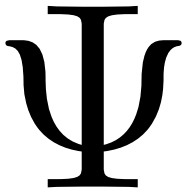
Materials for the simulation of quantum system behavior on quantum computer

Hyunseong Kim

qwqwhsnote@gm.gist.ac.kr



Contents

I Simulation of Quantum system **1**

1 Formalization and Evolution **3**
- 1.1 How does quantum computer simulate system? 3
- 1.2 Time evolution operator of the system 4
  - 1.2.1 Exponential Map 5
  - 1.2.2 Problem of commutation 8
  - 1.2.3 Suzuki Trotter expansion 10
  - 1.2.4 Terminology Note 12
  - 1.2.5 Further methods 12
- 1.3 Formalization of time evolution 13
  - 1.3.1 Type of systems 13
  - 1.3.2 Spin operator representation of General system 13
- 1.4 Circuit representation of the evolution operator 14
  - 1.4.1 Pauli-matrices and decomposition 14
  - 1.4.2 Evolution operator on single qubit system 15
  - 1.4.3 N-qubit system 17
  - 1.4.4 Implementation with Phase operator 18
- 1.5 Additional Note 19
  - 1.5.1 Clifford Group 19
  - 1.5.2 Pauli-Frame 19

2 Optimization of evolution circuit **21**
- 2.1 Introduction 21
- 2.2 Mutally commuting groups 21
- 2.3 Hamiltonian grouping problem 22
  - 2.3.1 Commutator of n-folded string 23
  - 2.3.2 Partition construction 24
- 2.4 Axis transformation weight 25
  - 2.4.1 Rough implementation of single qubit transformation 26
  - 2.4.2 Pauli-Frame metric 27
- 2.5 Krylov transformation 28
  - 2.5.1 Other techniques 28

2.6 VQE method and evolution 28

II Practical Examples 1

3 Simple quantum systems 3
3.1 Modeling of 1dim Particle 3
3.1.1 Modeling a Hamiltonian 5
3.1.2 Slite experiment 5
3.1.3 6

4 Adiabatic process on gate model 7
4.1 Formulation for Adiabatic computing 7
4.2 Implementation 8
4.2.1 QUBO 8
4.2.2 Ising model 8
4.3 Pennylane implmentation 8
4.3.1 Graph partitioning problem 8

5 Imaginary Time evolution 9
5.1 Imaginary time 9
5.2 Non-unitary on unitary 10
5.2.1 Embeding in unitary 10
5.2.2 VQE approximation 10
5.3 Spectrum Search 10

6 Variational method 11
6.1 Introduction 11
6.1.1 Variatiaonal Principle 11
6.1.2 Circuit as ansatz 11
6.1.3 Universality of Circuit 12
6.2 12
6.3 Ansatzs 12
6.3.1 Hardware Efficient 12
6.3.2 Evolution Based 12
6.4 Application 12
6.4.1 QUBO 12
6.4.2 Evolution 12
6.5 Implementation 12
6.5.1 Pennylane 12
6.5.2 Qiskit 12

7 Material Simulation 15
7.1 Hardness of the simulation 15

8 Tunneling Phenomenon **17**
8.1 Introduction to tunneling 17
8.1.1 alpha-decay 17
8.2 Tunneling time problem 17
8.3 Criticisims on tunneling time 18
8.3.1 Classical approach 18
8.3.2 Measurement- Experimental 18
8.3.3 Hartman effect and Superluminal velocity 18
8.4 Simulation and Measurement on Quantum circuit 18
8.4.1 Page Wootter Mechanics 19

III Further Topics **1**
8.5 Simulation Techniques 3
8.5.1 VQE evolution 3
8.5.2 Linearization 3
8.5.3 Quantum Walks 3
8.5.4 Qubitaization 3
8.5.5 Fractional Query 3
8.5.6 Taylorization 3

IV Appendix **1**

A Hermit and Unitary matrix **3**
A.1 Unitary matrix 3
A.1.1 Properties of unitary matrix 3
A.2 Hermit matrix 4
A.2.1 Properties of Hermite matrix 4
A.3 Additional properties 5

B Notation in Quantum circuit **7**
B.1 State 7

C Fourier Trasnformation **9**
C.1 DFT and FFT 9
C.2 QFT 9
C.3 QFT implementation on Circuit 10

D Fast Pauli Polynomial manipulation **11**
D.1 Decomposition 11
D.1.1 Tensorized method 11
D.1.2 Term-by-Term methods 11
D.2 Composition 12
D.2.1 PauliComposer 12
D.2.2 Inverse of the Tensorized method 12

E Miscellaneous Mathematics **15**
E.1 Tensor Product 15
E.2 Tensor product representation of quantum circuit 16
E.3 Lie-algebra 16

F NP-problems **17**
F.1 Traveling Purchaser problem 17
F.1.1 Traveling Saleman problem 17
F.2 Max-clique Problem 17

Preface

This document is a cheet-sheet and proof-notes about quantum simulation on quantum computer, especially, in gate model. If you know basic postulations and concepts of quantum computers and you want to implement the simulation of the given quantum system, it might be helpful.

Mainly focusing on Product-formula method and its optimization topics. They are basic materials for any kinds of quantum simulation and my Bachelor thesis subject. The formalization part is a basic materials for the product formula implementation minimizing errors on the formulation steps. The mathematical background of the evolution circuit and their representations were dicsussed. Optimization part is about the optimize the evolution circuit by using various techniques. In the practical part, we review the many practical techniques using quantum computer and their simple implementations. The modeling of the single-particle system, adiabatic process implementation, VQE introduction are dicussed. In the further topics part, we reviewed the advanced topics. Using 1 dim particle simulator, we dicussed about tunneling phanomenon and their simulation in quantum computer. Material modeling and physcial properties simulation in many-body physics, binary optimization in quantum computer briefly reviewed.

Written programs and algorithm implemnetations are also provided with refering repositories. In appendix, step by step example of simulation and measurement using some popular quantum frameworks;Qiskit, Pennylane, D-Wave et cetra.

Most contents are related with author's studies and research in the course of Quantum Graduate School internship established by 8 university. Thanks for Dr. Kim(GIST), Dr. Yee(GIST) and Dr. Choi(Korea).

- 2023 Summer Special Internship, held by Korea university
- Bachelor research internship on Quantum Field and Gravitation Lab of GIST
- Bachelor research internship on Computational Material Lab of GIST

Abbrevations

- QC: Quantum computation
- CC: Classcial computation

Part I

Simulation of Quantum system

CHAPTER

1

Formalization and Evolution

Section 1.1

How does quantum computer simulate system?

A quantum system basically lies in continuous Hilbert space of normalized complex functions, *wave function* or *quantum field*. However, common universal quantum computing model; *circuit model*, is not a continuous quantum model. It is a discrete computation model. There are some continuous computation model in quantum computer, such as *adiabatic model*. However, in this document we will focus on the circuit model and the adiabatic model would be treated in separated application chapter. Therefore, if we want to simulate a given quantum system, first thing to do is a discrete formalization to run on the gate model system.

The discretization is not a new concept in quantum computation. We can simulate various quantum systems in classic computation model already, and the implementation of the simulation on the system requires appropriate discretization techniques. The common notation and techniques of the quantum computation were adopted from classic computation technique. The differences are efficiency of computation and existence of the model. Many quantum systems do not have an any approximation model. For example, in condensed matter physics, Ising, and Hubbard Hamiltonian have been frequently used to describe the spin system of solid material. Even we just increase the dimension or lattice site, the problems become too complicated to solve or to compute their behavior[1]. However, quantum computer could simulate their behavior efficiently, comparing to the computational techniques in classic computers. In addition, even the most complicated system was given, at least, we can try the basic techniques by simulating time-evolution.

Then, what does the discrete model affect the computation and the modeling of the problem?. Considering a differential equation. All we do in quantum mechanics is getting solutions of the given differential equations. We want to approximate the solution, $f(x)$, of the equation, $D(x, f, f', \dots) = 0$, with given initial or boundary conditions, $(x_0, f(x_0))$. We first discrete the region of consideration with Δx and approximate the next points start from the initial value to the target value. This is called by *Euler method*.

$$(x_0, f(x_0)) \tag{1.1.1}$$

$$x_i = x_{i-1} + \Delta x \tag{1.1.2}$$

$$f(x_i) = f(x_{i-1}) + \sum \lambda_n f^{(n)}(x_i)(\Delta x)^n \tag{1.1.3}$$

[1] For Ising, only 1 and 2 dimension cases are exactly solvable, and the further dimensions have no solution.

There are several techniques to update the point value in each step, but the details are not a consideration in here, see numerical analysis textbook for the details[1]. The gate model simulation of quantum system is exactly same with the above discrete approximation. We update the intermediate state until we reach the state we want to observe. When we want to simulate the evolution of the system, first thing to do is preparing the initial system configuration, and we can obtain the target state by applying several evolution operators with evolution time, Δt.

These are main focus of the material. The quantum computer provides for us to simulate the complex systems in many areas not only for physics, but also for many engineering, especially computer science. We already have general frameworks to manipulate the operation, preparation, and dynamics of the quantum states. However, the exact process to simulate the given quantum state is still remained for us.

1. How to formulate *time-evolution* in quantum computing language?
2. How to implement the evolution process?
3. What error could be happened during the operation?
4. How to model a specific system to simulate with quantum computer?

There are many techniques and requirements for researchers and engineers to know. Thus, in here, we will overlook those techniques from solid mathematical backgrounds to programming implementation.

Section 1.2

Time evolution operator of the system

We start from the simplest situation; a time independent Schrödinger equation,

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle$$

In Schrödinger picture, where the operators are time dependent, We can express a solution of $|\psi(t)\rangle$ with unitary operator, U,

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle \tag{1.2.1}$$

What we have at the initial stage were an initial state and the given Hamiltonain. From the information, we have to get a $U(t)$ operator for simulate the dynamics of the given system. The solution of Eq (1.2.1) is achieved easily if the H has no time dependency[2].

$$U(t) = \exp\left(-i\frac{1}{\hbar}(H)t\right) \tag{1.2.2}$$

A more precise notation is $U(t_i, t_f)$ which indicates the initial time and final time. With the notation, the $U(t)$ is $U(0, t)$. We call it as a time-propagator[3] Here is why the gate model is called a discrete computation model. It is because we cannot directly implement the such general time-propagator per each system. Only thing we can manipulate is just an approximation of the propagator with basic block gates. Fortunately,

[1] Cheney, E Ward, or Burden et al are famous.

[2] In general, the solution is never gone like the Eq (1.2.1). We will see it in a further section.

[3] In mathematics, such propagator is a green function of solution of differential or integral equations. $G(s_n, t_n|s_0, t_0) = U(t_n, t_0)$

the universal gate set to represent approximate all unitary operator is well investigated. See *Solovay-Kitaev theorem.*

Definition 1 **Unitary approximation** For a given unitary gate, $U(t)$, approximation of $U(t)$ is a constructing an $U'(t)$ gate which consist of the universal gate set of the quantum machine, such that minimize the next,

$$\max\left(|U(t) - U'(t)|\right) \tag{1.2.3}$$

Figure 1.1. Sophus Lie

Then, what we need? What does the operator mean inside of exponential? It is called exponential mapping of operator, and it plays a key concept of time-evolution in quantum mechanics. The major properties and definiton were first investigated by *Sophus Lie* with his research on continuous group and differential geometry. It has tremendous interesting properties and applications in both physics and mathematics. However, in here, we only look up the operator map for evolution implementation in finite matrix group.

Subsection 1.2.1

Exponential Map

Definition 2 **Operator exponential**

$$\exp\left(\hat{X}\right) = \sum_{k=0} \frac{1}{k!}\hat{X}^k \tag{1.2.4}$$

The map is converges $\forall X \in \mathbf{M}_n(\mathbb{C})$.

Theorem 1 **Properties of Exponential Map**

- e^X is a continuous function.
- $e^0 = I$
- $(e^X)^* = e^{X^*}$
- e^X is always invertible, and $(e^X)^{-1} = e^{-X}$.
- $e^{(a+b)X} = e^{aX}e^{bX}$.
- If $[X, Y] = 0$, $e^{X+Y} = e^X e^Y = e^Y e^X$.
- $\forall M \in \mathrm{GL}(n;\mathbb{C}), e^{MXM^{-1}} = Me^X M^{-1}$.

If $\hat{A}$ was a digonal matrix, then the next relationship hold.

$$A = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}, e^A = \begin{bmatrix} e^{\lambda_1} & 0 & \dots & 0 \\ 0 & e^{\lambda_2} & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n} \end{bmatrix}$$

In general not only for the digonal matrix, the exponential map preserves the eigenvectors of the original matrix.

Exercise 1 For an eigen vector, $\mathbf{v}$, of the given matrix, X, with eigenvalue, λ,

$$X\mathbf{v} = \lambda\mathbf{v} \tag{1.2.5}$$

Show that

$$e^X \mathbf{v} = e^\lambda \mathbf{v}$$

In general case, we decompose the Hamiltonian as sum of local terms, and the time as sum of many intermediate steps. These slicing allow us to analysis the dynamics of the system evolution in specific local region or local time. First, the time slicing is not different with usual exponential function. The time-propagator from time t_0 to t_n can be decomposed to n intermediate step propagators,

$$\begin{aligned} U(t_n, t_0) &= U(t_0 + \Delta t, t_0) U(t_0 + 2\Delta t, t_0 + \Delta t) \cdots U(t_n, t_n - \Delta t) \\ &= \Pi_{i=0}^{n-1} U(t_{i+1}, t_i) \\ &= \Pi_{i=0}^{n-1} \exp\left(-i \Delta t H\right) \\ &= \exp\left(-i(t_n - t_0) H\right) \end{aligned}$$

However, decomposition with local Hamiltonian, $H = \sum_k H_k$ yields a problem. The non-commuting terms in $\sum_k H_k$ makes an error in the expotential map.

$$\begin{aligned} [H_i, H_k] &\neq 0 \\ e^{H_i + H_k} \neq e^{H_i} e^{H_k} \text{ or } e^{H_k} e^{H_i} \end{aligned}$$

Major problem arises in here. Since, we cannot directly implement the given evolution gate, but just applying sequential local term does not yield the correct gate. The problem would be investigated in the following subsection.

Pauli matrix and exponential

Before, we move to the non-commuting problem. Let us see basic properties of Pauli matrices as a basis set of matrix space. In the above paragraph, we read the Hamiltonian could be decomposed into several local Hamiltonians. Mostly, we decompose them with Pauli matrices. Since, Pauli matrices have good properties to decompose the general Hamiltonian.

Definition 3 **Pauli matrices**

$$\sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{1.2.6}$$

Usually, $\sigma_X = \sigma_1, \sigma_Y = \sigma_2, \sigma_Z = \sigma_3$. By the context, the notation could be different. Sometime just only use X, Y, Z.

Exercise 2 Show that,

$$[\sigma_j, \sigma_k] = 2i\epsilon_{jkl}\sigma_l$$

and

$$\sigma_j \cdot \sigma_k = \delta_{jk} I + i\epsilon_{jkl}\sigma_l$$

First, it forms a complete orthonormal basis in matrix space, with Hilbert-Schmidt inner product.

Definition 4 **Hilbert-Schmidt inner product** For two given matrices, $A, B \in \mathbf{M}(\mathcal{C})$, Hilbert-Schmidt inner product of two matrices is,

$$\langle A|B\rangle_{HS} := \frac{1}{N}\mathrm{Tr}(A^\dagger B) \tag{1.2.7}$$

The $1/N$ is a normalization factor.

Exercise 3 Prove that the Hilbert-Schmidt inner product satisifes inner product axiom.

- Positive definiteness: $\langle X|X\rangle \geq 0$.
- Linearity: $\langle aX + bY|Z\rangle = a\langle X|Z\rangle + b\langle Y|Z\rangle$.
- Conjugate symmetry: $\langle X|Y\rangle = \overline{\langle Y|X\rangle}$

Exercise 4 Show that any 2-by-2 square matrix can be decomposed into sum of Pauli matrices and it forms an orthonormal basis set of 2-by-2 matrices vector space.

Second, usual quantum computing systems are based on spin system. Therefore, Pauli and their generalized gates are universal for most quantum computing frameworks [4]. Third, the algebra of Pauli matrices is well studied so that the manipulation framework has been well established. Pauli decomposition allow us to deal the given Hamiltonian as an algebraic object.

The best operators to represent the Hamiltonian are unitary **and** simultaneously Hermit operators. If the given operator, $\hat{A}$ is unitary and hermit then, the next relationship is hold,

$$\hat{A} = \exp\Big(i\frac{\pi}{2}\hat{A}\Big). \tag{1.2.8}$$

Pauli-matrices, $\sigma_X, \sigma_Y, \sigma_Z$, are typical example of the such matrix. It is unitary, Hermit, and complete orthonormal basis set. In $2^n > 2$, Hilbert space, their tensor product also hold the properties. Therefore, any given Hamiltonian of $2^n \times 2^n$ dimension matrix can be expressed with n-length Pauli-string, P^n

$$\mathcal{H} = \sum_i^l \mathcal{H}_i = \sum_j^{2^n} \lambda_j P_j^n \tag{1.2.9}$$

where, P_j^n is a representing an element of n-folded Pauli matrices. For example, $P_1^3 = XYX$ is a $\sigma_X \otimes \sigma_Y \otimes \sigma_X$.

Somtimes we denote the Hamiltonian as vector notation whose basis is Pauli group.

$$\begin{aligned} H &= H \cdot \hat{\sigma} \\ c_0\sigma_0 + c_1\sigma_1 + c_2\sigma_2 + c_3\sigma_3 & \end{aligned} \tag{1.2.10}$$

Exercise 5 Suppose that basis of vectros is Pauli group.
Find a coefficient formula of inner product of two vector where the inner product of two vector, v, w is defined as

[4] There are some exceptions, but in those systems, we can still use Pauli gates.

$$v = \sum_{i=0}^{3} a_i \hat{\sigma}_i$$
$$w = \sum_{i=0}^{3} b_i \hat{\sigma}_i$$
$$v \cdot w = \sum_{j,k} a_j b_k (\hat{\sigma}_j \cdot \hat{\sigma}_k)$$

and so do on outter product.

$$v \times w = \sum_{j,k} a_j b_k (\hat{\sigma}_j \times \hat{\sigma}_k)$$

One thing you have to notice is that the coefficient λ_j are all real valued. It is a spectrum theorem for Hermit matrix. See Theorem 14 in Appendix A. It is because the complex valued linear combination does not preserve the hermiticity of the summation. You can check yourself with verify the $H_3 = H_1 + iH_2$, whether H_3 satisfies hermiticity or does not when H_1 and H_2 were Hermit matrices.

Subsection 1.2.2

Problem of commutation

Now, we have a practical formula for calculating the evolution operator of the given Hamiltonain. Where was a difficulty to implement the evolution operator on the circuit? The problem arise when the local terms are not commute each other in Eq (1.2.9). It is common if they are operators, matrices, or more widely general group elements. We are familiar with commutation group, Abelian group, however, for operators it is not common. For example, if they are commuting group element, typical example is real or complex number field; $\mathbb{R}, \mathbb{C}$. About the $x, y \in (+, \times, \mathbb{R} \text{ or } \mathbb{C})$,

$$\exp(x)\exp(y) = \exp(y)\exp(x) = \exp(x+y) \tag{1.2.11}$$

holds true. However, such expression does not hold in general case and moreover, $\exp(x)\exp(y) = \exp(z)$ solution may not exist.

For example, in matrix group, $\mathbf{M}_2(\mathbb{C})$,

$$X = \begin{pmatrix} 0 & i\pi \\ i\pi & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{1.2.12}$$
$$\exp(X)\exp(Y) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ 0 & -1 \end{pmatrix} = \exp(Z) \tag{1.2.13}$$

Z satisfying $\exp(Z)$ does not exist [1][4]. The exponential map is defined on whole $\mathrm{GL}(n, \mathbb{C})$, however, it is not surjective.

However, at least, the $\Pi_k \exp(H_k)$ seems appropriate for starting point to be closed to to $\exp(\sum_k H_k)$. Then, how much gap between the two operators? and how can we reduce the gap? What relationship do they have including non-commutting local terms? It is represented with *BCH formula*[2].

[4] *This is a Example 3.41 on page 67.*

Theorem 2 **Baker-Campbell-Hausdorff formula** For the next equation,

$$\exp(X)\exp(Y) = \exp(Z)$$

the solution Z is,

$$Z = X + Y + \Theta([X,Y]). \tag{1.2.14}$$

where, $\Theta([X,Y])$ is an error terms as

$$\Theta([X,Y]) = \frac{1}{2}[X,Y] + \frac{1}{12}[[X,Y],Y-X] + \dots \tag{1.2.15}$$

$$= \sum_n^\infty \frac{1}{n!}\left[\frac{\partial^n}{\partial\lambda^n}\ln\sum_{k=0}^\infty\sum_{j=0}^\infty \frac{\lambda^{k+j}}{k!j!}X^kY^j\right]_{\lambda=0}. \tag{1.2.16}$$

The BCH theorem provides us a reason why there is no solution Z in Eq(1.2.13). It hardly depends on the properties of X, Y operators. The direction is clear now. The $\exp(X)\exp(Y)$ does not exactly same with $\exp(X+Y)$ but, if $O(X,Y)$ term converges to finite value, It would be a good approximation of the original evolution. Suzuki analyzed convergence conditions of the BCH formula[2].

Theorem 3 **Convergence of BCH formula 1**
The BCH formula converges for $(||A|| + ||B||) < \ln 2$

where, $||\cdot||$ is Hilbert-Schmidt norm.

Definition 5 **Hilbert-Schmidt norm**
About the matrix, A over field $\mathbb{F}$, Hilbert-Schmidt norm is

$$||A|| := \sqrt{\sum_{i\in I}||Ae_i||^2}$$

where, $\{e_i\}_{i\in I}$ is an orthnormal basis.

Simply, we can rewrite the above norm as

$$||A|| = \sqrt{\sum_{ij}||a_{ij}||^2} = \sqrt{\mathrm{Tr}(A^*A)} \tag{1.2.17}$$

Theorem 4 **Convergence of BCH formula 2**
For any set of operators A and B in a Banach algebra, the expansion W in

$$\exp(A+B) = \exp(A)\exp(B)\exp(W) \tag{1.2.18}$$

$$W = \sum_{n=2}^\infty W_n \tag{1.2.19}$$

converges, at least, for

$$(||A|| + ||B||) < \frac{1}{2}\ln 2 \tag{1.2.20}$$

Exercise 6 The definition of Banach algebra is

Definition 6 **Banach Algebra** For the complete normed space $(X, |||\dot{|}|)$ and $x, y \in X$, the

$$||xy|| \leq ||x||||y||$$

Show that with Hilbert-Schmidt norm, the matrix space over field $\mathbb{C}$ satisfies Banach Algebra.

Since, **separable** Hilbert space satisfies the Banach Algebra axioms and every finite Hilbert space are separable. Therefore, we can freely use the above convergence formula for finite qubit-system.

This property generate errors, $\Theta(X, Y)$, between the true evolution operator and the product of exponentials of each Pauli-string component.

$$U = \exp(\mathcal{H}_i \Delta t) \neq \Pi \exp(\lambda_i P_i^n \Delta t) \tag{1.2.21}$$

It is not exactly same with the correct unitary operator, however, we can approximate it to the operator by slicing of time unit. It is a *Product Formula*.

Theorem 5 **Product Formula**

$$\exp(X + Y) = \lim_{n \to \infty} (\exp(X/n) \exp(Y/n))^n$$

The error bound is $O(t^2)$ for the evolution time t, thus

$$\exp(tA) \exp(tB) = \exp\big(t(A + B) + O(t^2)\big) \tag{1.2.22}$$

It was originally used in Monte-Carlo simulation for quantum system and was adopted in QC by Llyod[3].

Let us, analysis the above equation, $\exp(X/n) \exp(Y/n)$ guarantees that whatever $||X|| + ||Y||$ value it is, there exists N such that $\forall n > N$, $\frac{1}{n}(||X|| + ||Y||) < \ln(2)$ which satisfies the Theorem 4. In addition, for Z satisfies $\exp(X/n) \exp(Y/n) = \exp(Z)$, $\Pi_{i=1}^n \exp(Z) = \exp(nZ)$ because every operator commute with itself.

Generally, for $\mathcal{H} = \sum \mathcal{H}_i$,

$$\exp(-it\mathcal{H}) = \lim_{n \to \infty} \left(\Pi_{i=1}^n \exp\left(-i \frac{t}{n} \mathcal{H}_i \right) \right)^n \tag{1.2.23}$$

As an approximation, the time, T evolution of the given Hamiltonian, $\mathcal{H}$ can be modeled as

$$\exp(-i\mathcal{H}T) \approx [\exp(-i\mathcal{H}\Delta t)]^{T/(\Delta t)} \tag{1.2.24}$$

where, $n = T/(\Delta t)$ is called Trotter number. This technique is sometimes noted by *Suzuki-Trotter expansion* of 1st kind, ST1. There is a better error bound expansion, we call ST2 and further.

Subsection 1.2.3

Suzuki Trotter expansion

This method is called by *Fractal decomposition*[4].

Theorem 6 **2nd order Suzuki Trotter expansion**

$$\exp\left(t(F+G)\right) \approx [\exp\left(F\frac{\Delta t}{2}\right)\exp(G\Delta t)\exp\left(F\frac{\Delta t}{2}\right)] \tag{1.2.25}$$

With second order ST expansion, the error rate is $O(t^3)$.

Generally, the $H = \sum_k H_k$ term Hamiltonian could be expanded with ST2 expansion formula,

$$e^{-iHt} \approx \left(e^{-iH_1t/2}e^{-iH_2t/2}\cdots e^{-iH_nt/2}\right)\left(e^{-iH_nt/2}\cdots e^{-iH_2t/2}e^{-iH_1t/2}\right) \tag{1.2.26}$$

Be aware the oreder of the local Hamiltonians, in the 1st term, the order is ascending but, in the 2nd term it is descending order. It is a recursive formula of Eq (1.2.25).

For the higher expansion, it is defined with recursively

Theorem 7 **n-th order ST expansion**

$$S_{2n+2}(t;A,B) = S_{2n}(s_{2n}t;A,B)^2 S_{2n}((1-4s_{2n})t;A,B)S_{2n}(s_{2n}t;A,B)^2 \tag{1.2.27}$$

where, $S_2(t;A,B)$ is

$$S_2(t;A,B) = e^{tA/2}e^{tB}e^{tA/2} \tag{1.2.28}$$

and the error is $O(t^{2n+3})$ when

$$s_{2n} = \frac{1}{4 - \sqrt[2n+1]{4}} \tag{1.2.29}$$

However, mostly we only use ST1, Eq (5), or ST2 method. We can make a precise approximation quantum gate following Theorem 7, but the total number of terms are too long considering the computational resource, even in the quantum computer.

The BCH formula could be used for proving **Stone-von Neumann Theorem**, it is a different notation of Canonical Conjugation Relationship(CCR) of two observables. Since, the some modeling Hamiltonians are based on CCR observables, the theorem is worth to note here.

Stone-von Neumann Theorem

Theorem 8 For two Hermit operators satisfying canonical commutation relation, $[X,P] = iI$, and $a, b \in \mathbb{R}$

$$e^{iaX}e^{ibP} = e^{-iab}e^{ibP}e^{iaX} \tag{1.2.30}$$

This is a Weyl's version of CCR, where the original form in QM was,

$$[x,p] = i\hbar \tag{1.2.31}$$

The proof could be started from BCH formula, applying $[X,Y] = i\hbar I$. The error term $O(X,Y)$, in fact, depends on their commutator, $[X,Y]$, so that $O(X,Y) = O([X,Y])$[5]. With ST2, the next relationship hold for $[X,P] = i\hbar I$.

$$\exp(i(aX+bP)) = \exp(iaX/2)\exp(ibP)\exp(iaX/2) \tag{1.2.32}$$

Therefore, if we evolve the two canonical quantity, the error term does not affect to the result, except the global phase.

Subsection 1.2.4
Terminology Note

The Product formula, Eq (5), has been analyzed by many researchers by the various contexts and subjects. By the context, it has many names such as

- *Trotter formula*
- *Lie-Trotter formula*
- *Suzuki-Trotter formula* or *Suzuki-Trotter expansion of 1st kind*
- *Trotter product formula*
- *Llyod's product formula*
- *Product formula*

Trotter analyzed semi-groups of operator on a Banach space[6]. In his 1959 paper, we can see some basic properties of the decomposed operators. The terminologies were fully mathematician's words, but there is no advanced concepts for a modern undergraduate student in physics department.

Masuo Suzuki is a Japan physicist. He has contributed in mathematical and statistical physics. He generalized Trotter's formula for further order errors and precisely analysis the BCH formula, Theorem 2. In addition, he adopted the formula to the many-body Monte-Calro simulation. See [4] for his works, the most of the notation and knowledge in this chapter followed his work. Cohen et al wrote a doubt point of name "Lie" in some references in their paper[7], and assumed that it is a contribution of Lie's exponential product formula of Lie-algebra.

Lloyd introduced the above equation in quantum advantage and general quantum system simulation paper of him[3]. He noted the equation as "time-slicing technique" technique in Mote-Calro simulation on classical system.

Subsection 1.2.5
Further methods

This is not a only way to simulate behavior of the Hamiltonian system. There are many techniques to achieve the desire result of the given quantum system. Some techniques are more efficient than the product formalism but, it is still not only simple and practically useful method even in research area. In addition, some methods use their core architecture as Product formula. For example, Fractional query is a special case of Hamiltonian; sparse Hermite matrix.

Examples of methods to implement the evolution of the quantum system.

- Quantum walk.
- Qubitization.
- Fractional query.
- Taylor Series, or approximation.
- Linear Unitary method.

The brief introduction of the methods is written on Wolf's lecture note[8].

Section 1.3

Formalization of time evolution

Now, we have a basic material for simulation of general *time-independent* Hamiltonian. However, the real quantum world and our interests do not only live in time-independent system. General time-dependent, non-commuting by each time systems are also our intense of study.

Subsection 1.3.1

Type of systems

There are various quantum systems, but 3 large categories exist, by time-dependent and commutation.

- Time independent
- Time dependent-commuting
- Time dependent-non-commuting

The word, *commutation* in this section does not mean the relationship between local terms of the Hamiltonian. The commutation between two different time in time-dependent Hamiltonian.

$$[H(t_1), H(t_2)] \neq 0 \tag{1.3.1}$$

The most general situation in the real world, is time dependent and the Hamiltonian is non-commute in different time. In the case, the evolution operator is not represented with simple exponential term. It is represented with series, *Dyson Series*.

Theorem 9 **Dyson Series**

$$U(t, t_0) = \sum_n (-i\hbar^{-1})^n \int_{t_0}^{t} dt_1 \int_{t_0}^{t_1} dt_2 \cdots \int_{t_0}^{t_{n-1}} dt_n \Pi_{i=1} H(t_i)$$

With *Time ordered* method, we can upload the form to be exponential. The approximation of the time ordered exponential with several independent evolution gate is well-studied by Suzuki[9]. The answer is same with commuting Hamiltonian however, the theoretical background is well studied in his 1993 paper.

$$H(t) = \sum_i A_i(t) \tag{1.3.2}$$

Subsection 1.3.2

Spin operator representation of General system

Now, we have a well-defined controllable spin system. However, many quantum systems do not consist of only spin operators. In some case, naturally described by spin operator, For example, simulating Larmor-precession in uniform magnetic field $\mathbf{B} = B_0 \hat{z}$ is directly expressed by spin operator.

$$\mathcal{H} = g \frac{q}{2m} \mathbf{S} \cdot \mathbf{B} \tag{1.3.3}$$

and Ising model or Fermi-Hubbard models also represented with spin of the system elements,

$$\mathcal{H} = \sum \lambda_i (S_i)_z + \sum h_{ij} (S_i)_z (S_j)_z \tag{1.3.4}$$

However, common cases are represented with system dependence *annihilation* and *creation* operators; a_+, a_-. Such operators could be fermionic or bosonic. In those cases, we need a transformation to run such Hamiltonian on our system.

Operator transformation

We mainly focus on the *spin* system as base line to start the quantum computation. However, many quantum system do not consist of only spin system. Sometimes we only treat **fermion** system, such as electrons in the molecule, condensed matter, or we may want to simulate **boson** system [5]. Those fermion and bosons have their own algebra. In addition, we may want to simulate fermionic, boson and spin effect at once. How can we achieve those case in quantum system? The answer is mimicking the fermion and bosonic operators as spin operators, *operator transformation*.

If we can express the system with spin operators, we could directly implement their dynamics without any modification. Luckily, this topic has been an intense of study in simulation, and some common methods has already investigated. There are popular transformations for each fermion and boson operators.

Table 1.1. Transformation of each system to spin system

Fermion	Boson
Jordan-Wigner	Holstein-Primakoff
Bravyi-Kitaev[10]	-
Partiy	-

Details of the each transformation requires long chapter and section. We move back the content to further section in practical modeling of each specific system.

Section 1.4

Circuit representation of the evolution operator

This section is about a practical implementation of the evolution operator. From the above sections, we overlook some basic decomposition of evolution operators of general quantum systems.

Subsection 1.4.1

Pauli-matrices and decomposition

The general convention of Hamiltonian evolution operator is an inner product to Pauli-vector, $\hat{\sigma}$.

$$\mathcal{H} = \mathcal{H} \cdot \hat{\sigma} \tag{1.4.1}$$

. In 2-dimension system, it becomes $[\hat{\sigma_X}, \hat{\sigma_Y}, \hat{\sigma_Z}]^T$.

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.4.2}$$

Precisely, such convention is a Pauli-polynomial as shown in Eq (**??**). Now, how did the general evolution operators implemented on quantum circuit? It becomes rotations

[5] Of course even in those system, spin may have dominant affection to the behaviors. The key point is that what physical quantity we choose in the simulation.

on specific axes of the given Hilbert space. With the Pauli-polynomial representation of the given Hamiltonian, the next relationship is hold true, see proof in Appendix **??**.

$$\exp(-i\theta\mathcal{H}\cdot\hat{\sigma}) = \cos(\theta||\mathcal{H}||)\hat{I} - i\sin(\theta||\mathcal{H}||)\hat{\mathcal{H}}\cdot\hat{\sigma} \tag{1.4.3}$$

For general 2^n-dimension system, it is enough to show that the construction rule of every Pauli-n strings. We first look at single qubit cases and expand it to the general n-qubit system.

Subsection 1.4.2

Evolution operator on single qubit system

The Pauli-string of 1-qubit system are just $\{X, Y, Z\}$.

$$\mathcal{H} = \lambda_X X + \lambda_Y Y + \lambda_Z Z \tag{1.4.4}$$

Then, an evolution operator of total Hamiltonian consist of 3 rotation gates.

1. $\exp(-it_X X) = \cos(t_X)\hat{I} - i\sin(t_X)\hat{X} = RX(2t_X)$
2. $\exp(-it_Y Y) = \cos(t_Y)\hat{I} - i\sin(t_Y)\hat{Y} = RY(2t_Y)$
3. $\exp(-it_Z Z) = \cos(t_Z)\hat{I} - i\sin(t_Z)\hat{Z} = RZ(2t_Z)$

Be aware that they are not commute each other. You must use Product or ST2 formula when you deal with them simultaneously.

We have 3 rotation gates by the axis, but we can generate the other two rotation with one rotation gate and transformation gates. See an eigen decomposition form of each matrix.

$$A = QDQ^\dagger \tag{1.4.5}$$

where, $Q = [e_1, e_{-1}]$, e_λ is an eigenvector corresponding to eigenvalue, λ. Since, $\sigma_i, i \in [X, Y, Z]$ have same eigenvalues, $1, -1$, $D = Z$. The eigenvector of X, Y are

- X: $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$
- Y: $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}, \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$

then,

$$Q_X = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}, Q_Y = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\i & -i\end{pmatrix} \tag{1.4.6}$$

We can decompose the Q_Y with next procedure.

$$e_{1,Y} = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix} e_{1,x},\ e_{-1,Y} = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix} e_{-1,x} \tag{1.4.7}$$

$$Q_Y = \begin{pmatrix}e_{1,Y} & e_{-1,Y}\end{pmatrix} = \left(\begin{pmatrix}1 & 0\\0 & i\end{pmatrix} e_{1,x} \begin{pmatrix}1 & 0\\0 & i\end{pmatrix} e_{-1,x}\right) = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix} \begin{pmatrix}e_{1,X} & e_{-1,X}\end{pmatrix} \tag{1.4.8}$$

$$\therefore Q_Y = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix} Q_X \tag{1.4.9}$$

The each Q matrix components are standard quantum gates, Hadamard and S gates.

- Hadamard: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

- S: $S = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

Thus, we have next

$$X = HZH \tag{1.4.10}$$
$$Y = SHZHS^\dagger \tag{1.4.11}$$

.

From the above result we could represent X, Y rotation with Z, Hadamard and S gates.

$$RX(2\theta) = \exp(-i\theta X) = \exp(-i\theta(HZH)) = H\exp(-i\theta Z)H \tag{1.4.12}$$

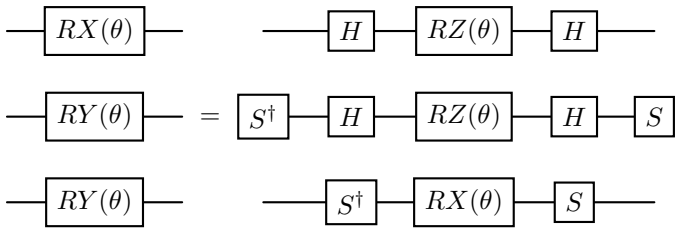


Figure 1.2. Circuit representation of each rotation gates.

Example (Bloch representation) The $\boxed{H}$ and $\boxed{S}$ are basis transformation matrices and in Bloch sphere representation, they are the rotation transformations of qubit state vector.

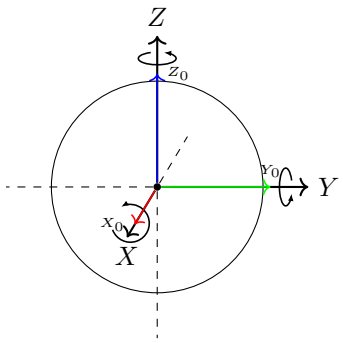


Figure 1.3. Bloch sphere of qubit. Each rotation arrow at the end of the axis indicates RX, RY, RZ rotation.

In Bloch sphere representation, Hadamard gate is a π rotation along, $\frac{1}{\sqrt{2}}(1, 0, 1)$, and S gate is a $\pi/2$ rotation along, Z axis.

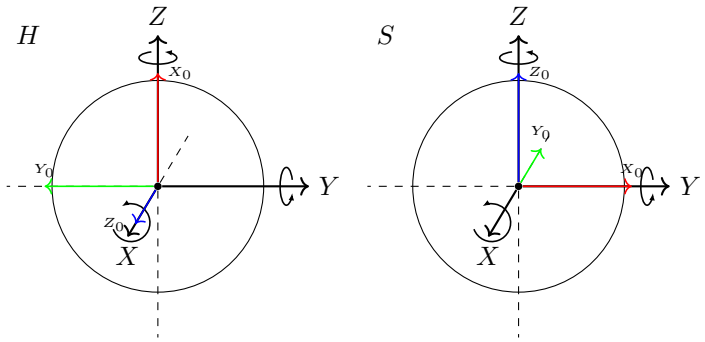


Figure 1.4. Applied axis transformation by Hadamard and S gates. The left is a Hadamard transformation and the right is a S gate transformation.

Subsection 1.4.3

N-qubit system

The extension of the single wires to N-wires on quantum circuit is an approximation of operators of N dimensional $\mathcal{H}$ with tensor product of 1-dim Hilbert spaces. In addition, such extension is efficient. *Solovay-Kitaev Theorem* guarantees that the possibility and efficiency such that "Any n-wires unitary operators could be represented with finite combination of universal quantum gate sets". The theorem permits the developers of the quantum computer to only implement few number of single and two qubit gates for universal applications.

In the previous exercise, we stated that the evolution on single Pauli term is same with the rotation along a specific axis in Hilbert-space, corresponding to the Pauli term. Same statement hold for n qubit wires.

$$Z_1Z_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, I_1Z_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$R(\theta; P_i) = \cos(\theta)I + i\sin(\theta)P_i$

CX structure

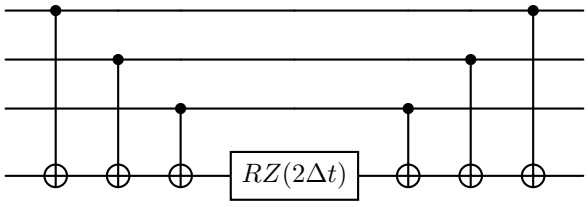


Figure 1.5. Δt evolution circuit of $H = ZZZZ$ Hamiltonian.

In the other manuscripts, the next type circuit also common.

The Fig 1.5 is called by *fountain* type, and Fig 1.6 is called by *chain* type.

Exercise 7 | Show that the above two circuits are identical.

If you construct the circuit in matrix form, you can verify that those two circuits are identical. In the end, the mixture of two type also identical and the position of the

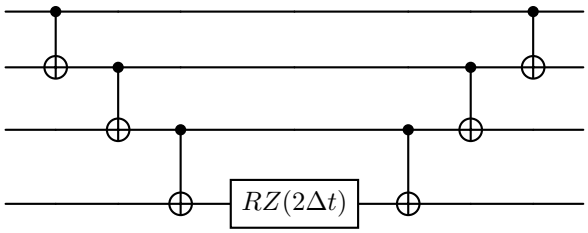
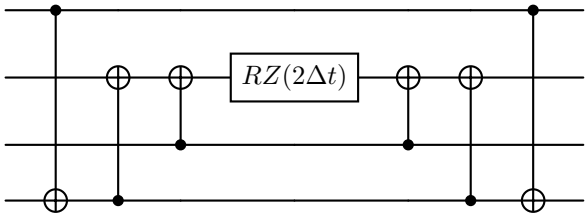


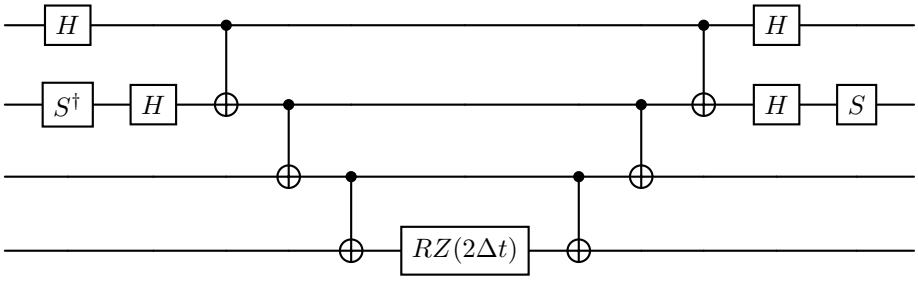
Figure 1.6. Δt evolution circuit of $H = ZZZZ$ Hamiltonian 2nd type.

RZ gate is also freely determined by the situation. The only thing we need is a CNOT gate path visit each i-th qubits in the Hamiltonian at once[6].



Basis transformation

$$\exp(-i\Delta t XYZZ) \tag{1.4.13}$$



Subsection 1.4.4

Implementation with Phase operator

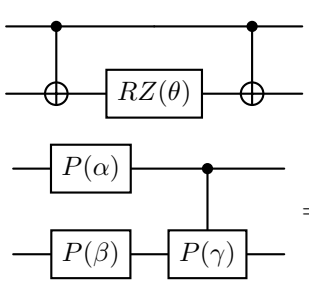
$$\mathrm{P}(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}, \mathrm{RZ}(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

However, in multi-qubit system a rotation z gate and phase gate work differently because

- $\boxed{P(\theta)}$

of the local phases. Even though, we can implement the same operation with phase gates without rotation z gates.

[6] Considering qubit rearangement.



$$= e^{-i\theta/2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\beta} & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & e^{i(\alpha+\beta)\gamma} \end{bmatrix}$$

If $\alpha = \beta = \theta$ and $\gamma = -2\theta$, the two circuit are identical without considering global phase. Similarly, we can arange RZ gates to construct the evolution circuit without CX gate, but using controlled-RZ gate.

Exercise 8 Show that the RZ and controlled-RZ could generate ZZ, IZ, ZI evolution gate on 2 qubit circuit.

Section 1.5

Additional Note

Subsection 1.5.1

Clifford Group

The evolution circuit comprise large 3 gate sets.

- Rotation gates: RZ
- Entaglement gates: $CNOT$
- Basis Transformation gates: H, S

For example, using two rotation gates, RZ, RX we can eliminate H, S gates in the evolution circuit. On the other hand, we can use 3 rotation gates without those basis change gates.

Note that, when you work with gate model, it is very common case that using an ancilla registers to reduce the circuit depth, or to implement some algorithms. During the process, you must take care of the relative phase between the original register and the ancilla register. Sometimes it does not affect but before the calculation you must ensure the consistency. Think about Ahrno-Bohm effect, it is not a strange effect that the local phase difference affect the overall result.

Subsection 1.5.2

Pauli-Frame

As a rotation, geographical representation only work for 1-qubit case. In N-qubit case, there is no representation method including all quantum information and state. Same hold for each rotation gate in Trotterization circuit. We accept the circuit in Fig () is a rotation around axis **nemo**. What about the next figure?

CHAPTER

2

Optimization of evolution circuit

Section 2.1

Introduction

The optimization of quantum circuit has various meaning by the contenxt. In this section, the term *optimization* is used as indicating two concept. First is error reducing technique of circuit representation of time-evolution dynamics, and Second is a direct circuit depth reduction techniques.

Section 2.2

Mutally commuting groups

In the Product formula, Eq(5), the total time evolution operator is approximated with product of several local Hamiltonian operators. It is just an approximation but actively adopted in many references and methods. The reason is that we don't know proper method to find exact evolution operator corresponding to the total Hamiltonian. Moreover, the method represents the locality of the given system well. In such representation, the product order of local operators does not affect the *physical* system or ther is no dependence on the system. It is purely error reduction technique of the implementation.

Start from the 3 terms, A, B, C hermite operators, the given Hamiltonian is $H = A + B + C$. By the BCH formula, the product of $\exp(A)\exp(B)\exp(C)$ has next error terms.

$$\exp(A)\exp(B) = \exp(A+B)\exp(\Theta(A,B)) \tag{2.2.1}$$
$$\exp(A+B)\exp(C) = \exp(A+B+C)\exp(\Theta((A+B),C)) \tag{2.2.2}$$
$$\exp(A)\exp(B)\exp(C) = \exp(A+B+C)\exp(\Theta((A+B),C) - \Theta(A,B)) \tag{2.2.3}$$

However, the error term $\Theta(A,B) = \Theta([A,B])$ and $\Theta(0) = 0$, it means that in the case of commuting operators $[A,B] = [B,C] = [A,C] = 0$, the product formula is exactly same the proper time evolution operator,

$$\exp(A)\exp(B)\exp(C) = \exp(A+B+C) \tag{2.2.4}$$

It is not a general case of the simulation, however, there are mutually commuting subsets exists generally. The situation of every local Hamitlonians are anti-commuting each other is also a rare case as much as the all commuting case.

Childs et al derived product formula error with commutator scaling[5].

Theorem 10 **Product error with commutator scaling**
Let, $\mathcal{L}_p(H,t)$ be a p-th order product formula of the given Hamiltonian, $H = \sum_i H_i$. Then, the error of the p-th order approximation is

$$\|\mathcal{L}_p(H,t) - \exp(-itH)\| = \mathcal{O}(\tilde{\alpha}_{com} t^{p+1}), \tag{2.2.5}$$

where, $\tilde{\alpha}_{com} := \sum_{\lambda_1,\lambda_2,\ldots,\lambda_{p+1}} \|[H_{\lambda_{p+1}}, \ldots [H_{\lambda_2}, H_{\lambda_1}]]\|$

The reduction of big-O error does not significantly large, however, mutually commuting group allow us some freedom to manipulate additional optimization about the circuit gates.

Section 2.3

Hamiltonian grouping problem

We can construct a graph of which edges are indicating commuting, anti-commuting relathion of Pauli nodes. Such graph is called *compatible graph*. Fig (2.1) is a example compatible graph of 1 qubit system.

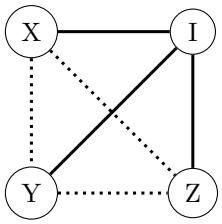


Figure 2.1. The straight line indicates commuting relationship of two ends of the edge and the dottted line indicates anti-commuting relathinship.

The n-qubit system also has such compatible graph of n-fold Pauli-strings. It is beacuse the n fold Pauli string always either anti-commute or commute each other.

$$[P_i^n, P_j^n] = 0 \,\text{or}\, \{P_i^n, P_j^n\} = 0 \tag{2.3.1}$$

where, [] is a commutator and {} is an anti-commutator. Since, any given Hamiltonian has a n-folded Pauli-polynomial represenation, a specific Hamiltonian would be represented as a subgraph, $G_{\mathcal{H}}$ of nodes in the n-fold compatible graph, G.

We can reduce a commuting error by minimizing the number of nested pair of anti-commuting Pauli-strings on the circuit. It is equivalent to finding a mutually commuting partition of Pauli-set.

Definition 7 **Pauli Partitioning Problem**
For a set of n fold Pauli strings, $\mathcal{P}^*$, and a given subcollection $\mathbf{S} \subseteq \mathcal{P}^*$, Pauli Partitioning Problem(PPP) is to return a partition of $\mathbf{S}$ into the fewest number of commuting parts.

$$[p_i, p_j] = 0 \,\forall p_i, p_j \in P_l \in \mathbf{S} \tag{2.3.2}$$

where, $\forall p_i$ is a pauli-string.

However, determining the commutation of the arbitary Pauli-strings and find a mutually commuting partition both of are not simple jobs.

are conceptually easy but computationally, it is a tedious work.

Subsection 2.3.1

Commutator of n-folded string

There are two common method to determine the commutation relationship of n folded Pauli-strings. *general commutativity*, and *qubit-wise-commutativity*[11].

Theorem 11 **General-Commutattivity**(GC)
For n-folded Pauli string, $P_j^n = \otimes_i^n p_i^j, P_k^n = \otimes_i^n p_i^k$,

$$[P_j^n, P_k^n] = 0 \Leftrightarrow \forall i, \text{occurence of } \{p_i^j, p_i^k\} = 0 \text{ is } 2l,\ l \in \mathbb{Z}_+ \tag{2.3.3}$$

Theorem 12 **Qubit-Wise-Commutattivity**(QWC)
For n-folded Pauli string, $P_j^n = \otimes_i^n p_i^j, P_k^n = \otimes_i^n p_i^k$,

$$\forall i, \{p_i^j, p_i^k\} = 0 \Rightarrow [P_j^n, P_k^n] = 0 \tag{2.3.4}$$

QWC is a sub-relationship of GC. QWC commuting or anti-commuting pair is a commuting or anti-commuting pair in GC. A reverse is not hold in general case.

The problem is for n-qubits system, there are 2^n number of Pauli-strings. Constructing commuting/anti-commuting map of the Pauli-strings requires next number of operations with GC.

$$\binom{2^n}{2} n = O(4^n n) \tag{2.3.5}$$

It has an expotential time complexity to achieve the compatible graph. Therefore, some frameworks only offer QWC method or providing pre-calculated commuting set in restricted dimension.

Reggio et al suggested acceleration technique in commuting term determinantion [12]. The similar result was introduced in 2009 from theories of Möbius pair of simplices by Havlicek et al[13]. They decompose the Pauli-term into two faimilies and represent the strings as product of two family memebers. For example, X, Z families are

- X-family: $IIIX$, $XIXI$, $IIXI$, $IXXX, \ldots$
- Z-family: $IIIZ$, $ZIZI$, $IIZI$, $IZZZ, \ldots$

then, every Pauli string, even a string containing Y elements, can be represented with a production of two families, $x_i \cdot z_j$. For example,

$$P_l = YZIX = (XIIX) \cdot (ZZII) = x_i \cdot z_j \tag{2.3.6}$$

Theorem 13 For a given pair of two n-fold Pauli strings, P_i, P_j, there is a X, Z family product representation as

- $P_i = x_k \cdot z_l$
- $P_j = x_m \cdot z_n$

.
then the given pair strings are commuting each other if and only if $[z_l, x_m] = [z_n, x_k]$.

Simply,

$$[P_i, P_j] = [x_k \cdot z_i, x_m \cdot z_n] = \begin{cases} 0 & \text{if } [z_i, x_m] = [x_k, z_n] \\ -P_i P_j & \text{otherwise} \end{cases} \tag{2.3.7}$$

This allows us to determine the commutation of two n fold string with only a few operations of 4 binary values. Unfortunately, this method does not allow us to avoid expotential cost increasing for larger n.

Sympletic representation of Pauli element

The XZ code is a kind of sympletic representation of Pauli group element.

Using this notation you can represent the various algebra with simple integer operations, not a $2^n \times 2^n$ dimension matrix operation.

- Group addition:
- Linear combination:
- Tensor product:

In addition, the XZ code itself is a specific matrix index where, the each Pauli terms are mapped into standard basis of matrix space.

$$P_i = e_i \tag{2.3.8}$$

Using this representation, you can fastly decompose the given Hamiltonian as Pauli-polynomial. See details in Append D.

Subsection 2.3.2

Partition construction

Once the compatible graph is constructed, the task at hand is to search for the mutually commuting partitions from the given compatible graph of Pauli-strigns. It is equivalent with Max-clique problem[3], unfortunately, it is a well known NP problem[14]. There have been many attemptions to solve or approximate the solution of the problem. However, in this documnet, we would like to indtroduce Adiabatic approximation technique suggested by Kurita et al[15].

Kurita et al used a graph partitioning Hamiltonian and constructed sequetial clique extracting algorithm. The compatible graph modeled as binary, 0 and 1, weighted complete graph by edges of commutation are marked as 0 and the anti-commutation ones are marked as 1. Basic procedure of Kurita et al is

1. Extract max-clique set, P_i, from the compatible graph.
2. Delete the nodes which were extracted from the step 1 from the graph.
3. Repeat until there is no remaining node after step 2.

.

The mutually commuting partitions are constructed by each max-cliques, $\{P_i\}_{i=1}^{N}$. The mutually commutation relationship is guaranteed by the next Hamiltonian, quadratic ising model.

$$\mathcal{H} = -\sum_i Z_i + \sum_{i>j} Z_i Z j \tag{2.3.9}$$

[3]See details of Max-clique problems in Appendix F.2

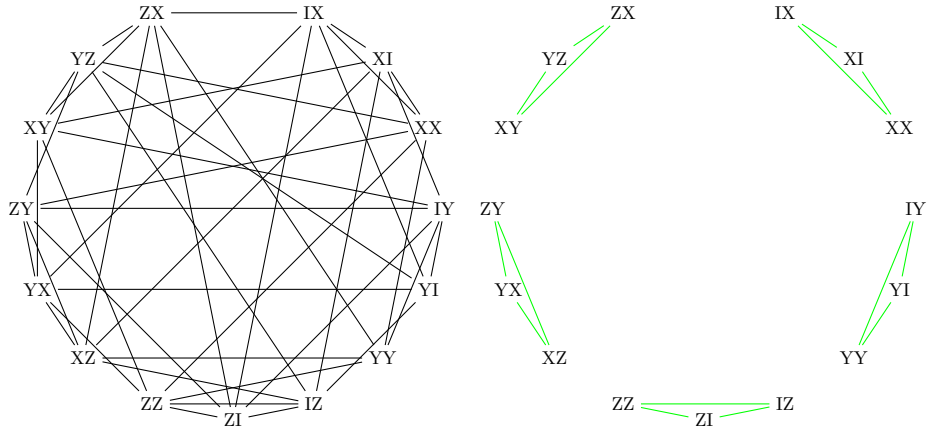


Figure 2.2. Left: Compatible graph of 2-qubit system Pauli-strings. Each edge indicates commuting relationship between the two ends. The edges in the figure weighted zero, and edges between disconnected nodes are weighted as 1. Right: The figure was refered from Kurita et al. 2023, redrawed by the author[15].

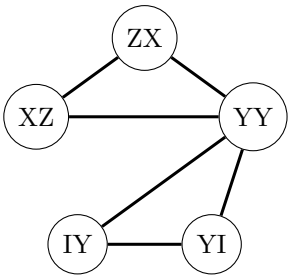
The first term, $-\sum_i Z_i$, reduces the state energy proportional to a number of nodes in the clique. The seocond term, $\sum_{i>j} Z_i Zj$ rasies the state energy as much of the number of the anti-commuting terms in the sample. At each step, the clique searching was conducted by stimulated annealing chip provided by Fujititu.

Section 2.4

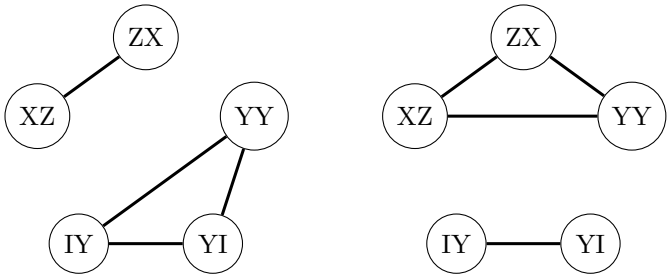
Axis transformation weight

The main idea of Kurita et al allow us to find mutually commuting partition of the given Hamiltonian set. However, there are some degeneracy in extracting cliques from the pauli-graph of the given Hamiltonian.

Consider the next Pauli graph.

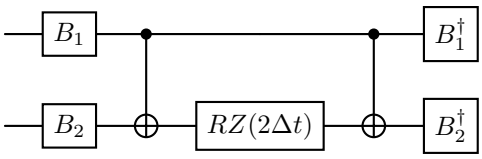


We have two possible partitions in the optimization process.



These two partition are indistinguishable in binary weighted graph suggested by Kurita et al. That means, the system has a degeneracy in the ground state. In general, each intermediate steps we would face such degeneracy in the max-clique finding problem. However, if we consider an additional implementation cost on the circuit, we can remove such degeneracy from the process. The author, Kim, suggested basis transformation weight to the Kurita's Hamiltonian.

In 2 qubit system, general evolution operator of length two Pauli-string on circuit would have next form.



By the axis of the rotation, the basis transformation gates rotate the qubit axes. However, some cases intermediate gates between rotation gates are not necessary to exist. If the nested rotation axes, mutually commuting by the method of implementation we can ignore the basis transformation circuit. See an example and details in *Pauli-Frame*.

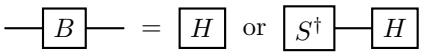
Two parameters could be considered in the problem.

- Circuit depth
- Number of gates

Subsection 2.4.1

Rough implementation of single qubit transformation

Transformation method to the graph clique problem and corresponding problem is well established in Kurita et al.



Definition 8 **Single qubit transformation depth**

The transformation depth of the nested axes, A_1, A_2 is defined as,

$$w(\cdot) : P^1 \times P^1 \to \mathbb{Z}_+ \tag{2.4.1}$$

where, $P^1 = \{I, X, Y, Z\}$.

Nested Axes	Transformation gate	Transformation Depth
A A	I	0
Z X	H	1
Z Y	$S^\dagger H$	2
X Y	$HS^\dagger H$	3

Expand them to the multi-qubit Hamiltonian, then

$$W(S_i, S_j) := \frac{1}{N}\sum_{k=1}^{N} w((S_i)_k, (S_j)_k) \tag{2.4.2}$$

$$\mathcal{H} = -\mu_o \sum Z_i + \mu_1 \sum_{i<j} h_{ij} Z_i Z_j + \mu_2 \sum_{i<j} w_{ij} Z_i Z_j \tag{2.4.3}$$

where, $h_{ij} \in \{0, 1\}$ and $w_{ij} \in [0, 1]$.

To avoid the reversed energy of the commuting and non-commuting states, the prior coefficients, μ_0, μ_1, μ_2 must satisfies the next constraints,

$$\begin{cases} ||\mu_1|| > N||\mu_0|| \\ ||\mu_0|| > \frac{1}{2}N(N-1)||\mu_2|| \end{cases} \tag{2.4.4}$$

The transofrmation weight could be differ by the basis gate set of th system and implementation unit. For example, the above construction was based on introductory level implementation of time-evolution gate using RZ, CNOT and Hadamard and S gates. However, there are many way to construct the gate. You can use RX or RY instead of RZ, simulatneously use two rotation gates, CZ, CY, or instead of CNOT, or instead of RZ gate, you can choose phase gate.

Subsection 2.4.2

Pauli-Frame metric

Yet's we even have no idea of quantum advantaged algorithm for TSP problem. Therefore, in this approach we seperated the problem in to several well-known , at least come conveience approximation method exists, problems.

Main idea is same with the Kurita et al's approach however, we combined them with Pauli-Frame transformation cost in the partitioning steps.

In the design of the graph optimization Hamiltonian, what is a proper ground state must be considered. For example, the actual path of each Pauli-strings would be formated after the partition generated. Then, must the highest weighted edge be cut in partition steps or formation steps? Answer, even we can find bypass route to avoid such path, it is more appropriate to cut the huge weighted edge in partition formation step.

The main idea is energy split the graph partitioning Hamiltonian by adding basis transformation cost. See an example case of degenerated situation in Kurita et al. The next compatible graph has two solution for max-clique extraction.

However, these two configurations are not same for the total error reduction in final result including implementation error.

Now, we are considering circuit-depth of the evolution circuit. What is a more suitable for depth reduction in this case?

Basis transform cancelling and multi-rotating pauli-frame could be a good estimation measure.

Considering a two

by the basis of each qubit, the transform layer depth is 0-3. in 0- 1 cases precisely defined CNOT gate allow more reduction of circuit.

See, Pauli-frame optimization.

Now, our job is applying such configuration in clique optimization.

insist With mutually commuting partition, the degree of the freedom in Pauli-frame in optimization process is greatly reduced.

The heuristhic assumption was if the most high cost edge exists in node set. It requires other edges than the lower cost edges to find minimum cost path travling all nodes.

Note: Idea The problem of Pauli-frame optimization is that it is a Traveling Shopper problem which is more complex and higher NP-Problem of TSP(Traveling Saleman Problem). The Pauli-frame optimization automatically satisfies the muttually commuting order optimization, since a Pauli-frame in steps represents the mutually commuting Pauli-strings groups. More precisely, Pauli-frame represent the simulatneously possible pauli-strings to be implemented on single operation step. Optimizing the Pauli-frame is already considering Basis-transform cost.

What if the pre-condition that the given pauli-strings are mutually commuting sets?

Section 2.5

Krylov transformation

Krylov method is a method of change the basis order by the significant of system evolution and drop the low-affection terms to reduce the size of the quantum system.

Subsection 2.5.1

Other techniques

Section 2.6

VQE method and evolution

Part II

Practical Examples

CHAPTER

3

Simple quantum systems

In this chapter, we will review some technique and materials for you to implement quantum simulation with quantum computer. Formulation of the system and requirements and progress are discussed in the below sections. The contents are dicussed in aiming a specific quantum systems, 1 dimensional particle systems. It is a very classical example of the physics system but has many attributes to study in quantum mechanics.

Note: It is not a full coverage of the quantum simulation with quantum computer. There are plenty of different quantum systems and they need individual modeling techniques.

Section 3.1

Modeling of 1dim Particle

Now let's take an example, this was introduced in *Box 4.12* in Nielsen and Chuang textbook. The particle in 1-dim system of momentum and mass, p, m and the potential, V can be expressed with next hamiltonian.

$$\mathcal{H} = \frac{p^2}{2m} + V \tag{3.1.1}$$

The wave function of the system is $|\psi\rangle$. If we discretize the line with n number of subregions, $x_1, x_2, \ldots x_n$, wave function have next representations.

$$|\psi_d\rangle = \int_{-\infty}^{\infty} |x\rangle\langle x||\psi\rangle \, dx \tag{3.1.2}$$

$$= \int_{-L}^{L} |x\rangle\langle x||\psi\rangle \, dx \tag{3.1.3}$$

$$\approx \sum \lambda_i |x_i\rangle \tag{3.1.4}$$

The last representation is a finite approximation of the quantum system. Using computation basis and their binary representation, where $0111 = 1 * 2^0 + 1 * 2^1 + 1 * 2^2 + 0 * 2^3 = 7$, we can simulate the $-d \leq x \leq d$ region particle wave function with

$$|\psi\rangle = \sum_{k=-d/\Delta x}^{d/\Delta x} \lambda_k |k\Delta x\rangle \tag{3.1.5}$$

The expectation value of the position is measured with the observable, $\mathbf{X}$

$$\mathbf{X} = \sum i\Delta x |i\Delta x\rangle\langle i\Delta x| \tag{3.1.6}$$

$$\langle x\rangle = \langle\psi_d|\mathbf{X}|\psi_d\rangle \tag{3.1.7}$$

Similarly, the momentum observable is

$$\mathbf{P} = \sum_j p_j |k_j\rangle\langle k_j| \tag{3.1.8}$$

In the representation the V term can be expressed with x_i operators directly, however if some observables are defined with $\hat{p}$, we needs a transformation. In continuous space, the canonical variables have next relationship on Fourier transformation.

$$\phi(p) = \frac{1}{\sqrt{2\pi\hbar}} \int \psi(x) \exp\left(-i\frac{px}{\hbar}\right) dx \tag{3.1.9}$$

$$\psi(x) = \frac{1}{\sqrt{2\pi\hbar}} \int \phi(p) \exp\left(i\frac{px}{\hbar}\right) dp \tag{3.1.10}$$

In matrix representation,

$$H = \sum_{ij} h_{ij} |x_j\rangle\langle x_i| \tag{3.1.11}$$

$$= P + V \tag{3.1.12}$$

$$= (\text{QFT})(\sum_i p_i |p_i\rangle\langle p_i|)(\text{QFT}^\dagger) + \sum_{ij} v_{ij} |x_j\rangle\langle x_i| \tag{3.1.13}$$

where, QFT is a quantum fourier transform.

We can discrete the wave function as n number of discreted vector, $\{|x_i\rangle\}_n$ indicating the probability of finding the particle in $x_i \pm \Delta x$ region. With manuplating V value by the time, we can simulate various experiment, such as slit experiment or quantum tunneling effects. See section 3.1 for details of implementation.

One thing you can confuse in the above matrx representation is canonical commutation relationship.

$$[\hat{x}, \hat{p}] = i\hbar \tag{3.1.14}$$

Now the observables are indicated in finite dimension matries, $\mathbf{X}, \mathbf{P}$. They would be diagonal matrix in each basis. Question is that "Is the canonical commutation relationship preserved in the matrix representation?" Unfortunately, the answer is **no**. You may think that it is because of the low precision of discrete Fourier transformation in finite dimension. It could be a reason, then how much precision is required to meet the $[\hat{x}, \hat{p}] = i\hbar$? The answer is ∞. The fact is that the operator $\hat{x}$ and $\hat{p}$ are not bounded operators.

We can prove the unboundness of the operators, easily.

$$[\hat{x}^n, \hat{p}] = i\hbar n \hat{x}^{n-1}$$

$$2||\hat{p}||||\hat{x}||^n \geq ||\hat{x}^n\hat{p}|| + ||\hat{p}\hat{x}^n|| \geq n\hbar||\hat{x}||^{n-1}$$

$$2||\hat{x}||||\hat{p}|| \geq n\hbar$$

the above inequality hold for **any** $n \geq 1$.

In fact, in the finite dimension, the canonical relationship should be $[\hat{x}, \hat{p}] = 0$, only if in the infinite dimension, $[\hat{x}, \hat{p}] = 1$ and if we approximate $n \to \infty$ the commutator $[\hat{x}, \hat{p}]$ would show Dirac-Delta like behavior[16].

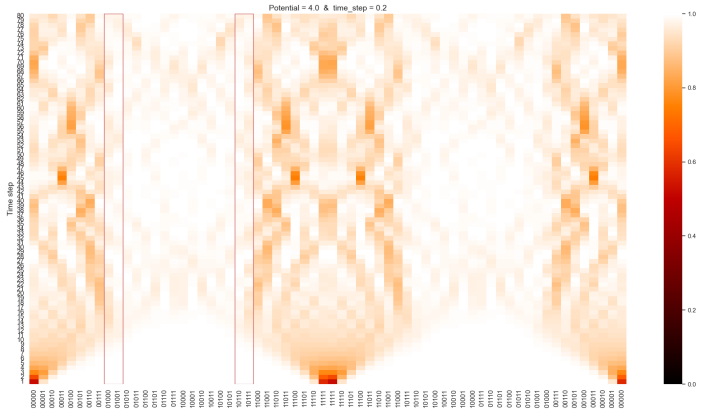


Figure 3.1. The symmetric potential.

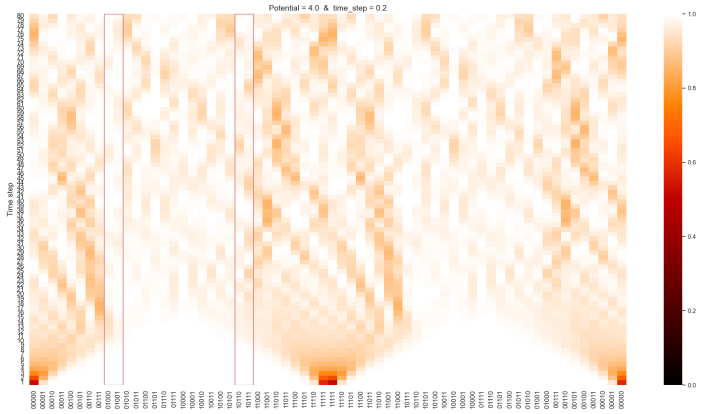


Figure 3.2. The reduced the right potential as 1/4 times of the left potential.

Subsection 3.1.1

Modeling a Hamiltonian

One thing you kepp in mind is that the finite dimension modeling with Weyl's clock and shift method. The position domain is not a straight long 1 dimension line. It would be a circular loop position, where $|2^n\rangle = |0\rangle$. Some 1 dim simulation have those error, especially, when they simulate the quantum tunneling. They did not considering the circular connected space and mis interpretated the amplitude of wave function.

The Fig 3.1, 3.2 would be helpful understand the situation.

Subsection 3.1.2

Slite experiment

Using a tunneling circuit and the time dependent Hamiltonian, the Slit experiment could be simulated with quantum computer where the time t indicate the position x, the axis where particle is moving.

Subsection 3.1.3

CHAPTER 4

Adiabatic process on gate model

Adiabatic computation

In practical implementation, adiabatic model of quantum computation has many advantanges in scale, speed and formulation of problem to run on the system. On the gate model, we consider many approximation formula and represetation of the evolution operator. The reason was that we cannot know exactly what form of the time-evolution operator form corresponding to the given hamiltonian matrix;hermite. However, in adiabatic computation, the unitary transformation is a **job of nature**. Only things we have to consider are how to apply the hamiltonian to the given qubit system and accelerating convergence speed enough to use in calculation. This computation model has very huge benefit in binary optimization problem than the gate model computation.

In this chapter, we are going to simulate adiabatic process on gate model computer and compare the result obtanined from a commerical adiabatic process machine.

Section 4.1

Formulation for Adiabatic computing

Adiabatic theorem states that if the system hamiltonian is varying slowly, the final state of the system is remained in eigenstate of finial hamiltonian which is corresponding eigenstate of the initial hamiltonian eigenstate.

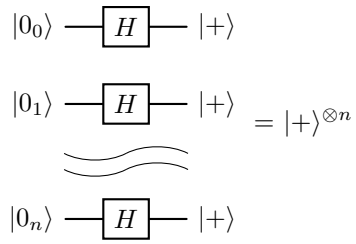
The term slowly is very important, it depends on the initial and final hamiltonian of the system. If the evolution time is T, and the eigenvalues of initial and final hamiltonian are E_i and, E_f, the evolution time must be satisfies next,

$$\frac{2\pi}{|E_f - E_i| << T} \tag{4.1.1}$$

That is to solve the system with adiabatic method, we have to manipulate the time, T. It is called by *adiabatic configuration*. The configuration depends on the initial and final system, so the determination requires some physical instuition by the researcher.

$$\mathcal{H}_{adia}(f(s), T) = T(1 - f(s))\mathcal{H}_{initial} + f(s)\mathcal{H}_{solve} \tag{4.1.2}$$

Our major concern is a ground state of $\mathcal{H}_{solve}$. We can expect the system to be a ground state through adiabatic process, if the initial system state was a ground state of $\mathcal{H}_{inital}$. That is, with adiabatic process, we can obtain the target system ground state using well-known system. The requirement is an appropriate Hamiltonian, $\mathcal{H}_{initial}$. It must have well-known ground state and their implementation should practically be efficient. Common Hamiltonian is a $\mathcal{H}_{inital} = -\sum_i \sigma_{X,i}$. Its ground state is a $|+\rangle^{\otimes n}$.



The common annealing solutions, such as D-Wave and QuEra, also apply the adiabatic theorem to their initialization process and Hamiltonian form. Quantum annealing is fundamentally based on the phenomenon of quantum fluctuation; however, the starting and final stages, as well as the evolving process, follow the adiabatic theorem. That is why they offer some annealing time and initial state parameters to the user API. Some problems require much more time to achieve the appropriate adiabatic process

Section 4.2
Implementation

The previous simulations were time-independent Hamiltonian systems, but adiabatic process is using a time-dependent Hamiltonian. The commutation problem still arises in here. If the $H(t_1)$ and $H(t_2)$ are not commuting, then we cannot apply the Still, product formula work well except the additional error term was added to the system.

Subsection 4.2.1
QUBO

Quadratic unconstrained binary optimization problem. It is widely tried in adiabatic computation and in gate-model computation VQE(Variational Quantum Eigensolver) method is dominant however, in this chapter we will solve the QUBO problem on gate model computer mimicing the adiabatic process.

Subsection 4.2.2
Ising model

Section 4.3
Pennylane implmentation

Subsection 4.3.1
Graph partitioning problem

CHAPTER

5

Imaginary Time evolution

Section 5.1

Imaginary time

Imaginary time evolution is frequently used in quantum system simulation especially, in statistical physics. In the statistical mechanics it is called by *Diffusion Montecarlo method*. The reason is that from the Schrödinger equation, we can directly obtain a progress to reach the ground state of the system. From the time-independent Schrödinger equation,

$$i\hbar\frac{\partial}{\partial}|\psi\rangle = H|\psi\rangle \tag{5.1.1}$$

takes $\tau = it$ then, $df/d\tau = (dt/d\tau)df/dt = idf/dt$, thus,

$$\frac{\partial}{\partial\tau}|\psi\rangle = H|\psi\rangle \tag{5.1.2}$$

$$|\psi(\tau)\rangle = e^{-\tau H}|\psi(0)\rangle \tag{5.1.3}$$

From the Eq (5.1.2), we get a general solution,

$$|\Psi\rangle = \sum_i c_i(0)e^{-E_i\tau}|\psi_i\rangle \tag{5.1.4}$$

The terms are decaying and does not oscillate. Therefore, with sufficient long time, τ, we get

$$|\Psi(\tau) \approx_{\tau\to\infty} c_g(0)e^{-E_g\tau}|\psi_g\rangle \tag{5.1.5}$$

However, the imaginary time evolution cannot be directly implemented in common gate model computer. Since, it is a non-unitary transformation, $\exp(-H\tau)^\dagger = \exp(-H\tau)$, Hermit. Therefore, the implementation of ITE is also a challenge in the quantum computation. There are several ways, one is approximate the non-unitary gate with unitary gate. It is a common method by VQE, we can approximate unitary process to reach the same result of the imaginary time steps. The other methods are a block-encoding, embeding the non-unitary gate into unitary gate, or using a measurement based approach. The method we used in the implementation was non-unitary Trotter circuit suggested by Leadbeater et al[**?**]. Once we get a proper ITE routine, then the quantum computer could find a ground state from the *given* states.

Section 5.2

Non-unitary on unitary

There are many method to obtain non-unitary operation on unitary system.

Subsection 5.2.1
Embeding in unitary

Even the whole gate was unitary, their sub-block matrix could be non-unitary. Using an ancilla register, we can obtain a non-unitary operation on the main register.

Especially, the measurement based operation suggested by is based on Trotterization.

Subsection 5.2.2
VQE approximation

Section 5.3
Spectrum Search

CHAPTER 6

Variational method

It is considered practically useful method in NISQ era[1].

Review [**?**].

Section 6.1

Introduction

Subsection 6.1.1

Variatiaonal Principle

Subsection 6.1.2

Circuit as ansatz

The term *ansatz* means an assumption in mathematics or physics to solve the given problem. In the physics, it is commonly refer a model representing or containing the solution of the problem. The statement of *Is the ansatz proper to solve the problem?* refers the whether is it possible that "The solution could be represented with the model" or, in optimization problem, "The ansatz achive the desire solution".

Using a paramerized circuit, we can treat the quantum circuit as a neural network in machine leanring. However, if the result is not bounded, the optimization is meaningless for any kinds purpose. That is why we need a *Variational principle*. Since, quantum circuit is a kind of quantum system and their ground state and energy is limited in finite value. Same statement holds for parameterized circuit, there always exists a ground state energy[2].

Whatever energy we measure from quantum circuit, it is always higher or equal with the ground state.

$$E_{ground} \leq \langle\psi|H|\psi\rangle$$

We can optimize the circuit without any worry about their convergence.

Principle 1 **Variational Principle**

$$E_g \leq \langle\psi|H|\psi\rangle \leq E_{\max} \tag{6.1.1}$$

Subsection 6.1.3

Universality of Circuit

[1] Noisy intermediate scale quantum era

[2] Not a state. There could be degenerated states.

One of the strong background of the neural network model is *Universal Approximation Theorem*.

How about quantum circuit? Do we have any those type of theorem? The answer is yes!. Quantum circuit is also a universal approximator. In addition, it is more powerful. Even with a single qubit we can achieve the property[17].

Section 6.2

Section 6.3

Ansatzs

Subsection 6.3.1

Hardware Efficient

Subsection 6.3.2

Evolution Based

Section 6.4

Application

Subsection 6.4.1

QUBO

Subsection 6.4.2

Evolution

Real time

Imaginary time

Section 6.5

Implementation

Subsection 6.5.1

Pennylane

Subsection 6.5.2

Qiskit

Bibliography

[1] B. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction.* Graduate Texts in Mathematics. Springer International Publishing, 2015.

[2] Masuo Suzuki. On the convergence of exponential operators—the Zassenhaus formula, BCH formula and systematic approximants. *Communications in Mathematical Physics*, 57(3):193–200, October 1977.

[3] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, August 1996.

[4] Naomichi Hatano and Masuo Suzuki. Finding Exponential Product Formulas of Higher Orders. In R. Beig, W. Beiglböck, W. Domcke, B.-G. Englert, U. Frisch, P. Hänggi, G. Hasinger, K. Hepp, W. Hillebrandt, D. Imboden, R. L. Jaffe, R. Lipowsky, H. V. Löhneysen, I. Ojima, D. Sornette, S. Theisen, W. Weise, J. Wess, J. Zittartz, Arnab Das, and Bikas K. Chakrabarti, editors, *Quantum Annealing and Other Optimization Methods*, volume 679, pages 37–68. Springer Berlin Heidelberg, Berlin, Heidelberg, November 2005. Series Title: Lecture Notes in Physics.

[5] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021.

[6] H. F. Trotter. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959.

[7] Joel E. Cohen, Shmuel Friedland, Tosio Kato, and Frank P. Kelly. Eigenvalue inequalities for products of matrix exponentials. *Linear Algebra and its Applications*, 45:55–95, June 1982.

[8] Ronald De Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019.

[9] Masuo Suzuki. General Decomposition Theory of Ordered Exponentials. *Proceedings of the Japan Academy, Series B*, 69(7):161–166, 1993.

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

[11] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T. Chong. $O(N^3)$ Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2020.

[12] Ben Reggio, Nouman Butt, Andrew Lytle, and Patrick Draper. Fast Partitioning of Pauli Strings into Commuting Families for Optimal Expectation Value Measurements of Dense Operators, June 2023. arXiv:2305.11847 [hep-lat, physics:hep-ph, physics:quant-ph].

[13] Hans Havlicek, Boris Odehnal, and Metod Saniga. Moebius Pairs of Simplices and Commuting Pauli Operators, August 2009. arXiv:0905.4648 [math-ph, physics:quant-ph].

[14] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Springer US, Boston, MA, 1972.

[15] Tomochika Kurita, Mikio Morita, Hirotaka Oshima, and Shintaro Sato. Pauli string partitioning algorithm with the ising model for simultaneous measurements. *The Journal of Physical Chemistry A*, 127(4):1068–1080, 2023. PMID: 36653017.

[16] T. S. Santhanam and A. R. Tekumalla. Quantum mechanics in finite dimensions. *Foundations of Physics*, 6(5):583–587, October 1976.

[17] Adrián Pérez-Salinas, David López-Núñez, Artur García-Sáez, P. Forn-Díaz, and José I. Latorre. One qubit as a universal approximant. *Phys. Rev. A*, 104:012405, Jul 2021.

[18] Don N. Page and William K. Wootters. Evolution without evolution: Dynamics described by stationary observables. *Phys. Rev. D*, 27:2885–2892, Jun 1983.

[19] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum time. *Phys. Rev. D*, 92:045033, Aug 2015.

[20] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, October 2019.

[21] Brian Gough. *GSL FFT Algorithms*, May 2023.

CHAPTER

7

Material Simulation

Section 7.1

Hardness of the simulation

Degree of freedom is high enough

CHAPTER

8

Tunneling Phenomenon

Quantum tunneling is one of non-classical phenomenon in real-world with entanglement and wave-particle duality. Suppose that the free particle of m, p mass and momentum pass through a potential varrier, V in 1-dimension.

Section 8.1

Introduction to tunneling

The solution of the Schrödinger equation yields,

$$\psi(x) = A\exp(\pm ikx) \tag{8.1.1}$$

The tunneling effect arises in the situation when $E < V$. In classically, at the $b < x$, the existence possibility of the particle is zero. However, it is not in quantum mechanics

Subsection 8.1.1

alpha-decay

The tunneling effect successfully explained the *alpha*-decay phenonmenon by George Gamow, and Condon and Gurney, indepedently. In the Columb potential of the nuclear, the alpha particle cannot obtain the enough energy to escape the nuclear binding force. However, in many experiment and observation alpha particle emission was occured

Section 8.2

Tunneling time problem

Definition 9

Tunneling time problem
How many time does it take to pass through the potential region, when the tunneling effect is occurred?

This simple question has not been solved yet.

Section 8.3

Criticisims on tunneling time

Figure 8.1.

The major problem is that there does not exist about some common definition of *time* in quantum mechanics. Time is just a parameter in quantum mechanics, meanwhile, in general relativity, spacetime dynamically interacts with matters.

Imaginary time evolution problem: Imaginary number has no proper order by the definition. How can we define proper flow direction of time in imaginary numbers?

There are some definitions of tunneling time. The concepts have their own criticisims about how they well explain the tunneling time problem, however, they are practically used in required field, such electrodynamics.

1. Dwell-time
2. Wignet-time
3. Larmor-time

A proper time observable, T_c, should satisfy the following properteis:

1. On average T_c should estimate te elaspsed time τ.
2. The variance in a measurement of T_c should be independent of the elapsed time τ.

Subsection 8.3.1
Classical approach

In the classsic scale, the potential is simply a hill.

Subsection 8.3.2
Measurement- Experimental

Larmor Precession

The spin is rotated in the uniform magentic field, $B_0\hat{z}$ with constant frequency ω. Such phenomenon is called *Larmor precession* and the frequency is called *Larmor frequency*. If the particle is

Subsection 8.3.3
Hartman effect and Superluminal velocity

Hartman effect

Section 8.4
Simulation and Measurement on Quantum circuit

Consider Larmor precession implementation on quantum circuit. The spatial propagation of 1 dimension is well established at section **??**. If we want to implement Larmor precession, we can add an additional qubit to act as a spin of the free particle, and modifying the Hamiltonian adding spiner term. Does it really a good simulation? Why we do a calculation and simulate the phenomenon on circuit? We want to obtain a complex system behavior through the simulation, however, designing an affecting circuit to such qubit by time-evolution effect is itself a problem we want to solve. In experiment, it could be a good clock system for the tunneling phenomenon. Meanwhile, in simulation, we cannot get a meaningful information from the model.

In addition, it is just a extracting time[4] information from the system register. Why it must be Larmor time depending on only a specific potential; magenetic potential. Tunneling is an universal phenonmenon that does not depends on the types of potential. A good clock simulator must act on the general situation whatever potnetial type it is.

Subsection 8.4.1

Page Wootter Mechanics

The previous definitions of clock and time of tunnleing effect were based on the wave function, probability current and their change by the time.

We are focusing on the situation of **the event occurred** stuation. The particle occurrence proability of the given space is closely related with the event occurrence. However, there is a gap between the stuations. Unfortunately, in the common Schrödinger equation cannot handle the case of our attention. It is because the Schrödinger equation didn't consider the relativity, so that the time is *not an observable quantity*. The dynamical interaction of space time with matter is described by general relativity. Related with gravity and quantum system.

However, Dirac/canonical quatization of gravity yields *Frozen Formalism*[1]. General realtivity can be represented with Hamiltonian form as Eq (8.4.1).

$$H_{GR}[\gamma, P] = 16\pi G G_{abcd}[\gamma] P^{ab} P^{cd} + V\gamma + \sqrt{\gamma}\rho = 0 \tag{8.4.1}$$

$$H_{GR}|\Psi\rangle = 0 \tag{8.4.2}$$

$$i\frac{d}{dt}|\Psi\rangle = H_{GR}|\Psi\rangle = 0 \tag{8.4.3}$$

What does the above equation means? The state vector $|\Psi\rangle$ is frozen which means there is no evolution. This is gravitation quantization problem related to the problem of time.

However, recently, there is a re-splotlighted perspective of quantum time in the field which has been known as *Page-Wootter formalizm*. In 1983, Page and Wootters presented extended version of the equation[18]. This is called PaW mechanics. PaW mechanics treat the time as a quantum degree of freedom in specific Hilbert space, and in the viewpoint, time flow is occurred from the entanglement between such space and the remained physical system. There were some critisim about the formalization of PaW mechanics;proper propagator, ... Giovannetti technically reformalize the PaW structure and showed PaW mechanics can be extended to give the time-independent Schrödinger equation[19].

Evolution as a basis change

The evolution is an unitary transformation, and every unitary transformation is a kind of basis transformation. Even the overall state is frozen by the time. The basis change does not affect the static state, however, locally there could be a dynamics.

Structure of PaW mechancis

The system, $\mathfrak{H}$ is consist of Hilbert space of the ordinary system, $\mathcal{H}_s$ and ancillary Hilbert space, $\mathcal{H}_T$, of clock or time.

$$\mathfrak{H} = \mathcal{H}_T \otimes \mathcal{H}_S \tag{8.4.4}$$

[4] Maybe events

[1] It was impeorted from , Smith, Aleander, "The Page-Wootters formalism: Where are we now?", 10.48660/22010079

$$\hat{\mathbb{J}} = \hbar\hat{\Omega} \otimes \mathbf{1}_S + \mathbf{1}_T \otimes \hat{H}_S \tag{8.4.5}$$

$$|\Psi\rangle\rangle = \int dt |t\rangle_T \otimes |\psi(t)\rangle_S \tag{8.4.6}$$

$$\hat{\mathbb{J}}|\Psi\rangle\rangle = 0 \tag{8.4.7}$$

From unitary time evolution operator on the system, $\hat{U}_S(t)$ we can construct a proper evolution of the system.

$$\mathbb{U} = \int dt |t\rangle_T\langle| \otimes \hat{U}_S(t) \tag{8.4.8}$$

$$= \hat{U}(\hat{T}) = \exp\Big(-i\hat{T} \otimes \hat{H}_s/\hbar\Big) \tag{8.4.9}$$

The main benefit of the PaW mechanics is that the time is an observable quantity in the mechanics and we can obtain a basyian probability on time-dimension about the occurred event following Boron's rule.

$$P(A|B) = \frac{\mathrm{tr}(P_A P_B \rho P_B)}{\mathrm{tr}(P_B \rho P_B)} \tag{8.4.10}$$

In the approach, the author treating the quantum tunneling time on the analysis of *arrival time*.

The goodness of the PaW mechanism is that it can be directly implemented on quantum circuit by ancilla qubit registers. The pure state is well-suited with qubit network and

Part III

Further Topics

Section 8.5

Simulation Techniques

Subsection 8.5.1

VQE evolution

VQE is an abbrevation of Variational Quantum Eigensolver.

In qiskit, they offer two type of evolution modules. One is a Product formula based implementation we already described in the previous contents. The other is a VQE based module that is called Variational quantum time evolution.

The theorem is based on the paper written by Yuan et al[20],

Subsection 8.5.2

Linearization

Subsection 8.5.3

Quantum Walks

Subsection 8.5.4

Qubitaization

Subsection 8.5.5

Fractional Query

Subsection 8.5.6

Taylorization

Bibliography

[1] B. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction.* Graduate Texts in Mathematics. Springer International Publishing, 2015.

[2] Masuo Suzuki. On the convergence of exponential operators—the Zassenhaus formula, BCH formula and systematic approximants. *Communications in Mathematical Physics*, 57(3):193–200, October 1977.

[3] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, August 1996.

[4] Naomichi Hatano and Masuo Suzuki. Finding Exponential Product Formulas of Higher Orders. In R. Beig, W. Beiglböck, W. Domcke, B.-G. Englert, U. Frisch, P. Hänggi, G. Hasinger, K. Hepp, W. Hillebrandt, D. Imboden, R. L. Jaffe, R. Lipowsky, H. V. Löhneysen, I. Ojima, D. Sornette, S. Theisen, W. Weise, J. Wess, J. Zittartz, Arnab Das, and Bikas K. Chakrabarti, editors, *Quantum Annealing and Other Optimization Methods*, volume 679, pages 37–68. Springer Berlin Heidelberg, Berlin, Heidelberg, November 2005. Series Title: Lecture Notes in Physics.

[5] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021.

[6] H. F. Trotter. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959.

[7] Joel E. Cohen, Shmuel Friedland, Tosio Kato, and Frank P. Kelly. Eigenvalue inequalities for products of matrix exponentials. *Linear Algebra and its Applications*, 45:55–95, June 1982.

[8] Ronald De Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019.

[9] Masuo Suzuki. General Decomposition Theory of Ordered Exponentials. *Proceedings of the Japan Academy, Series B*, 69(7):161–166, 1993.

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

[11] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T. Chong. $O(N^3)$ Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2020.

[12] Ben Reggio, Nouman Butt, Andrew Lytle, and Patrick Draper. Fast Partitioning of Pauli Strings into Commuting Families for Optimal Expectation Value Measurements of Dense Operators, June 2023. arXiv:2305.11847 [hep-lat, physics:hep-ph, physics:quant-ph].

[13] Hans Havlicek, Boris Odehnal, and Metod Saniga. Moebius Pairs of Simplices and Commuting Pauli Operators, August 2009. arXiv:0905.4648 [math-ph, physics:quant-ph].

[14] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Springer US, Boston, MA, 1972.

[15] Tomochika Kurita, Mikio Morita, Hirotaka Oshima, and Shintaro Sato. Pauli string partitioning algorithm with the ising model for simultaneous measurements. *The Journal of Physical Chemistry A*, 127(4):1068–1080, 2023. PMID: 36653017.

[16] T. S. Santhanam and A. R. Tekumalla. Quantum mechanics in finite dimensions. *Foundations of Physics*, 6(5):583–587, October 1976.

[17] Adrián Pérez-Salinas, David López-Núñez, Artur García-Sáez, P. Forn-Díaz, and José I. Latorre. One qubit as a universal approximant. *Phys. Rev. A*, 104:012405, Jul 2021.

[18] Don N. Page and William K. Wootters. Evolution without evolution: Dynamics described by stationary observables. *Phys. Rev. D*, 27:2885–2892, Jun 1983.

[19] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum time. *Phys. Rev. D*, 92:045033, Aug 2015.

[20] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, October 2019.

[21] Brian Gough. *GSL FFT Algorithms*, May 2023.

Part IV

Appendix

CHAPTER

A

Hermit and Unitary matrix

Section A.1

Unitary matrix

Let's think about there is a change, whatever it is, in the system. The $|\psi\rangle$ represent all the information of the system, so that it will be changed to $|\psi'\rangle$.

Any modification in the vector space can be represented with an *operator*, $\hat{U}$.

$$|\psi'\rangle = \hat{U}|\psi\rangle \tag{A.1.1}$$

Now, the modified state function also satisfies normalization, such as $\langle\psi'|\psi\rangle = \langle\psi|\psi\rangle$.

$$\begin{aligned}\langle\psi'|\psi\rangle &= \langle\hat{U}\psi||\hat{U}\psi\rangle \\ \langle\hat{U}\psi||\hat{U}\psi\rangle &= \langle\psi|\hat{U}^\dagger|\hat{U}\psi\rangle \\ \langle\psi|\hat{U}^\dagger|\hat{U}\psi\rangle &= \langle\psi|\hat{U}^\dagger\hat{U}|\psi\rangle\end{aligned}$$

we get,

$$\hat{U}^\dagger\hat{U} = \hat{I} \tag{A.1.2}$$

where, $\hat{I}$ is an identity operator.

That means that any state change event in the quantum system must be a unitary operator in vector space, in isolated system. With well defined basis, we can formulate the operator as matrix,

$$\begin{aligned}|\Psi\rangle &= \sum c_i|\psi_i\rangle \\ [\hat{U}]_{\psi_i} &= \sum(\langle\psi_j|\hat{U}|\psi_i\rangle)|\psi_i\rangle\langle\psi_j|\end{aligned}$$

It is little bit weired that the function operation as a matrix, however, we are treating basis function that generating all functions. About the those set of functions we can find well-ordered basis, of course it does not have to be finite. Even in the infinite dimensional vector space, we can find a subspace consist of discreted index basis. Think about the Fourier series of the L periodic function. The basis functions are $\cos(\lambda_n x), n \in \mathbb{Z}_+$.

That is why the unitary matrix is essential topic in quantum computation and simulation.

Subsection A.1.1

Properties of unitary matrix

- It preserves the inner product of two vector, $\mathbf{x}, \mathbf{y}, \langle\mathbf{x}|\mathbf{y}\rangle = \langle U\mathbf{x}|U\mathbf{y}\rangle$

- It is a normal operator: $AA^\dagger = A^\dagger A$.
- $U^\dagger = U^{-1}$
- There exists a Hermit matrix H such that $U = \exp(iH)$.
- Eigenvalues are *unimodular* which is their norms are 1. Therefore, $|\det(U)| = 1$.

In quantum systems symulation, finding a proper unitary operation on system is a significant work. In sometime, we only focus on the result of the operation, in that case we can find some equivalence unitary operators with less implmentation cost.

In the property of the unitary matrix, there is $U = \exp(iH)$. It is a very familiar term in quantum mechanics; time-evolution operator. In finite dimension, the Hamiltonian, H, is represented by *Hermit* matrix.

Section A.2

Hermit matrix

Suppose the Hamiltonian of the system is given as H. The Schrödinger equation yields next.

$$i\hbar\frac{d}{dt}|\psi\rangle = H|\psi\rangle \tag{A.2.1}$$

Hamiltonian is a kind of operator of measurement for energy of the system. It means that the eigenvalues of matrix are energy of the eiegenstates. Such that

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{A.2.2}$$

$$\langle\psi|\hat{H}|\psi\rangle = \langle\psi||\hat{H}\psi\rangle = \langle\hat{H}\psi||\psi\rangle \tag{A.2.3}$$
$$\langle\psi|\hat{H}^\dagger|\psi\rangle = E\langle\psi|\psi\rangle = E\langle\psi|\psi\rangle \tag{A.2.4}$$
$$\therefore E^* = E \tag{A.2.5}$$

$E^* = E, \forall E$, the only complex value satisfying this constraint is a real value. It means that the all eigenvalues of the matrix are real value. Such matrices are called self-adjoint matrix or *Hermit matrix*.

The definition of self-adjoint matrix is

$$H^\dagger = H. \tag{A.2.6}$$

It is equivalence to the all eigenvalues are real condition.

We refered a Hamiltonian as an example of measurement, however, any measurement quantity operators are represented with Hermit matrices. We call them as *observable*.

Subsection A.2.1

Properties of Hermite matrix

- All eigenvalues are real value.
- It is a self-adjoint matrix.
- All eigenvector having different eigenvalues are orthogonal to each other.
- Normal matrix.

- Closed under addition.
- If the two Hermite matrices are commute each other, then their product is Hermite matrix.

Theorem 14 Spectrum Theorem of Hermit matrix

Section A.3

Additional properties

Theorem 15 **Diagonalizability of Hermit matrix**
For a given H matrix, there exists an unitary matrix, U such that

$$H = UDU^\dagger$$

where, D is a diagonal matrix.

There are some common misconcept of the diagonalizability of the Hermit matrix, about the orthonormal basis of the Hermit matrix. The questions are

- Why do the orthonormal eigenvectors exist, since we cannot guarantee the singularity.
- How do we guarantee that the two eigenvectors sharing same eigenvalues are orthogonal?".

".

The two questions are related with a misconcept in linear algebra. The diagonalizability is related with singularity, of course, but it is not equivalent. There are diagonalizable but singular matrices, considering a matrix of which an eigen value is 0.

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

In other word, physically, we can get a 0 energy state by modifying the potential of the system. Therefore, the first question is solved. The second question is equivalent with the existence of the diagonal matrix of Hermit matrix. Since, the unitary operator preserve the inner product, the existence of the diagonal matrix, standard basis, guarantees the orthonormal basis.

Theorem 16 **Eigenvector conservation**
For a given non-singular Hermit matrix, H, with eigen values and their corresponding eigen vectors, $(\lambda, \mathbf{v})$,

$$H \cdot \mathbf{v} = \lambda \mathbf{v}$$

then,

$$e^H \mathbf{v} = e^\lambda \mathbf{v}$$

Proof From the taylor representation of matrix exponential,

$$e^A = \sum_{n=0}^{\infty} \frac{A}{n!}$$

then,

$$\begin{aligned} e^H \mathbf{v} &= \left[I + H + \frac{1}{2!}H^2 + \cdots + \frac{1}{k!}H^k + \ldots \right] \mathbf{v} \\ &= \left[I\mathbf{v} + H\mathbf{v} + \frac{1}{2!}H^2\mathbf{v} + \cdots + \frac{1}{k!}H^k\mathbf{v} + \ldots \right] \\ &= \left[\mathbf{v} + \lambda\mathbf{v} + \frac{1}{2!}\lambda^2\mathbf{v} + \cdots + \frac{1}{k!}\lambda^k\mathbf{v} + \ldots \right] \\ &= \left[1 + \lambda + \frac{1}{2!}\lambda^2 + \cdots + \frac{1}{k!}\lambda^k + \ldots \right] \mathbf{v} \\ &= e^\lambda \mathbf{v} \end{aligned}$$

□

Theorem 17 **Unitary on exponential**
For given unitary matrix, U,

$$e^{UAU^\dagger} = Ue^A U^\dagger.$$

Proof

$$\begin{aligned} e^{UAU^\dagger} &= I + UAU^\dagger + \frac{1}{2!}(UAU^\dagger)^2 + \cdots + \frac{1}{k!}(UAU^\dagger)^k + \ldots \\ &= UIU^\dagger + UAU^\dagger + \frac{1}{2!}(UAU^\dagger)^2 + \cdots + \frac{1}{k!}(UAU^\dagger)^k + \ldots \\ &= U(I + A + \frac{1}{2!}A^2 + \cdots + \frac{1}{k!}A^k + \ldots)U^\dagger \\ &= Ue^A U^\dagger \end{aligned}$$

□

CHAPTER

B

Notation in Quantum circuit

Section B.1

State

Decimal Notation Product Notation

CHAPTER

C

Fourier Trasnformation

Section C.1

DFT and FFT

Practical implementation of the FFT algorithm is well described in *GSL FFT Algorithm* [21].

Section C.2

QFT

Quantum Fourier transformation is a Fourier transoformation defined on the finite group.

$$|\psi\rangle = \sum_{j=1}^{n} x_j |j\rangle \rightarrow_{QFT} \sum_{j=1}^{n} y_k |k\rangle \tag{C.2.1}$$

where, $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left(2\pi i j \frac{k}{N}\right)$

Theorem 18 QFT is an unitary transformation.

Proof Let, $\hat{T}$ be an QFT defined on Eq (C.2.1),

$$\hat{T}\left(\sum_{j=0}^{N-1} x_j |j\rangle\right) = \sum_{k=0}^{N-1} y_k |k\rangle \tag{C.2.2}$$

$$= \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left(2\pi i j \frac{k}{N}\right) |k\rangle \tag{C.2.3}$$

$$= \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(2\pi i j \frac{k}{N}\right) |k\rangle\langle j|x_j|j\rangle \tag{C.2.4}$$

thereby, we can formulate the operator, $\hat{T}$ as $|k\rangle, |j\rangle$ states.

$$\hat{T} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(2\pi i j \frac{k}{N}\right) |k\rangle\langle j| \tag{C.2.5}$$

$$\hat{T}\hat{T}^{\dagger} = \frac{1}{N}\sum\sum \exp\left(2\pi i \frac{k}{N}(j-j')\right)|j'\rangle\langle j| \tag{C.2.6}$$

$$= \sum_{j,j'}\left(\frac{1}{N}\sum_{k=0}\exp\left(2\pi i \frac{k}{N}(j-j')\right)\right)|j'\rangle\langle j| \tag{C.2.7}$$

$$= \sum_{j=0}^{N-1}|j\rangle\langle j'| = \mathbb{1}, \square. \tag{C.2.8}$$

□

Section C.3

QFT implementation on Circuit

CHAPTER

D

Fast Pauli Polynomial manipulation

Hamiltonian could be decomposed to linear combination of several local Hamiltonians.

$$H = \sum_i \lambda_i H_i \tag{D.0.1}$$

The common decomposition is using Pauli terms, $H_i = P_{j(i)} = \Pi_k \sigma_l(k)^m$. Since, Pauli terms form an orthonormal basis of the matrix space, the decomposition is well-defined on general n qubit Hilbert space with Hilbert-Schmidt inner product.

$$\lambda_i = \langle P_{j(i)} | H \rangle = \frac{1}{2^n} \operatorname{tr}\big(P_{j(i)} * H\big) \tag{D.0.2}$$

The Pauli term representation is called *Pauli-polynimial* of the given Hamiltonian H. However, in large n, since the matrix dimension increases with exponential complexity, the decomposition process requires huge computational resources in Hamiltonian analysis and Pauli-polynomial manipulation.

The time complexity of the matrix multiplication is $O(8^n)$, for n qubit Hilbert space. In addition, it is only about single Pauli term, so that the total coefficient cost is $O(32^n)$. It is because that we cannot know what terms are zero in the given Hamiltonian and we have to test the all Pauli-terms.

Moreover, constructing the original Hamiltonian with matrix form from the Pauli polynomial also arises in many situation. Basic method is constructing each Pauli matrices with tensor products of 2-dim Pauli matrices, $\sigma_i, i \in \{0, 1, 2, 3\}$ and calculating a linear combination of them. It is called *term-by-term* method. The complexity of term-by-term method rely on the complexity of single n-fold Pauli matrix term, $f(n)$, and the number of non-zero terms in the polynomial, k. Therefore, the complexity of the composition is $O(k(f(n)+4^n))$. There are 4^n number of Pauli terms, the worst case is $O(16^n + 4^n f(n))$.

Section D.1

Decomposition

Subsection D.1.1

Tensorized method

Subsection D.1.2

Term-by-Term methods

Section D.2

Composition

Subsection D.2.1
PauliComposer

Subsection D.2.2
Inverse of the Tensorized method

Since, the tensorized method is just a basis transformation, the inverse transformation is well defined, however, without the coefficient matrix of the Pauli-polynomial the inverse algorithm could not be used to construct the original Hamiltonian matrix. In the original paper by Hantzko et al [**?**], they didn't find the way so that only mentioned about the decomposition method.

$$\begin{array}{rcl} c_0 &=& \frac{1}{2}(A_{11}+A_{22}) \\ c_1 &=& \frac{1}{2}(A_{12}+A_{21}) \\ c_2 &=& \frac{i}{2}(A_{12}-A_{21}) \\ c_3 &=& \frac{1}{2}(A_{11}-A_{22}) \end{array} \tag{D.2.1}$$

.

In 2024, Kim found a way based on the XZ representation of the Paili terms. In addition, he showed that XZ representation is one type of coefficient matrix basis.

Theorem 19 For a given simplex representation, (n_x, n_z) of the given Paili term, P, their index, (i, j), in coefficient matrix is determined as

$$(i, j) = (n_z, n_x^{\wedge} n_z)$$

where, $^{\wedge}$ is a XOR bitwise operator.

Proof From i-th iteration of the TPD algorithm of 2^n dim square matrix, the unit sub-matrix dimension is 2^{n-i} and there are 4 block matrices, see Figure **??**. With Eq(**??**), the result matrix of $i-th$ iteration is

$$\begin{bmatrix} \sigma_0 \cdot \sigma_0 & \sigma_1 \cdot \sigma_0 \\ \sigma_1 \cdot \sigma_3 & \sigma_0 \cdot \sigma_3 \end{bmatrix} = \begin{bmatrix} 0_x \cdot 0_z & 1_x \cdot 0_z \\ 1_x \cdot 1_z & 0_x \cdot 1_z \end{bmatrix} \tag{D.2.2}$$

where, $0_z, 1_z, 0_x, 1_x$ are XZ binary representation of Pauli term of i-th decimal. For row index, $2^i * nz_i, nz_i \in \{0_z, 1_z\}$ the Z-binary determine the row index movement, if $nz_i = 1_z$, the row location is changed else it is not. For column index, the column index changed by $+0$ if $(1_x, 0_z)$ or $(0_x, 1_z)$, else $+2^{n-i}$ if $(0_x, 0_z)$ or $(1_x, 1_z)$. It is a simple XOR binary opertor, therby $2^{n-i} * nz_i^{\wedge} nx_i$, $nz_i \in \{0_z, 1_z\}, nx_i \in \{0_x, 1_x\}$

Thus, we have (i, j) coefficient index of XZ representation by iteration from 1-th to n-th.

$$\begin{array}{rcl} i = \sum_{k=0}^{n-1} 2^k nz_k & = & nz \\ j = \sum_{k=0}^{n-1} 2^k nz_k^{\wedge} nx_k & = & nz^{\wedge} nx \end{array} \tag{D.2.3}$$

where nz_k, nx_k are k-th binary element of nz, nx binary representation of the given Pauli term □.

The time complexity of the TPD is $O(8^n)$ and it is not different for single term and the full term polynomial. Comparing to Qiskit, and Pennylane methods we can observe the efficient of the routine in the worst case.

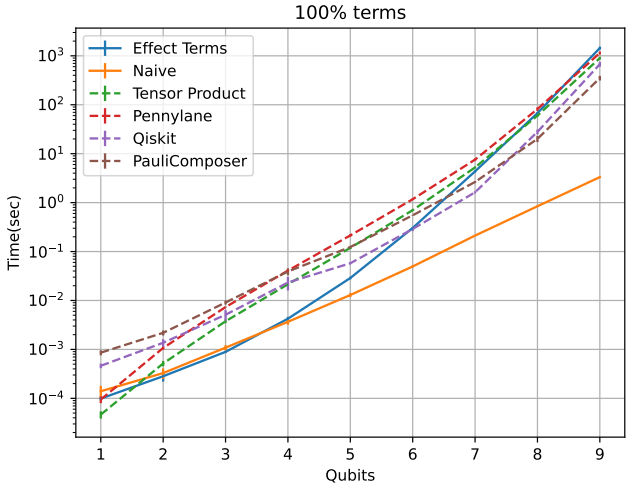


Figure D.1. Benchmarks for matrix composition of Puali polynomials with the algorithm 1, 2 with Qiskit, Pennylane, PauliComposer, and standard tensor product methods, for $n = 1$ to $n = 9$. The percentages of the each case represents how many coefficients are non-empty in 4^n number of spaces.

CHAPTER

E

Miscellaneous Mathematics

Section E.1

Tensor Product

Definition 10 **Tensor product** The given two vector space, V and W over the field $\mathbb{F}$, a tensor product is a bi-linear mapping with notation, $\otimes$, such that

$$\otimes : V \times W \to \mathcal{U} = V \otimes W \tag{E.1.1}$$

The generated space $\mathcal{U} = V \otimes W$ also be a vector space over field $\mathbb{F}$. The tensor product generates a larger vector space with two-given vector spaces.

Theorem 20 **Properties of Tensor producted space**

- Tensor product of the spanning sets of the each VS is a spanning set of the producted space.
- For finite VS $\dim(V) = n, \dim(W) = m$, $\dim(V \otimes W) = n \cdot m$.
- Dual space of the tensor producted space is a tensor product of dual spaces of each VS.

In matrix space, $\mathbf{M}(\mathbb{C})$ is form a Hilbert-space with Hilbert-Schmidt inner product.

Definition 11 **Hilbert-Schmidt Inner Product** For the given two matrices, A, B, their inner product is defined as

$$\langle A|B\rangle = \mathrm{Tr}(A^\dagger B) \tag{E.1.2}$$

A typical tensor product of matrix space is a *Kronecker product*.

Definition 12 **Kronecker Product**

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{pmatrix} \tag{E.1.3}$$

Section E.2

Tensor product representation of quantum circuit

Section E.3

Lie-algebra

Lie-group and Lie-algebra is a mathematical formulation to express the group sturcture in VS for a convience.

CHAPTER

F

NP-problems

Section F.1

Traveling Purchaser problem

Subsection F.1.1

Traveling Saleman problem

Section F.2

Max-clique Problem

Graph.

complment graph.

Additional topic: time evolution operator of Hamiltonian,

Index

Baker-Campbell-Hausdorff formula, 8
BCH-formula, 8

Dyson Series, 13

Fractal decomposition, 10

Hilbert-Schmidt inner product, 6

Kitaev
    Solovay-Kitaev theorem, 5

Lie, 5

Operator Exponential, 5

Pauli
    Polynomial, 14
Pauli matrices, 6
Product Formula, 10
    Lie-Trotter, 10
    Trotter, 10
    Trotter-Suzuki, 10

Solovay
    Solovay-Kitaev theorem, 5
Sophus Lie, 5