liquids, and molecules and in providing key interpretations to a variety of experimental results. However, it is often inadequate to describe strongly-correlated electronic states.

The boundary between strongly correlated and weakly correlated is imprecise from a quantitative point of view, but one may use the definition that a weakly-correlated system is one for which chemical accuracy can be obtained by coupled-cluster theory with single and double excitations and a perturbative treatment of connected triple excitations, CCSD(T), and a strongly-correlated system is one that requires higher excitations for chemical accuracy. This issue is particularly prevalent when two or more configurations are nearly degenerate. For this reason, strong correlation is often called near-degeneracy correlation.

For strongly-correlated systems the main difficulty lies in the size of the active spaces required for correctly describing them. This is because individual electron-electron interactions determine the behaviour of the whole system and they lead to complex structures that need high accuracy on a local scale, while methods like DFT or force fields (a method that model forces and bonds between atoms in a classical Newtonian parametrization) give results that are too approximative. This is particularly true for materials science, where systems, i.e. materials, consists of hundreds or thousands of atoms arranged in lattices which mostly interact locally through individual electrons or pair of electrons, e.g. superconductors.

Moreover, as we mentioned before, DFT can not be systematically improved, but one has to define a different functional for each simulated system. This can make the method impractical because it introduces empirical parameters that bind the result of the simulation to the experimental capabilities, which is the opposite of why a simulation method is used, that is, for the inference of chemical properties. In this condition QPE shows an advantage over classical methods since it leads to highly accurate results. That is because, as we described, it considers the system in a full quantum way, both in the definition of the wave function and in the implementation of the evolution through direct interactions between all the components.

For weakly-correlated large-size systems, for example drug molecules that interact with proteins, researched by the pharmaceutical industry, the methods aim to simulate an environment of hundreds of thousands of atoms. This makes the full quantum mechanical treatments of such systems out of reach. As a consequence, the most widely used computational techniques in pharma rely on force fields. However, while classical force fields capture most prominently bond lengths, bond angles, dihedrals as well as non-bonded electrostatic and van-der-Waals interactions, their traditional formulation does not account for finer electronic effects such as polarization, charge-transfer phenomena, aromatic stacking interactions or interactions with metal ions.

Thus, when more accurate treatment is required the approach that is traditionally used to describe the system is a hybrid method: a quantum mechanical (QM) technique is used to describe selected residues of the binding pocket and the drug, while the remainder of the system is simulated using molecular mechanics (MM). This is called QM/MM method.

QPE can bring to an advantage in this context, that is because of its asymptotic scaling. We can see that from eq. (1.29) and eq. (4.16) and (4.17) or otherwise from Figure 2.4. QPE can produce all the configurations in the Hilbert space generated by the qubits register and elaborates them in linear scaling width and depth.

Again simulation algorithms on quantum computers have a higher expressivity due to the intrinsic nature of the technology used to perform the calculation, but to prove a real quantum advantage the method must not introduce a significant overhead compared to classical methods, so that the improvements in the accuracy of results will come at a reasonable cost. This is why the main research on QPE revolves around finding an alternative core component to Trotterization that can lead to lower computational requirements.

6.3 Targets of simulation

6.3.1 Defining advantage

Quantum advantage in computational chemistry can be sought in a space of three dimensions: speed, accuracy, and molecule size. These define advantages that are not useful in practical or industrial applications [14]:

1. Irrelevance due to availability of accurate experimental results

These are the cases where quantum computers would have to compete with

experimental measurements that are accurate, fast, inexpensive and straightforward. It is obvious that in presence of readily available and reliable experimental data there is little need for simulated results;

2. Irrelevance due to availability of conventional computational results

These pertain to chemical systems or problems for which quantum chemical calculations on classical computers can produce chemical accuracy results in little time (seconds, minutes or even a few hours). These are exactly the types of applications on which quantum computing algorithms have been routinely validated so far;

3. Irrelevance due to real world complexity

Often the biggest problem in simulation research is not simply to complete single computations in reasonable time with sufficient accuracy. When simulated chemical processes are very complicated and involve potentially hundreds of intermediates, conformations or reaction paths, as in catalytic and metabolic pathways, the real research bottleneck lies in a combinatorial explosion of possibilities to probe with simulation. In such projects, even if the calculations themselves became orders of magnitude faster or more accurate, for example, through the exploitation of quantum computing, the whole project might enjoy only a modest speedup;

4. Irrelevance to industrial applications

These are molecular systems that, even if described accurately on a quantum computer, are likely to remain academic and theoretical interest and fail to lead to transformative changes in chemistry.

6.3.2 Applicability forecast

Now we focus on how the molecular size affects the capability of quantum computers to simulate molecular systems.

Despite the general impracticality of the largest calculations mentioned above, just the sheer fact that they have been accomplished on classical computers sets the bar very high for quantum computers.

The size of ABP, mentioned in the last section, makes it an attractive minimum viable system in real world practical applications. Systems of this size are regularly used for parameterizing force fields in areas of research like computational drug design.

We saw that accuracy is strictly linked to the basis set and especially on the number of functions chosen to model the system.

To give an example of the relation between basis sets and accuracy we take two simple molecules, the H₂ and He molecules. For these one does not achieve chemical accuracy at the FCI level with the cc-pVTZ basis set (28 basis functions or 56 spin-orbitals). One has to use either cc-pVQZ or both cc-pVDZ and cc-pVTZ in an extrapolation scheme for H₂, whereas for He at least the cc-pVTZ (14 functions or 28 spin-orbitals) and cc-pVQZ (30 functions or 60 spin-orbitals) basis set energies

are required as inputs to an extrapolation scheme.

However, in recent experiments with quantum computing hardware, applied to simulating molecular and materials chemistry, the number of qubits that have been used is significantly smaller than the one necessary to achieve chemical accuracy, for all molecules studied. Experiments are shown in Figure 6.5.

Reference	Year	Max # qubits	Systems	Platform	Methods
Peruzzo et al.	2013	2	HeH^+	Silicon Photonic	VQE-UCC
Shen et al.	2015	2	HeH^+	Trapped ion	VQE-UCC
Google	2015	2	H_2	Superconducting	VQE-UCC
Santagati et al.	2016	2	$\text{H}_2, \text{H}_3, \text{H}_3^+, \text{H}_4$	Silicon photonic	IPEA, VQE-UCC
IBM	2017	6	$\text{H}_2, \text{LiH}, \text{BeH}_2,$ Heisenberg model	Superconducting	Hardware-efficient VQE
Berkeley	2017	2	H_2 (excited states)	Superconducting	Hardware-specific VQE
Hempel et al.	2018	3	H_2, LiH	Trapped-ion	VQE-UCC
IBM	2018	4	Quantum magnetism H_2, LiH	Superconducting	Hardware-efficient VQE
OTI Lumionics	2018	4	H_2, LiH	Superconducting	Qubit CC
Li et al.	2019	2	H_2O	NMR	QPE
IonQ/JQI	2019	4	H_2O	Trapped-ion	VQE-UCC
Oak Ridge	2019	4	$\text{NaH}, \text{RbH}, \text{KH}$	Superconducting	Hardware-efficient VQE(-UCC)
Mitsubishi/IBM	2019	2	Lithium superoxide dimer	Superconducting	VQE-UCC
Smart & Mazzotti	2019	3	H_3	Superconducting	VQE-UCC
Google	2020	12	$\text{H}_6, \text{H}_8, \text{H}_{10}, \text{H}_{12}$ HNNH	Superconducting	VQE-HF
IBM	2020	2	PSPCz, 2F-PSPCz, 4F-PSPCz	Superconducting	qEOM-VQE VQD

Figure 6.5: Simulations on real quantum computers [14].

Moreover, in 2021 Huggins et al. [20] simulated a simplified version of the energy state of a molecule consisting of 12 hydrogen atoms with 12 qubits, each one representing one atom's single electron. They then modeled a chemical reaction in a molecule containing hydrogen and nitrogen atoms, including how that molecule's electronic structure would change when its hydrogen atoms shifted from one side to the other. That represents "the largest chemistry simulations performed with the help of quantum computers".

Elfving et al. [14] proposed the chromium dimer, Cr_2 , as a good target for benchmarking simulation on quantum computers. This molecule, with its very short formally sextuple bond and a peculiarly shaped dissociation curve, could be a critical test for electronic structure methods.

In their calculation on this target they chose the complete active space (CAS) approach, where they selected an increasing number of active orbitals and active electrons. The notation used to define the number of electrons and the number of orbitals used in the model is (N, N) . They carried on a resource estimation on ground state energy estimation, focusing especially on the core component of QPE algorithm and comparing Trotterization and qubitization methods with surface code

for error-correction.

This comparison is brought with reference to the number of T and Toffoli gates, i.e. the depth, and the number of logical qubits, i.e. the width. They showed that quibitization, especially in its sparse form, gives rise to a relatively large overhead in the number of logical qubits but it is optimized in terms of the non-Clifford gate complexity, leading to the conclusion that the trade-off between Trotterization and quibitization is the circuit depth versus the number of logical qubits.

After that, they compared the result on the quantum computer, in terms of wallclock time scaling, to a desktop PC simulation (corresponding to a Intel i9-10980XE, with ~ 1.2 TFLOPS) and a 10^5 x faster extrapolation (corresponding to a top-5 HPC, at ~ 125 PFLOPS). The results are shown in Figure 6.6 and 6.7. They considered error rates $p = 10^{-3}$ (already achieved in experiment) and $p = 10^{-6}$ (long-term hardware ambition).

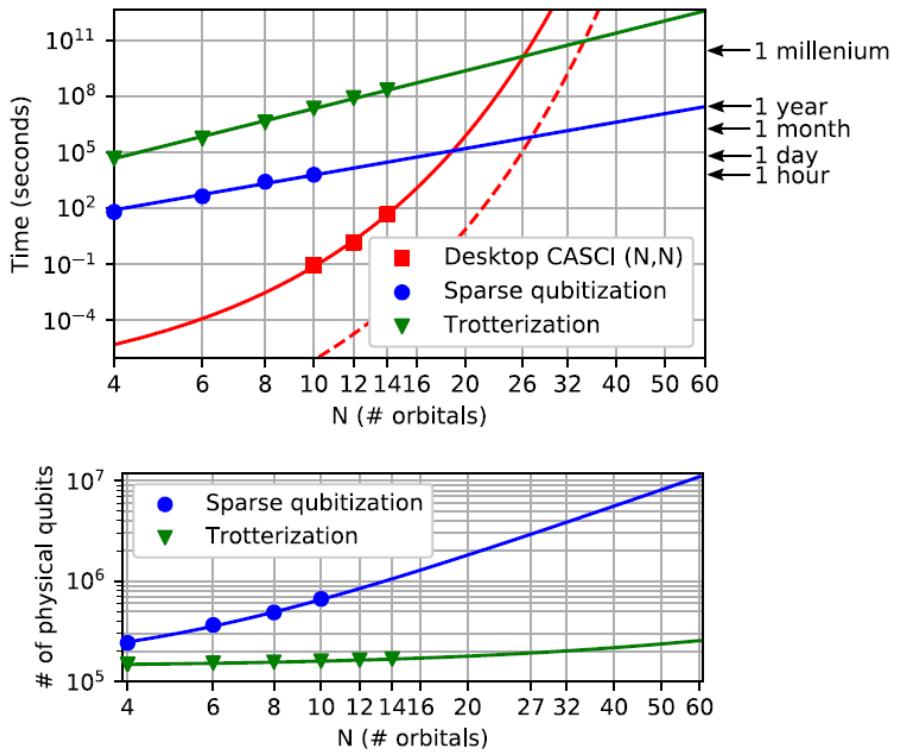


Figure 6.6: Runtime and physical qubit requirements for the chromium dimer (Cr_2) [14].

The figures show that the approximate size where a quantum computer solves the problem faster than a classical computer is for a (N, N) CAS size of around $N = 19 - 34$. For $N > 34$ any of the assessed quantum algorithms should be faster than any available classical computer. $N = 19$ implies a physical qubit count of $\sim 10^5$ for Trotterization and $\sim 3 \cdot 10^6$ for sparse qubitization. However, in the case of Trotterization the crossover point happens at a total runtime approaching a thousand years, which even with some optimization operations seems infeasible. The crossover for qubitization appears to happen at approximately the same point as the maximum

	Optimized for	Number of physical qubits	Total runtime
Trotterization $p = 10^{-3}$	space	3.8×10^5	1485 years
	time	1.6×10^6	161 years
Trotterization $p = 10^{-6}$	space	2.0×10^4	343 years
	time	8.6×10^4	37 years
Qubitization $p = 10^{-3}$	space	4.6×10^6	43 days
	time	7.1×10^6	110 hours
Qubitization $p = 10^{-6}$	space	2.7×10^5	11 days
	time	4.2×10^5	27 hours

Figure 6.7: Resource estimates for simulating the chromium dimer with a CAS space of (26,26) [14].

size which is still feasible at all on a classical supercomputer.

Although the resource estimates give an indication of the current state of the art in algorithms, these are not necessarily lower bounds and simulation algorithms may improve by orders of magnitude both in terms of complexity and prefactor.

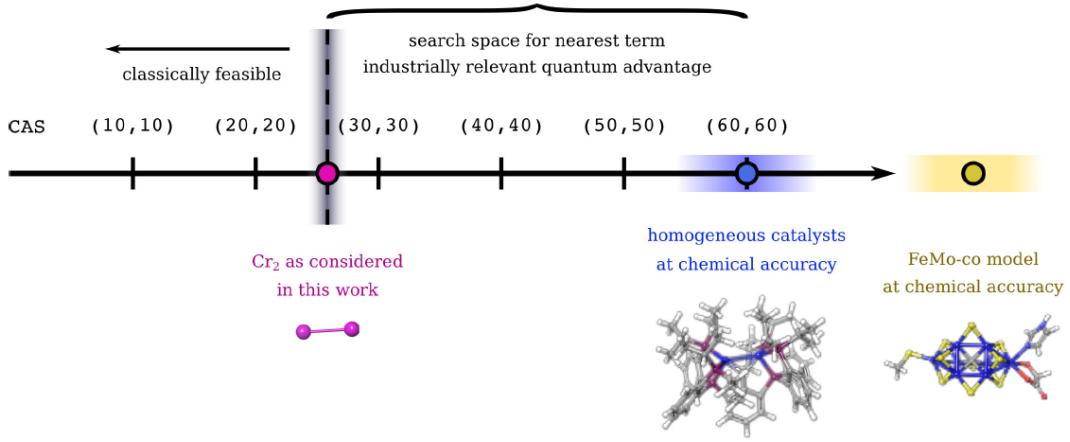
This makes it possible to delineate a range of possible targets of application for algorithms on quantum computers. In Figure 6.8 we show a scheme of the molecule systems with respect to the number of spatial orbitals used to carry on a simulation in the CAS approach.

Light atom diatomics are not put on the CAS axis because of the unclear boundaries between the dynamic and non-dynamic correlation in these molecules. The use of these methods for small molecules containing only light atoms, which have been an object of study and research direction in most of the quantum computing literature, does not appear to be promising.

On the other side of the spectrum are large and complicated molecules, the properties and chemistry of which stand little chance of being accurately described by classical computers. An example of such a molecular system is the FeMoco protein system. Although very relevant for possible industrial applications, FeMoco presents an overwhelming complexity for a short-term quantum computer.

For a relevant successful application of quantum computations a "sweet spot" lies somewhere in the middle between tiny light atom molecules and large complex multi-metal active sites of protein complexes.

The chromium dimer molecule, lacking industrial applications, can not be regarded as a good ultimate target for quantum computers, but it can be used as a model or rather as a testing ground on a way to the middle ground system to seek. One type of chemistry for such a middle ground can be found among medium-sized inorganic catalyst molecules which are a subject of homogeneous catalysis research. A report [10] investigated the applicability of quantum algorithms to the quantum



	Light atom diatomics	Cr ₂	Homogeneous catalysts	FeMo-co model
Chemical complexity	Minimal	Medium	High	Very high
Dynamic correlation	Medium	Medium	Difficult	Difficult
Non-dynamic correlation	Easy	Medium	Difficult	Very difficult
Industrial relevance	Irrelevant	Irrelevant	Highly relevant	Highly relevant
Relativistic effects	Non-existent	Not likely	Not likely	Likely
Protein matrix	No	No	No	Yes
Molecular geometry	Accurate	Accurate	Likely accurate	Unclear
Description on classical computer	Accurate	Sufficiently accurate	Not sufficiently accurate	Not sufficiently accurate

Figure 6.8: Possible short term quantum computer applications and schematic placement of their CAs [14].

chemical description of a ruthenium-containing catalyst designed to convert CO₂ into methanol. The authors proposed the treatment of active spaces that span 48-76 electrons and 52-65 orbitals. An attractive subclass of homogeneous catalysts on which early quantum advantage studies can be focused are biomimetics. These normally di- and tri-metal complexes borrow chemical insight from natural metal-containing enzymes and are designed for tackling industrially important chemical transformations such as C-H bond activation or the N₂ bond cleavage. Highly efficient C-H bond activation, for example, is at the heart of the idea of "methanol economy", which seeks to replace petroleum and coal by cleaner sources of energy and synthetic materials.

The estimates for the chromium dimer places quantum advantage for CAs of the structure (N, N) in the region after N ≈ 26. There is a gap between this value after which the quantum advantage can be expected and N ≈ 60, which is required for the accurate treatment of the non-dynamic correlation of homogeneous catalysts. The estimates for the latter type of molecules indicate that their treatment will require at least thousands of logical qubits.

Little is left for the VQE method. If we assume that one could reach CASCI levels of precision in a given basis set, by using a VQE method on a NISQ device with a linear circuit depth, one can estimate with similar assumptions as above, that tackling a (26, 26) CA problem would take about 1 hour per VQE iteration. Although this runtime does not seem completely prohibitive, the range of assumptions made,

prevents from drawing strong conclusions for practical and industrially relevant application.

Gonthier et al. [17] made an estimate from the extrapolation of results from their organic molecules benchmark set, leading to a total runtime of a few hours, which could allow for a full VQE optimization with considerable effort. The results are shown in Figure 6.9. However, Hamiltonian coefficients for heavier strongly-correlated atoms like Cr might be larger, which would result in larger values of K . Moreover, even if such a computation becomes possible, the transition to practically relevant advantage could require active spaces beyond 100 qubits.

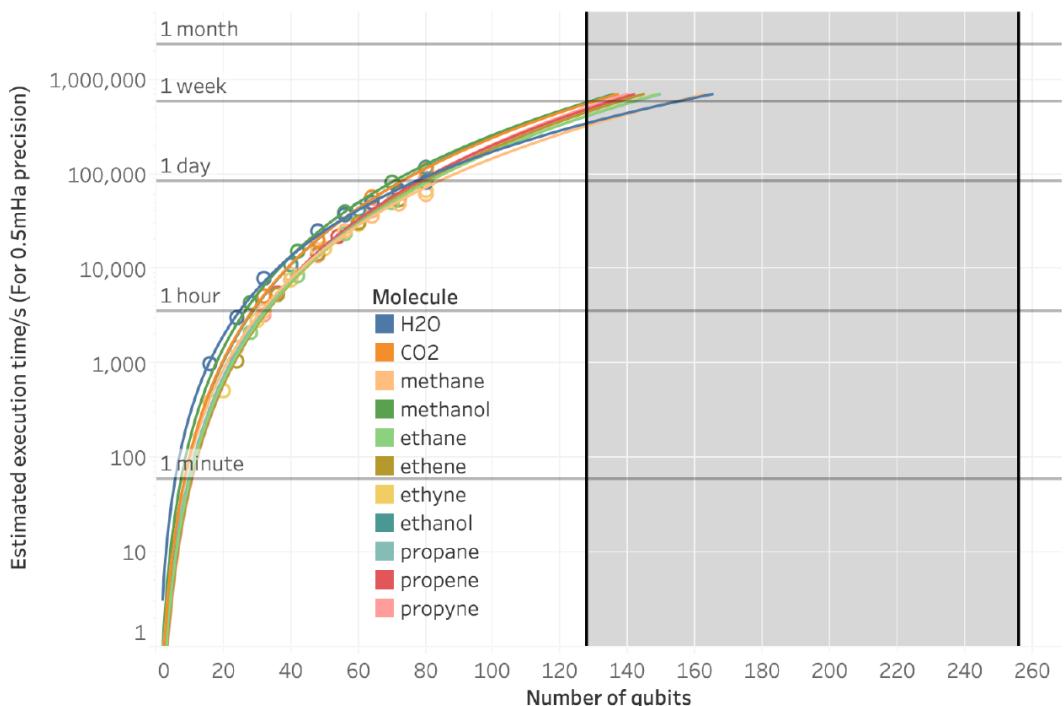


Figure 6.9: Extrapolated runtimes (in s) for the organic molecules benchmark set [17].

A more in-depth analysis of this problem has been made by Tilly et al. [46] which provided the estimated runtimes for the steps in the VQE, included the general scaling for the chromium dimer, with an active space of 26 electrons in 26 molecular orbitals (52 spin-orbitals and 52 qubits). They compiled a 52 qubit version of k-UpCCGD assuming $k = 1$ and full connectivity of the qubit register. The results are shown in Figure 6.10, where $L \sim 27\,000$ is the number of timesteps, i.e. the depth, $p = 170$ is the number of parameters, $T = 100$ ns is the assumed gate time, $P \sim 16N = 832$ is the number of operators and $S = 1/\epsilon^2 = 1\,000\,000$ is the number of shots used for estimation.

This demonstrates what we have already analyzed before in this chapter: the VQE will likely not show any advantages for practical and industrially relevant systems.

Operation	Scaling	Formula	Runtime
Single shot (using k-UpCCG SD, $k = 1$)	$\mathcal{O}(kN)$	$L \times T$	3 ms
One expectation at $\epsilon = 10^{-3}$ (using decomposed interaction method)	$\mathcal{O}(\frac{kN^2}{\epsilon^2})$	$L \times T \times P \times S$	25 days
Full iteration using Rotosolve	$\mathcal{O}(\frac{k^2 N^4}{\epsilon^2})$	$L \times T \times P \times S \times 3p$	35 years
Full iteration using gradient based method	$\mathcal{O}(\frac{k^2 N^4}{\epsilon^2})$	$L \times T \times P \times S \times 2p$	24 years

Figure 6.10: Estimates of one iteration of VQE for the simulation of the chromium dimer (Cr_2) molecule [46].

Elfving et al. [14] stressed that the real challenge is going from 10 to a million physical qubits, i.e. building a scalable platform, with logical qubits that can carry information and can be manipulated for virtually an infinite amount of time.

From a technological perspective this means that the next most important challenge to use these methods is to improve qubit error rates and error-correction codes. This is the approach that companies like Google (Figure 6.3) are pursuing and it is an unavoidable step for quantum computers to mature in their development.

This leads to the conclusion that the number of qubits alone is not a sufficient technological parameter to determine the quality of a real simulation. A precise forecast of applicability and improvability must take into account progresses in other parameters like maximum depth of a quantum circuit, T_1 and T_2 , as well as quality parameters like quantum volume, precedently described.

A forecast based on available roadmaps is that in the next 1 to 3 years quantum processors with hundreds to a thousand qubits and high-fidelity connectivity will be built. This will be the crucial period for the development of error-correction codes and first logical qubits, as well as first software applications on NISQ devices. Then, if quantum communication and scaling techniques for these structures will be ready, comprising a lower error rate, between 4 to 6 years from now quantum processors with thousands to hundreds of thousands of physical qubits and roughly tens to a hundred logical qubits will be available and it will be possible to test the first defined targets for computational advantage. After that, in 7 to 9 years it is expected to reach structures of up to a million physical qubits and hundreds to a thousand logical qubits. This will be the minimum necessary computational capability to provide computational advantages in practical and industrially relevant systems, as analyzed earlier in this chapter.

A more conservative forecast would be that the key milestone of achieving fault-tolerant large-scale quantum computers might occur in longer time, thus extending the period of their deployment to 10-15 years from now.

The technology development path at this limited level of maturity and with this level of complexity is unlikely to be linear. It is likely that a breakthrough will happen in the medium term and then the technological growth will be exponential. As Niels Bohr said: "Prediction is very difficult, especially if it's about the future".

6.4 Next steps

In the last section of this thesis we show some of the possible next steps in the research for quantum computational chemistry. We focus especially on methods and targets of simulation.

6.4.1 Problem fragmentation

To tackle complex chemistry and material science problems using NISQ computers one can search for methods to reduce the number of electrons treated explicitly at the highest level of accuracy. Building on the idea underpinning dynamical mean field theory (DMFT), one may simplify such problems by defining active regions or building blocks with strongly-correlated electronic states, embedded in an environment that may be described within mean-field theory.

These methods goes under the name of 'problem fragmentation' or 'problem decomposition', another example is density matrix embedding theory (DMET).

Ma et al. [29] proposed a quantum embedding theory built on one side on DFT, which is scalable to large systems and which includes the effect of exchange-correlation interactions of the environment on active regions, and on the other side on effective Hamiltonian for the active regions themselves using QPE or VQE. A general representation of this method is shown in Figure 6.11.

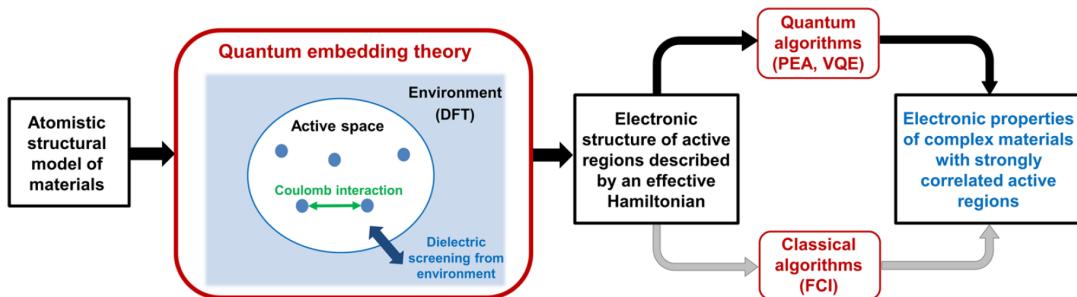


Figure 6.11: Quantum simulations of materials using quantum embedding theory [29].

The theory, inspired by the constrained random phase approximation (cRPA) approach, does not require the explicit evaluation of virtual electronic states, thus making the method scalable to materials containing thousands of electrons, and goes beyond RPA by explicitly including exchange-correlation effects.

The key part is to define the effective Hamiltonian

$$H^{\text{eff}} = \sum_{i,\alpha}^A t_{i\alpha}^{\text{eff}} a_i^\dagger a_\alpha + \frac{1}{2} \sum_{i>j,\alpha>\beta}^A V_{ij\alpha\beta}^{\text{eff}} a_i^\dagger a_j^\dagger a_\alpha a_\beta \quad (6.1)$$

and consequently the effective kinetic (t^{eff}) and potential (V^{eff}) terms. These terms are one- and two-body interaction terms that take into account effect of all the electrons that are part of the environment in a mean-field fashion, at the DFT level. An active space can be defined, for example, by solving the Kohn–Sham equations of the full system and selecting a subset of eigenstates among which electronic excitations of interest take place, e.g. defect states within the gap of a semiconductor or insulator.

They computed ground and excited state properties of several spin-defects in solids including the negatively charged nitrogen-vacancy (NV) center, the neutral silicon-vacancy (SiV) center in diamond, and the Cr impurity (4+) in 4H-SiC. Especially the investigation of the strongly-correlated electronic states of the NV center in diamond showed that quantum simulations yield results in agreement with those obtained with classical FCI calculations.

This method is similar to the one used by Reiher et al. [41] in the experiment described in Chapter 4, where the structure of the MoFe protein and the FeMoco active site are treated separately.

It could be essential to pursue this research for modelization and simulation of large-size systems like proteins.

6.4.2 Materials science

Throughout the thesis we focused especially on chemical systems as targets of quantum simulation methods, but systems and models which pertain to materials science could also be prominent targets of applicability.

These are mostly studied using lattices structures and plane wave basis sets. Their models all give rise to Hamiltonians with local interactions, thus efficiently simulatables on quantum devices. Moreover, large crystalline materials all exhibit quantum properties intrinsically related to the high level of correlation amongst their composing electrons [6].

Computing electronic band structures and phase diagrams to better understand these strongly correlated materials are grand challenges of condensed matter physics and materials science. This challenge is motivated both by practical considerations, such as the design and characterisation of novel materials, and by fundamental science.

Stanisic et al. [44] showed experimentally how to implement an efficient quantum algorithm for simulation of the Fermi-Hubbard model. This is the simplest system that includes non-trivial correlations not captured by classical methods. Although it is a highly simplified model of interacting electrons in a lattice, to date the largest Fermi-Hubbard system which has been solved exactly consisted of just 17 electrons on 22 sites.

The Hamiltonian is defined as

$$H = - \sum_{\langle i,j \rangle, \sigma} (a_{i\sigma}^\dagger a_{j\sigma} + a_{j\sigma}^\dagger a_{i\sigma}) + U \sum_i n_{i\uparrow} n_{i\downarrow}. \quad (6.2)$$

In their experiment they used a low-depth VQE implementation with few parameters to simulate 1×8 and 2×4 instances on the Rainbow superconducting quantum processor in Google Quantum AI's Sycamore architecture with 23 qubits available. In Figure 6.12 the qubit layout used is shown.

In the quantum circuit to be optimized, the number of ansatz layers scales like the number of sites, corresponding to several hundred layers of two-qubit gates. While substantially smaller than previous quantum circuit complexity estimates for post-classical simulation tasks, this is still beyond the capability of today's quantum computers.

They achieved accurate representations of the ground state of Fermi-Hubbard instances beyond classical exact diagonalization and observed physical properties expected for the ground state, such as the metal-insulator transition, Friedel oscillations, decay of correlations and antiferromagnetic order.

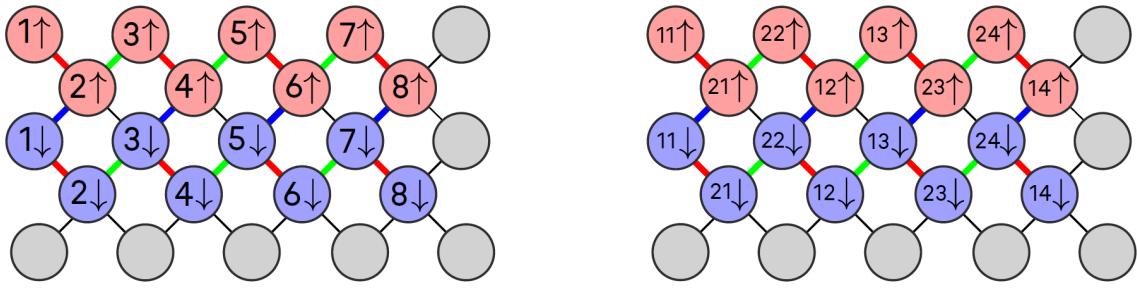


Figure 6.12: Qubit layout for implementing 1×8 (left) and 2×4 (right) Fermi-Hubbard instances [44].

The Fermi-Hubbard model has been widely proposed as an early target for quantum simulation algorithms. In the first place because of its direct application to understanding technologically-relevant correlated materials. Secondly, because the regularity and the relative simplicity of the Fermi-Hubbard Hamiltonian suggest that it may be easier to solve using a quantum computer than, for example, a large unstructured molecule.

Experimental results suggest that materials in the condensed phase and Fermi-Hubbard simulations require considerably fewer resources, nevertheless thousands or hundreds of thousands of physical qubits and order of 10^7 Toffoli or T gates are needed [31]. On the other hand, the challenge that it presents for classical methods makes it an excellent benchmark for quantum algorithms.

Conclusion

Quantum computational chemistry is likely to become one of the most researched applications of quantum computing. This could help to solve classically intractable chemistry and materials science problems and study phenomena like superconductivity, solid-state physics, transition metal catalysis and certain biochemical reactions. In turn, this increased understanding may help to refine, and perhaps even one day design, new compounds of scientific and industrial importance.

This advantage, however, will require quantum computers with high computational power, i.e. with logical qubits, high-fidelity connectivity and low gate time. Capability that will be achieved in a varying period of time, strictly depending on major breakthroughs in engineering of quantum processors.

Throughout the thesis we have given a brief introduction to quantum computing and we have discussed the main algorithms developed for classical and quantum devices to solve the electronic structure problem. We have analyzed those from the mathematical and the computational point of view and studied their technological requirements.

Finally, we have focused on the conditions to define a computational advantage and on the targets that could be researched to achieve it and we have provided a forecast of applicability for industrially relevant systems.

Future works on this topic should focus on performing more experiments to provide more and more precise estimates on computational requirements. Some frameworks for running simulations are already available and easily accessible, among the most popular are Qiskit-nature, PennyLane and tequila [24]. These could provide a more widely spread research on this topic, thus it is important to also develop them to collect more experimental results on ever-growing devices.

Although a fundamental portion of research must be dedicated to finding better techniques to make high-fidelity and scalable quantum processors, it is important that even mathematical and software algorithms are improved and optimized, i.e. better state preparation and orbitals representation, better quantum measurement process and hamiltonian simulation with a better scaling.

Acknowledgments

I am extremely grateful to Alberto Acuto, Antonio Policicchio and Nicolò Toscano as well as the whole group of Quantum development at NTT Data Italia for constantly following and advising me throughout the entire making of this thesis, from documentation to challenging ideas.

I must also thank my supervisor prof. Alberto Carlini for kindly guiding me through the choice of the topic.

Finally, I would like to thank my family and my dearest friends for always supporting me at whatever cost.

Bibliography

- [1] [Online; accessed 30. Aug. 2022]. July 2016. URL: <https://physlab.org/wp-content/uploads/2016/07/QComp-Photons-2015-J-Phys-B.pdf>.
- [2] Abhinav Anand et al. “A quantum computing view on unitary coupled cluster theory”. In: *Chem. Soc. Rev.* 51.5 (Mar. 2022), pp. 1659–1684. ISSN: 0306-0012. DOI: 10.1039/D1CS00932J.
- [3] Martin P. Andersson et al. “Quantum computing for chemical and biomolecular product design”. In: *Curr. Opin. Chem. Eng.* 36 (June 2022), p. 100754. ISSN: 2211-3398. DOI: 10.1016/j.coche.2021.100754.
- [4] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574 (Oct. 2019), pp. 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5.
- [5] Ala'n Aspuru-Guzik et al. “Simulated Quantum Computation of Molecular Energies”. In: *Science* 309.5741 (Sept. 2005), pp. 1704–1707. ISSN: 0036-8075. DOI: 10.1126/science.1113479.
- [6] Lindsay Bassman et al. “Simulating quantum materials with digital quantum computers”. In: *Quantum Sci. Technol.* 6.4 (Sept. 2021), p. 043002. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ac1ca6.
- [7] Bela Bauer et al. “Quantum Algorithms for Quantum Chemistry and Quantum Materials Science”. In: *Chem. Rev.* 120.22 (Nov. 2020), pp. 12685–12717. ISSN: 1520-6890. DOI: 10.1021/acs.chemrev.9b00829. eprint: 33090772.
- [8] Nick S. Blunt et al. “A perspective on the current state-of-the-art of quantum computing for drug discovery applications”. In: *arXiv* (June 2022). DOI: 10.48550/arXiv.2206.00551. eprint: 2206.00551.
- [9] Colin D. Bruzewicz et al. “Trapped-Ion Quantum Computing: Progress and Challenges”. In: *arXiv* (Apr. 2019). DOI: 10.1063/1.5088164. eprint: 1904.04178.
- [10] Vera von Burg et al. “Quantum computing enhanced computational catalysis”. In: *undefined* (2021). URL: <https://www.semanticscholar.org/paper/Quantum-computing-enhanced-computational-catalysis-Burg-Low/1df87ec3b118d35003aca194f5842129284f0a5d>.
- [11] Kieron Burke. “Perspective on density functional theory”. In: *J. Chem. Phys.* 136.15 (Apr. 2012), p. 150901. ISSN: 0021-9606. DOI: 10.1063/1.4704546.

- [12] Yudong Cao et al. “Quantum Chemistry in the Age of Quantum Computing”. In: *Chem. Rev.* 119.19 (Oct. 2019), pp. 10856–10915. ISSN: 0009-2665. DOI: 10.1021/acs.chemrev.8b00803.
- [13] Christopher J. Cramer. *Essentials of Computational Chemistry: Theories and Models, 2nd Edition*. Hoboken, NJ, USA: Wiley, Sept. 2004. ISBN: 978-0-470-09182-1. URL: <https://www.wiley.com/en-ie/Essentials+of+Computational+Chemistry%3A+Theories+and+Models%2C+2nd+Edition-p-9780470091821>.
- [14] V. E. Elfving et al. “How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?” In: *arXiv* (Sept. 2020). DOI: 10.48550/arXiv.2009.12472. eprint: 2009.12472.
- [15] Richard P. Feynman. “Simulating physics with computers”. In: *Int. J. Theor. Phys.* 21.6 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179.
- [16] Pranav Gokhale et al. “Minimizing State Preparations in Variational Quantum Eigensolver by Partitioning into Commuting Families”. In: *arXiv* (July 2019). DOI: 10.48550/arXiv.1907.13623. eprint: 1907.13623.
- [17] Jérôme F. Gonthier et al. “Identifying challenges towards practical quantum advantage through resource estimation: the measurement roadblock in the variational quantum eigensolver”. In: *arXiv* (Dec. 2020). DOI: 10.48550/arXiv.2012.04001. eprint: 2012.04001.
- [18] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: *STOC ’96: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*. New York, NY, USA: Association for Computing Machinery, July 1996, pp. 212–219. DOI: 10.1145/237814.237866.
- [19] Nicholas M. Harrison. Aug. 2006. URL: https://www.ch.imperial.ac.uk/harrison/Teaching/DFT_NATO.pdf.
- [20] William J. Huggins et al. “Unbiasing fermionic quantum Monte Carlo with a quantum computer”. In: *Nature* 603 (Mar. 2022), pp. 416–420. ISSN: 1476-4687. DOI: 10.1038/s41586-021-04351-z.
- [21] IBM Quantum. [Online; accessed 29. Sep. 2022]. Sept. 2022. URL: <https://quantum-computing.ibm.com/services/resources>.
- [22] Frank Jensen. *Introduction to Computational Chemistry*. Dec. 2011. ISBN: 978-0-47001187-4.
- [23] Isaac H. Kim et al. “Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules”. In: *ResearchGate* (Apr. 2021). URL: https://www.researchgate.net/publication/351046677_Fault-tolerant_resource_estimate_for_quantum_chemical_simulations_Case_study_on_Li-ion_battery_electrolyte_molecules.
- [24] Jakob S. Kottmann et al. “Tequila: A platform for rapid development of quantum algorithms”. In: *arXiv* (Nov. 2020). DOI: 10.1088/2058-9565/abe567. eprint: 2011.03057.

- [25] Joonho Lee et al. “Even more efficient quantum computations of chemistry through tensor hypercontraction”. In: *arXiv* (Nov. 2020). DOI: 10.1103/PRXQuantum.2.030305. eprint: 2011.03494.
- [26] Joonho Lee et al. “Generalized Unitary Coupled Cluster Wave functions for Quantum Computation”. In: *J. Chem. Theory Comput.* 15.1 (Jan. 2019), pp. 311–324. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.8b01004.
- [27] Seth Lloyd. “Universal Quantum Simulators”. In: *Science* 273.5278 (Aug. 1996), pp. 1073–1078. ISSN: 0036-8075. DOI: 10.1126/science.273.5278.1073.
- [28] Erik Lucero. “Unveiling our new Quantum AI campus”. In: *Google* (May 2021). URL: <https://blog.google/technology/ai/unveiling-our-new-quantum-ai-campus>.
- [29] He Ma, Marco Govoni, and Giulia Galli. “Quantum simulations of materials on near-term quantum computers”. In: *npj Comput. Mater.* 6.85 (July 2020), pp. 1–8. ISSN: 2057-3960. DOI: 10.1038/s41524-020-00353-z.
- [30] Nicholas Materise. “An Introduction to Superconducting Qubits and Circuit Quantum Electrodynamics”. In: *arXiv* (Aug. 2017). DOI: 10.1007/978-3-319-92726-8_10. eprint: 1708.07000.
- [31] Sam McArdle et al. “Quantum computational chemistry”. In: *Rev. Mod. Phys.* 92.1 (Mar. 2020), p. 015003. ISSN: 1539-0756. DOI: 10.1103/RevModPhys.92.015003.
- [32] Jarrod R. McClean et al. “The theory of variational hybrid quantum-classical algorithms”. In: *New J. Phys.* 18.2 (Feb. 2016), p. 023023. ISSN: 1367-2630. DOI: 10.1088/1367-2630/18/2/023023.
- [33] Mario Motta and Julia E. Rice. “Emerging quantum computing algorithms for quantum chemistry”. In: *WIREs Comput. Mol. Sci.* (Dec. 2021). ISSN: 1759-0876. DOI: 10.1002/wcms.1580.
- [34] M. Van den Nest. “Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond”. In: *arXiv* (Nov. 2008). DOI: 10.48550/arXiv.0811.0898. eprint: 0811.0898.
- [35] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge, England, UK: Cambridge University Press, Dec. 2010. ISBN: 978-0-51197666-7. DOI: 10.1017/CBO9780511976667.
- [36] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nat. Commun.* 5.4213 (July 2014), pp. 1–7. ISSN: 2041-1723. DOI: 10.1038/ncomms5213.
- [37] Nicolas Poirier. “Born-Oppenheimer molecular dynamics using the variational quantum eigensolver”. In: *eScholarship@McGill* (2020). URL: <https://escholarship.mcgill.ca/concern/theses/fn107335s>.
- [38] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (Aug. 2018), p. 79. DOI: 10.22331/q-2018-08-06-79. eprint: 1801.00862v3.

- [39] *Quantum computing | Arthur D. Little*. [Online; accessed 30. Sep. 2022]. Sept. 2022. URL: <https://www.adlittle.com/en/insights/viewpoints/quantum-computing>.
- [40] *Quantum computing with superconducting qubits — PennyLane documentation*. [Online; accessed 30. Aug. 2022]. Aug. 2022. URL: https://pennylane.ai/qml/demos/tutorial_sc_qubits.html.
- [41] Markus Reiher et al. “Elucidating reaction mechanisms on quantum computers”. In: *Proc. Natl. Acad. Sci. U.S.A.* 114.29 (July 2017), pp. 7555–7560. ISSN: 1091-6490. DOI: [10.1073/pnas.1619152114](https://doi.org/10.1073/pnas.1619152114). eprint: 28674011.
- [42] Yutaka Shikano et al. “Post-Hartree-Fock method in quantum chemistry for quantum computer”. In: *Eur. Phys. J. Spec. Top.* 230.4 (June 2021), pp. 1037–1051. ISSN: 1951-6401. DOI: [10.1140/epjs/s11734-021-00087-z](https://doi.org/10.1140/epjs/s11734-021-00087-z).
- [43] P. W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE, Nov. 1994, pp. 124–134. DOI: [10.1109/SFCS.1994.365700](https://doi.org/10.1109/SFCS.1994.365700).
- [44] Stasja Stanisic et al. “Observing ground-state properties of the Fermi-Hubbard model using a scalable algorithm on a quantum computer”. In: *arXiv* (Dec. 2021). DOI: [10.48550/arXiv.2112.02025](https://doi.org/10.48550/arXiv.2112.02025). eprint: 2112.02025.
- [45] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory (Dover Books on Chemistry)*. Mineola, NY, USA: Dover Publications, July 1996. ISBN: 978-0-48669186-2.
- [46] Jules Tilly et al. “The Variational Quantum Eigensolver: a review of methods and best practices”. In: *arXiv* (Nov. 2021). DOI: [10.48550/arXiv.2111.05176](https://doi.org/10.48550/arXiv.2111.05176). eprint: 2111.05176.
- [47] Libor Veis and Pittner. “Adiabatic state preparation study of methylene”. In: *J. Chem. Phys.* 140.21 (June 2014), p. 214111. ISSN: 0021-9606. DOI: [10.1063/1.4880755](https://doi.org/10.1063/1.4880755).
- [48] Andrew Wack et al. “Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers”. In: *arXiv* (Oct. 2021). DOI: [10.48550/arXiv.2110.14108](https://doi.org/10.48550/arXiv.2110.14108). eprint: 2110.14108.
- [49] Daochen Wang, Oscar Higgott, and Stephen Brierley. “Accelerated Variational Quantum Eigensolver”. In: *Phys. Rev. Lett.* 122.14 (Apr. 2019), p. 140504. ISSN: 1079-7114. DOI: [10.1103/PhysRevLett.122.140504](https://doi.org/10.1103/PhysRevLett.122.140504).
- [50] Nathan Wiebe et al. “Simulating quantum dynamics on a quantum computer”. In: *J. Phys. A: Math. Theor.* 44.44 (Oct. 2011), p. 445308. ISSN: 1751-8121. DOI: [10.1088/1751-8113/44/44/445308](https://doi.org/10.1088/1751-8113/44/44/445308).