

# COST FUNCTION DEPENDENT QUANTUM GATE SYNTHESIS FOR MOLECULAR SIMULATIONS

Master's Thesis

Master of Materials Science

FERDIANSYAH IQBAL RAFANDI

Supervisor:

Dr. Péter Rakyta



Eötvös Loránd University (ELTE)

Faculty of Science

Department of Physics of Complex Systems

Budapest, 2023

# Statement of Intellectual Property

Name : Ferdiansyah Iqbal Rafandi

Neptun : A877V0

ELTE Faculty of Science, specialization: MSc Materials Science

Title of diploma work: Cost Function Dependent Quantum Gate Synthesis for Molecular Simulations

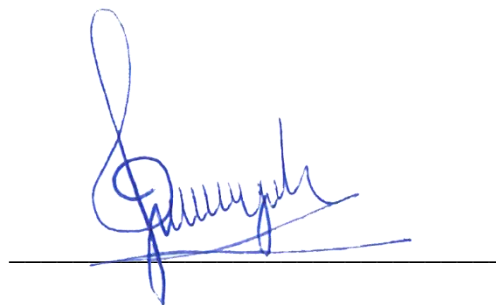
As the author of the diploma work, I declare, with disciplinary responsibility, that my thesis is my own intellectual product and the result of my own work. Furthermore, I declare that I have consistently applied the standard rules of references and citations.

I acknowledge that the following cases are considered plagiarism:

- using a literal quotation without quotation mark and adding citation;
- referencing content without citing the source;
- representing another person's published thoughts as my own thoughts.

Furthermore, I declare that the printed and electronic versions of the submitted diploma work are textually and contextually identical.

Budapest, 2023.05.27



*Signature of Student*

## Abstract

Quantum computers have the potential to revolutionize materials science by providing a new paradigm for simulating and designing materials, such as to simulate the behavior of molecules, providing insights into chemical reactions and could help to study the electronic structure of materials, which is critical to understanding their optical and electrical properties. However, high number quantum gate in circuit which represents material's system can cause abstract representation and tremendously difficult for quantum computer. Several works have been attempted to improve the simplicity of quantum circuits to be applied in current quantum devices.

In this work, adaptive circuit compression method is implemented for quantum gate synthesis. The decomposition process is optimized using ADAM and BFGS optimizers. 6 different cost functions are calculated with addition of correction order consisting of 2 cost function's norms: Frobenius and Hilbert-Schmidt. The decomposition processes are conducted in different iteration numbers and tolerance limits. The investigation reveals that Hilbert-Schmidt norm has outstanding stability when it is applied for decomposition of Q block quantum circuit, which has been used to obtain the ground state of P-Benzyne and Naphthalene. The execution time and the decomposition error of Hilbert-Schmidt norm is proportionally small and steady, whether it is observed in various maximum iterations values or tolerances. However, the existence of correction order doesn't give significant improvement in the effectivity and the efficiency of the cost function, yet the Hilbert-Schmidt norm of cost function is a good candidate to be applied since it contributes to the likelihood of observing small errors close to machine precision.

## Acknowledgements

First and foremost, I would like to thank God, the Almighty, who has granted me countless blessings, endless love, mercy, grace, strengths, and encouragements throughout all challenging moments, so that I have successfully been able to complete this thesis project.

Apart from my efforts, the success of this thesis was also largely achieved because of the encouragement and guidance from my thesis supervisor, Dr. Péter Rakyta. I would like to show my greatest appreciation to him, and I can't say thank you enough for his tremendous support and supervision. Without his instruction and insightful comments, this thesis would not have been possible.

Most of all, this thesis is dedicated to my beloved family in Indonesia for their constant prayers for me and my partner, JS, who has been extraordinarily supportive and always motivated me to finish this work in time and successfully. Special thanks also for my friends in Hungary, Misi, Ákos and Balázs, my Hungarian parents Ildikó and Ken, "The Other Direction" group: Jan and Andrés who also gave me support and encouragement when I was down and had no strength throughout the process. Last but not least, I want to express my thanks to my friends in Indonesia, Agnes, Alit, and Dayu for your constant call and the support I received even though you are far thousand miles away from Hungary.

Lot of love and thank to all of you.

# Table of Contents

Statement of Intellectual Property.....	I
Abstract.....	II
Acknowledgements .....	III
Table of Contents .....	IV
List of Figures .....	VI
<b>1  Introduction.....</b>	<b>1</b>
1.1 Brief Introduction to Quantum Computing .....	1
1.1.1 Qubit States.....	2
1.1.2 Single Qubit Gate.....	4
1.1.3 Multiple Qubit and Entangled States .....	7
1.2 Description of Variational Approach for Molecular System.....	8
1.3 Quantum Gate Synthesis and Optimization.....	15
1.3.1 Theoretical background behind the quantum gate synthesis .....	15
1.3.2 Adaptive Circuit Compression Method.....	18
<b>2  Methodology .....</b>	<b>21</b>
2.1 Quantum Circuits Generation .....	21
2.2 Brief description of optimization method.....	24
<b>3  Results and Discussion .....</b>	<b>26</b>
<b>4  Summary .....</b>	<b>34</b>

| **Appendix 1** ..... VIII

## List of Figures

<b>Figure 1.1</b> Qubit state representation on Bloch sphere.....	3
<b>Figure 1.2</b> Single qubit quantum gate illustration.....	6
<b>Figure 1.3</b> Quantum circuit of Rx, Ry and Rz gate.....	7
<b>Figure 1.4</b> Matrix form and quantum circuit of a) CNOT gate and b) CZ gate.....	8
<b>Figure 1.5</b> Schematic diagram of variational quantum Eigensolver.....	9
<b>Figure 1.6</b> Quantum circuit representation of Q building block.....	12
<b>Figure 1.7</b> Detail of four-qubits spatial orbital gate $QNPOR(\varphi)$ and four-qubits diagonal pair exchange gate $QNPPX(\theta)$ in which Given rotation G .....	14
<b>Figure 1.8</b> Controlled Ry gate which is generally expressed in terms of two CNOT and Ry gates. .....	19
<b>Figure 2.1</b> The qubits placement in the Sycamore processor's layout.....	22
<b>Figure 2.2</b> Quantum circuit representation of $Q = .QNPPX(\theta)QNPOR(\varphi)$ .....	23
<b>Figure 2.3</b> Decomposed quantum circuit representation of the $Q =$ $QNPOR(\pi)QNPPX(\theta)QNPOR(\varphi)$ gate on topology .....	24
<b>Figure 3.1</b> Iteration count over average execution time comparison for ADAM and BFGS optimizers with 6 different norms of cost functions in gate synthesis of Q gate structure. ....	26
<b>Figure 3.2</b> Decomposition Error result of Q gate in SQUANDER for both ADAM and BFGS optimizers.....	29
<b>Figure 3.3</b> Average decomposition time for 3 different Hilbert-Schmidt cost functions.....	32
<b>Figure 3.4</b> Decomposition error for 3 different Hilbert-Schmidt cost functions. ....	33

# **1 | Introduction**

Recent developments in computer technology have made remarkable advances in assisting scientists and engineers to simulate several phenomena in order to get exact data as similar as what actually happens in nature. A new breakthrough has been explored exhibiting a new possibility to deeply understand the physical and chemical properties of atoms, molecules, and materials using quantum computers, which is based on the quantum mechanic's principle. Quantum mechanics is the theory of physics that emerged to surpass the limitation of classical physics and this theory is the most accurate and complete theory in the world specially to describe physical and chemical phenomena. However, there is still some work that needs to be done to utilize this technology in computational physics and chemistry's field.

## **1.1 Brief Introduction to Quantum Computing**

Classically, information is encrypted by a series of bits, which represents by two different states and prominently labeled as 0 and 1 [1]. Quantum system has a similar idea to deliver information, however, the system is not necessarily in state 0 or 1. The system is rather in a superposition between 0 and 1 with mismatched probabilities. This means that the system can have the characteristics of both the states concurrently, at all times, until the measurement. To represent a bit in quantum world, two level systems are considered and usually called as quantum bit or qubit [2].

Qubits have characteristics of both states at the same time until they are measured. The superposition of qubits can facilitate broad computational space which can solve many complex problems because of its exponential space increment compared to bits [3]. This superiority can bring out quantum computing to handle large datasets with only a small number of qubits. Moreover, qubits can be also placed in entangled states. This means that qubits are intrinsically linked to each other, so that one qubit can dictate the possible measurement outcomes for another qubits, regardless how far apart



these objects are and this property is an important feature of qubit that can be exploited for quantum simulation of related systems [4].

### 1.1.1 Qubit States

A state of quantum system is described as vectors in a 2-dimensional Hilbert space, which are usually spanned by states  $|0\rangle$  and  $|1\rangle$ . These computational basis vectors correspond to the classical bits 0 and 1 respectively. The general state of this system can be expressed as an arbitrary state  $|\psi\rangle$  which is linearly weighted in orthonormal basis of  $\{|0\rangle, |1\rangle\}$  [5].

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \alpha_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}, \quad (1)$$

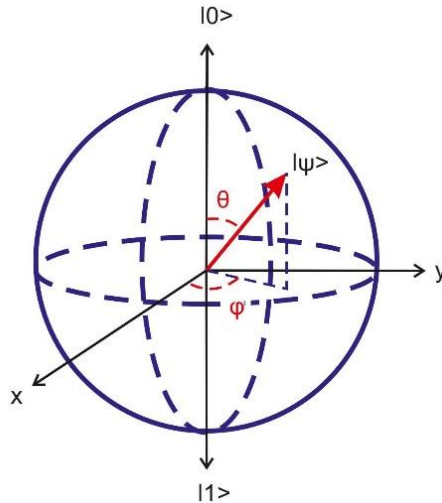
where the weighting factors  $\alpha_0, \alpha_1 \in \mathbb{C}$  are often called probability amplitudes and they must meet the normalization criteria:  $|\alpha_0|^2 + |\alpha_1|^2 = 1$ . This linearly weighted position of computational basis vectors is commonly termed as superposition. Therefore, when the qubit is measured, it can represent either “0” or “1” similar to the digital bits [6].  $|\alpha_0|^2$  shows the probability of the outcome  $|0\rangle$  which measures “0” while  $|\alpha_1|^2$  describes the probability for the outcome  $|1\rangle$  which measures “1”.

In spherical coordinates system, an arbitrary position of qubit state can be transformed in the term of two angles, denoted by  $\theta$  and  $\varphi$  respectively. Thus, the general state can be rewritten as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi} \sin\left(\frac{\theta}{2}\right)|1\rangle. \quad (2)$$

The state vector is generally depicted as a vector pointing to the surface of a 3-dimensional sphere, called Bloch sphere, as shown in Figure 1.1. Since state vectors are required to have a norm of 1 and to be equivalent up to the global phase, two real parameters  $\theta$  and  $\varphi$  are adequate to characterize a state vector [6]. The point on the surface of Bloch sphere can be also written in Cartesian coordinates as:

$$(x, y, z) = (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta). \quad (3)$$



**Figure 1.1** Qubit state representation on Bloch sphere.  $|0\rangle$  and  $|1\rangle$  represent the  $z$ -axis of Bloch sphere, which have the same analogy as North and South pole,  $|0\rangle$  is North and  $|1\rangle$  is South, and thus, any vector that doesn't represent directly  $|0\rangle$  or  $|1\rangle$  would be considered as superposition of both. While  $\theta$  and  $\phi$  are the angles of qubit state and  $|\psi\rangle$  is the single qubit vector.

Equally to classical devices, a collection of  $n$  qubits is called quantum register of size  $n$ . The amount of computational basis vectors to form arbitrary superposition of these vectors can be defined as  $N = 2^n$ . If the content of qubits in the quantum register is known, then the quantum state of quantum register can be calculated for independent qubits by means of a tensor product.

Suppose there are two independent qubits,  $|\psi_1\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$  and  $|\psi_2\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ , and these two qubits are combined, which results to the state of four-dimensional quantum register as follows:

$$\begin{aligned}
 |\psi\rangle &= |\psi_1\rangle|\psi_2\rangle = |\psi_1, \psi_2\rangle = |\psi_1\psi_2\rangle \\
 |\psi\rangle &= \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle}{2} \\
 &= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2},
 \end{aligned} \tag{4}$$

and from this equation, it can be observed that two qubits register consists of four weighted computational basis vectors which denotes as  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$  and  $|11\rangle$ . For a two-qubit system, a subset of qubits can be just measured, let's assume the first qubit is measured and  $|\psi_1\rangle = |0\rangle$  will be the outcome, the quantum register state will transform because of the measurement into:

$$|\psi\rangle = |0\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle}{\sqrt{2}}, \quad (5)$$

and proves that quantum register collapses to the subset of the whole Hilbert space and a general state for two qubits to an n-qubit register can written as [5]:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad (6)$$

where  $\alpha_i$  stands for probability amplitude and  $|i\rangle$  is its computational basis state.

### 1.1.2 Single Qubit Gate

Quantum gate can be defined as the operator in Hilbert space performed on the quantum registers and alters its state by rotation on the individual qubits [7]. This operator is unitary and the action of quantum gate,  $U$  can be described as a state transformation by this following equation:

$$|\psi'\rangle = U|\psi\rangle, \quad (7)$$

where  $|\psi\rangle$  is the starting state and  $|\psi'\rangle$  is the final state of qubit. The unitary operations of single qubit can be represented by 2 x 2 unitary matrices [8]. The most widely used single qubit gates are Identity gate, Pauli X gate, Pauli Y gate, Pauli Z gate, and Hadamard gate.

Identity gate is no-operation gate on one qubit. This gate doesn't create any modification on the basis state. In matrix form it is described as identity matrix.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (8)$$

Pauli X gate acts a  $\frac{\pi}{2}$  rotation along the x-axis. This gate is similar to a classical NOT operator, which flip the qubit from  $|0\rangle$  to  $|1\rangle$  or vice versa. The matrix form of Pauli X gate is given by:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (9)$$

Pauli Y gate creates rotation along y-axis on the qubit by  $\pi$  radians. This gate can be considered as the combination of Pauli X and Z gate [9]. This following equation shows the matrix form of Pauli Y gate.

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \quad (10)$$

Pauli Z gate rotates the qubit along z-axis by  $\pi$  radians. This gate is known as phase-flip gate, and it flips the phase of  $|1\rangle$  relative to  $|0\rangle$  state. The matrix of Pauli Z gate can be described as:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (11)$$

Hadamard gate gives rotation by  $\pi$  radians along diagonal axis to the xy plane, which makes this gate become the most interesting and useful among other single qubit gates [9]. Hadamard gate is known as phase-rotator gate or phase gate, and it performs with this following matrix operation.

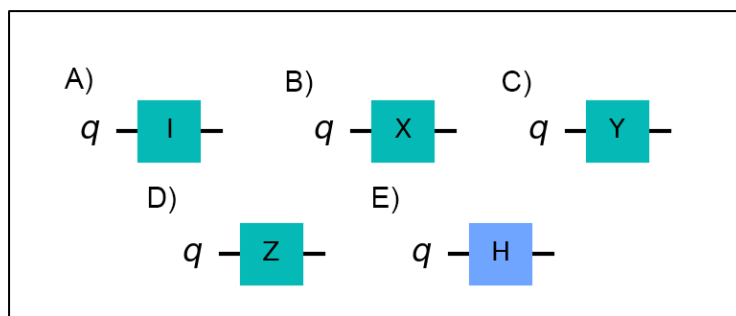
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (12)$$

However, Hadamard gate is not only unitary matrix, but it is also Hermitian matrix ( $H^\dagger = H$ ). This gate can create different probability amplitudes but the same probability vector [5]. The diagram of states of Identity gate, Pauli X gate, Pauli Y gate, Pauli Z gate, and Hadamard gate can be viewed in Figure 1.2.

Arbitrary rotation can also be performed by single qubit gate along x, y, or z-axis. This gate is usually called R gate. Rotation along x-axis by  $\theta$  radians is carried by RX-gate, while along y-axis by  $\theta$  radians by RY-gate, and  $\phi$  radians rotation along z-rotation by RZ-gate, which their diagrams of

state are depicted in Figure 1.3. Matrix representations of RX, RY and RZ gates can be written in the form as:

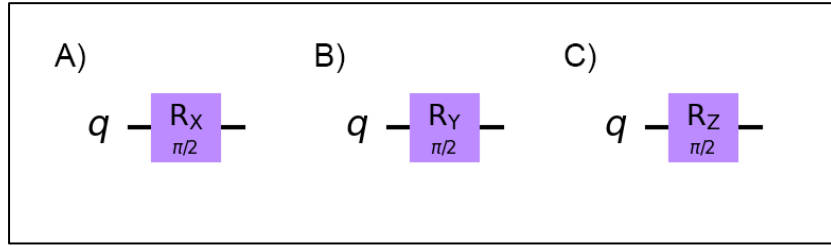
$$\begin{aligned}
 R_x(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i \sin\left(\frac{\theta}{2}\right) \\ -i \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \\
 R_y(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \\
 R_z(\phi) &= \begin{bmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{bmatrix}.
 \end{aligned} \tag{13}$$



**Figure 1.2** Single qubit quantum gate illustration.  $Q$  represents the qubit and the single line which goes through the square gate represents that the system contains one qubit. A) Identity gate, B) Pauli X gate, C) Pauli Y gate, D) Pauli Z gate and E) Hadamard gate.

These rotation gates can also be written in the Pauli matrix term as:

$$\begin{aligned}
 R_x(\theta) &= \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} X = e^{\frac{-i\theta X}{2}} \\
 R_y(\theta) &= \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Y = e^{\frac{-i\theta Y}{2}} \\
 R_z(\phi) &= \cos\frac{\phi}{2} I - i \sin\frac{\phi}{2} X = e^{\frac{-i\phi Z}{2}}.
 \end{aligned} \tag{14}$$



*Figure 1.3 Quantum circuit of Rx, Ry and Rz gate.  $\pi/2$  means the rotation that the gate applies along the corresponding axis.*

A general rotation around an arbitrary axis can be given as a product of the previous rotations which known as unitary gate, which can be stated as:

$$U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta), \quad (15)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are the Euler angles of U in ZYZ basis.

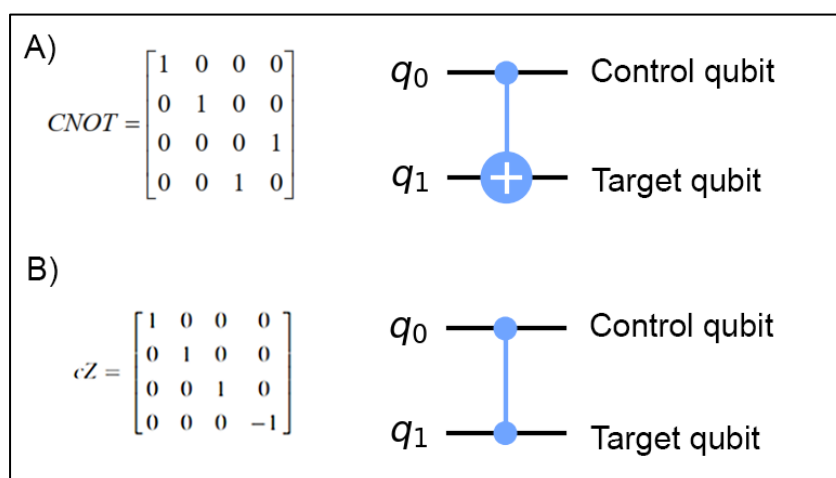
### 1.1.3 Multiple Qubit and Entangled States

Generally, quantum circuits contain two or more qubits. The qubits simultaneously store information, which most of this information can't be accessed during measurement. Nonetheless, this information can only be revealed by a joint measurement of the qubits [9]. Tensor product is used to describe the generate state of independent two or more qubits. For instance, when there are two qubits  $|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$  and  $|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle$  then the quantum state as tensor product of these two qubits system can be written in the equation as:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle. \quad (16)$$

However, for multiple qubit systems, the quantum state can't always be expressed as in the Equation (16). When 2 or more qubits are independently of each other and isolated, then it forms a closed system, and the state can be expressed as its product. Nevertheless, when qubits are not independent, then both qubits will bound together, and this special condition is well-known as entanglement [8].

Based on Equation (16), two-qubit state can be represented as 4 x 4 matrix form. Some examples of the most useful two qubit gates are controlled-NOT (CNOT) gate and controlled-Z(CZ) gate, which the matrix forms and the state diagram are describe in Figure 1.4. CNOT gate has two input qubits, the control qubit, and the target qubit. This gate implements an X-gate to the target qubit if control qubit is  $|1\rangle$  and there is no change in target qubit if control qubit is  $|0\rangle$ . Furthermore, CZ gate acts on two qubit gates and it amends the phase angle of target qubit by an angle of  $\varphi$  if the control qubit is in  $|1\rangle$  state. The target qubit is unchanged when the control qubit is  $|0\rangle$ .



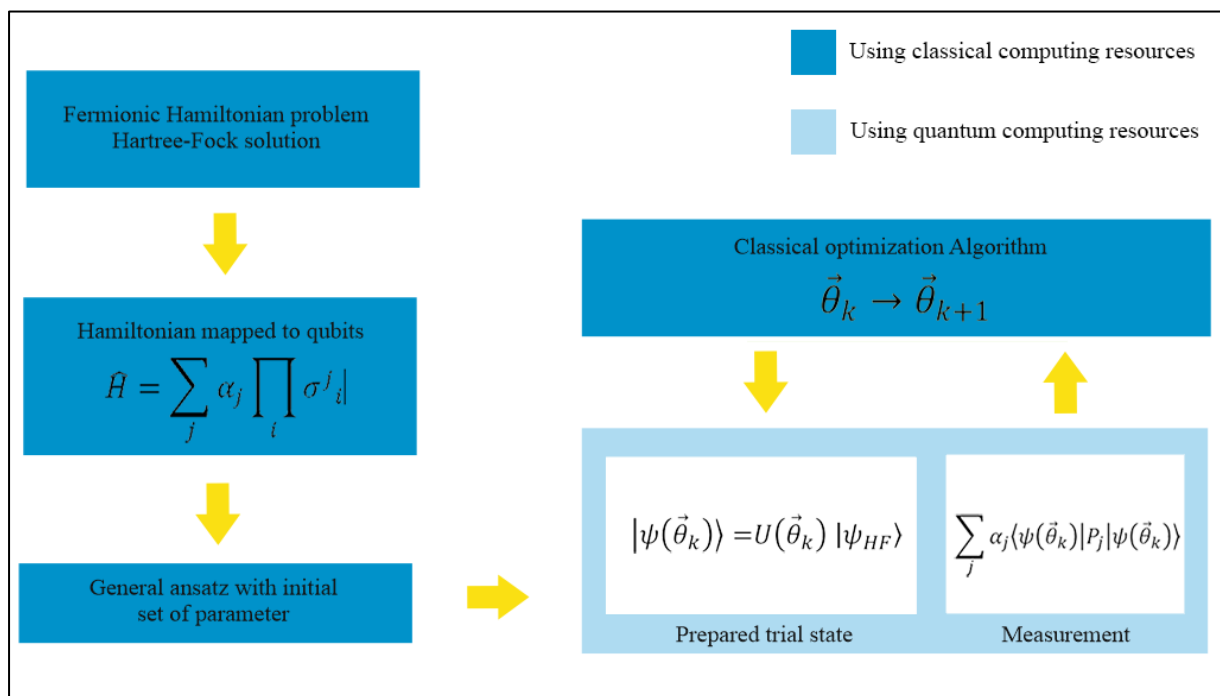
**Figure 1.4** Matrix form and quantum circuit of a) CNOT gate and b) CZ gate. These gates have two input qubits. CNOT gate flips the state of target qubit if the control qubit in  $|1\rangle$  and does nothing if it is in  $|0\rangle$  state. While CZ gate changes the phase angle of target qubit if the control qubit is in  $|1\rangle$  state.

## 1.2 Description of Variational Approach for Molecular System

Determining the molecular structure is the most debated topic among scientists because until this day there is still no efficient algorithm to solve the ground state energy of many-body interacting fermionic Hamiltonians. Several attempts have been conducted to solve this problem using quantum

computing which gives promising assets such as super rapid calculations compared to classical devices. To compute this problem in quantum computer, mapping between fermionic and qubit operators is considered and representing it as a local Hamiltonian problem on a set of qubits [10]. A k-local Hamiltonian H, consisted of terms acting on at most k qubits, is solved by finding the smallest eigenvalue,  $E_G$ ,

$$H |\psi\rangle = E_G |\psi\rangle . \tag{17}$$



**Figure 1.5** Schematic diagram of variational quantum Eigensolver. The energy of Hamiltonian by setting up using variational parameter  $\theta$ . The steps under darker blue box are performed with classical computer and lighter box with quantum computer. The Hamiltonian is then mapped, and initial set parameter is chosen for the initial wave function. The trial ansatz is then prepared on quantum computer with parameterized gates and then the process is repeated until the convergence criterion.



Variational quantum eigensolver (VQE) is one of the most prominent quantum algorithms and this algorithm's advantage surpasses other algorithms, in which don't require lot of qubits and gate for small system [11]. In VQE, both quantum and classical resources are used to discover variational solutions to eigenvalue problems, as the workload of VQE is shown in Figure 1.5. As a typical setup, a molecule's ground state trial wavefunction is built from classically computed Hartree-Fock wave functions for single and double excitations [12]. Subsequently, the trial wave function is adapted on quantum computer and the expected value of the Hamiltonian is measured. Next, the trial wave function's parameters are optimized on a classical computer based on the variational principle. This trial wave function or ansatz are variationally set until the expected value of the electronic Hamiltonian [11].

$$E \leq \frac{\langle \psi(\vec{\theta}) | \hat{H}_{el} | \psi(\vec{\theta}) \rangle}{\langle \psi(\vec{\theta}) | \psi(\vec{\theta}) \rangle}, \quad (18)$$

where  $\psi(\vec{\theta})$  is the ansatz with dependency on the vector of parameter  $\vec{\theta}$ ,  $E$  is the ground state of energy and  $\hat{H}_{el}$  is the electronic Hamiltonian. The Hamiltonian of indistinguishable fermions must be mapped onto Hamiltonian of distinguishable qubits to examine the energy on quantum computer by using a mapping method. There are 3 common mapping method such as Jordan-Wigner, Parity and Bravyi-Kitaev. Regardless of this mapping option, the resulting qubit Hamiltonian can be formulated as:

$$\hat{H} = \sum_j \alpha_j P_j = \sum_j \alpha_j \prod_i \sigma_i^j, \quad (19)$$

where  $\alpha_j$  are real scalar coefficients that depend on the single-electron or two-electron excitations.  $i$  represents which qubit the Pauli operator takes place and  $j$  is term of Hamiltonian.  $P_j$ 's are Pauli strings which are defined by a product of Pauli matrices  $\sigma_i^j$ ,  $P_j \in \{I, X, Y, Z\}^{\otimes N}$ , with  $N$  is the number of qubits which is used in the model of wavefunction. After the qubit Hamiltonian is

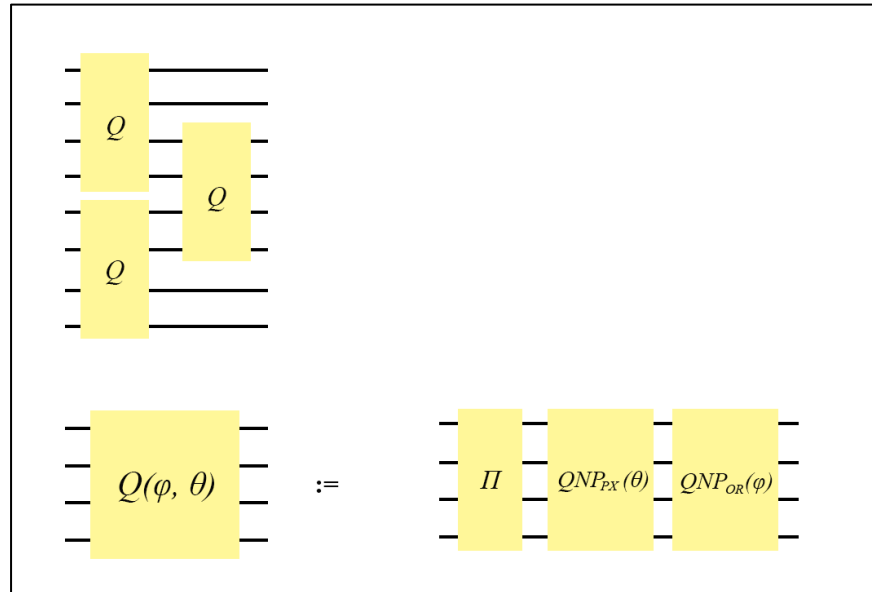
obtained from classical device and the ansatz is chosen, the trial wave function  $\psi(\vec{\theta})$  is arranged in quantum computer and the energy that is measured by the quantum computer can be expressed as:

$$E(\vec{\theta}) = \sum_j^N \alpha_j \langle \psi(\vec{\theta}) | \prod_i \sigma_i^j | \psi(\vec{\theta}) \rangle, \quad (20)$$

where N is number of terms in the Hamiltonian and  $\vec{\theta}$  is the vector of variational parameter. This Hamiltonian can possess up to  $M^4$  terms, where M is the number of basis functions. However, the state preparation step must be performed repeatedly for each term due to Pauli string operator representation of these terms. Additionally, to assemble enough expectation value statistics, all individual terms also must be measured in enough times. This energy calculation method is known as Hamiltonian averaging [13].

The main issue in VQE is determining the efficient entangler circuit. There are some characteristics which can lead to the efficient term, such as representing target quantum states in the limit of an intermediate depth efficiently, having a low number of independently realizable gate elements, showing a simple pattern when the gates are adapted, how compatible the circuit with device connectivity due to its sparse spatial connectivity and perform simple analytical gradient method and achieve numerical convergence during optimization of the VQE energy [14]. There are a variety of ansatz designs that satisfy the characteristic of efficient entangler circuit. This ansatz design method can be classified into three categories: i) Chemistry-inspired ansatzes which use the fundamental comprehension from quantum chemistry in a way that each term in ansatz defines the electron configuration. ii) The second is hardware-efficient ansatzes, which are built up from certain set of gates that can be easily implemented on quantum devices, however it can't represent a chemical interpretation for each term. iii) The last one is Hamiltonian variational ansatzes which are usually described as the intersection between chemistry-inspired and hardware-efficient.

In this work, we are focusing on Chemistry-inspired ansatz. The gate synthesis features the exact preservation of all quantum numbers for each individual element, which is represented as quantum number preserving (QNP). The gate synthesis is focused to spin-restricted fermionic symmetry under Jordan-Wigner mapping method. As a typical setup,  $M$  real orthogonal spatial orbitals are defined as  $\{|\phi_p\rangle\}_{p=0}^M$ . Subsequently,  $\alpha$  and  $\beta$  are denote as spins for each spatial orbitals,  $|\psi_p\rangle := |\phi_p\rangle|\alpha\rangle$  ( $|\psi_p\rangle := |\phi_p\rangle|\beta\rangle$ ) for a total of  $N := 2M$  spin orbitals in spin-constricted formalism. The number of spin orbitals is associated with the number of  $N$  qubits possessing interleaved ordering number ...  $|1_\beta\rangle |1_\alpha\rangle |0_\beta\rangle |0_\alpha\rangle$ . The creation and annihilation acting in fermionic state are described in the qubit operators acting as, X, Y and Z based on Jordan-Wigner mapping in the ordering of “ $\alpha$ -then- $\beta$ ”,  $p^\pm := (\hat{X}_p \mp i\hat{Y}_p)/2 \otimes_{q=0}^{p-1} \hat{Z}_q$  and  $\bar{p}^\pm := (\hat{X}_{\bar{p}} \mp i\hat{Y}_{\bar{p}})/2 \otimes_{q=0}^{p-1} \hat{Z}_{\bar{q}} \otimes_{q=0}^{M-1} \hat{Z}_q$ .



**Figure 1.6** Quantum circuit representation of  $Q$  building block. The four-bit  $Q$  gate is obtained from the four-local-nearest-neighbour qubits, and it is ordered alternatively. Each  $Q$  gate has two parameters, one parameter of four-qubit spatial orbital rotation gate  $QNP_{OR}(\varphi)$  and the other, four-qubit of pair exchange gate  $QNP_{PX}(\theta)$ . Adapted from [15].

According to [15], such ansatz gate operator on qubit is used, and 3 operators are introduced which associates with this ansatz, as  $N_\alpha, N_\beta$  and  $\hat{S}^2$ .  $N_\alpha$  and  $N_\beta$  are the projection of  $\alpha$  and  $\beta$  orbitals in XYZ, which can be expressed as:

$$\begin{aligned}\hat{N}_\alpha &:= \sum_p p^\dagger p = \sum_p \frac{\hat{I} - \hat{Z}_p}{2} \\ \hat{N}_\beta &:= \sum_p \bar{p}^\dagger \bar{p} = \sum_p \frac{\hat{I} - \hat{Z}_{\bar{p}}}{2},\end{aligned}\tag{21}$$

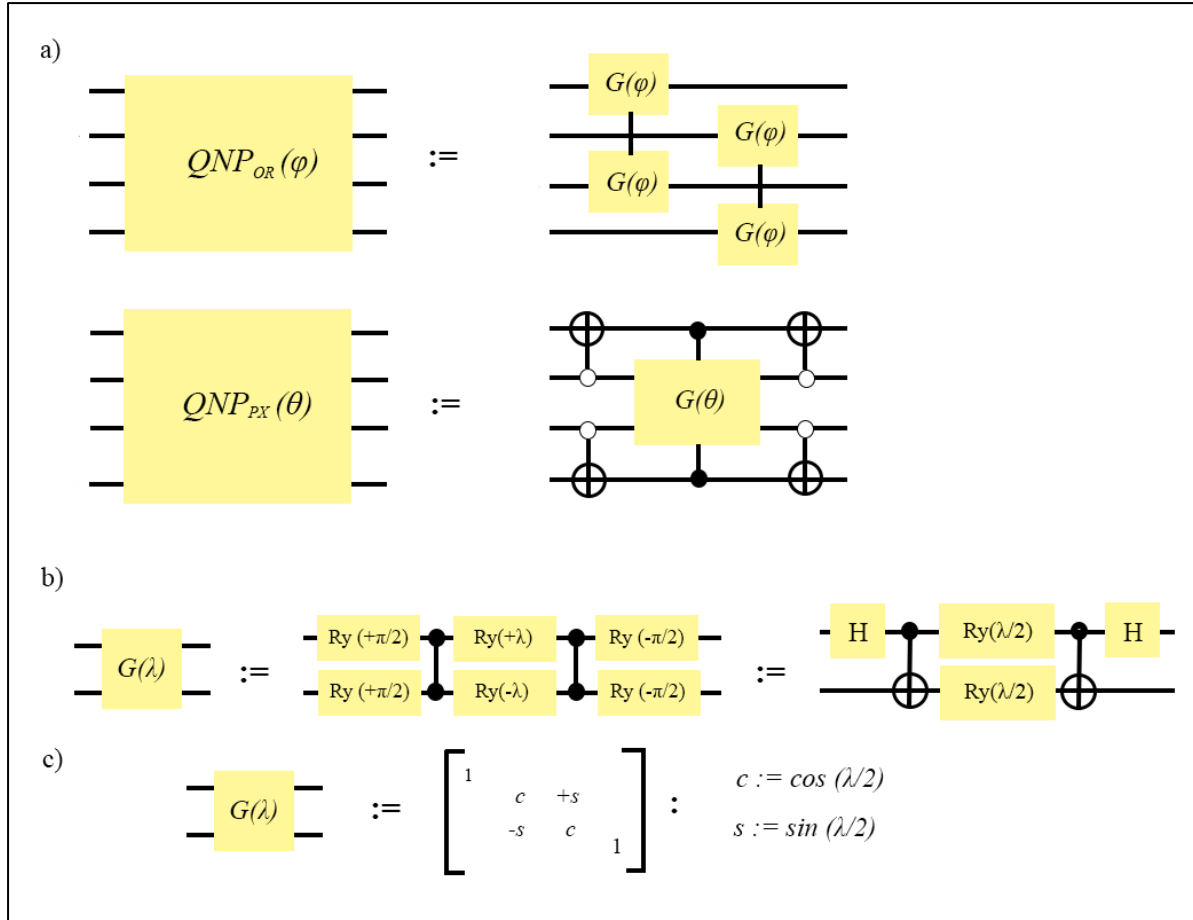
and the spin-squared operator,  $\hat{S}^2$  can be defined as:

$$\hat{S}^2 := \sum_{p q} p q^\dagger \bar{p}^\dagger \bar{q} + \frac{(\hat{N}_\alpha - \hat{N}_\beta)}{2} + \frac{(\hat{N}_\alpha - \hat{N}_\beta)^2}{4}.\tag{22}$$

However, this operator doesn't fulfill a local description in terms of Pauli operators in the Jordan-Wigner mapping.

One might expect that no local gate fabric could exactly preserve all three fermionic quantum numbers, because spin should be conserved until external force makes things different. The quantum-number-preserving gate synthesis should be achieved as demonstrated in Figure 1.6. In [15], gate structure was introduced, which represents multi electrons entanglement consists of many building blocks, labelled as  $\hat{Q}$ . This  $\hat{Q}$  building block was obtained from the reduction of five-parameter building block into two-parameter, one is the four-qubit spin-adapted spatial orbital rotation gate  $QNP_{OR}(\varphi)$  and the other parameter is the four-qubit diagonal pair exchange gate  $QNP_{PX}(\theta)$  as shown in Figure 1.6.  $QNP_{OR}(\varphi)$  applies the spatial orbital Given rotation  $G(\varphi)$ , which is described as 2-qubits space gate structure acting in  $\phi_0$  and  $\phi_1$ , where  $\phi_0$  and  $\phi_1$  are the basis vector as depicted in Figure 1.7 (b). This Given rotation  $G(\varphi)$  can be expressed as  $|\phi_0\rangle \rightarrow \cos\left(\frac{\varphi}{2}\right)|\phi_0\rangle + \sin\left(\frac{\varphi}{2}\right)|\phi_1\rangle$  and  $|\phi_1\rangle \rightarrow -\sin\left(\frac{\varphi}{2}\right)|\phi_0\rangle + \cos\left(\frac{\varphi}{2}\right)|\phi_1\rangle$  using same orbital rotation which are implemented in the  $\alpha$  and  $\beta$  orbitals. While  $QNP_{PX}(\theta)$  performs diagonal pair

Given rotation  $G(\theta)$  as  $|0011\rangle \rightarrow \cos\left(\frac{\theta}{2}\right)|0011\rangle + \sin\left(\frac{\theta}{2}\right)|1100\rangle$  and  $|1100\rangle \rightarrow -\sin\left(\frac{\theta}{2}\right)|0011\rangle + \cos\left(\frac{\theta}{2}\right)|1100\rangle$ . Figure 1.7 illustrates the details of  $QNP_{OR}(\varphi)$  and  $QNP_{PX}(\theta)$  gate structure constructed by Given rotation gate,  $G$ .



**Figure 1.7** Detail of four-qubits spatial orbital gate  $QNP_{OR}(\varphi)$  and four-qubits diagonal pair exchange gate  $QNP_{PX}(\theta)$  in which Given rotation  $G$  is implemented (a). The Given rotation is constructed from  $4 \times 4$  matrix which consist of  $c = \cos(\lambda/2)$  and  $s = \sin(\lambda/2)$ , where  $\lambda$  represents the angle of rotation (b and c). Adapted from [15].

## 1.3 Quantum Gate Synthesis and Optimization

### 1.3.1 Theoretical background behind the quantum gate synthesis

The value of an n-qubit register can be modified with quantum logic gates applied to selected qubits. However, implementing high number quantum gate in circuit can cause prohibitive difficulty for quantum computers. Therefore, an attempt should be performed to find a more efficient way by turning the desired quantum program acting on qubits into a low complexity circuit. This work is usually known as gate synthesis or circuit synthesis.

The common strategy of gate synthesis is to decompose the circuit into a sequence of two-qubit gates [16]. It is possible to decompose two-qubit gates into circuits containing one-qubit gates, and a canonical two-qubit gate known as the controlled-not (CNOT). There were several works that have been done to carry out two-qubit decomposition. The first algorithm was published by combining two-qubits gate decomposition with QR decomposition formula. The author investigated that any operator of n qubits can be fabricated with a circuit containing  $O(n^3 4^n)$  elementary quantum gates [17]. Under this work, a lower bound on the number of elementary two-qubit gates was presumed to create an arbitrary n-qubit operator:

$$\Omega(n) = \frac{1}{9} 4^n - \frac{1}{3} n - \frac{1}{9} \quad (23)$$

Other works demonstrated the advantages with different complexities [18, 19, 20, 21, 22]. However, all the mentioned methods considered full connectivity between the qubits during the composition and to implement the quantum algorithms in the latest quantum computers, the quantum circuits must follow the device's constraints. To achieve this, another method has been proposed such as introducing additional SWAP operations to make qubits adjoining each other by having two-qubits controlled gates. But this strategy can still cause error-correction, mapping and scheduling

measurement are still necessary to be applied when the quantum computational results are obtained [23].

The optimization of single qubit rotations and the gate structure is the most challenging and interesting field in simplifying gate decomposition problem. Madden and Simonetto have successfully eliminated trivial single-qubit gates. They derived a 14-CNOT 4-qubit Toffoli decomposition and compressed QSD by a factor of two without practical loss of fidelity [24]. Rakyta and Zimborás proposed a new algorithm based on iterative, sequential optimization (SO) of parameters to match it with quantum hardware limitations in inter-qubit connectivity. Without having extra SWAP gates, this method can be adopted to increase the total fidelity of the quantum circuit and it turned that 15 and 63 CNOT gates are enough to decompose general 3- and 4-qubits unitary [25]. However, this numerical approach only applies for general unitaries.

Another algorithm was also introduced by this team later which was found to be more efficient to decompose not only for general unitary, but also any special unitary, such as Variational Quantum Eigensolver (VQE), which possesses special gate. Their methodology is based on adaptive circuit compression, in which by compressing the quantum circuit by sequential removal of two-qubit gates and the remaining building blocks are transformed into reduced gate structure by iterated learning cycles. The algorithm was tested for decomposition 3-, 4- and 5-qubits unitaries from several references and the numerical experiment results exhibited that highly optimized gate counts was obtained and could reduce more than 50% in 21% percent of the decomposed unitaries [26]. This remarkable algorithm will be applied in this current work in SQUANDER [27] package.

To evaluate the distance of  $d \times d$  unitary  $V$  as a synthesized product from the original unitary  $U$ , Hilbert-Schmidt test can be utilized which is shown as follows:

$$C_{HST}(U, V) = 1 - \frac{1}{d^2} |\text{Tr}(V^\dagger U)|^2. \quad (24)$$

The fidelity of gate can be derived by aligning the state of fidelities from the output states (after the  $U$  and  $V$  transformation, respectively) with Haar distribution and it can be calculated via Hilbert-Schmidt test according to the equation.

$$\bar{F}(U, V) = 1 - \frac{d}{d+1} C_{HST}(U, V). \quad (25)$$

Alternatively, Madden implemented a different approach, Frobenius norm-based metric to measure the distance between two unitaries  $U$  and  $V$  :

$$f(U, V) = \frac{1}{2} \|V - U\|_F^2 = d - \text{Re} [\text{Tr}(U^\dagger V)], \quad (26)$$

and described a Frobenius based fidelity  $\bar{F}_F(U, V)$  by

$$\bar{F}_F(U, V) = 1 - \frac{d}{d+1} + \frac{1}{d(d+1)} (d - f(U, V))^2. \quad (27)$$

And it can be seen that  $\bar{F}_F(U, V) \leq \bar{F}(U, V)$ .

Equation (24) and (26) can be proficiently applied as a cost function in optimization process to find the best approximation of unitary  $U$ . Besides, the gradient descent optimization calculations can be improved numerically by calculating their gradient components analytically in relation to the free parameters of the decomposing quantum circuit, such as Broyden–Fletcher–Goldfarb–Shanno (BFGS) and Adaptive Moment (ADAM) optimization algorithms which is used in SQUANDER package which is used in this work as explained in | Appendix 1, and these 2 norms of cost function are observed and applied in the numerical optimization later on in this work.

Nevertheless, in order to be compatibly applied for full approximation in quantum compiling, the cost function needs to be extended and can be expressed in the term with “bit-flit”,  $X$  as proposed in [28] :



$$\begin{aligned}
& C_{lhs}^{circuit} \\
& = 1 \\
& - \frac{1}{d^2} \left[ |Tr(V^\dagger U)|^2 + \left(\frac{n-1}{n}\right) \sum_{j=1}^n |Tr(X_j V^\dagger U)|^2 \right. \\
& \left. + \left(\frac{n-2}{n}\right) \sum_{j<k} |Tr(X_j X_k V^\dagger U)|^2 + \dots + \frac{1}{n} \sum_{j<k<l<\dots} |Tr(X_j X_k X_l \dots V^\dagger U)|^2 \right].
\end{aligned} \tag{28}$$

This equation allows faster convergence when it is applied to classical compilation of quantum circuits. Unfortunately, the longer term in Equation (28) makes the gradient calculation even more difficult and a solution is introduced by simplifying the expression in Equation (28) than calculating each term explicitly [28]. Therefore,  $\alpha_1, \dots, \alpha_k$  is updated by using coefficient  $\alpha$  during the optimization and the cost function turns into:

$$C_L^{(1)}(\theta) = 1 - |\langle 0 | V^\dagger(\theta) | \psi_0 \rangle|^2 - \alpha_1 \sum_{j=1}^n |\langle 0 | X_j V^\dagger(\theta) | \psi_0 \rangle|^2, \tag{29}$$

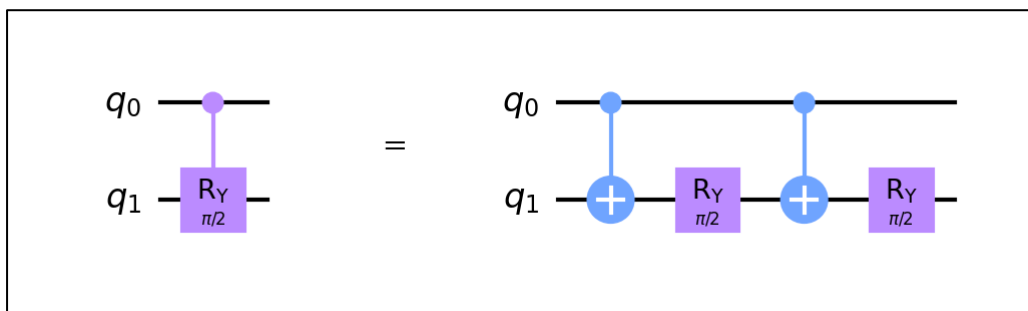
where  $\theta$  is the number of parameters. When  $C_L^{(1)}$  comes close to 0, then the value of  $\alpha$  is gradually declined and reduce the influence of final term in Equation (30) since the cost function comes near to zero. The value of  $\alpha(k) = \sqrt{C_L^{(1)}(k)}$  and  $\alpha(0) = 1$  is considered in this update. It can be considered that  $C_L^{(1)}(\alpha) =$  is always positive since  $|0\rangle$  and all “bit-flipped” states establish an orthogonal basis. The goal of this weighting scheme is to make the grip narrower around the state  $|0\rangle$  while  $C_L^{(1)}(\alpha)$  surrounds zero. In this condition, all the flip term are stopped and  $V(\theta)|0\rangle \approx |\psi_0\rangle$  [28].

### 1.3.2 Adaptive Circuit Compression Method

Adaptive circuit compression method is generally based on the elimination of parametric two-qubit gates from the circuit. The given quantum circuit is compressed, until there will be no additional

two-qubit components. Controlled rotation (CR) two-qubit gates are used in this algorithm, which can be tuned via continuous parameter. Moreover, these two-qubit gates are advantageous due to its versatility to represent quantum circuit elements [26]. CR gates can be mapped to special two-qubit gates such as CNOT and CZ gate. In principle, CR gates can be decomposed into two CNOT gates and any CNOT gate in the circuit can be transformed into CR gate, since CNOT gate is the special case of CR gate. In addition, any unitary  $U$  can be synthesized since the set of CR and general qubit rotation (U3) gates form a universal gate set. In particular, SQUANDER implements controlled  $R_y$  rotation gate, and this gate is further decomposed into elementary gates as shown in Figure 1.8. By taking this approach, we are able to reconstruct the structural combinatorial problem of placing the elementary two-qubit gates in a circuit as an optimization problem over continuous variables [26].

The meaning of ‘adaptive circuit compression’ is that CR gates correlate during compression, all of them respond when a two-qubit block is removed from the design, and our circuit compression strategy would not be limited to local cancellations of two-qubit gates. If the reduced gate structure optimization problem can be solved, then the two-qubit block chosen for the system can be eliminated as a dispensable one.



**Figure 1.8** Controlled  $R_y$  gate which is generally expressed in terms of two CNOT and  $R_y$  gates.

A quantum circuit which approximates unitary  $U$  is initially determined to commend the decomposition with optimized gate count by periodical repeated unit cells which are constructed

from two-qubit building blocks. Each building blocks are fabricated from two general qubit rotation ( $U3$ ) gates which run on each qubit and one single controlled  $R_y$  gates [26]. After the unitary  $U$  is successfully reproduced by trail circuit, the algorithm begins to run the compression cycles. For each iteration, random two-qubit building block is picked and eliminated from the system if it can follow to the changed structure by looking a new solution for the optimization process. In the end, when the algorithm can't find other removable building blocks, then CR gates is expanded and the gate synthesis of the unitary is terminated with a final optimization iteration [26].

## 2 | Methodology

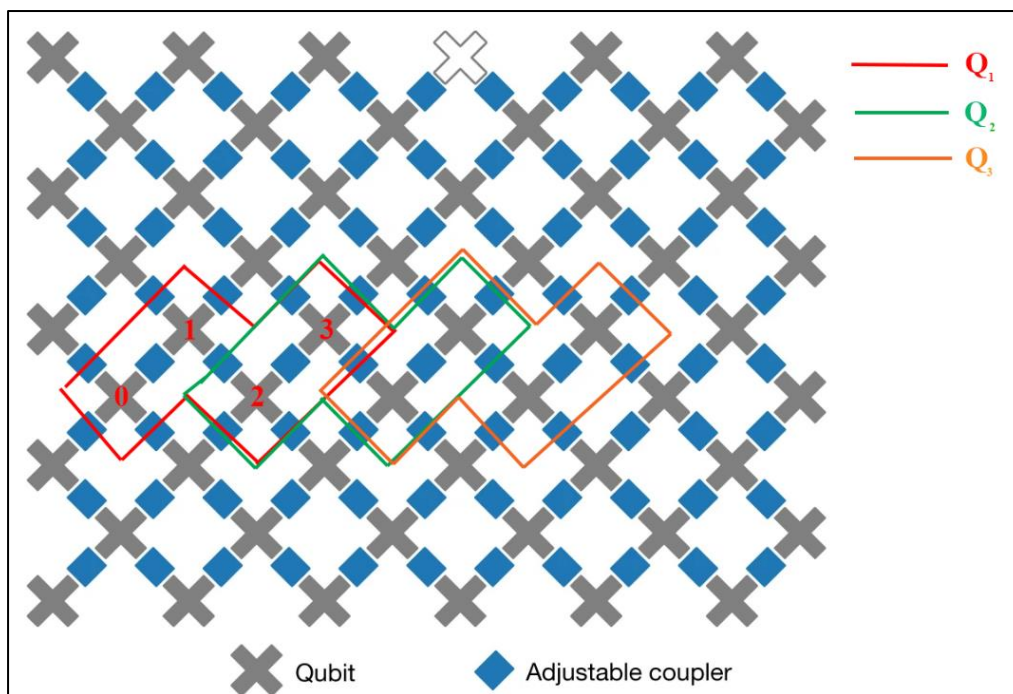
This chapter describes the details of work involved in the quantum gate synthesis circuit generating entangled state and calculating ground state energy for fermionic systems. Initially, a quantum circuits is constructed for  $Q$  building block as shown in Figure 1.6 and the unitary matrix representation of this  $Q$  gate is decomposed via adaptive circuit compression algorithm in SQUANDER along line topology, because this method can be run in the constraints of current quantum devices. 6 different cost functions with different number of correction factor,  $k$ , with  $k$  from 0 to 2 for each Frobenius and Hilbert-Schmidt norm of cost function based on Equation (29), are applied in two common optimizers, ADAM and BFGS for numerical optimization method and the goal of this work is to compare these optimization algorithms and observe which one can provide faster decomposition process for the given quantum circuit as proposed in [15].

### 2.1 Quantum Circuits Generation

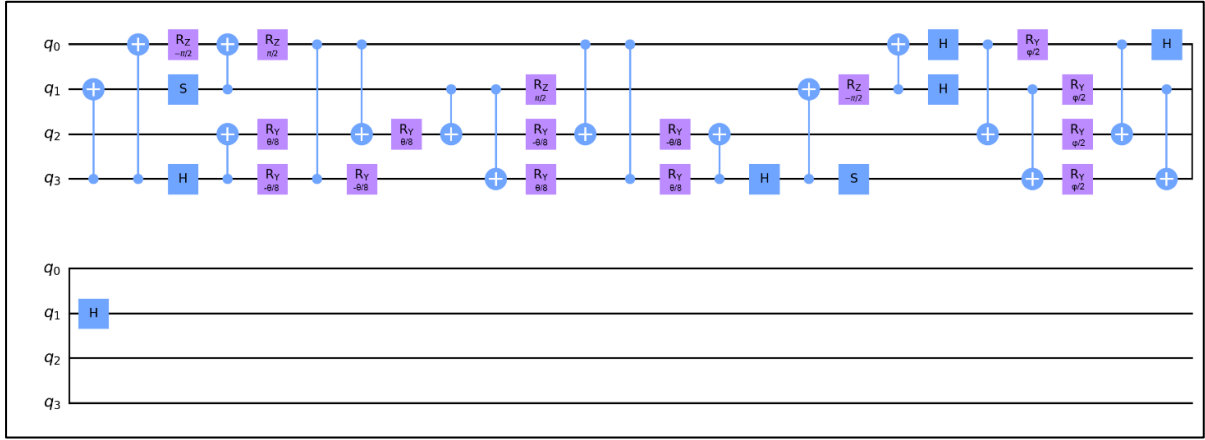
Numerical investigation has been conducted for the synthesized gate  $Q = \prod_i QNP_{PX}(\theta)QNP_{OR}(\varphi)$  as shown in Figure 1.6 to determine the ground state energy of p-benzyne and naphthalene [15]. P-benzyne, which possesses biradical open-shell singlet ground state, accompanied by two unpaired electrons and naphthalene, which is not basically biradical, however it shows several natural orbitals with significant deviations from Hartree-Fock (HF) natural orbital occupation number. The result revealed that the produced gate provided higher accuracy than either HF or doubly occupied configuration interaction (DOCI). However, this simulation was conducted with a full connectivity circuit of  $Q$  gate structure, where this method is rather be avoided since it is hard to be adapted with the constraint of quantum computer.

In this work the unitary matrix representation of  $Q$  gate structure as shown in Figure 1.6 is decomposed and optimized for line connectivity topology, because we need to adapt it with the

constraints of existing quantum devices. Two qubit gate operations are assumed as a pair during the decomposition, such as 0-1, 1-2, and 2-3 qubit pairs. With this decomposition, these qubit pairs can be implemented to existing quantum devices, for instance the Sycamore processor. Besides, the accomplishment of gate  $Q$  on line topology entitles for scaling up of the variational quantum circuit on Sycamore quantum chip by arranging the involved qubits into “Z” shaped geometry as depicted in Figure 2.1. In this Figure, it is also depicted that the connection between each  $Q$  block is exhibited since  $Q$  blocks need to overlap with other 2 qubits as previously elucidated in Figure 1.6.



**Figure 2.1** The qubits placement in the Sycamore processor's layout. Red contour determines the set of qubits on which block gates  $Q = \prod_i QNP_{PX}(\theta)QNP_{OR}(\varphi)$  and 0,1,2 and 3 numbers are the qubits that are associated with this  $Q$  block. Another colour represents another  $Q$  block and it can be viewed that between each  $Q$  blocks, overlapping exist on 2 qubits. Edited from [29].

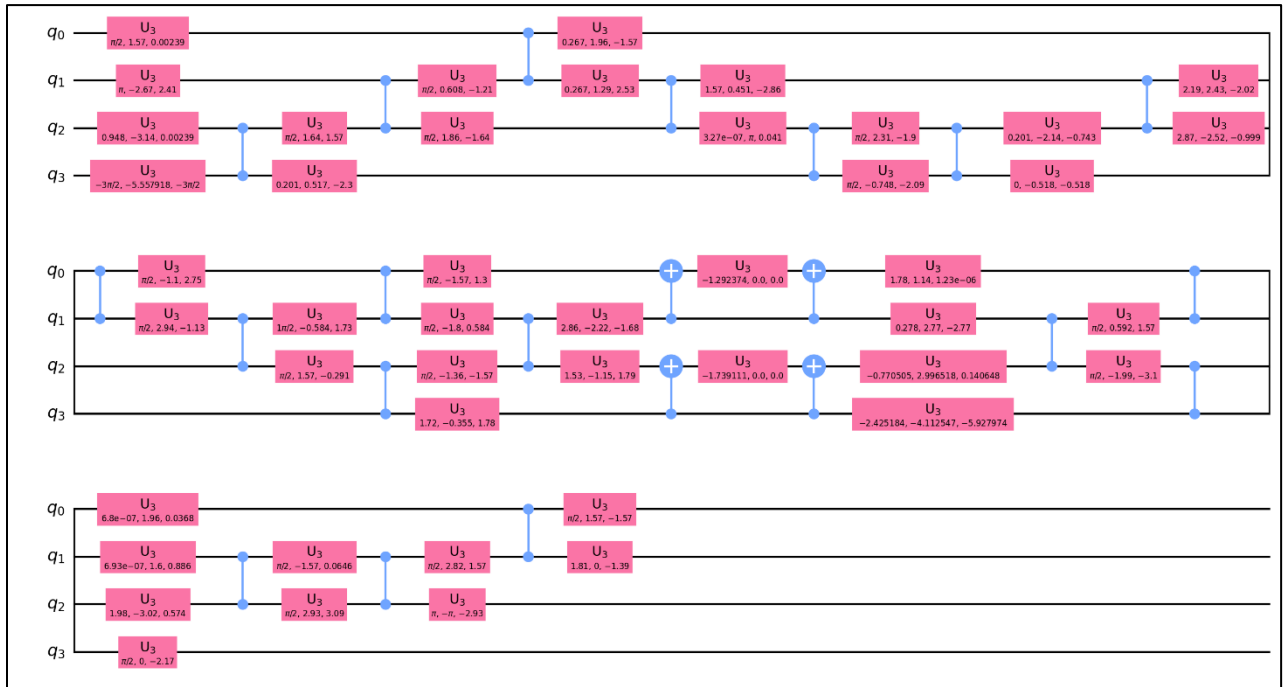


**Figure 2.2** Quantum circuit representation of  $Q = \prod \cdot QNP_{PX}(\theta)QNP_{OR}(\varphi)$  as proposed in [15]. In this figure, it can note that there is no limitation in the connectivity between qubits.

The proposed quantum gate structure as exhibited in Figure 2.2 is later decomposed onto architectures with line connectivity. Using the newly developed adaptive decomposition strategy of SQUANDER package, the optimized  $Q$  block circuit consists of pair exchange gate  $QNP_{PX}(\theta)$  and orbital gives rotation gate  $QNP_{OR}(\varphi)$  on topology can be achieved. This gate construction is constructed using Qiskit package in Python. Besides, as it was proposed in [15], by placing fixed spin-adapted orbital rotation gate,  $\prod = QNP_{OR}(\pi)$ , can increase the efficiency during gradient based parameter optimization,  $Q = QNP_{OR}(\pi)QNP_{PX}(\theta)QNP_{OR}(\varphi)$  block circuit was then generated and further investigated in this work due to more advantageous for gradient-based parameter optimization process as proposed in [15]. The line connectivity decomposition gate of  $Q = QNP_{OR}(\pi)QNP_{PX}(\theta)QNP_{OR}(\varphi)$  on topology with calculated for  $\theta = 0.68\pi$  and  $\varphi = 0.21\pi$  can be viewed in Figure 2.3.

## 2.2 Brief description of optimization method

In this work, the decomposition of the extended gate  $Q = QNP_{OR}(\pi)QNP_{PX}(\theta)QNP_{OR}(\varphi)$  was examined on line connectivity topology using the adaptive algorithm which is implemented in SQUANDER package. The simulation was performed in a computer device with Intel Core i5-4200U @1.6 GHzx4 processor having 8 GB memory. Two optimization algorithms were used, the BFGS and ADAM optimization algorithms. Gradient-based optimization exploits the information from cost function derivatives. First-order optimizers use only first-order derivatives of cost function, while second-order optimizers harness the Hessian of cost function, or the local curvature of learning landscape. Besides BFGS and ADAM, there are still some well-known optimizers such as Simple gradient descent and RMSProp, however in this work, we just limited the investigation only for BFGS and ADAM optimizers.



**Figure 2.3** Decomposed quantum circuit representation of the  $Q =$

$QNP_{OR}(\pi)QNP_{PX}(\theta)QNP_{OR}(\varphi)$  gate on topology for for the case  $\theta = 0.68\pi$  and  $\varphi = 0.21\pi$ . It can be seen that nearby qubits are connected over a line.

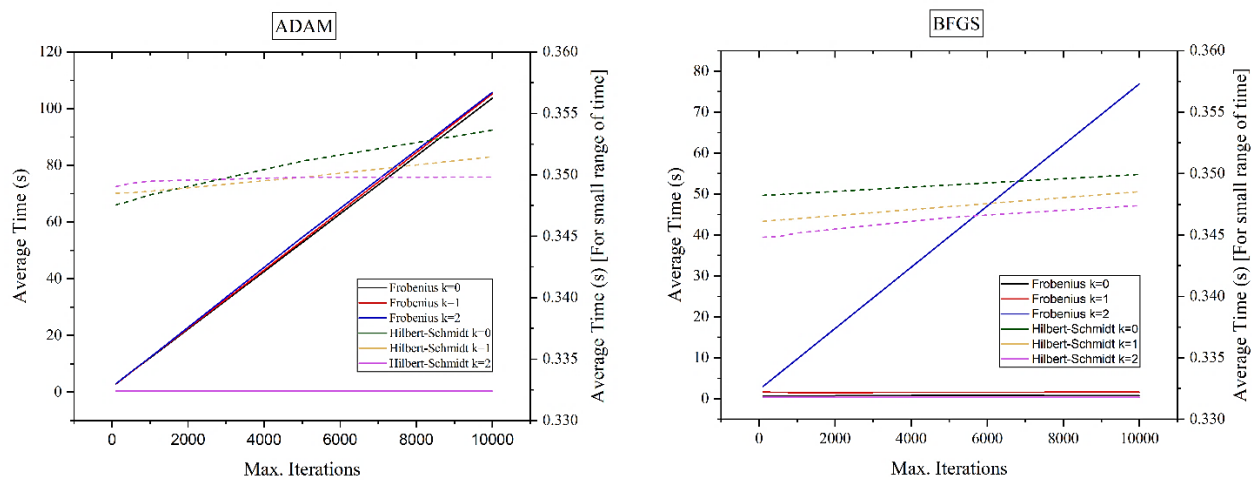
Quantum circuit of Q block,  $Q = QNP_{OR}(\pi)QNP_{PX}(\theta)QNP_{OR}(\varphi)$ , was prepared and parameter  $\theta$  was set to  $0,68\pi$  and  $\varphi$  was fixed to  $0.21\pi$ . Then we optimize the rotational parameters for randomly picked  $\theta$  value between 0 and  $2\pi$ . Once it was ready, the chosen optimization algorithm between BFGS and ADAM with 6 cost functions represents different term for each Frobenius and Hilbert-Schmidt norm of cost functions according to Equation (24) and (26), were run. The numerical analysis was conducted, and the iterations proceed as it was input. Lastly, when the algorithm does not find any progress after some number of iterations, the optimization will be terminated, and the gate synthesis process will be completed.

The observation was conducted in different iteration numbers varying from 100, 500, 1000, 5000 and 10000 with tolerance limit was set at  $10^{-8}$ . The average time and average decomposition error from the quantum gate synthesis were collected and plotted. Subsequently, the results were observed and compared between the 6 different norm of cost functions. Thus, from those two optimizers, an investigation was also undergone to examine the influence of tolerance variation:  $10^{-12}$ ,  $10^{-10}$ ,  $10^{-8}$ ,  $10^{-6}$ , and  $10^{-4}$  towards decomposition time and decomposition error observing at small iteration number 100 and higher iteration number 5000 and we can conclude which algorithm is the most effective one for the quantum gate synthesis process of the given quantum circuit as proposed in [15].



### 3 | Results and Discussion

Using Adaptive decomposition algorithm as presented in [26], the gate synthesis was successfully carried out as described in 2.2. All the obtained results were collected, and the average value of several executions were calculated and represented here. Besides, stable execution time and results were found out for several runs for each norm of cost functions in both optimizers.



**Figure 3.1** Iteration count over average execution time comparison for ADAM and BFGS optimizers with 6 different norms of cost functions in gate synthesis of  $Q$  gate structure. The dashed lines exhibit the enlarged view of 3 Hilbert-Schmidt norm of cost function and the average time range view of these 3-cost function is represented in the right side of the graph.

Figure 3.1 demonstrates the execution time plot for both ADAM and BFGS optimizers during the synthesis. As one can see, with Frobenius form of cost functions in ADAM optimizer, the synthesis completion time grows steadily as the iteration number increases. If we look thoroughly through this time result, with the addition of number of correction order number,  $k$ , in cost function, the execution time is also slightly more required which can be investigated by the increment of the gradient for 3 Frobenius cost function curves. Meanwhile, with Hilbert-Schmidt test of cost function, ADAM

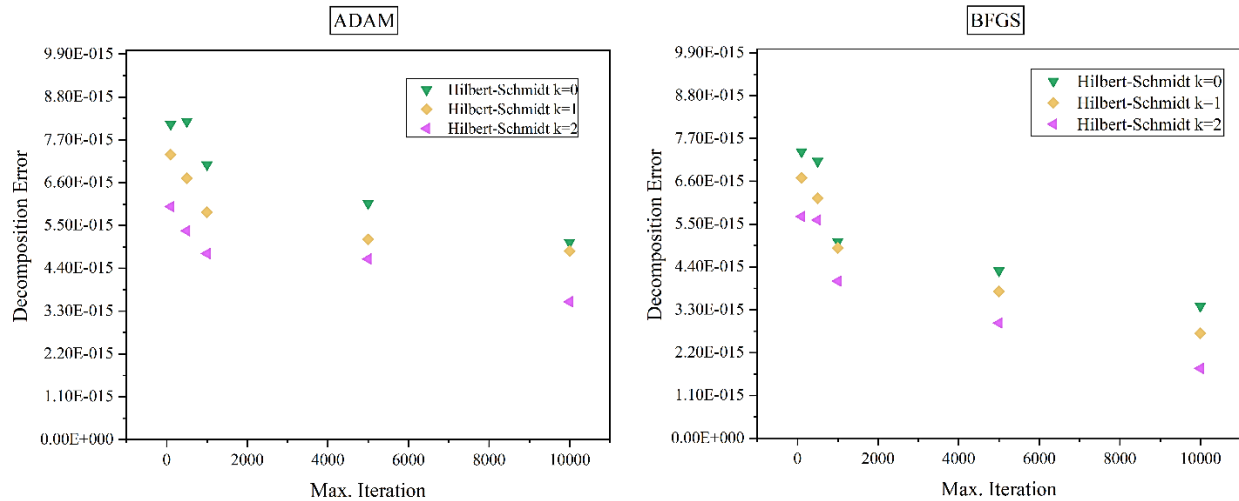
optimizer shows stable completion time for almost all of terms in Hilbert-Schmidt norm. Additionally, if we look through the closer view of the result from Hilbert-Schmidt cost function, the cost function without correction order reaches the completion time at low maximum iteration = 100 number about 0.35 seconds which is almost similar compared for  $k = 1$  and  $k = 2$ . Since these values are quite close to each other, it means that the effect of correction order becomes less important for Hilbert-Schmidt norm of cost function. Nonetheless, when the maximum iteration number is set to a higher number, the cost function with correction order possesses almost similar decomposition time. It can be noted that at maximum iteration number equals to 10000, the cost function without correction order reaches the execution time near to 0.35 seconds, similarly when the correction order is added to 1 and 2.

Furthermore, BFGS optimizer exhibits different trends in execution time during this gate synthesis procedure. With Frobenius norm of cost function, the completion time is quite rapid close to 0 second, when the number of correction order,  $k$ , is not introduced. But, when  $k$  is equal to 1, the average time is somewhat escalates as depicted in Figure 3.1 about to around 1.6 seconds which remains stable until maximum iteration number is set to higher value. The red line curve, which portrays Frobenius cost function with  $k = 1$ , is moderately higher than the curve represented Frobenius cost function without any correction factor, represented by black line, can reach the decomposition under 0.74 seconds when the maximum iteration is set to 10000. Interestingly, dramatic surge of time required is found when the number of correction factor,  $k$ , is increased to 2 as the number of iterations is ascended and it can be noted that the gate synthesis completion time increases linearly, therefore in the further investigation, this cost function will be excluded, looking to the completion time produced, we won't reach fast decomposition if we apply this algorithm. Whilst, , the execution time of Hilbert-Schmidt cost functions applied in BFGS optimizer are almost as fast as in ADAM and interestingly when we look up to the enlarged curves of the 3 cost function

of Hilbert-Schmidt, the decomposition time is only diverged in the scale of  $10^{-3}$  compared to the ADAM optimizer, without the presence of correction order, the completion time can be reached at around 0.35 seconds when maximum iteration equals to 100, while the other cost functions, which contain correction order,  $k = 1$  and  $k = 2$ , exhibit results with more or less the same value. But, when this maximum iteration value is increased, the decomposition time is practically similar, and it reaches around 0.35 seconds at maximum iteration number = 10000 and with the addition of correction order, the decomposition is completed at the same time as depicted in Figure 3.1.

Decomposition error results were also obtained and depicted in Figure 3.2. Unfortunately, although Frobenius norm of cost functions without correction order or with the correction order equals to 1 produces fast execution time, however, their decomposition error is far beyond the expectation because they display decomposition error at high number, above 1. This value is not demanded as an efficient algorithm to be applied for the quantum gate synthesis. Therefore, in the further step, these two cost functions are disbarred from the investigation.

In Figure 3.2 left graph, all Hilbert-Schmidt norm of cost functions exhibit small value of decomposition error in ADAM optimizer. One can note that even when the maximum iteration is enhanced, the errors most likely remain the same. Although, the correction order number is added, the error stays a in the nearby value, that the presence of correction factor doesn't give any significant change to the convergence of decomposition process and these errors are comparable to the machine precision, means that the computed error is very small and lies within the range of precision that can be represented by the numerical system used by the machine. In detail, Hilbert-Schmidt function without the addition of correction order possesses close to  $\sim 1E-15$  of error at low iteration even at high numbers of maximum iteration. While, with correction order  $k = 1$  or  $k = 2$ , the decomposition error remains stable.



**Figure 3.2** Decomposition Error result of  $Q$  gate in SQUANDER for both ADAM and BFGS optimizers. The Y axis is set to only  $1E-11$ . In this range of decomposition errors, only 3 Hilbert-Schmidt norm of cost function which can be included to the criteria as efficient logarithm which can be applied since they generate very small number of decomposition errors.

For BFGS Optimizer, the errors produced are as low as in ADAM Optimizer, which is portrayed in Figure 3.2 the right graph. By introducing the correction order, the decomposition errors persist in the same range of value although the maximum iteration numbers grow. The trend obtained is similar to ADAM optimizer, the error reached is in the same level around  $\sim 1E-15$ . The presence of correction order is the same less essential, the Hilbert-Schmidt norm still possesses the constant value of error. Thus, looking from the given error values, the Hilbert-Schmidt cost function generates the error value close to each other even with BFGS optimizer and it can be concluded even with BFGS optimizer, that the Hilbert-Schmidt cost function error is close to machine precision.

As it is described in 2.2, the next observation is to investigate the execution time and decomposition error for 3 Hilbert-Schmidt cost functions, which seems to be a good candidate in 2 optimizers towards varied tolerance,  $10^{-12}$ ,  $10^{-10}$ ,  $10^{-8}$ ,  $10^{-6}$ , and  $10^{-4}$ . Figure 3.3 exhibits the decomposition time

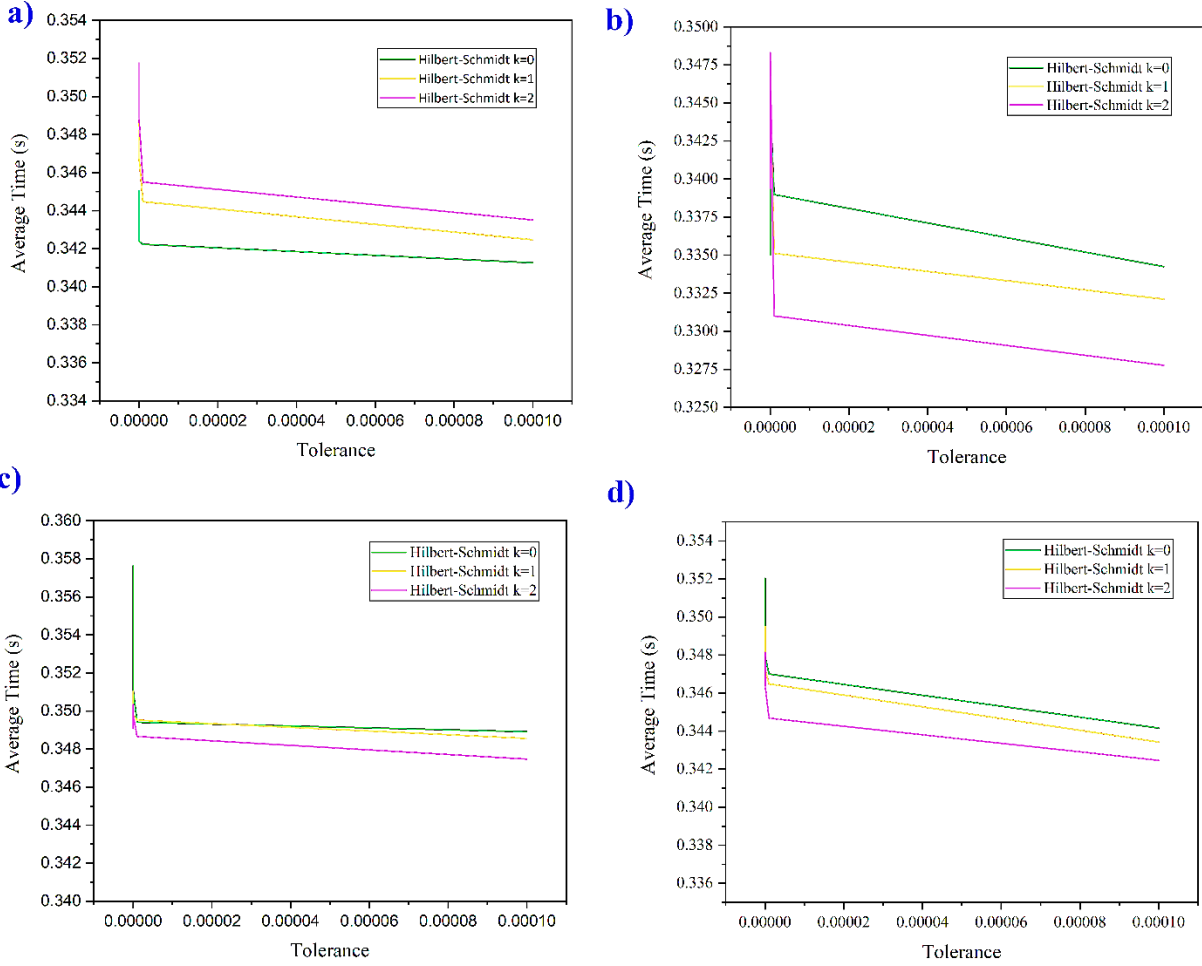
in both optimizers with 2 different maximum iteration numbers, 100 and 5000. As can be observed, Hilbert-Schmidt norm of cost functions requires short time to execute the decomposition in all range of given tolerances. At small tolerance number, the decomposition time in both optimizers is almost close to each other even though when the tolerance number is set in high number. One can be noticed that in ADAM optimizer, when the maximum iteration number is adjusted to 100 (Figure 3.3 a), the decomposition time can be achieved still around 0.35 seconds at small tolerance number,  $10^{-12}$ , even when the correction order is counted ( $k = 1$ ) and ( $k = 2$ ). At high tolerance number, it still reaches indistinguishable result around 0.34 seconds when  $k$  equals to 0, simultaneously with the correction order, ( $k = 1$ ) and ( $k = 2$ ) the required time are similar. As well as, this trend is not completely diverged when the maximum iteration is set to higher number, for instance at 5000 as depicted in Figure 3.3 b. The correction order effect is almost negligible, proved by the similar completion time whether at low or higher tolerance number. When the tolerance limit is set to  $10^{-12}$ , around 0.35 seconds of completion is achieved when  $k = 0$ , similarly at higher correction order,  $k = 1$ , and  $k = 2$ , indifferent decomposition time are recorded. At maximum iteration equals to 5000, the decomposition time required is also around 0.35 seconds for Hilbert-Schmidt cost function ( $k = 0$ ), alike when the correction orders are introduced, ( $k = 1$ ) and ( $k = 2$ ), the execution time needed is analogous.

In BFGS optimizer, the existence of correction factor is still trivial, which is proved by the completion time produced are still in the same range shown by ADAM optimizer. In Figure 3.3 b., despite the fact that there is a gap between the three curves, however the difference between these three curves is quite small, in the range of  $10^{-3}$ . At this low maximum iteration number (max. iteration = 100), Hilbert-Schmidt function can complete the decomposition under 0.34 seconds with the absence of correction order, when it is observed at low tolerance number,  $10^{-12}$ . Likewise, when the correction order is added to  $k = 1$  and  $k = 2$ . Although at high tolerance number, the

completion time is reached in the same range value. For instance, when it is set to higher number, ( $k = 2$ ), the decomposition can be completed under 0.33 seconds, and when  $k = 1$  also  $k = 0$ . By increasing the maximum iteration number to the higher value, the decomposition still remains stable, as shown in Figure 3.3 d. Starting at low tolerance number,  $10^{-12}$ , the decomposition is undergone after 0.35 seconds when the correction order is not introduced. Uniformly, when it is added, it reaches a similar execution time for the first correction and for the second correction. In higher tolerance number, the tendency is the same, while the highest correction order,  $k = 2$ , possesses decomposition time at 0.34 seconds, comparably when it is run for  $k = 1$  and  $k = 0$ .

Decomposition errors are also measured for various tolerance number's range and plotted in Figure 3.4. In ADAM optimizer, correction order doesn't give some refinement effect to the convergence achieved during the decomposition, showing the decomposition errors are still close to each other. When tolerance is set to  $10^{-12}$ , cost function without the presence of correction order still has the decomposition error  $\sim 1E-15$ , even at high tolerance number, it exhibits the similar value of decomposition (showed by the star red shape). Furthermore, when we observe at higher maximum iteration number (showed by the blue colour), the tolerance number equals to  $10^{-12}$ , the Hilbert-Schmidt cost function with  $k = 0$  still has identical decomposition error. Nonetheless, the trend remains in the same range of  $10^{-15}$  even with the addition of correction order.

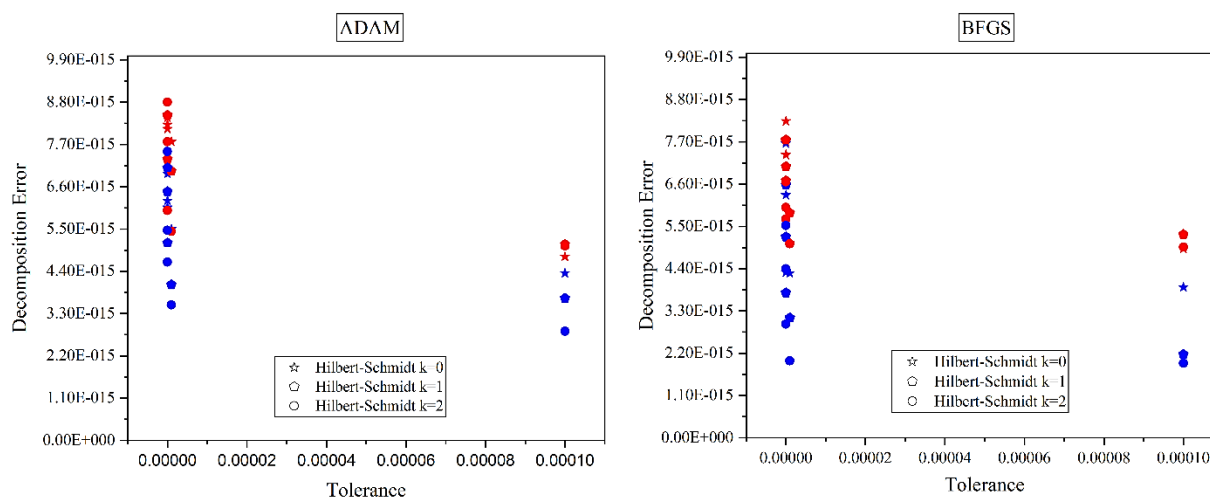
Like in ADAM optimizers, in spite of added correction order, the decomposition error immovably catches to  $\sim 1E-15$  for  $k = 1$  and for  $k = 2$ , or even without the correction order at tolerance equals to  $10^{-12}$  and maximum iteration number is 100. Even at higher value, the introduction of correction order doesn't give any significant alteration to the decomposition error. Despite the fact when the tolerance is set to  $10^{-4}$  and the maximum iteration number equals to 5000.



**Figure 3.3** Average decomposition time for 3 different Hilbert-Schmidt cost functions. a) with 100 iterations in ADAM Optimizer, b) with 100 iterations in BFGS Optimizer, c) with 5000 iterations in ADAM Optimizers, and d) with 5000 iterations in BFGS Optimizer.

As it can be seen from all the observation result, Hilbert-Schmidt norm proves an efficient and effective to be applied in gate synthesis process. Hilbert-Schmidt norm possesses low decomposition error and the execution time required is quite short. This can also be caused because when approximating a matrix using a lower rank matrix, the Hilbert-Schmidt norm encourages the approximation to preserve more of the structure of the original unitary matrix, whereas the Frobenius norm may allow more of this structure to be lost in the approximation [30]. Besides, Hilbert-Schmidt

norm provides a more complete measure of the size of a matrix than the Frobenius norm and this can be particularly important in quantum system, where the wave function can be high-dimensional and complex [31].



**Figure 3.4** Decomposition error for 3 different Hilbert-Schmidt cost functions. Red colour indicates the result for 100 iterations, blue colour indicates the result for 5000 iterations.

It can be seen from the obtained result that Hilbert-Schmidt norm cost function has a closeness to machine precision when it is applied for quantum gate synthesis of Q block gate as proposed in [15], unfortunately the introduction of the correction order as proposed in [28] doesn't give a significant impact to the Hilbert-Schmidt cost function whether it is applied in ADAM or BFGS optimizer. However, it is still important to note that the Hilbert-Schmidt cost function error being close to machine precision is not guaranteed in all cases. Depending on the specific problem and implementation details, the error may vary.



## 4 | Summary

The main goal for this work is to observe and find the most efficient and effective cost function to be applied between 6 different cost functions which in 2 commonly used optimizers, ADAM and BFGS for quantum gate synthesis of molecular simulations case. The Q block quantum circuit was prepared as proposed in [15] and the unitary matrix of this circuit was decomposed in SQUANDER using Adaptive Circuit Compression Method along line topology, because this method can follow quantum device's constraint. These 6 different cost functions consist of Frobenius norm without correction factor ( $k = 0$ ) and the other form with correction factor ( $k = 1$ ) and ( $k = 2$ ). The rest are Hilbert-Schmidt norm without correction factor ( $k = 0$ ), and with correction factor ( $k = 1$ ) and ( $k = 2$ ). They were then executed in ADAM and BFGS optimizers with different number of iterations. Further investigation was also conducted into various number of tolerance limits.

Hilbert-Schmidt norm revealed good stability during the decomposition process of Q block quantum circuit by showing the steady execution time and decomposition error whether the correction factor is introduced or not. Even at higher iteration number, Hilbert-Schmidt norm of cost functions exhibited short execution and excellent decomposition error result. This property still can be also observed at small number of tolerance limit. Thus, Hilbert-Schmidt norm of cost function can be comparable to machine precision when it is applied for quantum gate synthesis of Q-block gate.

## | Reference

- [1] J. Stolze and D. Suter, Quantum Computing A Short Course from Theory to Experiment, Weinheim: WILEY-VCH GmbH, 2004.
- [2] M. Hirvensalo, Quantum Computing, Heidelberg: Springer-Verlag Berlin, 2001.
- [3] T. Jawaid, "Quantum Computing and the Future Internet," *ISSA Journal*, pp. 19-24, 2020.
- [4] E. a. M. National Academies of Sciences, Quantum Computing: Progress and Prospects, Washington D.C.: The National Academies Press, 2019.
- [5] S. Imre and F. Balázs, Quantum Computing and Communications: An Engineering Approach, Chichester: John Wiley & Sons Ltd, 2005.
- [6] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge: Cambridge University Press, 2010.
- [7] P. I. Hagouel and I. G. Karafyllidis, "Quantum Computers: Registers, Gates and Algorithms," in *International Conference on Microelectronics* , Nis, 2012.
- [8] V. Kasirajan, Fundamentals of Quantum Computing, Cham: Springer Nature Switzerland AG, 2021.
- [9] D. F. V. James, P. G. Kwiat, W. J. Munro and A. G. White, "On the Measurement of Qubits," *Phys Rev A* , vol. 64, no. 5, 2001.
- [10] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen and M. Steffen, "Simple All-Microwave

Entangling Gate for Fixed-Frequency Superconducting Qubits," *Physical Review Letters* 107, 080502, pp. 080502-1-5, 2011.

[11] D. A. Fedorov, B. Peng, N. Govind and Y. Alexeev, "VQE method: a short survey and recent developments," *Materials Theory* 6:2, pp. 1-21, 2022.

[12] A. G. Taube and R. J. Bartlett, "New Perspectives on Unitary Coupled-Cluster Theory," *International Journal of Quantum Chemistry*, Vol 106, p. 3393–3401, 2006.

[13] R. Babbush, D. W. Berry, J. R. McClean and H. Neven, "Quantum simulation of chemistry with sublinear scaling in basis size," *npj Quantum Information* volume 5, 92, 2019.

[14] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications* volume 9, 4812, 2018.

[15] G.-L. R. Anselmeti, D. Wierichs, C. Gogolin and R. M. Parrish, "Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems," *New Journal of Physics* , vol. 23, no. 113010, 2021.

[16] D. DiVincenzo, "Two-bit gates are universal for quantum computation," *Physical Review A*, 15:1015, 1995.

[17] A. Barenco, C. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin and H. Weinfurter, "Elementary gates for quantum computation," *Physical Review A* vol. 52, p. 3457–3467, 1995.

[18] R. R. Tucci, "A rudimentary quantum compiler(2cnd ed.)," *arXiv preprint quantph/9902062*, 1999.

- [19] V. V. Shende, S. S. Bullock and I. L. Markov, "Synthesis of quantum logic circuits," in *Proceedings of the 2005 Conference on Asia South Pacific Design Automation, ASP-DAC 2005*, Shanghai, China, 2005.
- [20] M. Mottonen and J. J. Vartiainen, "Decompositions of general quantum gates," *arXiv preprint*, p. 2005.
- [21] M. Amy, P. Azimzadeh and M. Mosca, "On the controlled-NOT complexity of controlled-NOT-phase circuits," *Quantum Science and Technology*, 4(1), 2018.
- [22] V. Shende, I. L. Markov and S. S. Bullock, "Minimal universal two-qubit controlled-notbased circuits," *Phys. Rev. A*, vol. 69, p. 062321, 2004.
- [23] S. Lee, S.-J. Lee, T. Kim and J.-S. Lee, "The cost of quantum gate primitives," *Journal of Multiple-valued Logic and Soft Computing*, vol. 12, no. 5, pp. 561-573, 2006.
- [24] L. Madden and A. Simonetto, "Best Approximate Quantum Compiling Problems," *ACM Transactions on Quantum Computing*, vol. 3, no. 2, pp. 1-29, 2021.
- [25] P. Rakyta and Z. Zimboras, "Approaching the theoretical limit in quantum gate decomposition," *Quantum* 6, 710, 2022.
- [26] P. Rakyta and Z. Zimborás, "Efficient quantum gate decomposition via adaptive circuit compression," *arXiv:2203.04426*, 2022.
- [27] " Sequential quantum gate decomposer.," 2021. [Online]. Available: <https://github.com/rakytap/sequential-quantum-gate-decomposer>.

- [28] N. F. Robertson, A. Akhriev, J. Vala and S. Zhuk, "Escaping barren plateaus in approximate quantum compiling," *10.48550/arXiv.2210.09191*, 2022.
- [29] F. Arute and et.al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505-510, 2019.
- [30] F. Pes, "On the computation of the minimal-norm solution of linear and nonlinear problems," *Universita Degli Studi Di Cagliari, Cagliari*, 2021.
- [31] M.-L. Hu, X. Hu, J. Wang, Y. Peng, Y.-R. Zhang and H. Fan, "Quantum coherence and geometric quantum discord," *Physics Reports*, Vols. 762-764, pp. 1-100, 2018.
- [32] M. Cerezo, A. Sone, T. Volkoff, L. Cincio and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature communications*, vol. 12, no. 1, pp. 1-12, 2021.

## | Appendix 1

Adaptive moment (ADAM) optimizer is an extensively applied optimization strategy which was adapted from deep learning neural networks. The ADAM optimizers generate efficient stochastic optimization with only first-order gradient and employ adaptive moment estimation. The most highlighted feature of ADAM optimizer is the momentum, which is influenced by the ball's momentum with friction. ADAM overcomes some local minima by reusing past parameter upgrades with an exponential decay towards the past.

ADAM algorithm can be briefly summarized as follows: First hyperparameters  $\{\eta, \beta_1, \beta_2, \varepsilon\}$ , momentum  $m^{(0)} = 0$ , average square gradient  $v^{(0)} = 0$  are set.  $\beta_1$  are usually known as moving average parameter for past gradients and  $\beta_2$  is moving average parameter for past squared gradients.

At t-th step, the momentum and the cumulated squared gradient are updated as:

$$m^{(t)} = \frac{\beta_1 - \beta_1^t}{1 - \beta_1^t} m^{(t-1)} + \frac{1 - \beta_1^t}{1 - \beta_1^t} \nabla (E(\theta^{(t)})), \quad (30)$$

$$v^{(t)} = \frac{\beta_2 - \beta_2^t}{1 - \beta_2^t} v^{(t-1)} + \frac{1 - \beta_2^t}{1 - \beta_2^t} \nabla (E(\theta^{(t)}))^{\odot 2}, \quad (31)$$

where  $()^{\odot 2}$  represents the elementwise square of a function. The parameter update is then formulated into update quantities through:

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta}{\sqrt{v^{(t)}} + \varepsilon} m^{(t)}. \quad (32)$$

The square root of  $v^{(t)}$  taken elementwise.  $\varepsilon$  denotes as regularizer, which prevents unreasonably large updates in flat regions and dividing by zero at initialization. Due to the utilization of moving average for both momentum and magnitude estimations, ADAM algorithm can be applied to conditions with noisy gradients.

Second-order optimization utilizes the second-order derivative of the objective function to decide the descent direction. In this type of optimizers, Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is focused. A global line search is performed in the direction of the gradient transformed by the Hessian inverse, using only first-order resources. Even conventional problems with large parameter spaces are difficult to calculate precisely with the Hessian. The optimization is commenced by setting of starting point  $\theta^{(0)}$  and first approximation of Hessian of cost function  $H^{(0)}$  to the identity. The gradient of each step brings the direction which is formulated as:

$$n^{(t)} = H^{(t)-1} \nabla E(\theta^{(t)}), \quad (33)$$

and implement a line search on  $\{\theta^{(t)} + \eta n^{(t)} | \eta \in \mathbb{R}\}$  which This gives the best update in that direction and can optionally be limited to a bounded parameter space. With the new point in parameter space,  $\theta^{(t+1)}$ , the gradient change is computed as:

$$D^{(t)} = \nabla E(\theta^{(t+1)}) - \nabla E(\theta^{(t)}), \quad (34)$$

and this gradient change is utilized to update the approximate Hessian as:

$$H^{(t+1)} = H^{(t)} + \frac{D^{(t)} D^{(t)T}}{\eta^{(t)} D^{(t)T} n^{(t)}} - \frac{H^{(t)} n^{(t)} n^{(t)T} H^{(t)}}{n^{(t)T} H^{(t)} n^{(t)}}. \quad (35)$$

Although analytical methods can overcome the calculation of Hessian in variational ansatz, it is still strenuous to estimate the full Hessian. Therefore, BFGS method is usually employed as early stage of conventional algorithm or base comparison for the newest developed optimizer.