Mahn-Soo Choi (Korea University)

A Quantum Computation Workbook

May 26, 2021

Springer

Preface

The book will be published later this year, and obviously, this draft edition is very rough and far from complete for the moment.

N.B. This draft is provided for educational purposes with the permission of Springer. Redistribution of this draft is strictly prohibited.

Seoul, Korea
April 2021 *Mahn-Soo Choi*

Contents

1 The Postulates of Quantum Mechanics **9**
- 1.1 Quantum States . . . 10
  - 1.1.1 Pure States . . . 10
  - 1.1.2 Mixed States . . . 15
- 1.2 Time Evolution of Quantum States . . . 22
  - 1.2.1 Unitary Dynamics . . . 23
  - 1.2.2 Quantum Noisy Dynamics . . . 26
- 1.3 Measurements on Quantum States . . . 27
  - 1.3.1 Projection Measurements . . . 28
  - 1.3.2 Generalized Measurements . . . 31

2 Quantum Computation: Overview **37**
- 2.1 Single-Qubit Gates . . . 38
  - 2.1.1 Pauli Gates . . . 38
  - 2.1.2 Hadamard Gate . . . 42
  - 2.1.3 Rotations . . . 45
- 2.2 Two-Qubit Gates . . . 48
  - 2.2.1 CNOT, CZ, and SWAP . . . 48
  - 2.2.2 Controlled-U Gate . . . 58
  - 2.2.3 General Unitary Gate . . . 65
- 2.3 Multi-Qubit Controlled Gates . . . 73
  - 2.3.1 Gray Code . . . 73
  - 2.3.2 Multi-Qubit Controlled-NOT . . . 75
- 2.4 Universal Quantum Computation . . . 81
- 2.5 Measurements . . . 83

3 Virtual Realizations of Quantum Computers **89**
- 3.1 Quantum Bits . . . 90
- 3.2 Dynamical Scheme . . . 92
  - 3.2.1 Implementation of Single-Qubit Gates . . . 93
  - 3.2.2 Implementation of CNOT . . . 99
- 3.3 Geometric/Topological Scheme . . . 102
- 3.4 Measurement-Based Scheme . . . 107

3.4.1 Single-Qubit Rotations . . . 110
3.4.2 CNOT Gate . . . 113
3.4.3 Graph States . . . 114
3.5 Spin-Boson Model* . . . 116

4 Quantum Algorithms: Introduction 121
4.1 Quantum Teleportation . . . 122
4.1.1 Nonlocality in Entanglement . . . 122
4.1.2 Implementation of Quantum Teleportation . . . 124
4.2 Deutch-Jozsa Algorithm & Variants . . . 127
4.2.1 Quantum Oracle . . . 128
4.2.2 Deutsch-Jozsa Algorithm . . . 134
4.2.3 Bernstein-Vazirani Algorithm . . . 136
4.2.4 Simon's Algorithm . . . 137
4.3 Quantum Fourier Transform (QFT) . . . 141
4.3.1 Definition and Physical Meaning . . . 142
4.3.2 Quantum Implementation . . . 143
4.3.3 Semiclassical Implementation . . . 147
4.4 Quantum Phase Estimation (QPE) . . . 152
4.4.1 Definition . . . 152
4.4.2 Implementation . . . 153
4.4.3 Simulation of von Neumann Measurement . . . 156
4.5 Applications . . . 158
4.5.1 The Period-Finding Algorithm . . . 158
4.5.2 The Order-Finding Algorithm . . . 160

5 Decoherence 163
5.1 Quantum Operations . . . 164
5.1.1 The Kraus Representation . . . 165
5.1.2 Unitary Representation . . . 175
5.1.3 Examples . . . 176
5.2 Generalized Measurements as Quantum Operations . . . 179
5.3 Quantum Master Equation . . . 179
5.3.1 Derivation . . . 181
5.3.2 Examples . . . 183
5.3.3 Solution Methods . . . 185
5.4 Fidelity and Trace Distance . . . 192
5.5 Entanglement, Entropy, Mutual Information . . . 192

6 Quantum Error Correction Codes: Introduction 193
6.1 Discretization of errors . . . 193
6.2 9-Qubit Code (Shor's Code) . . . 193
6.3 Fault-Tolerant Quantum Computation . . . 193
6.4 CSS Code (Optional) . . . 193

6.5 Stabilizer Code (Optional) . . . 193
6.6 Surface Code (Optional) . . . 193

A Linear Algebra **195**
A.1 Vectors . . . 195
A.1.1 Vector Space . . . 195
A.1.2 Hermitian Product . . . 196
A.1.3 Basis . . . 197
A.1.4 Representations . . . 198
A.2 Linear Operators . . . 200
A.2.1 Linear Maps . . . 200
A.2.2 Representations . . . 201
A.2.3 Hermitian Conjugate of Operators . . . 202
A.3 Dirac's Bra-Ket Notation . . . 204
A.4 Spectral Theorems . . . 207
A.4.1 Spectral Decomposition . . . 207
A.4.2 Functions of Operators . . . 208
A.5 Tensor-Product Spaces . . . 210
A.5.1 Vectors in a Product Space . . . 210
A.5.2 Operators on a Product Space . . . 212

B Superoperators **215**
B.1 Operators as Vectors . . . 215
B.2 Superoperators . . . 221
B.2.1 Matrix Representation . . . 221
B.2.2 Operator-Sum Representation . . . 222
B.2.3 Choi Isomorphism . . . 225
B.3 Partial Trace . . . 227
B.4 Partial Transposition . . . 228

C Mathematica Application Q3 **231**
C.1 Installation . . . 231
C.2 Quick Start . . . 232

Bibliography **233**

Index **238**

Chapter 1

The Postulates of Quantum Mechanics

- April 20, 2021 (v1.9)

The great compilation "Elements" (see Figure 1.1) by Euclid of Alexandria in Ptolemaic Egypt circa 300 BC established a unique logical structure for mathematics, and every mathematical theory is built upon elementary axioms and definitions with propositions and proofs following. Theories in physics also take a similar structure. For example, classical mechanics is based on Sir Isaac Newton's three laws of motion. Called "laws", they are in fact elementary hypotheses, that is, axioms. While this is a remarkably different custom in physics compared with the mathematical counterpart, it should not be surprising to call assumptions as laws or principles because they do not only provide physical theories with logical foundation but also determine their fate whether to describe the Nature properly or have mere existence as an intellectual framework. After all, the true value of a physical science is to understand the Nature.

Embracing the wave-particle duality and the complementarity principle, quantum mechanics has been founded on the three fundamental postulates. The founders of quantum mechanics could be more ambitious to call them laws instead of plain postulates, but every single of them defies our intuition to such an extent that it sounds more natural. For an overview, here are the fundamental postulates of quantum mechanics:

Postulate 1 The *quantum state* of a system is completely described by a state vector in the Hilbert space associated with the system.

Postulate 2 The *time evolution* of a closed quantum system is governed by the Schrödinger equation.

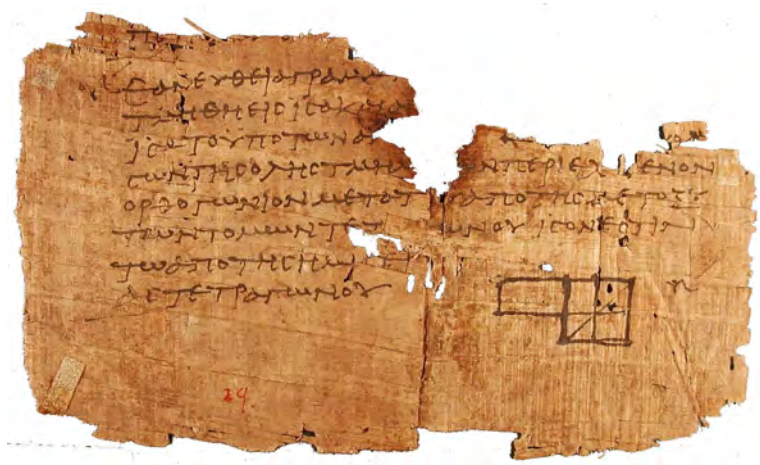


Figure 1.1: A fragment of Euclid's Elements on part of the Oxyrhynchus papyri located at the University of Pennsylvania. Courtesy of WikiMedia Commons.

Postulate 3 A physical quantity is described by an "observable"—a Hermitian operator. Upon the *measurement* of the quantity, the outcome is one of the eigenvalues of the observable and determined *probabilistically*. Right after the measurement, the state "collapses" to the eigenstate of the observable corresponding to the measurement outcome.

In the subsequent sections of this chapter, we will detail the physical aspects of each postulate and their relevance to quantum computing and quantum information.

1.1 Quantum States

The first postulate is about how to describe the *state* of a system mathematically. Recall that in classical mechanics, the state of a particle in motion is described by the simple values of its position and momentum (or, equivalently, velocity). In quantum mechanics, the description is formulated at two different levels depending on the physical situation.

1.1.1 Pure States

Postulate 1 The quantum state of a closed system is completely described by a state vector in the Hilbert space associated with the system.

The most common example of the state vector is a "wave function"—a member of the Hilbert space of square integrable functions—originally put forward by Schrödinger. Modern approaches associate an abstract vector space with the system and the specific characters of the particular system are reflected in the

choices of basis (see Appendix A.1). When the state is known exactly, the system is said to be in the *pure state*, and the above description in comprehensive. In many cases, however, it is difficult to know the state exactly. We thus need a more general description.

This postulate immediately raises a mind blowing question: What is the physical meaning of the state vector or its components in a given basis (or the wave function)? Quantum mechanics has never offered a direct physical meaning of the state vector. It was Born (1926) who proposed a partial resolution to the question and inspired the probabilistic interpretation of quantum mechanics as formulated in Postulate 3 concerning measurement. This work awarded him the 1954 Nobel Prize in Physics.

Postulate 1 leaves another baffling question: Given a physical system, there is no general prescription to figure out the Hilbert space associated with it. While it is rather technical, it is nevertheless an important and serious question when one encounters a new (or yet-to-be-understood) system and tries to describe it quantum mechanically.

For a *qubit*—an idealistic two-level quantum system—the Hilbert space is two dimensional. A basis of two logical states $|0\rangle$ and $|1\rangle$ is assumed and is called the *logical basis* of the qubit.

Consider a group of two-level quantum systems, indicated by the symbol `S`.

```
Let[Qubit, S]
```

Different qubits can be specified by the flavor indices, the last of which has a special meaning (see the documentation of `Qubit`).

```
In[•]:= {S[1, None], S[2, None]}
        S[{1, 2}, None]
Out[•]= {S1, S2}
Out[•]= {S1, S2}
```

The associated Hilbert space is two dimensional. For many functions dealing with qubits, the final index `None` can be dropped.

```
In[•]:= bs = Basis[S[1, None]]
        bs = Basis[S[1]]
Out[•]= {|␣⟩, |1S1⟩}
Out[•]= {|␣⟩, |1S1⟩}
```

For the efficiency reasons, the default value 0 of any qubit is removed from the data structure. For a more intuitively appealing form with all default values, `LogicalForm` can be used.

```
In[•]:= LogicalForm[bs]
Out[•]= {|0S1⟩, |1S1⟩}
```

Each state in the logical basis can also be specified manually.

```
In[•]:= vec = Ket[S[1] → 1, S[2] → 0];
        LogicalForm[vec, {S[1], S[2]}]
Out[•]= |1_{S_1} 0_{S_2}⟩
```

```
In[•]:= vec = Ket[{S[1], S[2]} → {1, 0}];
        LogicalForm[vec, {S[1], S[2]}]
Out[•]= |1_{S_1} 0_{S_2}⟩
```

A general quantum state of `S[1,None]` is a linear combination of the two basis states with two complex coefficients `c[0]` and `c[1]`.

```
In[•]:= Let[Complex, c]
        vec = Ket[S[1] → 0] × c[0] + Ket[S[1] → 1] × c[1];
        vec // LogicalForm
Out[•]= c_0 |0_{S_1}⟩ + c_1 |1_{S_1}⟩
```

A two-dimensional state vector is often visualized as a point (called the *Bloch vector*) on the *Bloch sphere*. The Bloch sphere is a geometrical representation of a two-dimensional vector space. Any state vector $|\psi\rangle$ is expanded in the logical basis as

$$|\psi\rangle = |0\rangle\,\psi_0 + |1\rangle\,\psi_1 \quad (\psi_0, \psi_1 \in \mathbb{C})\,. \tag{1.1}$$

The normalization condition, $|\psi_0|^2 + |\psi_1|^2 = 1$, tells us that the state vector can be expressed up to a global phase factor by

$$|\psi\rangle = |0\rangle\cos(\theta/2) + |1\rangle\sin(\theta/2)e^{i\phi} \tag{1.2}$$

with θ and ϕ specifying the relative magnitude and phase, respectively, of the expansion coefficients ($\theta, \phi \in \mathbb{R}$). The Bloch vector associated with the state vector $|\psi\rangle$ is defined by $\boldsymbol{b} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$. The Bloch vector can equivalently be obtained in terms of the expectation values of the Pauli operators, $\boldsymbol{b} = (\langle\hat{\sigma}^x\rangle, \langle\hat{\sigma}^y\rangle, \langle\hat{\sigma}^z\rangle)$. Indeed, with $|\psi\rangle$ in the form (1.2), one can show that $\langle\hat{\sigma}^x\rangle = \sin\theta\cos\phi$, $\langle\hat{\sigma}^y\rangle = \sin\theta\sin\phi$, and $\langle\hat{\sigma}^z\rangle = \cos\theta$. Therefore, any state vector in a two-dimensional Hilbert space corresponds uniquely (up to a global phase factor) to a point on the sphere of unit radius, the Bloch sphere.

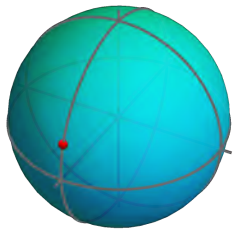
A two-dimensional pure state is represented by a point on the Bloch sphere. For example, consider a pure state.

```
In[•]:= vec = Ket[] × Sqrt[2] - I Ket[S[1] → 1];
        vec // LogicalForm
Out[•]= √2 |0_{S_1}⟩ - i |1_{S_1}⟩
```

This visualize the state vector on a Bloch sphere. `BlochVector` converts the state vector to a three-dimensional vector. `BlochSphere` is a shortcut for `Graphics3D` with an visualization of the Bloch sphere.

```
In[•]:= BlochSphere[{Red, Bead@BlochVector[vec]},
        ImageSize → Small]
```

Out[•]=



Many quantum systems like a system of many particles are composed of several parts with independent degrees of freedom. For such a system, the overall Hilbert space is built up from the Hilbert spaces of individual parts by means of the tensor product (see Appendix A.5). For example, consider a system of two parts and suppose that they are associated with the vector spaces $\mathcal{V}$ and $\mathcal{W}$, respectively. The total Hilbert space is given by the tensor product $\mathcal{V} \otimes \mathcal{W}$, which is defined to be the vector space spanned by the tensor-product basis

$$\{|v_i\rangle \otimes |w_j\rangle : i = 0, \ldots, m-1;\ j = 0, \ldots, n-1\}\,, \tag{1.3}$$

where $\{|v_i\rangle\}$ and $\{|w_j\rangle\}$ are bases of $\mathcal{V}$ and $\mathcal{W}$ of dimensions m and n, respectively. The dimension of $\mathcal{V} \otimes \mathcal{W}$ for the total system is obviously given by mn, and in general a state vector of the total system is a linear superposition consisting of mn terms

$$|\Psi\rangle = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |v_i\rangle \otimes |w_j\rangle\, \Psi_{ij}\,. \tag{1.4}$$

Some state vectors are factored into the form

$$(|v_1\rangle\, c_1 + |v_2\rangle\, c_2 + \cdots) \otimes (|w_1\rangle\, d_1 + |w_2\rangle\, d_2 + \cdots)\,. \tag{1.5}$$

Such a state is said to be *separable*. For example, the superposition of all the logical basis states of two qubits

$$|0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle \tag{1.6}$$

is factored into

$$(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle), \tag{1.7}$$

and is a separable state. On the other hand, the state

$$|0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle \tag{1.8}$$

can never be factored. Such a state is said to be *entangled*. The *Schmidt decomposition* is a systematic way to test whether a given quantum state $|\Psi\rangle$ in a tensor-product space is separable or not. The Schmidt decomposition—see Appendix A.5.1 for the mathematical details—is a method to choose proper bases

$\{|v_i'\rangle\}$ and $\{|w_j'\rangle\}$ of $\mathcal{V}$ and $\mathcal{W}$, respectively, to rewrite the given state vector $|\Psi\rangle$ in the least number of terms of the form

$$|\Psi\rangle = \sum_{k=0}^{R-1} |v_k'\rangle \otimes |w_k'\rangle \, s_k \,, \tag{1.9}$$

where the coefficients are all definitely positive, $s_k > 0$. Here the so-called *Schmidt rank* R of the quantum state $|\Psi\rangle$ cannot be larger than the smaller of m and n, $R \leq \min(m, n)$. If the Schmidt rank is 2 or larger, then the state is entangled.

Consider the following state of a two-qubit state.

```
In[•]:= ket = Ket[] + Ket[S[1] → 1] + Ket[S[2] → 1];
        ket // LogicalForm
Out[•]= |0_S1 0_S2⟩ + |0_S1 1_S2⟩ + |1_S1 0_S2⟩
```

This gives the Schmidt decomposition of the state. It turns out that its Schmidt rank is two and the state is an entangled state.

```
In[•]:= {ww, uu, vv} = SchmidtDecomposition[ket, S[1], S[2]];
        ww
Out[•]= {Sqrt[1/2 (3 + Sqrt[5])], Sqrt[1/2 (3 - Sqrt[5])]}
```

`SchmidtForm` presents the Schmidt decomposition in a more intuitively-appealing form. For a thorough analysis of the result, use `SchmidtDecomposition`.

```
In[•]:= new = SchmidtForm[ket, {S[1]}, {S[2]}]
```

$$\sqrt{\frac{1}{2}\left(3-\sqrt{5}\right)}\left(\frac{\left(1+\frac{1}{2}\left(1-\sqrt{5}\right)\right)\left|0_{S_1}\right\rangle}{\sqrt{\frac{1}{4}\left(1-\sqrt{5}\right)^2+\left(1+\frac{1}{2}\left(1-\sqrt{5}\right)\right)^2}}+\frac{\left(1-\sqrt{5}\right)\left|1_{S_1}\right\rangle}{2\sqrt{\frac{1}{4}\left(1-\sqrt{5}\right)^2+\left(1+\frac{1}{2}\left(1-\sqrt{5}\right)\right)^2}}\right)\otimes\left(\frac{\left|0_{S_2}\right\rangle}{2\sqrt{1+\frac{1}{4}\left(1-\sqrt{5}\right)^2}}+\frac{\left|1_{S_2}\right\rangle}{\sqrt{1+\frac{1}{4}\left(1-\sqrt{5}\right)^2}}\right)+\sqrt{\frac{1}{2}\left(3+\sqrt{5}\right)}\left(\frac{\left(1+\frac{1}{2}\left(1+\sqrt{5}\right)\right)\left|0_{S_1}\right\rangle}{\sqrt{\frac{1}{4}\left(1+\sqrt{5}\right)^2+\left(1+\frac{1}{2}\left(1+\sqrt{5}\right)\right)^2}}+\frac{\left(1+\sqrt{5}\right)\left|1_{S_1}\right\rangle}{2\sqrt{\frac{1}{4}\left(1+\sqrt{5}\right)^2+\left(1+\frac{1}{2}\left(1+\sqrt{5}\right)\right)^2}}\right)\otimes\left(\frac{\left|0_{S_2}\right\rangle}{2\sqrt{1+\frac{1}{4}\left(1+\sqrt{5}\right)^2}}+\frac{\left|1_{S_2}\right\rangle}{\sqrt{1+\frac{1}{4}\left(1+\sqrt{5}\right)^2}}\right)$$

The Schmidt decomposition is incredibly complicated for such a simple-looking system. Let us take an approximation to get an impression of how the state is entangled.

```
In[•]:= new // N // Simplify
Out[•]= 0.618034 (0.525731 |0_S1⟩ - 0.850651 |1_S1⟩) ⊗ (-0.525731 |0_S2⟩ + 0.850651 |1_S2⟩) +
          1.61803 (0.850651 |0_S1⟩ + 0.525731 |1_S1⟩) ⊗ (0.850651 |0_S2⟩ + 0.525731 |1_S2⟩)
```

As pointed out for the first time by Einstein *et al.* (1935), entangled quantum states have many intriguing properties that are difficult to understand intuitively and have raised many questions concerning the foundation of quantum mechanics. The non-local property—the very property pointed out by Einstein *et al.* (1935)—is the representative property of entangled states, and illustrated in Section 4.1.1. Quantum entanglement also provides a fundamental explanation for quantum decoherence—the process where the quantum system loses the quantum effects—as is discussed in Chapter 5. In quantum computation, quantum information and quantum communication, quantum entanglement is regarded as a valuable resource that enables many amazing quantum effects such quantum speed-up and unconditional security which are impossible in classical world. One of the most illustrative example—the quantum teleportation—is discussed in Section 4.1.

1.1.2 Mixed States

One often encounters a situation where the state of a system is not known completely. Common example is the case where the system is interacting with its surroundings. In this case, the system is said to be in a *mixed state*. A mixed state is a statistical mixture of pure states and characterized in terms of statistical ensemble, where different state vectors $|\psi_\mu\rangle$ are found with probabilities p_μ.[1] Such an ensemble is efficiently represented by a *density operator* (often just called *density matrix* for historical reasons) which is constructed as

$$\hat{\rho} = \sum_\mu |\psi_\mu\rangle \, p_\mu \, \langle\psi_\mu| \tag{1.10}$$

Sometimes, it is convenient to describe the statistical mixture in terms of a set of *unnormalized* vectors $\{|\psi'_j\rangle := |\psi_\mu\rangle \sqrt{p_\mu}\}$ absorbing the probabilities into the vectors—$\langle\psi'_\mu|\psi'_\mu\rangle$ gives the probability to find the state $|\psi'_\mu\rangle$ in the ensemble. The corresponding density operator is constructed as

$$\hat{\rho} = \sum_\mu |\psi'_\mu\rangle \, \langle\psi'_\mu| \ , \tag{1.10′}$$

and is certainly equivalent to the one in Eq. (1.10).

Here is an example of the density matrix for a statistical mixture of two pure states.

[1] Here, note that the different pure states $|\psi_\mu\rangle$ in the mixture do not have to be orthogonal to each other. They do not span the Hilbert space, either, in general. They are completely general.

```
In[•]:= vecs = {
          v = Ket[],
          w = (Ket[] - I Ket[S[1] → 1]) / Sqrt[2]
        };
      vecs // LogicalForm
      probs = {1 / 3, 2 / 3}
```

Out[•]= $\left\{\left|0_{S_1}\right\rangle, \frac{\left|0_{S_1}\right\rangle - i\left|1_{S_1}\right\rangle}{\sqrt{2}}\right\}$

Out[•]= $\left\{\frac{1}{3}, \frac{2}{3}\right\}$

From the specifications of the ensemble, this constructs the density operator for the mixed state.

```
In[•]:= ρ = (vecs ** Dagger[vecs]).probs // Garner;
      ρ // LogicalForm
```

Out[•]= $\frac{2}{3}\left|0_{S_1}\right\rangle\left\langle 0_{S_1}\right| + \frac{1}{3} i \left|0_{S_1}\right\rangle\left\langle 1_{S_1}\right| - \frac{1}{3} i \left|1_{S_1}\right\rangle\left\langle 0_{S_1}\right| + \frac{1}{3}\left|1_{S_1}\right\rangle\left\langle 1_{S_1}\right|$

This gives the matrix representation -- the "density matrix" -- of the density operator In the logical basis.

```
In[•]:= Matrix@ρ // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{2}{3} & \frac{i}{3} \\ -\frac{i}{3} & \frac{1}{3} \end{pmatrix}$$

This gives the expression of the density operator in terms of the Pauli operators.

```
In[•]:= Elaborate@ExpressionFor[Matrix@ρ, S[1]]
```

Out[•]= $\frac{1}{2} - \frac{S_1^y}{3} + \frac{S_1^z}{6}$

It is interesting to note that the density operator and the relevant physical properties do not depend on all the details in the specification of the statistical ensemble, which in some sense makes the description of mixed states in terms of density operator powerful and efficient: Supppose that two statistical ensembles be specified by the sets $\{|\alpha_\mu\rangle : \mu = 1, \ldots, m\}$ and $\{|\beta_\nu\rangle : j = \mu, \ldots, n\}$, respectively, of unnormalized vectors as in Eq. (1.10′). Without loss of generality, assume that $m \leq n$. Then, the density operators describing the ensembles are identical, that is,

$$\sum_{\mu=1}^{m} |\alpha_\mu\rangle \langle\alpha_\mu| = \sum_{\nu=1}^{n} |\beta_\nu\rangle \langle\beta_\nu| \tag{1.11}$$

if and only if there exists an $n \times n$ unitary matrix $U_{\mu\nu}$ such that

$$|\beta_\nu\rangle = \sum_{\mu} |\alpha_\mu\rangle U_{\mu\nu} \tag{1.12}$$

for all $\nu = 1, \ldots, n$. Although providing rigorous proofs is out of the main scope of the book, the impact of the unitary freedom—or ambiguity—in the specification of the mixed states is wide spread, and similar unitary freedom is observed in the

decription of decoherence and related effects (Section 5). It is thus heuristic to take a moment here to prove it.

It is clear that if the states from the two sets satisfy the relation (1.12), then the identity (1.11) holds. To see the converse, suppose that the two density operators are the same and equal to $\hat{\rho}$. Write $\hat{\rho}$ in a spectral decomposition (Appendix A.4),

$$\hat{\rho} = \sum_{\lambda} |\lambda\rangle \langle\lambda| \,, \tag{1.13}$$

where $|\lambda\rangle$ are the (unnormalized) eigenstates of $\hat{\rho}$ belonging to non-zero eigenvalues λ. We have used the properties (Theorem 19) of positive operators and normalized $|\lambda\rangle$ by their own eigenvalues, that is, $\langle\lambda|\lambda\rangle = \lambda$. We first note that $\{|\alpha_\mu\rangle\}$ and $\{|\lambda\rangle\}$ span the same subspace, and hence $|\alpha_\mu\rangle$ can be expanded in $|\lambda\rangle$,

$$|\alpha_\mu\rangle = \sum_{\lambda} |\lambda\rangle V_{\lambda\mu}. \tag{1.14}$$

Putting it into (1.11), we require that

$$\sum_{\lambda\lambda'} |\lambda\rangle \langle\lambda'| \sum_{\mu} V_{\lambda\mu} V^*_{\lambda'\mu} = \sum_{\lambda} |\lambda\rangle \langle\lambda| \,. \tag{1.15}$$

Recall that unlike $|\alpha_\mu\rangle$, which are not orthogonal to each other in general, $|\lambda\rangle$ are orthogonal (Appendix A.4). Equation (1.15) thus implies that the rows of the matrix V are orthogonal to each other. By adding additional rows, if necessary, that are orthogonal to existing rows, the matrix V can be extended into a unitary matrix. For the same reason, $|\beta_\nu\rangle$ can be expanded as

$$|\beta_\nu\rangle = \sum_{\lambda} |\lambda\rangle W_{\lambda\nu} \,, \tag{1.16}$$

and the matrix $W_{\lambda\nu}$ can be extended into a unitary matrix. Overall, $|\alpha_\mu\rangle$ and $|\beta_\nu\rangle$ satisfy the relation (1.12), where $U = W^\dagger V$.[2] Finally, here is a demonstration of such freedom:

Let us consider an example. Consider a statistical mixture of the following three pure states.

```
In[•]:= v1 = Ket[];
        v2 = (Ket[] + I Ket[S → 1]) / Sqrt[2];
        v3 = (Ket[] 2 + Ket[S → 1] I) / Sqrt[5];
        LogicalForm@{v1, v2, v3}
```

$$\text{Out[•]=}\ \left\{ |0_S\rangle,\ \frac{|0_S\rangle + i\,|1_S\rangle}{\sqrt{2}},\ \frac{2\,|0_S\rangle + i\,|1_S\rangle}{\sqrt{5}} \right\}$$

These are the associated probabilities.

[2] In this proof, we have implicitly assumed that $m = n$. However, the proof can be extended easily by adding null vectors in the set $\{\alpha_\mu\}$ until $m = n$.

```
In[•]:= p1 = 1 / 8;
        p2 = 1 / 4;
        p3 = 5 / 8;
        {p1, p2, p3}
Out[•]= {1/8, 1/4, 5/8}
```

The mixed state is described by the density operator.

```
In[•]:= ρ = Total@Multiply[{v1, v2, v3}, {p1, p2, p3}, Dagger@{v1, v2, v3}];
        ρ // LogicalForm
Out[•]= 3 |0_S⟩ ⟨0_S| / 4 - 3/8 i |0_S⟩ ⟨1_S| + 3/8 i |1_S⟩ ⟨0_S| + |1_S⟩ ⟨1_S| / 4
```

Next consider another set of pure states.

```
In[•]:= w1 = (Ket[] 2 + Ket[S → 1] I) / Sqrt[5];
        w2 = Ket[S → 1];
        LogicalForm@{w1, w2}
Out[•]= {(2 |0_S⟩ + i |1_S⟩)/√5, |1_S⟩}
```

The associated probabilities are as following.

```
In[•]:= q1 = 15 / 16;
        q2 = 1 / 16;
        {q1, q2}
Out[•]= {15/16, 1/16}
```

The mixture leads to the same density operator.

```
In[•]:= σ = Total@Multiply[{w1, w2}, {q1, q2}, Dagger@{w1, w2}]
Out[•]= 3 |␣⟩ ⟨␣| / 4 - 3 i |␣⟩ ⟨1_S| / 8 + 3 i |1_S⟩ ⟨␣| / 8 + |1_S⟩ ⟨1_S| / 4
```

The two sets are related by the following unitary matrix

```
In[•]:= U = Topple@{
            {1, 1, 2} / Sqrt[6],
            {1, -1, 0} / Sqrt[2],
            {1, 1, -1} / Sqrt[3]
           };
        U // MatrixForm
Out[•]//MatrixForm=
        ⎛ 1/√6    1/√2    1/√3  ⎞
        ⎜ 1/√6   -1/√2    1/√3  ⎟
        ⎝ √(2/3)   0     -1/√3  ⎠
```

Recall that two-dimensional pure states are visualized *on* a Bloch sphere. A mixed state $\hat{\rho}$ for a qubit can be visualized similarly, but in general the representing point resides *inside* the Bloch sphere: As for a pure state, the Bloch vector $\boldsymbol{b}$ corresponding to a mixed state $\hat{\rho}$ is defined by $\boldsymbol{b} = (\langle\hat{\sigma}^x\rangle, \langle\hat{\sigma}^y\rangle, \langle\hat{\sigma}^z\rangle)$. Recalling

that any operator on a two-dimensional vector space is a linear superposion of the Pauli operators, we decompose a density operator into the form

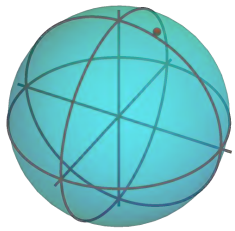
$$\hat{\rho} = \frac{1}{2}\left(\hat{\sigma}^0 + x\hat{\sigma}^x + y\hat{\sigma}^y + z\hat{\sigma}^z\right) \tag{1.17}$$

with $x, y, z \in \mathbb{R}$. By evaluating the averages, $\langle\hat{\sigma}^\mu\rangle = \mathrm{Tr}\,\hat{\rho}\hat{\sigma}^\mu$, of the Pauli operators with respect to $\hat{\rho}$, one can see that the Bloch vector corresponding to $\hat{\rho}$ is just given by $\boldsymbol{b} = (x, y, z)$. Clearly, $\sqrt{x^2 + y^2 + z^2} \leq 1$, and it may lie inside the Bloch sphere in general.

This gives a visualization of the mixed state by a point -- Bloch vector -- in a Bloch sphere.

```
In[•]:= BlochSphere[{Red, Bead@BlochVector@ρ}, "Opacity" → 0.4, ImageSize → Small]
```

Out[•]=



By construction, any density operator $\hat{\rho}$ is Hermitian, $\hat{\rho}^\dagger = \hat{\rho}$. Obviously, a density operator is an operator acting on the state vectors in the vector space $\mathcal{H}$. However, itself should be regarded as a vector in the vector space $\mathcal{L}(\mathcal{H})$ of all linear operators on $\mathcal{H}$ (see Appendix B.1).

Each diagonal element $\rho_{jj} := \langle v_j|\,\hat{\rho}\,|v_j\rangle$ in a given basis $\{|v_j\rangle\}$ carries an important physical meaning, that is, the probability to find the system in the basis state $|v_j\rangle$. This can be seen by noting that the probability to find the system in $|v_j\rangle$ is $|\,\langle v_j|\psi_\mu\rangle\,|^2$ under the condition that the state is $|\psi_\mu\rangle$, and the latter has the chance of probability p_μ. As the above argument holds for any arbitrary basis, it implies that any density operator $\hat{\rho}$ is not merely hermitian but also *positive* (Definition 12) with unit trace: $0 \leq \hat{\rho} \leq 1$ (i.e., any eigenvalue of $\hat{\rho}$ lies between 0 and 1) and $\mathrm{Tr}\,\hat{\rho} = 1$.

Off-diagonal elements of a density operator are responsible for coherence effects as we will observe in various interference experiments later. It is important to note here that coherence is a basis-dependent effect.

Exercise 1 Consider a density operator $\hat{\rho}$ of the general form in (1.10). Recall that as noted in Footnote 1, the pure states $|\psi_\mu\rangle$ in the mixture are completely arbitrary. Prove that the eigenvalues of $\hat{\rho}$ are all non-negative.

A mixed state arises naturally when the system interacts with its environment: As a closed system, the total system is described by a pure state $|\Psi\rangle \in \mathcal{H}\otimes\mathcal{E}$, where $\mathcal{H}$ and $\mathcal{E}$ are the Hilbert spaces associated with the system and the environment,

respectively. In accordance with the *Schmidt decomposition* (see Appendix A.5), it can be written as

$$|\Psi\rangle = \sum_{j=1}^{R} |\alpha_j\rangle \otimes |\beta_j\rangle \sqrt{p_j}\,, \tag{1.18}$$

where $|\alpha_j\rangle \in \mathcal{H}$, $|\beta_j\rangle \in \mathcal{E}$, $0 < p_j < 1$, and R is the *Schmidt rank* of $|\Psi\rangle$. Without access to the environment, one cannot tell in which state among $|\alpha_j\rangle$ the system is. One can only tell the chances. The probability for the system to be found in $|\alpha_j\rangle$ is equal to p_j. Therefore, the density operator that describes the situation is given by

$$\hat{\rho} = \sum_j |\alpha_j\rangle \langle\alpha_j|\, p_j \tag{1.19}$$

Now noting that $\langle\beta_i|\beta_j\rangle = \delta_{ij}$ and

$$|\Psi\rangle \langle\Psi| = \sum_{ij} |\alpha_i\rangle \langle\alpha_j| \otimes |\beta_i\rangle \langle\beta_j| \sqrt{p_i p_j}\,, \tag{1.20}$$

we can see that the expression in (1.19) is equivalent to taking a partial trace over $\mathcal{E}$ (see Appendix B.3),

$$\hat{\rho} = \operatorname*{Tr}_{\mathcal{E}} |\Psi\rangle \langle\Psi| \,. \tag{1.21}$$

In this sense, taking a partial trace over a particular part of the total system corresponds physically to "ignoring" that part.

Suppose that two qubits are coupled and that the total system is in the following state. We can regard the first qubit as the "system" and the second as the "reservoir".

```
In[•]:= total = (Ket[] - Ket[S[1] → 1] + Ket[S[2] → 1]) / Sqrt[3];
        LogicalForm[total]
```

$$\text{Out[•]=} \quad \frac{\left|0_{S_1} 0_{S_2}\right\rangle + \left|0_{S_1} 1_{S_2}\right\rangle - \left|1_{S_1} 0_{S_2}\right\rangle}{\sqrt{3}}$$

The first qubit is in a mixed state. The density operator is given by the partial trace over the second qubit.

```
In[•]:= ρ = Elaborate@PartialTrace[total ** Dagger[total], S[2]]
```

$$\text{Out[•]=} \quad \frac{1}{2} - \frac{S_1^x}{3} + \frac{S_1^z}{6}$$

This is the matrix representation of the density operator in the logical basis.

```
In[•]:= MatrixForm@Matrix@ρ
```

$$\text{Out[•]//MatrixForm=} \quad \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

A simple yet important aspect of the above observation is that when $|\Psi\rangle$ of the total system is a separable state, $|\Psi\rangle = |\alpha\rangle \otimes |\beta\rangle$, the reduced state

$$\hat{\rho} = \sum_{\mathcal{E}} |\alpha\rangle \langle\alpha| \otimes |\beta\rangle \langle\beta| = |\alpha\rangle \langle\alpha| \tag{1.22}$$

is a pure state, retaining full coherence. It means that the *entanglement* between the system and the environment is responsible of decoherence. In the history of quantum physics, it was a mystery why quantum effects can only be observable in microscopic systems but not in macroscopic systems. It was entanglement that has provided a key clue to the resolution of the mystery. We will see later entanglement also provides valuable resources for quantum information processing and quantum communications; see, e.g., Section 4.1.

When the system is in a pure state $|\psi\rangle$, the density operator is simply given by $\hat{\rho} = |\psi\rangle \langle\psi|$. For a pure state $|\psi\rangle = |0\rangle c_0 + |1\rangle c_1$, the matrix representation of $\hat{\rho}$ is given by

$$\hat{\rho} \doteq \begin{bmatrix} c_0 c_0^* & c_0 c_1^* \\ c_1 c_0^* & c_1 c_1^* \end{bmatrix}. \tag{1.23}$$

Given a density operator, there is a simple test whether it is a pure state or a mixed state.

Exercise 2 Show that a density operator $\hat{\rho}$ is a pure state if and only if $\mathrm{Tr}\,\hat{\rho}^2 = 1$. That is, for a mixed state $\mathrm{Tr}\,\hat{\rho}^2$ is strictly smaller than 1.

Let us examine the density operator corresponding to a pure state. As an example, consider the following pure state.

```
In[•]:= vec = Ket[S[1] → 0] × c[0] + Ket[S[1] → 1] × c[1];
        vec // LogicalForm
Out[•]= c0 |0S1⟩ + c1 |1S1⟩
```

This gives the density operator corresponding to the pure state.

```
In[•]:= ρ = vec ** Dagger[vec];
        ρ // LogicalForm
Out[•]= c0 c0* |0S1⟩ ⟨0S1| + c0 c1* |0S1⟩ ⟨1S1| + c1 c0* |1S1⟩ ⟨0S1| + c1 c1* |1S1⟩ ⟨1S1|
```

The matrix representation of the density operator gives the typical form of the density matrix for a pure state.

```
In[•]:= mat = Matrix[ρ];
        mat // MatrixForm
Out[•]//MatrixForm=
        ( c0 c0*   c0 c1* )
        ( c1 c0*   c1 c1* )
```

This illustrates that $\mathrm{Tr}[\rho^2] = 1$ for pure states.

```
In[•]:= Tr[mat.mat] // Simplify // MatrixForm
Out[•]//MatrixForm=
        (c0 c0* + c1 c1*)^2
```

A far more general way to characterize a mixed state $\hat{\rho}$ is the *von Neumann entropy*. The von Neumann entropy of a density operator $\hat{\rho}$ is defined by

$$S(\hat{\rho}) := -\operatorname{Tr} \hat{\rho} \log_2 \hat{\rho} = -\sum_j \lambda_j \log_2 \lambda_j \,, \tag{1.24}$$

where λ_j are the non-vanishing eigenvalues of $\hat{\rho}$.

Exercise 3 (a) Show that the von Neumann entropy of any pure state is zero.

(b) Show that

$$0 \le S(\hat{\rho}) \le \log_2 N \,, \tag{1.25}$$

where N is the Hilbert space dimension. In particular, for an n-qubit system $S(\hat{\rho}) \le n$.

Exercise 4 Let $|\Psi\rangle$ be a vector in $\mathcal{V} \otimes \mathcal{E}$, and $\hat{\rho} := \operatorname{Tr}_{\mathcal{E}} |\Psi\rangle \langle\Psi|$. Show that

$$S(\hat{\rho}) \le \log_2 R \,, \tag{1.26}$$

where R is the *Schmidt rank* of $|\Psi\rangle$.

As a vector describing a mixed state, one can ask whether a density operator is separable or not. Consider a system consisting of two subsystems A and B. A density operator $\hat{\rho}$ (and the associated mixed state) is said to be separable if it can be written as a *convex linear* superposition

$$\hat{\rho} = \sum_j \hat{\sigma}_j \otimes \hat{\tau}_j \, p_j \,, \quad 0 \le p_j \le 1 \,, \quad \sum_j p_j = 1 \,, \tag{1.27}$$

where $\hat{\sigma}_j$ and $\hat{\tau}_j$ are the density operators of the subsystems A and B. Unlike pure states, for which the Schmidt decomposition provides a simple test of entanglement (see Appendix A.5), it is hard in general and remains an open question to tell whether a given mixed state is separable or entangled. One might be tempted to use the Schmidt-like decomposition (A.57) of operators for the test of mixed-state entanglement. However, the operators $\hat{A}_\mu$ and $\hat{B}_\mu$ in the sum are not guaranteed to be density operators.

1.2 Time Evolution of Quantum States

The state of a classical system changes with time in accordance with Newton's second law of motion, that is, the evolution is governed by the celebrated equation of motion, $F = ma$, by Newton. In quantum mechanics, Newton's equation of motion is replaced with Schrödinger's.

1.2.1 Unitary Dynamics

Postulate 2 The time evolution of the state $|\psi\rangle$ of a *closed* quantum system is governed by the Schrödinger equation

$$i\hbar\frac{d}{dt}|\psi\rangle = \hat{H}|\psi\rangle \ , \tag{1.28}$$

where $\hbar$ is the Planck constant and $\hat{H}$ is the Hamiltonian of the system.

Throughout the book, we will use a unit system where $\hbar = 1$. In such a unit system, energy and frequency have the same dimension, the inverse of the dimension of time. The Hamiltonian is a Hermitian operator, physically, describing the energy of the system. In general, it is difficult to find the precise Hamiltonian for a particular system. In most cases, a model Hamiltonian is constructed and tested against experimental observations. If desired, it is corrected by changing the existing terms or including additional terms, and tested again.

Postulate 2 expresses the time evolution in terms of a differential equation. An equivalent way is to describe it by means of unitary transformation: Suppose that the system is initially in a state $|\psi(0)\rangle$, then the state $|\psi(t)\rangle$ at later time $t > 0$ is related to the initial state $|\psi(0)\rangle$ by a unitary operator $\hat{U}(t)$

$$|\psi(t)\rangle = \hat{U}(t)|\psi(0)\rangle \, . \tag{1.29}$$

The unitary operator $\hat{U}(t)$ is called the *time-evolution operator*. When the Hamiltonian $\hat{H}$ of the system is independent of time, the time-evolution operator is given by (recall that we have put $\hbar = 1$)

$$\hat{U}(t) = \exp[-it\hat{H}] \, . \tag{1.30}$$

The best way to evaluate the exponential function of a normal operator is to use its *spectral decomposition* (Appendix A.3). With the eigenstates $|E\rangle$ and corresponding eigenvalues E of $\hat{H}$, the Hamiltonian itself reads as $\hat{H} = \sum_E |E\rangle\langle E| \, E$ and hence its exponential function is given by $\exp(-it\hat{H}) = \sum_E |E\rangle\langle E| \, e^{-itE}$. In general, especially when the system is driven externally, for example, to actively process quantum information, the Hamiltonian depends on time. In this case, the relation between the time-evolution operator and the Hamiltonian is more complicated and will be discussed later.

Consider a two - level quantum state, denoted by the symbol S . Some real parameters will be denoted by the symbol B .

```
Let[Qubit, S]
Let[Real, B]
```

A time-independent Hamiltonian can be expressed in terms of the Pauli operators.

```
In[•]:= H = B[0] + S[1] × B[1] + S[2] × B[2] + S[3] × B[3]
Out[•]= B0 + B1 S^x + B2 S^y + B3 S^z
```

In this case, the time-evolution operator is given by

```
In[•]:= U[t_] = MultiplyExp[-I t H]
Out[•]= e^(-i t (B0 + B1 S^x + B2 S^y + B3 S^z))
```

The exponential function of operators can be evaluated by means of the spectral decomposition. Q3 has an internal mechanism to facilitate the spectral decomposition method. It is implemented through the function **Elaborate**.

```
In[•]:= Elaborate[U[t]] // ExpToTrig // Garner
```

$$\text{Out}[\bullet]= \cos\left[t\sqrt{B_1^2+B_2^2+B_3^2}\right]\,(\cos[t\,B_0]-i\,\sin[t\,B_0])-\frac{i\,B_3\,S^z\,(\cos[t\,B_0]-i\,\sin[t\,B_0])\,\sin\left[t\sqrt{B_1^2+B_2^2+B_3^2}\right]}{\sqrt{B_1^2+B_2^2+B_3^2}}-\frac{i\,(B_1-i\,B_2)\,S^+\,(\cos[t\,B_0]-i\,\sin[t\,B_0])\,\sin\left[t\sqrt{B_1^2+B_2^2+B_3^2}\right]}{\sqrt{B_1^2+B_2^2+B_3^2}}+\frac{(-i\,B_1+B_2)\,S^-\,(\cos[t\,B_0]-i\,\sin[t\,B_0])\,\sin\left[t\sqrt{B_1^2+B_2^2+B_3^2}\right]}{\sqrt{B_1^2+B_2^2+B_3^2}}$$

For simplicity, consider the following specific case. We have assumed a ceratin choice of units.

```
B[0] = 0; B[1] = B[3] = 1; B[2] = -1;
```

```
In[•]:= op[t_] = Elaborate[U[t]] // ExpToTrig // Garner
```

$$\text{Out}[\bullet]= \cos\left[\sqrt{3}\,t\right]-\frac{i\,S^z\,\sin\left[\sqrt{3}\,t\right]}{\sqrt{3}}+\frac{(1-i)\,S^+\,\sin\left[\sqrt{3}\,t\right]}{\sqrt{3}}-\frac{(1+i)\,S^-\,\sin\left[\sqrt{3}\,t\right]}{\sqrt{3}}$$

Suppose that the initial state is the eigenstate of the Pauli X operator, here denoted by **S[1]**.

```
In[•]:= v0 = (Ket[] + Ket[S → 1]) / Sqrt[2];
        v0 // LogicalForm
```

$$\text{Out}[\bullet]= \frac{|0_S\rangle+|1_S\rangle}{\sqrt{2}}$$

This is the state vector at a later time **t>0**.

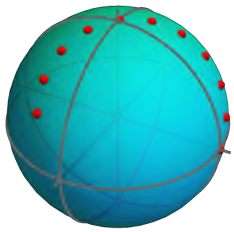
```
In[•]:= vec[t_] = op[t] ** v0;
        vec[t] // LogicalForm
```

$$\text{Out}[\bullet]= |1_S\rangle\left(\frac{\cos\left[\sqrt{3}\,t\right]}{\sqrt{2}}-\frac{\sin\left[\sqrt{3}\,t\right]}{\sqrt{6}}\right)+|0_S\rangle\left(\frac{\cos\left[\sqrt{3}\,t\right]}{\sqrt{2}}+\frac{(1-2\,i)\,\sin\left[\sqrt{3}\,t\right]}{\sqrt{6}}\right)$$

This visualizes the evolution of the state under the Hamiltonian on the Bloch sphere.

```
In[•]:= vv = Bead /@ BlochVector /@ Table[vec[t], {t, 0, 1, 0.1}] // Chop;
        BlochSphere[{Red, vv}, ImageSize → Small]
```

Out[•]=



An important point to bear in mind about unitary dynamics is that it does not depend on the history (for a time-independent Hamiltonian). This is reflected in the obvious property of the time-evolution operator,

$$\hat{U}(t+t') = \hat{U}(t)\hat{U}(t') \tag{1.31}$$

for any t, t'. Physically, it implies that the evolution depends only on the duration of time, but not on when it starts or ends. In this respect, it is also natural for a time-evolution operator to have the property $\hat{U}^\dagger(t) = \hat{U}(-t)$.

So far, we have discussed the time evolution of a pure state. What if the system is initially in a mixed state $\hat{\rho}(0)$ for a certain reason? As a statistical mixture of pure states, the mixed state can be expressed as

$$\hat{\rho}(0) = \sum_j |\psi_j(0)\rangle \langle\psi_j(0)| \, p_j \tag{1.32}$$

with $0 \le p_j \le 1$. If the system is closed, then each pure state $|\psi(0)\rangle$ evolves into $\hat{U}(t)|\psi(0)\rangle$. Overall, the state $\hat{\rho}(t)$ at later time t is given by

$$\hat{\rho}(t) = \hat{U}(t)\hat{\rho}(0)\hat{U}^\dagger(t). \tag{1.33}$$

In short, as long as the system remains closed later on, the dynamics is still unitary regardless of whether the system is prepared initially in a pure or mixed state.

Let us consider the same system and Hamiltonian. However, the initial state is now a mixed state.

```
In[•]:= ρ0 = v0 ** Dagger[v0] * 3 / 4 + Ket[] ** Bra[] ** 1 / 4 // LogicalForm // Garner
```

Out[•]=

$$\frac{5\,|0_s\rangle\langle 0_s|}{8} + \frac{3\,|0_s\rangle\langle 1_s|}{8} + \frac{3\,|1_s\rangle\langle 0_s|}{8} + \frac{3\,|1_s\rangle\langle 1_s|}{8}$$

Its matrix representation in the logical basis is given by the following.

```
In[•]:= mat = Matrix[ρ0];
        mat // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{5}{8} & \frac{3}{8} \\ \frac{3}{8} & \frac{3}{8} \end{pmatrix}$$

This is the state vector at a later time `t>0`.

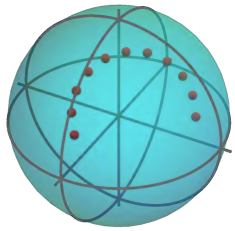
```
In[•]:= Let[Real, t]
        rho[t_] = op[t] ** ρ0 ** Dagger[op[t]];
        rho[t] // LogicalForm
```

$$\text{Out}[\bullet]= \frac{5}{8}\,\text{Cos}\left[\sqrt{3}\,t\right]^2 \left|0_S\right\rangle\left\langle 0_S\right| + \frac{17}{24}\left|0_S\right\rangle\left\langle 0_S\right| \text{Sin}\left[\sqrt{3}\,t\right]^2 + \frac{1}{8}\sqrt{3}\left|0_S\right\rangle\left\langle 0_S\right| \text{Sin}\left[2\sqrt{3}\,t\right] +$$
$$\frac{1}{48}\,i\left|0_S\right\rangle\left\langle 1_S\right| \left((16+2i)\,\text{Sin}\left[\sqrt{3}\,t\right]^2 - (6+3i)\sqrt{3}\,\text{Sin}\left[2\sqrt{3}\,t\right]\right) +$$
$$\frac{1}{48}\left|1_S\right\rangle\left\langle 0_S\right| \left(18\,\text{Cos}\left[\sqrt{3}\,t\right]^2 - (5+2i)\sqrt{3}\,\text{Sin}\left[2\sqrt{3}\,t\right]\right) +$$
$$\frac{1}{48}\left|0_S\right\rangle\left\langle 1_S\right| \left(18\,\text{Cos}\left[\sqrt{3}\,t\right]^2 - (5-2i)\sqrt{3}\,\text{Sin}\left[2\sqrt{3}\,t\right]\right) +$$
$$\frac{1}{24}\left|1_S\right\rangle\left\langle 1_S\right| \left(8+\text{Cos}\left[2\sqrt{3}\,t\right] - 3\sqrt{3}\,\text{Sin}\left[2\sqrt{3}\,t\right]\right) +$$
$$\frac{1}{48}\left|1_S\right\rangle\left\langle 0_S\right| \left((-2-16i)\,\text{Sin}\left[\sqrt{3}\,t\right]^2 + (3+6i)\sqrt{3}\,\text{Sin}\left[2\sqrt{3}\,t\right]\right)$$

This visualizes the evolution of the state under the Hamiltonian on the Bloch sphere. Note that the magnitude of the Bloch vectors are preserved by the unitary dynamics.

```
In[•]:= ρρ = Bead /@ BlochVector /@ Table[rho[t], {t, 0, 1, 0.1}] // Chop;
        BlochSphere[{Red, ρρ}, "Opacity" → 0.4, ImageSize → Small]
```

Out[•]=



1.2.2 Quantum Noisy Dynamics

When a system interacts with its environment, the dynamics of the system alone cannot be described by the Schrödinger equation any longer, and more importantly, it is not unitary. At a first glance, it may not be surprising as similar effects also occur in classical mechanics. For example, a ball thrown up in the air interacts with various molecules and small particles in the air, and it does not follow Newton's equation of motion. The dynamics becomes dissipative, which is described by additional damping terms in the equation of motion. In effect, the damping terms arise because we are ignoring the small molecules and particles disturbing the ball. However, ignoring the environmental degrees of freedom brings about far more profound effects in quantum theory. It causes not only the dissipation of energy of the system to the environment, but also the effect of so-called *decoherence*, the loss of *quantumness* (Zurek, 1991, 2002). Here we outline the basic formalism of decoherence process, but the details will be discussed in Section 5.1.

Suppose that the system is initially in a state $|\psi\rangle \in \mathcal{H}$ and the environment in $|\xi\rangle \in \mathcal{E}$, and thus the initial state of the total system is $|\Psi(0)\rangle = |\psi\rangle \otimes |\xi\rangle \in \mathcal{H} \otimes \mathcal{E}$. Then let the system interacts with the environment. As a closed system, the

evolution of the total system is governed by a unitary time-evolution operator $\hat{U}_{\rm tot}(t)$, $|\Psi(t)\rangle = \hat{U}_{\rm tot}(t)\,|\psi\rangle \otimes |\xi\rangle$, in accordance with Postulate 2. As the system-environment coupling tends to build entanglement, the state $|\Psi(t)\rangle$ cannot be factorized in general. Without a comprehensive knowledge of the environment, the state of the system alone is thus a mixed state as discussed in Section 1.1.2. Specifically, the density matrix corresponding to the mixed state is obtained by reducing the total state $|\Psi(t)\rangle$ as

$$\hat{\rho}(t) = \operatorname*{Tr}_{\mathcal{E}} |\Psi(t)\rangle \langle\Psi(t)| = \operatorname*{Tr}_{\mathcal{E}} \hat{U}_{\rm tot}(t) \left(|\psi\rangle \langle\psi| \otimes |\xi\rangle \langle\xi|\right) \hat{U}^{\dagger}_{\rm tot}(t)\,, \tag{1.34}$$

where the partial trace over the environment physically corresponds to ignoring the environment degrees of freedom. In general, the dynamics of $\hat{\rho}(t)$ is thus very complicated, but there are two essential points to be noted here: First, the dynamics is not unitary any longer. Second, through the entanglement with the environment, the quantum state of the system loses coherence (i.e., quantumness).

The non-unitary dynamics of a system subject to coupling to the environment as expressed in (1.34) can be described most efficiently in terms of *quantum operations*. According to the operator-sum representation theorem, which will be detailed in Section 5.1.1, the effect of a quantum operation can be written in the form

$$\hat{\rho}(t) = \sum_{\mu} \hat{F}_{\mu}(t) \hat{\rho}(0) \hat{F}^{\dagger}_{\mu}(t)\,, \tag{1.35}$$

where $\hat{F}_{\mu}(t)$ are operators on $\mathcal{H}$ such that $\operatorname{Tr} \hat{F}^{\dagger}_{\mu} \hat{F}_{\nu} = 0$ for $\mu \neq \nu$. The conservation of probability $\operatorname{Tr} \hat{\rho}(t) = 1$ requires an addition condition that $\sum_{\mu} \hat{F}^{\dagger}_{\mu}(t) \hat{F}_{\mu}(t,0) = \hat{I}$ at any time t. Compared with the unitary dynamics in (1.33), the dissipative dynamics in (1.35) has several distinctive features: First, the operators $\hat{F}_{\mu}$ are not unitary. Second, it consists of more than one terms. The unitary case in (1.33) is a special case with $\hat{F}_1 = \hat{U}$ and all other $\hat{F}_{\mu}$ vanishing. Third and most importantly, the evolution depends on the whole history of its own, in stark contrast with the unitary dynamics as demonstrated in (1.31). Under certain conditions, the history can be ignored and the dynamics can be described to a good approximation by a so-called *quantum master equation*, a master-type differential equation for the density operator. Such a limit is called the *Markov limit*. Quantum master equations will be discussed further in Section 5.3.

1.3 Measurements on Quantum States

In classical mechanics, just like the state of motion, a physical quantity is described by a simple number (or a set of numbers when it is a vector quantity). As such, measurement is merely about assessing the number and only casts technological questions concerning measuring devices. In quantum mechanics, it is totally different and measurement is deeply intermingled into the quantum theory itself.

1.3.1 Projection Measurements

Postulate 3 A physical quantity is described by an "observable"—a Hermitian operator. Upon the *measurement* of an observable $\hat{A}$, the outcome is one of the eigenvalues a of $\hat{A}$ and determined *probabilistically*. When the system is in the state $|\psi\rangle$ right before the measurement, the probaility for a particular outcome a is given by $P_a = |\langle a|\psi\rangle|^2$, where $|a\rangle$ is the eigenstate of $\hat{A}$ belonging to the eigenvalue a. Right after the measurement, the state "collapses" to the eigenstate $|a\rangle$ corresponding to the outcome a.

The postulate points out several striking features that put quantum mechanics in stark contrast with classical mechanics. First, a measurable physical quantity (i.e., an observable) is described by an operator, not by a simple number, acting on the vectors that describe the quantum states of the system (see Postulate 1).

Second, measurement brings about a sudden collapse of the state vector. The unavoidable disturbance by a mere measurement causes obvious obstacles to naively examining quantum states, but at the same time it opens new opportunities such as secure quantum communication. Further, the collapse is a peculiar type of dynamics of quantum states, an alternative to the one governed by the Schrödinger equation (Postulate 2), which turns out to be useful in measurement-based quantum computing architecture (Section 3.4). Even in common dynamical schemes of quantum computation, measurement is often used to initialize the quantum registers to the logical basis states.

Third, the postulate makes it clear that even if the state vector $|\psi\rangle$ is known, which comprises the complete description of the state according to Postulate 1, one cannot infer the measurement result even in principle, but only the probabilities of possible outcomes. The description is inherently statistical and one need to measure an *ensemble* of identically prepared systems and extract the value of an ensemble in terms of the statistical moments such as the expectation value. Given a state vector $|\psi\rangle$, the expectation value of the operator $\hat{A}$ is obtained using the elementary theory of probability (see also Dirac's Bra-Ket notation in Appendix A.3)

$$\langle \hat{A} \rangle = \sum_q q P_q = \sum_q \langle \psi|q\rangle \, q \, \langle q|\psi\rangle = \langle \psi| \, \hat{A} \, |\psi\rangle \tag{1.36}$$

In the above statement of the postulate it was assumed that the system was initially in a pure state. It is naturally extended to the case of mixed state: When a system is in a mixed state $\hat{\rho}$ right before the measurement, the probability is given by $P_a = \langle a|\hat{\rho}|a\rangle$ and the state right after the measurement becomes $|a\rangle \langle a|$. By virtue of the elementary theory of probability, we observe again that $0 \le \hat{\rho} \le 1$ and $\mathrm{Tr}\, \hat{\rho} = 1$ as we have already seen in Section 1.1.2. Further, the expectation value of the observable is given by

$$\langle \hat{A} \rangle = \sum_a a P_a = \sum_a a \, \langle a| \, \hat{\rho} \, |a\rangle = \mathrm{Tr}\, \hat{A}\hat{\rho} \, . \tag{1.37}$$

Von Neumann envisioned an idealistic procedure to realize a projection measurement on physical systems. For this reason, projection measurement is also called as von Neumann measurement. The *von Neumann scheme* founded the quantum theory of measurement and has inspired various methods to push the measurement precision to the limit intrinsically put by quantum mechanics. The von Neumann scheme of projection measurement can be directly implemented in a quantum circuit model (see Section 4.4). Here briefly summarize the procedure: Suppose that we want to measure a quantity described by the Hermitian operator $\hat{A}$. Let $|a\rangle$ be the eigenvectors with eigenvalues a of $\hat{A}$ so that $\hat{A}$ has the spectral decomposition (see Appendix A.3)

$$\hat{A} = \sum_a |a\rangle\, a\, \langle a| \,. \tag{1.38}$$

The system is in a state $|\psi\rangle$, which is expanded as

$$|\psi\rangle = \sum_a |a\rangle\, \psi_a \tag{1.39}$$

in the eigenbasis of $\hat{A}$. Typically, it is supposed that the distribution $P_a := |\psi_a|^2$ is sharply peaked around a certain unknown eigenstate $|a_*\rangle$ and the measurement is supposed to reveal a_*.

We choose a measuring device the "position" $\hat{X}$ of which can be directly observed, and prepare it in an approximate eigenstate $|\xi\rangle$ of $\hat{X}$. In terms of the eigenstates $|x\rangle$ with eigenvalue x of $\hat{X}$, $|\xi\rangle$ is expanded as

$$|\xi\rangle = \int dx\ |x\rangle\, \xi(x) \tag{1.40}$$

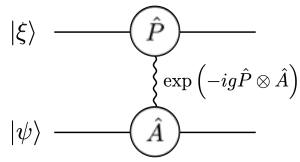
with $\xi(x)$ peaked sharply around a certain fixed point x_0. It will get clear below that the sharper the distribution $|\xi(x)|^2$ in the position is, the more accurate the measurement result gets. At this stage, the overall state of the total system composed of the system in question and the measurement device is given by $|\psi\rangle \otimes |\xi\rangle$. Now couple the observable $\hat{A}$ in question to the "momentum" $\hat{P}$ (not $\hat{X}$ itself) of the measurement device. Note that $\hat{P}$ and $\hat{X}$ are canonical conjugates of each other, $[\hat{X}, \hat{P}] = i$ (we have chosen a unit system such that $\hbar = 1$). The coupling is described by the interaction Hamiltonian

$$\hat{H}_{\text{int}} = J\hat{P} \otimes \hat{A}\,, \tag{1.41}$$

where J is the coupling strength. One has to let the system and the measurement device interact for sufficiently long time τ such that the position of the measurement device gets distinguished from the initial position. Let $g := J\tau$ be the dimensionless coupling constant. Due to the coupling, the total system composed of the system in question and the measurement device evolves in time, which is described by the unitary operator

$$\hat{U}_{\text{int}} = \exp\left(-ig\hat{P} \otimes \hat{A}\right). \tag{1.42}$$

This situation is depicted diagrammatically as following



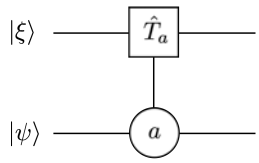
It is instructive to expand the interaction unitary operator $\hat{U}_{\text{int}}$ using the spectral decomposition [see Eq. (A.42)] of the observable $\hat{A}$ as

$$\hat{U}_{\text{int}} = \sum_a e^{-iga\hat{P}} \otimes |a\rangle \langle a| \tag{1.43}$$

The operator $\hat{T}_a := e^{-iga\hat{P}}$ is a translation operator with the amount of translation depending on the eigenvalue a of $\hat{A}$, and to make the physical interpretation clearer, we rewrite the interaction unitary operator into

$$\hat{U}_{\text{int}} = \sum_a \hat{T}_a \otimes |a\rangle \langle a| . \tag{1.44}$$

This picture is illustrated in the following schematic diagram



. (1.45)

After the interaction, the state vector of the total system becomes

$$\hat{U}_{\text{int}} |\Psi\rangle = \sum_a \left(\hat{T}_a |\xi\rangle\right) \otimes (|a\rangle \psi_a) = \int dx\ |x\rangle \otimes \left(\sum_a |a\rangle \xi(x - ga)\psi_a\right). \tag{1.46}$$

In general, the final state is an entangled state. The probability to register x out of the measurement on the measurement device is given by

$$P(x) = \sum_a |\xi(x - ga)|^2 |\psi_a|^2. \tag{1.47}$$

When the system starts from a definite eigenstate $|a\rangle$ of $\hat{A}$, the system and the measurement device remain unentangled after the system-measurement device interaction, with the probability distribution determined solely by the given eigenvalue a

$$P(x) = |\xi(x - ga)|^2 . \tag{1.48}$$

Manifestly, the measurement accuracy is best in this case. As illustrated in Fig. 1.2, one can extract the eigenvalue a of the observable from the amount of translation ga of the final wave function relative to the initial wave function assuming that the coupling strength g is known (it can be calibrated separately).

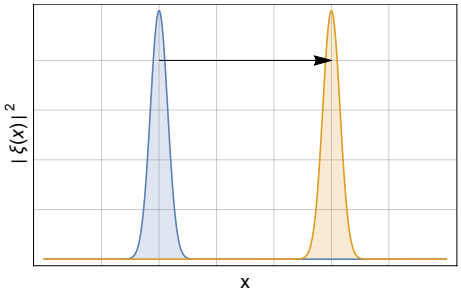


Figure 1.2: The change of the wave function $\xi(x)$ of the measurement device from the initial state (blue) to the final state (orange) after it interacts with the system. Illustrated is the case where the system is in a definite eigenstate $|a\rangle$ of the observable $\hat{A}$ to be measured. The amount of translation is given by ga, the coupling constant times the eigenvalue associated with $|a\rangle$.

Figure 1.2 suggests that the sharper the initial wave function of the measurement device is, the more precise the extracted value of a becomes. Roughly speaking, this is indeed one direction of efforts to improve the precision of measurement (Caves, 1981). An obstacle to precision measurement that is not directly apparent in Fig. 1.2 is the inherent statistical nature of quantum mechanics. As the probability distribution has a finite width, any measurement in quantum mechanics should be repeated many times. The statistical error decreases as $1/\sqrt{N}$ with the number N of repeated measurements. This is called the *standard quantum limit*. Interestingly, if one prepares a set of measurement devices in a proper entangled state, one can improve the statistical error to $1/N$ (Giovannetti *et al.*, 2006). This enhanced accuracy due to quantum entanglement is called the *Heisenberg limit*, and it puts the ultimate limit on the measurement precision that quantum mechanics allows.

1.3.2 Generalized Measurements

We have already introduced a Postulate 3 concerning measurement, and it is usually the form of the measurement postulate discussed in most textbooks on quantum mechanics. In quantum information context, however, it is more convenient to generalize it for various reasons to be discussed shortly.

Postulate 3′ Quantum measurements are described by a set $\{\hat{M}_m\}$ of *measurement operators* corresponding to *measurement outcomes* m satisfying the completeness relation $\sum_m \hat{M}_m^\dagger \hat{M}_m = \hat{I}$. If the quantum state of the system is $\hat{\rho}$ right before the measurement, then the probability for outcome m is equal to $P_m = \mathrm{Tr}\, \hat{M}_m \hat{\rho} \hat{M}_m^\dagger$, and the state right after measurement becomes $P_m^{-1} \hat{M}_m \hat{\rho} \hat{M}_m^\dagger$.

In general, the physical meaning of the measurement operators $\hat{M}_m$ is not immediately clear until the specifics of the measurement in question are given. It

is even more obscure how one can set up a physical device for the measurement described by the given set of operators $\hat{M}_m$. The projection measurement discussed previously is a special case of the generalized measurement. In this case, the measurement operators are the projection $|a\rangle\langle a|$ into the eigenstate of the observable $\hat{A}$ being measured. In fact, one can show that projective measurements (extended to a larger system and supplemented with additional unitary operations) are equivalent to generalized measurements. In many applications, however, generalized measurements are more mathematically convenient because they are less restrictive. A common example to illustrate this point is the problem of distinguishing quantum states (see Example 5). One important physical motivation for generalized measurements can be seen by noting that for a projective measurement, once the measurement outcome a is obtained, all subsequent measurements will give the same outcome. This property is inherently implied in any projective measurement. For many measurement in reality, however, repetition is not possible. Consider a photo-detector as an example, where the photon is immediately destroyed upon the observation and obviously repetition is not physically allowed.

Example 5 Consider a fixed set of n *mutually orthogonal* quantum states

$$\{|\psi_j\rangle : j = 1, \ldots, n\} . \tag{1.49}$$

Suppose that a state is drawn from the set, and your task is to tell which of the n state it is. Define n operators $\hat{M}_j := |\psi_j\rangle\langle\psi_j|$ for $j = 1, \ldots, n$ and another operator $\hat{M}_0 := \hat{I} - \sum_j \hat{M}_j$. They satisfy the completeness relation $\sum_{j=0}^{n} \hat{M}_j^\dagger \hat{M}_j = \hat{I}$ and describe a measurement. Now when the particular state $|\psi_k\rangle$ is taken, the probability for the measurement outcome k, $P_k = \langle\psi_k|\hat{M}_k|\psi_k\rangle = 1$, is unity.

Measurement can also be regarded as a special case of quantum operations (see Sections 1.2.2 and 5.1). The mapping $\mathscr{E}_m \in \mathcal{L}(\mathcal{V})$ defined by

$$\mathscr{E}_m(\hat{\rho}) = \hat{M}_m \hat{\rho} \hat{M}_m^\dagger \tag{1.50}$$

for each m is obviously a quantum operation. This is natural as a measurement process involves the interaction of the system with the measuring devices. Note that the quantum operation $\mathscr{E}_m$ does not preserve the trace in general, $0 \leq \mathrm{Tr}\,\mathscr{E}_m(\rho) \leq 1$. The measurement given above is a *selective measurement*. A *non-selective measurement* is represented by the quantum operation (Breuer & Petruccione, 2002)

$$\mathscr{F}(\hat{\rho}) := \sum_m \mathscr{F}_m(\hat{\rho}) = \sum_m \hat{M}_m \hat{\rho} \hat{M}_m^\dagger . \tag{1.51}$$

In this case, the trace is preserved: $\mathrm{Tr}\,\mathscr{F}(\hat{\rho}) = 1$ for any $\hat{\rho}$. It follows from the completeness relation satisfied by the measurement operators.

Postulate 3 or 3′ describes not only the probability for a particular measurement outcome but also the state corresponding to the outcome after the measurement. In many experiments, however, the system is measured only once, and the state after the measurement is irrelevant—of the primary concern is the probabilities. Often, it is even impossible to assign a state to the system. For example, in an experiment to measure the position of photon, the photon is destroyed at the photo-screen and it is meaningless to ask about the wave function of the photon after measurement. As long as one is focused on the probabilities, the description in Postulate 3′ can be drastically simplified: Since the trace is invariant under cyclic permutation of the operators, the probability for the outcome m is given by

$$P_m = \mathrm{Tr}\, \hat{M}_m \hat{\rho} \hat{M}_m^\dagger = \mathrm{Tr}\, \hat{M}_m^\dagger \hat{M}_m \hat{\rho}\,. \tag{1.52}$$

Therefore, as long as the probabilities are concerned, all we need are the operators $\hat{E}_m := \hat{M}_m^\dagger \hat{M}_m$ but not the measurement operators $\hat{M}_m$. Note that $\hat{M}_m$ may contain far more details that cannot be extracted from $\hat{E}_m$ only. Infinitely many different measurement operators can lead to the same $\hat{E}_m$. In analogy, $\hat{M}_m$ is to $\hat{E}_m$ what $|\psi_j\rangle$ and P_j are to $\hat{\rho}$. The discard of certain details in $\hat{M}_m$ for more compact operators $\hat{E}_m$ is possible because we not care about the post-measurement state. By construction, $\hat{E}_m$ are positive semi-definite operators and satisfy $\sum_m \hat{E}_m = \hat{I}$. The set $\{\hat{E}_m\}$ of such operators is called a *positive operator-valued measure* or more often just a *POVM*. The individual operators $\hat{E}_m$ are called the POVM elements. In many measurement experiments, the POVM formalism allows far more efficient description of the measurement.

As a non-trivial example of the application of the POVM formalism, let us consider the problem of quantum state discrimination of non-orthogonal states (Bergou *et al.*, 2004; Chefles, 2004). To be specific, suppose that we are given either of the two states

$$|v\rangle = |0\rangle\ , \quad |w\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\,. \tag{1.53}$$

We want to figure out which it is. Apparently, these two states are not orthogonal, $\langle v|w\rangle \neq 0$, and it is impossible to discriminate them with certainty—see Example 5. Instead, the task is to perform a measurement that tells which of the two, but never misidentifies a wrong state. Consider a POVM including the following three elements

$$\hat{E}_1 = \frac{1}{\sqrt{1+\langle v|w\rangle}} |1\rangle\langle 1|\ , \tag{1.54a}$$

$$\hat{E}_2 = \frac{1}{\sqrt{1+\langle v|w\rangle}} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(\frac{\langle 0| - \langle 1|}{\sqrt{2}}\right) \tag{1.54b}$$

$$\hat{E}_3 = \hat{I} - \hat{E}_1 - \hat{E}_2 \tag{1.54c}$$

It is straightforward to check that they form a POVM. Suppose that the state $|v\rangle$ is given. As it is orthogonal to $|1\rangle$, the measurement outcome $m = 1$—corresponding

to the POVM element $\hat{E}_1$—never occurs. It means that once $m = 2$—corresponding to $\hat{E}_2$—occurs, the state must be $|w\rangle$. For the same reason, the result $m = 1$ implies that the state is definitely $|v\rangle$. Of course, it is possible to get the outcome $m = 3$, in which case the measurement tells nothing. However, the important point is that there is no chance to make a mistake.

Problems

1. Consider a set of *arbitrary* state vectors $\{|\psi_\mu\rangle : \mu = 0, 1, \ldots, n-1\}$ in a vector space. Note that the vectors are not normalized nor orthogonal to each other. Let $\hat{A}$ be a linear operator defined by

$$\hat{A} = \sum_{\mu=0}^{n-1} |\psi_\mu\rangle \langle\psi_\mu| \,. \tag{1.55}$$

   Show that the eigenvalues of $\hat{A}$ are all non-negative.

2. Consider a two-level quantum system. Let $|0\rangle$ and $|1\rangle$ be the logical basis of the Hilbert space associated with the system. Let $\hat{S}^\mu$ $(\mu = 0, x, y, z)$ be the Pauli operator, defined by

$$\hat{S}^x |0\rangle = |1\rangle \,, \qquad \hat{S}^y |0\rangle = +i\,|1\rangle \,, \qquad \hat{S}^z |0\rangle = +\,|0\rangle \,, \tag{1.56a}$$

$$\hat{S}^x |1\rangle = |0\rangle \,, \qquad \hat{S}^y |1\rangle = -i\,|0\rangle \,, \qquad \hat{S}^z |1\rangle = -\,|1\rangle \,, \tag{1.56b}$$

   and $\hat{S}^0 = \hat{I}$.

   (a) Find the matrix representations of $\hat{S}^\mu$ $(\mu = 0, x, y, z)$ in the logical basis.

   (b) Find the matrix representations of $\hat{S}^\mu$ in the new basis

$$|\pm\rangle := \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}. \tag{1.57}$$

3. Consider a spin 1/2 in an external magnetic field. We describe the states of the spin with the vector space spanned by the logical basis states $|0\rangle \equiv |\uparrow\rangle$ and $|1\rangle \equiv |\downarrow\rangle$. In terms of the Pauli operators in Eq. (1.56), the Hamiltonian is given by

$$\hat{H} = B_x \hat{S}^x + B_y \hat{S}^y + B_z \hat{S}^z \,, \tag{1.58}$$

   where B_μ $(\mu = x, y, z)$ are the parameters proportional to the external magnetic field—in this expression, they have the same dimension as the energy. Suppose that

$$B_x = B_0 \sin\theta \cos\phi \,, \quad B_y = B_0 \sin\theta \sin\phi \,, \quad B_z = B_0 \cos\theta \tag{1.59}$$

   with $B_0 > 0$ and $\theta, \phi \in \mathbb{R}$.

(a) Find the eigenvalues and corresponding eigenstates of $\hat{H}$.
Hint: Direct evaluation of the Hamiltonian in Eq. (1.58) is straightforward. Another method is to use the commutation relations of the Pauli operators.

(b) Display the two eigenstates on the Bloch sphere.

4. Let $|\psi\rangle$ be a vector in a two-dimensional vector space. Show that the Bloch vector $\boldsymbol{b} := (\langle\hat{S}^x\rangle, \langle\hat{S}^y\rangle, \langle\hat{S}^z\rangle) \in \mathbb{R}^3$ has the magnitude of unity, $|\boldsymbol{b}| = 1$.

5. Let $\hat{\rho}$ be a density operator on a two-dimensional vector space. It can be written as

$$\hat{\rho} = \frac{1}{2}\left(\hat{S}^0 + x\hat{S}^x + y\hat{S}^y + z\hat{S}^z\right), \tag{1.60}$$

where $\hat{S}^\mu$ are the Pauli operators—see Eq. (1.57)—and $x, y, z \in \mathbb{R}$.

(a) Show that the Bloch vector defined by $\boldsymbol{b} := (\langle\hat{S}^x\rangle, \langle\hat{S}^y\rangle, \langle\hat{S}^z\rangle)$ is just given by $\boldsymbol{b} = (x, y, z)$.

(b) Show that $|\boldsymbol{b}| \leq 1$.

(c) Show that $\hat{\rho}$ is a pure state if and only if $|\boldsymbol{b}| = 1$.
Hint: See Problem 3.

6. Consider a two-qubit system, and suppose that it is in the state

$$|\psi\rangle = \frac{|0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle)}{\sqrt{3}}. \tag{1.61}$$

(a) What is the probability p_0 to find the first qubit in $|0\rangle$ (regardless of the second qubit)? Similarly, what is the probability p_2 to find the first qubit in the state $|1\rangle$?

(b) What is the probability p_0' to find the first qubit in $(|0\rangle + |1\rangle)/\sqrt{2}$? What about the probability p_1' to find the first qubit in $|0\rangle$?

(c) Suppose that you do not have an access to the second qubit, and that you want to construct the density operator $\hat{\rho}$ for the first qubit correponding to the physical situation. Which of the two data sets from (6a) and (6b) would you use for the construction of $\hat{\rho}$? That is, if the resulting density operators are different, which one is correct and why?

(d) Calculate the Bloch vector $\boldsymbol{b}$ corresponding to $\hat{\rho}$ in (b), and display it in the Bloch sphere.

7. Consider a system of two two-level susystems. The logical basis and the Pauli operators for each subsystem are defined as in Problem 2.

(a) Find the matrix representations of the following operator

$$\hat{H} = \hat{S}^x \otimes \hat{S}^x + \hat{S}^y \otimes \hat{S}^y + \hat{S}^z \otimes \hat{S}^z \tag{1.62}$$

in the standard tensor-product basis $\{|i\rangle \otimes |j\rangle : i, j = 0, 1\}$.

(b) Find all eigenvalues and eigenvectors of $\hat{H}$.

(c) Calculate the following unitary operator

$$\hat{U} = \exp(-it\hat{H}), \quad t \in \mathbb{R}, \tag{1.63}$$

which corresponds physically to the time-evolution operator, and express it in terms of either $\hat{S}^\mu \otimes \hat{S}^\nu$ or the dyadic products, $|i\rangle\langle j| \otimes |k\rangle\langle l|$ $(i, j, k, l = 0, 1)$.

(d) Evaluate the state

$$|\psi(t)\rangle := \hat{U}(t)\,|+\rangle \otimes |-\rangle\,, \tag{1.64}$$

where $|\pm\rangle$ are defined in Eq. (1.57).

(e) Evaluate the expectation values

$$S_1^\mu(t) := \langle\psi(t)|\,\hat{S}^\mu \otimes \hat{S}^0\,|\psi(t)\rangle\,, \tag{1.65}$$

for $\mu = x, y, z$, and plot them as functions of t.

(f) Evaluate the expectation values

$$S^\mu(t) := \langle\psi(t)|\left(\hat{S}^\mu \otimes \hat{S}^0 + \hat{S}^0 \otimes \hat{S}^\mu\right)|\psi(t)\rangle\,, \tag{1.66}$$

for $\mu = x, y, z$, and plot them as functions of t.

8. Consider a two-qubit system in the following state

$$|\psi\rangle = \frac{1}{2}\sum_{x=0}^{3}|x\rangle\,. \tag{1.67}$$

Here we have used a short-hand notation $|x\rangle := |x_1\rangle \otimes |x_2\rangle$, where x_j $(j = 1, 2)$ are the binary digits of x, $x = (x_1x_2)_2$. For example, $|2\rangle = |1\rangle \otimes |0\rangle$. Suppose that we measure an observable

$$\hat{H} = \sum_{\mu\in\{x,y\}} \hat{S}^\mu \otimes \hat{S}^\mu\,, \tag{1.68}$$

where $\hat{S}^\mu$ $(\mu = x, y, z)$ are the Pauli operators.

(a) Find all possible values of the outcome of the measurement according to the postulates of quantum mechanics.

(b) For each of the values found in (a), find the probability to actually observe the value as the measurement outcome. Plot the probabilities versus the possible values, say, in a bar chart to compare them directly.

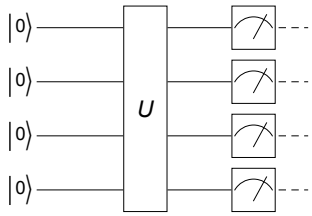
(c) For each of the values found in (a), find the post-measurement state when the value actually is registered as the measurement outcome.

Chapter 2

Quantum Computation: Overview

- April 29, 2021 (v1.14)

In the simplest physical terms, quantum computation is an implementation of an arbitary unitary operation on a finite collection of *quantum bits* or *qubits* for short—a two-level quantum system. It is typically depicted in a *quantum circuit diagram* as following:



(2.1)

Each qubit is associated with a line which indicates the time evolution of the state specified on the left. The time flows from the left to the right. The *quantum logic gate operations* (or *gates* for short) on single or multiple qubits are denoted by a rectangular box often with labels indicating the types of the gates. Measurements are denoted by square boxes with needles. After a measurement, the time-evolution is represented by dashed lines to remind that the information is classical—no superposition.

The input state is prepared in one of the states in the logical basis, typically $|0\rangle \otimes \cdots \otimes |0\rangle$. After the unitary operation, the resulting state is measured in the logical basis, and the readouts are supposed to be the result of computation.

In order for a quantum computer to be programmable, it is required to implement a given unitary operator $\hat{U}$ in a combination of other more elementary unitary operators

$$\hat{U} = \hat{U}_1 \hat{U}_2 \cdots \hat{U}_L \,, \tag{2.2}$$

where each $\hat{U}_j$ is chosen from a small fixed set of elementary gate operations. The latter operations are called the *elementary quantum logic gates* for the quantum computer. In this chapter, we will examine widely used choices for elementary gates, and illustrate how a set of elementary gates achieve the arbitrary unitary operation, the so-called *universal quantum computation*.

Throughout the chapter, we denote by $\mathcal{S}$ the Hilbert space associated with a single qubit. The Hilbert space of an n-qubit system is given by $\mathcal{S}^{\otimes n}$, a tensor-product space of multiple $\mathcal{S}$. Each element $|x\rangle$ in the logical basis of $\mathcal{S}^{\otimes n}$ will be labeled by an integer $x = 0, 1, \cdots, 2^n - 1$, which should be understood to enumerate the tensor product form

$$|x\rangle \equiv |x_1 x_2 \cdots x_n\rangle := |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \tag{2.3}$$

in terms of the binary digits x_j $(j = 1, 2, \cdots, n)$ of x, that is, $x \equiv (x_1 x_2 \ldots x_n)_2$.

2.1 Single-Qubit Gates

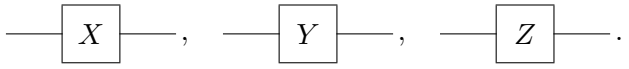
Unitary operators on the two-dimensional vector space $\mathcal{S}$ associated with a singel qubit form the *unitary group* U(2). In the standard basis, they are represented by a 2×2 unitary matrices. We first take a look at some special examples and discuss the general properties of the single-qubit unitary operations.

2.1.1 Pauli Gates

The Pauli gate opertaions (or Pauli operators for short) are defined by the corresponding Pauli matrices

$$\hat{X} \doteq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \hat{Y} \doteq \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \hat{Z} \doteq \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.4}$$

They form the most elementary single-qubit gate operations and are frequently used in many quantum algorithms. In this book, the Pauli gates will be sometimes denoted by $\hat{X}$, $\hat{Y}$, and $\hat{Z}$, and othertimes by $\hat{S}^x$, $\hat{S}^y$, and $\hat{S}^z$, depending on the context. In the quantum circuit model, they are typically depicted by the circuit elements



(2.5)

Pauli $\hat{X}$ maps the logical basis states as

$$\hat{X} : |0\rangle \mapsto |1\rangle\,, \; |1\rangle \mapsto |0\rangle\,, \tag{2.6}$$

and is similar to the classical logic gate NOT. It is also customary to write Pauli $\hat{X}$ in the bra-ket notation as

$$\hat{X} = |1\rangle\langle 0| + |0\rangle\langle 1|\,. \tag{2.7}$$

It is important to remember that like any other quantum gate operations, it can take a linear superposition as input and transform the two logical basis states "simultaneously",

$$\hat{X}(|0\rangle\, c_0 + |1\rangle\, c_1) = |1\rangle\, c_0 + |0\rangle\, c_1\,, \tag{2.8}$$

which is not possible with the classical counterpart NOT.

The Pauli X gate is represented by `S[...,1]`.

```
In[•]:= S[1]
Out[•]= S^x
```

It corresponds to the Pauli X matrix.

```
In[•]:= Matrix[S[1]] // MatrixForm
Out[•]//MatrixForm=
        ( 0  1 )
        ( 1  0 )
```

In the quantum circuit model, it is denoted by the following quantum circuit element.

```
In[•]:= QuantumCircuit[S[1]]
Out[•]=  ─┤ X ├─
```

Operating on the logical basis states, it flips the states and is similar to the classical logical gate NOT.

```
In[•]:= bs = Basis[S];
        out = S[1] ** bs;
        Thread[bs → out] // LogicalForm // TableForm
Out[•]//TableForm=
        |0_S⟩ → |1_S⟩
        |1_S⟩ → |0_S⟩
```

Operating on a superposition state, it flips the state "simultaneously".

```
In[•]:= in = Ket[] × c[0] + Ket[S → 1] × c[1];
        in // LogicalForm
        out = S[1] ** in;
        out // LogicalForm
Out[•]= c_0 |0_S⟩ + c_1 |1_S⟩
Out[•]= c_1 |0_S⟩ + c_0 |1_S⟩
```

Operating on the logical basis states, Pauli $\hat{Z}$ only changes the relative phase of $|1\rangle$,

$$\hat{Z} : |0\rangle \mapsto |0\rangle\,,\ |1\rangle \mapsto -|1\rangle\,, \tag{2.9}$$

and hence in the bra-ket notation, it reads as

$$\hat{Z} = |0\rangle\langle 0| - |1\rangle\langle 1|\,. \tag{2.10}$$

The phase change is meaningless on classical bits, but it makes a significant difference on a superposition as illustrated in the following example

$$\hat{Z}(|0\rangle\, c_0 + |1\rangle\, c_1) = |0\rangle\, c_0 - |1\rangle\, c_1\,. \tag{2.11}$$

The Pauli Z gate is represented by `S[...,3]`.

```
In[•]:= S[3]
Out[•]= S^z
```

It corresponds to the Pauli Z matrix.

```
In[•]:= Matrix[S[3]] // MatrixForm
Out[•]//MatrixForm=
        ( 1   0 )
        ( 0  -1 )
```

In the quantum circuit model, it is denoted by the following quantum circuit element.

```
In[•]:= QuantumCircuit[S[3]]
Out[•]=  —[ Z ]—
```

Operating on the logical basis states, it "flips the phase", that is, it changes the phase factor to `-1` of the logical basis state $|1\rangle$.

```
In[•]:= bs = Basis[S];
        out = S[3] ** bs;
        Thread[bs → out] // LogicalForm // TableForm
Out[•]//TableForm=
        |0_S⟩ → |0_S⟩
        |1_S⟩ → - |1_S⟩
```

Here is an example how the Pauli Z gate acts on a superposition state.

```
In[•]:= in = Ket[] × c[0] + Ket[S → 1] × c[1];
        in // LogicalForm
        out = S[3] ** in;
        out // LogicalForm
Out[•]= c_0 |0_S⟩ + c_1 |1_S⟩
Out[•]= c_0 |0_S⟩ - c_1 |1_S⟩
```

Pauli $\hat{Y}$ combines the bit-flip feature of $\hat{X}$ and the phase-flip feature of $\hat{Z}$ to get

$$|0\rangle \mapsto i\,|1\rangle\ , \quad |1\rangle \mapsto -i\,|0\rangle\ . \tag{2.12}$$

This can also be seen in the operator identity, $\hat{Y} = i\hat{X}\hat{Z}$. In the bra-ket notation, it reads as

$$\hat{Y} = i\,|1\rangle\langle 0| - i\,|0\rangle\langle 1|\ . \tag{2.13}$$

The Pauli Y gate is represented by `S[...,2]`.

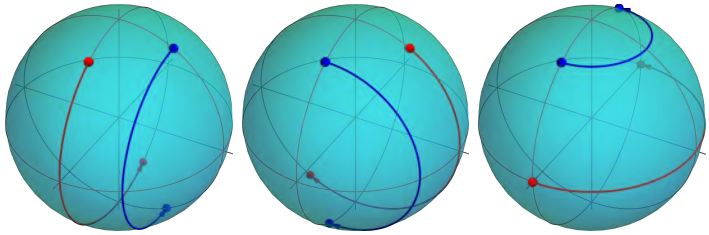


Figure 2.1: Illustration of the actions of Pauli gates as rotaions by angle π. From the left the actions of Pauli $\hat{X}$, $\hat{Y}$, $\hat{Z}$.

```
In[•]:= S[2]
Out[•]= S^y
```

It corresponds to the Pauli Y matrix.

```
In[•]:= Matrix[S[2]] // MatrixForm
Out[•]//MatrixForm=
( 0   -i )
( i    0 )
```

In the quantum circuit model, it is denoted by the following quantum circuit element.

```
In[•]:= QuantumCircuit[S[2]]
Out[•]=  —[ Y ]—
```

Pauli Y is a combination of the bit-flip (Pauli X) and phase-flip (Pauli Z) operation.

```
qc = QuantumCircuit[S[1], S[3]]
op = ExpressionFor[qc]
Out[•]=  —[ X ]—[ Z ]—
Out[•]= i S^y
```

This shows more explicitly how Pauli Y "flips" both the bit and phase of the logical basis states.

```
In[•]:= bs = Basis[S];
out = S[2] ** bs;
Thread[bs → out] // LogicalForm // TableForm
Out[•]//TableForm=
|0_S⟩ → i |1_S⟩
|1_S⟩ → -i |0_S⟩
```

Here is an example how the Pauli Y gate acts on a superposition state.

```
In[•]:= in = Ket[] × c[0] + Ket[S → 1] × c[1];
in // LogicalForm
out = S[2] ** in;
out // LogicalForm
Out[•]= c_0 |0_S⟩ + c_1 |1_S⟩
Out[•]= -i c_1 |0_S⟩ + i c_0 |1_S⟩
```

The Pauli gates can also be regarded as rotations by π around the x-, y-, and z-axis, respectively, in the Bloch sphere as illustrated in Fig. 2.1 and demonstrated in the following:

The Pauli gates also correspond, up to a global phase factor ($-i$), to rotations by the corresponding axes by angle π. Here `QuissoRotation[ϕ,S[...,μ]]` gives the rotation operator around the μ-axis by angle ϕ.

```
In[•]:= QuissoRotation[Pi, S[1]]
Out[•]= - i S^x

In[•]:= QuissoRotation[Pi, S[2]]
Out[•]= - i S^y

In[•]:= QuissoRotation[Pi, S[3]]
Out[•]= - i S^z
```

Here we are mainly focusing on their roles as unitary operators. However, the Pauli operators play another important role as an orthogonal *basis vectors* of the vector space of all linear operators on a two-dimensional vector space—see Section 2.1.3 and Appendix B.1.

2.1.2 Hadamard Gate

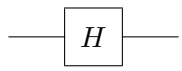
The Hadamard gate is one of the most frequently used elementary gates in many quantum algorithms. The Hadamard gate $\hat{H}$ is defined by the mapping:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.14}$$

That is, it constructs linear superpoistions of the two logical basis states. It is this feature that makes the Hadamard gate so useful, and is exploited in a wide range of quantum algorithms. In the logical basis, it is represented by the 2×2 Hadamard matrix

$$\hat{H} \doteq \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.15}$$

In the quantum circuit model, the Hadamard gate is depicted by an element with label "H"



(2.16)

Note that the output states in (2.14) corresponds to the eigenstates of the Pauli X operator. One can thus regard the Hadamard gate as a basis transformation from the logical basis to the eigenbasis of the Pauli X gate.

The Hadamard gate is represented by S[..., 6] .

```
In[•]:= S[1, 6]
Out[•]= S_1^H
```

Let us consider all the logical basis states.

```
In[•]:= bs = Basis[S[1]];
        bs // LogicalForm
Out[•]= { |0_S1⟩ , |1_S1⟩ }
```

Operating the Hadamard gate on them gives the two superposition states.

```
In[•]:= out = S[1, 6] ** bs;
        out // LogicalForm
Out[•]= { |0_S1⟩/√2 + |1_S1⟩/√2 , |0_S1⟩/√2 - |1_S1⟩/√2 }
```

In the quantum circuit model, it is denoted by the following circuit element.

```
In[•]:= QuantumCircuit[S[1, 6]]
Out[•]= ─┤ H ├─
```

It is also ingsightful to note that it can be regarded (up to a global phase factor) as a rotation by angle π around the axis $(1, 0, 1)$ on the Bloch sphere. This can be seen from the following:

$$\hat{H} = \frac{1}{\sqrt{2}}\left(\hat{X} + \hat{Z}\right) = i \exp\left[-i\frac{\pi}{2}(\hat{X} + \hat{Z})\right] \tag{2.17}$$

This is illustrated in Fig. 2.2.

Geometrically, the Hadamard gate can be regarded as a rotation around the axis `(1,0,1)` in the Pauli space by angle π.

```
In[•]:= op = I QuissoRotation[π, S, {1, 0, 1}] // Garner
        mat = Matrix[op];
        mat // MatrixForm
Out[•]= S^x/√2 + S^z/√2

Out[•]//MatrixForm=
        ⎛ 1/√2   1/√2 ⎞
        ⎝ 1/√2  -1/√2 ⎠
```

An obvious but very useful feature is that it makes a linear superposition of *all* states in the logical basis: Consider a *quantum register* consisting of n qubits. When applied to each qubit in $|0\rangle$, it generates a linear superposition of all states in the logical basis

$$\hat{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \cdots \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \ , \tag{2.18}$$

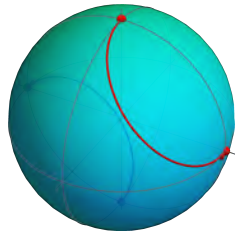


Figure 2.2: Illustration of the action of the Hadamard gate on the Bloch sphere. The Hadamard gate correponds to a rotation around the axis $(1, 0, 1)$ in the xz-plane by angle π.

where $|x\rangle := |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$ for an integer x represented by $x = (x_1 x_2 \ldots x_n)_2$ in the binary digits. More generaly, for an arbitrary state $|y\rangle$ in the logical basis,

$$\hat{H}^{\otimes n} |y\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \, (-1)^{x \cdot y} \,, \tag{2.19}$$

where we have used a short-hand notation

$$x \cdot y := x_1 y_1 + \cdots + x_n y_n \pmod 2. \tag{2.20}$$

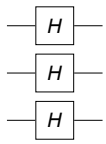
Suppose the Hadamard gates are applied to three qubits.

```
In[•]:= op = HoldForm@Multiply[S[1, 6], S[2, 6], S[3, 6]]
Out[•]= S_1^H S_2^H S_3^H
```

This shows the overall operation in the quantum circuit model.

```
In[•]:= qc = QuantumCircuit[S[{1, 2, 3}, 6], Null]
```

Out[•]=



Operating the Hadamard gate on each qubit produces a superposition state consisting all logical basis states.

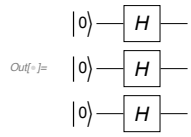
```
In[•]:= out = ReleaseHold[op] ** Ket[];
        out // LogicalForm
```

$$\text{Out}[\bullet] = \frac{|0_{s_1} 0_{s_2} 0_{s_3}\rangle}{2\sqrt{2}} + \frac{|0_{s_1} 0_{s_2} 1_{s_3}\rangle}{2\sqrt{2}} + \frac{|0_{s_1} 1_{s_2} 0_{s_3}\rangle}{2\sqrt{2}} + \frac{|0_{s_1} 1_{s_2} 1_{s_3}\rangle}{2\sqrt{2}} + \frac{|1_{s_1} 0_{s_2} 0_{s_3}\rangle}{2\sqrt{2}} + \frac{|1_{s_1} 0_{s_2} 1_{s_3}\rangle}{2\sqrt{2}} + \frac{|1_{s_1} 1_{s_2} 0_{s_3}\rangle}{2\sqrt{2}} + \frac{|1_{s_1} 1_{s_2} 1_{s_3}\rangle}{2\sqrt{2}}$$

This show the same result in the quantum circuit model.

```
qc = QuantumCircuit[LogicalForm[Ket[], S@{1, 2, 3}], S[{1, 2, 3}, 6]]
ExpressionFor[qc] // LogicalForm
```

Out[•]=



$$\frac{\left|0_{S_1}0_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|0_{S_1}0_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|0_{S_1}1_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|0_{S_1}1_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}+$$
$$\frac{\left|1_{S_1}0_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|1_{S_1}0_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|1_{S_1}1_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|1_{S_1}1_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}$$

Let us compare it with an explicit construction. To do it first prepare the indices of the logical basis states in binary digits.

```
nn = Range[0, 2^3 - 1];
bit = IntegerDigits[nn, 2, 3]
```

Out[•]=

```
{{0, 0, 0}, {0, 0, 1}, {0, 1, 0},
 {0, 1, 1}, {1, 0, 0}, {1, 0, 1}, {1, 1, 0}, {1, 1, 1}}
```

This gives an explicit construction (unnormalized) of the superposition of all local basis states.

```
vec = Total[Ket[S@{1, 2, 3} → #] & /@ bit];
vec // LogicalForm
```

Out[•]=

$$\left|0_{S_1}0_{S_2}0_{S_3}\right\rangle+\left|0_{S_1}0_{S_2}1_{S_3}\right\rangle+\left|0_{S_1}1_{S_2}0_{S_3}\right\rangle+$$
$$\left|0_{S_1}1_{S_2}1_{S_3}\right\rangle+\left|1_{S_1}0_{S_2}0_{S_3}\right\rangle+\left|1_{S_1}0_{S_2}1_{S_3}\right\rangle+\left|1_{S_1}1_{S_2}0_{S_3}\right\rangle+\left|1_{S_1}1_{S_2}1_{S_3}\right\rangle$$

On other elements of the logical basis, the sign of each term is determined by the bitwise dot product of the its bit-string with that of the input state.

```
in = Ket[S[{1, 2, 3}] → {1, 0, 1}];
in = LogicalForm[in, S@{1, 2, 3}]
```

Out[•]=

$$\left|1_{S_1}0_{S_2}1_{S_3}\right\rangle$$

```
out = ReleaseHold[op] ** in;
out // LogicalForm
```

Out[•]=

$$\frac{\left|0_{S_1}0_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}-\frac{\left|0_{S_1}0_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|0_{S_1}1_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}-\frac{\left|0_{S_1}1_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}-$$
$$\frac{\left|1_{S_1}0_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|1_{S_1}0_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}-\frac{\left|1_{S_1}1_{S_2}0_{S_3}\right\rangle}{2\sqrt{2}}+\frac{\left|1_{S_1}1_{S_2}1_{S_3}\right\rangle}{2\sqrt{2}}$$

2.1.3 Rotations

Any unitary operator $\hat{U}$ can always be written in the form $\hat{U} = \exp(-i\hat{H})$ with a Hermitian operator $\hat{H}$. On a two-dimensional vector space $\mathcal{S}$ associated with a qubit, any Hermitian operator $\hat{H}$ can be expanded in terms of the Pauli operators $\hat{S}^\mu$ as

$$\hat{H} = \phi_0 + \hat{S}^x B_x + \hat{S}^y B_y + \hat{S}^z B_z \tag{2.21}$$

where ϕ_0, B_x, B_y, B_z are real parameters. Regarding $\boldsymbol{B} := (B_x, B_y, B_z)$ as a three-dimensional vector, we consider the unit vector $\boldsymbol{n}$ pointing to the same direction

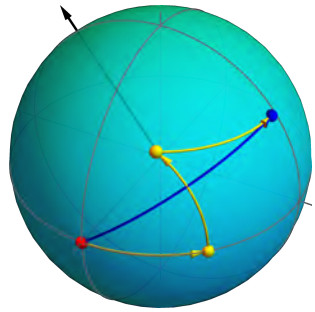


Figure 2.3: Visualization of transformations of states under the single-qubit operations. Up to a global phase factor, a single-qubit unitary operation is a rotation on the Bloch sphere. The rotation around the axis indicated by the black arrow is depicted by the blue arrow. The same rotation can be achieved by combining three rotations around y- or z-axis depicted by the yellow arrows.

as $\boldsymbol{B}$ and another real parameter $\phi := 2|\boldsymbol{B}|$, where the factor 2 is just for later convenience. In terms of these new parameterization, $\hat{H}$ reads as

$$\hat{H} = \phi_0 + \hat{\boldsymbol{S}} \cdot \boldsymbol{n}\, \phi/2\,, \tag{2.22}$$

where $\hat{\boldsymbol{S}} := (\hat{S}^x, \hat{S}^y, \hat{S}^z)$. In short, any unitary operator on $\mathcal{S}$ has the form $\hat{U} = e^{-i\phi_0}\hat{U}_{\boldsymbol{n}}(\phi)$ with

$$\hat{U}_{\boldsymbol{n}}(\phi) := \exp(-i\hat{\boldsymbol{S}} \cdot \boldsymbol{n}\, \phi/2). \tag{2.23}$$

Here $e^{-i\phi_0}$ changes the global phase factor and is physically irrelevant. More important and interesting is the part $\hat{U}_{\boldsymbol{n}}(\phi)$, which as we will see below, describes a "rotation" around the axis $\boldsymbol{n}$ by the angle ϕ. The rotations here are on the Bloch sphere—see Fig. 2.3 for an illustration—corresponding to the two-dimensional vector space $\mathcal{S}$, not in the real three-dimensional world. We will further denote the rotations around the μ-axis—$\boldsymbol{n}$ parallel to the μ-axis—of the Bloch sphere by $\hat{U}_\mu(\phi)$.

To see that the unitary operation $\hat{U}_{\boldsymbol{n}}(\phi)$ in (2.23) corresponds to a rotation, recall that the Pauli operators $\hat{S}^\mu$ are the spin angular momentum operators of spin 1/2. That is, they are the generators of rotations and satisfy the commutation relations

$$[\hat{S}^\mu, \hat{S}^\nu] = 2i \sum_\lambda \hat{S}^\lambda \epsilon_{\lambda\mu\nu}. \tag{2.24}$$

The connection of the unitary operator $\hat{U}_\lambda(\phi)$ to rotation is seen more explicitly in the equivalent relation

$$\hat{U}_\lambda(\phi)\hat{S}^\nu\hat{U}_\lambda^\dagger(\phi) = \sum_\mu \hat{S}^\mu [R_\lambda(\phi)]_{\mu\nu}\,, \tag{2.25}$$

where $R_\lambda(\phi)$ is the 3×3 orthogonal matrix describing the rotation of three-dimensional coordinates around the λ-axis by angle ϕ.

The Pauli operators are generators of the rotational transformations in a two-dimensional complex vector space, and hence satisfy the fundamental commutation relations of angular momentum operators (up to a normalization factor).

```
In[•]:= op = S[All]
Out[•]= {S^x, S^y, S^z}

In[•]:= in = Outer[HoldForm@*Commutator, op, op];
        out = ReleaseHold[in];
        Thread[Flatten[in] → Flatten[out]] // TableForm
Out[•]//TableForm=
        Commutator[S^x, S^x] → 0
        Commutator[S^x, S^y] → 2 i S^z
        Commutator[S^x, S^z] → -2 i S^y
        Commutator[S^y, S^x] → -2 i S^z
        Commutator[S^y, S^y] → 0
        Commutator[S^y, S^z] → 2 i S^x
        Commutator[S^z, S^x] → 2 i S^y
        Commutator[S^z, S^y] → -2 i S^x
        Commutator[S^z, S^z] → 0
```

In $\mathbb{R}^3$, any 3×3 rotation matrix can be decomposed into three factors

$$R = R_z(\alpha) R_y(\beta) R_z(\gamma)\,, \tag{2.26}$$

where α, β, γ are the so-called *Euler angles* and such a combination of rotations is called the *Euler rotation*. In the same manner, any unitary operator on $\mathcal{S}$ can also be written as

$$\hat{U} = e^{-i\phi_0} \hat{U}_z(\alpha) \hat{U}_y(\beta) \hat{U}_z(\gamma)\,, \tag{2.27}$$

that is, a combination of elementary "rotations" around the y- and z-axis and an additional overall phase shift. The unitary operator $\hat{U}(\alpha, \beta, \gamma) := \hat{U}_z(\alpha) \hat{U}_y(\beta) \hat{U}_z(\gamma)$ is called the Euler rotation in the two-dimensional vector space $\mathcal{S}$. Figure 2.3 illustrates an Euler rotation $\hat{U}(\pi/3, -\pi/3, \pi/4)$. It transforms $|v\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ (red dot in Fig. 2.3) is transformed to $|w\rangle = \hat{U}\,|v\rangle$ (blue dot in Fig. 2.3). The transformation is displayed by the blue arrow. The yellow dots and arrows depicts the transformations under $\hat{U}_z(\alpha)$, $\hat{U}_y(\beta)$, and $\hat{U}_z(\gamma)$, which combine to reproduce $\hat{U}$.

Consider a unitary operator represented by the following matrix.

```
In[•]:= mat = {
            {3 - I Sqrt[3], I - Sqrt[3]},
            {I + Sqrt[3], 3 + I Sqrt[3]}
          } / 4;
        mat // MatrixForm
Out[•]//MatrixForm=
        ⎛ 1/4 (3 - i √3)   1/4 (i - √3) ⎞
        ⎝ 1/4 (i + √3)     1/4 (3 + i √3) ⎠
```

This gives its expression in terms of the Pauli operators.

```
op = ExpressionFor[mat, S]
```

$$\frac{3}{4} + \frac{i\, S^x}{4} - \frac{1}{4} i \sqrt{3}\, S^y - \frac{1}{4} i \sqrt{3}\, S^z$$

```
Dagger[op] ** op
```

```
1
```

This gives the Euler angles of the unitary operator.

```
angs = TheEulerAngles[mat]
```

$$\left\{\frac{\pi}{3}, \frac{\pi}{3}, 0\right\}$$

Indeed, the Euler angles reproduces the original unitary operator.

```
new = QuissoEulerRotation[angs, S]
op - new
```

$$\frac{3}{4} + \frac{i\, S^x}{4} - \frac{1}{4} i \sqrt{3}\, S^y - \frac{1}{4} i \sqrt{3}\, S^z$$

```
0
```

2.2 Two-Qubit Gates

Let us next consider quantum logic gate operations acting on two qubits. Such operations are represented by 4×4 unitary matrices. We will see that any two-qubit gate operations can be decomposed into controlled-U gates. A controlled-U gate acts a unintary operator on one qubit depending on the logical state of the other qubit. A controlled-U gate on two qubits can be further decomposed into factors including only CNOT gate and single-qubit rotation gates. In this sense, the CNOT gate alone is sufficient for any two-qubit gate.

The controlled-U and CNOT gate have various interesting properties that make them useful in the implementation of quantum algorithms. In this section, we will first examine the basic properties of the CNOT gate, in particular, how it is used to generate an entanglement between two qubits. We then discuss the properties of the controlled-U gate and how to implement a controlled-U gate in terms of the CNOT gate and the single-qubit rotations. Finally, we discuss how an arbitrary two-qubit unitary operation can be decomposed into controlled-U gates.

2.2.1 CNOT, CZ, and SWAP

The CNOT or controlled-NOT gate is a quantum logic gate on two qubits that maps the logical basis states as

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes |c \oplus t\rangle \ ; \quad c, t \in \{0, 1\} \ , \tag{2.28}$$

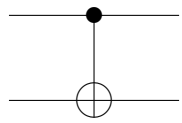
where the first qubit is typically called the *control qubit* (c) and the second qubit the *target qubit* (t). It has the following matrix representation in the logical basis

$$\mathrm{CNOT} \doteq \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{bmatrix}, \tag{2.29}$$

and expressed in terms of the Pauli operators on the control and target qubit as

$$\mathrm{CNOT} = \frac{1}{2}\left(1 + \hat{S}_c^z + \hat{S}_t^z - \hat{S}_c^z \hat{S}_t^x\right). \tag{2.30}$$

In a quantum circuit model, it is represented as the following circuit element:



, (2.31)

where the smaller filled circle indicates the dependence on the state of the control qubit and the circled-plus sign denotes the conditional NOT action on the target qubit.

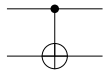
CNOT[*control*, *target*] denotes the CNOT gate in the quantum circuit model.

```
In[•]:= op = CNOT[S[1], S[2]]
Out[•]= CNOT[{S_1}, {S_2}]
```

This displays the quantum circuit model of the CNOT gate.

```
In[•]:= qc = QuantumCircuit[op]
```

Out[•]=



This is the explicit expression of the CNOT gate in terms of the Pauli operators.

```
In[•]:= op = ExpressionFor[qc]
Out[•]= 1/2 - 1/2 S_1^z S_2^x + S_1^z/2 + S_2^x/2
```

This is the matrix representation of the CNOT gate in the logical basis.

```
In[•]:= Matrix[op] // MatrixForm
Out[•]//MatrixForm=
(1 0 0 0)
(0 1 0 0)
(0 0 0 1)
(0 0 1 0)
```

This shows how the CNOT gate operates on the logical basis states.

```
In[•]:= in = Basis[S@{1, 2}];
        out = op ** in;
        Thread[in → out] // LogicalForm // TableForm
Out[•]//TableForm=
        |0_{s_1} 0_{s_2}⟩ → |0_{s_1} 0_{s_2}⟩
        |0_{s_1} 1_{s_2}⟩ → |0_{s_1} 1_{s_2}⟩
        |1_{s_1} 0_{s_2}⟩ → |1_{s_1} 1_{s_2}⟩
        |1_{s_1} 1_{s_2}⟩ → |1_{s_1} 0_{s_2}⟩
```

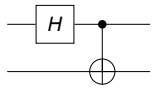
A simple yet important feature of CNOT is to copy the logical state of the control qubit to the target bit provided that the target bit is initially set to $|0\rangle$; or the reversed state when the target qubit is in $|1\rangle$. A vital implication is that CNOT generates an entangled state when the control qubit is in a superposition:

$$(|0\rangle\, c_0 + |1\rangle\, c_1) \otimes |0\rangle \mapsto |0\rangle \otimes |0\rangle\, c_0 + |1\rangle \otimes |1\rangle\, c_1. \tag{2.32}$$

Applying a single qubit rotation such the Hadamard gate on the control qubit prior to CNOT, one can thus generate entangled states from logical states. In this sense, such a circuit is called *quantum entangler circuit*.

As an example of the application of the CNOT gate, this shows an entangler quantum circuit.

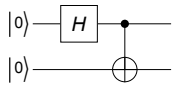
```
In[•]:= entangler = QuantumCircuit[S[1, 6], CNOT[S[1], S[2]]]
Out[•]=
```



For example, when the input state is $|0\rangle \otimes |0\rangle$, the outcome of the circuit is given by one of the so-called Bell states.

This demonstrates the generation of an entangled state from a product state.

```
In[•]:= new = QuantumCircuit[LogicalForm[Ket[S@{1, 2} → {0, 0}], S@{1, 2}], entangler]
        vec = ExpressionFor[new];
        vec // LogicalForm
Out[•]=
```




```
Out[•]= |0_{s_1} 0_{s_2}⟩/√2 + |1_{s_1} 1_{s_2}⟩/√2
```

In general, the product states in the logical basis are transformed to the Bell states as you can see in the demonstration below.

This lists the mapping between the standard tensor-product basis states and the Bell states.

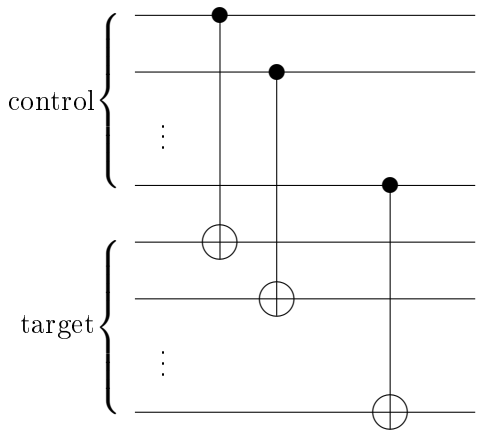


Figure 2.4: A quantum circuit model which makes a copy of logical state of the "control" quantum register to the "target" quantum register. The quantum circuit model transforms the logical basis states as $|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes |c \oplus t\rangle$, where c and t are b-bit strings.

```
In[•]:= bs = Basis@S@{1, 2};
        op = ExpressionFor[entangler];
        out = op ** bs;
        table = Thread[bs → out];
        table // LogicalForm // TableForm
```

Out[•]//TableForm=

$$\left|0_{S_1} 0_{S_2}\right\rangle \to \frac{\left|0_{S_1} 0_{S_2}\right\rangle}{\sqrt{2}} + \frac{\left|1_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}}$$
$$\left|0_{S_1} 1_{S_2}\right\rangle \to \frac{\left|0_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}} + \frac{\left|1_{S_1} 0_{S_2}\right\rangle}{\sqrt{2}}$$
$$\left|1_{S_1} 0_{S_2}\right\rangle \to \frac{\left|0_{S_1} 0_{S_2}\right\rangle}{\sqrt{2}} - \frac{\left|1_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}}$$
$$\left|1_{S_1} 1_{S_2}\right\rangle \to \frac{\left|0_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}} - \frac{\left|1_{S_1} 0_{S_2}\right\rangle}{\sqrt{2}}$$

One can further generalize the above procedure to generate an maximally entangled states between larger systems: Consider two quantum registers each of which consisting of n qubits. We call them the "control" and "target" register, respectively, for the reason to get clear below. Upon applying the CNOT gate on each pair of the corresponding qubits in the two registers as the quantum circuit model in Fig. 2.4, the logical basis states are transformed as

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes |c \oplus t\rangle \ . \tag{2.33}$$

The above association rule is *formally* the same as the one in (2.28) for a single pair of qubits. Here, however, $|c\rangle := |c_1\rangle \otimes |c_2\rangle \otimes \cdots \otimes |c_n\rangle$ and $|t\rangle := |t_1\rangle \otimes |t_2\rangle \otimes \cdots \otimes |t_n\rangle$, and $c \oplus t$ denotes the bit-wise exclusive OR (or XOR),

$$c \oplus t := (c_1 \oplus t_1, c_2 \oplus t_2, \ldots, c_n \oplus t_n), \tag{2.34}$$

for the n-bit strings $c \equiv (c_1, c_2, \ldots, c_n)$ and $t \equiv (t_1, t_2, \ldots, t_n)$. When the target register is prepared in the state $|0\rangle \equiv |0\rangle^{\otimes n}$, the logical state of the control qubit

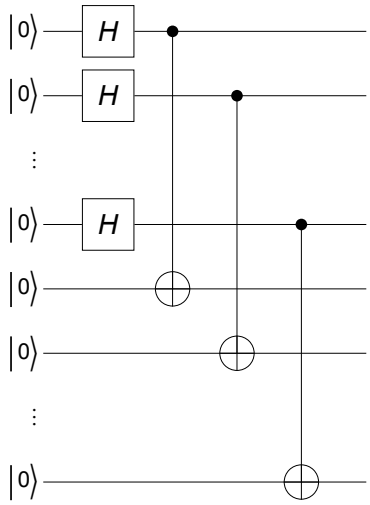


Figure 2.5: A quantum circuit model generating a maximally entangled states between two quantum registers each of which consisting of n qubits.

iis copied to the target state,

$$|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |x\rangle \tag{2.35}$$

under the set of CNOT gates on paired qubits. More interestingly, when the control regisgter is prepared in a superposition, say, $2^{-n/2}\sum_{x=0}^{2^n-1}|x\rangle$, and the target register in the state $|0\rangle \equiv |0\rangle$, the transformation in (2.33) makes a copy of each logical state of the control register to the target register, and leads to a maximally entangled state,

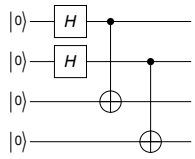
$$\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle \otimes |0\rangle \mapsto |\Phi\rangle := \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle \otimes |x\rangle \tag{2.36}$$

between the two regisgters (rather than single qubits). Since the superposition $2^{-n/2}\sum_{x=0}^{2^n-1}|x\rangle$ is obtained just by the Hadamard gates as shown in (2.18), the maximally entagnled state $|\Phi\rangle$ in (2.36) can be generated from the logical state $|0\rangle \otimes |0\rangle$ through the quantum circuit model illustrated in Fig. 2.5.

Here we want to generate a maximally entangled state between two quantum registers each of which consisting of two qubits.

```
In[•]:= qc = QuantumCircuit[LogicalForm[Ket[], S@{1, 2, 3, 4}],
          S[{1, 2}, 6], CNOT[S[1], S[3]], CNOT[S[2], S[4]],
          PlotRangePadding → 0, ImagePadding → {{36, 36}, {5, 5}}]
```

Out[•]=




```
In[•]:= out = ExpressionFor[qc];
        out // LogicalForm
```

$$\text{Out}[\bullet]= \frac{1}{2}\left|0_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle + \frac{1}{2}\left|0_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle + \frac{1}{2}\left|1_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle + \frac{1}{2}\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle$$

This example illustrate that the entanglement depends on the partition of the system. Indeed, the above system is a product state for the partition (1,3) and (2,4) qubits.

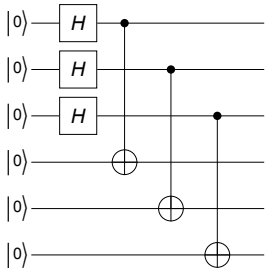
```
In[•]:= QuissoFactor[out]
```

$$\text{Out}[\bullet]= \frac{1}{2}\left(\left|0_{S_1}0_{S_3}\right\rangle + \left|1_{S_1}1_{S_3}\right\rangle\right) \otimes \left(\left|0_{S_2}0_{S_4}\right\rangle + \left|1_{S_2}1_{S_4}\right\rangle\right)$$

The above construction can be generalized for a pair of n-qubit systems.

```
In[•]:= n = 3;
        qc = QuantumCircuit[LogicalForm[Ket[], S@Range[2 n]],
          S[Range[n], 6], Sequence @@ Table[CNOT[S[j], S[n + j]], {j, 1, n}]]
```

Out[•]=




```
In[•]:= out = ExpressionFor[qc];
        out // LogicalForm
```

$$\text{Out}[\bullet]= \frac{\left|0_{S_1}0_{S_2}0_{S_3}0_{S_4}0_{S_5}0_{S_6}\right\rangle}{2\sqrt{2}} + \frac{\left|0_{S_1}0_{S_2}1_{S_3}0_{S_4}0_{S_5}1_{S_6}\right\rangle}{2\sqrt{2}} + \frac{\left|0_{S_1}1_{S_2}0_{S_3}0_{S_4}1_{S_5}0_{S_6}\right\rangle}{2\sqrt{2}} + \frac{\left|0_{S_1}1_{S_2}1_{S_3}0_{S_4}1_{S_5}1_{S_6}\right\rangle}{2\sqrt{2}} +$$
$$\frac{\left|1_{S_1}0_{S_2}0_{S_3}1_{S_4}0_{S_5}0_{S_6}\right\rangle}{2\sqrt{2}} + \frac{\left|1_{S_1}0_{S_2}1_{S_3}1_{S_4}0_{S_5}1_{S_6}\right\rangle}{2\sqrt{2}} + \frac{\left|1_{S_1}1_{S_2}0_{S_3}1_{S_4}1_{S_5}0_{S_6}\right\rangle}{2\sqrt{2}} + \frac{\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}1_{S_5}1_{S_6}\right\rangle}{2\sqrt{2}}$$

Quantum entanglement is a valuable resource in quantum information processing and quantum communication. The most popular example is quantum teleportation to be discussed in Section 4.1. Inherently, the features eluciated in (2.35) and (2.36)—and the related quantum circuit models in Figs. 2.4 and 2.5—will be frequently used in later parts of the book.

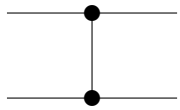
An interesting variant of CNOT gates is the so-called CZ or controlled-Z gate: It is a quantum logic gate on two qubits that maps the logical basis states as

$$\mathrm{CZ} : |c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes |t\rangle\, (-1)^{ct}\,. \tag{2.37}$$

The matrix representation of the CZ gate in the logical basis is given by

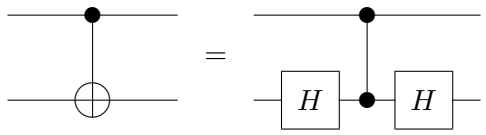
$$\mathrm{CZ} \doteq \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{bmatrix}. \tag{2.38}$$

As it is symmetric for the two qubits, the distinction of the control and taget qubit is meaningless. Accordingly, in the quantum circuit model, the gate is depicted by the following quantum circuit element



. (2.39)

The filled circles on both qubit lines–rather than a square box on either qubit—indicates that the bit values of the both qubits remain unchanged. Noting the identity $\hat{H}\hat{Z}\hat{H} = \hat{X}$, one can regard that the CZ gate is equal to the CNOT gate up to the Hadamard gate on the target qubit. The relation between the CNOT and CZ gate is expressed in the following quantum circuit model



. (2.40)

Depending on the Hamiltonian of particular physical system, the direct realization of the CNOT gate may be significantly difficult while the CZ gate is relatively easier to realize. In such a case, the identity (2.40) offers a straightforward workaround for the physical implementation of the CNOT gate.

The CZ (or controlled-Z) gate is a variant of the CNOT gate.

```
In[•]:= op = CZ[S[1], S[2]]
Out[•]= CZ[S1, S2]
```

This shows how it transforms the logical basis states.

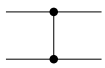
```
In[•]:= bs = Basis@S@{1, 2};
        out = op ** bs;
        Thread[bs → out] // LogicalForm // TableForm
Out[•]//TableForm=
        |0S1 0S2⟩ → |0S1 0S2⟩
        |0S1 1S2⟩ → |0S1 1S2⟩
        |1S1 0S2⟩ → |1S1 0S2⟩
        |1S1 1S2⟩ → - |1S1 1S2⟩
```

Here is the matrix representation of the CZ gate.

```
In[•]:= mat = Matrix[Elaborate@op];
        mat // MatrixForm
Out[•]//MatrixForm=
        ⎛1  0  0   0⎞
        ⎜0  1  0   0⎟
        ⎜0  0  1   0⎟
        ⎝0  0  0  -1⎠
```

This is the quantum circuit model of the CZ gate.

```
In[•]:= cz = QuantumCircuit[CZ[S[1], S[2]]]
Out[•]=
```

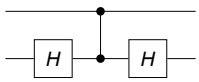


Note the following identity.

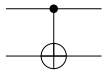
```
In[•]:= expr = HoldForm[S[1, 6] ** S[1, 3] ** S[1, 6] == S[1, 1]]
        ReleaseHold[expr] // Elaborate
Out[•]= S[1,H] ** S[1,z] ** S[1,H] == S[1,x]
Out[•]= True
```

It leads to the following relation between the CNOT gate and CZ gate.

```
In[•]:= new = QuantumCircuit[S[2, 6], cz, S[2, 6]]
Out[•]=
```




```
In[•]:= cnot = QuantumCircuit[CNOT[S[1], S[2]]]
Out[•]=
```



```
In[•]:= Elaborate[new - cnot]
Out[•]= 0
```

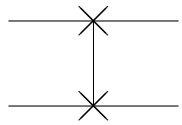
Another interesting two-qubit gate is the SWAP gate: The SWAP gate "swaps" the states of the two qubits, and maps the logical basis states as

$$\text{SWAP} : |x_1\rangle \otimes |x_2\rangle \mapsto |x_2\rangle \otimes |x_1\rangle \ . \tag{2.41}$$

As the states $|0\rangle \otimes |0\rangle$ and $|1\rangle \otimes |1\rangle$ are not altered by the operation, the matrix representation is given by

$$\text{SWAP} \doteq \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & 1 & 0 & \\ & & & 1 \end{bmatrix} . \tag{2.42}$$

In the qunatum circuit model, it is depicted as



(2.43)

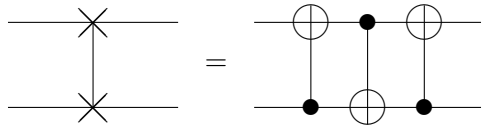
The SWAP gate can be implemented using the CNOT gate. To see this, first note that the simulaneous exchange of second and forth columns and rows of the matrix in (2.42) leads to

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{bmatrix}, \tag{2.44}$$

which is nothing but the matrix representation of the CNOT gate. On the other hand, the exchange of the second and forth columns and rows is described by the transformation matrix

$$\begin{bmatrix} 1 & & & \\ & 0 & & 1 \\ & & 1 & \\ & 1 & & 0 \end{bmatrix}, \tag{2.45}$$

which flips the bit values of the first qubit only when the second qubit is set to $|1\rangle$, and hence it corresponds to the CNOT gate with the second and first qubit as the control and target qubit, respectively. In short, the SWAP gate can be achieved by a combination of the CNOT gates as following



(2.46)

The SWAP gate exchanges the states of two qubits.

```
In[•]:= op = SWAP[S[1], S[2]]
Out[•]= SWAP[S1, S2]

In[•]:= bs = Basis@S@{1, 2};
        new = op ** bs;
        Thread[bs → new] // LogicalForm // TableForm
Out[•]//TableForm=
        |0S1 0S2⟩ → |0S1 0S2⟩
        |0S1 1S2⟩ → |1S1 0S2⟩
        |1S1 0S2⟩ → |0S1 1S2⟩
        |1S1 1S2⟩ → |1S1 1S2⟩
```

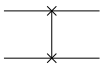
This is the matrix representation of the SWAP gate in the logical basis.

```
In[•]:= Matrix[Elaborate@op] // MatrixForm
Out[•]//MatrixForm=
```

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

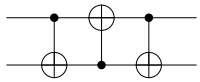
In the quantum circuit model, the SWAP gate is represented as following.

```
In[•]:= qc = QuantumCircuit[SWAP[S[1], S[2]]]
Out[•]=
```



The SWAP gate can be implemented by means of the CNOT gate.

```
In[•]:= new = QuantumCircuit[CNOT[S[1], S[2]], CNOT[S[2], S[1]], CNOT[S[1], S[2]]]
Out[•]=
```



```
In[•]:= Elaborate[qc - new]
Out[•]= 0
```

Interestingly, the SWAP gate itself is not universal, but the $\sqrt{\text{SWAP}}$ gate—the gate when squared equals to SWAP—is. That is, any quantum gate on a multi-qubit system can be implemented using $\sqrt{\text{SWAP}}$ and single-qubit rotations—see also Section 2.4 and Section 3.2.2. Indeed, one can combine the $\sqrt{\text{SWAP}}$ gate with single-qubit rotations to construct the CZ gate (Loss & DiVincenzo, 1998). As discussed above, the CZ gate just requires two more Hadamard gates to implement the CNOT gate and hence is universal.

Let us construct the CZ gate with the $\sqrt{\text{SWAP}}$ gate. This is the matrix representation of the the $\sqrt{\text{SWAP}}$ gate.

```
In[•]:= mat = MatrixPower[Matrix@Elaborate@SWAP[S[1], S[2]], 1 / 2];
        mat // MatrixForm
Out[•]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

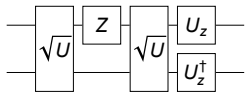
This is an explicit operator expression of the $\sqrt{\text{SWAP}}$ gate in terms of the Pauli operators.

```
In[•]:= sqrtSWAP = {ExpressionFor[mat, S@{1, 2}], , "Label" → "√U"}
Out[•]= {(3/4 + I/4) + (1/4 - I/4) S1^z S2^z + (1/2 - I/2) S1^+ S2^- + (1/2 - I/2) S1^- S2^+, Null, Label → √U}
```

This is a quantum circuit model to construct the CZ gate from the $\sqrt{\text{SWAP}}$ gate and single-qubit gates. In this diagram, $\sqrt{U}$ denotes the $\sqrt{\text{SWAP}}$ gate and U_z the rotation around the z-axis by angle $\pi/2$.

```
In[•]:= qc = QuantumCircuit[sqrtSWAP, S[1, 3], sqrtSWAP,
           {Rotation[Pi / 2, S[1, 3]], Rotation[-Pi / 2, S[2, 3], "Label" → "U_z^†"]}]
```

Out[•]=



To check if it indeeds implement the CZ gate, take a look at the matrix representation of the quantum circuit model.

```
In[•]:= Matrix[qc] // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

2.2.2 Controlled-U Gate

Consider two qubits, again called as the control and target qubit. Let $\hat{U}$ be a unitary operator on the target qubit. The controlled-U gate is a unitary operator on the two-qubit Hilbert space defined analogously to CNOT by

$$\mathrm{Ctrl}(\hat{U}) : |c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes \hat{U}^c |t\rangle \ , \tag{2.47}$$

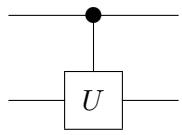
or equivalently, by

$$\mathrm{Ctrl}(\hat{U}) := |0\rangle\langle 0| \otimes \hat{I} + |1\rangle\langle 1| \otimes \hat{U} \, . \tag{2.48}$$

If the conrol qubit is in $|0\rangle$, then it does nothing. On the other hand, if the control qubit is in $|1\rangle$, the unitary operator $\hat{U}$ acts on the target qubit. In analogy with the CNOT gate, the matrix representation of the controlled-U gate is thus given by

$$\mathrm{Ctrl}(\hat{U}) \doteq \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & U_{11} & U_{12} \\ & & U_{21} & U_{22} \end{bmatrix} \ , \tag{2.49}$$

where U is the matrix representation of $\hat{U}$. In the quantum circuit model, a controlled-U gate is depicted as



(2.50)

The filled circle connected to the quantum circuit element on the target qubit indicates the conditional operation of the element.

Consider a single-qubit rotation acting on the qubit `S[2,None]`. It is a rotation around the y-axis by angle ϕ.

```
In[•]:= U = Rotation[ϕ, S[2, 2], "Label" → "U"];
        U // Elaborate // Matrix // MatrixForm
```

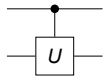
Out[•]//MatrixForm=

$$\begin{pmatrix} \text{Cos}\left[\frac{\phi}{2}\right] & -\text{Sin}\left[\frac{\phi}{2}\right] \\ \text{Sin}\left[\frac{\phi}{2}\right] & \text{Cos}\left[\frac{\phi}{2}\right] \end{pmatrix}$$

This shows the controlled-U gate.

```
In[•]:= qc = QuantumCircuit[ControlledU[S[1], U]]
```

Out[•]=



This is the explicit expression of the controlled-U gate operation in terms of the Pauli operators.

```
In[•]:= op = Elaborate[qc]
```

Out[•]=

$$\text{Cos}\left[\frac{\phi}{4}\right]^2 + S_1^z \,\text{Sin}\left[\frac{\phi}{4}\right]^2 + \frac{1}{2}\, i\, S_1^z S_2^y \,\text{Sin}\left[\frac{\phi}{2}\right] - \frac{1}{2}\, i\, S_2^y \,\text{Sin}\left[\frac{\phi}{2}\right]$$

The controlled-U gate maps the logical basis states as following.

```
In[•]:= bs = Basis[S@{1, 2}];
        bs // LogicalForm
        out = op ** bs;
        out // LogicalForm
```

Out[•]= $\left\{\left|0_{S_1}0_{S_2}\right\rangle,\ \left|0_{S_1}1_{S_2}\right\rangle,\ \left|1_{S_1}0_{S_2}\right\rangle,\ \left|1_{S_1}1_{S_2}\right\rangle\right\}$

Out[•]= $\left\{\left|0_{S_1}0_{S_2}\right\rangle,\ \left|0_{S_1}1_{S_2}\right\rangle,\ \text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_1}0_{S_2}\right\rangle + \left|1_{S_1}1_{S_2}\right\rangle \,\text{Sin}\left[\frac{\phi}{2}\right],\right.$

$\left.\text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_1}1_{S_2}\right\rangle - \left|1_{S_1}0_{S_2}\right\rangle \,\text{Sin}\left[\frac{\phi}{2}\right]\right\}$

To make the mapping clearer, this tabulates the above result.

```
In[•]:= new = QuissoFactor[#, S[1]] & /@ out;
        Thread[bs → new] // LogicalForm // TableForm
```

Out[•]//TableForm=

$$\begin{aligned}
&\left|0_{S_1}0_{S_2}\right\rangle \to \left|0_{S_1}\right\rangle \otimes \left|0_{S_2}\right\rangle \\
&\left|0_{S_1}1_{S_2}\right\rangle \to \left|0_{S_1}\right\rangle \otimes \left|1_{S_2}\right\rangle \\
&\left|1_{S_1}0_{S_2}\right\rangle \to \left|1_{S_1}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|0_{S_2}\right\rangle + \left|1_{S_2}\right\rangle \,\text{Sin}\left[\frac{\phi}{2}\right]\right) \\
&\left|1_{S_1}1_{S_2}\right\rangle \to \left|1_{S_1}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_2}\right\rangle - \left|0_{S_2}\right\rangle \,\text{Sin}\left[\frac{\phi}{2}\right]\right)
\end{aligned}$$

Let us take a look at the mapping more closely. When the first qubit is set to `Ket[0]`, it does nothing.

```
In[•]:= Let[Complex, c]
        vec = Ket[] × c[0] + Ket[S[2] → 1] × c[2];
        LogicalForm[vec, S@{1, 2}]
```

Out[•]= $c_0\ \left|0_{S_1}0_{S_2}\right\rangle + c_2\ \left|0_{S_1}1_{S_2}\right\rangle$

```
In[•]:= new = op ** vec;
        LogicalForm[new, S@{1, 2}]
```

Out[•]= $c_0\ \left|0_{S_1}0_{S_2}\right\rangle + c_2\ \left|0_{S_1}1_{S_2}\right\rangle$

When the control qubit -- the first qubit in this case -- is set to `Ket[1]`, it operates the unitary operator on the second qubit.

```
In[•]:= vec = Ket[S[1] → 1] ** (Ket[] × c[0] + Ket[S[2] → 1] × c[2]);
        LogicalForm[vec, S@{1, 2}]
Out[•]= c0 |1S1 0S2⟩ + c2 |1S1 1S2⟩

In[•]:= new = op ** vec;
        LogicalForm[new, S@{1, 2}]
Out[•]= |1S1 1S2⟩ (c2 Cos[ϕ/2] + c0 Sin[ϕ/2]) + |1S1 0S2⟩ (c0 Cos[ϕ/2] - c2 Sin[ϕ/2])
```

When the control qubit is in a superposition, the resulting state is an entangled state in general.

```
In[•]:= vec = Ket[] + Ket[S[1] → 1];
        LogicalForm[vec, S@{1, 2}]
Out[•]= |0S1 0S2⟩ + |1S1 0S2⟩

In[•]:= new = op ** vec;
        LogicalForm[new, S@{1, 2}]
Out[•]= |0S1 0S2⟩ + Cos[ϕ/2] |1S1 0S2⟩ + |1S1 1S2⟩ Sin[ϕ/2]
```

An important aspect of the controlled-U operator is that it induces relative phase shifts on the *control* qubits when the target is prepared in an eigenstate of $\hat{U}$. At a first glace, it may sound counter intuitive as the definition in (2.47) seems to indicate that it changes the target qubit depending on the state of the control qubit and leaves the latter intact. This is another feature distinguishing quantum gates from the classical counterparts. Let us take a closer look to see how it works. Prepare the control register in a superposition $|\psi\rangle = |0\rangle + |1\rangle$ and the target register in a eigenstate $|u\rangle$ of $\hat{U}$ with eigenvalue $e^{i\phi}$. The controlled-U gate transforms the state to

$$\begin{aligned} |\psi\rangle \otimes |u\rangle &\to |0\rangle \otimes |u\rangle + |1\rangle \otimes \hat{U}\,|u\rangle \\ &= |0\rangle \otimes |u\rangle + |1\rangle \otimes |u\rangle\, e^{i\phi} = \left(|0\rangle + |1\rangle\, e^{i\phi}\right) \otimes |u\rangle\,. \end{aligned} \tag{2.51}$$

This feature is extended to multi-qubit controlled-U gates and plays a crucial role in many quantum algorithms. In particular, the quantum phase estimation algorithm (Section 4.4) is a direct consequence of this feature.

When the target qubit is set to an eigenstate of the unitary operator, it does not change but the control qubit acquires the phase factor given by the eigenvalue of the target state.

```
In[•]:= vec = (Ket[] + Ket[S[1] → 1]) ** (Ket[] - I Ket[S[2] → 1]);
        LogicalForm[QuissoFactor@vec, S@{1, 2}]
Out[•]= (|0S1⟩ + |1S1⟩) ⊗ (|0S2⟩ - i |1S2⟩)

In[•]:= new = op ** vec // TrigToExp;
        LogicalForm[QuissoFactor@new, S@{1, 2}]
Out[•]= (|0S1⟩ + e^(iϕ/2) |1S1⟩) ⊗ (|0S2⟩ - i |1S2⟩)
```

Example 6 Suppose that a qubit is known to be in one of the two eigenstates of the unitary operator

$$\hat{U} = \hat{\sigma}^0 \cos(\phi/2) - i\hat{\sigma}^x \sin(\phi/2) \tag{2.52}$$

with the angle ϕ *known*. Construct a quantum circuit model to figure out the unknown state using an additional qubit.

Hint: Use a controlled-U gate to acquire the one-bit information. This is a simplified version of the quantum phase estimation procedure (see Section 4.4), but it can be worked out without resorting to it.

Solution: The eigenstates of `U` are the same as those of `Pauli[1]`.

```
In[•]:= U = Rotation[ϕ, S[1, 1]] // Elaborate
```

$$\text{Out[•]=} \quad \text{Cos}\left[\frac{\phi}{2}\right] - \text{i}\, S_1^x \,\text{Sin}\left[\frac{\phi}{2}\right]$$

```
In[•]:= vec = S[1, 6] ** Ket[];
        vec // LogicalForm
```

$$\text{Out[•]=} \quad \frac{\left|0_{S_1}\right\rangle}{\sqrt{2}} + \frac{\left|1_{S_1}\right\rangle}{\sqrt{2}}$$

```
In[•]:= U ** vec // LogicalForm // TrigToExp // Simplify
```

$$\text{Out[•]=} \quad \frac{e^{-\frac{i\phi}{2}}\left(\left|0_{S_1}\right\rangle + \left|1_{S_1}\right\rangle\right)}{\sqrt{2}}$$

```
In[•]:= vec = S[1, 6] ** Ket[S[1] → 1];
        vec // LogicalForm
```

$$\text{Out[•]=} \quad \frac{\left|0_{S_1}\right\rangle}{\sqrt{2}} - \frac{\left|1_{S_1}\right\rangle}{\sqrt{2}}$$

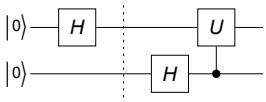
```
In[•]:= U ** vec // LogicalForm // TrigToExp // Simplify
```

$$\text{Out[•]=} \quad \frac{e^{\frac{i\phi}{2}}\left(\left|0_{S_1}\right\rangle - \left|1_{S_1}\right\rangle\right)}{\sqrt{2}}$$

The ancillary qubit takes a relative phase shift depending on in which eigenstate the native qubit is.

```
In[•]:= Let[Real, ϕ];
        qc = QuantumCircuit[LogicalForm[Ket[], S@{1, 2}], S[1, 6], "Separator",
          S[2, 6], ControlledU[S[2], Rotation[ϕ, S[1, 1]], "Label" → "U"]]
```

Out[•]=



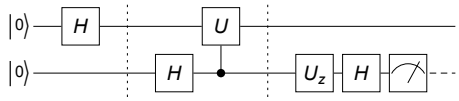

```
In[•]:= out = ExpressionFor[qc] // TrigToExp;
        LogicalForm[QuissoFactor@out, S@{1, 2}]
```

$$\text{Out[•]=} \quad \frac{1}{2} e^{-\frac{i\phi}{2}}\left(\left|0_{S_1}\right\rangle + \left|1_{S_1}\right\rangle\right) \otimes \left(e^{\frac{i\phi}{2}}\left|0_{S_2}\right\rangle + \left|1_{S_2}\right\rangle\right)$$

Make a basis change to detect it.

```
In[•]:= qc = QuantumCircuit[LogicalForm[Ket[], S@{1, 2}], S[1, 6], "Separator",
          S[2, 6], ControlledU[S[2], Rotation[ϕ, S[1, 1]], "Label" → "U"],
          "Separator", Rotation[ϕ / 2, S[2, 3]], S[2, 6], Measurement[S[2]]]
```

Out[•]=




```
In[•]:= out = ExpressionFor[qc] // TrigToExp;
        LogicalForm[QuissoFactor@out, S@{1, 2}]
```

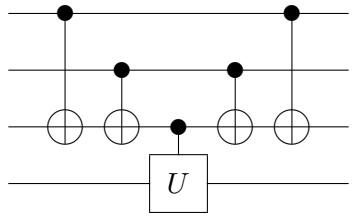
Out[•]=

$$\frac{e^{-\frac{i\phi}{4}}\left(\left|0_{S_1}0_{S_2}\right\rangle + \left|1_{S_1}0_{S_2}\right\rangle\right)}{\sqrt{2}}$$

Combining the CNOT gate and the controlled-U gate, one can achieve a variety of conditional gate operations: For example, consider a system consisting of n control qubits and one target qubit, and suppose that you want to operate a unitary gate $\hat{U}$ on the target qubit when one and only one of the control qubits is set to $|1\rangle$,

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes \hat{U}^{c_1 \oplus \cdots \oplus c_n} |t\rangle \ . \tag{2.53}$$

First operate the CNOT gates with the first $(n-1)$ qubits in the control register consecutively as the control qubit and the last qubit in the control register as the target qubit. It transforms the nth qubit to $|c_1 \oplus \cdots \oplus c_n\rangle$—Problem 6. Then, by applying the controlled-U gate controlled by the nth qubit, the desired operation is implemented. To return the control qubits back to the original state, operate the CNOT gates in the reverse order. Overall, the following quantum circuit model implements the conditional operation



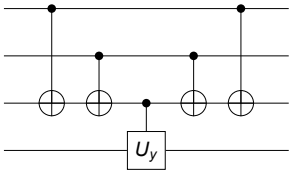
(2.54)

for the case of $n = 3$. This method is used for the implementation of multi-qubit controlled-U gate based on the Gray code—see Section 2.3.

This shows a quantum circuit model conditionally operating the logic gate U on the target qubit.

```
In[•]:= $n = 3;
        cc = Table[CNOT[S[j], S[$n]], {j, 1, $n - 1}];
        op = Rotation[ϕ, S[$n + 1, 2]];
        cU = ControlledU[S[$n], op];
        qc = QuantumCircuit[Sequence @@ Flatten@{cc, cU, Reverse@cc}]
```

Out[•]=



This shows how the above quantum circuit model maps the logical basis states. It affects the target qubit only when $c_1 \oplus c_2 \oplus c_3 = 1$.

```
In[•]:= ss = S[Range[$n], None];
        bs = Basis[S@Range[$n + 1]];
        out = Elaborate[qc] ** bs;
        new = QuissoFactor[#, ss] & /@ out;
        Thread[bs → new] // LogicalForm // TableForm
```

Out[•]//TableForm=

$$\left|0_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle \to \left|0_{S_1}0_{S_2}0_{S_3}\right\rangle \otimes \left|0_{S_4}\right\rangle$$
$$\left|0_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle \to \left|0_{S_1}0_{S_2}0_{S_3}\right\rangle \otimes \left|1_{S_4}\right\rangle$$
$$\left|0_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle \to \left|0_{S_1}0_{S_2}1_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|0_{S_4}\right\rangle + \left|1_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|0_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle \to \left|0_{S_1}0_{S_2}1_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_4}\right\rangle - \left|0_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|0_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle \to \left|0_{S_1}1_{S_2}0_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|0_{S_4}\right\rangle + \left|1_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|0_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle \to \left|0_{S_1}1_{S_2}0_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_4}\right\rangle - \left|0_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|0_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle \to \left|0_{S_1}1_{S_2}1_{S_3}\right\rangle \otimes \left|0_{S_4}\right\rangle$$
$$\left|0_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle \to \left|0_{S_1}1_{S_2}1_{S_3}\right\rangle \otimes \left|1_{S_4}\right\rangle$$
$$\left|1_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle \to \left|1_{S_1}0_{S_2}0_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|0_{S_4}\right\rangle + \left|1_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|1_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle \to \left|1_{S_1}0_{S_2}0_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_4}\right\rangle - \left|0_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|1_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle \to \left|1_{S_1}0_{S_2}1_{S_3}\right\rangle \otimes \left|0_{S_4}\right\rangle$$
$$\left|1_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle \to \left|1_{S_1}0_{S_2}1_{S_3}\right\rangle \otimes \left|1_{S_4}\right\rangle$$
$$\left|1_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle \to \left|1_{S_1}1_{S_2}0_{S_3}\right\rangle \otimes \left|0_{S_4}\right\rangle$$
$$\left|1_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle \to \left|1_{S_1}1_{S_2}0_{S_3}\right\rangle \otimes \left|1_{S_4}\right\rangle$$
$$\left|1_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle \to \left|1_{S_1}1_{S_2}1_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|0_{S_4}\right\rangle + \left|1_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$
$$\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle \to \left|1_{S_1}1_{S_2}1_{S_3}\right\rangle \otimes \left(\text{Cos}\left[\frac{\phi}{2}\right]\ \left|1_{S_4}\right\rangle - \left|0_{S_4}\right\rangle \text{Sin}\left[\frac{\phi}{2}\right]\right)$$

How can you implement a controlled-U gate? The operation involves only two qubits, and in principle, it should be possible to implement any specific controlled-U gate. However, as it will get clearer in Chapter 3, the requirements for physical implementation of two-qubit gates is far more difficult to fulfill on realistic systems than single-qubit gates. Fortunately, any controlled-U gate can be implemented using only CNOT gate and single-qubit gates This is one of the basic steps in an establishment of the universal quantum computation.

Let $\hat{U}$ a unitary gate on the second (target) qubit controlled by the first (control) qubit. Suppose that $\hat{U}$ has Euler angles α, β, and γ and an additional

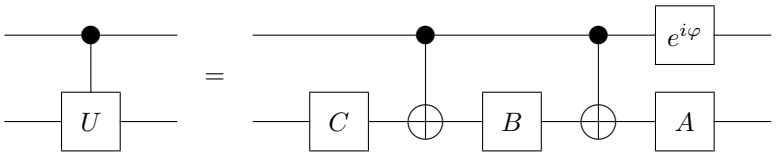
phase factor $e^{i\varphi}$ [see (2.27)], $\hat{U} = e^{i\varphi}\hat{U}_z(\alpha)\hat{U}_y(\beta)\hat{U}_z(\gamma)$. Then one can find three unitary operators $\hat{A}$, $\hat{B}$, and $\hat{C}$ such that

$$\hat{U} = e^{i\varphi}\hat{A}\hat{X}\hat{B}\hat{X}\hat{C}\,, \quad \hat{A}\hat{B}\hat{C} = \hat{I}\,, \tag{2.55}$$

where $\hat{X}$ is the Pauli X operator. More explicitly, one common choice is

$$\hat{A} = \hat{U}_z(\alpha)\hat{U}_y(\beta/2)\,, \tag{2.56a}$$
$$\hat{B} = \hat{U}_y(-\beta/2)\hat{U}_z(-(\alpha+\gamma)/2)\,, \tag{2.56b}$$
$$\hat{C} = \hat{U}_z(-(\alpha-\gamma)/2). \tag{2.56c}$$

Since $\hat{U}_{y/z}(\phi) = \hat{X}\hat{U}_{y/z}(-\phi)\hat{X}$ for any ϕ, the above choice satisfies the desired properties in (2.55). The properties imply that the controlled-U gate can be implemented as in the following quantum circuit model



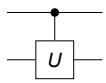
, (2.57)

where the last gate on the first qubit is the *relative* phase shift by φ, $|0\rangle\langle 0| + |1\rangle\langle 1|\, e^{i\varphi}$. Indeed, when the control qubit is in $|0\rangle$, the two CNOT gates in the middle do nothing, and the combined operator $\hat{A}\hat{B}\hat{C}$ on the target qubit ends up with the identity operator. With the control qubit in $|1\rangle$, on the other hand, the two CNOT gates are operational and the overall operator on the target qubit becomes $\hat{A}\hat{X}\hat{B}\hat{X}\hat{C} = e^{-i\varphi}\hat{U}$, where the phase factor is cancel by the opposite phase from the control qubit.

Consider a controlled-U gate.

```
In[•]:= matU = RandomUnitary[2];
        matU // MatrixForm
Out[•]//MatrixForm=
        ( -0.275548 + 0.720471 i   -0.630417 - 0.0870079 i )
        ( -0.622038 - 0.134405 i    0.0118329 + 0.771275 i )

In[•]:= opU = ExpressionFor[matU, S[2]];
        qc1 = QuantumCircuit[ControlledU[S[1], opU, "Label" → "U"]]
Out[•]=
```



For the decomposition, first find the Euler angles of the unitary operator.

```
In[•]:= detU = Det[matU];
        vphi = Arg[detU] / 2;
        {a, b, c} = TheEulerAngles[matU / Sqrt[detU]]
Out[•]= {1.41831, 1.37962, 4.48425}
```

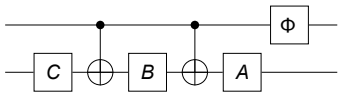
From the Euler angles, choose the component operators.

```
opA = EulerRotation[{a, b / 2, 0}, S[2], "Label" → "A"];
opB = EulerRotation[{0, -b / 2, -(a + c) / 2}, S[2], "Label" → "B"];
opC = EulerRotation[{0, 0, -(a - c) / 2}, S[2], "Label" → "C"];
opD = Phase[vphi, S[1]];
```

Finally, construct the equivalent quantum circuit model.

```
In[•]:= qc2 = QuantumCircuit[opC, CNOT[S[1], S[2]], opB, CNOT[S[1], S[2]], opA, opD]
```

Out[•]=



Check the result.

```
In[•]:= Elaborate[qc1 - qc2] // Chop
Out[•]= 0
```

2.2.3 General Unitary Gate

Here we decompose an arbitray two-qubit unitary gate into factors of controlled-U gates only. This is a remarkable advantage when one tries to build a quantum computer as you can just focus on how to realize the CNOT gate and single-qubit gates. Moreover, the same idea eventually leads to the proof of the universal quantum computation on a larger system, which will be discussed in Section 2.4.

Consider an aribtrary two-qubit unitary operator $\hat{U}$. In the logical basis, it is represented by a unitary matrix

$$U = \begin{bmatrix} U_{11} & U_{12} & U_{13} & U_{14} \\ U_{21} & U_{22} & U_{23} & U_{24} \\ U_{31} & U_{32} & U_{33} & U_{34} \\ U_{41} & U_{42} & U_{43} & U_{44} \end{bmatrix}. \tag{2.58}$$

To break the unitary operation $\hat{U}$ down into more elementary quantum logic gates, we make use of *two-level unitary transformations*. A two-level unitary transformation is a unitary operation the matrix representation of which acts only on the two columns and rows of other matrices or column or row vectors. The discriptive word "two-level" should not confused with "two-qubit". A two-level transformation acts on multiple qubits, but it just transforms only two rows or

columns at a time in the representation. For example, consider a two-level unitary trasformation of the form

$$T_1 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \tilde{U}_{13}^* & \tilde{U}_{14} \\ & & \tilde{U}_{14}^* & -\tilde{U}_{13} \end{bmatrix}, \tag{2.59}$$

where $\tilde{U}_{ij} \propto U_{ij}$ with normalization factor (unspecified) such that $T_1^\dagger T_1 = T_1 T_1^\dagger = I$. When it multiplies U from the right, it does not change the first two columns of U. It only alters the last two columns; hence the name two-level transformation. The elements in the lower-right subblock of T_1 have been chosen so that it the first element of the last column is canceled:

$$UT_1 = \begin{bmatrix} U_{11} & U_{12} & U'_{13} & 0 \\ U_{21} & U_{22} & U'_{23} & U'_{24} \\ U_{31} & U_{32} & U'_{33} & U'_{34} \\ U_{41} & U_{42} & U'_{43} & U'_{44} \end{bmatrix}. \tag{2.60}$$

Now take another two-level unitary transformation, this time, of the form

$$T_2 = \begin{bmatrix} 1 & & & \\ & \tilde{U}_{12}^* & \tilde{U}'_{13} & \\ & \tilde{U}'^*_{13} & -\tilde{U}_{12} & \\ & & & 1 \end{bmatrix}. \tag{2.61}$$

It removes U'_{13} while keeeping the first and last column as they are:

$$UT_1T_2 = \begin{bmatrix} U_{11} & U''_{12} & 0 & 0 \\ U_{21} & U''_{22} & U''_{23} & U'_{24} \\ U_{31} & U''_{32} & U''_{33} & U'_{34} \\ U_{41} & U''_{42} & U''_{43} & U'_{44} \end{bmatrix}. \tag{2.62}$$

Similarly, we go further with the twol-level unitary matrix

$$T_2 = \begin{bmatrix} \tilde{U}_{12}^* & \tilde{U}'_{13} & & \\ \tilde{U}'^*_{13} & -\tilde{U}_{12} & & \\ & & 1 & \\ & & & 1 \end{bmatrix}. \tag{2.63}$$

to remove the element U''_{12} and get

$$UT_1T_2T_3 = \begin{bmatrix} U'''_{11} & 0 & 0 & 0 \\ 0 & U'''_{22} & U''_{23} & U'_{24} \\ 0 & U'''_{32} & U''_{33} & U'_{34} \\ 0 & U'''_{42} & U''_{43} & U'_{44} \end{bmatrix}. \tag{2.64}$$

At this stage, all elements except for the first of the first column vanish —$U_{21}''' = U_{31}''' = U_{41}''' = 0$—because the product $UT_1T_2T_3$ is a unitary matrix. Repeating the procedure, we can remove all off-diagonal elements to get

$$UT_1T_2 \ldots T_L = I. \tag{2.65}$$

As T_j are all unitary, it enables us to rewrite U in the combination of two-level unitary transformation

$$U = T_L^\dagger \ldots T_2^\dagger T_1^\dagger. \tag{2.66}$$

Consider a Hermitian operator on a two-qubit system. Physically, it corresponds to a transverse-field Ising model.

```
In[•]:= H = S[1, 1] ** S[2, 1] + S[1, 2] + S[2, 2] + S[1, 3] + S[2, 3]
        matH = Matrix[H];
        matH // MatrixForm
```

Out[•]= $S_1^x S_2^x + S_1^y + S_1^z + S_2^y + S_2^z$

Out[•]//MatrixForm=

$$\begin{pmatrix} 2 & -i & -i & 1 \\ i & 0 & 1 & -i \\ i & 1 & 0 & -i \\ 1 & i & i & -2 \end{pmatrix}$$

This is the unitary operator generated by the above Hermitian operator.

```
In[•]:= U = MultiplyExp[-I H Pi / 4]
```

Out[•]= $e^{-\frac{1}{4} i \pi \left(S_1^x S_2^x + S_1^y + S_1^z + S_2^y + S_2^z\right)}$

This shows the matrix representation of `U` in the logical basis.

```
In[•]:= matU = Matrix[U] // Simplify;
        matU // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -\frac{\frac{1}{2}+\frac{5i}{6}}{\sqrt{2}} & -\frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}} & -\frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}} & \frac{\frac{1}{2}-\frac{i}{2}}{\sqrt{2}} \\ \frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}} & \frac{\frac{1}{2}+\frac{i}{6}}{\sqrt{2}} & -\frac{\frac{1}{2}+\frac{5i}{6}}{\sqrt{2}} & -\frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}} \\ \frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}} & -\frac{\frac{1}{2}+\frac{5i}{6}}{\sqrt{2}} & \frac{\frac{1}{2}+\frac{i}{6}}{\sqrt{2}} & -\frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}} \\ \frac{\frac{1}{2}-\frac{i}{2}}{\sqrt{2}} & \frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}} & \frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}} & -\frac{\frac{1}{2}-\frac{i}{2}}{\sqrt{2}} \end{pmatrix}$$

This shows the decomposition of the unitary matrix into two-level matrices (displaying the first three elements). For the purpose of efficiency, the resulting list is given in terms of `TwoLevelU`.

```
In[•]:= twl = TwoLevelDecomposition[matU] // Simplify;
        twl[[ ;; 3]]
```

Out[•]= $\Big\{$`TwoLevelU`$\left[\{\{1, 0\}, \{0, 1\}\}, \{3, 4\}, 4\right]$,

`TwoLevelU`$\left[\left\{\left\{\frac{\frac{1}{2}-\frac{3i}{2}}{\sqrt{7}}, -\frac{\frac{3}{2}+\frac{3i}{2}}{\sqrt{7}}\right\}, \left\{\frac{\frac{3}{2}-\frac{3i}{2}}{\sqrt{7}}, \frac{\frac{1}{2}+\frac{3i}{2}}{\sqrt{7}}\right\}\right\}, \{3, 4\}, 4\right]$,

`TwoLevelU`$\left[\left\{\left\{-\frac{1-3i}{\sqrt{38}}, \sqrt{\frac{14}{19}}\right\}, \left\{-\sqrt{\frac{14}{19}}, -\frac{1+3i}{\sqrt{38}}\right\}\right\}, \{2, 3\}, 4\right]\Big\}$

For a more intuitive form, you can convert `TwoLevelU` into the normal matrix form using `Matrix`. Here the first three are shown.

```
In[•]:= MatrixForm /@ Matrix /@ twl[[ ;; 3]]
```

Out[•]=

$$\left\{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{\frac{1}{2}-\frac{3i}{2}}{\sqrt{7}} & -\frac{\frac{3}{2}+\frac{3i}{2}}{\sqrt{7}} \\ 0 & 0 & \frac{\frac{3}{2}-\frac{3i}{2}}{\sqrt{7}} & \frac{\frac{1}{2}+\frac{3i}{2}}{\sqrt{7}} \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1-3i}{\sqrt{38}} & \sqrt{\frac{14}{19}} & 0 \\ 0 & -\sqrt{\frac{14}{19}} & -\frac{1+3i}{\sqrt{38}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\right\}$$

Indeed, they reconstruct the original matrix.

```
In[•]:= new = Dot @@ Matrix /@ twl;
        matU - new // Simplify // MatrixForm
```

Out[•]//MatrixForm=

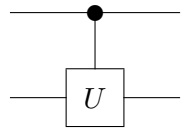
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

We are still to express the two-level unitary transformation in terms of a controlled-U gate: The two-level unitary matrix—for example, T_1 in (2.59)—of the form

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & U_{11} & U_{12} \\ & & U_{21} & U_{22} \end{bmatrix}, \tag{2.67}$$

is already in the form of the matrix representation of a controlled-U gate in (2.49), and just correponds to a single controlled-U:

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & U_{11} & U_{12} \\ & & U_{21} & U_{22} \end{bmatrix} = \quad . \tag{2.68}$$



Consider a two-level matrix of the form.

```
In[•]:= matU = TheHadamard[];
        code = TwoLevelU[matU, {3, 4}, 4];
        full = Matrix[code];
        full // MatrixForm
```

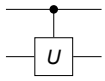
Out[•]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

It corresponds to a single controlled-U gate.

```
In[•]:= ctrlU = GrayTwoLevelU[matU, {3, 4}, S@{1, 2}];
        QuantumCircuit[ctrlU]
```

Out[•]=

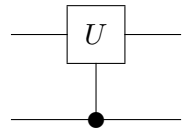


A two-level unitary matrix of the form

$$\begin{bmatrix} 1 & & & \\ & U_{11} & & U_{12} \\ & & 1 & \\ & U_{21} & & U_{22} \end{bmatrix} \tag{2.69}$$

also corresponds to a single controlled-U gate with the control and target qubit exchanged—here the first qubit is the target qubit and the second the control qubit:

$$\begin{bmatrix} 1 & & & \\ & U_{11} & & U_{12} \\ & & 1 & \\ & U_{21} & & U_{22} \end{bmatrix} = \tag{2.70}$$



.

Consider a two-level matrix of the form.

```
In[•]:= matU = TheHadamard[];
        code = TwoLevelU[matU, {2, 4}, 4];
        full = Matrix[code];
        full // MatrixForm
```

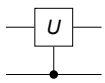
Out[•]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

It corresponds to a single controlled-U gate with the control and target qubit exchanged.

```
In[•]:= ctrlU = GrayTwoLevelU[matU, {2, 4}, S@{1, 2}];
        QuantumCircuit[ctrlU]
```

Out[•]=

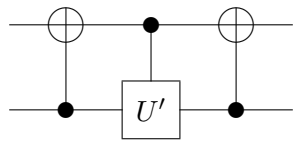


More complicated is the two-level unitary matrix of the form

$$\begin{bmatrix} 1 & & & \\ & U_{11} & U_{12} & \\ & U_{21} & U_{22} & \\ & & & 1 \end{bmatrix}. \tag{2.71}$$

As it affects both qubits simultaneously, it cannot be represented by a single controlled-U gate. However, it is possible to bring it to the form (2.68) by exchanging the second and forth columns and rows. The specified exchanges correspond to flipping the bit values of the first qubit only when the second qubit has the value 1, that is, the CNOT gate with the second qubit as the control qubit and the first qubit as the target qubit. Through the exchange, the unitary matrix U itself is modified and the rows and columns are exchanged, which corresponds to the basis change by the Pauli X matrix, $U \to U' = XUX$. Putting all together, the two-level unitary matrix is implemented as

$$\begin{bmatrix} 1 & & & \\ & U_{11} & U_{12} & \\ & U_{21} & U_{22} & \\ & & & 1 \end{bmatrix} = \quad .$$



(2.72)

Consider a two-level matrix of the form.

```
In[•]:= matU = TheHadamard[];
        code = TwoLevelU[matU, {2, 3}, 4];
        full = Matrix[code];
        full // MatrixForm
```

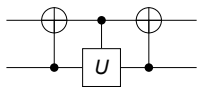
Out[•]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In this case, we need to apply the CNOT gate before and after the controlled-U gate.

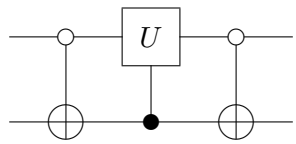
```
In[•]:= ctrlU = GrayTwoLevelU[matU, {2, 3}, S@{1, 2}];
        QuantumCircuit[ctrlU]
```

Out[•]=



In a similar manner, we can implement another form of two-level unitary matrix as

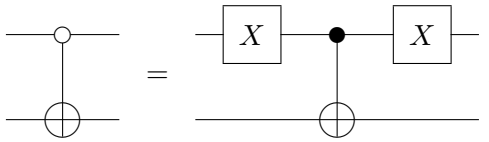
$$\begin{bmatrix} U_{11} & & & U_{12} \\ & 1 & & \\ & & 1 & \\ U_{21} & & & U_{22} \end{bmatrix} = \quad .$$



(2.73)

Here we have adopted a short-hand diagram for the modified-CNOT gate, which flips the bit value of the target qubit when the control qubit is in the state $|0\rangle$

rather than $|1\rangle$. It is achieved by operating the Pauli X gate before and after the usual CNOT gate,



(2.74)

Consider a two-level matrix of the form.

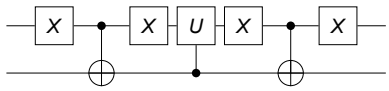
```
In[•]:= matU = TheHadamard[];
        code = TwoLevelU[matU, {1, 4}, 4];
        full = Matrix[code];
        full // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

```
In[•]:= ctrlU = GrayTwoLevelU[matU, {1, 4}, S@{1, 2}];
        QuantumCircuit @@ ctrlU
```

Out[•]=



So far, we have figured out the implementations of 4×4 two-level unitary matrices of different forms just by simple inspection. For more than two qubits, the size of the two-level unitary matrices is much larger and it is difficult to find proper implementations in such a way. Fortunately, there is a systematic way based on the Gray code, which will be discussed later in Section 2.4.

In summary, an arbitrary two-qubit unitary gate can be carried out by first decomposing its matrix representation into two-level unitary matrices and then implementing the two-level unitary matrices by means of the CNOT gate and the controlled-U gate.

Let us consider again the two-qubit model demonstrated before.

```
In[•]:= H = S[1, 1] ** S[2, 1] + S[1, 2] + S[2, 2] + S[1, 3] + S[2, 3]
        U = MultiplyExp[-I H Pi / 4]
        matU = Matrix[U];
```

Out[•]= $S_1^x S_2^x + S_1^y + S_1^z + S_2^y + S_2^z$

Out[•]= $e^{-\frac{1}{4} i \pi \left(S_1^x S_2^x + S_1^y + S_1^z + S_2^y + S_2^z\right)}$

This is the decomposition into the two-level matrices, just showing the first four.

```
In[•]:= twl = TwoLevelDecomposition[matU] // Simplify;
        MatrixForm /@ Matrix /@ twl[[ ;; 3]]
```

Out[•]=

$$\left\{\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix},\ \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&\frac{\frac{1}{2}-\frac{3i}{2}}{\sqrt{7}}&-\frac{\frac{3}{2}+\frac{3i}{2}}{\sqrt{7}}\\0&0&\frac{\frac{3}{2}-\frac{3i}{2}}{\sqrt{7}}&\frac{\frac{1}{2}+\frac{3i}{2}}{\sqrt{7}}\end{pmatrix},\ \begin{pmatrix}1&0&0&0\\0&-\frac{1-3i}{\sqrt{38}}&\sqrt{\frac{14}{19}}&0\\0&-\sqrt{\frac{14}{19}}&-\frac{1+3i}{\sqrt{38}}&0\\0&0&0&1\end{pmatrix}\right\}$$

The two-level matrices are written in terms of the controlled-U and CNOT gate, again showing the first four. The first of the list `twl` happens to be the identity matrix in this case, and it is excluded in the further analysis.

```
In[•]:= gates = GrayTwoLevelU[#, S@{1, 2}] & /@ Rest[twl];
        gates[[ ;; 3]]
```

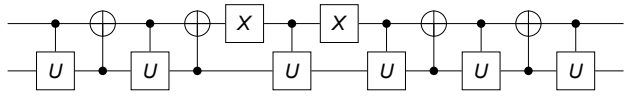
Out[•]=

$$\left\{\left\{\text{ControlledU}\left[\{S_1\},\ \frac{1}{2\sqrt{7}}-\frac{3i\,S_2^x}{2\sqrt{7}}-\frac{3i\,S_2^y}{2\sqrt{7}}-\frac{3i\,S_2^z}{2\sqrt{7}},\ \text{Label}\to U\right]\right\},\ \left\{\text{CNOT}[\{S_2\},\{S_1\}],\right.$$
$$\text{ControlledU}\left[\{S_1\},\ -\frac{1}{\sqrt{38}}-i\sqrt{\frac{14}{19}}\,S_2^y-\frac{3i\,S_2^z}{\sqrt{38}},\ \text{Label}\to U\right],\ \text{CNOT}[\{S_2\},\{S_1\}]\Big\},$$
$$\left.\left\{\text{ControlledU}\left[\{S_1\},\ -\frac{5}{14\sqrt{2}}+\frac{3}{14}i\sqrt{19}\,S_2^y-\frac{5i\,S_2^z}{14\sqrt{2}},\ \text{Label}\to U\right]\right\}\right\}$$

This shows the overall quantum circuit model. Here the same label “U” has been used for different controlled-U gates just for
convenience. Do not forget to reverse the order before plugging the gates in the quantum circuit model.

```
In[•]:= qc = QuantumCircuit @@ Reverse@Flatten[gates]
```

Out[•]=



Check the above quantum circuit by converting it into the matrix representation.

```
In[•]:= new = Matrix[qc] // Simplify;
        new // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix}-\frac{\frac{1}{2}+\frac{5i}{6}}{\sqrt{2}}&-\frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}}&-\frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}}&\frac{\frac{1}{2}-\frac{i}{2}}{\sqrt{2}}\\ \frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}}&\frac{\frac{1}{2}+\frac{i}{6}}{\sqrt{2}}&-\frac{\frac{1}{2}+\frac{5i}{6}}{\sqrt{2}}&-\frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}}\\ \frac{\frac{1}{6}-\frac{i}{2}}{\sqrt{2}}&-\frac{\frac{1}{2}+\frac{5i}{6}}{\sqrt{2}}&\frac{\frac{1}{2}+\frac{i}{6}}{\sqrt{2}}&-\frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}}\\ \frac{\frac{1}{2}-\frac{i}{2}}{\sqrt{2}}&\frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}}&\frac{\frac{1}{2}+\frac{i}{2}}{\sqrt{2}}&-\frac{\frac{1}{2}-\frac{i}{2}}{\sqrt{2}}\end{pmatrix}$$

```
In[•]:= new - matU // Simplify // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$$

2.3 Multi-Qubit Controlled Gates

Let $\mathcal{S}^{\otimes m}$ and $\mathcal{S}^{\otimes n}$ be the Hilbert spaces of the control and target register consisting of m and n qubits, respectively. Suppose that $\hat{U}$ is a unitary operator on the target register. The multi-qubit controlled-U operator is defined by

$$\mathrm{Ctrl}(\hat{U}) : |c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes \hat{U}^{c_1 c_2 \cdots c_m} |t\rangle \ , \tag{2.75}$$

where $c \equiv (c_1 c_2 \ldots c_m)_2$. The unitary transformation $\hat{U}$ acts on the target qubits only when every control qubit is set to $|1\rangle$. We will be most interested in the case with $n = 1$, where the matrix representation takes the form

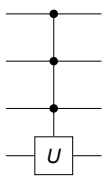
$$\mathrm{Ctrl}(\hat{U}) \doteq \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & U_{11} & U_{12} \\ & & & U_{21} & U_{22} \end{bmatrix} . \tag{2.76}$$

It is the prototype form of a two-level unitary matrix on $(m + 1)$ qubits. Indeed, any two-level unitary matrix can be put into this form by exchanging columns and rows—equivalent to basis changes. Multi-qubit controlled-U gates thus arise naturally as we discuss the universal quantum computation in Section 2.4.

This is a three-qubit controlled-U gate.

```
In[•]:= qc = QuantumCircuit[ControlledU[S@{1, 2, 3}, S[4, 1], "Label" → "U"]]
```

Out[•]=

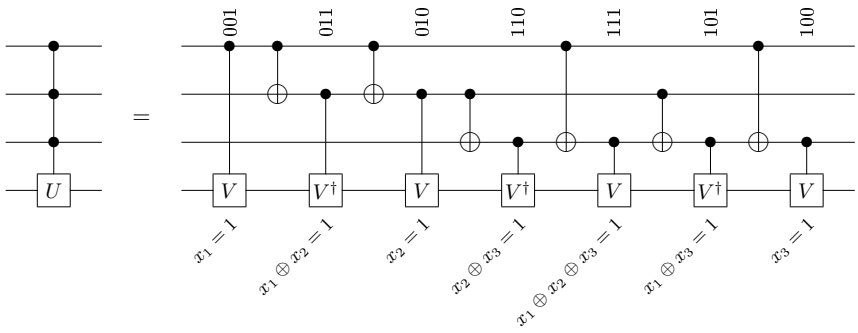


A reliable implementation of multi-qubit controlled-$\hat{U}$ gate is essential in many quantum algorithms. A notable example is the quantum oracle (see Section 4.2.1), which is a key component of quantum decision problems as well as quantum search problems. Here we introduce some widely known methods to implement it efficiently.

2.3.1 Gray Code

We first discuss a systematic method based on the Gray code to decompose a multi-qubit controlled-U gate into factors of either the single-qubit controlled-U

or CNOT gate: For example, consider a 3-qubit controlled-U gate as shown in the following quantum circuit model (Barenco *et al.*, 1995)



(2.77)

where $\hat{V}$ is another unitary operator such that $\hat{V}^4 = \hat{U}$. When every control qubit is set to $|1\rangle$, all $\hat{V}$ gates in the diagram take effects on the target qubit while $\hat{V}^\dagger$ gates are ineffective. To examine the case with the contol qubits in a general state $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$, note that the gates on the target qubit are effective under the conditions specified in terms of the bit values at the bottom of the diagram—see also Eq. (2.54). The conditions are fulfilled systematically by following the Gray code sequence,[1] the strings of which are indicated at the top of the diagram. The identity for the bitwise AND,

$$2^{n-1}(x_1 x_2 \cdots x_n) = \sum_{k_1} x_{k_1} - \sum_{k_1<k_2} (x_{k_1} \oplus x_{k_2}) + \sum_{k_1<k_2<k_3} (x_{k_1} \oplus x_{k_2} \oplus x_{k_3}) - \cdots + (-1)^{n-1}(x_1 \oplus x_2 \oplus \cdots \oplus x_n)\,, \quad (2.78)$$

ensures that the quantum circuit model on the right-hand side of (2.77) reproduces the desired multi-qubit controlled-U gate on the left-hand side.

In general, a n-qubit controlled-U gate can be implemented by combining 2^{n-1} controlled-V gates, $(2^{n-1}-1)$ controlled-$V^\dagger$ gates, and $(2^n - 2)$ CNOT gates, where $\hat{V}$ is a unitary operator satisfying $\hat{V}^{2^{n-1}} = \hat{U}$. For relatively small n ($n \leq 8$), the Gray code is known to be the most efficient method. However, it grows exponentially and eventually becomes impractical. For those cases, several methods have been proposed where the computational cost increases quadratically with the size of the quantum register (Barenco *et al.*, 1995; Nielsen & Chuang, 2011).

Consider a three-qubit register as an example.

```
$L = 3;
jj = Range[$L];
cc = S[jj, None];
```

[1] The *Gray code sequence* is an arrange of bits such that two successive values differ in only one bit.

This is the gate to operate on the target qubit.

```
In[•]:= T = S[$L + 1, None];
        op = QuissoRotation[Pi / 3, T[1]]
```

$$\text{Out[•]=}\quad \frac{\sqrt{3}}{2} - \frac{\mathbb{i}\, S_4^x}{2}$$

This decomposes the multi-qubit controlled-U gate based on the Gray code -- showing only a part of it. Each component is either CNOT or a two-qubit controlled-U gate.

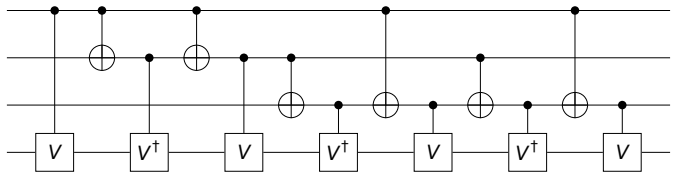
```
In[•]:= gc = GrayControlledU[cc, op]; gc[[ ;; 3]]
```

$$\text{Out[•]=}\quad \left\{\text{ControlledU}\left[\{S_1\}, \frac{2^{1/4}\, e^{-\frac{\mathbb{i}\pi}{24}} + 2^{1/4}\, e^{\frac{\mathbb{i}\pi}{24}}}{2 \times 2^{1/4}} + \frac{\left(2^{1/4}\, e^{-\frac{\mathbb{i}\pi}{24}} - 2^{1/4}\, e^{\frac{\mathbb{i}\pi}{24}}\right) S_4^x}{2 \times 2^{1/4}}, \text{Label} \to V\right],\right.$$
$$\text{CNOT}[\{S_1\}, \{S_2\}],$$
$$\left.\text{ControlledU}\left[\{S_2\}, \frac{2^{1/4}\, e^{-\frac{\mathbb{i}\pi}{24}} + 2^{1/4}\, e^{\frac{\mathbb{i}\pi}{24}}}{2 \times 2^{1/4}} + \frac{\left(-2^{1/4}\, e^{-\frac{\mathbb{i}\pi}{24}} + 2^{1/4}\, e^{\frac{\mathbb{i}\pi}{24}}\right) S_4^x}{2 \times 2^{1/4}}, \text{Label} \to V^\dagger\right]\right\}$$

This is a quantum circuit model of the decomposition.

```
In[•]:= qc = QuantumCircuit[gc]
```

Out[•]=



Finally check the result.

```
In[•]:= expr = ExpressionFor[qc];
        expr2 = QuissoControlledU[cc, op];
        expr - expr2 // Garner
```

```
Out[•]= 0
```

2.3.2 Multi-Qubit Controlled-NOT

The multi-qubit controlled-NOT gate is an important special case of the multi-qubit controlled-U gate, where the unitary operator $\hat{U}$ equals to the Pauli $\hat{X}$. It transforms the logical basis states as [see Eq. (2.75)]

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes \hat{X}^{c_1 c_2 \cdots c_m} |t\rangle \ , \tag{2.79a}$$

or equivalently,

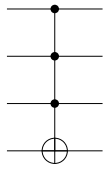
$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes |(c_1 c_2 \cdots c_n) \oplus t\rangle \ . \tag{2.79b}$$

The multi-qubit controlled-NOT gate commonly occurs when one converts the two-level unitary transformation—see Section 2.2.3—on more than two qubits. Indeed, we will generalize the procedure and discuss a systematic way of conversion based on the Gray code in Section 2.4.

This shows a generalized CNOT gate, controlled by three qubits rather than a single qubit.

```
In[•]:= qc = QuantumCircuit[CNOT[S@{1, 2, 3}, S[4]]]
```

Out[•]=



Here is the explicit expression of the multi-qubit CNOT gate in terms of the Pauli operators.

```
In[•]:= op = ExpressionFor[qc]
```

$$\text{Out}[\bullet]= \frac{7}{8} - \frac{1}{8} S_1^z S_2^z - \frac{1}{8} S_1^z S_3^z - \frac{1}{8} S_1^z S_4^x - \frac{1}{8} S_2^z S_3^z - \frac{1}{8} S_2^z S_4^x - \frac{1}{8} S_3^z S_4^x + \frac{1}{8} S_1^z S_2^z S_3^z + \frac{1}{8} S_1^z S_2^z S_4^x + \frac{1}{8} S_1^z S_3^z S_4^x + \frac{1}{8} S_2^z S_3^z S_4^x - \frac{1}{8} S_1^z S_2^z S_3^z S_4^x + \frac{S_1^z}{8} + \frac{S_2^z}{8} + \frac{S_3^z}{8} + \frac{S_4^x}{8}$$

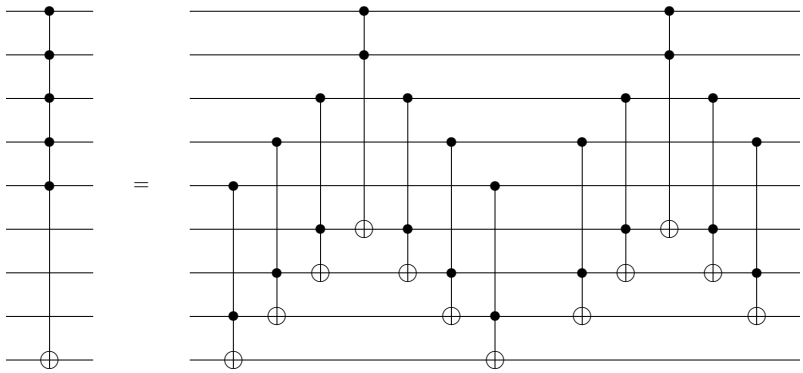
Here we compare it with the expression explicitly constructed in terms of a projection operator.

```
In[•]:= prj = Multiply @@ S[{1, 2, 3}, 11];
        new = (1 - prj) + prj ** S[4, 1]
        op - new // Elaborate
```

$$\text{Out}[\bullet]= 1 - (|1\rangle\langle 1|)_{S_1} (|1\rangle\langle 1|)_{S_2} (|1\rangle\langle 1|)_{S_3} + (|1\rangle\langle 1|)_{S_1} (|1\rangle\langle 1|)_{S_2} (|1\rangle\langle 1|)_{S_3} S_4^x$$

$$\text{Out}[\bullet]= 0$$

An efficient implementation of a multi-qubit CNOT gate uses additional qubits not directly involvined in the gate operation itself as in the followign quantum circuit model (Barenco *et al.*, 1995):



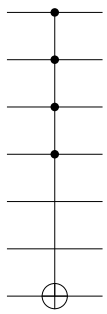
(2.80)

It is emphasized that the quantum states of the extra qubits should not be confused with the so-called "ancillary qubits" in the sense that they do not have to be initialized in a certain fixed quantum state and their state is not altered. The desired gate operation is performed properly regardless of the initial state of the extra qubits and their quantum state is restored after the gate operation.

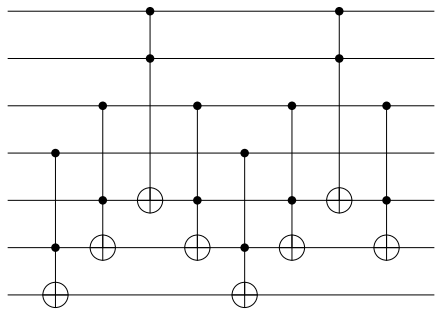
Here we demonstrate a multi-qubit CNOT gate.

```
$n = 4;
$m = 2;
cc = Range[$n];
aa = Range[$n + 1, $n + $m];
tt = $n + $m + 1;
```

```
In[•]:= qc = QuantumCircuit[CNOT[S[cc], S[tt]], "Visible" → S[aa]]
```

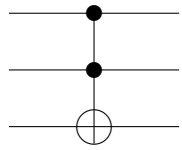


```
In[•]:= tofa = Table[Toffoli[S[$n - j + 1], S[$n + $m - j + 1], S[tt - j + 1]], {j, 1, $m}];
        tofb = Toffoli[S[1], S[2], S[$n + 1]];
        tofc = Rest@tofa;
        new = QuantumCircuit[
          Sequence @@ Flatten@{tofa, tofb, Reverse@tofa, tofc, tofb, Reverse@tofc}]
```



```
In[•]:= Elaborate[qc - new]
Out[•]= 0
```

In the quantum circuit model (2.80), we have introduced another special case of multi-qubit CNOT gates, which is called the *Toffolli gate*, denoted by the quantum circuit element



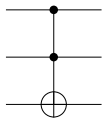
. (2.81)

The Toffoli gate has attracted interested as it is univesal for classical reversible computation. Unfortunately, however, it is not universal for quantum computation.

This is a quantum circuit model of the Toffoli gate.

```
In[•]:= qc = QuantumCircuit[Toffoli[S[1], S[2], S[3]]]
        toff = ExpressionFor[qc]
```

Out[•]=

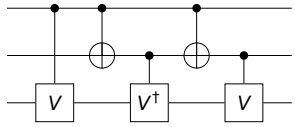


$$\text{Out[•]=}\quad \frac{3}{4}-\frac{1}{4}S_1^z S_2^z-\frac{1}{4}S_1^z S_3^x-\frac{1}{4}S_2^z S_3^x+\frac{1}{4}S_1^z S_2^z S_3^x+\frac{S_1^z}{4}+\frac{S_2^z}{4}+\frac{S_3^x}{4}$$

It can be implemented by a combination of two-qubit gates.

```
In[•]:= gray = GrayControlledU[S@{1, 2}, S[3, 1]];
        qc = QuantumCircuit[gray]
        expr = ExpressionFor[qc]
        toff - expr
```

Out[•]=



$$\text{Out[•]=}\quad \frac{3}{4}-\frac{1}{4}S_1^z S_2^z-\frac{1}{4}S_1^z S_3^x-\frac{1}{4}S_2^z S_3^x+\frac{1}{4}S_1^z S_2^z S_3^x+\frac{S_1^z}{4}+\frac{S_2^z}{4}+\frac{S_3^x}{4}$$

$$\text{Out[•]=}\quad 0$$

Here $V = \sqrt{X}$, and its matrix representation looks like this.

```
In[•]:= matV = MatrixPower[ThePauli[1], 1 / 2];
        matV * 2 / (1 + I) // MatrixForm
        opV = Elaborate@ExpressionFor[matV, S[3]]
```

Out[•]//MatrixForm=

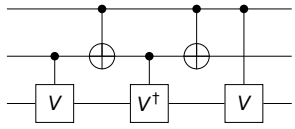
$$\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$$

$$\text{Out[•]=}\quad \left(\frac{1}{2}+\frac{i}{2}\right)+\left(\frac{1}{2}-\frac{i}{2}\right)S_3^x$$

Reversing the above circuit gives the identical result.

```
In[•]:= qc = QuantumCircuit[Reverse@gray]
        expr = ExpressionFor[qc]
        toff - expr
```

Out[•]=



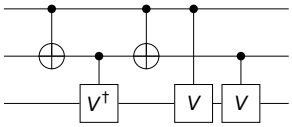
$$\text{Out[•]=}\quad \frac{3}{4}-\frac{1}{4}S_1^z S_2^z-\frac{1}{4}S_1^z S_3^x-\frac{1}{4}S_2^z S_3^x+\frac{1}{4}S_1^z S_2^z S_3^x+\frac{S_1^z}{4}+\frac{S_2^z}{4}+\frac{S_3^x}{4}$$

$$\text{Out[•]=}\quad 0$$

As noted by Smolin & DiVincenzo (1996), it can also be reordered as following. This rearrangement is useful in optimizing the *Fredkin gate*, another universal gate for classical reversible computation.

This shows another slightly different implementation of the Toffoli gate.

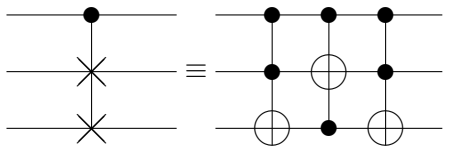
```
In[•]:= qc = QuantumCircuit[Permute[gray, Cycles[{{4, 3, 2, 1}}]]]
        expr = ExpressionFor[qc]
        toff - expr
```



$$\text{Out}[\bullet]= \frac{3}{4} - \frac{1}{4} S_1^z S_2^z - \frac{1}{4} S_1^z S_3^x - \frac{1}{4} S_2^z S_3^x + \frac{1}{4} S_1^z S_2^z S_3^x + \frac{S_1^z}{4} + \frac{S_2^z}{4} + \frac{S_3^x}{4}$$

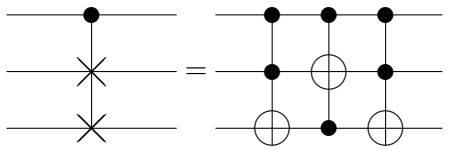
$$\text{Out}[\bullet]= 0$$

The Fredkin gate "swaps" the states of the two target qubits when the control qubit is set in the state $|1\rangle$, as depicted by the following two equivalent quantum circuit elements



$$\equiv \quad . \tag{2.82}$$

In fact, the relation between the SWAP gate and the CNOT gate suggests that there is another equivalent quantum circuit model of the Fredkin gate,

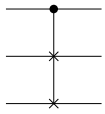


$$= \quad , \tag{2.83}$$

which is simpler than the above. This equivalent quantum circuit model was used by Smolin & DiVincenzo (1996) for an efficient implement of the Fredkin gate in terms of the elementary gates: Like the Toffoli gate, the Fredkin gate is not universal for quantum computation.

This shows the quantum circuit model of the quantum Fredkin gate.

```
In[•]:= qc = QuantumCircuit[Fredkin[S[1], S[2], S[3]]]
```



This is an explicit operator expression of the Fredkin gate in terms of the Pauli operators.

```
In[•]:= op = Elaborate[qc]
```

$$\text{Out[•]}= \frac{3}{4}+\frac{1}{4}S_2^x S_3^x+\frac{1}{4}S_2^y S_3^y+\frac{1}{4}S_2^z S_3^z-\frac{1}{4}S_1^z S_2^x S_3^x-\frac{1}{4}S_1^z S_2^y S_3^y-\frac{1}{4}S_1^z S_2^z S_3^z+\frac{S_1^z}{4}$$

This is the matrix representation of the Fredkin gate in the logical basis.

```
In[•]:= mat = Matrix[qc];
        mat // MatrixForm
```

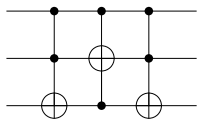
Out[•]//MatrixForm=

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

The Fredkin gate is equivalent to a combination of three Toffoli gates.

```
In[•]:= qc2 = QuantumCircuit[
          Toffoli[S[1], S[2], S[3]],
          Toffoli[S[1], S[3], S[2]],
          Toffoli[S[1], S[2], S[3]]]
```

Out[•]=



```
In[•]:= op2 = Elaborate[qc2]
```

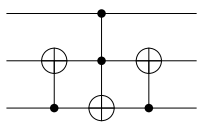
$$\text{Out[•]}= \frac{3}{4}+\frac{1}{4}S_2^x S_3^x+\frac{1}{4}S_2^y S_3^y+\frac{1}{4}S_2^z S_3^z-\frac{1}{4}S_1^z S_2^x S_3^x-\frac{1}{4}S_1^z S_2^y S_3^y-\frac{1}{4}S_1^z S_2^z S_3^z+\frac{S_1^z}{4}$$

```
In[•]:= op - op2
Out[•]= 0
```

In fact, the relation between the SWAP gate and the CNOT gate suggests that there is another equivalent quantum circuit model of the Fredkin gate.

```
In[•]:= new = QuantumCircuit[
          CNOT[S[3], S[2]],
          Toffoli[S[1], S[2], S[3]],
          CNOT[S[3], S[2]]
         ]
```

Out[•]=



```
In[•]:= qc - new // Elaborate
Out[•]= 0
```

2.4 Universal Quantum Computation

In classical computation, it is known that a finite set of logic gates—typically including AND, OR, and NOT—is sufficient to calculate any binary function. The set is said to be *universal* for classical computation. One can ask whether there exists a universal set of elementary quantum logic gates for quantum computation that enables to implement any arbitrary quantum unitary operation? In this section, we examine this question.

Consider a system of n qubits. Suppose that we want to implement an arbitrary unitary operation $\hat{U}$ on the n-qubit register. We start by decomposing $\hat{U}$ into a set of two-level unitary transformations, which we discussed for the case of two-qubit systems in Section 2.2.3.

Consider a three-qubit system, and an arbitrary unitary operation on it.

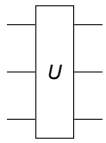
```
In[•]:= $n = 3;
        SS = S[Range[$n], None];
        mat = RandomUnitary[2^$n];
        mat[[ ;; 3, ;; 3]] // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -0.33896+0.246376\,i & 0.130988+0.0604394\,i & 0.0749685+0.112423\,i \\ -0.357331-0.0573556\,i & 0.10117+0.293136\,i & -0.279531+0.0506324\,i \\ 0.0368121-0.400436\,i & 0.0174221+0.0806826\,i & 0.149-0.435898\,i \end{pmatrix}$$

```
In[•]:= op = ExpressionFor[mat, SS];
        qc = QuantumCircuit[{op, "Label" → "U"}]
```

Out[•]=



```
In[•]:= twl = TwoLevelDecomposition[mat];
        MatrixForm /@ Matrix /@ twl[[ ;; 3]] // TableForm
```

Out[•]//TableForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.+0.\,i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.847915-0.530133\,i \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.494159+0.441522\,i & 0.601185-0.446589\,i \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.601185-0.446589\,i & -0.494159-0.441522\,i \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.830166-0.00942158\,i & 0.557437 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.557437 & 0.830166+0.00942158\,i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Now a remaining question is how to implement the two-level unitary transformation in terms of elementary gates.

Let us consider a particular two-level unitary matrix on a three-qubit system. We want to implement it in terms of elementary quantum logic gates.

```
In[•]:= x = 4; y = 5;
        U = TheRotation[Pi / 3, 2];
        mat = Matrix@TwoLevelU[U, {x, y}, 2^$n];
        mat // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

```
In[•]:= op = ExpressionFor[mat, SS]
```

Out[•]=

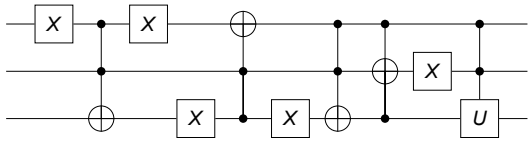
$$\frac{1}{8}\left(6+\sqrt{3}\right)+\frac{1}{8}\left(2-\sqrt{3}\right)\,S_1^z\,S_2^z+\frac{1}{8}\left(2-\sqrt{3}\right)\,S_1^z\,S_3^z+\frac{1}{8}\left(-2+\sqrt{3}\right)\,S_2^z\,S_3^z-\frac{1}{2}\,S_1^+\,S_2^-\,S_3^-+\frac{1}{2}\,S_1^-\,S_2^+\,S_3^+$$

Our implementation is based on the Gray code sequence. Notice the function **Reverse**.

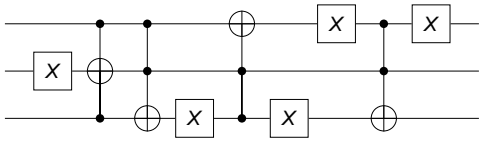
```
In[•]:= gates = Flatten@GrayTwoLevelU[U, {x, y}, SS]
Out[•]= {S_1^x, CNOT[{S_1, S_2}, {S_3}], S_1^x, S_3^x, CNOT[{S_2, S_3}, {S_1}],
  S_3^x, CNOT[{S_1, S_2}, {S_3}], CNOT[{S_1, S_3}, {S_2}], S_2^x,
  ControlledU[{S_1, S_2}, Sqrt[3]/2 + i S_3^y/2, Label → U], S_2^x, CNOT[{S_1, S_3}, {S_2}],
  CNOT[{S_1, S_2}, {S_3}], S_3^x, CNOT[{S_2, S_3}, {S_1}], S_3^x, S_1^x, CNOT[{S_1, S_2}, {S_3}], S_1^x}

In[•]:= new = Apply[Dot, Matrix[#, SS] & /@ Elaborate /@ gates] // Simplify;
        new // MatrixForm
Out[•]//MatrixForm=
⎛1  0  0  0    0    0  0  0⎞
⎜0  1  0  0    0    0  0  0⎟
⎜0  0  1  0    0    0  0  0⎟
⎜0  0  0  √3/2 -1/2 0  0  0⎟
⎜0  0  0  1/2  √3/2 0  0  0⎟
⎜0  0  0  0    0    1  0  0⎟
⎜0  0  0  0    0    0  1  0⎟
⎝0  0  0  0    0    0  0  1⎠

In[•]:= qc1 = QuantumCircuit[gates[[ ;; 10]]]
        qc2 = QuantumCircuit[gates[[11 ;;]]]
Out[•]=
```




```
Out[•]=
```




```
In[•]:= new = Matrix[qc2].Matrix[qc1] // Simplify;
        new // MatrixForm
Out[•]//MatrixForm=
⎛1  0  0  0    0    0  0  0⎞
⎜0  1  0  0    0    0  0  0⎟
⎜0  0  1  0    0    0  0  0⎟
⎜0  0  0  √3/2 -1/2 0  0  0⎟
⎜0  0  0  1/2  √3/2 0  0  0⎟
⎜0  0  0  0    0    1  0  0⎟
⎜0  0  0  0    0    0  1  0⎟
⎝0  0  0  0    0    0  0  1⎠
```

2.5 Measurements

We conclude this chapter with a few discussions on measurement. In quantum computers, measurement is assumed to be performed independently on individual qubits in the logical basis, $\{|x\rangle : x = 0, 1, \ldots, 2^n - 1\}$.

What if a measurement in another basis, say, $\{|\alpha_x\rangle = \hat{U}\,|x\rangle\}$ is required? We require that the input state $|\alpha_x\rangle$ should end up with the logical basis state $|x\rangle$ with

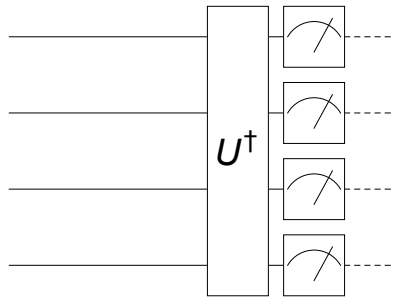


Figure 2.6: Measurement in a basis $\{|\alpha_x\rangle\}$ other than the logical basis. The unitary operator $\hat{U}$ here corresponds to the basis change $|\alpha_y\rangle = \hat{U}\,|y\rangle = \sum_x |x\rangle\,\langle x|\alpha_y\rangle$.

unit probability so that the measurement yields the outcome x. This process is described by the measurment operator $\hat{P}_x := |x\rangle\,\langle\alpha_x| = |x\rangle\,\langle x|\,\hat{U}^\dagger$. Evidently they satisfy the condition $\sum_x \hat{P}_x^\dagger \hat{P}_x = \hat{I}$ for measurement operators (see Postulate 1.3). Now we note that $\hat{E}_x := |x\rangle\,\langle x|$ is nothing but the measurement operators in the logical basis state. This implies that simply by applying the inverse unitary operation $\hat{U}^\dagger$ before the measurement, the measurement in the new basis $\{|\alpha_x\rangle\}$ can be achieved through the measurement in the logical basis; see Fig. 2.6. For example, suppose that a qubit is in the state $|\psi\rangle = |0\rangle\,c_0 + |1\rangle\,c_1$. By default, a measurement is assumed to be in the logical basis and the measurement statistics reflects the probability distribution $P_0 = |c_0|^2$ and $P_1 = |c_1|^2$ as illustrated in the demonstration below.

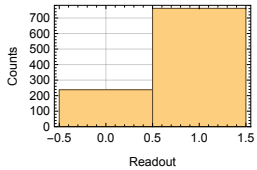
First consider a measurement in the logical basis.

```
In[•]:= Let[Complex, c]
        vec = ProductState[S[1] → {c[0], c[1]}];
        qc = QuantumCircuit[vec, "Spacer", Measurement[S[1]], "PortSize" → {2, 0.2}]
```

Out[•]= $|0\rangle\,c_0 + |1\rangle\,c_1$ —— 

```
In[•]:= Block[
         {c, data},
         c[0] = 1 / 2;
         c[1] = Sqrt[3] / 2;
         data = Table[out = ExpressionFor[qc]; Readout[out, S[1]], 1000];
         Histogram[data, FrameLabel → {"Readout", "Counts"}, ImageSize → Small]
        ]
```

Out[•]=



We now want to make a measurement on the eigenbasis $\{|\pm\rangle\}$ of the Pauli X operator. In this case, it is the Hadamard gate that gives the desired basis change.

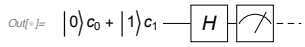
In the new basis, the state vector $|\psi\rangle$ is given by

$$|\psi\rangle = |+\rangle \frac{c_0 + c_1}{\sqrt{2}} + |-\rangle \frac{c_0 - c_1}{\sqrt{2}}, \tag{2.84}$$

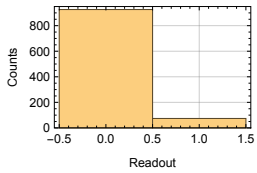
Now the measurement statistics is in accordance with the probability distribution $P_\pm = |c_0 \pm c_1|^2/2$.

Now consider a measurement in the eigen-basis of the Pauli X operator.

```
In[•]:= Let[Complex, c]
        vec = ProductState[S[1] → {c[0], c[1]}];
        qc = QuantumCircuit[vec, S[1, 6], Measurement[S[1]], "PortSize" → {2, 0.2}]
```

Out[•]= 


```
In[•]:= Block[
         {c, data},
         c[0] = 1 / 2;
         c[1] = Sqrt[3] / 2;
         data = Table[out = ExpressionFor[qc]; Readout[out, S[1]], 1000];
         Histogram[data, FrameLabel → {"Readout", "Counts"}, ImageSize → Small]
        ]
```

Out[•]= 

Another interesting example is the *Bell measurement*, that is, the measurement on two qubits in the basis of Bell states,

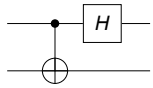
$$\begin{aligned}
|\beta_0\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\
|\beta_1\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\
|\beta_2\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}} \\
|\beta_3\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}
\end{aligned} \tag{2.85}$$

Recall that the Bell states can be generated from the logical basis states by the so-called entangler, a combination of the Hadamard gate and the CNOT gate; see Section 2.2.1. Therefore, the Bell measurement can be achieved by applying the inverse of the entangler operation.

This is the "disentangler" quantum circuit, which is the inverse of the entangler quantum circuit .

```
In[•]:= disentangler = QuantumCircuit[CNOT[S[1], S[2]], S[1, 6]]
```

Out[•]=



This shows that the disentangler quantum circuit maps the Bell states into the logical basis states.

```
In[•]:= op = ExpressionFor[disentangler];
        bs = BellState@S@{1, 2};
        Thread[bs → op ** bs] // LogicalForm // TableForm
```

Out[•]//TableForm=

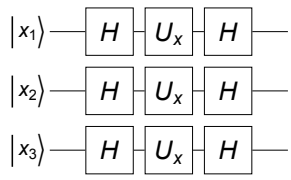
$$\frac{|0_{S_1}0_{S_2}\rangle + |1_{S_1}1_{S_2}\rangle}{\sqrt{2}} \to |0_{S_1}0_{S_2}\rangle$$
$$\frac{|0_{S_1}1_{S_2}\rangle + |1_{S_1}0_{S_2}\rangle}{\sqrt{2}} \to |0_{S_1}1_{S_2}\rangle$$
$$\frac{|0_{S_1}1_{S_2}\rangle - |1_{S_1}0_{S_2}\rangle}{\sqrt{2}} \to |1_{S_1}1_{S_2}\rangle$$
$$\frac{|0_{S_1}0_{S_2}\rangle - |1_{S_1}1_{S_2}\rangle}{\sqrt{2}} \to |1_{S_1}0_{S_2}\rangle$$

For fast quantum algorithms and quantum error corrections, more sophisticated meausurements may be necessary. A common example is the quantum phase estimation, which is one of the core parts of Shor's factorization algorithm. We will discuss it in Section 4.4.

Problems

1. Let $\hat{\Phi}(\phi)$ be the *phase gate*, which gives rise to a *relative* phase shift by $\phi \in [0, 2\pi)$,
$$\hat{\Phi}(\phi) : |0\rangle \mapsto |0\rangle \ , \quad |1\rangle \mapsto |1\rangle \, e^{i\phi} \, . \tag{2.86}$$
   (a) Show that on a n-qubit quantum register,
$$[\hat{\Phi}(\phi)]^{\otimes n} \, |x\rangle = |x\rangle \, e^{i\phi(x_1 + \cdots + x_n)} \tag{2.87}$$
   for any $x = 0, 1, \ldots, 2^n - 1$.
   (b) Evaluate explicitly the state $[\hat{\Phi}(\phi)]^{\otimes n} \hat{H}^{\otimes n} \, |0\rangle$, where $\hat{H}$ is the Hadamard gate.

2. Let $\hat{U}_x(\phi)$ be the rotatin around the x-axis by angle ϕ on a single qubit. Analyze and evaluate explicitly the following quantum circuit model



, (2.88)

where $|x\rangle := |x_1\rangle \otimes |x_2\rangle \otimes |x_3\rangle$ is a 3-qubit logical basis state and the labels "H" and "U_x" indicate the Hadamard gate $\hat{H}$ and the rotation gate $\hat{U}_x(\phi)$, respectively. Generalize the result and show that

$$\hat{H}^{\otimes n}[\hat{U}_x(\phi)]^{\otimes n}\hat{H}^{\otimes n}\,|x\rangle = e^{-i\phi n/2}\,|x\rangle\, e^{i\phi(x_1+\cdots x_n)} \tag{2.89}$$

for any $x = 0, 1, \ldots, 2^n - 1$.

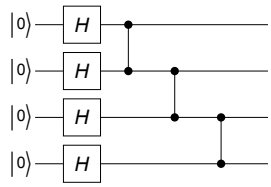
3. Let $\hat{S}^\mu$ be the Pauli operarors. Show that

$$e^{i\hat{S}^\mu\phi/2}\hat{S}^\nu e^{-i\hat{S}^\mu\phi/2} = \hat{S}^\nu\cos(\phi) + \hat{S}^\lambda\epsilon_{\lambda\mu\nu}\sin(\phi) \tag{2.90}$$

for all $\mu, \nu = x, y, z$ and $\mu \neq \nu$.

4. Consider a quantum register of four qubits.

   (a) Analyze the following quantum circuit model



(2.91)

   and evaluate the resulting state $|\Psi\rangle$ explicitly. The state is a so-called *cluster state* or *graph state*, a crucial resource in the measurement-based quantum computation—see Section 3.4.

   (b) Show that in the state $|\Psi\rangle$ from (b), every qubit is *maximally entangled* with the rest qubits, that is, the *reduced density matrix* $\hat{\rho}_j$ of the jth qubit, $\hat{\rho}_j := \mathrm{Tr}_{k\neq j}\,|\Psi\rangle\langle\Psi|$, is given by $\hat{\rho}_j = \hat{I}/2$—it exhibits coherence in no basis.

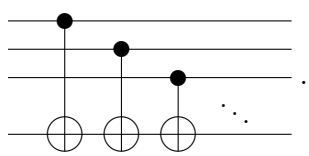
5. Suppose that a two-qubit system is known to be in one of the four eigenstates of the unitary operator

$$\hat{U} = e^{i\phi}\left(|0\rangle\langle 0| + i\,|1\rangle\langle 1| - |2\rangle\langle 2| - i\,|3\rangle\langle 3|\right). \tag{2.92}$$

   Construct a quantum circuit model to figure out the unknown state using two additional qubits.

   Hint: Use the property (2.18) of the Hadamard gate and those of the controlled-U gates, where a unitary operator acts on a two-qubit system controlled by a single qubit.

6. Consider the following quantum circuit model consisting of the CNOT gates on a n-qubit quantum register



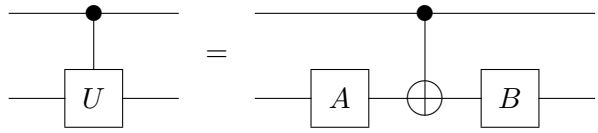
(2.93)

(a) Find the output state for the input of a logical basis state $|x_1\rangle \otimes \cdots \otimes |x_n\rangle \in \mathcal{S}^{\otimes n}$.

(b) Find the output state for the input state $|+\rangle \otimes \cdots \otimes |+\rangle \otimes |-\rangle$, where $|\pm\rangle := (|0\rangle \pm |1\rangle)/\sqrt{2}$.

7. Let $\hat{U}$ be a unitary operator on a single qubit. Show that the following three statements are equivalent:

(a) There exist unitary gates $\hat{A}$ and $\hat{B}$ such that



(2.94)

(b) There exists a unitary operator $\hat{W}$ such that $\hat{U} = \hat{W}\hat{Z}\hat{W}^\dagger$.

(c) $\operatorname{Tr}\hat{U} = 0$ and $\det\hat{U} = -1$.

8. Let $P(i \leftrightarrow j)$ be the matrix exchanging the ith and jth rows (columns) of vectors/matrices. For example, on a four-dimenisonal space,

$$P(1 \leftrightarrow 2) = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}. \tag{2.95}$$

$\hat{P}(i \leftrightarrow j)$ is the corresponding operator.

(a) Find the quantum circuit model for $\hat{P}(2 \leftrightarrow 3)$ on a two-qubit system (four-dimensional vector space $\mathcal{S} \otimes \mathcal{S}$) using CNOT gates only.

(b) Find the quantum circuit model for $\hat{P}(2 \leftrightarrow 3)$ on a three-qubit system ($\mathcal{S}^{\otimes 3}$) using only mult-qubit CNOT gates only.

(c) Find the quantum circuit model for $\hat{P}(4 \leftrightarrow 7)$ on a three-qubit system ($\mathcal{S}^{\otimes 3}$) using only multi-qubit CNOT gates onlys.

Chapter 3

Virtual Realizations of Quantum Computers

- April 29, 2021 (v1.9)

In the previous chapter, we have seen how quantum computation works under the assumption that the elementary quantum logic gate operations are available. But how can one build a quantum computer, a machine, that allows such quantum logic gates? Quantum computers are physical systems and the implementation of all the quantum logic gates are governed by the laws of physics. In this section, we discuss the basic physical principles that are directly involved in the implementation of quantum logic gates. Through the course of discussion, it will become clearer what basic conditions and requirements should be fulfilled to build a quantum computer.

By now there are many quantum computer architectures that are not only proposed and tested at the research level but also actually running. However, to understand each architecture requires a certain level of knowledge about the physical systems. For example, to underdestand a quantum computer based on superconducting circuits, one has to first understand the superconductivity, the Josephson effect, the flux quantization, the Josephson inductance (a sort of non-linear kinetic inductance), and the interaction of superconducting circuits to electromagnetic fields. Such discussions often hinder access to the essential part of the operating principle of the quantum computer, and are out of the scope of this workbook.

Here we consider an idealistic and minimal quantum system that is suitable for quantum computation and discuss how to control it to implement the desired quantum logic gates. It is certainly impractical. Nevertheless it will point out the crucial requirements when one wants to actually develop an quantum computer based on realistic systems and devices. Through the discussions, we will indicate how the relevant parts are related to actual quantum computer architectures.

Figure 3.1: An internal view of IBQ quantum computer based on superconducting circuits.

3.1 Quantum Bits

We have already noted several times that the building blocks and basic computational units of a quantum computer are qubits. Ideally, a qubit is associated with a two-dimensional Hilbert space. In reality, the Hilbert space for any realisitc system is infinite-dimensional, and a qubit usually refers to *certain degrees of freedom* that are relatively independent of other degrees of freedom. For example, the spin of electron or the polarization of photon has exactly two-dimensional Hilbert space. In many cases, a qubit may also refer to a *certain two-dimensional subspace* of a larger Hilbert space that are decoupled or separated relatively well from the rest. For example, a superconducting charge qubit refers to a two lowest-energy charnging states in a small—hundreds nanometers in lateral size—superconducting island.

However, a well-defined two-dimensional Hilbert space (or subspace) does not necessarily mean that the degrees of freedom in question qualifies as a qubit. For example, consider the spin of neutron. Although its Hilbert space dimension is certainly two, you recognize that it can hardly be used for quantum computatoin. It is hard to isolate a neutron, and even more so to manipulate its spin in a reliable and tunable manner. Then what requirements should qubits—individually and as a whole collection—meet to build a practical quantum computer? Apart from specific technical issues in specific systems, these are the basic requirements–the so-called DiVincenzo criteria—commonly rated to assess the potential of a particular architecture in consideration (DiVincenzo, 2000):

(a) The qubits should be well characterized and form a scalable system. For each qubit, the Hilbert space should be well defined in the sense mentioned above and its internal Hamiltonian including the parameters needs to be accurately known. The qubits must also admit genuine interactions among

them and maintain the characteristics up to a sufficiently large scale for practical computation.

(b) The qubits should allow initialization to a fixed logical basis state. Even though any quantum algorithm assumes superposition in the middle of the process, all computations must start from a known value. This straightforward requirement is the same even for classical computers. One of the common approaches for initialization is to cool down the system and wait for it to relax to the ground state. Another method is to perform a projection measurement in the logical basis so as for the state to collapse to the logical basis state corresponding to the measurement outcome.

(c) The qubits should maintain coherence long enough for the desired gate operations. The superposition between different logical basis states is a crucial difference dinstinguishing quantum computers from classical computers. Unfortunately, qubits are subject to various decoherence effects due to external control circuits and measuring devices and eventually lose quantumness. The system should maintain the coherence during the desired gate operations to get a reliable result out of the computation.

(d) The system of qubits should allow a *universal* set of quantum gate operations. As discussed in Chapter 2, quantum computation is to achieve a desired unitary transformation with a combination of certain elementary gate operations that are acting on a single qubit or two qubits at a time. Below we will discuss the physical implementations of those elementary quantum logic gates.

(e) The system should allow measurements in the logical basis. At the end of a computation, the result needs to be read out, and it is achieved by performing measurements on specific qubits. The capability of accurate measurement is called the *quantum efficiency*. Ideal measurement has 100% quantum efficiency. Less than 100 % quantum efficiency in measurements leads to a tradeoff with other resources. For example, if a computation is desired with 97 % reliability while measurements have only 90 % quantum efficiency, then one must repeat the computation three times or more.

In the rest of the chapter, let us now focus on the manipulation of quantum states of qubits, which naturally forms the largest part of quantum computation.

Consiser a quantum computer consisting of n qubits. Let $\mathcal{S}_j$ $(j = 1, \ldots, n)$ be the 2-dimentional Hilbert space associated with the jth single qubit. An ideal quantum computer would realize a Hamiltonian on $\mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_n$ of the form

$$\hat{H}(t) = \sum_j \sum_\mu B_j^\mu(t) \hat{S}_j^\mu + \sum_{ij} \sum_{\mu\nu} J_{ij}^{\mu\nu}(t) \hat{S}_i^\mu \hat{S}_j^\nu \,, \tag{3.1}$$

where $\hat{S}_j^\mu$ $(\mu = x, y, z)$ are the Pauli operators—see Section 2.1.1—on $\mathcal{S}_j$.

The parameter B_j^μ directly controls the jth qubit; physically, it plays the same role as the magnetic field on a spin. In realistic systems, it may be hard to address single qubits individually. How freely single qubits can be manipulated strongly depends on how many of the parameters B_j^μ the system allows to be tunable accurately. See, for example, Section 3.2 below.

The control parameters $J_{ij}^{\mu\nu}$ describe the (hypothetical) exchange coupling between the ith and jth qubits. In principle, any type of interaction between two qubits can be used to implement CNOT gate (see Section 3.2 for examples) although the actual implementation may require more than one interactions and many additional single-qubit operations depending on the particular type of coupling. Therefore, an accurate control of the coupling parameters $J_{ij}^{\mu\nu}$ between a specific pair of qubits is essential for universal quantum computation. In realistic systems, the coupling parameters $J_{ij}^{\mu\nu}$ are even more difficult to realize. First of all, in many architectures the connectivity of qubits that are not in direct priximity of each other is seriously limited. Further, dynamically turning on and off the coupling is often forbidden. In many cases, in order to achieve a sizable strength, the exchange couplings are kept turned on throughout the whole computation processes. Such difficulties and imperfections all contribute to the errors in the computational outputs.

We will be denoting each qubit by the symbol S and accompanying indices .

```
Let[Qubit, S]
```

The Pauli operators are specified by the last index. For example, the Pauli operator S_j^x is denoted by S[j,1].

```
In[•]:= S[j, 1]
Out[•]= S_j^x

In[•]:= S[j, 1] ** S[j, 2]
        S[j, 1] ** S[k, 2]
Out[•]= i S_j^z

Out[•]= S_j^x S_k^y
```

3.2 Dynamical Scheme

The dynamic scheme implements the desired quantum gate operations by means of the time-evolution operator governing the dynamics of the physical qubits in a quantum computer; hence the name. It is the most common scheme of quantum computation and a majority of quantum computers demonstrated so far are based on it.

3.2.1 Implementation of Single-Qubit Gates

Conceptually, the most straightforward way to contol a single qubit is to apply the static parameters B^x, B^y, and B^z for a certain period τ of operation time. We refer to them collectively by a vector $\boldsymbol{B} = (B^x, B^y, B^z)$, which can be regarded as a fictitious magnetic field—here, the dimension of $\boldsymbol{B}$ is energy unlike the real magnetic field. The Hamiltonian for the qubit is given by

$$\hat{H} = \frac{1}{2}\boldsymbol{B} \cdot \hat{\boldsymbol{S}}, \tag{3.2}$$

where $\hat{\boldsymbol{S}} := (\hat{S}^x, \hat{S}^y, \hat{S}^z)$ is the vector of the Pauli operators on $\mathcal{S}$. The time evolution is then governed by the unitary operator

$$\hat{U}(t) = \exp(-it\hat{H}) = \exp(-it\boldsymbol{B} \cdot \hat{\boldsymbol{S}}/2). \tag{3.3}$$

It has the same form as the single-qubit rotation operator in Eq. (2.23), Section 2.1.3: It describes the rotation in the Pauli spce—or the Bloch spahere—around the axis parallel to the vector $\boldsymbol{B}$ by the angle $\phi(t) = |\boldsymbol{B}|t$. When the involved two-level system is a true spin, such a rotation in the Pauli space is called the *Larmor precession*.

Consider a qubit. Let us apply a (fictitious) magnetic field.

```
In[•]:= Let[Real, B]
        opH = Dot[B@{1, 2, 3}, S@{1, 2, 3}] / 2
```

$$\text{Out[•]=} \quad \frac{1}{2}\left(B_1 S^x + B_2 S^y + B_3 S^z\right)$$

To be specific, we consider the case with the magnetic field between the *x*- and *z*-axis in the *xz*-plane. The factor of $1/\sqrt{2}$ is to set the energy (frequency) scale to unit.

```
In[•]:= B[1] = B[3] = 1 / Sqrt[2]; B[2] = 0;
        opH
```

$$\text{Out[•]=} \quad \frac{1}{2}\left(\frac{S^x}{\sqrt{2}} + \frac{S^z}{\sqrt{2}}\right)$$

```
In[•]:= opU[t_] = MultiplyExp[-I t opH]
        opU[t] // Elaborate
```

$$\text{Out[•]=} \quad e^{-\frac{1}{2} i t \left(\frac{S^x}{\sqrt{2}} + \frac{S^z}{\sqrt{2}}\right)}$$

$$\text{Out[•]=} \quad \text{Cos}\left[\frac{t}{2}\right] - \frac{i\, S^x\, \text{Sin}\left[\frac{t}{2}\right]}{\sqrt{2}} - \frac{i\, S^z\, \text{Sin}\left[\frac{t}{2}\right]}{\sqrt{2}}$$

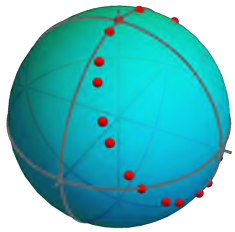
```
In[•]:= vec[t_] = opU[t] ** Ket[] // Elaborate
```

$$\text{Out[•]=} \quad -\frac{i\,\left|1_S\right\rangle\, \text{Sin}\left[\frac{t}{2}\right]}{\sqrt{2}} + \left|\_\right\rangle\left(\text{Cos}\left[\frac{t}{2}\right] - \frac{i\, \text{Sin}\left[\frac{t}{2}\right]}{\sqrt{2}}\right)$$

This illustrates the Larmor precession (with the qubit regarded as a "spin").
Technical Note: You need `Chop` because numerical errors sometimes give `0.+Ket[...]`, which cannot be handled properly by `BlochVector`.

```
In[•]:= vv = BlochVector /@ Table[Chop@vec[t], {t, 0, 8, 0.5}];
        BlochSphere[{Red, Bead /@ vv}, ImageSize → Small]
```

Out[•]=



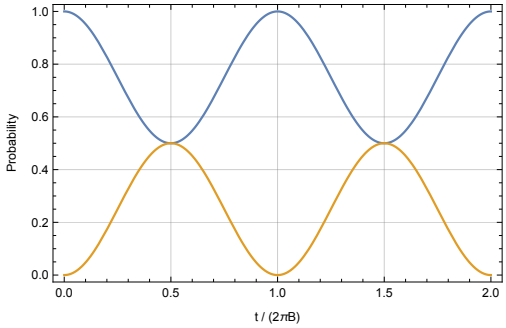
Let us now examine the probabilities for the qubit to be found in the logical basis states.

```
In[•]:= prob[t_] = Abs[Normal@Matrix@vec[t]]^2
```

$$\text{Out[•]=}\quad \left\{\left|\text{Cos}\left[\frac{t}{2}\right] - \frac{i\,\text{Sin}\left[\frac{t}{2}\right]}{\sqrt{2}}\right|^2,\ \frac{1}{2}\left|\text{Sin}\left[\frac{t}{2}\right]\right|^2\right\}$$

```
In[•]:= Plot[Evaluate@prob[2 Pi t], {t, 0, 2},
          FrameLabel → {"t / (2πB)", "Probability"}]
```

Out[•]=



Although conceptionally simple, the above method has limited applications in many realistic systems. For example, in the presence of other levels, one cannot apply the method selectively between the two levels in question. More widely applicable method is to apply an oscillating field: Suppose that

$$B^x = B_\perp \cos(\omega t)\,, \quad B^y = B_\perp \sin(\omega t)\,, \quad B^z = B_0\,. \tag{3.4}$$

One can regaard it as a fictitious magnetic field precessing around the z-axis with frequency ω. The Hamiltonian now depends on time and is given by

$$\hat{H}(t) = \frac{1}{2}B_\perp\left[\cos(\omega t)\hat{S}^x + \sin(\omega t)\hat{S}^y\right] + \frac{1}{2}B_0\hat{S}^z\,. \tag{3.5}$$

Recalling the property in Eq. (2.25) of the Pauli operators as the generators of rotation, we observe that

$$\hat{U}_z^\dagger(\omega t)\hat{H}(t)\hat{U}_z(\omega t) = \frac{1}{2}B_\perp\hat{S}^x + \frac{1}{2}B_0\hat{S}^z \tag{3.6}$$

does not depend on time any longer. This observation suggests that the dynamics looks simpler in the rotating frame. As the rotating frame is not an inertial frame,

one has to take into account the non-inertial effect—corresponding to the classical *inertial force*: The state vectors $|\psi(t)\rangle$ and $|\psi_R(t)\rangle$ in the lab and rotating frame, respectively, are related by

$$|\psi_R(t)\rangle = \hat{U}_z^\dagger(\omega t)\,|\psi(t)\rangle\ . \tag{3.7}$$

Putting it into the original Schrödinger equation for $|\psi(t)\rangle$,

$$i\partial_t\,|\psi\rangle = \hat{H}(t)\,|\psi\rangle\ , \tag{3.8}$$

and operating $\hat{U}_z^\dagger(t)$ from the left on both sides, one can get the Schrödinger equation in the rotating frame,

$$i\partial_t\,|\psi_R\rangle = \hat{H}_R\,|\psi_R\rangle\ , \tag{3.9}$$

where the Hamiltonian in the rotating frame is given by

$$\hat{H}_R(t) := U_z^\dagger(\omega t)\hat{H}(t)\hat{U}_z(\omega t) - \hat{U}_z^\dagger(\omega t)[i\partial_t\hat{U}_z(\omega t)]\,. \tag{3.10}$$

The second term in $\hat{H}_R$ is responsible for the non-inertial effect. As already expected in the above observation, the rotating-frame Hamiltonian does not depend on time any longer,

$$\hat{H}_R = \frac{1}{2}B_\perp\hat{S}^x + \frac{1}{2}(B_0-\omega)\hat{S}^z\,. \tag{3.11}$$

The time-evolution operator in the rotating frame,

$$\hat{U}_R(t) = \exp(-it\hat{H}_R) \tag{3.12}$$

is *formally* the same as the one (3.3) in the lab frame. It describes a precession around the axis parallel to $\boldsymbol{B}_R := (B_\perp, 0, B_0-\omega)$ with the frequency

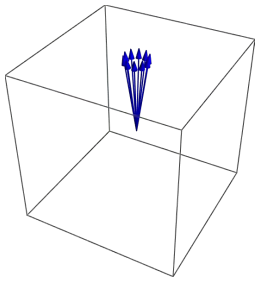
$$\Omega_R := \sqrt{B_\perp^2 + (B_0-\omega)^2}\,. \tag{3.13}$$

The precession in the rotating frame is called the *Rabi oscillation* and the frequency in (3.13) is called the *Rabi frequency*. The fictitious magnetic field in the rotating frame—compare Eqs. (3.2) and (3.10)—is almost along the x-axis for $\omega \approx B_0$, that is, the time-evolution $\hat{U}_R(t)$ in the rotating frame corresponds to the rotation around the x-axis in the Bloch sphere. In this case, the qubit can make a full transition from the initial state $|0\rangle$ to the orthogonal state $|1\rangle$ by properly tuning the operation time and/or the parameter $B_\perp$. In this sense, when $\omega \approx B_0$, the system is said to be at *resonance*. As the driving frequency ω gets away off the resonance, the maximum transition probability becomes smaller and smaller. This resonance behavior allows to induce transitions between a selected pair of two levels among many others.

Let us now apply a time-dependent field. It precesses around the z-axis. Note that typically, $B_3 \gg B_1, B_2$.

```
In[•]:= ω = 2 Pi;
        B[1] = Cos[ω t];
        B[2] = Sin[ω t];
        B[3] = 1.1 ω; (* near resonance *)
        Graphics3D[{Blue, Table[Arrow@Tube@{{0, 0, 0}, B@{1, 2, 3}}, {t, 0, 1, 0.1}]},
         PlotRange → ω {{-1, 1}, {-1, 1}, {-1, 1}}, ImageSize → Small]
```

Out[•]=



```
In[•]:= opH = Dot[B@{1, 2, 3}, S@{1, 2, 3}] / 2
```

$$\text{Out[•]=}\quad \frac{1}{2}\left(\text{Cos}[2\,\pi\,t]\;S^x + 6.9115\,S^z + S^y\,\text{Sin}[2\,\pi\,t]\right)$$

```
In[•]:= opHR =
         Rotation[-ω t, S[3]] ** opH ** Rotation[ω t, S[3]] - ω S[3] / 2 // Elaborate // Chop
```

$$\text{Out[•]=}\quad \frac{S^x}{2} + 0.314159\,S^z$$

```
In[•]:= opUR[t_] = MultiplyExp[-I t opHR]
```

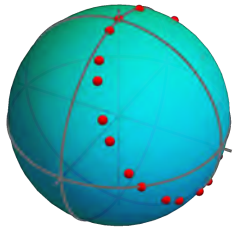
$$\text{Out[•]=}\quad e^{-i\,t\left(\frac{S^x}{2}+0.314159\,S^z\right)}$$

```
        vec[t_] = opUR[t] ** Ket[] // Elaborate;
```

This illustrates the precession in the rotating frame, which is called the Rabi oscillation.

```
In[•]:= vv = BlochVector /@ Table[Chop@vec[t], {t, 0, 8, 0.5}];
        BlochSphere[{Red, Bead /@ vv}, ImageSize → Small]
```

Out[•]=

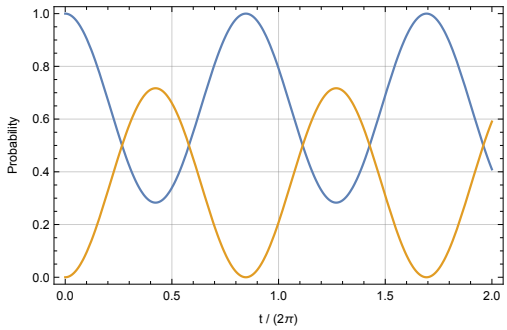


This shows the transition probabilities in the logical basis states. Note that the probabilities are the same both in the lab and rotating frame. The Rabi oscillation frequency is in this particular example is close to one---it is exactly one at resonance.

```
        prob[t_] = Abs[Normal@Matrix@vec[t]]^2;
```

```
In[•]:= Plot[Evaluate@prob[2 Pi t], {t, 0, 2},
         FrameLabel → {"t / (2π)", "Probability"}]
```

Out[•]=



Let us consider the case exactly at the resonance.

```
ω = 2 Pi;
B[1] = Cos[ω t];
B[2] = Sin[ω t];
B[3] = ω; (* resonance *)
```

```
In[•]:= opH = Dot[B@{1, 2, 3}, S@{1, 2, 3}] / 2
        opHR =
         Rotation[-ω t, S[3]] ** opH ** Rotation[ω t, S[3]] - ω S[3] / 2 // Elaborate // Chop
        opUR[t_] = MultiplyExp[-I t opHR]
```

Out[•]=
$$\frac{1}{2}\left(\text{Cos}[2\,\pi\,t]\;S^x + 2\,\pi\,S^z + S^y\,\text{Sin}[2\,\pi\,t]\right)$$

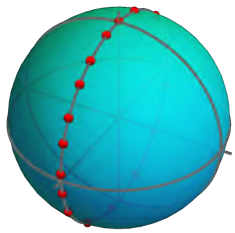
Out[•]=
$$\frac{S^x}{2}$$

Out[•]=
$$e^{-\frac{1}{2}\,i\,t\,S^x}$$

The Rabi oscillation now corresponds to the Larmor precession around the x-axis in the rotating frame.

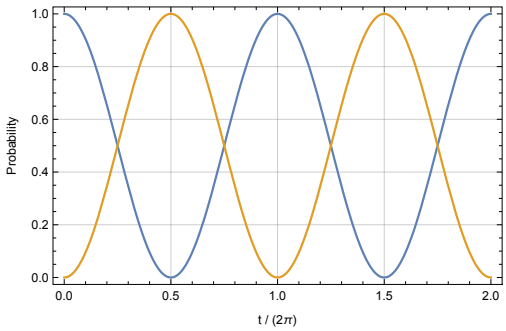
```
In[•]:= vec[t_] = opUR[t] ** Ket[] // Elaborate;
        vv = BlochVector /@ Table[Chop@vec[t], {t, 0, 8, 0.5}];
        BlochSphere[{Red, Bead /@ vv}, ImageSize -> Small]
```

Out[•]=



As the precession axis is exactly along the x-axis, the maximum transition probability can reach unity.

```
In[•]:= prob[t_] = Abs[Normal@Matrix@vec[t]]^2;
        Plot[Evaluate@prob[2 Pi t], {t, 0, 2},
         FrameLabel → {"t / (2π)", "Probability"}]
```



In the above discussion of Rabi oscillation, the two parameters are deliberately manipulated periodically with relative phase difference of $\pi/2$. It may not always be possible in practical experimental situations. Fortuantely, however, the requirement is not so stringent: Suppose that only one parameter can be driven periodically, say, $B_y(t) = 2B_\perp \sin(\omega t)$ (notice the fator 2 introduced here for convenience). The time-dependent Hamiltonian

$$\hat{H}(t) = B_\perp \sin(\omega t)\hat{S}^y + \frac{1}{2}B_0\hat{S}^z \tag{3.14}$$

looks seemingly simpler than the one in (3.5), but it does not allow for an exact solution. Instead, let us rewrite it into the form

$$\hat{H}(t) = \frac{1}{2}B_0\hat{S}^z + \frac{1}{2}B_\perp\left[\cos(\omega t)\hat{S}^x + \sin(\omega t)\hat{S}^y\right] - \frac{1}{2}B_\perp\left[\cos(-\omega t)\hat{S}^x + \sin(-\omega t)\hat{S}^y\right], \tag{3.15}$$

which is obviously the same as (3.14). While the second term describes a fictitious magnetic field rotating in the counterclockwise direction, the field in the last term rotates in the clockwise direction. In the frame rotating in the counterclockwise sense, the Hamiltonian reads as

$$\hat{H}_R(t) = \frac{1}{2}(B_0 - \omega)\hat{S}^z + \frac{1}{2}B_\perp\hat{S}^x - \frac{1}{2}B_\perp\left[\cos(-2\omega t)\hat{S}^x + \sin(-2\omega t)\hat{S}^y\right]. \tag{3.16}$$

The first two terms describe the Rabi oscillation discussed above. On the other hand, the last term oscillates fast with frequency 2ω, which is typically much larger than Ω_R. Such an oscillation is therefore too fast for the system to respond to, and its effect is negligible as the typical time scale of the system is fixed by the Rabi frequency Ω_R in (3.13). On this ground, assuming $\omega \approx B_0$, one often drops the fast oscillating term from the Hamiltonian (3.16) so that

$$\hat{H}(t) = B_\perp \sin(\omega t)\hat{S}^y + \frac{1}{2}B_0\hat{S}^z \approx \frac{1}{2}B_\perp\left[\cos(\omega t)\hat{S}^x + \sin(\omega t)\hat{S}^y\right] + \frac{1}{2}B_0\hat{S}^z. \tag{3.17}$$

Such an approximation is called the *rotating-wave approximation*. The approximation is valid as long as $B_0 \gg \Omega_R$.

3.2.2 Implementation of CNOT

CNOT is a vital gate operation in any quantum algorithms. Seemingly simple, it is not trivial to physically realize in realistic system. A typical obstacle is that the Hamiltonian, in particular the coupling between two qubits, is restricted to certain limited forms. Here we take a few examples of the qubit-qubit interaction and see how one can use it to implement the CNOT gate. These examples should be enough to give general idea about what is required for the implementation of CNOT or similar gate operations in a given realistic physical system.

One of the most common form of the exchange interaction between two qubits is the so-called *Heisenberg exchange interaction*

$$\hat{H} = J(\hat{S}_1^x \hat{S}_2^x + \hat{S}_1^y \hat{S}_2^y + \hat{S}_1^z \hat{S}_2^z)\,. \tag{3.18}$$

As one can see from its matrix representation

$$\hat{H} \doteq J \begin{bmatrix} 1 & & & \\ & -1 & 2 & \\ & 2 & -1 & \\ & & & 1 \end{bmatrix}, \tag{3.19}$$

it operates nontrivially only within the subspace spanned by $|01\rangle$ and $|10\rangle$ just like the SWAP gate discussed in Section 2.2.1—see Eq. (2.42). This is because the Heisenberg exchange coupling conserves the total angular momentum. More explicitly, up to an constant shift and multiplication factor, the Heisenberg exchange Hamiltonian equals to the SWAP gate

$$\frac{J + \hat{H}}{2J} \doteq \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & 1 & 0 & \\ & & & 1 \end{bmatrix} \doteq \text{SWAP}\,. \tag{3.20}$$

The SWAP gate behaves like the NOT gate within the subspace spanned by $|01\rangle$ and $|10\rangle$, and hence the rotation around the x-axis—in the Bloch sphere corresponding to the subspace—by angle π results in the NOT gate. It implies that by tuning the exchange coupling constant and the operation time τ such that $J\tau = \pi/4$ achieves the SWAP gate (Loss & DiVincenzo, 1998)

$$\text{SWAP} = \exp\left[-\frac{i\pi}{4}\left(\hat{S}_1^x \hat{S}_2^x + \hat{S}_1^y \hat{S}_2^y + \hat{S}_1^z \hat{S}_2^z - 1\right)\right] \tag{3.21}$$

As pointed out in Section 2.2.1, the SWAP gate itself is not universal. However, combining the $\sqrt{\text{SWAP}}$ gate with single qubit gates, one can implement the CZ gate (Loss & DiVincenzo, 1998). The CZ gate is universal. Or, it take two Pauli X gates to construct the CNOT gate from the CZ gate.

Consider the Heisenberg exchange coupling.

```
In[•]:= op = MultiplyDot[S[1, All], S[2, All]]
```

$$\text{Out[•]=}\quad S_1^x S_2^x + S_1^y S_2^y + S_1^z S_2^z$$

This is the matrix representation.

```
In[•]:= mat = Matrix[(1 + op) / 2];
        mat // MatrixForm
```

$$\text{Out[•]//MatrixForm=}\quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
In[•]:= opU = MultiplyExp[-I (Pi / 4) * (op - 1)]
        opU // Elaborate
```

$$\text{Out[•]=}\quad e^{-\frac{1}{4} i \pi \left(-1+S_1^x S_2^x+S_1^y S_2^y+S_1^z S_2^z\right)}$$

$$\text{Out[•]=}\quad \frac{1}{2} + \frac{1}{2} S_1^x S_2^x + \frac{1}{2} S_1^y S_2^y + \frac{1}{2} S_1^z S_2^z$$

```
In[•]:= matU = Matrix[opU];
        matU // MatrixForm
```

$$\text{Out[•]//MatrixForm=}\quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Unfortunately, the SWAP gate is not universal—the limitation originates from the fact that the Heisenberg exchange interaction conserves the total angular momentum. However, the $\sqrt{\text{SWAP}}$ gate is universal. In fact, one can construct the CZ gate, which is almost equivalent to the CNOT gate, from the $\sqrt{\text{SWAP}}$ gate as discussed in Section 2.2.1. Summing up, when the Heisenberg exchange interaction is available in a system of qubits, one can achive the CZ gate using two $\sqrt{\text{SWAP}}$ gates and three single-qubit gates—if desired, one can apply two additional Hadamard gates to implement the CNOT gate.

There are two additional types of qubit-qubit echange interaction, the XY and Isign exchange interaction. The *XY exchange interaction*—also known as the *planar exchange interaction*—is described by the Hamiltonian

$$\hat{H} = J(\hat{S}_1^x \hat{S}_2^x + \hat{S}_1^y \hat{S}_2^y)\,. \tag{3.22}$$

The matrix representation

$$\hat{H} \doteq \begin{bmatrix} 0 & & & \\ & 0 & 2 & \\ & 2 & 0 & \\ & & & 0 \end{bmatrix} \tag{3.23}$$

suggests that one can use the XY exchange interaction in an essentially the same way as the Heisenberg exchange interaction. An explicit construction is left for an exercise—see Problem 2.

Next, consider the XY exchange interaction.

```
In[•]:= op = MultiplyDot[S[1, {1, 2}], S[2, {1, 2}]]
Out[•]= S[1]^x S[2]^x + S[1]^y S[2]^y

In[•]:= mat = Matrix[op];
        mat // MatrixForm
Out[•]//MatrixForm=
        ( 0 0 0 0 )
        ( 0 0 2 0 )
        ( 0 2 0 0 )
        ( 0 0 0 0 )

In[•]:= opU = MultiplyExp[-I ϕ op]
Out[•]= e^(-i ϕ (S[1]^x S[2]^x + S[1]^y S[2]^y))

In[•]:= Matrix[opU] // MatrixForm
Out[•]//MatrixForm=
        (  1          0             0          0 )
        (  0      Cos[2 ϕ]     -i Sin[2 ϕ]     0 )
        (  0     -i Sin[2 ϕ]      Cos[2 ϕ]     0 )
        (  0          0             0          1 )
```

The *Ising exchange interaction* has the Hamiltonian

$$\hat{H} = J\hat{S}_1^z \hat{S}_2^z \,, \tag{3.24}$$

where the z-component has no special meaning, and the x- or y-component has the same effect. The Ising exchange interaction allows for a more direct implementation of the CZ gate: Although a direct investigation is enough to see it, let us here take another view of the CZ gate for the heuristic purposes: Recall that the CZ gate is defined as

$$\mathrm{CZ} = \sum_{x=0,1} |x\rangle\langle x| \otimes \hat{Z}^x = i \sum_x |x\rangle\langle x| \otimes e^{-i\pi x \hat{Z}/2} \tag{3.25}$$

As $x = 0, 1$ are the eigenvalues of $|1\rangle\langle 1| = (\hat{I} - \hat{Z})/2$,

$$\mathrm{CZ} = e^{i\pi/4} \exp\left[-\frac{i\pi}{4} (\hat{Z} \otimes \hat{I} + \hat{I} \otimes \hat{Z} - \hat{Z} \otimes \hat{Z}) \right] \tag{3.26}$$

We note that the exponent involves the Ising exchange interaction between the two qubits. Therefore, one can implement the CZ gate with the first and second qubit as the control and target qubit, respectively, in terms of the Hamiltonian (up to an irrelevant global phase factor $e^{i\pi/4}$) of the form

$$\hat{H} = B(\hat{S}_1^z + \hat{S}_2^z) - J\hat{S}_1^z \hat{S}_2^z \,. \tag{3.27}$$

Note that the exchange coupling—apart from the first, single-qubit term—is of the Ising type. Putting $B = J$, the time-evolution operator governed by the Hamiltonian is given by

$$\hat{U}(t) = \exp\left[-iJt(\hat{S}_1^z + \hat{S}_2^z - \hat{S}_1^z \hat{S}_2^z) \right] . \tag{3.28}$$

Therefore, the CZ gate is achieved by tuing the parameters B and J and the operation time such that

$$J\tau = B\tau = \frac{\pi}{4} \,. \tag{3.29}$$

Let us now consider the Ising exchange interaction. Here we have introduced an additional single-qubit term controlling the second qubit.

```
In[•]:= op = S[1, 3] + S[2, 3] - S[1, 3] ** S[2, 3]
Out[•]= -S1z S2z + S1z + S2z

In[•]:= mat = Matrix[op];
        mat // MatrixForm
Out[•]//MatrixForm=
        ( 1  0  0  0 )
        ( 0  1  0  0 )
        ( 0  0  1  0 )
        ( 0  0  0 -3 )

In[•]:= opU = MultiplyExp[-I ϕ op]
        opU // Matrix // MatrixForm
Out[•]= e^(-i ϕ (-S1z S2z+S1z+S2z))
Out[•]//MatrixForm=
        ( e^(-i ϕ)    0        0        0      )
        (    0     e^(-i ϕ)    0        0      )
        (    0        0     e^(-i ϕ)    0      )
        (    0        0        0     e^(3 i ϕ) )

In[•]:= opU = Exp[I Pi / 4] × MultiplyExp[-I Pi / 4 op]
        opU // Matrix // MatrixForm
Out[•]= e^(i π/4) e^(-1/4 i π (-S1z S2z+S1z+S2z))
Out[•]//MatrixForm=
        ( 1  0  0  0 )
        ( 0  1  0  0 )
        ( 0  0  1  0 )
        ( 0  0  0 -1 )
```

3.3 Geometric/Topological Scheme

When a system undergoes a cyclic adiabatic process starting from a particular eigenstate of the Hamiltonian, the system remains in the same energy level without making a transition to other energy levels. However, the quantum state of the system acquires a phase factor from two contributions. One is responsible for the usual dynamical accumulation and the other results from the geometric properties of the parameter space. The latter is called the *geometric phase* of the cyclic adiabatic process (Berry, 1984). When the energy level is degenerate and associated with a multi-dimensional eigen-subspace, the geometric phase becomes non-Abelian, that is, the quantum state evolves to another state within the subsapce. The unitary transformation between the initial and final state is called the *non-Abelian geometric phase* (Wilczek & Zee, 1984). Non-Abelian geometric phase can be extended to any *cyclic evolution*, without restriction by the adiabatic condition (Aharonov & Anandan, 1987; Anandan, 1988).

The *geometric scheme* of quantum computation (or simpley *geometric quantum computation* for short) is to implement the unitary gate operations by means of the non-Abelian geometric phases (Zanardi & Rasetti, 1999; Sjöqvist *et al.*, 2012). The geometric scheme is stable against random fluctuations of the parameters as

it depends on the geometric path in the parameter space rather than the detailed time-dependence of the parameters.

In this section, we will give a brief overview of the geometric scheme. Consider a cyclic process where the Hamiltonian changes in time through the control parameters λ_μ with $\mu = 1, 2, \dots$

$$\hat{H}(t) = \hat{H}(\boldsymbol{\lambda}(t)) \tag{3.30}$$

where we adopted a row-vector notation

$$\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_\mu, \dots) \tag{3.31}$$

to collectively denote the control parameters. When the Hamiltonian $\hat{H}(t)$ is the idealistic one in (3.1) for qubits, of course, the control parameters $\boldsymbol{\lambda}$ refer to the parameters B_j^μ and J_{ij}^μ. However, the Hamiltonian for a geometric quantum computation is usually much more general and does not contains elements menifesting peculiar form of qubits. The logical qubits of a geometric scheme is often implicitly encoded into the subsapce undergoing the cyclic evolution.

Let the states $|\alpha_j(t)\rangle$ $(j = 1, 2, \dots)$ form an *instantaneous basis* of the Hilbert space $\mathcal{H}$. These states are often chosen to be the instantaneous eigenstates of the Hamiltonian $\hat{H}(t)$, but the choice is completely arbitary as long as they are *cylic*, $|\alpha_j(\tau)\rangle = |\alpha_j(0)\rangle$, and *smooth* enough to be differentiable with respect to time (ane hence the parameters λ_μ) and . Suppose that a physical state $|\psi(t')\rangle$ at a given time t' is expanded in the instantaneous basis $|\alpha_j(t')\rangle$ as

$$|\psi(t')\rangle = \sum_j |\alpha_j(t')\rangle \langle\alpha_j(t')|\psi(t')\rangle \,. \tag{3.32}$$

At later time $t > t'$, it must be expanded in another instantaneous basis $|\alpha_i(t)\rangle$ in the form

$$|\psi(t)\rangle = \sum_i |\alpha_i(t)\rangle \, U_{ij}(t, t') \, \langle\alpha_j(t')|\psi(t')\rangle \;, \tag{3.33}$$

where $U_{ij}(t, t')$ is a unitary matrix which describes the basis change to $|\alpha_i(t)\rangle$ from $|\alpha_j(t')\rangle$—see Eqs. (A.4) and (A.5). Putting (3.33) into the Schrödinger equation, one can see that the unitary matrix $U(t, t')$ satisfies the dynamical equation

$$i\frac{\partial}{\partial t}U(t, t') = [H(t) - iA(t)]U(t, t') \,, \tag{3.34}$$

where $H(t)$ is the matrix representation of the Hamiltonian $\hat{H}(t)$ in the instantaneous basis $|\alpha_j(t)\rangle$ and the matrix $A(t)$ has been defined by

$$A_{ij}(t) := \langle\alpha_i(t)| \frac{d}{dt} |\alpha_j(t)\rangle \,. \tag{3.35}$$

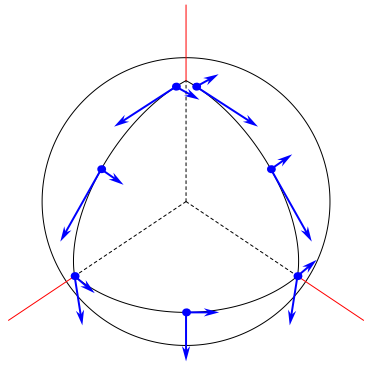


Figure 3.2: Illustration of a parallel transport on the unit sphere.

The additional term $-iA(t)$ in (3.34) is a non-inertial effect similar to the one we observed—see Eq. (3.10)—in the rotating frame in the Rabi oscillation. The solution to the equation (3.34) is formally given by

$$U(t,t') = \mathrm{Texp}\left[-\int_{t'}^{t} ds\,\{A(s)+iH(s)\}\right], \tag{3.36}$$

where T denotes the time ordering.

So far, everything has been completely general. A geometric quantum computation requires to identify a dynamic subspace $\mathcal{K}(t)\subset\mathcal{H}$ such that

(a) $\mathcal{K}(t)$ undergoes a *cyclic evolution*, $\mathcal{K}(\tau)=\mathcal{K}(0)$;

(b) $\hat{H}(t)$ vanishes within $\mathcal{K}(t)$ at all time $0<t<\tau$.

With these two conditions satisfied, the unitary matrix governing the evolution of the state vector $|\psi(t)\rangle$ in (3.33) is determined solely by the matrix $A(t)$. Furthermore, using the chain rule for the total derivative with respect to t, we can rewreite the matrix A into the form

$$A_{ij}(t) = \sum_{\mu}\frac{d\lambda_\mu}{dt}\,\langle\alpha_i|\tfrac{\partial}{\partial\lambda_\mu}|\alpha_j\rangle = \sum_{\mu}\frac{d\lambda_\mu}{dt}A^{\mu}_{ij}\,, \tag{3.37}$$

where we have put

$$A^{\mu}_{ij} := \langle\alpha_i|\tfrac{\partial}{\partial\lambda_\mu}|\alpha_j\rangle\ . \tag{3.38}$$

It implies that the unitary matrix $U(\tau,0)$ for the whole cycle is completely determined by the closed path in the parameter space that is traversed by the parameters $\lambda_\mu(t)$. That is, for a given loop $\mathcal{C}$ in the parameter space, the unitary matrix $U(\tau,0)=U(\mathcal{C})$ is given by

$$U(\mathcal{C}) = \mathrm{Pexp}\left[-\sum_{\mu}\oint_{\mathcal{C}} d\lambda_\mu\, A^{\mu}\right], \tag{3.39}$$

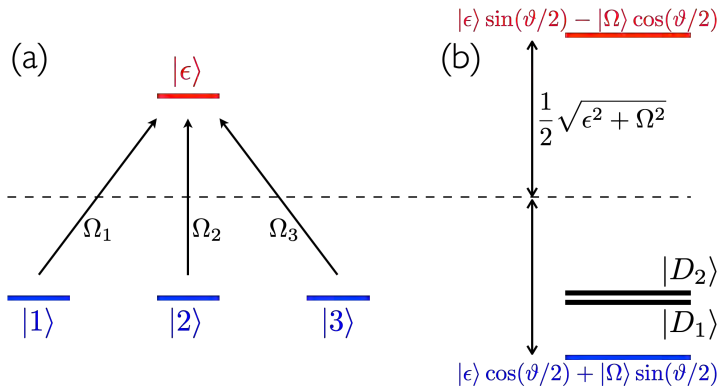


Figure 3.3: A schematic of the level structure of a toy model (see the text) for the geometric quantum computation. (a) The couplings between the ground-state levels and the excited-state level. (b) The eigenstates and corresponding eigenenergies of the model with two dark states $|D_1\rangle$ and $|D_2\rangle$. Here $\Omega := \sqrt{\Omega_1^2 + \Omega_2^2 + \Omega_3^2}$, $|\Omega\rangle := \Omega^{-1} \sum_{j=1}^{3} |j\rangle \, \Omega_j$, and $\tan\vartheta := \Omega/\epsilon$.

where P denotes the path ordering.

Suppose that one chooses a different basis, say,

$$|\beta_j\rangle := \hat{V} \, |\alpha_j\rangle = \sum_i |\alpha_i\rangle \, V_{ij} \, , \tag{3.40}$$

where $\hat{V}$ is a unitary operator and V is its unitary representation in the original basis $\{|\alpha_j\rangle\}$. Under the change of basis, the matrix A^μ transforms as

$$A^\mu \mapsto V^\dagger A^\mu V + V^\dagger \frac{\partial V}{\partial \lambda_\mu} \, , \tag{3.41}$$

the typical way a vector potential transforms under the gauge transformation in quantum mechanics. In this sense, A^μ is called the *non-Abelian gauge potential*, and describes the connection between the bases, $|\alpha_i(\lambda_\mu)\rangle$ and $|\alpha_j(\lambda_\mu + \delta\lambda_\mu)\rangle$, at infinitesimally different points along the path $\mathcal{C}$ in the parameter space. The gauge connection A^μ and the corresponding non-Abelian geometric phase $U(\mathcal{C})$ are in close analogy with the parallel transport in curved space illustrated in Fig. 3.2.

In short, the geometric quantum computation is to implement the quantum gate operations by means of the unitary transformation $U(\mathcal{C})$ in (3.39). The computational space is identified by the subspace $\mathcal{K}$ undergoing a cyclic evolution. Different choices of the closed path $\mathcal{C}$ results in different quantum logic gates.

As an example, let us consider a toy mode (Choi, 2003)

$$\hat{H} = \epsilon \, |\epsilon\rangle \, \langle\epsilon| - \frac{1}{2} \sum_{j=1}^{3} (\Omega_j \, |j\rangle \, \langle\epsilon| + h.c.) \tag{3.42}$$

consisting of four levels—see Fig. 3.3. For simplicity, we assume that $\Omega_j \in \mathbb{R}$. Put

$$|\Omega\rangle := \frac{1}{\Omega}\sum_{j=1}^{3} |j\rangle\, \Omega_j\,, \tag{3.43}$$

where $\Omega := \sqrt{\Omega_1^2 + \Omega_2^2 + \Omega_3^2}$. Then, the Hamiltonian reads as

$$\hat{H} = \epsilon\, |\epsilon\rangle\, \langle\epsilon| - \frac{1}{2}\Omega(|\Omega\rangle\, \langle\epsilon| + |\epsilon\rangle\, \langle\Omega|)\,. \tag{3.44}$$

One can immediately identify two eigenstates $|\epsilon\rangle \cos(\vartheta/2) + |\Omega\rangle \sin(\vartheta/2)$ and $|\epsilon\rangle \sin(\vartheta/2) - |\Omega\rangle \cos(\vartheta/2)$, where $\tan\vartheta := \Omega/\epsilon$, with eigenenergies $(\epsilon \mp \sqrt{\epsilon^2 + \Omega^2})/2$. More interesting is the fact that the two states $|\epsilon\rangle$ and $|\Omega\rangle$—equivalently, the two eigenstates mentioned above—spans only part of the four-dimensional Hilbert space, and there must be two more states. These additional states, which we denote by $|D_j\rangle$ $(j = 1, 2)$, do not appear in the Hamiltonian at all—they are completely decoupled from the rest. In this sense, the two states $|D_1\rangle$ and $|D_2\rangle$ are called the "dark state" whereas the state $|\Omega\rangle$, which couples to the excited-state $|\epsilon\rangle$, is called the "bright state". Note that the dark states are orthogonal to both $|\Omega\rangle$ and $|\epsilon\rangle$.

In this toy model, the two dark states, $|D_1\rangle$ and $|D_2\rangle$, form a degenerate subspace $\mathcal{K}$ of our interest. It is convenient to take the parameterization

$$\Omega_1 = \Omega \sin\theta \cos\phi\,, \quad \Omega_2 = \Omega \sin\theta \sin\phi\,, \quad \Omega_3 = \Omega\cos\theta \tag{3.45}$$

which leads to (unnormalized)

$$|\Omega\rangle = |1\rangle \sin\theta\cos\phi + |2\rangle \sin\theta\sin\phi + |3\rangle\cos\theta \tag{3.46}$$

and (unnormalized)

$$|D_1\rangle = |1\rangle \sin\phi - |2\rangle\cos\phi \tag{3.47a}$$

$$|D_2\rangle = |1\rangle \cos\theta\cos\phi + |2\rangle \cos\theta\sin\phi - |3\rangle \sin\theta \tag{3.47b}$$

Now consider the non-Abelian geometric phase for the path with fixed Ω_3—with θ fixed. We see that

$$A^{\phi}_{jk} := \langle D_j | \partial_\phi D_k\rangle = -i\cos\theta\, Y_{jk}\,, \tag{3.48}$$

where Y is the Pauli Y matrix. As a result, the cyclic evolution gives rise to the following unitary transformation

$$U = \mathrm{Pexp}\left(-\oint d\phi\, A^{\phi}\right) = \exp(i\sigma^y\, 2\pi\cos\theta) = U_y(4\pi\cos\theta)\,, \tag{3.49}$$

where $U_y(\varphi)$ is the rotation around the y-axis by angle φ. We have just demonstrated a single-qubit unitary gate operation can be achieved by purely geometric

means. Two-qubit gate operations can also be implemented in a similar way (Choi, 2003).

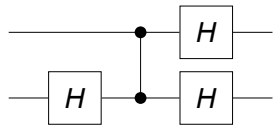
The *topological scheme* of quantm computation is a peculiar case of geometric scheme.

...

3.4 Measurement-Based Scheme

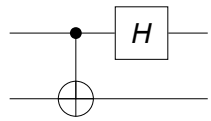
The fundametal postulates of quantum mechanics discussed at the very beginning of the book states that the quantum state of a system changes as a consequence of two difference causes. One is the time-evolution governed by the Schrödinger equation. The dynamical ang geometric scheme of quantum computation are both ultimately based on the time evolution. The other cause for the change of quantum states is the collapse of quantum states after measurement (see Postulate 3). Can we also use it for quantum computation? At a first glance, it look difficult because of the uncontrolable nature of the quantum state collapse. Recently, it have been found that quantum computation is possible just by means of measurements in different bases provided the system is prepared in a special quantum state called the *cluster state* or, more generally, *graph state* (Raussendorf & Briegel, 2001; Raussendorf *et al.*, 2002). Such a scheme of quantum computation is called the *measurement-based quantum computation* or *one-way quantum computation*—"one-way" for the reason to be clear below. There are several methods to technically realize the measurement-based quantum computation. Here we just introduce the elementary idea, and refer interested readers to the aforementioned articles and follow-up literature.

To get an idea how the measurement-based scheme works, first consider the quantum circuit model of the simple form



. (3.50)

Of course, the first three quantum circuit elements combine to construct the CNOT gate,



, (3.51)

but here we deliberately use the CZ gate for a better connection to the measurement-based scheme. Suppose that the two qubits are initially prepared in a product

state $|\psi\rangle \otimes |0\rangle$, where the state $|\psi\rangle : |0\rangle c_0 + |1\rangle c_1$ is completely general. The output state through the quantum circuit model is given by

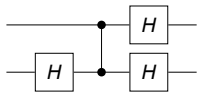
$$\frac{|0\rangle \otimes |\psi\rangle + |1\rangle \otimes (\hat{Z}\,|\psi\rangle)}{\sqrt{2}}\,. \tag{3.52}$$

When the first qubit is measured, the state of the second qubit is set to different states depending on the measurement outcome: When the outcome is 0, the state is identitical to the input state of the first qubit whereas it is $\hat{Z}\,|\psi\rangle$ if the outcome is 1. In any case, the state of the second qubit can be set to the (unknown) input state of the first qubit—with an additional operation $\hat{Z}$ if necessary—as we know the outcome out of the measurement.

Consider the following quantum circuit model.

```
qc = QuantumCircuit[S[2, 6], CZ[S[1], S[2]], S[{1, 2}, 6]]
```

Out[•]=



Suppose that the input state of the first qubit is an arbitrary superposition state while the second qubit is set to $\left|0\right\rangle$.

```
In[•]:= Let[Complex, c]
        in = Ket[] × c[0] + Ket[S[1] → 1] × c[1];
        QuissoFactor[LogicalForm[in, S@{1, 2}]]
```

Out[•]= $\left(c_0 \left|0_{S_1}\right\rangle + c_1 \left|1_{S_1}\right\rangle\right) \otimes \left|0_{S_2}\right\rangle$

This is the output state.

```
In[•]:= out = qc ** in
```

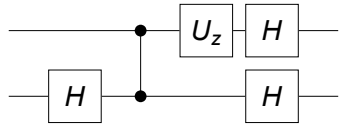
Out[•]= $\frac{c_0 \left|\_\right\rangle}{\sqrt{2}} + \frac{c_0 \left|1_{S_1}\right\rangle}{\sqrt{2}} - \frac{c_1 \left|1_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}} + \frac{c_1 \left|1_{S_2}\right\rangle}{\sqrt{2}}$

As you can see here, when the first qubit is measured and the outcome is 0, then the second qubit is identical to the input state of the first qubit. In case the measurement outcome is 1, the second qubit is set to the initial state of the first qubit with the Pauli Z operated.

```
In[•]:= QuissoFactor[out, S[1]] // LogicalForm
```

Out[•]= $\left|0_{S_1}\right\rangle \otimes \left(\frac{c_0 \left|0_{S_2}\right\rangle}{\sqrt{2}} + \frac{c_1 \left|1_{S_2}\right\rangle}{\sqrt{2}}\right) + \left|1_{S_1}\right\rangle \otimes \left(\frac{c_0 \left|0_{S_2}\right\rangle}{\sqrt{2}} - \frac{c_1 \left|1_{S_2}\right\rangle}{\sqrt{2}}\right)$

Next, consider a slightly different quantum circuit model

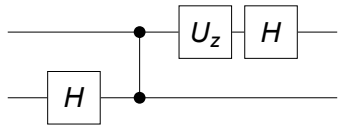


. (3.53)

Compared with the previous quantum circuit model, it has a single-qubit rotation $\hat{U}_z$ around the z-axis on the first qubit. For the same initial product state $|\psi\rangle \otimes |0\rangle$, the output state is given by

$$\frac{|0\rangle \otimes (\hat{U}_z |\psi\rangle) + |1\rangle \otimes (\hat{Z}\hat{U}_z |\psi\rangle)}{\sqrt{2}} . \tag{3.54}$$

Upon the measurement of the first qubit, the output state of the second qubit becomes either $\hat{U}_z |\psi\rangle$ or $\hat{Z}\hat{U}_z$ depending on the measurement outcome. In either case, the state $\hat{U}_z |\psi\rangle$ can be achived on the second qubit (if necessary) with an additional operation $\hat{Z}$. Here the important point is that we have achieve the state with the unitary gate $\hat{U}_z$ applied on it just by means of measurement. As the inverse of the Hadamard gate equals to itself, $\hat{H}^{-1} = \hat{H}$, removing the second Hadamdard gate on the second qubit,



, (3.55)

leads to the output state

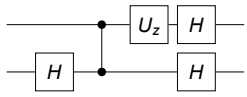
$$\frac{1}{\sqrt{2}} \sum_{x_1=0,1} |x_1\rangle \otimes (\hat{H}\hat{Z}^{x_1}\hat{U}_z |\psi\rangle) . \tag{3.56}$$

We will see below that the trailing Hadamard gate on the second qubit plays an important role to achieve an arbitary single-qubit rotation in the measurement-based scheme.

Consider the following quantum circuit model.

```
qc = QuantumCircuit[S[2, 6], CZ[S[1], S[2]], Rotation[ϕ, S[1, 3]], S[{1, 2}, 6]]
```

Out[•]=



Suppose that the input state of the first qubit is an arbitrary superposition state while the second qubit is set to $|0\rangle$.

In[•]:=
```
Let[Complex, c]
in = Ket[] × c[0] + Ket[S[1] → 1] × c[1];
QuissoFactor[LogicalForm[in, S@{1, 2}]]
```

Out[•]= $\left(c_0 \left|0_{S_1}\right\rangle + c_1 \left|1_{S_1}\right\rangle\right) \otimes \left|0_{S_2}\right\rangle$

This is the output state.

In[•]:=
```
out = qc ** in // TrigToExp
```

Out[•]= $\frac{e^{-\frac{i\phi}{2}} c_0 |\_\rangle}{\sqrt{2}} + \frac{e^{-\frac{i\phi}{2}} c_0 \left|1_{S_1}\right\rangle}{\sqrt{2}} - \frac{e^{\frac{i\phi}{2}} c_1 \left|1_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}} + \frac{e^{\frac{i\phi}{2}} c_1 \left|1_{S_2}\right\rangle}{\sqrt{2}}$

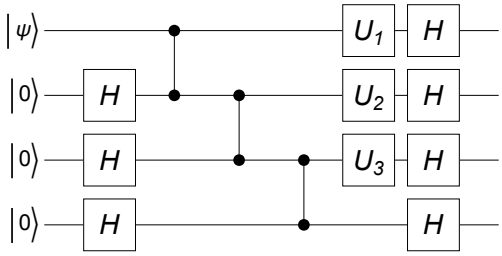
When the first qubit is measured, the output state of the second qubit depends on the measurement outcome. In either case, however, the input state of the first qubit with the unitary gate operated can be recovered in the second qubit with an operation of the Pauli Z if the measurement outcome is 1.

```
In[•]:= QuissoFactor[out, S[1]] // LogicalForm
```

$$\text{Out[•]=}\quad |0_{S_1}\rangle \otimes \left(\frac{e^{-\frac{i\phi}{2}} c_0 |0_{S_2}\rangle}{\sqrt{2}} + \frac{e^{\frac{i\phi}{2}} c_1 |1_{S_2}\rangle}{\sqrt{2}} \right) + |1_{S_1}\rangle \otimes \left(\frac{e^{-\frac{i\phi}{2}} c_0 |0_{S_2}\rangle}{\sqrt{2}} - \frac{e^{\frac{i\phi}{2}} c_1 |1_{S_2}\rangle}{\sqrt{2}} \right)$$

3.4.1 Single-Qubit Rotations

Let us examine the following quantum circuit model



, (3.57)

where the label U_j denotes the single-qubit rotations $\hat{U}_z(\phi_j)$ around the z-axis by angle ϕ_j. The output state of the above quantum circuit model can be analyzed by consecutively applying the result in (3.56) from (3.55),

$$\sum_{x_1,x_2,x_3} |x_1\rangle \otimes |x_2\rangle \otimes |x_3\rangle \otimes (\hat{Z}^{x_3}\hat{U}_z(\phi_3)\hat{H}\hat{Z}^{x_2}\hat{U}_z(\phi_2)\hat{H}\hat{Z}^{x_1}\hat{U}_z(\phi_1)\,|\psi\rangle) \tag{3.58}$$

Using the identity $\hat{H}\hat{Z}\hat{H} = \hat{X}$, the overal state on the forth qubit is given by

$$|\Psi\rangle_4 = \hat{Z}^{x_3}\hat{U}_z(\phi_3)\hat{X}^{x_2}\hat{U}_x(\phi_2)\hat{Z}^{x_1}\hat{U}_z(\phi_3)\,|\psi\rangle \tag{3.59}$$

Further, using the identities $\hat{X}\hat{Z}\hat{X} = -\hat{Z}$ and $\hat{Z}\hat{X}\hat{Z} = -\hat{X}$, it follows that

$$|\Psi\rangle_4 = \hat{Z}^{x_3}\hat{X}^{x_2}\hat{Z}^{x_1}\hat{U}_z((-1)^{x_2}\phi_3)\hat{U}_x((-1)^{x_1}\phi_2)\hat{U}_z(\phi_1)\,|\psi\rangle\ . \tag{3.60}$$

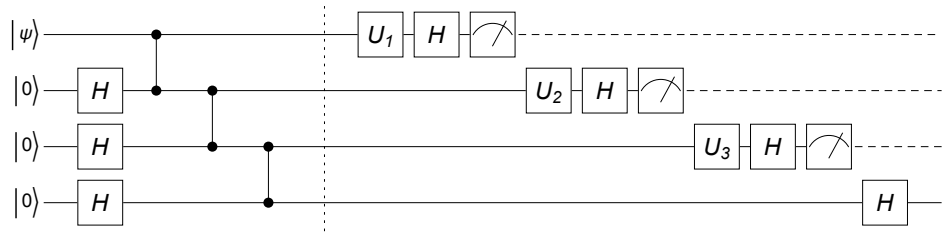
The key point to notice here that upon the measurement on the first three qubits, the forth qubit takes on the state with a single-qubit rotation

$$\hat{U}_z((-1)^{x_2}\phi_3)\hat{U}_x((-1)^{x_1}\phi_2)\hat{U}_z(\phi_3) \tag{3.61}$$

operated on the initial state $|\psi\rangle$ of the first qubit. The rotation consists of three rotations around perpendicular axes and can realize any arbitrary single-qubit rotations.

The rotation in (3.61) is not deterministic as it depends on the measurement outcome. This can be deterministic by delaying the operations on the second

and third qubits. That is, let us consider a modified quantum circuit model as following



(3.62)

Suppose that we want to implement a single-qubit Euler rotation[1]

$$\hat{U}(\alpha,\beta,\gamma) := \hat{U}_z(\alpha)\hat{U}_x(\beta)\hat{U}_z(\gamma)\,. \tag{3.63}$$

We first set $\phi_1 = \gamma$. Then, depending the measurement outcome m_1 on the first qubit, we set $\phi_2 = (-1)^m\beta$. Similarly, depending on the measurement outcome m_2 on the second qubit, we set $\phi_3 = (-1)^{m_2}\alpha$. The final state on the forth qubit will become

$$|\Psi\rangle_4 = \hat{Z}^{m_3}\hat{X}^{m_2}\hat{Z}^{m_1}\hat{U}(\alpha,\beta,\gamma)\,|\psi\rangle \tag{3.64}$$

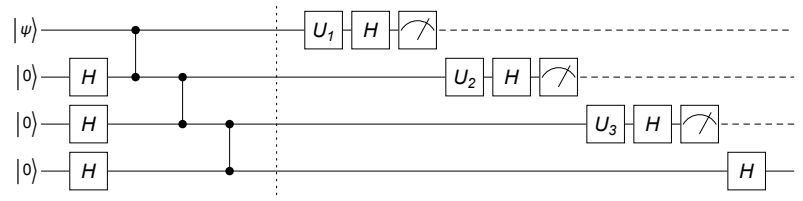
with the single-qubit rotation fixed *deterministically*. There are still undesired operations, $\hat{Z}^{m_3}\hat{X}^{m_2}\hat{Z}^{m_1}$. However, these operations are irrelevant at the end of quantum computation, and do not have to be corrected. For example, if the final state is measured in the logical basis state, $\hat{Z}$ does not affect the readout at all, and the effect of $\hat{X}$ can be handled by a classical post-processing.

Let us construct a quantum circuit model for an arbitrary single-qubit rotation.

```
Let[Real, c, ϕ]
```

```
qc = QuantumCircuit[ProductState[S[1] → {c[0], c[1]}, "Label" → Ket["ψ"]],
  LogicalForm[Ket[], S@{2, 3, 4}], S[{2, 3, 4}, 6],
  CZ[S[1], S[2]], CZ[S[2], S[3]], CZ[S[3], S[4]], "Separator",
  Rotation[ϕ[1], S[1, 3], "Label" → "U₁"], S[1, 6], Measurement[S[1]],
  Rotation[ϕ[2], S[2, 3], "Label" → "U₂"], S[2, 6], Measurement[S[2]],
  Rotation[ϕ[3], S[3, 3], "Label" → "U₃"], S[3, 6], Measurement[S[3]], S[{4}, 6]]
```

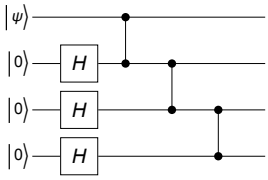
Out[ ]=



[1] Here we have adopted a different convention for the Euler rotation.

```
qc0 = QuantumCircuit[
  ProductState[S[1] → {c[0], c[1]}, "Label" → Ket["ψ"]],
  LogicalForm[Ket[], S@{2, 3, 4}],
  S[{2, 3, 4}, 6], CZ[S[1], S[2]], CZ[S[2], S[3]], CZ[S[3], S[4]]]
```

Out[•]=




```
out1 = QuantumCircuit[qc0, Rotation[ϕ[1], S[1, 3], "Label" → "U1"],
    S[1, 6], Measurement[S[1]]] // Elaborate;
m1 = Readout[out1, S[1]]
```

Out[•]= 0

```
out2 = QuantumCircuit[out1, Rotation[(-1)^m1 * ϕ[2], S[2, 3], "Label" → "U2"],
    S[2, 6], Measurement[S[2]]] // Elaborate;
m2 = Readout[out2, S[2]]
```

Out[•]= 0

```
out3 = QuantumCircuit[out2, Rotation[(-1)^m2 * ϕ[3], S[3, 3], "Label" → "U3"],
    S[3, 6], Measurement[S[3]], S[4, 6]] // Elaborate;
out3 = out3 /. {c[0]^2 + c[1]^2 → 1};
m3 = Readout[out3, S[3]]
```

Out[•]= 1

In[•]:=
```
expr = MultiplyPower[S[4, 3], m3] **
    MultiplyPower[S[4, 1], m2] ** MultiplyPower[S[4, 3], m1] **
    Rotation[ϕ[3], S[4, 3]] ** Rotation[ϕ[2], S[4, 1]] **
    Rotation[ϕ[1], S[4, 3]] ** (Ket[S[1] → m1, S[2] → m2, S[3] → m3] **
      ProductState[S[4] → {c[0], c[1]}]) // Elaborate
```

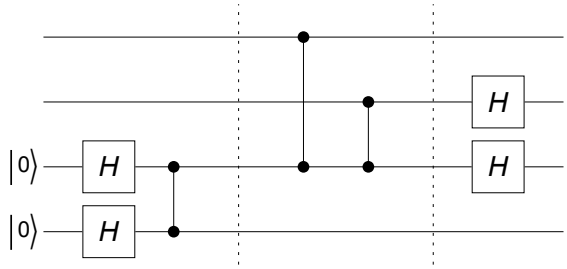
Out[•]=
$$\left|1_{S_3}\right\rangle \left(\text{Cos}\left[\frac{\phi_1}{2}\right] \left(c_0 \text{Cos}\left[\frac{\phi_2}{2}\right] - i\, c_1 \text{Sin}\left[\frac{\phi_2}{2}\right]\right) + \text{Sin}\left[\frac{\phi_1}{2}\right] \left(-i\, c_0 \text{Cos}\left[\frac{\phi_2}{2}\right] + c_1 \text{Sin}\left[\frac{\phi_2}{2}\right]\right)\right)$$
$$\left(\text{Cos}\left[\frac{\phi_3}{2}\right] - i \text{Sin}\left[\frac{\phi_3}{2}\right]\right) + \left|1_{S_3} 1_{S_4}\right\rangle$$
$$\left(\text{Cos}\left[\frac{\phi_1}{2}\right] \left(-c_1 \text{Cos}\left[\frac{\phi_2}{2}\right] + i\, c_0 \text{Sin}\left[\frac{\phi_2}{2}\right]\right) + \text{Sin}\left[\frac{\phi_1}{2}\right] \left(-i\, c_1 \text{Cos}\left[\frac{\phi_2}{2}\right] + c_0 \text{Sin}\left[\frac{\phi_2}{2}\right]\right)\right)$$
$$\left(\text{Cos}\left[\frac{\phi_3}{2}\right] + i \text{Sin}\left[\frac{\phi_3}{2}\right]\right)$$

In[•]:=
```
out3 - expr
```

Out[•]= 0

3.4.2 CNOT Gate

To get an idea how to implement the CNOT gate in the measurement-based scheme, consider the following quantum circuit model



(3.65)

For any input of the form

$$|\Psi\rangle_{\text{in}} = |x_1\rangle \otimes |x_2\rangle \otimes |0\rangle \otimes |0\rangle \tag{3.66}$$

the output state is given by

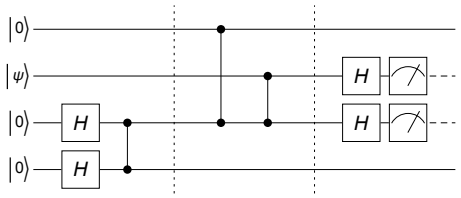
$$|\Psi\rangle_{\text{out}} = \sum_{y_2,y_3} \left(\hat{Z}^{y_2} |x_1\rangle\right) \otimes |y_2\rangle \otimes |y_3\rangle \otimes \left(\hat{X}^{y_3} \hat{Z}^{y_2} |x_1 \oplus x_2\rangle\right) \tag{3.67}$$

The input state of the target qubit is prepared in the second qubit and the output state appears in the forth qubit. Both the input and output state of the control qubit are stored in the first qubit. In many practical cases, especially in the middle of quantum computation, it is more convenient to transfer the target qubit to another qubit as well. It can be acheive using similar methods with 10 or 15 quibts (Raussendorf & Briegel, 2001; Raussendorf *et al.*, 2003).

Here we demonstrate the CNOT gate based on the measurement-based scheme.

```
in = ProductState[S[2] → {c[0], c[1]}, "Label" → Ket["ψ"]];
qc = QuantumCircuit[LogicalForm[Ket[S[1] → 0], S[1]],
   in, LogicalForm[Ket[], S@{3, 4}], S[{3, 4}, 6],
   CZ[S[3], S[4]], "Separator", CZ[S[1], S[3]], CZ[S[2], S[3]],
   "Separator", S[{2, 3}, 6], Measurement[S@{2, 3}]]
```

Out[•]=




```
In[•]:= out = Elaborate[qc] /. {c[0] × Conjugate[c[0]] + c[1] × Conjugate[c[1]] → 1};
        {m2, m3} = Readout[out, S@{2, 3}]
        QuissoFactor[out, S@{1, 2, 3}] // LogicalForm
```

Out[•]= {1, 1}

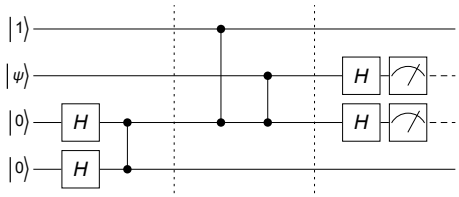
Out[•]= $\left|0_{S_1} 1_{S_2} 1_{S_3}\right\rangle \otimes \left(-c_1 \left|0_{S_4}\right\rangle + c_0 \left|1_{S_4}\right\rangle\right)$

```
In[•]:= new = MultiplyPower[S[1, 3], m2] ** MultiplyPower[S[4, 1], m3] **
           MultiplyPower[S[4, 3], m2] ** CNOT[S[1], S[4]] **
           (Ket[S@{2, 3} → {m2, m3}] ** ProductState[S[4] → {c[0], c[1]}]);
         QuissoFactor[new, S@{1, 2, 3}] // LogicalForm
Out[•]= |0_{S1} 1_{S2} 1_{S3}⟩ ⊗ (-c1 |0_{S4}⟩ + c0 |1_{S4}⟩)

In[•]:= new - out // Garner
Out[•]= 0

         in = ProductState[S[2] → {c[0], c[1]}, "Label" → Ket["ψ"]];
         qc = QuantumCircuit[LogicalForm[Ket[S[1] → 1], S[1]],
           in, LogicalForm[Ket[], S@{3, 4}], S[{3, 4}, 6],
           CZ[S[3], S[4]], "Separator", CZ[S[1], S[3]], CZ[S[2], S[3]],
           "Separator", S[{2, 3}, 6], Measurement[S@{2, 3}]]
```

Out[•]=




```
In[•]:= out = Elaborate[qc] /. {c[0] × Conjugate[c[0]] + c[1] × Conjugate[c[1]] → 1};
         {m2, m3} = Readout[out, S@{2, 3}]
         QuissoFactor[out, S@{1, 2, 3}] // LogicalForm
Out[•]= {1, 1}
Out[•]= |1_{S1} 1_{S2} 1_{S3}⟩ ⊗ (c0 |0_{S4}⟩ - c1 |1_{S4}⟩)

In[•]:= new = MultiplyPower[S[1, 3], m2] ** MultiplyPower[S[4, 1], m3] **
           MultiplyPower[S[4, 3], m2] ** CNOT[S[1], S[4]] ** (Ket[S[1] → 1] **
             Ket[S@{2, 3} → {m2, m3}] ** ProductState[S[4] → {c[0], c[1]}]);
         QuissoFactor[new, S@{1, 2, 3}] // LogicalForm
Out[•]= |1_{S1} 1_{S2} 1_{S3}⟩ ⊗ (c0 |0_{S4}⟩ - c1 |1_{S4}⟩)

In[•]:= new - out // Garner
Out[•]= 0
```

3.4.3 Graph States

For the measurement-based quantum computation, the crucial resource is the state prepared by the quantum circuit elements on the left of the vertial dashed line in (3.62). Once the state is prepared, from then on the unitary logical gate can be implemented solely by measurements in rotated bases. Such a state is called the *graph state*.

A graph state is created by applying the CZ gate on the two qubits linked by the graph.

For example, consider a linear graph.

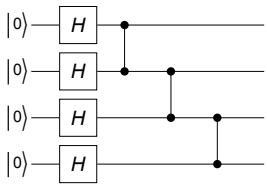
```
In[•]:= g = Graph[{S[1] ↔ S[2], S[2] ↔ S[3], S[3] ↔ S[4]}, VertexLabels → "Index"]
```

Out[•]=

The corresponding graph state is created by the following quantum circuit model.

```
qc = QuantumCircuit[LogicalForm[Ket[], S@{1, 2, 3, 4}], g]
```

Out[•]=




```
vec = Elaborate[qc];
vec // LogicalForm
```

Out[•]=

$$\frac{1}{4}\left|0_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle+\frac{1}{4}\left|0_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle+\frac{1}{4}\left|0_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle-\frac{1}{4}\left|0_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle+$$
$$\frac{1}{4}\left|0_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle+\frac{1}{4}\left|0_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle-\frac{1}{4}\left|0_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle+\frac{1}{4}\left|0_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle+$$
$$\frac{1}{4}\left|1_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle+\frac{1}{4}\left|1_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle+\frac{1}{4}\left|1_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle-\frac{1}{4}\left|1_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle-$$
$$\frac{1}{4}\left|1_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle-\frac{1}{4}\left|1_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle+\frac{1}{4}\left|1_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle-\frac{1}{4}\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle$$

The same state can be directly obtained using `GraphState`.

```
new = GraphState[g]
```

Out[•]=

$$\frac{1}{4}\left(\left|\_\right\rangle+\left|1_{S_1}\right\rangle-\left|1_{S_1}1_{S_2}\right\rangle+\left|1_{S_1}1_{S_2}1_{S_3}\right\rangle-\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle-\left|1_{S_1}1_{S_2}1_{S_4}\right\rangle+\left|1_{S_1}1_{S_3}\right\rangle-\left|1_{S_1}1_{S_3}1_{S_4}\right\rangle+\right.$$
$$\left.\left|1_{S_1}1_{S_4}\right\rangle+\left|1_{S_2}\right\rangle-\left|1_{S_2}1_{S_3}\right\rangle+\left|1_{S_2}1_{S_3}1_{S_4}\right\rangle+\left|1_{S_2}1_{S_4}\right\rangle+\left|1_{S_3}\right\rangle-\left|1_{S_3}1_{S_4}\right\rangle+\left|1_{S_4}\right\rangle\right)$$

```
vec - new // Simplify
```

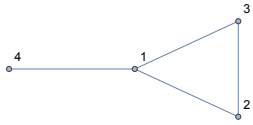
Out[•]=

```
0
```

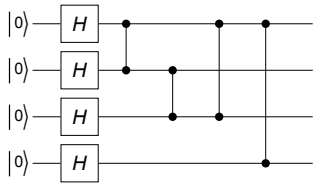
As another example, consider the following graph and corresponding quantum circuit model.

```
g = Graph[{S[1] ↔ S[2], S[2] ↔ S[3], S[1] ↔ S[3], S[1] ↔ S[4]},
  ImageSize → Small, VertexLabels → "Index"]
QuantumCircuit[LogicalForm[Ket[], S@{1, 2, 3, 4}], g]
```

Out[•]=



Out[•]=



Mathematically, a graph state can be defined for any graph connecting the qubits. However, such a state is difficult to generate in realistic systems. *Cluster states* are a special subclass of graph states. The underlying graph is an d-dimensional square grid, and the regular connectivity of the underlying graph

makes the states far more feasible. Indeed, a number of physical methods to generate cluster states have been proposed.

3.5 Spin-Boson Model*

In many architectures of quantum computers, the coupling between qubits are indirectly achieved through a bosonic mode shared by the qubits. Common examples include the quantum computers based on superconducting qubits and trapped ions. The bosonic mode can also be used for a quantum non-demolition (QND) measurement of quantum states as in the quantum computers based on superconducting circuits. Here we discuss the properties of such a spin-boson model at the elementary level.

Let us first examine the elementary properties of the model with a single-qubit coupled to a bosonic mode. The Hamiltonian is given by

$$\hat{H} = \omega \hat{a}^\dagger \hat{a} + \frac{1}{2}\Omega \hat{S}^z + g(\hat{a}^\dagger + \hat{a})\hat{S}^x \,, \tag{3.68}$$

where $\hat{a}$ and $\hat{a}^\dagger$ are the annihilation and creation operators, respectively, of the bosonic mode.

Denote the qubits by S and the cavity mode by a.

```
Let[Qubit, S]
Let[Boson, a]
```

This is the Hamiltonian.

```
In[•]:= opH = Dagger[a] ** a + Ω S[3] / 2 + g (Dagger[a] + a) ** S[1]
```

$$\text{Out[•]=}\quad a^\dagger\, a + g\left(a\, S^x + a^\dagger\, S^x\right) + \frac{\Omega\, S^z}{2}$$

Here is a *truncated* basis.

```
In[•]:= bs = Basis[{a, S}];
        LogicalForm@bs
Out[•]= {|0a0S⟩, |0a1S⟩, |1a0S⟩, |1a1S⟩, |2a0S⟩,
         |2a1S⟩, |3a0S⟩, |3a1S⟩, |4a0S⟩, |4a1S⟩, |5a0S⟩, |5a1S⟩}
```

Here is the matrix representation of the Hamiltonian (showing only part of it), and the eigenenergies for a particular set of parameters. The matrix representation is not exact as the basis has been truncated.

```
In[•]:= Ω = 1;
        g = 1 / 10;
        matH = Matrix[opH];
        matH[[ ;; 6, ;; 6]] // MatrixForm
        val = N@Eigenvalues[matH]
```

Out[•]//MatrixForm=

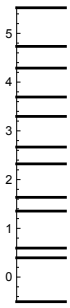
$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{10} & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{10} & 0 & 0 & 0 \\ 0 & \frac{1}{10} & \frac{3}{2} & 0 & 0 & \frac{1}{5\sqrt{2}} \\ \frac{1}{10} & 0 & 0 & \frac{1}{2} & \frac{1}{5\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{5\sqrt{2}} & \frac{5}{2} & 0 \\ 0 & 0 & \frac{1}{5\sqrt{2}} & 0 & 0 & \frac{3}{2} \end{pmatrix}$$

```
Out[•]= {5.52494, 4.7328, 4.2874, 3.69409, 3.29609, 2.66746,
         2.32239, 1.63601, 1.35389, 0.594847, -0.505013, 0.395102}
```

Here you can see the (approximate) energy levels of the spin-boson model.

```
In[•]:= levels = Line@Thread@{Thread@{0, val}, Thread@{1, val}};
        Graphics[levels,
         BaseStyle → Thick,
         Axes → {None, True}]
```

Out[•]=



...

Next we demonstrate the QND measurement of quatum states using the bosonic mode.

...

We demonstrate a basic method to induce a coupling between two qubits by sharing the single bosonic mode.

$$\hat{H} = \omega \hat{a}^\dagger \hat{a} + \frac{1}{2}\Omega \sum_{j=1}^{2} \hat{S}_j^z + g(\hat{a}^\dagger + \hat{a}) \sum_j \hat{S}_j^x \tag{3.69}$$

...

Problems

1. As the single-parameter controlled Hamiltonian in Eq. (3.16) has two fictitious magnetic fields rotating in opposite senses, it seems to be a matter of choice to pick either of the two. Argue physically why it does not work to choose the frame rotating in the clockwise sense.

2. Consider a system of two qubits interacting with each other through the XY exchange coupling in (3.22). For the given coupling constant J, find the way to physically implement the SWAP gate by tuning the operation time τ. If necessary, you can apply additional single-qubits gates. Ignore a global phase factor (if any). Once the SWAP gate is carried out, you can combine the $\sqrt{\text{SWAP}}$ gate and single-qubit gates to construct the CZ gate—see Section 2.2.1.

3. Consider a toy model similar to the one described in Eq. (3.42) and Fig. 3.3, but with two ground-state levels:

$$\hat{H} = \epsilon \left|\epsilon\right\rangle \left\langle\epsilon\right| - \frac{1}{2}\sum_{j=1}^{2}(\Omega_j \left|j\right\rangle \left\langle\epsilon\right| + h.c.)\,. \tag{3.70}$$

   Assume that

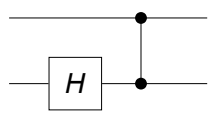
$$\Omega_1 = \cos(\theta/2)e^{-i\phi/2}\,, \quad \Omega_2 = \sin(\theta/2)e^{+i\phi/2} \quad (\theta, \phi \in \mathbb{R}). \tag{3.71}$$

   (a) Find all eigenstates and corresponding eigenvalues of the Hamiltonian $\hat{H}$. Show that there exists an eigenstate $|D\rangle$ the eigenvalue of which is always zero regardless of Ω_1 and Ω_2. $|D\rangle$ is the "dark state" of the model.

   (b) Calculate the (Abelian) gauge potential $A^{\phi}(\theta, \phi)$ for a fixed θ for the one-dimensional subspace spanned by $|D\rangle$. Plot A^{ϕ} as a function of θ and ϕ.

   (c) Calculate the (Abelian) geometric phase $U(\mathcal{C})$ for the path $\mathcal{C}$ such that θ is fixed and ϕ changes from 0 to 2π.

4. Consider the following quanutm circuit model:



   (a) Suppose that the input state is $(|0\rangle\, c_0 + |1\rangle\, c_1) \otimes |0\rangle$. Show that the output state is given by $|0\rangle \otimes |+\rangle\, c_0 + |1\rangle \otimes |-\rangle\, c_1$, where $|\pm\rangle := (|0\rangle \pm |1\rangle)/\sqrt{2}$ is the eigenstates (with eigenvalues ± 1) of the Pauli X operator.

   (b) Find the output state for the input state $(|0\rangle\, c_0 + |1\rangle\, c_1) \otimes |1\rangle$.

5. Let $\hat{U}_\mu(\phi)$ be the single-qubit rotaion around the μ-axis by angle ϕ in the Pauli space. Show

   (a) that
   $$\hat{X}\hat{U}_z(\phi)\hat{X} = \hat{U}_z(-\phi)\,, \tag{3.72}$$
   where $\hat{X}$ is the Pauli X operator; and

   (b) that
   $$\hat{H}\hat{U}_z(\phi)\hat{H} = \hat{U}_x(\phi)\,, \tag{3.73}$$
   where $\hat{H}$ is the Hadamard operator.

Chapter 4

Quantum Algorithms: Introduction

- May 22, 2021 (v1.8)

In Chapter 2, we have discussed how to carry out arbitrary computation by composing elementary quantum logic gates. In Chapter 3, we have seen what is required to physically implement those elementary quantum logic gates. The discussions are enough to establish physical and realistic models of quantum computation. However, so far, we have disregarded an important issue, that is, the efficiency of the implementations. Quantum computers turn out to be technically hard to build and error rates are still a fundamental concern for quantum computers while aforementioned calculations can be performed, in principle, on classical computers anyway. Why should quantum computation be attractive?

Not surprisingly, it was Peter Shor's factorization algorithm (Shor, 1994, 1997) that brought quantum computation to such great attention even of the public at the turn of the millennium. The factorization of large numbers was the first practically important task that is not feasible on a classical computer but can be performed efficiently on a quantum computer. In this chapter, we explore several elementary examples of quantum algorithms that efficiently solve the problems that are known to be exponentially hard with classical algorithms. Some of them may be of little use for practical applications. Nevertheless, these examples are still interesting as one can take a glimpse of ideas and features behind quantum algorithms distinguished from classical algorithms through them. In the discussion, included is quantum teleportation. It is a quantum commmunication protocol rather than a quantum algorithm. Nonetheless, we include it here because it is a simple yet fascinating example demonstrating what one can do with quantum states that is not possible at all with classical information.

Throughout the chapter, we keep using the same notation in Chapter 2, Eq. (2.3) in particular.

4.1 Quantum Teleportation

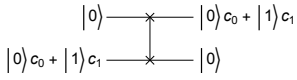
Quantum teleportation is a communication protocol to send quantum information to a distant party making use of a pre-shared pair of entangled particles. The protocol has attracted public interest as it closely resembles the (hypothetical) teleportation in the sense that an unknown quantum state is trasmitted to a system far away while it disappears from the original system. From physical point of view, it is one of the first quantum information protocols that have vividly illustrated the significance of quantum entanglement as a valuable resource.

Of course, the task will be almost trivial if the two parties are in direct contact with each other. For example, one can simply use the SWAP gate.

When one has access to both qubits, a SWAP gate for example is enough to send state from one qubit to the other.

```
In[•]:= Let[Complex, c]
        qc =
         QuantumCircuit[{LogicalForm[Ket[], S[1]], ProductState[S[2] → {c[0], c[1]}]},
          SWAP[S[1], S[2]], {LogicalForm[Ket[], S[2]],
           ProductState[S[1] → {c[0], c[1]}]}, "PortSize" → 2]
```

Out[•]=




```
In[•]:= out = QuissoFactor@ExpressionFor[qc]
```

Out[•]= $\left(c_0 \left|0_{S_1}\right\rangle + c_1 \left|1_{S_1}\right\rangle\right) \otimes \left|0_{S_2}\right\rangle$

In quantum teleportation it is assumed that there is no quantum channel available between the two parties.

In this section, we discuss the physical principle behind quantum teleportation and demonstrate its protocol. Before discussing quantum teleportation, we look into one of the intriguing properties of entangled quantum states. Another closely related quantum communication protocol is the so-called *superdense coding*. As the idea and protocol are in parallel with the quantum teleporation protocol, here we do not discuss it and refer the readers to the original work (Bennett & Wiesner, 1992).

4.1.1 Nonlocality in Entanglement

Quantum entanglement is a key resource in quantum teleportation as we will see shortly. However, it also reveals an intriguing nature of quantum mechanics, that is, *nonlocality*. Quantum entanglement and accompanying nonlocality is a largely unexpected consequence of the superposition principle of quantum states which was pointed out first by Einstein *et al.* (1935). Before we discuss an implementation of quantum teleportation, here we take a brief look at the nonlocal property buried in quantum entanglement.

Suppose that two people, Alice and Bob, share a pair of qubits that is in an entangled state

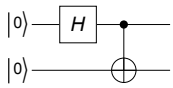
$$|\Psi\rangle = \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle}{\sqrt{2}} \tag{4.1}$$

Recall that it can be generated using the quantum entangler circuit in Section 2.2.1.

This is an entangler quantum circuit.

```
In[•]:= qc = QuantumCircuit[LogicalForm[Ket[], S@{1, 2}], S[1, 6], CNOT[S[1], S[2]]]
```

Out[•]=




```
In[•]:= vec = ExpressionFor[qc];
        vec // LogicalForm
```

Out[•]=

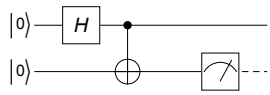
$$\frac{\left|0_{S_1} 0_{S_2}\right\rangle}{\sqrt{2}} + \frac{\left|1_{S_1} 1_{S_2}\right\rangle}{\sqrt{2}}$$

When Bob measures his qubit, the measurement readout is just random and can be either 0 or 1.

This is a quantum circuit model for the Bel measurement.

```
In[•]:= qc2 = QuantumCircuit[qc, "Spacer", Measurement@S[2]]
```

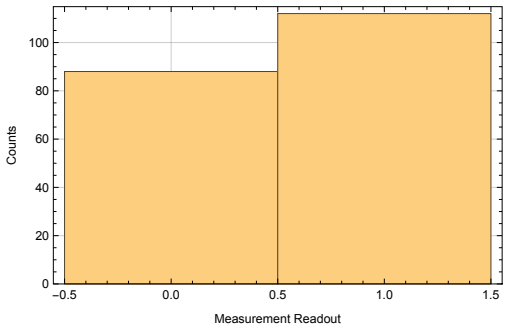
Out[•]=



This shows that the measurement outcome is just random.

```
In[•]:= val = Table[out = Elaborate[qc2] // Garner;
          Readout[out, S[2]], {200}];
        Histogram[val, FrameLabel → {"Measurement Readout", "Counts"},
         ImageSize → Automatic, FrameStyle → Automatic]
```

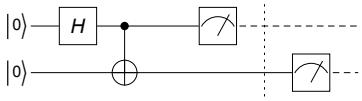
Out[•]=



However, if Alice performs a measurement on her qubit anytime before Bob's measurement, Bob's result is completely fixed by the result of Alice's measurement.

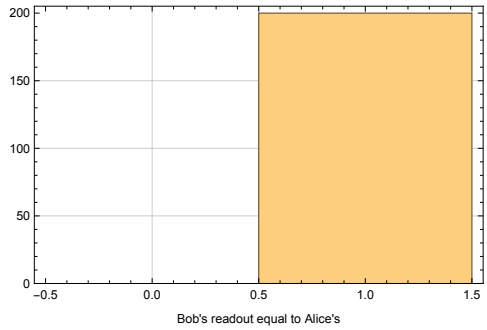
Here Alice -- possessing the first qubit -- measures her qubit before Bob measures his qubit.

```
In[•]:= qc3 = QuantumCircuit[qc, "Spacer",
           Measurement@S[1], "Separator", Measurement@S[2]]
```



This illustrates that Bob's measurement results are affected by Alice measurement.

```
In[•]:= val = Boole@Table[out = Elaborate[qc2] // Garner;
          Equal @@ Readout[out, S@{1, 2}], {200}];
        Histogram[val, {-0.5, 1.5, 1},
         FrameLabel → {"Bob's readout equal to Alice's", "Counts"},
         ImageSize → Automatic, FrameStyle → Automatic]
```



The above conclusion holds however far Alice and Bob are separated and however soon Bob measures after Alice does. Somehow Alice's measurement affects Bob's measurement "instantaneously". It seemingly violates Einstein's special theory of relativity which dictates that nothing can travel faster than light. This forced many people to hesitate to believe quantum mechanics until John Bell proposed an experimental test in terms of inequality (Bell, 1966) and actual experiments verified quantum mechanics with a result violating Bell's inequality (Aspect *et al.*, 1981). Later, another interesting test of the non-local nature of quantum mechanics was proposed by Hardy (1992) and demonstrated by other experiments.

4.1.2 Implementation of Quantum Teleportation

Now let us turn back to the quantum teleportation protocol. Suppose that Bob wants to send one bit of quantum information, say, $|\psi\rangle_C = |0\rangle_C \psi_0 + |1\rangle_C \psi_1$ stored in Charlie's qubit, to Alice residing in a place far away from Bob (and Charlie). It is important to note here that $|\psi\rangle$ is an unknown state. As Zurek (2000) declared, "You can clone a sheep,[1] but not a quantum state," the *no-cloning theorem* of quantum states dictates that it is impssible to make a copy of an unknown quantum state (Wooters & Zurek, 1982). The no-cloning nature of quantum state is one of key features of quantum states that provide *unconditional* security in quantum communication.

[1] It refers to Dolly, the sheep cloned by a research group (Wilmut *et al.*, 1997).

To teleport a quantum state, Alice and Bob need an entangled pair of qubits generated and shared sometime before, say, $|\beta_0\rangle_{AB} = (|00\rangle_{AB} + |11\rangle_{AB})/\sqrt{2}$, which is one of the four Bell states. Recall that the entanglement is generated by a combination of the Hadamard and CNOT gate—see Section 2.2.1. Thus the initial state $|\Psi\rangle$ of the three qubits at the beginning of the procedure is given by

$$|\Psi\rangle = |\beta_0\rangle_{AB} \otimes |\psi\rangle_C = \frac{|000\rangle\psi_0 + |110\rangle\psi_0 + |001\rangle\psi_1 + |111\rangle\psi_1}{\sqrt{2}} \tag{4.2}$$

Rewriting the parts consisting of the qubits B and C in the Bell basis—the basis consisting of the four Bell states—leads to

$$\begin{aligned}|\Psi\rangle = (|0\rangle\,\psi_0 + |1\rangle\,\psi_1)_A \otimes |\beta_0\rangle_{BC} + (|1\rangle\,\psi_0 + |1\rangle\,\psi_0)_A \otimes |\beta_1\rangle_{BC} \\ - (|1\rangle\,\psi_0 - |1\rangle\,\psi_0)_A \otimes |\beta_2\rangle_{BC} + (|0\rangle\,\psi_0 - |1\rangle\,\psi_1)_A \otimes |\beta_3\rangle_{BC}\end{aligned} \tag{4.3}$$

The crucial point here is that the state of A in the first term is identical to the quantum state $|\psi\rangle$, and those in the rest are also closely related to $|\psi\rangle$ by the Pauli operators. More explicitly, one can rewrite the total state vector as

$$\begin{aligned}|\Psi\rangle = |\psi\rangle_A \otimes |\beta_0\rangle_{BC} + \hat{S}_A^x\,|\psi\rangle_A \otimes |\beta_1\rangle_{BC} \\ + i\hat{S}_A^y\,|\psi\rangle_A \otimes |\beta_2\rangle_{BC} + \hat{S}_A^z\,|\psi\rangle_A \otimes |\beta_3\rangle_{BC}\end{aligned} \tag{4.4}$$

Now Bob performs a measurement on the two qubits B and C, owned by Bob, *in the Bell basis*. The measurement will yield outcomes $\mu = 0, 1, 2, 3$ and collapse the total state vector into the corresponding term, $\hat{S}_A^\mu\,|\psi\rangle_A \otimes |\beta_\mu\rangle$ (there is an additional factor of i for $\mu = 2$ but the global phase factor is physically irrelevant). The remaining task is for Bob to inform Alice of the outcome μ so that Alice recover the desired state $|\psi\rangle$ by operating the inverse operator $\hat{S}_A^\mu$ on her qubit. The information about the measurement outcome amounts to two bits, and requires only a classical channel for transmission. In short, the quantum teleportation protocol consists of the following steps:

1. Alice and Bob generate an entangled pair of qubits, A and B, and share the pair betwen them. This can be done anytime before the procedure actually starts. Bob prepares a quantum state to send in a separate qubit C.

2. Bob makes a Bell measuremen, i.e., the measurement in the Bell basis, on his two qubits B and C.

3. Bob sends the two-bit information of the measurement outcome to Alice through a classical communication channel.

4. Alice operates a proper inverse operator to recover the desired quantum state.

Here is a simulation of the quantum teleportation protocol using Q3.

The initial state of the total system at the beginning of the protocol.

```
In[•]:= Let[Complex, ψ]
        vec = Ket[] × ψ[0] + Ket[S[3] → 1] × ψ[1];
        vec // LogicalForm
```

Out[•]= $\left|0_{S_3}\right\rangle \psi_0 + \left|1_{S_3}\right\rangle \psi_1$

```
In[•]:= tot = BellState[S@{0, 2}, 0] ** vec;
        tot // LogicalForm
```

Out[•]= $\frac{\left|0_{S_0}0_{S_2}0_{S_3}\right\rangle \psi_0}{\sqrt{2}} + \frac{\left|1_{S_0}1_{S_2}0_{S_3}\right\rangle \psi_0}{\sqrt{2}} + \frac{\left|0_{S_0}0_{S_2}1_{S_3}\right\rangle \psi_1}{\sqrt{2}} + \frac{\left|1_{S_0}1_{S_2}1_{S_3}\right\rangle \psi_1}{\sqrt{2}}$

The total state is rewritten in the Bell basis for qubits `S[2,None]` and `S[3,None]`.

```
In[•]:= bs = BellState[S@{2, 3}];
        prj = Dyad[#, #] & /@ bs;
        QuissoFactor /@ (prj ** tot) // TableForm
```

Out[•]//TableForm=

$\frac{\left(\left|0_{S_2}0_{S_3}\right\rangle + \left|1_{S_2}1_{S_3}\right\rangle\right) \otimes \left(\left|0_{S_0}\right\rangle \psi_0 + \left|1_{S_0}\right\rangle \psi_1\right)}{2\sqrt{2}}$

$\frac{\left(\left|0_{S_2}1_{S_3}\right\rangle + \left|1_{S_2}0_{S_3}\right\rangle\right) \otimes \left(\left|1_{S_0}\right\rangle \psi_0 + \left|0_{S_0}\right\rangle \psi_1\right)}{2\sqrt{2}}$

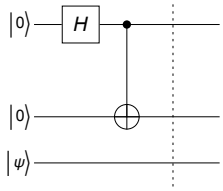
$\frac{\left(-\left|0_{S_2}1_{S_3}\right\rangle + \left|1_{S_2}0_{S_3}\right\rangle\right) \otimes \left(\left|1_{S_0}\right\rangle \psi_0 - \left|0_{S_0}\right\rangle \psi_1\right)}{2\sqrt{2}}$

$\frac{\left(\left|0_{S_2}0_{S_3}\right\rangle - \left|1_{S_2}1_{S_3}\right\rangle\right) \otimes \left(\left|0_{S_0}\right\rangle \psi_0 - \left|1_{S_0}\right\rangle \psi_1\right)}{2\sqrt{2}}$

Step 1. Alice and Bob generates an entangled pair. Bob prepares a quantum state in a separate qubit.

```
In[•]:= qc1 = QuantumCircuit[LogicalForm[Ket[], S@{1, 3}],
          ProductState[S[4] → {ψ[0], ψ[1]}, "Label" → Ket[ψ]],
          S[1, 6], CNOT[S[1], S[3]], "Separator", "Invisible" → S[2]]
```

Out[•]=



Step 2. Bob performs a Bell measurement on his qubits. This can be done by reversing the entangler circuit.

```
In[•]:= qc2a = QuantumCircuit[CNOT[S[3], S[4]], S[3, 6]];
        op = ExpressionFor[qc2a];
        bs = BellState[S@{3, 4}];
        ls = op ** bs;
        LogicalForm@Transpose@Thread[bs → ls] // TableForm
```

Out[•]//TableForm=

$\frac{\left|0_{S_3}0_{S_4}\right\rangle + \left|1_{S_3}1_{S_4}\right\rangle}{\sqrt{2}} \to \left|0_{S_3}0_{S_4}\right\rangle$

$\frac{\left|0_{S_3}1_{S_4}\right\rangle + \left|1_{S_3}0_{S_4}\right\rangle}{\sqrt{2}} \to \left|0_{S_3}1_{S_4}\right\rangle$

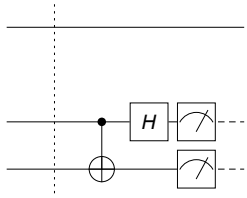
$\frac{\left|0_{S_3}1_{S_4}\right\rangle - \left|1_{S_3}0_{S_4}\right\rangle}{\sqrt{2}} \to \left|1_{S_3}1_{S_4}\right\rangle$

$\frac{\left|0_{S_3}0_{S_4}\right\rangle - \left|1_{S_3}1_{S_4}\right\rangle}{\sqrt{2}} \to \left|1_{S_3}0_{S_4}\right\rangle$

This is the corresponding quantum circuit model.

```
In[•]:= qc2 = QuantumCircuit["Separator", qc2a,
          Measurement@S@{3, 4}, "Visible" → S[1], "Invisible" → S[2]]
```

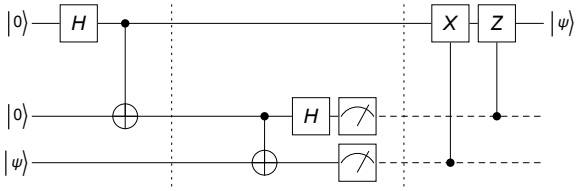
Out[•]=



Step 3. Bob sends the measurement outcome to Alice through a classical channel.
Step 4. Alice applies a proper Combining these leads to the whole protocol. These steps are simulated by a feedback control.
Combining all these steps comprises, one can transmit a quantum state to a remote party without a quantum channel.

```
In[•]:= qc = QuantumCircuit[in = ProductState[S[4] → {ψ[0], ψ[1]}, "Label" → Ket[ψ]],
          LogicalForm[Ket[], S@{1, 3}], S[1, 6], CNOT[S[1], S[3]],
          "Separator", "Spacer", CNOT[S[3], S[4]], S[3, 6],
          Measurement[S@{3, 4}], "Separator",
          ControlledU[S[4], S[1, 1]], ControlledU[S[3], S[1, 3]],
          ProductState[S[1] → {ψ[0], ψ[1]}, "Label" → Ket[ψ]], "Invisible" → S[2]]
```

Out[•]=



Check the result.

```
In[•]:= {ψ[0], ψ[1]} = Normalize@Re@RandomVector[];
        out = ExpressionFor[qc];
        in -> LogicalForm@QuissoFactor[out, S@{3, 4}]
Out[•]= (-0.956687 |0⟩ + 0.291117 |1⟩)_S4 → |0_S3 0_S4⟩ ⊗ (-0.956687 |0_S1⟩ + 0.291117 |1_S1⟩)
```

4.2 Deutch-Jozsa Algorithm & Variants

The Deutsch-Jozsa algorithm (Deutsch, 1985; Deutsch & Jozsa, 1992) is known to be the first quantum algorithm that is faster than the best classical counterpart. Although it is not useful for practical applications, it is still interesting as it illustrates some aspects of *quantum parallelism* providing a glimpse of a hint why quantum computers are faster than classical computers. It has inspired the two variants, the Bernstein-Vazirani algorithm, and Simon's algorithm.

4.2.1 Quantum Oracle

An *oracle* in computer science is a "black box" operation with certain unknown property. In decision problems such as the Deutsch-Jozsa and related problems, you are supposed to figure out the unknown property by running the oracle. Before going further, let us first examine a *quantum oracle*—a quantum mechanical implementation of an oracle.

Typically, a classical oracle is described by a binary function

$$f : \{0,1\}^m \to \{0,1\}^n \,, \tag{4.5}$$

which maps an m-bit input to an n-bit output. The function $f(x)$ may not be invertible in general.

Given a classical oracle f, a proper quantum implementation should operate on qubits and allow superposition in the input states. One naive approach is to define an operator $\hat{O}$ by $\hat{O}\,|x\rangle = |f(x)\rangle$ for any state in the logical basis of the m-qubit register. It does not work because, as mentioned above, the function $f(x)$ is not invertible in general and hence the operator $\hat{O}$ cannot be unitary.

To overcome such an issue, first extend the mapping f by adding an auxiliary register of n bits to the input and keeping the original input value so that both the input and output registers have $(m+n)$ bits. The extended mapping, $\{0,1\}^{m+n} \to \{0,1\}^{m+n}$, is then defined by the association

$$(x, y) \mapsto (x, f(x) \oplus y) \,, \tag{4.6}$$

where $x \in \{0,1\}^m$ and $y \in \{0,1\}^n$ are the bit strings of the m-bit native register and the n-bit auxiliary register, respectively. Although the function $f(x)$ itself may not be invertible, the extended mapping in (4.6) is always one-to-one regardless of the function $f(x)$—Problem 1. Due to this property, the extended mapping in (4.6) is widely used to convert a classical code to a form that is suitable for (classical) *reversible computation*.[2]

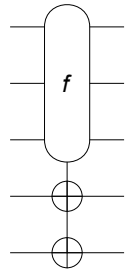
The *quantum oracle* corresponding to the classical oracle f is simply an implementation of the extended mapping (4.6) on quantum registers: It is a quantum gate operation defined by the association

$$\hat{U}_f : |x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |f(x) \oplus y\rangle \;, \tag{4.7}$$

where $|x\rangle$ and $|y\rangle$ are the logical basis states belonging to the native and auxiliary register of m and n qubits, respectively. As the extended mapping (4.6) is one-to-one and the logical basis states are orthonormal, the operator $\hat{U}_f$ is unitary—Problem 1. It is important to recall that $\hat{U}_f$ is a linear operator and can act on any

[2]For a complete reversiable computation, there is another important step required to remove the "garbage" bits.

arbitrary superposition states. In the quantum circuit model, a quantum oracle is depicted diagramatically as following



(4.8)

In this particular example, the first three qubits are from the native register and the last two qubits belong to the auxiliary register. The bit values of the auxiliary qubits are flipped conditionally depending on the value $f(x)$ as a function of the bit values x of the native qubits.

Here is a simple example, which is used in the Deutsch-Jozsa algorithm.

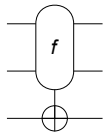
Consider a balanced function *f* as a classical oracle.

```
f[0, 0] = f[1, 1] = 0;
f[0, 1] = f[1, 0] = 1;
```

Here is the corresponding quantum oracle.

In[•]:=
```
cc = S@{1, 2};
tt = S[3];
qc = QuantumCircuit[Oracle[f, cc, tt]]
```

Out[•]=



In[•]:=
```
op = ExpressionFor[qc]
```

Out[•]=
$$\frac{1}{2}+\frac{1}{2}\,S_1^z\,S_2^z-\frac{1}{2}\,S_1^z\,S_2^z\,S_3^x+\frac{S_3^x}{2}$$

In[•]:=
```
bs = Basis@Join[cc, {tt}];
bs // LogicalForm
```

Out[•]=
$$\left\{\left|0_{S_1}0_{S_2}0_{S_3}\right\rangle,\ \left|0_{S_1}0_{S_2}1_{S_3}\right\rangle,\ \left|0_{S_1}1_{S_2}0_{S_3}\right\rangle,\ \left|0_{S_1}1_{S_2}1_{S_3}\right\rangle,\ \left|1_{S_1}0_{S_2}0_{S_3}\right\rangle,\ \left|1_{S_1}0_{S_2}1_{S_3}\right\rangle,\ \left|1_{S_1}1_{S_2}0_{S_3}\right\rangle,\ \left|1_{S_1}1_{S_2}1_{S_3}\right\rangle\right\}$$

In[•]:=
```
out = op ** bs;
out // LogicalForm
```

Out[•]=
$$\left\{\left|0_{S_1}0_{S_2}0_{S_3}\right\rangle,\ \left|0_{S_1}0_{S_2}1_{S_3}\right\rangle,\ \left|0_{S_1}1_{S_2}1_{S_3}\right\rangle,\ \left|0_{S_1}1_{S_2}0_{S_3}\right\rangle,\ \left|1_{S_1}0_{S_2}1_{S_3}\right\rangle,\ \left|1_{S_1}0_{S_2}0_{S_3}\right\rangle,\ \left|1_{S_1}1_{S_2}0_{S_3}\right\rangle,\ \left|1_{S_1}1_{S_2}1_{S_3}\right\rangle\right\}$$

```
In[•]:= Thread[bs → out] // LogicalForm // TableForm
```

Out[•]//TableForm=

$$\begin{aligned}
&\left|0_{s_1}0_{s_2}0_{s_3}\right\rangle \to \left|0_{s_1}0_{s_2}0_{s_3}\right\rangle\\
&\left|0_{s_1}0_{s_2}1_{s_3}\right\rangle \to \left|0_{s_1}0_{s_2}1_{s_3}\right\rangle\\
&\left|0_{s_1}1_{s_2}0_{s_3}\right\rangle \to \left|0_{s_1}1_{s_2}1_{s_3}\right\rangle\\
&\left|0_{s_1}1_{s_2}1_{s_3}\right\rangle \to \left|0_{s_1}1_{s_2}0_{s_3}\right\rangle\\
&\left|1_{s_1}0_{s_2}0_{s_3}\right\rangle \to \left|1_{s_1}0_{s_2}1_{s_3}\right\rangle\\
&\left|1_{s_1}0_{s_2}1_{s_3}\right\rangle \to \left|1_{s_1}0_{s_2}0_{s_3}\right\rangle\\
&\left|1_{s_1}1_{s_2}0_{s_3}\right\rangle \to \left|1_{s_1}1_{s_2}0_{s_3}\right\rangle\\
&\left|1_{s_1}1_{s_2}1_{s_3}\right\rangle \to \left|1_{s_1}1_{s_2}1_{s_3}\right\rangle
\end{aligned}$$

Let us consider another example, which is used in Simon's algorithm.

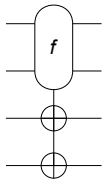
Consider a two-to-one function as a classical oracle.

```
f[0, 0] = f[1, 1] = {0, 1};
f[0, 1] = f[1, 0] = {1, 1};
```

Here is an implementation of the corresponding quantum oracle.

```
In[•]:= cc = S@{1, 2};
        tt = S@{3, 4};
        qc = QuantumCircuit[Oracle[f, cc, tt]]
```

Out[•]=




```
In[•]:= op = Elaborate@QuissoOracle[f, cc, tt]
```

Out[•]=

$$\frac{1}{2}\,S_3^x\,S_4^x+\frac{1}{2}\,S_1^z\,S_2^z\,S_4^x-\frac{1}{2}\,S_1^z\,S_2^z\,S_3^x\,S_4^x+\frac{S_4^x}{2}$$

```
In[•]:= bs = Basis@Join[cc, tt];
        bs // LogicalForm
```

Out[•]=

$$\begin{aligned}
&\{\left|0_{s_1}0_{s_2}0_{s_3}0_{s_4}\right\rangle, \left|0_{s_1}0_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|0_{s_1}0_{s_2}1_{s_3}0_{s_4}\right\rangle, \left|0_{s_1}0_{s_2}1_{s_3}1_{s_4}\right\rangle,\\
&\left|0_{s_1}1_{s_2}0_{s_3}0_{s_4}\right\rangle, \left|0_{s_1}1_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|0_{s_1}1_{s_2}1_{s_3}0_{s_4}\right\rangle, \left|0_{s_1}1_{s_2}1_{s_3}1_{s_4}\right\rangle,\\
&\left|1_{s_1}0_{s_2}0_{s_3}0_{s_4}\right\rangle, \left|1_{s_1}0_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|1_{s_1}0_{s_2}1_{s_3}0_{s_4}\right\rangle, \left|1_{s_1}0_{s_2}1_{s_3}1_{s_4}\right\rangle,\\
&\left|1_{s_1}1_{s_2}0_{s_3}0_{s_4}\right\rangle, \left|1_{s_1}1_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|1_{s_1}1_{s_2}1_{s_3}0_{s_4}\right\rangle, \left|1_{s_1}1_{s_2}1_{s_3}1_{s_4}\right\rangle\}
\end{aligned}$$

```
In[•]:= out = op ** bs;
        out // LogicalForm
```

Out[•]=

$$\begin{aligned}
&\{\left|0_{s_1}0_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|0_{s_1}0_{s_2}0_{s_3}0_{s_4}\right\rangle, \left|0_{s_1}0_{s_2}1_{s_3}1_{s_4}\right\rangle, \left|0_{s_1}0_{s_2}1_{s_3}0_{s_4}\right\rangle,\\
&\left|0_{s_1}1_{s_2}1_{s_3}1_{s_4}\right\rangle, \left|0_{s_1}1_{s_2}1_{s_3}0_{s_4}\right\rangle, \left|0_{s_1}1_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|0_{s_1}1_{s_2}0_{s_3}0_{s_4}\right\rangle,\\
&\left|1_{s_1}0_{s_2}1_{s_3}1_{s_4}\right\rangle, \left|1_{s_1}0_{s_2}1_{s_3}0_{s_4}\right\rangle, \left|1_{s_1}0_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|1_{s_1}0_{s_2}0_{s_3}0_{s_4}\right\rangle,\\
&\left|1_{s_1}1_{s_2}0_{s_3}1_{s_4}\right\rangle, \left|1_{s_1}1_{s_2}0_{s_3}0_{s_4}\right\rangle, \left|1_{s_1}1_{s_2}1_{s_3}1_{s_4}\right\rangle, \left|1_{s_1}1_{s_2}1_{s_3}0_{s_4}\right\rangle\}
\end{aligned}$$

```
In[•]:= Thread[Rule[bs, out]] // LogicalForm // TableForm
Out[•]//TableForm=
```

$$
\begin{array}{l}
|0_{s_1}0_{s_2}0_{s_3}0_{s_4}\rangle \to |0_{s_1}0_{s_2}0_{s_3}1_{s_4}\rangle \\
|0_{s_1}0_{s_2}0_{s_3}1_{s_4}\rangle \to |0_{s_1}0_{s_2}0_{s_3}0_{s_4}\rangle \\
|0_{s_1}0_{s_2}1_{s_3}0_{s_4}\rangle \to |0_{s_1}0_{s_2}1_{s_3}1_{s_4}\rangle \\
|0_{s_1}0_{s_2}1_{s_3}1_{s_4}\rangle \to |0_{s_1}0_{s_2}1_{s_3}0_{s_4}\rangle \\
|0_{s_1}1_{s_2}0_{s_3}0_{s_4}\rangle \to |0_{s_1}1_{s_2}1_{s_3}1_{s_4}\rangle \\
|0_{s_1}1_{s_2}0_{s_3}1_{s_4}\rangle \to |0_{s_1}1_{s_2}1_{s_3}0_{s_4}\rangle \\
|0_{s_1}1_{s_2}1_{s_3}0_{s_4}\rangle \to |0_{s_1}1_{s_2}0_{s_3}1_{s_4}\rangle \\
|0_{s_1}1_{s_2}1_{s_3}1_{s_4}\rangle \to |0_{s_1}1_{s_2}0_{s_3}0_{s_4}\rangle \\
|1_{s_1}0_{s_2}0_{s_3}0_{s_4}\rangle \to |1_{s_1}0_{s_2}1_{s_3}1_{s_4}\rangle \\
|1_{s_1}0_{s_2}0_{s_3}1_{s_4}\rangle \to |1_{s_1}0_{s_2}1_{s_3}0_{s_4}\rangle \\
|1_{s_1}0_{s_2}1_{s_3}0_{s_4}\rangle \to |1_{s_1}0_{s_2}0_{s_3}1_{s_4}\rangle \\
|1_{s_1}0_{s_2}1_{s_3}1_{s_4}\rangle \to |1_{s_1}0_{s_2}0_{s_3}0_{s_4}\rangle \\
|1_{s_1}1_{s_2}0_{s_3}0_{s_4}\rangle \to |1_{s_1}1_{s_2}0_{s_3}1_{s_4}\rangle \\
|1_{s_1}1_{s_2}0_{s_3}1_{s_4}\rangle \to |1_{s_1}1_{s_2}0_{s_3}0_{s_4}\rangle \\
|1_{s_1}1_{s_2}1_{s_3}0_{s_4}\rangle \to |1_{s_1}1_{s_2}1_{s_3}1_{s_4}\rangle \\
|1_{s_1}1_{s_2}1_{s_3}1_{s_4}\rangle \to |1_{s_1}1_{s_2}1_{s_3}0_{s_4}\rangle
\end{array}
$$

There are several interesting features of quantum oracle to be noticed immediately from the definition in (4.7). In Section 2.2.1, we noted that the CNOT gate makes a copy of the logical state of the control register to the target register when the latter is initially prepared in the state $|0\rangle$—see Eq. (2.35). It is exploited in many applications, expecially, for the generation of entanglement—see Eqs. (2.32) and (2.36). Quantum oracle has a similar property: However, the quantum oracle makes a copy of the image $|f(x)\rangle$ rather than the state $|x\rangle$ itself of the native register to the ancillary register,

$$|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |f(x)\rangle \ . \tag{4.9}$$

Suppose that the native quantum register in the superposition $\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle$ and the ancillary quantum register in the state $|0\rangle \equiv |0\rangle^{\otimes n}$. The quantum oracle transforms the state as

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \otimes |0\rangle \mapsto \frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \otimes |f(x)\rangle \ . \tag{4.10}$$

Just like the state in (2.32) from the CNOT gate, the state (4.10) from the quantum oracle is also entangled unless f is constant. In this case, the entanglement is controlled by the (classical) oracle f.

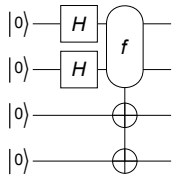
Here is demonstrate the feature of quantum oracle that makes a copy of the image f(x) of the native register to the ancillary qubit.

In this particular example, we consider a one-to-one function, but it can be arbitrary (but not constant if an entanglement is necessary).

```
f[0, 0] = {1, 1};
f[0, 1] = {1, 0};
f[1, 0] = {0, 1};
f[1, 1] = {0, 0};
```

```
In[•]:= cc = {1, 2};
tt = {3, 4};
qc = QuantumCircuit[
  LogicalForm[Ket[], Join[S@cc, S@tt]], S[cc, 6], Oracle[f, S@cc, S@tt]]
```

Out[•]=



The output state is an entangled state (unless the classical oracle `f` is constant).

```
In[•]:= out = ExpressionFor[qc];
LogicalForm[out, Join[S@cc, S@tt]]
```

$$\text{Out}[\bullet]= \frac{1}{2}\left|0_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle + \frac{1}{2}\left|0_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle + \frac{1}{2}\left|1_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle + \frac{1}{2}\left|1_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle$$

To make clearer the copies made tot the ancillary register, it may be useful to rewrite the state vector in a form that distinguishes the native and ancillary register.

```
In[•]:= QuissoFactor[out, S@cc] // LogicalForm
```

$$\text{Out}[\bullet]= \left|0_{S_1}0_{S_2}\right\rangle \otimes \left(\frac{1}{2}\left|1_{S_3}1_{S_4}\right\rangle\right) + \left|0_{S_1}1_{S_2}\right\rangle \otimes \left(\frac{1}{2}\left|1_{S_3}0_{S_4}\right\rangle\right) + \left|1_{S_1}0_{S_2}\right\rangle \otimes \left(\frac{1}{2}\left|0_{S_3}1_{S_4}\right\rangle\right) + \left|1_{S_1}1_{S_2}\right\rangle \otimes \left(\frac{1}{2}\left|0_{S_3}0_{S_4}\right\rangle\right)$$

In Section 2.2.2, we have seen that the controlled-U gate induces a phase shift on the control register—rather than on the target register—when the target register is in an eigenstate of the unitary operator. Similar method can be used to induce a phase shift conditionally on every terms that satisfy a certain condtion. For example, let $f : \{0,1\}^n \to \{0,1\}$ is a classical oracle, and suppose that we are given a state

$$|\psi\rangle = \sum_{x=0}^{2^n-1} |x\rangle\, \psi_x \tag{4.11}$$

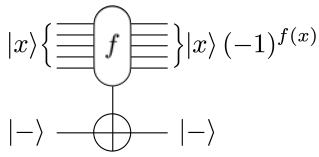
on an n-qubit register. We want to get a new state where every term $|x\rangle$ satisfying $f(x) = 1$ flips its amplitude ψ_x to $-\psi_x$

$$|\psi'\rangle = \sum_{x=0}^{2^n-1} |x\rangle\, (-1)^{f(x)} \psi_x \tag{4.12}$$

In other words, we want an *effective* quantum gate that maps the logical basis states as

$$|x\rangle \mapsto |x\rangle\, (-1)^{f(x)}\,, \quad x = 0, 1, 2, \ldots, 2^n - 1\,. \tag{4.13}$$

This can be achieved using the quantum oracle corresponding to the function f and preparing an auxiliary qubit in the state $|-\rangle := (|0\rangle - |1\rangle)/\sqrt{2}$, the eigenstate of the Pauli X operator belonging to the eigenvalue -1, as depicted in the quantum circuit model



(4.14)

Indeed, for a logical sitate $|x\rangle$ with $f(x) = 1$,

$$|x\rangle \otimes |-\rangle = \frac{|x\rangle \otimes |0\rangle - |x\rangle \otimes |1\rangle}{\sqrt{2}} \mapsto \frac{|x\rangle \otimes |1\rangle - |x\rangle \otimes |0\rangle}{\sqrt{2}} = -\,|x\rangle \otimes |-\rangle\ , \quad (4.15)$$

while nothing happens for $|x\rangle$ with $f(x) = 0$. One can find an example of more general conditional phase shift in Problem 2.

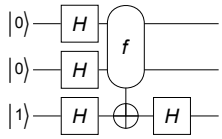
This is an interesting twist of the quantum oracle. It is useful in Grover's search algorithm.

The classical oracle mark the solutions, `{0,0}=0` and `{1,1}=1` in this particular case.

```
f[0, 0] = 1; f[0, 1] = 0; f[1, 0] = 0; f[1, 1] = 1;
```

This quantum circuit model implements the desired mapping.

In[•]:=
```
cc = {1, 2};
tt = {3};
ct = Join[cc, tt];
qc = QuantumCircuit[LogicalForm[Ket[S[tt] → 1], S[ct]],
  S[ct, 6], Oracle[f, S@cc, S@tt], S[tt, 6]]
```

Out[•]=


In[•]:=
```
out = ExpressionFor[qc];
QuissoFactor[out, S[tt]] // LogicalForm
```

Out[•]= $\left|1_{S_3}\right\rangle \otimes \left(-\frac{1}{2}\left|0_{S_1}0_{S_2}\right\rangle + \frac{1}{2}\left|0_{S_1}1_{S_2}\right\rangle + \frac{1}{2}\left|1_{S_1}0_{S_2}\right\rangle - \frac{1}{2}\left|1_{S_1}1_{S_2}\right\rangle\right)$

Check the result.

In[•]:=
```
bs = Basis[S@cc];
ff = f@@@IntegerDigits[Range[0, 2^2 - 1], 2, 2];
Thread[bs → Power[-1, ff]] // LogicalForm
```

Out[•]= $\left\{\left|0_{S_1}0_{S_2}\right\rangle \to -1,\ \left|0_{S_1}1_{S_2}\right\rangle \to 1,\ \left|1_{S_1}0_{S_2}\right\rangle \to 1,\ \left|1_{S_1}1_{S_2}\right\rangle \to -1\right\}$

In[•]:=
```
new = bs.Power[-1, ff] / 2;
new // LogicalForm
```

Out[•]= $\frac{1}{2}\left(-\left|0_{S_1}0_{S_2}\right\rangle + \left|0_{S_1}1_{S_2}\right\rangle + \left|1_{S_1}0_{S_2}\right\rangle - \left|1_{S_1}1_{S_2}\right\rangle\right)$

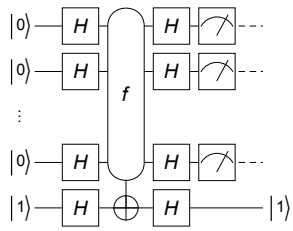
4.2.2 Deutsch-Jozsa Algorithm

Let us now discuss the Deutsch-Jozsa algorithm. Proposed first by Deutsch & Jozsa (1992), the algorithm determines whether a given function is balanced or constant. It is of little use in practice, but it is one of the first examples of a quantum algorithm that is exponentially faster than any possible classical algorithms. Further, it is so simple that it reveals clearly some aspects of *quantum parallelism* and the operational principle of quantum oracle.

The Deutsch-Jozsa problem is defined as following: Consider a classical function $f : \{0,1\}^n \to \{0,1\}$. It is a classical oracle taking n-bit inputs and returning 0 or 1. It is known that the function is either constant—either 0 or 1 for all inputs—or ballanced—0 for one half of the possible inputs and 1 for the other half. The task is to determine whether f is constant or balanced by using the oracle the least number of times.

In classical algorithms, it is known that one has to evaluate the oracle $(2^{n-1} + 1) \approx 2^n/2$ times in the worst case. That is, one has to try almost half of possible inputs before reliably determining the unknown property of the oracle. As we will see now, the Deutsch-Jozsa algorithm figures out the property from a single query to the *quantum oracle*.

The Deutsch-Jozsa algorithm is summarized in the quantum circuit model



. (4.16)

The last qubit is an ancillary qubit to induce the conditional phase shifts on the first n qubits. The Hadamard gate on it ensures that the acillary qubit is in the eigenstate $|-\rangle$ of the Pauli X operator just before the quantum oracle operates. According to Eq. (2.18), the Hadamard gates before the quantum oracle create the linear superposition of all the logical basis states of the n-qubit native register (normalization ignored)

$$|0\rangle \xrightarrow{\hat{H}^{\otimes n}} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \tag{4.17}$$

Next, as seen in Eq. (4.14), the quantum oracle induces the conditional phase shifts

$$\sum_{x=0}^{2^n-1} |x\rangle \to \sum_{x} |x\rangle \, (-1)^{f(x)} \tag{4.18}$$

Another operation of the Hadamard gates shuffles and causes addtional phase factors in accordance with (2.19), and leads to the output state to be measured

$$\sum_x |x\rangle\,(-1)^{f(x)} \xrightarrow{\hat{H}^{\otimes n}} \frac{1}{2^n}\sum_{y=0}^{2^n-1}|y\rangle\sum_{x=0}^{2^n-1}(-1)^{f(x)+x\cdot y} \tag{4.19}$$

To see the effect of the function f on the final state, suppose that f is a constant function. Then, the output state is given by

$$\frac{(-1)^{f(0)}}{2^n}\sum_{y=0}^{2^n-1}|y\rangle\sum_{x=0}^{2^n-1}(-1)^{x\cdot y} = (-1)^{f(0)}\,|0\rangle \tag{4.20}$$

Every measurement on each of the n qubits should yield zero with unit probability. To make the analysis more explicit, consider the probability to find the n-qubit register in the state $|0\rangle \equiv |0\rangle\otimes|0\rangle\otimes\cdots\otimes|0\rangle$

$$P_0 = \frac{1}{2^n}\left|\sum_{x=0}^{2^n-1}(-1)^{f(x)}\right|^2. \tag{4.21}$$

It assures that $P_0 = 1$ for a constant function f. When the function f is balanced, there are as many terms with -1 as 1, and the sum is always zero. There must be at least one qubit in the state $|1\rangle$ if the function f is balanced. Therefore, to determin whether a given function f is constant or balanced, one needs to run the quantum oracle just once, and check if the measurement outcome is 0. If the outcome is 0, then the function must be constant; and balanced otherwise.

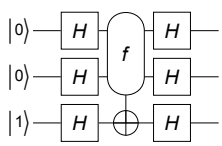
Consider a balanced function as an example.

```
f[0, 0] = f[1, 1] = 0;
f[0, 1] = f[1, 0] = 1;
```

Here is a quantum circuit model of the Deutsch-Jozsa algorithm. The final Hadamard gate on the third qubit is not necessary, but we put it here to make the output state more readable.

In[•]:=
```
cc = {1, 2};
tt = {3};
all = {1, 2, 3};
qc = QuantumCircuit[LogicalForm[Ket[S[3] → 1], S@all],
  S[all, 6], Oracle[f, S@cc, S@tt], S[all, 6]]
```

Out[•]=



In[•]:=
```
out = ExpressionFor[qc]
```

Out[•]= $\left|1_{S_1}1_{S_2}1_{S_3}\right\rangle$

4.2.3 Bernstein-Vazirani Algorithm

Suppose that we are given a binary function $f : \{0,1\}^n \to \{0,1\}$, the value of which is determined by a secrete string s of n bits

$$f : x \mapsto x \cdot s \pmod 2 . \tag{4.22}$$

The Bernstein-Vazirani problem is to find the secrete bit string s by making queries to the oracle f. Classically, one needs n queries to the function f to infer the secrete string s.

The Bernstein-Vazirani algorithm Bernstein & Vazirani (1993, 1997) is implemented in the same quantum circuit model (4.16) as the Deutsch-Jozsa algorithm. Only the analysis of the final readout is different. The first set of the Hadamard gates and the quantum oracle in (4.16) leads the n-qubit register to the state

$$\sum_{x=0}^{2^n-1} |x\rangle \, (-1)^{x \cdot s} . \tag{4.23}$$

Taking the inverse of the mapping in (2.19), one can observe that the second set of the Hadamard gates converts the above state to the final state $|s\rangle$ set by the secrete bit string. Therefore, a simple measurement of the final state in the logical basis will just reveal the secrete string.

Consider a secrete string of bits. The task is to find the secrete string.

```
string = {0, 1};
```

In the Bernstein-Vazirani algorithm, the value of the classical oracle *f* is determined by the given secrete string.

```
Clear[f];
f[x__] := Mod[{x}.string, 2]
```

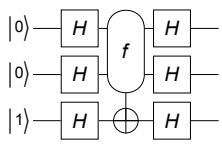
For example, `f[1,1]=0·1 + 1·1=1`.

In[•]:=
```
f[1, 1]
```
Out[•]= 1

Here is a quantum circuit model of the Bernstein-Vazirani algorithm. The final Hadamard gate on the third qubit is not necessary and we put it here to make the output state more readable.

In[•]:=
```
cc = {1, 2};
tt = {3};
all = {1, 2, 3};
qc = QuantumCircuit[LogicalForm[Ket[S[3] → 1], S@all],
  S[all, 6], Oracle[f, S@cc, S@tt], S[all, 6]]
```

Out[•]=



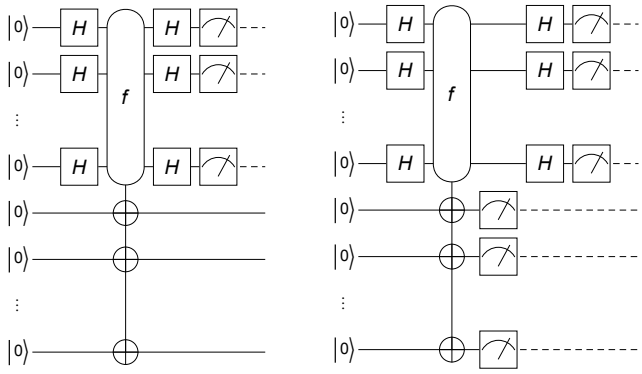


Figure 4.1: Quantum circuit models for Simon's algorithm. The measurment on the acillary register in (b) is not essential. It can be performed later than the second Hadarmard gate on the native register, or even dropped off completely as in (a).

```
In[•]:= out = ExpressionFor[qc];
        LogicalForm[out, S@cc]
Out[•]= |0_{S_1} 1_{S_2} 1_{S_3}⟩

        Here is the secrete string successfully retrieved.

In[•]:= answer = out[S@cc]
Out[•]= {0, 1}
```

4.2.4 Simon's Algorithm

Finally, let us turn to Simon's algorithm (Simon, 1997). It was the first quantum algorithm featuring an exponential speed-up over the best known classical algorithm for a specific problem. The algorithm is known to have inpsired Peter Shor's factorization algorithm. Furthermore, it has recently been shown that Simon's algorithm can be used to break the symmetric-key cryptosystems.

In Simon's problem, we are given a function $f : \{0,1\}^n \to \{0,1\}^n$ and a secrete string s of n bits. It is known that for all $x, y \in \{0,1\}^n$, $f(x) = f(y)$ if and only if $y = x \oplus s$. Note that f is either one-to-one ($s = 0$) or two-to-one ($s \neq 0$). The task is to find the secrete string s with as few queries to the function f as possible. Classically, one needs queries to $f(x)$ with up to $2^{n-1} + 1$ different inputs.[3] Unlike the Deutsch-Jozsa problem, Simon's problem is known to be hard to solve even probabilistically.

Simon's algorithm is summarized in the two slightly different quantum circuit models in Fig. 4.1. The measurement on the ancillary (second) register in the

[3]More precisely, one needs order of $\sqrt{2^n}$ queries to encounter a pair of two bit strings leading to the same result with probability greater than 1/2. The number being $\sqrt{2^n}$ rather than 2^n is related to the so-called "birthday paradox".

quantum circuit model in Fig. 4.1 (b) can be delayed until the measurement on the native (first) register, or even dropped off completely as in the quantum circuit model in Fig. 4.1 (a). That is, it is not essential, but depending on your taste, it may simplify the analsys of the algorithm. Here we will adopt the quantum circuit model in Fig. 4.1 (a). The first Hadamard gate on the native register transforms the input state of the whole system—again, see Eq. (2.18)—as

$$|0\rangle \otimes |0\rangle \xrightarrow{\hat{H}^{\otimes n}} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle \ . \tag{4.24}$$

As we noted in (4.10),the quantum oracle makes a copy of the image $|f(x)\rangle$ of the state $|x\rangle$ of the native register to the anscillary register, and leads to

$$\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle \ . \tag{4.25}$$

Finally, the second set of Hadamard gates on the native register—see Eq. (2.19)—maps the above state into

$$\sum_{y=0}^{2^n-1} |y\rangle \otimes \frac{1}{2^n} \sum_{x=0}^{2^n-1} |f(x)\rangle \, (-1)^{x\cdot y}. \tag{4.26}$$

The measurement on the native register yields an n-bit string y. The probability for a particular string y is determined by the squared norm, $P_y = \langle \psi_y | \psi_y \rangle$, of the y-dependent state $|\psi_y\rangle$ of the ancillary register

$$|\psi_y\rangle := \frac{1}{2^n} \sum_{x=0}^{2^n-1} |f(x)\rangle \, (-1)^{x\cdot y} \, . \tag{4.27}$$

Let us examine different cases: First, suppose that the secrete string $s = 0$. In this case, the function f is one-to-one, and the state $|\psi_y\rangle$ in (4.27) simply consists of terms that is a rearrangement of logical basis states

$$|\psi_y\rangle := \frac{1}{2^n} \sum_{z=0}^{2^n-1} |z\rangle \, (-1)^{y\cdot f^{-1}(z)} \, , \tag{4.28}$$

where f^{-1} is the inverse of f, and hence $P_y = 2^{-n}$ for all y. In other words, for $s = 0$, the measurement produces a random bit string y with uniform probablity. Now, suppose that $s \neq 0$. In this case, the function f is two-to-one such that $f(x) = f(x \oplus s)$. The terms in the summation in (4.27) appear in pairs giving the same state, $|f(x)\rangle = |f(x \oplus s)\rangle$. To be more explicit, let $\mathcal{B} := f(\{0,1\}^n)$ be the image of f. Then,

$$|\psi_y\rangle := \frac{1}{2^n} \sum_{z \in \mathcal{B}} |z\rangle \left\{ (-1)^{y\cdot a_z} + (-1)^{y\cdot b_z} \right\} \, , \tag{4.29}$$

where a_z and b_z are the elements in the preimage (or inverse image) of z under f, $f(a_z) = f(b_z) = z$. Since $b_z = a_z \oplus s$ and $(a_z \oplus s) \cdot y = (a_z \cdot y) \oplus (s \oplus y)$, it follows that

$$\left\{(-1)^{a_z \cdot y} + (-1)^{b_z \cdot y}\right\} = (-1)^{y \cdot a_z} \left\{1 + (-1)^{y \cdot s}\right\} . \tag{4.30}$$

If $y \cdot s$ is odd, $y \cdot s = 1 \pmod 2$, then $|\psi_y\rangle$ is a null vector, and $P_y = 0$ for such a bit string y. On the other hand, if $y \cdot s$ is even, $y \cdot s = 0 \pmod 2$, then $|\psi_y\rangle$ is a finite vector and independent of y. That is, $P_y = 2^{1-n}$ regardless of y as long as $y \cdot s$ is even.

In short, the bit string y resulting from the measurement on the native (first) register always satisfy $y \cdot s = 0 \pmod 2$ for any secrete bit string s. To find $s \equiv (s_1 s_2 \ldots s_n)_2$, one needs to run the algorithms repeated to get $(n-1)$ linearly independent bit strings $y^{(i)} \equiv (y_1^{(i)} y_2^{(i)} \ldots y_n^{(i)})_2$ $(i = 1, 2, \ldots, n-1)$, and then solve the set of equations

$$\sum_{j=1}^{n} y_j^{(i)} s_j = 0 \pmod 2. \tag{4.31}$$

It can be argued that the probability to get $n-1$ linearly independent bit strings out of n runs is slightly larger than $1/4$. Therefore, the required number of queries to the quantum oracle is order of n, exponentially smaller than the classical algorithm.

Consider again a secrete bit string.

```
string = {1, 1};
```

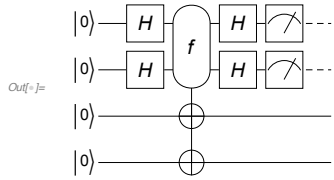
Consider a two-to-one function obeying the rule (specified in Simon's problem).

```
f[0, 0] = f[1, 1] = {0, 1};
f[0, 1] = f[1, 0] = {1, 1};
```

Here is an implementation of the corresponding quantum oracle.

```
In[•]:= cc = {1, 2};
        tt = {3, 4};
        all = Join[cc, tt];
        qc = QuantumCircuit[LogicalForm[Ket[], S@all],
          S[cc, 6], Oracle[f, S@cc, S@tt], S[cc, 6], Measurement[S@cc]]
```

Out[•]=



```
In[•]:= out = ExpressionFor[qc];
        LogicalForm[out, S@all]
        result = Readout[out, S@cc]
```

$$\text{Out}[\bullet]= \frac{\left|1_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle}{\sqrt{2}} - \frac{\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle}{\sqrt{2}}$$

```
Out[•]= {1, 1}
```

```
In[•]:= eqs = Table[out = ExpressionFor[Matrix[qc], S@all];
          result = Readout[out, S@cc], {2}]
```

```
Out[•]= {{1, 1}, {0, 0}}
```

Now let us examine a lager system. Suppose that we are given a secrete bit string.

```
string = {1, 1, 0};
```

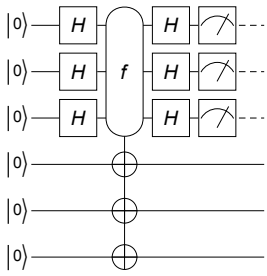
This is a function consistent with the above secrete bit string.

```
f[0, 0, 0] = f[1, 1, 0] = {0, 1, 1};
f[0, 0, 1] = f[1, 1, 1] = {1, 1, 1};
f[0, 1, 0] = f[1, 0, 0] = {1, 0, 0};
f[0, 1, 1] = f[1, 0, 1] = {0, 0, 1};
```

Here is an implementation of the corresponding quantum oracle.

```
In[•]:= cc = {1, 2, 3};
        tt = {4, 5, 6};
        all = Join[cc, tt];
        qc1 = QuantumCircuit[LogicalForm[Ket[], S@all],
           S[cc, 6], Oracle[f, S@cc, S@tt], S[cc, 6]];
        qc2 = QuantumCircuit[qc1, Measurement[S@cc]]
```

Out[•]=



This is one way to get the measurement outcome.

```
In[•]:= out = ExpressionFor[qc2];
        LogicalForm[out, S@all]
        result = Readout[out, S@cc]
```

$$\text{Out}[\bullet]= -\frac{1}{2}\left|0_{S_1}0_{S_2}1_{S_3}0_{S_4}0_{S_5}1_{S_6}\right\rangle + \frac{1}{2}\left|0_{S_1}0_{S_2}1_{S_3}0_{S_4}1_{S_5}1_{S_6}\right\rangle + \frac{1}{2}\left|0_{S_1}0_{S_2}1_{S_3}1_{S_4}0_{S_5}0_{S_6}\right\rangle - \frac{1}{2}\left|0_{S_1}0_{S_2}1_{S_3}1_{S_4}1_{S_5}1_{S_6}\right\rangle$$

```
Out[•]= {0, 0, 1}
```

To make repeated measurements, it is more efficient to first compute the state just before the measurement.

```
new = ExpressionFor[qc1];
```

Now we perform the measurement repeatedly.

```
In[•]:= data = Table[out = Measurement[new, S@cc];
          Readout[out, S@cc], {12}];
        data // TableForm
Out[•]//TableForm=
        1   1   1
        1   1   1
        0   0   1
        1   1   1
        0   0   1
        1   1   1
        0   0   1
        0   0   0
        1   1   0
        1   1   0
        0   0   1
        0   0   1
```

As two linearly independent vectors (bit strings), we choose these:

```
In[•]:= mat = {{1, 1, 0}, {0, 0, 1}}
Out[•]= {{1, 1, 0}, {0, 0, 1}}
```

Then, the linear equation, `mat.ss=0 (mod 2)`, for the Boolean variables `ss:={s1,s2,s3}` is given by the following, which agrees with the given secrete bit string.

```
In[•]:= ss = {1, 1, 0}
Out[•]= {1, 1, 0}

In[•]:= Mod[mat.ss, 2]
Out[•]= {0, 0}
```

4.3 Quantum Fourier Transform (QFT)

The quantum Fourier transform is a unitary transformation of quantum states. It is analogous to the *discrete Fourier transform* of a finite set of numbers. In the case of quantum Fourier transform, the numbers are replaced with quantum states.

The quantum Fourier transform on a quantum computer consisting of n qubits can be performed efficiently with only $\mathcal{O}(n^2)$ elementary quantum gate operations, compared to the $\mathcal{O}(n2^n)$ gates for the best known classical algorithm of discrete Fourier transform. This exponential speedup of the quantum Fourier transform algorithm compared with the classical counterpart enabled the celebrated Shor's factorization algorithm to achieve a similar efficiency. Apart from the factorization algorithm, the quantum Fourier transformation is a key part of many other quantum algorithms such as the quantum phase estimation (Section 4.4), the order-finding problem, the discrete logarithm, and most importantly the hidden subgroup problem. Such a wide range of applications of the quantum Fourier transform stem from the fact that all konwn quantum algorithms featuring exponential speedup over classical algorithms are variations of the hidden subgroup problem and, as

first realized by Kitaev (1996), the key step to solve the latter problem is the quantum Fourier transform.

4.3.1 Definition and Physical Meaning

To make the physical meaning underlying the quantum Fourier transform, in this section we will take a slightly different notation for the logical basis states, and denote them by

$$|X_x\rangle := |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \ , \quad x = 0, 1, 2, \ldots, 2^n - 1 \, , \tag{4.32}$$

where as usual, x_j are the binary digits of the index $x = (x_1 x_2 \ldots x_n)_2$. The quantum Fourier transform (QFT for short) is a unitary trasformation defined by the association

$$\hat{U}_{\rm QFT} \, |X_y\rangle := \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |X_x\rangle \, e^{ixp_y} \quad (y = 0, 1, 2, \ldots, 2^n - 1), \tag{4.33}$$

where p_y is the yth "wave number" defined by

$$p_y := \frac{2\pi y}{2^n} = 2\pi (0.y_1 y_2 \ldots y_n)_2 \, . \tag{4.34}$$

The expression in the right-hand side of (4.33) is of familiar form of discrete Fourier transform except that instead of numerical coefficients to the exponential factor e^{ixp_y} there appear state vectors. Therefore, one may think of the quantum Fourier transform as a *vector-valued* discrete Fourier transformation.

In many physical applications, however, there is a more important aspect of the quantum Fourier transform than the formal similarity to the discrete Fourier transform: It turns out to be useful and inspiring to regard the QFT as a basis change, a unitary transformtion from the logical basis

$$\{|X_x\rangle : x = 0, 1, \ldots, 2^n - 1\} \tag{4.35}$$

to the so-called "conjugate basis"

$$\{|P_y\rangle := \hat{U}_{\rm QFT} \, |X_y\rangle : y = 0, 1, \ldots, 2^n - 1\} \, . \tag{4.36}$$

The key observation is that in the "continuum" limit ($n \to \infty$), the logical basis corresponds to the eigenbasis of "position" and the conjugate basis to that of "momentum". Indeed, one can see that the following two observables

$$\hat{X} := \sum_x |X_x\rangle \, \langle X_x| \, x \, , \quad \hat{P} := \sum_y |P_y\rangle \, \langle P_y| \, p_y \tag{4.37}$$

bearing eigenvalues x and p_y, respetively, satisfy the relation

$$e^{-ia\hat{P}} \hat{X} e^{+ia\hat{P}} = \hat{X} - a \, . \tag{4.38}$$

It implies that $e^{-ia\hat{P}}$ is a translation operator and $\hat{P}$ is the generator of translation, i.e., the momentum. The QFT appears frequently in many quantum algorithms (notably the quantum phase estimation algorithm in Section 4.4), either in the explicit or disguised form. The relation between the logical and conjugate basis defined above is useful to understand the principle behind the particular algorithms.

4.3.2 Quantum Implementation

The quantum Fourier transform (QFT) algorithm is to efficiently implement the unitary operator (4.32) in terms of elementary gates.

Getting back to the definition, the quantum Fourier transform maps the quantum states by

$$\hat{U}_{\mathrm{QFT}} |X_y\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |X_x\rangle \, e^{ixp_y} \tag{4.39}$$

with the wave number

$$p_y := \frac{2\pi y}{2^n} = 2\pi (0.y_1 y_2 \ldots y_n)_2 \,. \tag{4.40}$$

As the characteristic properties of the Fourier transform of any kind are attributed to the oscillatory phase factor e^{ixp_y}, we start with elaborating it. The product xp_y in the exponent is given by

$$xp_y = \frac{2\pi xy}{2^n} = 2\pi (0.x_1 x_2 \ldots x_n)_2 y = 2\pi y \sum_{k=1}^{n} x_k 2^{-k} \,, \tag{4.41}$$

which recasts the phase factor into the form

$$e^{ixp_y} = \prod_{k=1}^{n} e^{2\pi i y x_k 2^{-k}} \,. \tag{4.42}$$

Putting it back in (4.39) and rewriting the sum over the integer values x with the sum over the bit values $x_j \in \{0, 1\}$ lead to

$$\hat{U}_{\mathrm{QFT}} |X_y\rangle = \bigotimes_{k=1}^{n} \sum_{x_k} |x_k\rangle \, e^{2\pi i x_k y/2^k} . \tag{4.43}$$

For a fixed k, the ratio $y/2^k$ is represented in the binary digits as

$$y/2^k = (y_1 y_2 \cdots y_{n-k} . y_{n-k+1} \cdots y_{n-1} y_n)_2 \,. \tag{4.44}$$

The integer part does not affect the sum because $e^{2\pi i m} = 1$ for any integer m; only the fractional part $(0.y_{n-k+1} \cdots y_{n-1} y_n)_2$ does. Depriving the phase factors $e^{ix_k p_y}$ of the integer parts, we arrive at the expression for the result of the transformation

$$\hat{U}_{\mathrm{QFT}} |X_y\rangle = \bigotimes_{k=1}^{n} \sum_{x_k} |x_k\rangle \, e^{2\pi i x_k (0.y_{n-k+1} \cdots y_{n-1} y_n)_2} . \tag{4.45}$$

Explicitly writing the sums over bit values x_j, it reads as

$$\hat{U}_{\rm QFT}\,|X_y\rangle = \left(\frac{|0\rangle + |1\rangle\, e^{2\pi i 0.y_n}}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle\, e^{2\pi i 0.y_{n-1}y_n}}{\sqrt{2}}\right) \otimes \cdots \otimes \left(\frac{|0\rangle + |1\rangle\, e^{2\pi i 0.y_1 y_2 \cdots y_n}}{\sqrt{2}}\right). \quad (4.46)$$

Interestingly, the state resulting from the quantum Frourier transform is clearly a product state. The first factor is stored in the first qubit of the quantum register, and consecutive factors in the corresponing qubits in the same order. For the later analysis, it is useful to reverse the order. It is equivalent to perform the qubit-reversing unitary transformation

$$\hat{U}_{\rm QBR}\,|y_1\rangle \otimes |y_2\rangle \otimes \cdots \otimes |y_n\rangle = |y_n\rangle \otimes \cdots \otimes |y_2\rangle \otimes |y_1\rangle\,. \quad (4.47)$$

Applying first the qubit-resersing transformation before the quantum Fourier transform, we arrive at the final expression that we analyse subsequently

$$\hat{U}_{\rm QFT}\hat{U}_{\rm QBR}\,|X_y\rangle = \left(\frac{|0\rangle + |1\rangle\, e^{2\pi i\, 0.y_1}}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle\, e^{2\pi i\, 0.y_2 y_1}}{\sqrt{2}}\right) \otimes \cdots \otimes \left(\frac{|0\rangle + |1\rangle\, e^{2\pi i\, 0.y_n \cdots y_2 y_1}}{\sqrt{2}}\right). \quad (4.48)$$

So far, we have done nothing but re-exressing the defining equation (4.39) of the quantum Fourier transform. Now we want to identify the elementary quantum logic gates that compose the transform. The product for in (4.48) is particularly useful to analyse: Let us examine the state stored on the kth qubit,

$$\frac{|0\rangle + |1\rangle\, e^{2\pi i\,(0.y_k \cdots y_2 y_1)_2}}{\sqrt{2}}. \quad (4.49)$$

Noting that

$$2\pi\,(0.y_k \ldots y_2 y_1)_2 = \sum_{j=1}^{k} y_j 2^{j-k}\pi, \quad (4.50)$$

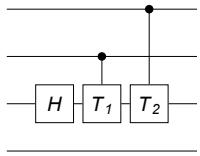
we see that the qubits $1, 2, \ldots k$ makes relative phase shifts $2^{1-k}\pi, 2^{2-k}\pi, \cdots, \pi$, respectively, on the state in (4.49). Among these, the phase shift π depends on the state $|y_k\rangle$ of the kth qubit itself, and is equivalent to the Hadamard gate. Therefore, we can regard that the state (4.49) stored on the kth qubit is resulting from the transformation

$$|y_1\rangle \otimes \cdots \otimes |y_{k-1}\rangle \otimes |y_k\rangle \mapsto |y_1\rangle \otimes \cdots \otimes |y_{k-1}\rangle \otimes \prod_{j=1}^{k-1} \hat{T}^{y_j}(2^{j-k}\pi)\hat{H}\,|y_k\rangle\,, \quad (4.51)$$

where $\hat{T}(\phi)$ denotes the relative phase shift

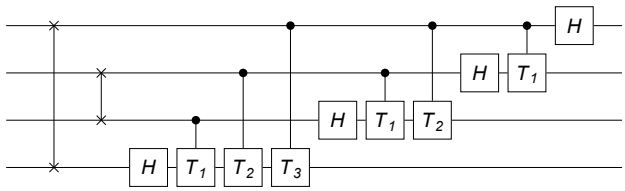
$$\hat{T}(\phi) = |0\rangle\langle 0| + |1\rangle\langle 1| e^{i\phi} \doteq \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \tag{4.52}$$

As the actual operation of $\hat{T}^{y_j}(2^{j-k}\pi)$ is determined by the logical state $|y_j\rangle$ of the jth qubit, it is a controlled unitary gate. For example, the state store on the third qubit is represented in the quantum circuit model as



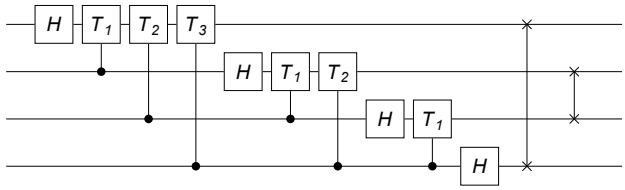
(4.53)

where we have used the short-hand notation $\hat{T}_l := \hat{T}(\pi/2^l)$. Overall, the quantum Fourier transformation can be represented in the quantum circuit model as



, (4.54)

where we have implemented the qubit-reversing transformation $\hat{U}_{\mathrm{QBT}}$ by applying the SWAP gates on the pairs of qubits from the first and second half of the quantum register. In the above quantum circuit model, we have applied $\hat{U}_{\mathrm{QBR}}$ at the begining. One can apply it at the end as well: As $\hat{U}_{\mathrm{QBR}}$ corresponds to simply reversing the order of qubits, the quantum circuit model in (4.54) is identical to the following model



. (4.55)

This quantum circuit model for the quantum Fourier transformation is found more commonly in the literature.

Here we simulate the quantum Fourier transform on a quantum register of 4 qubits.

```
$n = 4;
```

Here defined are the controlled-phase gates.

```
CP[j_, j_] := S[j, 6]
CP[j_, k_] := ControlledU[S[j],
    Phase[Pi Power[2, k - j], S[k], "Label" → Subscript["T", j - k]]
  ] /; j > k
CP[j_, k_] := ControlledU[S[j],
    Phase[Pi Power[2, j - k], S[k], "Label" → Subscript["T", k - j]]
  ] /; j < k
```

We denote by the label T_k the relative phase shift by $\pi/2^k$: $T_k := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^k} \end{pmatrix}$.

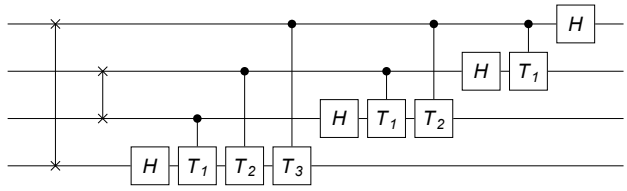
This is just a short-hand function.

```
SW[j_] := SWAP[S[j], S[$n - j + 1]]
SW[All] := Table[SW[j], {j, 1, $n / 2}]
```

This is the standard implementation, which can be seen commonly in textbooks.

```
In[•]:= gates = Flatten@Table[CP[j, k], {k, $n, 1, -1}, {j, k, 1, -1}];
        qc1 = QuantumCircuit[Sequence @@ SW[All], Sequence @@ gates]
```

Out[•]=



To verify the above quantum circuit model, we compare its matrix representation with the matrix for the discrete Fourier transform.

```
In[•]:= mat = Matrix[qc1] // TrigToExp;
        new = FourierMatrix[Power[2, $n]];
        new - mat // Simplify // MatrixForm
```

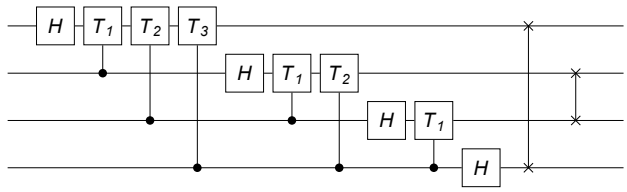
Out[•]//MatrixForm=

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

One can also reverse the order of qubits later to get the following quantum circuit model for the quantum Fourier transform.

```
In[•]:= gates = Flatten@Table[CP[j, k], {k, 1, $n}, {j, k, $n}];
        qc2 = QuantumCircuit[Sequence @@ gates, Sequence @@ SW[All]]
```

Out[•]=



Again, verify it by comparing the matrix representation with the matrix for the discrete Fourier transform.

```
In[•]:= mat = Matrix[qc2] // TrigToExp;
        new = FourierMatrix[Power[2, $n]];
        new - mat // Simplify // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4.3.3 Semiclassical Implementation

The semiclassical implementations is not only just interesting from a physics point of view but also extremely interesting with the current level of technology (Griffiths & Niu, 1996). As it does not require any entanglement, the most fragile quantum information resource and hence difficult to maintain, many experimenters are using this implementation whenever the QFT and/or the quantum phase estimation (see Section 4.4) is in needs (Chiaverini, 2005; Higgins *et al.*, 2007).

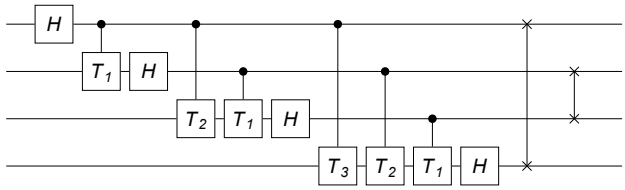
The line of arguments (Griffiths & Niu, 1996) in the original work to derive the semiclassical implementation of the quantum Fourier transformation is interesting in its own right, and can offer another derivation of the quantum Fourier transform algorithm. Here we will not repeat the original arguments. Instead, we make use the relation between the quantum Fourier transform and its inverse. The inverse quantum Fourier tranform is given by

$$\hat{U}_{\mathrm{QFT}}^{\dagger} |X_y\rangle = \frac{1}{2^{n/2}} \sum_x |X_x\rangle \, e^{-ixp_y} \,, \tag{4.56}$$

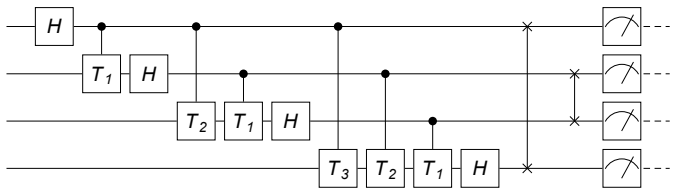
which follows directly from the orthgonality relation

$$2^n \sum_y e^{i(x-x')p_y} = \delta_{xx'} \,. \tag{4.57}$$

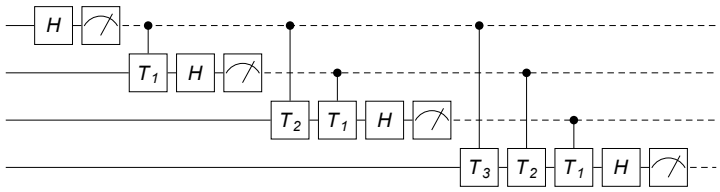
The expression (4.56) suggests that the inverse quantum Fourier transform is just another form of quantum Fourier transform—this is indeed a characteristic property common to any kind of Fourier transform. It implies that one can recover the origianl quantum Fourier transform from the inverse quantum Fourier transform, and vice versa, in several ways. For example, note that $\hat{U}^\dagger_{\rm QFT}\,|X_y\rangle$ is an element of the conjugate basis because $e^{-ixp_y} = e^{ix(2\pi - p_y)}$ for any x and y. In fact, $\hat{U}^\dagger_{\rm QFT}\,|X_y\rangle = |P_{2^n-y}\rangle$. More useful in the context of the semiclassical implementation of the quantum Fourier transform is to note that the inverse quantum Fourier transform $\hat{U}^\dagger_{\rm QFT}$ is nothign but the quantum Fourier transform with the phase factor e^{ixp_y} replaced by e^{-ixp_y}. It corresonds to replacing the relative phase shifts $\hat{T}(\phi)$ in (4.54) by $T^\dagger(\phi)$. Conversely, if we start with the quantum circuit model in (4.54), take its Hermitian conjuate (i.e., inverse), and replace $\hat{T}^\dagger(\phi) \equiv \hat{T}(-\phi)$ with $\hat{T}(\phi)$, then we should recover the original quantum Fouriert transform. Through this procedure, we get the following quantum circuit model



for the quantum Fourier transform. Compared with the quantum circuit model in (4.54), the above quantum circuit model has the Hadamard gate on a fixed qubit coming before the qubit "controls" the operation of the relative phase shift $\hat{T}(\phi)$ on other qubits. This difference brings a significant consequence. To see it, we put the measurements on the qubits explicitly,



According to the *pinciple of deffered measurement*, the measurement can be performed before the controlled unitary gates



This form is exactly the circuit model derived in Griffiths & Niu (1996). The important point is that after the measurement, the controlled unitary gates become just classical feedback controls, and require no entanglement.

Here we simulate the semiclassical implementation of the quantum Fourier transform. Again, we consider a quantum register of four qubits.

```
$n = 4;
SS = S[Range[$n], None];
```

Here defined are the controlled-phase gates.

```
CP[j_, j_] := S[j, 6]
CP[j_, k_] := ControlledU[S[j],
    Phase[Pi Power[2, k - j], S[k], "Label" → Subscript["T", j - k]]
  ] /; j > k
CP[j_, k_] := ControlledU[S[j],
    Phase[Pi Power[2, j - k], S[k], "Label" → Subscript["T", k - j]]
  ] /; j < k
```

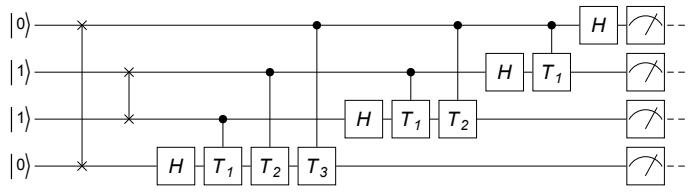
In[•]:=
```
in = Ket @@ Thread[SS -> RandomChoice[{0, 1}, $n]];
LogicalForm[in, SS]
```

Out[•]= $|0_{S_1} 1_{S_2} 1_{S_3} 0_{S_4}\rangle$

This is a quantum circuit model for the quantum Fourier transformation.

In[•]:=
```
gates = Flatten@Table[CP[j, k], {k, $n, 1, -1}, {j, k, 1, -1}];
swaps = Table[SWAP[S[k], S[$n - k + 1]], {k, 1, $n / 2}];
qc1 = QuantumCircuit[LogicalForm[in, SS],
  Sequence @@ swaps, Sequence @@ gates, Measurement[SS]]
```

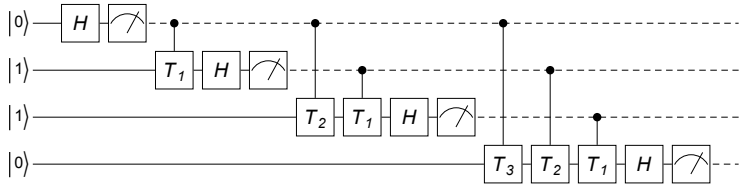
Out[•]=



This is the semiclassical implementation of the quantum Fourier transform.

```
In[•]:= gates = Table[CP[j, k], {k, 1, $n}, {j, 1, k}];
        gates = Flatten@MapIndexed[Append[#1, Measurement@S[#2]] &, gates];
        swaps = Table[SWAP[S[k], S[$n - k + 1]], {k, 1, $n / 2}];
        qc2 = QuantumCircuit[LogicalForm[in, SS], Sequence @@ gates]
```

Out[•]=

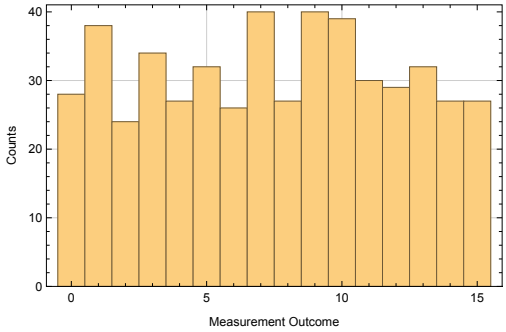


To verify the equivalence of the two quantum circuit models, we compare the probability distributions.

```
In[•]:= Timing[data1 = Table[out = ExpressionFor[qc1];
            FromDigits[Readout[out, SS], 2], {500}];]
Out[•]= {63.5074, Null}
```

```
In[•]:= Histogram[data1, FrameLabel → {"Measurement Outcome", "Counts"}]
```

Out[•]=

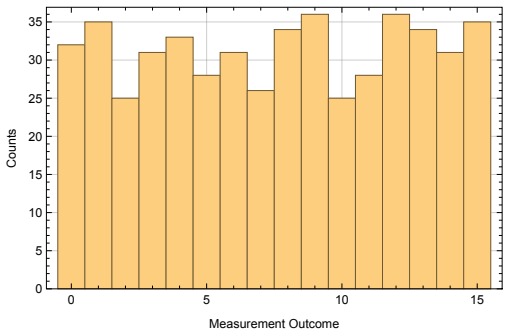


Note that the bit strings of the measurement outcomes from the semiclassical model must be reversed .

```
In[•]:= Timing[data2 = Table[out = ExpressionFor[qc2];
            FromDigits[Reverse@Readout[out, SS], 2], {500}];]
Out[•]= {18.8853, Null}
```

```
In[•]:= Histogram[data2, FrameLabel → {"Measurement Outcome", "Counts"}]
```

Out[•]=



The discrepancy between the two distributions is due to the finite sample size.

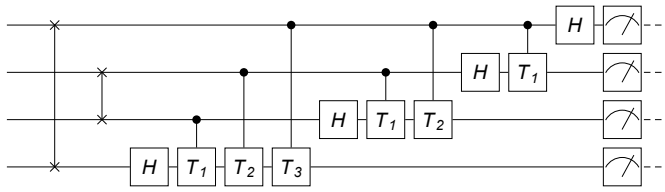
Another way to test the equivalence of the fully quantum and semiclassical implementation of the quantum Fourier transformation is to take as the input state one that ends up in a product state.

```
In[•]:= vecs = FourierMatrix[Power[2, $n], FourierParameters → {0, -1}].Basis[SS];
        in = RandomChoice[vecs];
        in // LogicalForm
```

$$\text{Out}[\bullet]= \frac{1}{4}\left|0_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{\frac{3i\pi}{8}}\left|0_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle + \frac{1}{4}e^{\frac{3i\pi}{4}}\left|0_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{-\frac{7i\pi}{8}}\left|0_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle -$$
$$\frac{1}{4}i\left|0_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{-\frac{i\pi}{8}}\left|0_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle + \frac{1}{4}e^{\frac{i\pi}{4}}\left|0_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{\frac{5i\pi}{8}}\left|0_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle -$$
$$\frac{1}{4}\left|1_{S_1}0_{S_2}0_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{-\frac{5i\pi}{8}}\left|1_{S_1}0_{S_2}0_{S_3}1_{S_4}\right\rangle + \frac{1}{4}e^{-\frac{i\pi}{4}}\left|1_{S_1}0_{S_2}1_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{\frac{i\pi}{8}}\left|1_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle +$$
$$\frac{1}{4}i\left|1_{S_1}1_{S_2}0_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{\frac{7i\pi}{8}}\left|1_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle + \frac{1}{4}e^{-\frac{3i\pi}{4}}\left|1_{S_1}1_{S_2}1_{S_3}0_{S_4}\right\rangle + \frac{1}{4}e^{-\frac{3i\pi}{8}}\left|1_{S_1}1_{S_2}1_{S_3}1_{S_4}\right\rangle$$

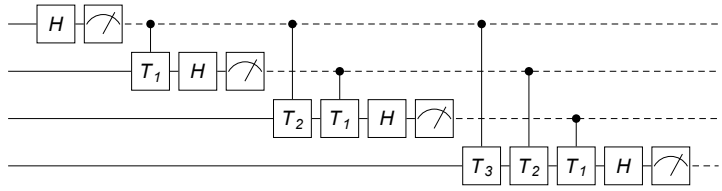
```
In[•]:= gates = Flatten@Table[CP[j, k], {k, $n, 1, -1}, {j, k, 1, -1}];
        swaps = Table[SWAP[S[k], S[$n - k + 1]], {k, 1, $n / 2}];
        qc1 = QuantumCircuit[{in, "Label" → None},
          Sequence @@ swaps, Sequence @@ gates, Measurement[SS]]
```

Out[•]=




```
In[•]:= gates = Table[CP[j, k], {k, 1, $n}, {j, 1, k}];
        gates = Flatten@MapIndexed[Append[#1, Measurement@S[#2]] &, gates];
        swaps = Table[SWAP[S[k], S[$n - k + 1]], {k, 1, $n / 2}];
        qc2 = QuantumCircuit[{in, "Label" → None}, Sequence @@ gates]
```

Out[•]=




```
In[•]:= out1 = ExpressionFor[qc1];
        LogicalForm[out1, SS]
```

$$\text{Out}[\bullet]= \left|1_{S_1}1_{S_2}0_{S_3}1_{S_4}\right\rangle$$

Again, recall that the order of qubits are reversed in the semiclassical implementation.

```
In[•]:= out2 = ExpressionFor[N@qc2] // Chop;
        LogicalForm[out2, SS]
```

$$\text{Out}[\bullet]= 1.\left|1_{S_1}0_{S_2}1_{S_3}1_{S_4}\right\rangle$$

4.4 Quantum Phase Estimation (QPE)

Quantum phase estimation is a procedure to estimate the phase of the unknown eigenvalue of an eigenstate of a unitary operator. Like the quantum Fourier transformation, the quantum algorithm for quantum phase estimation is one of the key elements of many quantum algorithms including Shore's factorization algorithm. In fact, the quantum Fourier transform is a part of the quantum phase estimation algorithm, and almost always appear in that way rather than independently. In this sense, quantum phase estimation is the key subroutine in most quantum algiorithms. Indeed, all quntum algorithms known so far can be regarded as quantum phase estimation in one form or another (Cleve *et al.*, 1998).[4]

4.4.1 Definition

Let $\hat{U}$ be a unitary operator on the Hilbert space associated with a register consisting of n qubits. Suppose that the quantum register is known to be prepared in an eigenstate state $|\phi\rangle$ of $\hat{U}$. Its eigenvalue $e^{-i\phi}$ is desired but *unkown*. The quantum phase estimation (QPE) procedure is to get the phase value ϕ (and hence the eigenvalue).

Note that the quantum phase estimation is directly related to the measurement. Recall that finding the eigenvalue $e^{-i\phi}$ reveals the corresponding eigenstate $|\phi\rangle$. As an unitary operator can be written as $\hat{U} = e^{-i\hat{Q}}$ for some observable $\hat{Q}$, the estimation of phase corresponds to the measurement of the quantity $\hat{Q}$. In this sense, one can regard the quantum phase estimation procedure as a realization of quantum measurement on quantum computers. We will discuss this aspect in more detail in Section 4.4.3.

To get a general idea behind the quantum phase esitmation procedure, take an ancillary register of m qubits, and prepare it in the superposition $\propto \sum_x |x\rangle$ of all elements in the logical basis. The superposition state can be generated by applying the Hadamard gate—see Eq. (2.18). The state vector of the total system becomes

$$\left(\hat{H}^{\otimes n}|0\rangle\right) \otimes |\phi\rangle = \frac{1}{2^{m/2}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |\phi\rangle \tag{4.58}$$

Next, perform a unitary transformation on the total system defined by

$$\hat{U}_{\mathrm{QPE}} : |x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes \left(\hat{U}^x |y\rangle\right) , \tag{4.59}$$

where $|x\rangle$ $(x = 0, 1, \cdots, 2^m - 1)$ belongs to the ancillary register and $|y\rangle$ $(y = 0, 1, \ldots, 2^n - 1)$ to the native register. The transformation is a controlled unitary gate, performing the transformation $\hat{U}$ repeatedly depending on the value of y on

[4] Mathematically speaking, they belong to the class called the hidden subgroup problem.

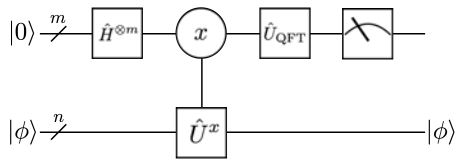
the native register. Upon the controlled unitary transformation, the state vector becomes

$$\frac{1}{2^{m/2}}\sum_x |x\rangle \otimes \left(\hat{U}^x |\phi\rangle\right) = \frac{1}{2^{m/2}}\sum_x |x\rangle \left(|\phi\rangle\, e^{-ix\phi}\right) = \left(\frac{1}{2^{m/2}}\sum_{x=0}^{2^m-1} |x\rangle\, e^{-ix\phi}\right) \otimes |\phi\rangle \quad (4.60)$$

Note that the original register still remains in the same state $|\phi\rangle$, whereas the ancillary register has picked up the relative phase shift proportional to x, $e^{-ix\phi}$. The state in (4.60) stored on the ancillary register is thus formally the same as the (inverse) quantum Fourier transform—see (4.33) and (4.56). For the moment, let us assume that $\phi/2\pi$ ($0 \le \phi/2\pi < 1$) takes one of the discrete values

$$\frac{0}{2^m}, \frac{1}{2^m}, \frac{2}{2^m}, \cdots, \frac{2^m-1}{2^m}. \quad (4.61)$$

In this case, performing the quantum Fourier trasnform $\hat{U}_{\rm QFT}$ on the ancillary register will bring it to the logical basis state $|y\rangle$ with $y = 2^m\phi/2\pi$, and the measurement on the logical basis will reveal the value ϕ. The procedure described above is summarized in the following diagram.



(4.62)

The normalize phase $\phi/2\pi$ is continuous parameter, and in general, has values other than the discrete values in (4.61). It means that the number m of qubits in the ancillary register is not sufficient to accurately estimate the phase ϕ.

4.4.2 Implementation

Let us now discuss the actual implementation of the individual steps. The preparation of the ancillary register in the overal superposition state is achieved by using the Hadamard gate (see Section 2.1.2)

$$|0\rangle^{\otimes m} \to \hat{H}^{\otimes m}|0\rangle^{\otimes m} = \frac{1}{2^{m/2}}\sum_{x=0}^{2^m-1} |x\rangle\ . \quad (4.63)$$

Explicitly writing, the controlled-$\hat{U}^x$ transformation in Eq. (4.59) reads as

$$|x\rangle \otimes \hat{U}^x |y\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_m\rangle \otimes \hat{U}^{x_1 2^{m-1}} \hat{U}^{x_2 2^{m-2}} \cdots \hat{U}^{x_m 2^0} |y\rangle\ . \quad (4.64)$$

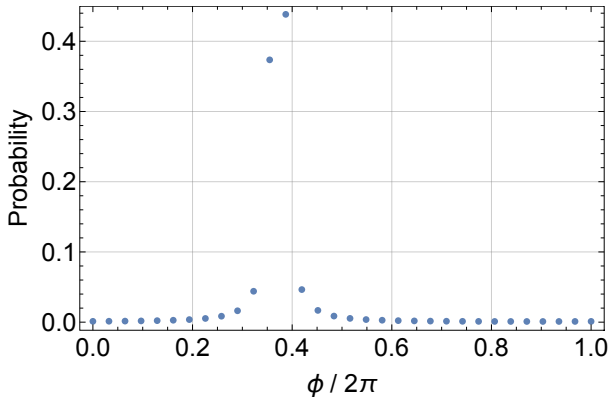
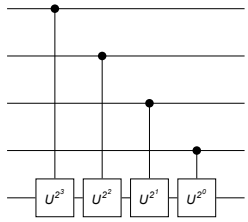


Figure 4.2: Probability to estimate the value of phase using $m = 5$ ancillary qubits when the true phase value is $\phi = 18\pi/25$.

It is a product of controlled-$\hat{U}^{2^j}$ gates depending on the value x_j of the jth ancillary qubit. It is thus implemented by the following quantum circuit model



(4.65)

Here it is noted that the efficient implementation of quantum phase estimation requires an efficient implementation of the controlled-U^{2^j} gates. One of the most common unitary transformations that allows for efficient implementation is the *modular multiplication*, which is used in the order-finding problem.

The controlled-U^x gate brings the state of the total system to

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \otimes |\phi\rangle \to \left(\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \, e^{-i\phi x} \right) \otimes |\phi\rangle \tag{4.66}$$

The final quantum Fourier transformation on the ancillary register produces

$$\frac{1}{2^{m/2}} \sum_{x} |x\rangle \, e^{-ix\phi} \to \frac{1}{2^m} \sum_{x,y} |y\rangle \, e^{i(2\pi y/2^m - \phi)x} \tag{4.67}$$

Upon the measurement in the logical basis, the probability to get the outcome y is given by

$$P_y = \left| \frac{1}{2^m} \sum_{x=0}^{2^m-1} e^{i2\pi(y-p)x/2^m} \right|^2 , \tag{4.68}$$

where $p := 2^m \frac{\phi}{2\pi}$. For reasonably large m, the probability P_y is sharply peaked around p (see Fig. 4.2): It is precisely 1 when p is one of the integers $0, 1, \ldots, 2^m - 1$. In general, $p = (p_1 p_2 \ldots p_m . p_{m+1} p_{m+2} \ldots)_2$, has a fractional part

$$\delta := \sum_{j=1}^{\infty} p_{m+j} 2^{-j} \,. \tag{4.69}$$

The fractional part δ cannot be stored in the m-qubit probe register, and results in the error of the phase estimation procedure.

We take an ancillary register consisting of three qubits.

```
m = 4;
ancilla = Range[m];
sys = m + 1;
```

We consider a rotation around the z-axis for the unitary operator. In this case, the logical basis consists of the eigenstates of the unitary operator.

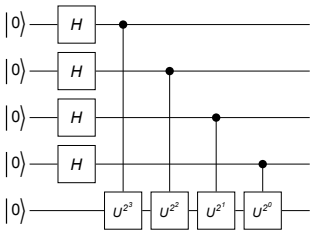
```
p = 18 / 25;
U = Rotation[p 2 Pi, S[sys, 3]];
```

Here is the controlled – $U^{2^{m-j}}$ operator to be used in the phase estimation algorithm.

```
ctrlU[j_] := ControlledU[S[j], MultiplyPower[U, Power[2, m - j]],
  "Label" → Superscript["U", Superscript["2", ToString[m - j]]],
  "LabelSize" → 0.65]
```

In[•]:=
```
qc = QuantumCircuit[LogicalForm[Ket[], S@ancilla], LogicalForm[Ket[], S@sys],
  {S[ancilla, 6], "LabelSize" → 0.8}, ctrlU[1], ctrlU[2], ctrlU[3], ctrlU[4]]
```

Out[•]=



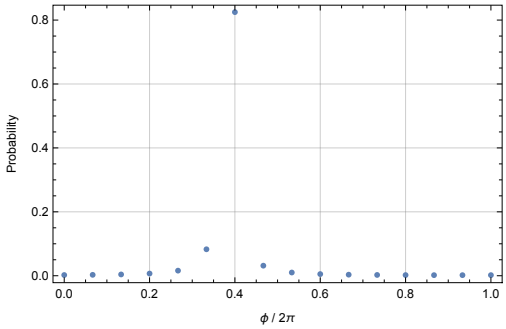
As the QFT step is clear, we do not simulate the step here and just process the data using the built-in function `Fourier`.

In[•]:=
```
vec = ExpressionFor[qc];
out = Fourier@Matrix[vec, S@ancilla]
```

Out[•]=
```
{0.0146117 - 0.0449701 i, 0.00625173 - 0.0528206 i,
 -0.00498774 - 0.0633752 i, -0.0225159 - 0.0798352 i,
 -0.0573411 - 0.112538 i, -0.17816 - 0.225994 i, 0.690635 + 0.589858 i,
 0.154844 + 0.0867168 i, 0.0955664 + 0.0310514 i, 0.0715134 + 0.00846417 i,
 0.057667 - 0.00453849 i, 0.0480647 - 0.0135556 i, 0.0405163 - 0.0206441 i,
 0.0339768 - 0.0267851 i, 0.027817 - 0.0325695 i, 0.0215406 - 0.0384635 i}
```

```
In[•]:= Prob = Abs[out]^2
Out[•]= {0.00223581, 0.0028291, 0.00404129, 0.00688062, 0.0159529,
  0.0828144, 0.824909, 0.0314965, 0.0100971, 0.00518581, 0.00334608,
  0.00249397, 0.00206775, 0.00187186, 0.00183456, 0.00194343}

In[•]:= ListPlot[Prob, PlotRange → All,
  DataRange → {0, 1}, FrameLabel → {"ϕ / 2π", "Probability"},
  FrameStyle → Automatic, ImageSize → Automatic]
```

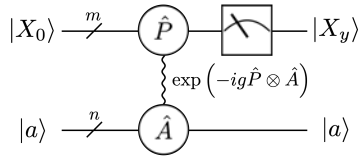


4.4.3 Simulation of von Neumann Measurement

The von Neumann scheme of measurement (see Section 1.3.1) is an idealistic mechanism to implement a projection measurement. As already mentioned, the quantum phase estimation procedure is in spirit equivalent to the von Neumann scheme of measurement. In fact, quantum phase estimation was put forward to simulate the von Neumann scheme on quantum computers (Kitaev, 1996, 1997). The equivalence of the von Neumann scheme of measurement and the quantum phase estimation procedure have also inspired the method of direct tomography of wave functions and its variants (Lundeen *et al.*, 2011; Vallone & Dequal, 2016). Therefore, it will be interesting for the scientific as well as heuristic purposes to examine the equivalence in more detail.

We consider a "system" consisting of n qubits, and pick a "probe" of m qubits. To maintain the physical nature of the von Neumann scheme, for the probe let us consider two bases, $\{|X_x\rangle\}$ and $\{|P_x\rangle\}$ ($x = 0, 1, 2, \ldots, 2^m - 1$), that are conjugate to each other and related by the quantum Fourier transform $|P_x\rangle = \hat{U}_{\mathrm{QFT}} |X_x\rangle$. We call them the "position" and "momentum" basis, respectively (see Section 4.3.1). To make the argument simpler, we assume the system is in a definite yet unkown eigenstate $|a\rangle$ of $\hat{A}$. We want to find the eigenvalue a by a measurement procedure.

Following the von Neumann scheme, we prepare the probe in the state $|X_0\rangle$, which refers to a definite position. We let the system and probe interact with each other by coupling the "momentum" operator $\hat{P}$ of the probe to the observable $\hat{A}$ of the system with strength g. After the interacion for a finite duration of time, the probe is measured in the basis $|X_y\rangle$. This is illustrated in the following schematic diagram



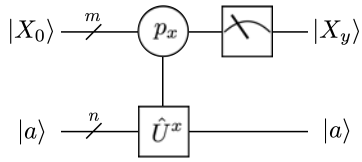
We rewrite the interaction unitary operator as [to be compared with (1.44)]

$$\hat{U}_{\text{int}} = \exp\left(-ig\hat{P}\otimes\hat{A}\right) = \sum_x |P_x\rangle\langle P_x| \otimes e^{-igp_x\hat{A}} = \sum_x |P_x\rangle\langle P_x| \otimes \hat{U}^x \tag{4.70}$$

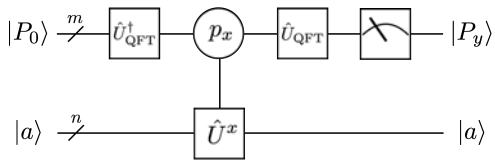
where the operator $\hat{U}$ acting on the system has been defined by

$$\hat{U} = \exp\left[-i\left(\frac{2\pi g}{2^n}\right)\hat{A}\right]. \tag{4.71}$$

In this new interpretation, the operator $\hat{U}$ is repeatedly opperated x times on the system depending on the wave number p_x (eigenvalue of $\hat{P}$) in the probe. This interpretation is seeminly opposite to the original idea of the von Neumann scheme, where the translation operator $\hat{T}_a$ is operated on the probe depending on the eigenvalue a of $\hat{A}$ in the system. This reinterpretation is depicted schematically in the following diagram [to be compared with the diagram in (1.45)]

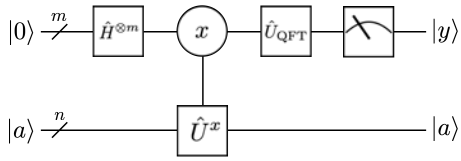


Note that the probe register controls the operation of $\hat{U}$ on the system in the "momentum" basis $\{|P_y\rangle\}$, and hence the momentum basis is the computational basis (logical basis). On the other hand, the von Neumann scheme requires the measurement to be performed in the "position" basis $\{|X_y\rangle\}$. As discussed in Section 2.5, any measurment in a basis different from the logical basis can be achieved by applying a proper unitary transformation corresponding to the basis change. In this case, it is the quantum Fourier transform. Similarly, one can obtain the input state $|X_0\rangle$ by acting $\hat{U}^\dagger_{\text{QFT}}$ on $|P_0\rangle$. These changes lead to the schematic diagram



Finally, with all bases changed to the logical basis $\{|P_x\rangle\}$, we can drop the addition labels 'P' and 'p' from the states and eigenvalues. Recall also that the action of the quantum Fourier transform on $|P_0\rangle$ can be replaced by the Hadamard gates,

$\hat{U}_{\mathrm{QFT}}^{\dagger}|P_0\rangle = \hat{H}^{\otimes m}|P_0\rangle$ (see Problem 4). Therefore, the von Neumann measurement can be depicted as the following digram, which is identical to the diagram (4.62) corresponding to the quantum phase estimation procedure.



4.5 Applications

Before concluding the chapter, let us examine two example problems where the quantum Fourier transform and the quantum phase estimation can be applied. Fourier transform is particularly useful for periodic effects. It is thus natural to use the quantum Fourier transform to find the unknown period of a given function. The order-finding problem is a specific example, where the function is the *modular exponentiation*. As mentioned earlier, however, the quantum Fourier transform cannot be used independently although it is the key part. One needs a procedure to induce the relative phase shifts, which can be extracted with the quantum Fourier transform. The procedure is the quantum phase esitmation. Mathematically, the period-finding and order-finding problem belong to a wider class of problem known as the *hidden subgroup problem*.

4.5.1 The Period-Finding Algorithm

Let $f : \{0,1\}^m \to \{0,1\}^n$ be a periodic function, and

$$f(x+\lambda) = f(x) \tag{4.72}$$

for all x and fixed λ. The period-finding problem is to find the unkown (primary) period λ with the least queries to the function f. For simplicity, we assume that λ is a divider of 2^m. Here we also assume that the quantum oracle $\hat{U}_f$ corresponding to the function f is efficiently implemented for the mapping

$$\hat{U}_f : |x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y \oplus f(x)\rangle\,. \tag{4.73}$$

Given the sequence of function values $f(0), f(1), \ldots, f(\lambda-1)$, one can perform the discrete Fourier transform of the sequence to get

$$F_k := \frac{1}{\sqrt{\lambda}} \sum_{k=0}^{\lambda-1} f(x) e^{-2\pi i k x/\lambda} \tag{4.74}$$

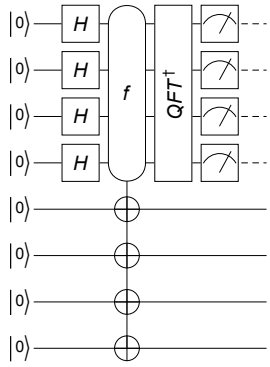


Figure 4.3: A quantum circuit model for the period-finding algorithm.

Analogously, it is also possible to define new states[5]

$$|F_k\rangle = \frac{1}{\sqrt{\lambda}} \sum_{k=0}^{\lambda-1} |f(x)\rangle \, e^{+2\pi i k x/\lambda} \tag{4.75}$$

for $x = 0, 1, 2, \ldots, \lambda - 1$, using the discrete Fourier tansform of the sequence of states $|f(x)\rangle$ with $x = 0, 1, 2, \ldots, \lambda - 1$. It should be noted that in general, the new states $|F_k\rangle$ may not be linearly independent with each other because $|f(x)\rangle = |f(y)\rangle$ for some $0 \leq x, y < \lambda$.[6]

The period-finding algorithm is summarized in the quantum circuit model in Fig. 4.3. We prepare the control register in the state $2^{-m/2} \sum_x |x\rangle$. The quantum oracle (4.73) then makes the transformation

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \otimes |0\rangle \rightarrow \frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \otimes |f(x)\rangle \tag{4.76}$$

We now replace the states $|f(x)\rangle$ with $|F_k\rangle$ in (4.75) using the inverse relation

$$|f(x)\rangle = \frac{1}{\sqrt{\lambda}} \sum_{k=0}^{\lambda-1} |F_k\rangle \, e^{-2\pi i k x/\lambda} \,, \tag{4.77}$$

which leads to the expression for the state of the total system

$$\frac{1}{\sqrt{\lambda}} \sum_{k=0}^{\lambda-1} \left(\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle \, e^{-2\pi i k x/\lambda} \right) \otimes |F_k\rangle \tag{4.78}$$

[5]Here we take the opposite sign in the exponent of the phase factor $e^{2\pi i k x/\lambda}$ to be consistent with the quantum Fourier transform in (4.33).

[6]In this respect, the period-finding problem in the presented form is slightly different from the hidden subgroup problem. In the hidden subgroup problem, the function $f : \mathcal{G} \rightarrow \mathcal{S}$ from a group $\mathcal{G}$ to a set $\mathcal{S}$ is assumed to separate the cosets. In other words, $f(x) = f(y)$ *if and only if* x and y belong to the same coset of the given subgroup $\mathcal{H} \subset \mathcal{G}$.

It reveals that the quantum oracle has induced the relative phase shifts in the control register that is proportional to the logical values x. The period-finding problem can thus be regarded to fall in the category of quantum phase estimation. As usual in quantum phase estimation, the relative phase shifts can be extract by applying the quantum Fourier transform, which in this case leads to

$$\frac{1}{\sqrt{\lambda}} \sum_{k=0}^{\lambda-1} |2^m k/\lambda\rangle \otimes |F_k\rangle \,. \tag{4.79}$$

Finally, the measurement on the control register produces the outcome $y = 2^m k/\lambda$ ($k = 0, 1, 2, \ldots, \lambda - 1$) randomly, from which one can find the period λ. Here note that the probability P_k for the particular measurement output $2^m k/\lambda$, given by

$$P_k = \frac{\langle F_k | F_k \rangle}{\lambda}, \tag{4.80}$$

depends on k.[7] Again, this is because it is possible $f(x) = f(y)$ for some $0 \leq x, y < \lambda$.

4.5.2 The Order-Finding Algorithm

In the previous section, the function f was not specified explicitly, but the corresponding quantum oracle (4.73) was assumed to be implemented efficiently. Here we consider a specific function, the *modular exponentiation*

$$f(x) = a^x \pmod{N} \tag{4.81}$$

for fixed positive integers a and N. The two integers a and N are assumed to be coprimes. The function is a periodic function, and the primary period (the smallest period) is called the order of a modulo N. The order-fining problem is to determine the order for given a and N, and is known to be hard to solve classicaly.

Unlike the period-finding algorithm, the order-finding algorithm does not resort to quantum oracle, and implement the function with elementary gates. More precisely, it uses the quantum phase estimation algorithm to estimate the phase of the unitary operator

$$\hat{U} : |x\rangle \mapsto |ax \pmod{N}\rangle \,, \tag{4.82}$$

the quantum mechanical extension of the *modular multiplication*.

The order-finding algorithm is closely related to the factorization algorithm: The factorization algorithm is composed of two parts, one quantum mechanical and the other classical. The quantum mechanical part is just the order-finding algorithm. The classical part is to determine the factors from the order based on the number theory.

[7] This is another difference from the hidden subgroup problem.

Problems

1. Consider the quantum oracle defined in (4.7).

   (a) Classically (operating only on the basis states without any superposition of them), the mapping in (4.6) is one-to-one regardless of the function f.

   (b) Show that for any function f, the transformation $\hat{U}_f$ in (4.7) is unitary.

2. **(conditional phase shift)** Let $f : \{0,1\}^n \to \{0,1\}$ be a (classical) oracle. Supppose that we have a quantum computer consisting of an n-qubit "control register" and a single-qubit "target register". Using a quantum oracle, construct a quantum circuit model which shifts the phase by the factor $e^{i\phi}$ of every term in $|x\rangle$ satisfying $f(x) = 1$ of the n-qubit register, but keeps the second single-qubit register intact. The quantum circuit model effectively transforms the states of the control qubit as

$$\sum_x |x\rangle \mapsto \sum_x |x\rangle \, e^{i\phi f(x)}. \tag{4.83}$$

   The simple application of quantum oracle corresponds to $\phi = \pi$.

   Hint: See the implementation of the controlled-U gate in Section 2.2.2.

3. Using the orthogonality relation

$$\sum_{z=0}^{2^n-1} e^{i(x-y)p_z} = 2^n \delta_{xy}\,, \tag{4.84}$$

   prove Eq. (4.56).

4. Consider the logical state

$$|X_0\rangle \equiv |0\rangle^{\otimes n} \tag{4.85}$$

   of an n-qubit register. Show that

$$\hat{U}_{\rm QFT}\,|0\rangle^{\otimes n} = \hat{U}^\dagger_{\rm QFT}\,|0\rangle^{\otimes n} = \hat{H}^{\otimes n}\,|0\rangle^{\otimes n} = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |X_x\rangle = |P_0\rangle\,. \tag{4.86}$$

Chapter 5

Decoherence

• May 26, 2021 (v1.11)

In the previous chapters, our dicussions and arguments have been mainly based on the principles of quantum physics for closed systems. However, no realistic system is closed. A system is naturally subject to interaction with the surrounding system, which is commonly called the *environment*. There is also a more fundamental reason for the notion of an *open quantum system* in quantum mechanics. The theory of quantum mechanics is intrisincally probabilistic. It means that the verification of any quantum principle should be tested statistically through repeated measurements and the resulting data. The measurement process inevitably requires to couple the system to a measuring device. Further, in quantum computation and more general quantum information processing, we desire the preparation, manipulation, and measurement of quantum states. All those procedures require the system to be coupled to external equipments.

In principle, one can regard the combined system enclosing both the system and the environment as a closed system, and apply the quantum mechanical principles to the total system. However, the environment is typically large—since perfect isolation is impossible, the total system is eventually the whole universe—and involves a huge number of degrees of freedom. A complete microscopic description incorporating the environmental degrees of freedom is not only impractical but also of little use. First of all, such a description is tremendously complicated and hard to solve. A solution, if any, would lead to an intractable amount of information, the vast majority of which would be irrelevant to the physical effects exhibited by the system itself.

A more reasonable and practical approach is thus to seek an effective description of open quantum systems in terms of the system degrees of freedom only. The effective theory is achieved in two stages: First, due to the ignorance of the environmental degrees of freedom brings about the statiscal mixture of pure states for the system. The state of the system is not a pure state any longer, and described by the density operator. We have already introduced this description

in Section 1.1.2. Second, the influence of the environment should be reflected on the (effective) dynamical evolution of the density operator in a way that does not depend on the details of the environment and the system-environment coupling. A powerful mathematical tool is provided by the formalism of quantum operations.

In this chapter, we first introduce the quantum operations formalism. The two common and complementary representtaions of quantum operations are discussed together with simple examples. The quantum operations is used not only for dynamical processes of open quantum systems but also for the quantum theory of generalized measurement. Next, we will turn to the quantum master equation approach to open quantum systems. It is an approximate approach of the quantum operations formalism under the Markovian assumption. While the quantum operations formalism provide the most general mathematical tool, it is not always possible to find the quantum operations explicitly for given specific systems. It is far simpler and insightful to construct the quantum master equation and examine the solution to understand the behaviors of the open quantum systems in question. In the remaining part of the chapter, we introduce several concepts such as entropy and fidelity to quantify and characterize quantum information. These information theoretic concepts will be useful in the next chapter, when we discuss the quantum error correction code.

5.1 Quantum Operations

Under a certain physical process, the state of a given system evolves into another state. The time evolution of a closed system is described by unitary operators. What about an open quantum system, which interacts with its environment?

Dynamical processes of open quantum systems are described by a special kind of supermaps called *quantum operations*: A supermap maps density operators to other density operators while preserving the elementary properties of density operators. In particular, as density operators are positive,[1] a quantum operation needs to preserve positivity. However, it turns out that merely preserving positivity is not sufficient and a much stronger condition is required. Essentially, a quantum operation needs to preserve not only the positivity of density operators of a given system but also all density operators of any extended system including the system itself and its surrounding systems. Mathematically, such a condition is satisfied by *completely positive* supermaps (see Appendix 24).

Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces, and $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ a supermap. $\mathscr{F}$ is called a *quantum operation* if it satisfies the following three axioms

(a) Whenever $\hat{\rho}$ is a *density operator*[2] on $\mathcal{V}$, $0 \leq \mathscr{F}(\hat{\rho}) \leq 1$. That is, the trace is non-increasing.

[1] Recall that a positive operator is Hermitian by definition.

[2] That is, $\hat{\rho}$ is positive semi-definite and $\operatorname{Tr}\hat{\rho} = 1$. See 1.1.2 for the precise definition and properties of a density operator.

(b) $\mathscr{F}$ is *convex linear*. That is, for any probabilities p_j[3] and density operators $\hat{\rho}_j$ on $\mathcal{V}$,

$$\mathscr{F}\left(\sum_j \hat{\rho}_j p_j\right) = \sum_j \mathscr{F}(\hat{\rho}_j) p_j \,. \tag{5.1}$$

(c) $\mathscr{F}$ is a *completely positive supermap*.[4] That is, not only $\mathscr{F}(\hat{\rho})$ itself is positive for any positive operator $\hat{\rho}$ on $\mathcal{V}$, but $(\mathscr{F} \otimes \mathscr{I})(\hat{\rho})$ is also positive for any positive operator on $\mathcal{V} \otimes \mathcal{E}$ with arbitrary vector space $\mathcal{E}$.

Most quantum operations preserve the trace—$\mathrm{Tr}\,\mathscr{F}(\hat{\rho}) = 1$ for all density operators $\hat{\rho}$. An important exception is the process associated with a (generalized selective) measurement. When the trace is not preserved, $\mathrm{Tr}\,\mathscr{F}(\hat{\rho})$ gives the probability for the process $\mathscr{F}$ to occur.

In quantum information theory, quantum operations preserving trace—completely positive and trace-preserving or CPTP supermaps—are called *quantum channels*. Physically, they describe communication channels which can transmit quantum information, as well as classical information.

Another important class of physical phenomena described by quantum operations is *quantum decoherence* or just *decoherence* for short, referring to the loss of quantum coherence: Consider a quantum state and its representation in a certain basis. The components of the representation are complex numbers in general. As long as there exits a definite phase relation between different components, the state is said to be coherent. For various reasons, which are eventually traced back to interaction with the environment, the state loses coherence and the quantum effects disappear in the system. In this case, the input and output Hilbert space coincide, $\mathcal{V} = \mathcal{W}$, and the relevant quantum operations are *superoperators*.

In this section, we introduce two mathematical methods to describe quantum operations, the Kraus representation in Section 5.1.1 and the unitary representation in Section 5.1.2. We conclude this section giving examples of these representations for single-qubit systems in Section 5.1.3.

5.1.1 The Kraus Representation

The Kraus representation is an efficient and powerful method to write a quantum operation as a sum of operators. For a physical motivation, let us consider a system interacting with its environment: The system and environment are associated with the Hilbert spaces $\mathcal{V}$ and $\mathcal{E}$, respectively. For simplicity, assume that the total system is initially in the product state $\hat{\rho} \otimes \hat{\sigma}$. The total system is a closed system, and the dynamical process afterwards due to the system-environment interaction is described an overall unitary operator $\hat{U}$ acting on the total system, $\hat{\rho} \otimes \hat{\sigma} \mapsto \hat{U}(\hat{\rho} \otimes \hat{\sigma})\hat{U}^\dagger$. Without access to the environment, one has to take the

[3] That is, $0 \le p_j \le 1$ and $\sum_j p_j = 1$.

[4] In most literature,it is called a completely positive "map".

partial trace of the final overall state over the environment to obtain the state of the system. Putting all together, the quantum operation $\mathscr{F}$ describing the process is written as

$$\mathscr{F}(\hat{\rho}) = \operatorname*{Tr}_{\mathcal{E}} \hat{U}(\hat{\rho} \otimes \hat{\sigma})\hat{U}^\dagger = \sum_\mu \langle \varepsilon_\mu | \hat{U}(\hat{\rho} \otimes \hat{\sigma})\hat{U}^\dagger | \varepsilon_\mu \rangle \ , \tag{5.2}$$

where $\{|\varepsilon_\mu\rangle\}$ is an orthonormal basis of $\mathcal{E}$. On the right-hand side of (5.2), the Hermitian product is applied partially and only on $\mathcal{E}$, and the expression still remains to be an operator on $\mathcal{V}$. To further investigate the quantum operation, we take the spectral decomposition (see Appendix A.4) of $\hat{\sigma}$

$$\hat{\sigma} = \sum_\nu |s_\nu\rangle \langle s_\nu| \ , \tag{5.3}$$

where the eigenvectors $|s_\mu\rangle$ have been normalized by their own eigenvalues $s_\mu = \langle s_\mu | s_\mu \rangle$—see Eq. (A.43)—as $\hat{\rho}$ is a postive semidefinite operator. Now, define linear operators $\hat{F}_\mu$ on $\mathcal{V}$ by

$$\hat{F}_\mu := \sum_\nu \operatorname*{Tr}_{\mathcal{E}} \hat{U} \left(\hat{I} \otimes |s_\nu\rangle \langle \varepsilon_\mu| \right) = \sum_\nu \langle \varepsilon_\mu | \hat{U} | s_\nu \rangle \ . \tag{5.4}$$

Then, these linear operators satisfy the closure relation

$$\sum_\mu \hat{F}_\mu^\dagger \hat{F}_\mu = \hat{I} \, , \tag{5.5}$$

and the quantum operation is rewritten as

$$\mathscr{F}(\hat{\rho}) = \sum_\mu \hat{F}_\mu \hat{\rho} \hat{F}_\mu^\dagger \, . \tag{5.6}$$

Equation (5.6) is called a *Kraus representation* of the quantum operation $\mathscr{F}$ and the linear operators $\hat{F}_\mu$ are called the *Kraus elements* or the *Kraus operators* associated with the Kraus representation. Here we have derived a Kraus representation for a quantum operation resulting from a system-plus-environment model initially in a product state, $\hat{\rho} \otimes \hat{\sigma}$. Another notable point is that the Kraus elements $\hat{F}_\mu$ resulting from the overall unitary transformation on the total system, are not orthogonal—with respect to the trace Hermitian product in (B.4) —to each other. In particular, the number of Kraus elements may be as huge as the dimension of the environmental Hilbert space $\mathcal{E}$—the dimension of $\mathcal{E}$ is infinite for any realistic environment. However, the above method just provides a formal derivation of the Kraus elements based on a system-plus-environment model. As we will see now, the Kraus elements can be optimized and chosen to be mutually orthogonal.

Before going further, let us take an example. Consider a chain of three qubits. Suppose that the Hamiltonian of the chain is given by

$$\hat{H} = \frac{1}{2} B \hat{S}_1^z + \frac{1}{2} J \left(\hat{S}_1^x \hat{S}_2^x + \hat{S}_2^x \hat{S}_3 \right) . \tag{5.7}$$

We regard the first qubit as the "system", and the other two qubits form the "environment". This model is overly artificial as the environment is not only finite but also very small, just twice larger than the system. However, it is enough to demonstrate the main idea. We consider the whole chain is initially in the product state $|\Psi(0)\rangle = |L\rangle \otimes |L\rangle \otimes |L\rangle$, where $|L\rangle := (|0\rangle + i\,|1\rangle)/\sqrt{2}$ is the "left" state—it is often used to denote the left-circularly polarized state of a photon. When focused on the system only, the initial state is $|L\rangle$, or equivalently, $\hat{\rho}(0) = |L\rangle\langle L| = \frac{1}{2} + \frac{1}{2}S_1^y$. At later time t, the state of the chain is given by

$$|\Psi(t)\rangle = \hat{U}(t)\,|\Psi(0)\rangle \tag{5.8}$$

with $\hat{U}(t) = \exp(-it\hat{H})$. Ignoring the two qubits in the environment, we get the (mixed) state of the sysstem (the first qubit)

$$\hat{\rho}(t) = \operatorname*{Tr}_{\mathcal{E}} |\Psi(t)\rangle\langle\Psi(t)| = \frac{1}{2} - \frac{1}{2}\frac{\sin(\Omega t)}{\Omega}\hat{S}_1^x + \frac{1}{2}\cos(\Omega t)\hat{S}_1^y\,, \tag{5.9}$$

where $\Omega := \sqrt{B^2 + J^2}$. Now, let us describe the evolution $\hat{\rho}(0) \to \hat{\rho}(t)$ in terms of quantum operation $\mathscr{F}$. More specifically, we want to find the Kraus elements $\hat{F}_\mu(t)$ for the quantum operation so that

$$\hat{\rho}(t) = \sum_\mu \hat{F}_\mu(t)\hat{\rho}(0)\hat{F}_\mu^\dagger(t). \tag{5.10}$$

Following the prescription in (5.4), we get the Kraus elements

$$\hat{F}_\mu(t) = \operatorname*{Tr}_{\mathcal{E}} \hat{U}(t)\,\left(\hat{I} \otimes |L\rangle\langle\mu_1| \otimes |L\rangle\langle\mu_2|\right)\,, \tag{5.11}$$

where μ_j are the binary digits of $\mu = (\mu_1\mu_2)_2$. More explicitly, they are given by

$$\begin{aligned}
\hat{F}_0(t) &= \frac{Je^{-\frac{1}{2}iJt}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^x}{2\Omega} - \frac{ie^{\frac{iJt}{2}}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^z}{2\Omega} + \frac{1}{2}e^{\frac{iJt}{2}}\cos\left(\frac{\Omega t}{2}\right),\\
\hat{F}_1(t) &= \frac{Je^{\frac{iJt}{2}}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^x}{2\Omega} - \frac{ie^{-\frac{1}{2}iJt}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^z}{2\Omega} + \frac{1}{2}e^{-\frac{1}{2}iJt}\cos\left(\frac{\Omega t}{2}\right),\\
\hat{F}_2(t) &= \frac{Je^{\frac{iJt}{2}}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^x}{2\Omega} + \frac{ie^{-\frac{1}{2}iJt}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^z}{2\Omega} - \frac{1}{2}e^{-\frac{1}{2}iJt}\cos\left(\frac{\Omega t}{2}\right),\\
\hat{F}_3(t) &= \frac{Je^{-\frac{1}{2}iJt}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^x}{2\Omega} + \frac{ie^{\frac{iJt}{2}}\sin\left(\frac{\Omega t}{2}\right)\hat{S}_1^z}{2\Omega} - \frac{1}{2}e^{\frac{iJt}{2}}\cos\left(\frac{\Omega t}{2}\right).
\end{aligned}$$

One can convince oneself by direct evaluations that these Kraus elements indeed reproduces the dynamical evolution in (5.10), and that $\sum_\mu \hat{F}_\mu^\dagger(t)\hat{F}_\mu(t) = \hat{I}$. These Kraus elements are not mutually orthogonal, but they can be orthogonalized by following the lines in Theorem 23.

We consider a chain of three qubits. The first qubit is regarded as the "system", and the other two form the "environment".

```
$L = 3;
jj = Range[$L];
sys = 1;
env = Range[2, $L];
```

Here is the Hamiltonian describing the chain. We are measuring the energy in units of B (B=1).

```
In[•]:= Let[Real, J]
        H = S[sys, 3] / 2 + J / 2 * Total[ChainBy[S[jj, 1], Multiply]]
```

$$\text{Out[•]}= \frac{1}{2}\,\mathtt{J}\,\left(\mathtt{S}_1^{\mathtt{x}}\,\mathtt{S}_2^{\mathtt{x}}+\mathtt{S}_2^{\mathtt{x}}\,\mathtt{S}_3^{\mathtt{x}}\right)+\frac{\mathtt{S}_1^{\mathtt{z}}}{2}$$

The system has a discrete symmetry. It is invariant under the rotation around the z-axis by angle π.

```
In[•]:= V = Rotation[Pi, S[1, 3]] ** Rotation[Pi, S[2, 3]] ** Rotation[Pi, S[3, 3]]
        V ** H ** Dagger[V]
```

$$\text{Out[•]}= \mathtt{i}\,\mathtt{S}_1^{\mathtt{z}}\,\mathtt{S}_2^{\mathtt{z}}\,\mathtt{S}_3^{\mathtt{z}}$$

$$\text{Out[•]}= \frac{1}{2}\,\mathtt{J}\,\mathtt{S}_1^{\mathtt{x}}\,\mathtt{S}_2^{\mathtt{x}}+\frac{1}{2}\,\mathtt{J}\,\mathtt{S}_2^{\mathtt{x}}\,\mathtt{S}_3^{\mathtt{x}}+\frac{\mathtt{S}_1^{\mathtt{z}}}{2}$$

The symmetry leads to the degeneracy of the eigenvalues of the Hamiltonian.

```
In[•]:= ProperValues[H]
```

$$\text{Out[•]}= \left\{\frac{1}{2}\left(-\mathtt{J}-\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(-\mathtt{J}-\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(\mathtt{J}-\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(\mathtt{J}-\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(-\mathtt{J}+\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(-\mathtt{J}+\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(\mathtt{J}+\sqrt{1+\mathtt{J}^2}\right),\ \frac{1}{2}\left(\mathtt{J}+\sqrt{1+\mathtt{J}^2}\right)\right\}$$

The time-evolution operator of the chain is evaluated.

```
Let[Real, t]
U[t_] = Elaborate@MultiplyExp[-I t H];
```

We suppose that the chain is initially in the state $\left|\mathtt{L}\right\rangle\otimes\left|\mathtt{L}\right\rangle\otimes\left|\mathtt{L}\right\rangle$, where $\left|\mathtt{L}\right\rangle = \left(\left|0\right\rangle + \mathtt{i}\,\left|1\right\rangle\right)/\sqrt{2}$.

```
In[•]:= vec[0] = ProductState[S@jj → {1, I} / Sqrt[2]]
        vec[t_] = U[t] ** Elaborate[vec[0]];
```

$$\text{Out[•]}= \left(\frac{\left|0\right\rangle}{\sqrt{2}}+\frac{\mathtt{i}\,\left|1\right\rangle}{\sqrt{2}}\right)_{\mathtt{S}_1}\otimes\left(\frac{\left|0\right\rangle}{\sqrt{2}}+\frac{\mathtt{i}\,\left|1\right\rangle}{\sqrt{2}}\right)_{\mathtt{S}_2}\otimes\left(\frac{\left|0\right\rangle}{\sqrt{2}}+\frac{\mathtt{i}\,\left|1\right\rangle}{\sqrt{2}}\right)_{\mathtt{S}_3}$$

Here is a set of replacement rules to be used later to simplify expressions.

```
In[•]:= rules = {Sqrt[1 + J^2] → Ω, 1 / Sqrt[1 + J^2] → 1 / Ω}
```

$$\text{Out[•]}= \left\{\sqrt{1+\mathtt{J}^2}\to\Omega,\ \frac{1}{\sqrt{1+\mathtt{J}^2}}\to\frac{1}{\Omega}\right\}$$

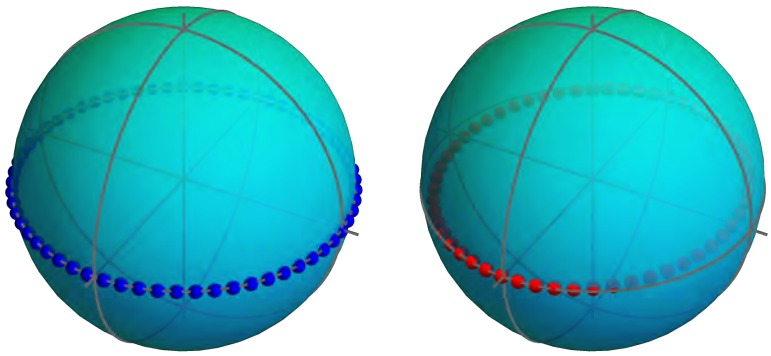
```
In[•]:= rho[t_] = PartialTrace[vec[t], S@env] // Elaborate // ExpToTrig // Garner;
        rho[t] /. rules
```

$$\text{Out[•]}= \frac{1}{2}+\frac{1}{2}\,\mathtt{Cos}[\mathtt{t}\,\Omega]\,\mathtt{S}_1^{\mathtt{y}}-\frac{\mathtt{S}_1^{\mathtt{x}}\,\mathtt{Sin}[\mathtt{t}\,\Omega]}{2\,\Omega}$$

Take a look at the dynamical evolution of the state of the "system" (the first qubit), after tracing out the "environment" (the other two qubits). On the left, shown is the evolution in the absence of the coupling to the environment. On the right, the evolution is not coherent due to the coupling to the environment. The evolution is still periodic because the environment is finite.

```
In[•]:= bv0 = Block[{J = 0.}, Table[BlochVector[rho[t]], {t, 0, 2 Pi, 0.1}]];
        bv = Block[{J = .5}, Table[BlochVector[rho[t]], {t, 0, 2 Pi, 0.1}]];
        GraphicsRow@{BlochSphere[{Blue, Bead /@ bv0}], BlochSphere[{Red, Bead /@ bv}]}
```

Out[•]=



Now, let us examine the evolution in terms of the supermap. This is the initial state of the system.

```
In[•]:= in = Elaborate@ProductState[S[sys] → {1, I} / Sqrt[2]];
        in = Elaborate@Dyad[in, in]
```

Out[•]= $\frac{1}{2} + \frac{S_1^y}{2}$

To find the Kraus elements, consider the initial state of the environment.

```
In[•]:= sgm = Elaborate@ProductState[S@env → {1, I} / Sqrt[2]];
        sgm // LogicalForm
```

Out[•]= $\frac{1}{2}\left|0_{S_2}0_{S_3}\right\rangle + \frac{1}{2}\,i\left|0_{S_2}1_{S_3}\right\rangle + \frac{1}{2}\,i\left|1_{S_2}0_{S_3}\right\rangle - \frac{1}{2}\left|1_{S_2}1_{S_3}\right\rangle$

Finally, this is the Kraus elements of the quantum operation.

```
bs = Basis[S@env];
prj = Map[Dyad[sgm, #, S@env] &, bs];
ops = U[t] ** prj // Elaborate;
```

```
In[•]:= kraus = PartialTrace[#, S@env] & /@ ops // ExpToTrig // Elaborate // Garner;
        kraus /. rules
```

$$\Big\{\frac{1}{2}\,\text{Cos}\Big[\frac{t\,\Omega}{2}\Big]\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]+i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)+\frac{J\,S_1^x\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]-i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega}+\frac{S_1^z\,\Big(-i\,\text{Cos}\Big[\frac{J\,t}{2}\Big]+\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega},\ \frac{1}{2}\,\text{Cos}\Big[\frac{t\,\Omega}{2}\Big]\,\Big(i\,\text{Cos}\Big[\frac{J\,t}{2}\Big]+\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)+\frac{S_1^z\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]-i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega}+\frac{i\,J\,S_1^x\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]+i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega},\ \frac{1}{2}\,\text{Cos}\Big[\frac{t\,\Omega}{2}\Big]\,\Big(i\,\text{Cos}\Big[\frac{J\,t}{2}\Big]+\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)+\frac{S_1^z\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]-i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega}+\frac{J\,S_1^x\,\Big(-i\,\text{Cos}\Big[\frac{J\,t}{2}\Big]+\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega},\ -\frac{1}{2}\,\text{Cos}\Big[\frac{t\,\Omega}{2}\Big]\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]+i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)+\frac{J\,S_1^x\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]-i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega}+\frac{i\,S_1^z\,\Big(\text{Cos}\Big[\frac{J\,t}{2}\Big]+i\,\text{Sin}\Big[\frac{J\,t}{2}\Big]\Big)\,\text{Sin}\Big[\frac{t\,\Omega}{2}\Big]}{2\,\Omega}\Big\}$$

```
In[•]:= new[t_] = Elaborate@Supermap[kraus][in];
        new[t] /. rules
```

$$\frac{1}{2}+\frac{1}{2}\,\text{Cos}[t\,\Omega]\,S_1^y-\frac{S_1^x\,\text{Sin}[t\,\Omega]}{2\,\Omega}$$

```
In[•]:= new[t] - rho[t] // Elaborate // Garner
Out[•]= 0
```

So far, we have discussed the Kraus representation of a quantum operation based on a system-plus-environment model. Kraus representation is far more general: Indeed, as indicated by axiom (c), a quantum operation is a completely positive supermap, which is mathematically gauranteed to have a Kraus representation—see Appendix B.2.2: For any *quantum operation* $\mathscr{F}:\mathcal{L}(\mathcal{V})\to\mathcal{L}(\mathcal{W})$, there exists a set of Kraus elements $\hat{F}_\mu\in\mathcal{L}(\mathcal{V},\mathcal{W})$ such that for all $\hat{\rho}\in\mathcal{L}(\mathcal{V})$

$$\mathscr{F}(\hat{\rho})=\sum_\mu \hat{F}_\mu\hat{\rho}\hat{F}_\mu^\dagger\,. \tag{5.12}$$

Furhter, the Kraus elements can always be chosen—see Eqs. (5.30) and (5.31) below—to be *mutually orthogonal* with respect to the trace Hermitian product,

$$\text{Tr}\,\hat{F}_\mu^\dagger\hat{F}_\nu=0\quad(\mu\neq\nu)\,. \tag{5.13}$$

A set of mutually orthogonal Kraus elements is optimal in the sense that it has no more elements than $(\dim\mathcal{V})\times(\dim\mathcal{W})$. The trace-decreasing condition in axiom (a) imposes the inequalities

$$0\le\sum_\mu \hat{F}_\mu^\dagger\hat{F}_\mu\le 1\,. \tag{5.14}$$

It is clear that any supermap given in the above form satisfy the three axioms, and is a quantum operation. The converse is more complicated to prove. There are

three common ways, each of which is interesting on its own right. The first method—see Exercise 11—is directly based on the general operator-sum representation in (B.16). Here we will discuss the other two methods.

Before discussing the proofs, we introduce an interesting isomorphism between completely positive supermaps and density operators. This isomorphism is a further refinement of the *Choi isomorphism* between supermaps and operators (Appendix B.2.3). To exploit the condition of $\mathscr{F}$ being completely positive later, take a copy of the original vector space $\mathcal{V}$ as a reference space (or any vector space $\mathcal{R}$ of the same dimension as $\mathcal{V}$), and construct the tensor-product space $\mathcal{V} \otimes \mathcal{V}$ of the original and reference space. Then, consider a maximally entangled state

$$|\Phi\rangle := \sum_k |v_k\rangle \otimes |v_k\rangle \ , \tag{5.15}$$

where $\{|v_k\rangle\}$ is an orthonormal basis of $\mathcal{V}$. Its density operator is given by

$$|\Phi\rangle \langle\Phi| = \sum_{kl} |v_k\rangle \langle v_l| \otimes |v_k\rangle \otimes \langle v_l| \, . \tag{5.16}$$

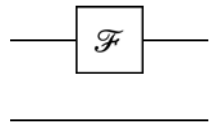
Now operate an extended supermap $\mathscr{F} \otimes \mathscr{I}$, where $\mathscr{I}$ is the identity superoperator, on $|\Phi\rangle \langle\Phi|$ to get

$$\hat{C}_{\mathscr{F}} := (\mathscr{F} \otimes \mathscr{I})(|\Psi\rangle \langle\Psi|) = \sum_{ij} \sum_{kl} |w_i v_k\rangle \langle w_j v_l| \, C_{ik;jl} \in \mathcal{L}(\mathcal{W} \otimes \mathcal{V}), \tag{5.17}$$

where C is the *Choi matrix* (see Appendix B.2.3) associated with $\mathscr{F}$,

$$\mathscr{F}(|v_k\rangle \langle v_l|) = \sum_{ij} |w_i\rangle \langle w_j| \, C_{ik;jl} \, . \tag{5.18}$$

Equation (5.17) implies that $\hat{C}_{\mathscr{F}}$ is an operator (not a superoperator) on $\mathcal{W} \otimes \mathcal{V}$ with the matrix representation given just by the Choi matrix C—the operator $\hat{C}_{\mathscr{F}}$ and the matrix C are essentially the same mathematical objects. We call $\hat{C}_{\mathscr{F}}$ the *Choi operator* associated with the supermap $\mathscr{F}$. The association $\mathscr{F} \mapsto \hat{C}_{\mathscr{F}}$ is an isomorphism between supermaps and operators, and called the *Choi isomorphism*. The Choi operator $\hat{C}_{\mathscr{F}}$ (and hence the Choi matrix C) completely characterizes the associated supermap $\mathscr{F}$. To see it in a more physically transparent way, it is useful to represent the Choi operator $\hat{C}_{\mathscr{F}}$ in a quantum circuit model of the form

$\hat{C}_{\mathscr{F}} =$ $|\Phi\rangle$  (5.19)

The Choi operator $\hat{C}_{\mathscr{F}}$ is the result of the evolution of the maximally entangled state $|\Phi\rangle$ in $\mathcal{V} \otimes \mathcal{V}$ composed of the original and reference space evolves under the supermap $\mathscr{F}$ acting only on the original system. One way to quantitatively

characterize the supermap $\mathscr{F}$ is thus to compare $\hat{C}$ with the initial maximally entangled state $|\Phi\rangle$,

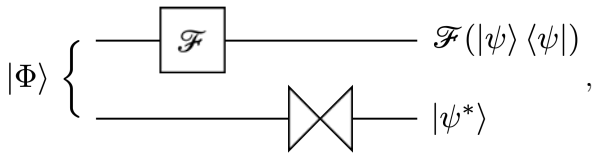
$$F_E := \mathrm{Tr}\,|\Phi\rangle\langle\Phi|\,\hat{C}_{\mathscr{F}} = \langle\Phi|\,\hat{C}_{\mathscr{F}}\,|\Phi\rangle\,. \tag{5.20}$$

The quantitative measure F_E is called the *entanglement fidelity*[5] of the supermap $\mathscr{F}$. It quantifies how well the supermap preserves the initial quantum information. The Choi operator also reflects the properties of the associated supermap: When $\mathscr{F}$ is completely positive, $\hat{C}_{\mathscr{F}}$ is a positive operator. Further, when the supermap $\mathscr{F}$ is a *quantum operation*, satisfying all the axioms (a)–(c), $\hat{C}_{\mathscr{F}}$ is a density operator. Interestingly, the Choi isomorphism remains to hold between *quantum operations* and *density operators*.[6] This refined isomorphism is called the *channel-state duality*. Below we provide two additional proofs of the Kraus representation theorem using the channel-state duality.

Let us get back to the proofs of the Kraus representation theorem. The first method relies on the property of the Choi operator $\hat{C}_{\mathscr{F}}$: For any pure state $|\psi\rangle = \sum_j |v_j\rangle\,\psi_j \in \mathcal{V}$, define its conjugate state $|\psi^*\rangle := \sum_j |v_j\rangle\,\psi_j^*$, and observe that

$$\mathscr{F}(|\psi\rangle\langle\psi|) = \mathop{\mathrm{Tr}}_{\mathcal{V}}\left[\left(\hat{I}\otimes|\psi^*\rangle\langle\psi^*|\right)\hat{C}_{\mathscr{F}}\right] = \langle\psi^*|\,\hat{C}_{\mathscr{F}}\,|\psi^*\rangle\,. \tag{5.21}$$

Note that the Hermitian product in $\langle\psi^*|\,\hat{C}_{\mathscr{F}}\,|\psi^*\rangle$ is merely a short-hand notation for the partial trace, and $\langle\psi^*|\,\hat{C}_{\mathscr{F}}\,|\psi^*\rangle$ is an operator on $\mathcal{V}$ (not a number). Recalling the quantum circuit representation (5.19) of the Choi isomorphism, the identity (5.21) can be described by the quantum circuit model



, (5.22)

where the quantum circuit element —⋈— represents the projection onto the state specified at the output port. Since $\hat{C}_{\mathscr{F}}$ is a positive operator for a quantum operation $\mathscr{F}$, rewrite it in a spectral decomposition

$$\hat{C}_{\mathscr{F}} = \sum_{\mu} |\varphi_\mu\rangle\langle\varphi_\mu|\,, \tag{5.23}$$

where each vector $|\varphi_\mu\rangle$ has been normalized so that $\langle\varphi_\mu|\varphi_\mu\rangle$ gives the corresponding (positive) eigenvalue of $\hat{C}_{\mathscr{F}}$, $\hat{C}_{\mathscr{F}}\,|\varphi_\mu\rangle = |\varphi_\mu\rangle\langle\varphi_\mu|\varphi_\mu\rangle$. We define a linear map $\hat{F}_\mu : \mathcal{V} \to \mathcal{W}$ by the association

$$\hat{F}_\mu\,|\psi\rangle = \langle\psi^*|\varphi_\mu\rangle\,. \tag{5.24}$$

[5] It is a special case of more general notion of *fideltity* to be discussed in Section 5.4.

[6] Note that an isomorphism between two spaces does not necessarily hold between their subspaces.

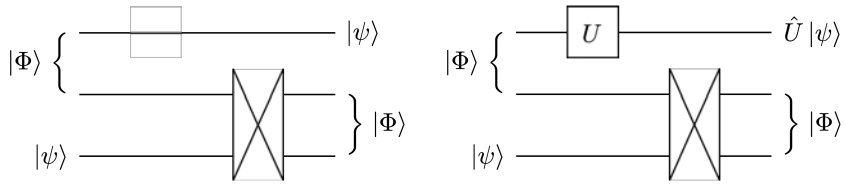


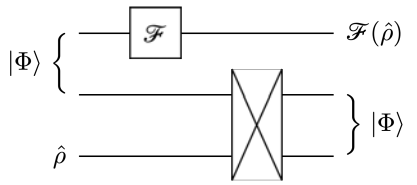
Figure 5.1: Comparison of quantum teleportation and quantum gate teleportation. (a) A simplified quantum circuit model of quantum teleportation. The Bell measurement is replaced by the projection to $|\Phi\rangle$, and the success probability is 1/4. (b) A quantum circuit model for quantum gate teleportation. The input state $|\psi\rangle$ on the third qubit results in the unitary-transformed state $\hat{U}|\psi\rangle$ on the first qubit.

Note that on the right-hand side of the relation, the Hermitian product is applied partialy and only on $\mathcal{V}$—the remaining part is a vector belonging to $\mathcal{W}$. Putting (5.23) and (5.24) into (5.21), we confirm that

$$\mathscr{F}(|\psi\rangle\langle\psi|) = \sum_{\mu} \hat{F}_{\mu}|\psi\rangle\langle\psi|\hat{F}_{\mu}^{\dagger}. \tag{5.25}$$

As $\mathscr{F}$ is linear and $|\psi\rangle$ is arbitrary, this proves the statement in the theorem.

Now, turn to the second proof: It is based on the so-called *quantum gate teleportation* protocol. Figure 5.1 (a) shows a simplified quantum circuit model of the quantum teleportation protocol. Compared with the typical quantum teleportation protocol discussed in Section 4.1, the Bell measurement has been replaced with the projection onto a single Bell state $|\Phi\rangle$. The protocol is not deterministic any longer, and with the success probability 1/4, the input state $|\psi\rangle$ on the third qubit is "teleported" to the first qubit. At the end of the protocol, one can apply any unitary transformation $\hat{U}$ to get $\hat{U}|\psi\rangle$. The result does not change even if one applies $\hat{U}$ even before the projection. This variation leads to the quantum circuit model depicted in Fig. 5.1 (b), which is commonly called the quantum gate teleportation protocol. Using the quantum resourse of maximal entanglement,[7] $\hat{U}\otimes\hat{I}|\Phi\rangle$, it moves the input state $|\Phi\rangle$ on the third qubit to the unitary-transformed state $\hat{U}|\psi\rangle$ on the first qubit. The success probability of the protocol is 1/4. The quantum gate teleportation protocol can be generalized (Problem 1) to supermaps for the gate operation and to mixed states for the input state as in the following quantum circuit model



. (5.26)

[7] As $\hat{U}\otimes\hat{I}$ only operates *locallly*, it does not modify the entanglement characteristics of $|\Phi\rangle$. Note also that the state$|U\rangle := \hat{U}\otimes\hat{I}|\Phi\rangle$ is the *Choi vector*associated with the unitary operator $\hat{U}$—see Appendix B.2.3.

Now consider a state $|\psi\rangle$. In accordance with the quantum gate teleportation protocol in (5.26) and the Choi isomorphism in (5.19), one has

$$\mathscr{F}(|\psi\rangle\langle\psi|)|v_j\rangle = \sum_i |v_i\rangle\left(\langle v_i|\otimes\langle\Phi|\right)\left(\hat{C}_{\mathscr{F}}\otimes|\psi\rangle\langle\psi|\right)\left(|v_j\rangle\otimes|\Phi\rangle\right) \tag{5.27}$$

Again, as $\hat{C}_{\mathscr{F}}$ is positive if $\mathscr{F}$ is a quantum operation, we use the spectral decomposition (5.23) of $\hat{C}_{\mathscr{F}}$. Finally, define a set of linear operators $\hat{F}_\mu$ by

$$\hat{F}_\mu : |\psi\rangle \mapsto \sum_i |v_i\rangle\left(\langle v_i|\otimes\langle\Phi|\right)\left(|\varphi_\mu\rangle\otimes|\psi\rangle\right). \tag{5.28}$$

Then, we find that

$$\mathscr{F}(|\psi\rangle\langle\psi|) = \sum_\mu \hat{F}_\mu|\psi\rangle\langle\psi|\hat{F}_\mu^\dagger, \tag{5.29}$$

which proves the Kraus representation theorem.

We close this subsection by noting that the Kraus repesentation is not unique and there exists unitary freedom for the choice of the Kraus elements: Suppoe that the two quantum operations $\mathscr{F}$ and $\mathscr{G}$ are associated with the Kraus elements $\{\hat{F}_\mu\}$ and $\{\hat{G}_\nu\}$, respectively.[8] Then, $\mathscr{F}=\mathscr{G}$, that is,

$$\sum_\mu \hat{F}_\mu\hat{\rho}\hat{F}_\mu^\dagger = \sum_\nu \hat{G}_\nu\hat{\rho}\hat{G}_\nu^\dagger \tag{5.30}$$

for all $\hat{\rho}\in\mathcal{L}(\mathcal{V})$ if and only if there exists a unitary matrix U such that

$$\hat{G}_\nu = \sum_\mu \hat{F}_\mu U_{\mu\nu}. \tag{5.31}$$

This is analgous the unitary freedom for the choice of pure states in the specification of a mixed state—see Eqs. (1.11) and (1.12). In fact, the underlying mathematical principles are the same. As we have already established the proof of the unitary freedom in the mixed state, here let us use it to prove the unitary freedom in the Kraus representation.

If the two sets of Kraus elements satisfy the relation (5.31), it is straight forward to prove the two quantum operations are identical—it immediately follows from the defining property of a unitary matrix. Let us prove the converse. Suppose that $\mathscr{F}=\mathscr{G}$. Then the corresponding Choi operators—see Appendix B.2.3—should be identical as well, $\hat{C}_{\mathscr{F}}=\hat{C}_{\mathscr{G}}$. Given the Kraus elements, one can evaluate the Choi operators explicitly starting from the maximally entangled state in Eq. (5.15),

$$\hat{C}_{\mathscr{F}} = \sum_\mu(\hat{F}_\mu\otimes\hat{I})|\Phi\rangle\langle\Phi|(\hat{F}_\mu\otimes\hat{I})^\dagger = \sum_\mu |F_\mu\rangle\langle F_\mu| \tag{5.32a}$$

[8] The Kraus elements here are not orthogonal, $\mathrm{Tr}\,\hat{F}_\mu^\dagger\hat{F}_\nu\neq 0$ and $\mathrm{Tr}\,\hat{G}_\mu^\dagger\hat{G}_\nu\neq 0$, in general.

$$\hat{C}_{\mathscr{G}} = \sum_{\nu} (\hat{G}_\nu \otimes \hat{I}) \, |\Phi\rangle \, \langle\Phi| \, (\hat{G}_\nu \otimes \hat{I})^\dagger = \sum_{\nu} |G_\nu\rangle \, \langle G_\nu| \, , \tag{5.32b}$$

where $|F_\mu\rangle \, , |G_\nu\rangle \in \mathcal{W}\otimes\mathcal{V}$ are the Choi vectors—see Appnedix B.2.3—corresponding to the linear maps $\hat{F}_\mu$ and $\hat{G}_\nu$, respectively,

$$|F_\mu\rangle := (\hat{F}_\mu \otimes \hat{I}) \, |\Phi\rangle \, , \quad |G_\nu\rangle := (\hat{G}_\nu \otimes \hat{I}) \, |\Phi\rangle \, . \tag{5.33}$$

According to Eqs. (1.11) and (1.12),

$$\sum_{\mu} |F_\mu\rangle \, \langle F_\mu| = \sum_{\nu} |G_\nu\rangle \, \langle G_\nu| \tag{5.34}$$

implies that there exists a unitary matrix U such that

$$|G_\nu\rangle = \sum_{\mu} |F_\mu\rangle \, U_{\mu\nu} \, . \tag{5.35}$$

Finally, we note—Appendix B.2.3—that for arbitrary $|\psi\rangle \in \mathcal{V}$

$$\hat{G}_\nu \, |\psi\rangle = \langle\psi^*|F_\nu\rangle = \sum_{\mu} \langle\psi^*|E_\mu\rangle \, U_{\mu\nu} = \sum_{\mu} \hat{F}_\mu \, |\psi\rangle \, U_{\mu\nu} \, . \tag{5.36}$$

This asserts the relation (5.31).

In some cases, say, motivated by the (unperturbed) Hamiltonian of the isolated system, there may be a preferred basis. Can we exploit the unitary freedom to change the given set of Kraus elements to another set of Kraus elements that is consistent with the preferred basis? Unfortunately, given two sets of Kraus elements, it is not trivial to check if they are equivalent or not. It is because the Kraus elements in the relation (5.31) are not normalized. In terms of the normalized Kraus elements, $\hat{F}'_\mu = \hat{F}_\mu \sqrt{p_\mu}$ and $\hat{G}'_\nu = \hat{G}_\nu \sqrt{q_\nu}$, the relation reads as

$$\hat{G}'_\nu = \sum_{\mu} \hat{F}'_\mu \sqrt{p_\mu} U_{\mu\nu} / \sqrt{q_\nu} . \tag{5.37}$$

As $\hat{F}'_\mu$ and $\hat{G}'_\nu$ are orthonormal, the matrix $\sqrt{p_\mu} U_{\mu\nu} / \sqrt{q_\nu}$ should also be unitary. In general, it is not trivial to find a unitary matrix U which allows $\sqrt{p_\mu} U_{\mu\nu} / \sqrt{q_\nu}$ to be unitary as well.

5.1.2 Unitary Representation

A quantum operation can be regarded as a unitary operator on an extended system, which involves an "environment" in addition to the original "system". Although it is not particularly useful in practical applications, the unitary representation provides a clear physical insight into the underlying physical processes described by the quantum operation.

Suppose that we are given a quantum operation $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ represented by the Kraus representation in Eq. (5.12) in terms of the Kraus elements $\hat{F}_\mu$ ($\mu = 0, 1, \ldots, m-1$): We want to construct a system-plus-environment model which gives the same effect as $\mathscr{F}$ when the environment is traced out. We need to find a proper vector space $\mathcal{E}$ for the environment and an overall unitary operator $\hat{U}$ acting on the total system $\mathcal{V} \otimes \mathcal{E}$ such that

$$\mathscr{F}(\hat{\rho}) = \operatorname*{Tr}_{\mathcal{E}} \hat{U} \left(\hat{\rho} \otimes |\varepsilon_0\rangle \langle\varepsilon_0| \right) \hat{U}^\dagger \tag{5.38}$$

for all $\hat{\rho} \in \mathcal{L}(\mathcal{V})$ and a particular state $|\varepsilon_0\rangle \in \mathcal{E}$.[9] We first construct the vector space $\mathcal{E}$ associated with the environment by choosing an orthonormal basis $\{|\varepsilon_\mu\rangle : \mu = 0, \ldots, m-1\}$.[10] We note that the dimension of $\mathcal{E}$ is the same as the number of the Kraus elements $\hat{F}_\mu$ in the Kraus representation (5.12) and no larger than $(\dim \mathcal{V}) \times (\dim \mathcal{W})$. Define a unitary operator $\hat{U}$ on $\mathcal{V} \otimes \mathcal{W}$ by requiring that

$$\hat{U} |\psi\rangle \otimes |\varepsilon_0\rangle = \sum_\mu (\hat{F}_\mu |\psi\rangle) \otimes |\varepsilon_\mu\rangle \tag{5.39}$$

for any $|\psi\rangle \in \mathcal{V}$. Clearly, taking the partial trace over the environment reproduces $\mathscr{F}(|\psi\rangle \langle\psi|)$ as one can see from an explicit evaluation

$$\begin{aligned} \operatorname*{Tr}_{\mathcal{E}} \hat{U} \left(|\psi\rangle \langle\psi| \otimes |\varepsilon_0\rangle \langle\varepsilon_0| \right) \hat{U}^\dagger &= \operatorname*{Tr}_{\mathcal{E}} \sum_{\mu\nu} \left(\hat{F}_\mu |\psi\rangle \langle\psi| \hat{F}_\nu^\dagger \right) \otimes |\varepsilon_\mu\rangle \langle\varepsilon_\nu| \\ &= \sum_\mu \hat{F}_\mu |\psi\rangle \langle\psi| \hat{F}_\mu^\dagger \end{aligned} \tag{5.40}$$

This relation is linear and holds for arbitrary vector $|\psi\rangle$, and it should holds for any mixed state $\hat{\rho} = \sum_j |\psi_j\rangle p_j \langle\psi_j|$.

5.1.3 Examples

Phase damping The phase damping process is a decoherence process without involving any relaxation of energy or change in the population over the states. In this sense, it may be regarded as a pure decoherence process without involving any energy relaxation. For this reason, it is also called a *dephasing* process. The unitary representation of a phase damping process in a single-qubit system is given by

$$|0\rangle \otimes |\varepsilon_0\rangle \mapsto |0\rangle \otimes |\varepsilon_0\rangle \tag{5.41}$$

$$|1\rangle \otimes |\varepsilon_0\rangle \mapsto |1\rangle \otimes |\varepsilon_0\rangle \sqrt{1-p} + |1\rangle \otimes |\varepsilon_1\rangle \sqrt{p}\,, \tag{5.42}$$

[9] The choice of $|\varepsilon_0\rangle$ is completely arbitrary, and one can even choose a mixed state.

[10] Here, just for convenience, we have chosen the basis so as for it to include $|\varepsilon_0\rangle$, but it is not necessary.

where $\{|\varepsilon_0\rangle, |\varepsilon_1\rangle\}$ is an orthonormal basis of the vector space $\mathcal{E}$ associated with the environment. It describes that when and only when the system is in $|1\rangle$, the environment changes its state from the initial state $|\varepsilon_0\rangle$ to another orthogonal state $|\varepsilon_1\rangle$ with probability p. Note that the system remains in the same state in both cases. The key point is that nevertheless, the environment "knows" which state the system is in and this knowledge leads to the loss of coherence in the state of the system.

Indeed, the total unitary operator is a controlled-U operator with

$$\hat{U} \doteq \begin{bmatrix} \sqrt{1-p} & -\sqrt{p} \\ \sqrt{p} & \sqrt{1-p} \end{bmatrix} \tag{5.43}$$

in the basis $\{|\varepsilon_0\rangle, |\varepsilon_1\rangle\}$. As a result, when the system is prepared in a coherent superposition, the controlled-U operation creates an entanglement between the system and the environment (see Sections 2.2.1 and 2.2.2)

$$(|0\rangle c_0 + |1\rangle c_1) \otimes |\varepsilon_0\rangle \mapsto |0\rangle \otimes |\varepsilon_0\rangle c_0 + |1\rangle \otimes (\hat{U}|\varepsilon_0\rangle) c_1 \tag{5.44}$$

or, more explicitly,

$$(|0\rangle c_0 + |1\rangle c_1) \otimes |\varepsilon_0\rangle \mapsto \left(|0\rangle c_0 + |1\rangle c_1 \sqrt{1-p}\right) \otimes |\varepsilon_0\rangle + |1\rangle \otimes |\varepsilon_1\rangle c_1 \sqrt{p}\,. \tag{5.45}$$

Due to the entanglement, the final state of the system alone cannot be a pure state (see Section 1.1.2) and coherence in the initial state has been lost through the process.

With the prescription in Section 5.1.1, the Kraus elements are given by

$$\hat{E}_0 \doteq \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, \quad \hat{E}_1 \doteq \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{p} \end{bmatrix} \tag{5.46}$$

and the corresponding quantum operation is given by

$$\mathscr{F}(\hat{\rho}) = \hat{E}_0 \hat{\rho} \hat{E}_0^\dagger + \hat{E}_1 \hat{\rho} \hat{E}_1^\dagger \tag{5.47}$$

Note that they are not orthogonal, $\mathrm{Tr}\, \hat{E}_0^\dagger \hat{E}_1 = \sqrt{p(1-p)} \neq 0$. It is more convenient and efficient to choose mutually orthogonal Kraus elements

$$\hat{F}_0 = \sqrt{\frac{1+\sqrt{1-p}}{2}}\, \hat{I}\,, \quad \hat{F}_1 = \sqrt{\frac{1-\sqrt{1-p}}{2}}\, \hat{Z} \tag{5.48}$$

In short, the quantum operation for the phase damping process is written as

$$\mathscr{F}(\hat{\rho}) = \frac{1+\sqrt{1-p}}{2} \hat{\rho} + \frac{1-\sqrt{1-p}}{2} \hat{Z} \hat{\rho} \hat{Z} \tag{5.49}$$

Amplitude damping The amplitude damping process describes the spontaneous decay of the excited state $|1\rangle$ of the system to the ground state $|0\rangle$. The decay is accompanied with the emission of a photon. One can regards that the photon "observes" the system and the information of the system acquired by the photon leads to decoherence. In the unitary representation, the process is described by the overall unitary operator such that

$$|0\rangle \otimes |\varepsilon_0\rangle \mapsto |0\rangle \otimes |\varepsilon_0\rangle \,, \tag{5.50a}$$

$$|1\rangle \otimes |\varepsilon_0\rangle \mapsto |1\rangle \otimes |\varepsilon_0\rangle \sqrt{1-p} + |0\rangle \otimes |\varepsilon_1\rangle \sqrt{p} \,. \tag{5.50b}$$

The decay occurs with probability p provided that the system is in the state $|1\rangle$. The Kraus elements are given by

$$\hat{F}_0 \doteq \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix} , \quad \hat{F}_1 \doteq \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix} . \tag{5.51}$$

They are already orthogonal to each other.

Depolarizing In the depolarizing process, the decoherence occurs symmetrically, and there is no distinction of the specific types of actual decoherence process. The system undergoes an incoherent process with probility p whereas it reamins intact with probability $1-p$. The incoherent process may cause the system to flip the bit value, the phase, or both with equal probability.

The situation can be best described in the unitary representation. Suppose that the system and the environment is in the product state $|\psi\rangle \otimes |\varepsilon_0\rangle$. The decoherence process causes the transition

$$\begin{aligned} |\psi\rangle \otimes |\varepsilon_0\rangle \mapsto |\psi\rangle \otimes |0\rangle \sqrt{1-p} \\ + \left(\hat{X} |\psi\rangle \otimes |\varepsilon_1\rangle + \hat{Y} |\psi\rangle \otimes |\varepsilon_2\rangle + \hat{Z} |\psi\rangle \otimes |\varepsilon_3\rangle \right) \sqrt{\frac{p}{3}} \end{aligned} \tag{5.52}$$

The environment evolves to one of the four mutually orthogonal states. The final states of the environment enables to recognize what process (bit flip, phase flip, or both) has occurred, and hence cause the decoherence on the state of the system.

From the above unitary representation, we can get the Kraus elements

$$\hat{F}_0 = \sqrt{1-p}\hat{I} \,, \quad \hat{F}_1 = \sqrt{\frac{p}{3}}\hat{X} \,, \quad \hat{F}_2 = \sqrt{\frac{p}{3}}\hat{Y} \,, \quad \hat{F}_3 = \sqrt{\frac{p}{3}}\hat{Z} \,. \tag{5.53}$$

The Kraus elements are already orthogonal to each other. One can also check that they satisfy the completeness relation

$$\sum_{\mu=0}^{3} \hat{F}_\mu^\dagger \hat{F}_\mu = \hat{I} \tag{5.54}$$

as they should. In the Kraus representation with the above Kraus elements, a density operator $\hat{\rho}$ is transformed under the decoherence process as

$$\hat{\rho} \mapsto (1-p)\hat{\rho} + \frac{p}{3} \left(\hat{X}\hat{\rho}\hat{X} + \hat{Y}\hat{\rho}\hat{Y} + \hat{Z}\hat{\rho}\hat{Z} \right) . \tag{5.55}$$

5.2 Generalized Measurements as Quantum Operations

This subsection provides a reminder of the generalized measurement, this time, in terms of the quantum operations formalism.

5.3 Quantum Master Equation

Consider an open quantum system, interacting with its environment. The system is inevitably subject to decoherence processes. Suppose that the system is in $\hat{\rho}(t)$ at time t. To understand the decoherence processes, we want to examine the state $\hat{\rho}(t')$ at later times $t' > t$. The evolution from $\hat{\rho}(t)$ to $\hat{\rho}(t')$ is described by a quantum operation—in this case, a completely positive and trace-preserving superoperator. The operator-sum representation (5.12) guarantees the existence of operators $\hat{F}_\mu(t', t)$ such that

$$\hat{\rho}(t') = \sum_\mu \hat{F}_\mu(t', t)\hat{\rho}(t)\hat{F}_\mu^\dagger(t', t)\,, \tag{5.56a}$$

and satisfying the probability-conserving—trace-preserving—condition

$$\sum_\mu \hat{F}_\mu^\dagger(t', t)\hat{F}_\mu(t', t) = \hat{I} \tag{5.56b}$$

and the orthogonality condition

$$\mathrm{Tr}\,\hat{F}_\mu^\dagger\hat{F}_\nu = 0 \quad (\mu \neq \nu). \tag{5.56c}$$

However, it turns out that under a specific physical situation, mostly it is difficult to figure out the relevant operators $\hat{F}_\mu(t', t)$ properly describing the given situation. It may be because the approach attempts to directly determine $\hat{\rho}(t')$ as a function of time t' given the initial condition set by $\hat{\rho}(t)$. It would be more convenient and efficient to express the process in a differential form—a rate equation. After all, both Newton's classical equation of motion and Schrödinger's equation for quantum states are differential equations, describing the rate of changes in the state variables.

Can one express an quantum operation in a set of differential equation that is equivalent to the operator-sum representation? Unfortunately, the answer is "No," in general. However, under many physically relevant conditions,[11] the operators $\hat{F}_\mu(t', t)$ depend only on the time span $\delta t := t' - t$ but not on the individual instances t' and t. Physically, it implies that the underlying process does not depend on the history, and the assumption is commonly called the *Markov approximation*. Under

[11] A notable exception is the case where time-dependent external fields are applied on the system.

such conditions, the quantum operation in (5.56) can reformulated in a differential form and the resulting equation,

$$\frac{d\hat{\rho}}{dt} = \mathscr{L}(\hat{\rho})\,, \tag{5.57}$$

is called the *Lindblad equation* or *quantum master equation*. Here the superoperator $\mathscr{L}$ defined by

$$\mathscr{L}(\hat{\rho}) := -i[\hat{H}, \hat{\rho}] + \sum_{\mu} \left(\hat{L}_\mu \hat{\rho} \hat{L}_\mu^\dagger - \frac{1}{2} \hat{L}_\mu^\dagger \hat{L}_\mu \hat{\rho} - \frac{1}{2} \hat{\rho} \hat{L}_\mu^\dagger \hat{L}_\mu \right)\,, \tag{5.58}$$

generates the *quantum Markovian dynamics*, and is called the *Lindblad generator*. The Hermitian operator $\hat{H}$ in (5.58) describes the unitary part of the dynamics. For this reason, $\hat{H}$ is often called the *effective Hamiltonian* of the system, but in general it is not the same as the Hamiltonian when the system is isolated. The operators $\hat{L}_\mu$ in (5.58) are responsible for non-unitary dynamics and called the *Lindblad operators* or *quantum jump operators*.

It is also customary to rewrite the Lindblad generator (5.58) into the form

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}, \hat{\rho}] - \{\hat{G}, \hat{\rho}\} + \sum_{\mu} \hat{L}_\mu \hat{\rho} \hat{L}_\mu^\dagger, \tag{5.58$'$}$$

where

$$\hat{G} := \frac{1}{2} \sum_{\mu} \hat{L}_\mu^\dagger \hat{L}_\mu\,. \tag{5.59}$$

Interestingly, ignoring the last term in the quantum jump operators in (5.58′), the solution to the Lindblad equation (5.58′) is simply given by

$$\hat{\rho}(t) = e^{-it\hat{H}_{\rm non}} \hat{\rho}(0) e^{it\hat{H}_{\rm non}^\dagger}\,, \tag{5.60}$$

which resembles the unitary dynamics in (1.33) with the Hamiltonian $\hat{H}$ replaced with the effective *non-Hermitian Hermitonian*

$$\hat{H}_{\rm non} := \hat{H} - i\hat{G}. \tag{5.61}$$

The additional term in $\hat{G}$ of the non-Hermitian Hamiltonian makes a significant difference in the evolution governed by Eq. (5.60) as it causes damping—the irreversible population loss in the eigenstates of $\hat{H}$. In this sense, we call $\hat{G}$ the *effective damping operator*. Although the non-Hermitian Hamiltonian approach does not explain all decoherence processes, it lays out an intuitively appealing picture of open systems and is widely used to describe the effects of finite life time of (effective) energy levels. The non-Hermitian Hamiltonian also provides a good starting point for various more elaborate methods to investigate decoherence processes. A common example is the so-called *quantum jump approach*. It is an approximate method to solve the Lindblad equation combining the non-unitary

evolution in (5.60) due to the non-Hermitian Hamiltonian and the "quantum jumps" due to the quantum jump operators $\hat{L}_\mu$ (Dum *et al.*, 1992; Plenio & Knight, 1998).

The choice of the Lindblad operators $\hat{L}_\mu$ and the effective Hamiltonian $\hat{H}$ is not unique (Breuer & Petruccione, 2002): First, the two sets of Lindblad operators $\{\hat{L}_\mu\}$ and $\{\hat{L}'_\nu\}$ give the same Lindblad equation when

$$\hat{L}'_\nu = \sum_\mu \hat{L}_\mu U_{\mu\nu} \,, \tag{5.62}$$

where U is a unitary matrix—recall similar unitary freedom for the choice of the Kraus operators in the specification of quantum operations—see Eqs. (5.30) and (5.31)—as well as for the choice of pure states in the specification of mixed states—see Eqs. (1.11) and (1.12). Thanks to the unitary freedom, one can always choose the quantum jump operators to be *mutually orthogonal*,

$$\operatorname{Tr} \hat{L}_\mu^\dagger \hat{L}_\nu = 0 \quad (\mu \neq \nu) \tag{5.63}$$

for all μ and ν. Such a choice is optimal in the sense that $(d^2 - 1)$ quantum jump operators, where $d := \dim \mathcal{V}$, is sufficient for any Lindblad equation. The unitary freedom in (5.62) inherits from the unitary freedom for the choice of the Kraus elements in (5.31). The proof is left for an exercise. Second, the Lindblad generator is also invariant under the inhonogeneous transformations

$$\hat{L}_\mu \to \hat{L}'_\mu = \hat{L}_\mu + a_\mu \,, \tag{5.64a}$$

$$\hat{H} \to \hat{H}' = \hat{H} + \frac{1}{2i} \sum_\mu (a_\mu^* \hat{L}_\mu - a_\mu \hat{L}_\mu^\dagger) + b \tag{5.64b}$$

for any $a_\mu \in \mathbb{C}$ and $b \in \mathbb{R}$. Due to the translational freedom, it is always possible to choose the Lindblad operators to be *traceless*, $\operatorname{Tr} \hat{L}_\mu = 0$. Furthermore, for a given Lindblad equation, it is common to impose the condition $\operatorname{Tr} \hat{H} = 0$ on the effective Hamiltonian $\hat{H}$ to make it unique. It is straightforward to prove the translational freedom, and again left for an exercise.

As we have pointed out concerning the unitary freedom in the Kraus representation, the unitary freedom does not necessarily imply that one can exploit it to change a given set of Lindblad operators to any arbitrary preferred set of Lindblad operators. This is because the Lindblad operators in (5.62) are not normalized. A notable exception is the two-dimensional case—see Section 5.3.2.

In the remaining of the section, we derive the quantum master equation (5.58) and discuss methods to solve it.

5.3.1 Derivation

It is straightforward to derive the Lindblad equation (5.58) starting from the Kraus representation (5.12) under the Markov assumption—see Breuer & Petruccione

(2002) for example. Here we will take a heuristic approach, which is more useful to understand the underlying physics:

As $t' \to t$ ($\delta t \to 0$), it is physically required that $\hat{\rho}(t') \to \hat{\rho}(t)$. It implies that one and only one of $\hat{F}_\mu(\delta t)$ must approach $\hat{I}$. Let us denote it by $\hat{F}_0(\delta t)$. Up to the first order in δt,

$$\hat{F}_0(\delta t) \approx \hat{I} + \hat{L}_0\, \delta t\,. \tag{5.65}$$

The rest should vanish $\hat{F}_\mu(\delta t) \to 0$ for any $\mu > 0$. Since we physically expect that $\hat{\rho}(t') \approx \hat{\rho}(t) + \mathcal{O}(\delta t)$, $\hat{F}_\mu(\delta t)$ must vanish like $\sqrt{\delta t}$ with δt so that $\hat{F}_\mu(\delta t)\hat{\rho}\hat{F}_\mu^\dagger(\delta) \to \delta t$. We put

$$\hat{F}_\mu(\delta t) \approx \hat{L}_\mu \sqrt{\delta t} \quad (\mu > 0)\,. \tag{5.66}$$

As $\hat{L}_\mu$ ($\mu > 0$) directly proportional to $\hat{F}_\mu$, they are all traceless and mutually orthogonal—$\mathrm{Tr}\, \hat{L}_\mu^\dagger \hat{L}_\nu = 0$ for $\mu \neq \nu$. The probability conservation condition, Eq. (5.56b), implies that

$$\hat{L}_0 + \hat{L}_0^\dagger = -\sum_{\mu \neq 0} \hat{L}_\mu^\dagger \hat{L}_\mu\,. \tag{5.67}$$

It suggests that it will be convenient to split $\hat{L}_0$ into the Hermitian and anti-Hermitian part

$$\hat{L}_0 = -\hat{G} - i\hat{H}\,, \tag{5.68}$$

where the Hermitian part $\hat{G}$ is fixed by the operators $\hat{L}_\mu$ with the relation ($\mu > 0$)

$$\hat{G} = \frac{1}{2} \sum_{\mu=1}^{N^2-1} \hat{L}_\mu^\dagger \hat{L}_\mu\,. \tag{5.69}$$

whereas $\hat{H}$ remains arbitrary—only determined by $\hat{F}_0$, which is linearly independent of $\hat{L}_\mu$ ($\mu > 0$). Finally, putting Eqs. (5.65), (5.66), (5.68), and (5.69) into Eq. (5.56) leads to the desired equation (5.58) or, equivalently, to (5.58). Note that in this particular derivation, the Lindblad operators $\hat{L}_\mu$ turn out to be traceless and mutually orthogonal automatically without exploiting the unitary freedom in (5.62)—here the properties inherit from the orthogonality (5.56c) of the Kraus elements.

As mentioned at the beginning of the section, in practice it is difficult to explicitly figure out the quantum operations $\hat{\rho}(t) \mapsto \hat{\rho}(t')$ as a function of time. More common approach is to derive the Lindblad equation, often approximately, by reducing the unitary dynamics of the total system of system plus environment (see also Section 5.1.2). Some examples including perturbative methods are discussed in Breuer & Petruccione (2002).

5.3.2 Examples

Phase damping The Lindblad equation for the phase damping process can be obtained from the Kraus elements in (5.48). We assume that the probability p for the process to occur is proportional to time t, $p = \gamma t$, where γ is the rate of the process per unit time. Expanding the Kraus elements for small t,

$$\hat{F}_0 \approx \hat{I} - \frac{\gamma}{8}\hat{I}\,. \tag{5.70}$$

According to (5.68), we identify the effective Hamiltonian and damping operator with

$$\hat{H} = 0\,, \quad \hat{G} = \frac{\gamma}{8}\hat{I}\,, \tag{5.71}$$

respectively. Further,

$$\hat{F}_1 \approx \frac{\sqrt{\gamma t}}{2}\hat{Z} \tag{5.72}$$

implies that there is one Lindblad operator

$$\hat{L}_1 = \frac{\sqrt{\gamma}2}{\hat{Z}}\,. \tag{5.73}$$

Overall, the Lindblad generator for the phase damping process is given by

$$\mathscr{L}(\hat{\rho}) = \frac{\gamma}{8}\hat{\rho} + \frac{\gamma}{4}\hat{Z}\hat{\rho}\hat{Z}\,. \tag{5.74}$$

Amplitude damping The Kraus elements for the amplitude damping process are given in (5.51). In the infinitesimal time t,

$$\hat{F}_0 \approx I - \frac{\gamma}{2}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\,, \quad \hat{F}_1 \approx \sqrt{\gamma}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{5.75}$$

Therefore, while the effective Hamiltonian $\hat{H}$ vanishes, the effective damping operator $\hat{G}$ and the Lindblad operator are given by

$$\hat{G} = \frac{\gamma}{4}(1 - \hat{S}^z)\,, \quad \hat{L}_1 = \sqrt{\gamma}\hat{S}^+ \tag{5.76}$$

The Lindblad generator for the amplitude damping is given by

$$\mathscr{L}(\hat{\rho}) = \frac{\gamma}{4}\left\{(1 - \hat{S}^z)\hat{\rho} + \hat{\rho}(1 - \hat{S}^z)\right\} + \gamma\hat{S}^+\hat{\rho}\hat{S}^- \tag{5.77}$$

Depolarizing The Kraus elements for the depolarizing process have been obtained in (5.53). Again, assuming $p = \gamma t$ and expanding the Kraus elements for small t give

$$\hat{F}_0 \approx \hat{I} - \frac{\gamma t}{2}\hat{I}\,, \quad \hat{F}_\mu = \sqrt{\frac{\gamma t}{3}}\hat{S}^\mu \quad (\mu = x, y, z). \tag{5.78}$$

Therefore, the Lindblad generator for the depolarizing process is given by

$$\mathscr{L}(\hat{\rho}) = \frac{\gamma}{2}\hat{\rho} + \frac{\gamma}{3}\sum_{\mu=x,y,z}\hat{S}^\mu\hat{\rho}\hat{S}^\mu\,. \tag{5.79}$$

General damping For a single qubit, any master equation can be put into the form

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}, \hat{\rho}] - \{\hat{G}, \hat{\rho}\} + \Gamma_+ \hat{S}^+ \hat{\rho} \hat{S}^- + \Gamma_- \hat{S}^- \hat{\rho} \hat{S}^+ + \Gamma_0 \hat{S}^z \hat{\rho} \hat{S}^z \,, \tag{5.80a}$$

where the effective Hamiltonian $\hat{H}$ is arbitrary as longas as it is Hermitian, the effective damping operator is given by

$$\hat{G} := \frac{\Gamma_- + \Gamma_+ 2\Gamma_0}{4} + \left(\frac{\Gamma_- - \Gamma_+}{4} \right) \hat{S}^z \,, \tag{5.80b}$$

the real positive parameters $\Gamma_\pm, \Gamma_0$ are the rates at which the decoherence processes associated with the quantum jump operators $\hat{S}^\pm$ and $\hat{S}^z$ occur. In this form, the quantum jump operators, $\hat{S}^\pm$ and $\hat{S}^z$, describe the "simple" transitions—no mixture of different transitions—between the fixed set of states $|0\rangle$ and $|1\rangle$.[12] In general, those states are not the eigenstates of the effective Hamiltonian $\hat{H}$ in the coherent part of the master equation.

To see that the form in Eq. (5.80) is the most general form of the quantum master equation for a single-qubit system, let us start from the general Lindblad equation

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}, \hat{\rho}] - \{\hat{G}, \hat{\rho}\} + \sum_{\mu=1}^{3} \gamma_\mu \hat{A}_\mu \hat{\rho} \hat{A}_\mu^\dagger \,, \tag{5.81}$$

where

$$\hat{G} := \frac{1}{2} \sum_\mu \gamma_\mu \hat{A}_\mu^\dagger \hat{A}_\mu \,. \tag{5.82}$$

The three Lindblad operators are *traceless* and *orthonormal* and $\gamma_\mu \geq 0$. Consider another set of orthonormal operators

$$\hat{L}_1 = \hat{S}^+ \,, \quad \hat{L}_2 = \hat{S}^- \,, \quad \hat{L}_3 = \frac{1}{\sqrt{2}} \hat{S}^z \,. \tag{5.83}$$

As both sets $\{\hat{A}_\mu\}$ and $\{\hat{L}_\mu\}$ are orthonormal, there exits a unitary matrix U such that

$$\hat{A}_\nu = \sum_\mu \hat{L}_\mu U_{\mu\nu} \tag{5.84}$$

for all $\nu = 1, 2, 3$. In turn, this implies that there exists a unitary operator $\hat{U} \in \mathcal{L}(\mathcal{V})$ such that

$$\hat{A}_\nu = \hat{U} \hat{L}_\nu \hat{U}^\dagger = \sum_\mu \hat{L}_\mu U_{\mu\nu} \tag{5.85}$$

[12] $\hat{S}^z$ makes a transition to the same state, but it induces different phase factors depending on the states.

for all $\nu = 1, 2, 3$. Putting (5.85) into the Kraus representation (5.81),

$$\hat{U}^\dagger \frac{d\hat{\rho}}{dt} \hat{U} = -i\hat{U}^\dagger[\hat{H}, \hat{\rho}]\hat{U} - \hat{U}^\dagger\{\hat{G}, \hat{\rho}\}\hat{U} + \sum_\mu \gamma_\mu \hat{L}_\mu^\dagger \hat{U}^\dagger \hat{\rho} \hat{U} \hat{L}_\mu^\dagger. \tag{5.86}$$

Finally, redefining the operators as[13]

$$\hat{\rho}' := \hat{U}^\dagger \hat{\rho} \hat{U}\,, \quad \hat{H}' := \hat{U}^\dagger \hat{\rho} \hat{U}\,, \quad \hat{G}' := \hat{U}^\dagger \hat{G} \hat{U} \tag{5.87}$$

and rescalingt the parameters as

$$\Gamma_+ := \gamma_1\,, \quad \Gamma_- := \gamma_2\,, \quad \Gamma_- := \frac{1}{2}\gamma_3 \tag{5.88}$$

one arrives at the Lindblad equation of the form in (5.80) for $\hat{\rho}'$. Since $\hat{U}$ is a unitary transformation, it is nothing but a basis change, and $\hat{\rho}$ and $\hat{\rho}'$ are essentially the same. After solving the Lindblad equation, one can easily get $\hat{\rho}$ by applying the inverse transformation. In this sense, the Lindblad equation in (5.80) is the most general form for a single-qubit system.

5.3.3 Solution Methods

The Lindblad equation is a linear equation without explicit time dependence, and can always be solved by means of common methods for linear differential equations. More explicitly, in the standard basis, the Lindblad equation can be written as

$$\dot{\rho}_{jk} = \sum_{j'k'} M_{jk;j'k'} \rho_{j'k'}\,, \tag{5.89}$$

with

$$M_{jk;j'k'} := -i(H_{jj'}\delta_{kk'} - \delta_{jj'}H^*_{kk'}) - (G_{jj'}\delta_{kk'} + \delta_{jj'}G^*_{kk'}) + \sum_\mu L_{\mu;jj'} L^*_{\mu;kk'} \tag{5.90}$$

Regarding $\mu := (jk)$ and $\nu := (j'k')$ as collective indices, Eq. (5.89) reads as

$$\dot{\rho}_\mu = \sum_\nu M_{\mu\nu}\rho_\nu, \tag{5.91}$$

which is a typical first-order differential equation for the column vector ρ_μ.

Technically, the differential equation (5.91) is not adequate yet to solve because of the conditions that $\rho_{jk} = \rho^*_{kj}$ and that $\sum_j \rho_{jj} = 1$. The latter conditon is reflected in the fact that the determinant of the matrix M is always zero. For example, consider a single-qubit system. Suppoe that the Lindblad equation is characterized by the effective Hamiltonian

$$\hat{H} = \frac{1}{2}\Omega\hat{S}^z + \frac{1}{2}\Delta\hat{S}^x \tag{5.92}$$

[13] Note that $\hat{G}'$ equals to the expression in Eq. (5.80b).

and the Lindblad operators $\sqrt{\Gamma_\pm}\hat{S}^\pm$. In the matrix form, the Lindlblad equation is given by

$$\frac{d}{dt}\begin{bmatrix}\rho_{11}\\ \rho_{12}\\ \rho_{21}\\ \rho_{22}\end{bmatrix} = \begin{bmatrix} -\Gamma_- & \frac{i\Delta}{2} & -\frac{i\Delta}{2} & \Gamma_+ \\ \frac{i\Delta}{2} & -\frac{\Gamma_-}{2}-\frac{\Gamma_+}{2}-i\Omega & 0 & -\frac{i\Delta}{2} \\ -\frac{i\Delta}{2} & 0 & -\frac{\Gamma_-}{2}-\frac{\Gamma_+}{2}+i\Omega & \frac{i\Delta}{2} \\ \Gamma_- & -\frac{i\Delta}{2} & \frac{i\Delta}{2} & -\Gamma_+ \end{bmatrix}\begin{bmatrix}\rho_{11}\\ \rho_{12}\\ \rho_{21}\\ \rho_{22}\end{bmatrix} \tag{5.93}$$

Imposing the conditions, $\rho_{11}+\rho_{22}=0$ and $\rho_{12}=\rho_{21}^*$, the above equation reads as

$$\frac{d}{dt}\begin{bmatrix}\rho_{11}\\ \mathrm{Re}\,\rho_{21}\\ \mathrm{Im}\,\rho_{21}\end{bmatrix} = \begin{bmatrix} -\Gamma & 0 & 0 \\ 0 & -\Gamma/2 & -i\Gamma/2 \\ -i\Delta & -i\Omega & -\Omega \end{bmatrix}\begin{bmatrix}\rho_{11}\\ \mathrm{Re}\,\rho_{21}\\ \mathrm{Im}\,\rho_{21}\end{bmatrix} + \begin{bmatrix}\Gamma_+\\ 0\\ \Delta/2\end{bmatrix}, \tag{5.94}$$

where $\Gamma := \Gamma_+ + \Gamma_-$. It is a typical inhomogeneous first-order differential equation and can be solved using the standard methods. If necessary, various numerical methods can also be applied. This method is extensively discussed in Blum (2012).

In the above discussion, the constraints $\hat{\rho}^\dagger = \hat{\rho}$ and $\mathrm{Tr}\,\hat{\rho}=1$ have been handled in an ad hoc fashion. They can be dealt with in a systematic way by choosing an appropriate orthonormal basis $\{\hat{B}_\mu : \mu = 0,1,2,\ldots,n^2-1\}$ for $\mathcal{L}(\mathcal{V})$, where n is the dimension of the vector space $\mathcal{V}$, such that (i) $\hat{B}_0 = \hat{I}/\sqrt{n}$ and (ii) $\hat{B}_\mu^\dagger = \hat{B}_\mu$. Note that the condition (i) implies that the rest elements of the basis are all traceless—$\mathrm{Tr}\,\hat{B}_\mu = 0$ for $\mu = 1,2,\ldots,n^2-1$. Such a basis is called a *Lindblad basis*. For example, the following operators form a Lindblad basis for a three-level atom:

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{6}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & -\sqrt{\frac{2}{3}} \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$\begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 \end{bmatrix},$$
$$\begin{bmatrix} 0 & -\frac{i}{\sqrt{2}} & 0 \\ \frac{i}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -\frac{i}{\sqrt{2}} \\ 0 & 0 & 0 \\ \frac{i}{\sqrt{2}} & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{i}{\sqrt{2}} \\ 0 & \frac{i}{\sqrt{2}} & 0 \end{bmatrix}. \tag{5.95}$$

Let us examine how various quantities are represented in a Lindblad basis: The components of a density operator $\hat{\rho}$ in the expansion

$$\hat{\rho} = \sum_{\mu=0}^{n^2-1} \hat{B}_\mu \rho_\mu \tag{5.96}$$

are given by $\rho_\mu := \mathrm{Tr}\, \hat{B}_\mu \hat{\rho}$, and they are all real, and in particular, $\rho_0 = 1/\sqrt{n}$. More importantly, the Lindblad equation nows reads as

$$\dot{\rho}_\mu = \sum_{\nu=1}^{n^2-1} K_{\mu\nu}\rho_\nu + b_\mu \quad (\mu = 1, 2, \ldots, n^2-1)\,, \tag{5.97}$$

with the generator matrix K given by

$$K_{\mu\nu} := \mathrm{Tr}\, \hat{B}_\mu^\dagger \mathscr{L}(\hat{B}_\nu)\,, \tag{5.98}$$

where $\mathscr{L}$ is the Lindblad generator in (5.58) or (5.58′), and the inhomogeneous term given by

$$b_\mu := \mathrm{Tr}\, \hat{B}_\mu^\dagger \mathscr{L}(\hat{B}_0)/\sqrt{n}\,. \tag{5.99}$$

The inhomogeneous differential equation (5.97) has the solution of the form

$$\rho_\mu(t) = \sum_{\nu=1}^{n^2-1} \left[e^{Kt}\right]_{\mu\nu} \rho_\nu(0) + \sum_\nu \left[\int_0^t ds\, e^{Ks}\right]_{\mu\nu} b_\nu\,. \tag{5.100}$$

Let us consider a single qubit, and examine a master equation. We assume that the system is initially prepared in a pure state $\left(|0\rangle + |1\rangle\right)/\sqrt{2}$.

```
In[•]:= init = (1 + S[1]) / 2;
        init // MatrixForm
Out[•]//MatrixForm=
        1/2 (1 + S^x)
```

This is the effective Hamiltonian.

```
        opH = S[3];
```

These are the Lindblad operators.

```
In[•]:= Let[Real, Γ]
        opL = {Sqrt[Γ["+"]] × S[4], Sqrt[Γ["-"]] × S[5]}
Out[•]= {S^+ √Γ_+ , S^- √Γ_- }
```

This is the Lindblad basis we are going to use. It happens to be equivalent to the basis consisting of the Pauli operators.

```
In[•]:= lbs = LindbladBasis[S]
        MatrixForm /@ (Matrix[#, S] &) /@ lbs
Out[•]= {1/√2, S^z/√2, S^x/√2, S^y/√2}
Out[•]= {((1/√2  0), (0  1/√2)), ((1/√2  0), (0  -1/√2)), ((0  1/√2), (1/√2  0)), ((0  -i/√2), (i/√2  0))}
```

Here are the generator matrix K and the inhomogeneous term when the Lindblad equation is written in the standard form of an inhomogeneous first-order differential equation.

```
In[•]:= {mat, vec} = LindbladConvert[opH, opL];
        mat // MatrixForm
        vec // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -\Gamma_- - \Gamma_+ & 0 & 0 \\ 0 & \frac{1}{2}\left(-2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) + \frac{1}{2}\left(2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) & -\frac{1}{2}\,i\left(-2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) + \frac{1}{2}\,i\left(2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) \\ 0 & \frac{1}{2}\,i\left(-2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) - \frac{1}{2}\,i\left(2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) & \frac{1}{2}\left(-2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) + \frac{1}{2}\left(2\,i - \frac{\Gamma_-}{2} - \frac{\Gamma_+}{2}\right) \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{-\Gamma_- + \Gamma_+}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix}$$

This solves the differential equation based on the generator matrix K and the inhomogeneous term.

```
In[•]:= Clear[ρ]
        ρ[t_] = Block[
            {Γ, ρ},
            Γ["+"] = .3;
            Γ["-"] = .7;
            LindbladSolve[opH, opL, init, t] // Chop
          ];
        ρ[t] // MatrixForm
```

Out[•]//MatrixForm=

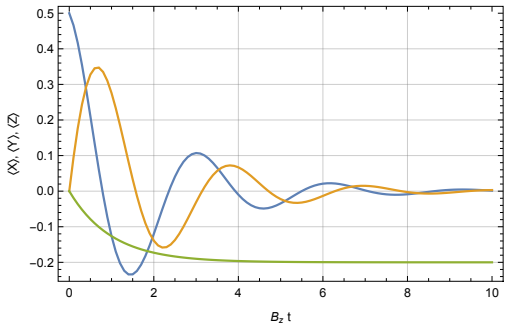
$$0.5 + 0.5\,e^{-0.5\,t}\,\text{Cos}[2.\,t]\,S^x + \left(-0.2 + 0.2\,e^{-1.\,t}\right)S^z + 0.5\,e^{-0.5\,t}\,S^y\,\text{Sin}[2.\,t]$$

To investigate the physical properties of the solution, we calculate the expectation values of the Pauli operators.

```
In[•]:= {avgX[t_], avgY[t_], avgZ[t_]} = Coefficient[ρ[t], #] & /@ S@{1, 2, 3}
Out[•]= {0.5 e^-0.5 t Cos[2. t], 0.5 e^-0.5 t Sin[2. t], -0.2 + 0.2 e^-1. t}

In[•]:= data = Transpose@Table[
            {{t, avgX[t]}, {t, avgY[t]}, {t, avgZ[t]}},
            {t, 0, 10, 0.1}
           ];
        ListLinePlot[data,
         FrameLabel → {"B_z t", "⟨X⟩, ⟨Y⟩, ⟨Z⟩"}
        ]
```

Out[•]=



Consider a three-level atom with the Λ-type level structure.

```
Let[Qudit, A]
```

In the interaction picture, the Hamiltonian looks like this. We have put the two Rabi transition amplitudes to 1.

```
In[•]:= opH = (1 / 10) A[1 → 1] + (2 / 10) A[2 → 2] + A[2 → 0] + A[0 → 2] + A[2 → 1] + A[1 → 2];
        matH = Matrix[opH];
        matH // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & \frac{1}{10} & 1 \\ 1 & 1 & \frac{1}{5} \end{pmatrix}$$

```
In[•]:= Let[Real, Γ]
        opL = {
          Sqrt[Γ[0, "-"]] * A[2 → 0],
          Sqrt[Γ[0, "+"]] * A[0 → 2],
          Sqrt[Γ[1, "-"]] * A[2 → 1],
          Sqrt[Γ[1, "+"]] * A[1 → 2]}
        matL = Matrix[opL];
        MatrixForm /@ matL
```

Out[•]= $\{(|0\rangle\langle 2|)\sqrt{\Gamma_{0,-}}, (|2\rangle\langle 0|)\sqrt{\Gamma_{0,+}}, (|1\rangle\langle 2|)\sqrt{\Gamma_{1,-}}, (|2\rangle\langle 1|)\sqrt{\Gamma_{1,+}}\}$

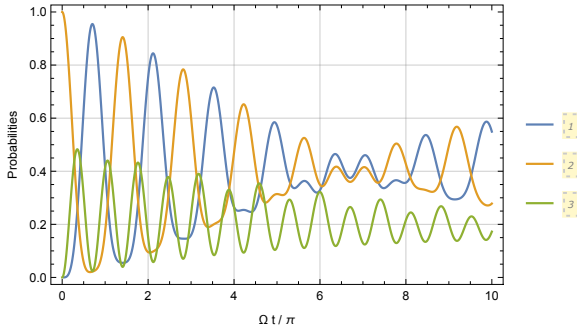
Out[•]= $\left\{\begin{pmatrix} 0 & 0 & \sqrt{\Gamma_{0,-}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \sqrt{\Gamma_{0,+}} & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \sqrt{\Gamma_{1,-}} \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \sqrt{\Gamma_{1,+}} & 0 \end{pmatrix}\right\}$

```
In[•]:= Timing[
         ρ[t_] = Block[
            {Γ, init},
            Γ[0, "-"] = 0.05;
            Γ[0, "+"] = 0.01;
            Γ[1, "-"] = 0.025;
            Γ[1, "+"] = 0.005;
            init = DiagonalMatrix[{0, 1, 0}];
            LindbladSolve[matH, matL, init, t]
           ];
        ]
```

Out[•]= {2.99647, Null}

```
In[•]:= Plot[Evaluate@Diagonal@ρ[t Pi], {t, 0, 10},
         FrameLabel → {"Ω t / π", "Probabilities"},
         PlotRange → All,
         PlotLegends → Automatic]
```

Out[•]=



There are two drawbacks in the above approaches: First, the size of the generator matrix K increases exponentially with the system size, and in many

cases, even numerical methods become intractable. In such cases, the quantum jump approach mentioned earlier—Section 5.3—is often used. Second, the resulting solution is given in an explicit matrix representation, and putting the solution in the Kraus representation is tedious. Therefore, the physical interpretation of the solution is not always clear.

Interestingly, there is a special class of Lindblad equations that allow for a solution directly in the Kraus representation (Nakazato *et al.*, 2006): Let $\{|j\rangle : j = 0, \ldots, n-1\}$ be the eigenbasis from the effective Hamiltonina $\hat{H}$ so that

$$\hat{H} = \sum_j E_j |j\rangle \langle j| \,. \tag{5.101}$$

We consider a Lindblad equation of the form

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}, \hat{\rho}] - \{\hat{G}, \hat{\rho}\} + \sum_{jk} \gamma_{jk} \hat{L}_{jk} \hat{\rho} \hat{L}_{jk}^\dagger \,, \tag{5.102}$$

where every (normalized) quantum jump operator $\hat{L}_{jk}$ corresponds to an incoherent transition between a pair of eigenstates, $\hat{L}_{jk} := |j\rangle \langle k|$, and γ_{jk} charaterizes the rate of the process. Note here that the effective damping operator $\hat{G}$,

$$\hat{G} := \frac{1}{2} \sum_{jk} \gamma_{jk} \hat{L}_{jk}^\dagger \hat{L}_{jk} = \frac{1}{2} \sum_k \gamma_k |k\rangle \langle k| \tag{5.103}$$

with

$$\gamma_k := \sum_j \gamma_{jk}, \tag{5.104}$$

commutes with the effective Hamiltonian $\hat{H}$.

To solve the Lindblad equation (5.102), we first recall that the non-unitary evolution in Eq. (5.60) governed by the non-Hermitian Hamiltonian in Eq. (5.61) corresponds to the solution in the absence of the quantum jump operators ($\gamma_{jk} = 0$). It is therefore natural to define the *generalized interaction picture*

$$\hat{\rho}_I(t) := e^{it\hat{H}_{\rm non}} \hat{\rho}(t) e^{-it\hat{H}_{\rm non}^\dagger} \tag{5.105}$$

with respect to the non-Hermitian Hamiltonian $\hat{H}_{\rm non}$. In this interaction picture, the Lindblad equation reads as

$$\frac{d\hat{\rho}_I}{dt} = \sum_{jk} \gamma_{jk} e^{(\gamma_j - \gamma_k)t} \hat{L}_{jk} \hat{\rho}_I \hat{L}_{jk}^\dagger = \sum_{jk} \gamma_{jk} \hat{R}_{jk}(t) \,, \tag{5.106}$$

where we have defined

$$\hat{R}_{jk}(t) := e^{(\gamma_j - \gamma_k)t} \hat{L}_{jk} \hat{\rho}_I(t) \hat{L}_{jk}^\dagger \,. \tag{5.107}$$

Since the differential equation (5.106) is equivalent to the intergral equation

$$\hat{\rho}_I(t) = \hat{\rho}_I(0) + \sum_{jk} \gamma_{jk} \int_0^t ds\, \hat{R}_{jk}(s)\,, \tag{5.108}$$

it is now a matter of calculating the new operator $\hat{R}_{jk}(t)$. It follows from (5.106) that the operator $\hat{R}_{jk}(t)$ satisfies the differential equation

$$\frac{d\hat{R}_{jk}}{dt} = \sum_l \Gamma^{(j)}_{kl} \hat{R}_{jl} \tag{5.109}$$

with the matrix $\Gamma^{(j)}$ defined by $\Gamma^{(j)}_{kl} := \delta_{kl}(\gamma_j - \gamma_k) + \gamma_{kl}$. The solution is given by

$$\hat{R}_{jk}(t) = \sum_l \left[e^{t\Gamma^{(j)}}\right]_{kl} \hat{R}_{jl}(0)\,. \tag{5.110}$$

Putting it back into the integral equation (5.108), we finally obtain

$$\hat{\rho}_I(t) = \hat{\rho}(0) + \sum_{jkl} \gamma_{jk} W^{(j)}_{kl}(t) \hat{L}_{jl} \hat{\rho}(0) \hat{L}^\dagger_{jl} \tag{5.111}$$

with

$$W^{(j)}(s) := \int_0^t ds\, e^{s\Gamma^{(j)}}. \tag{5.112}$$

More explicitly, for the purpose of illustration, the Kraus represenation of $\hat{\rho}(t)$ reads as

$$\hat{\rho}(t) = \hat{F}_0(t)\hat{\rho}(0)\hat{F}_0^\dagger(t) + \sum_{jk} \hat{F}_{jk}(t)\hat{\rho}(0)\hat{F}^\dagger_{jk}(t)\,, \tag{5.113a}$$

where the Kraus elements are given by

$$\hat{F}_0(t) := e^{-it\hat{H}_{\rm non}}\,, \quad \hat{F}_{jk}(t) := e^{-it\hat{H}_{\rm non}} |j\rangle \sqrt{\textstyle\sum_l \gamma_{jl} W^{(j)}_{lk}(t)}\, \langle k| \,. \tag{5.113b}$$

Let us discuss some examples. First, consider the following Lindblad equation for a single qubit:

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}, \hat{\rho}] - i\{\hat{G}, \hat{\rho}\} + \gamma_- |0\rangle \langle 1| \hat{\rho} |1\rangle \langle 0| + \gamma_1 |1\rangle \langle 1| \hat{\rho} |1\rangle \langle 1| \,, \tag{5.114}$$

where

$$\hat{H} = E_0 |0\rangle \langle 0| + E_1 |1\rangle \langle 1| \,, \quad \hat{G} = \frac{1}{2}(\gamma_- + \gamma_1) |1\rangle \langle 1| \,. \tag{5.115}$$

Apparently, γ_- and γ_1 are responsible for the amplitude and phase damping process, respectively, as discussed in Sections 5.1.3 and 5.3.2.

. . . . . .

. . . . . .

Next, consider an atom with a Λ-type level structure. Suppose that it is subject to an interaction to its environment, which is described by the following Lindblad equation

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}, \hat{\rho}] - i\{\hat{G}, \hat{\rho}\} + \gamma_0 \, |0\rangle \langle 2| \, \hat{\rho} \, |2\rangle \langle 0| + \gamma_1 \, |1\rangle \langle 2| \, \hat{\rho} \, |2\rangle \langle 1| \, , \tag{5.116}$$

where $|0\rangle$ and $|1\rangle$ are the ground-state levels and $|2\rangle$ is the excited state,

$$\hat{H} = E_1 \, |1\rangle \langle 1| + E_2 \, |2\rangle \langle 2| \quad (0 \leq E_1 \ll E_2) \, , \tag{5.117}$$

and

$$\hat{G} = \frac{1}{2}(\gamma_0 + \gamma_1) \, |2\rangle \langle 2| \, . \tag{5.118}$$

We have set the ground-state energy to zero $E_0 = 0$.

. . . . . .

. . . . . .

5.4 Fidelity and Trace Distance

5.5 Entanglement, Entropy, Mutual Information

Problems

1. By explicitly evaluating the quantum circuit model in (5.26), prove the quantum gate teleportation protocol.

2. Consider a quantum register of two qubits. Let $\mathscr{F}$ be a quantum operation on the quantum register, specified by the Kraus elements

$$\hat{F}_0 = \sqrt{1 - p_1 - p_2 - p_3}\hat{I} \, ,$$
$$\hat{F}_1 = \sqrt{p_1}\hat{S}_1^- \, , \quad \hat{F}_2 = \sqrt{p_2}\hat{S}_2^- \, , \quad \hat{F}_3 = \sqrt{p_3}\hat{S}_1^- \hat{S}_2^- \, . \tag{5.119}$$

   (a) Construct a quantum circuit model to generate the Choi operator $\hat{C}_{\mathscr{F}}$ associated with the quantum operation $\mathscr{F}$. Note that the input state of the system (including an auxiliary quantum register if necessary) must be $|0\rangle \equiv |0\rangle \otimes |0\rangle \otimes \cdots$.
   Hint: See the quantum circuit model representation in (5.19) to generate the Choi operator. You have to generate the maximally entangled state starting from $|0\rangle$.

   (b) Write down the Choi operator $\hat{C}_{\mathscr{F}}$ in terms of the Pauli operators $\hat{S}_1^\mu$ and $\hat{S}_2^\nu$ on the two qubits.

Chapter 6

Quantum Error Correction Codes: Introduction

6.1 Discretization of errors

6.2 9-Qubit Code (Shor's Code)

6.3 Fault-Tolerant Quantum Computation

6.4 CSS Code (Optional)

6.5 Stabilizer Code (Optional)

6.6 Surface Code (Optional)

Appendix A

Linear Algebra

• May 26, 2021 (v1.15)

Linear algebra is an elementary language to describe quantum mechanics mathematically. This appendix summarizes, without rigorous proofs, the concepts, definitions, theorems and properties of linear algebra that are frequently used in quantum information physics.

Many textbooks on linear algebra focus on arithmetic techniques related to properties of matrices, such as the Gauss elimination and the formula of determinant. In most areas of physics, notably quantum mechanics, more relevant are the fundamental concepts and algebraic structures of vector spaces and linear operators. Lang (1987)—an abridged edition Lang (1986) is also available—is one of the textbooks that introduce and discuss the latter subjects at a level adequate to physicists.

A.1 Vectors

A.1.1 Vector Space

Arguably, the most distinguished feature of quantum states compared with classical states is superposition inherited from the wave-particle duality. It is thus natural to describe quantum states mathematically by vectors. Vectors can be multiplied by numbers (called scalars) and added with each other, exactly the way the superposition principle dictates. We first need a field, a set of scalars with addition and multiplication.

Definition 1 (*field*) A set $\mathbb{F}$ of elements is called a *field* if it satisfies the following conditions:

(a) (addition) If $x, y \in \mathbb{F}$, then $x + y \in \mathbb{F}$.

(b) (multiplication) If $x, y \in \mathbb{F}$, then $xy \in \mathbb{F}$.

(c) (zero) $0 \in \mathbb{F}$ and $1 \in \mathbb{F}$.

(d) (inverse) If $x \in \mathbb{F}$, then $-x \in \mathbb{F}$.

(e) (inverse) If $x \in \mathbb{F}$ and $x \neq 0$, then $x^{-1} \in \mathbb{F}$.

The elements of the given field are called the *scalars*.

In the simplest terms, vectors represent physical quantities with both magnitude and direction. In quantum mechanics, more important feature of vectors is superposition. Here is the formal definition with mathematical rigor.

Definition 2 (*vector space*) A set $\mathcal{V}$ of elements is called a *vector space* over a field $\mathbb{F}$, if it satisfies the following conditions:

(a) If $v_1, v_2 \in \mathcal{V}$, then $v_1 + v_2 \in \mathcal{V}$.

(b) If $v_1 \in \mathcal{V}$ and $x \in \mathbb{F}$, then $xv_2 \in \mathcal{V}$.

(c) If $v_1, v_2, v_3 \in \mathcal{V}$, then $(v_1 + v_2) + v_3 = v_1 + (v_2 + v_3)$.

(d) There is an element $0 \in \mathcal{V}$ (a *null vector*) such that for all $v_2 \in \mathcal{V}$ $0 + v_2 = v_2 + 0 = v_2$. Note that both "zero" in $\mathbb{F}$ or the null vector in $\mathcal{V}$ are denoted by '0'.

(e) For a given $v_2 \in \mathcal{V}$, there exists $-v_2 \in \mathcal{V}$ such that $v_2 + (-v_2) = 0$. The subtraction between vectors are defined by $v_1 - v_2 := v_1 + (-v_2)$.

(f) For $x, y \in \mathbb{F}$ and $v_1, v_2 \in \mathcal{V}$,

$$x(v_1 + v_2) = xv_1 + xv_2 \,, \tag{A.1a}$$

$$(x + y)v_2 = xv_2 + yv_2 \,, \tag{A.1b}$$

$$(xy)v_2 = x(yv_2) \,. \tag{A.1c}$$

The elements of a vector space are called *vectors*.

Common examples include vector spaces over the fields of rational numbers ($\mathbb{Q}$), real numbers ($\mathbb{R}$), complex numbers ($\mathbb{C}$), and quarternions ($\mathbb{H}$). Note that the set of integer numbers ($\mathbb{Z}$) with the standard arithmetic rules is not a field. As is the case mostly in quantum mechanics, ***vector spaces will be assumed to be over the field of complex numbers*** $\mathbb{C}$ ***throughout this book unless mentioned otherwise.***

A.1.2 Hermitian Product

In the definition of vector space, the multiplication of vectors has not be defined. The *inner product* gives a special kind of multiplication between vectors and provides the vector space with a geometric structure, i.e., the orthogonality of vectors.

Inner product is usually a bilinear product. In many fields of physics (e.g., quantum mechanics), however, we will be dealing with vector spaces over $\mathbb{C}$ (the field of complex numbers). To preserve the notion of positive definiteness, we need to adopt a slightly different definition of inner product. This modified inner product is called a "Hermitian product" to distinguish it from a usual inner product.

Definition 3 (*Hermitian product*) Let $\mathcal{V}$ be a vector space. A *Hermitian product* on $\mathcal{V}$ is a function $\langle\cdot,\cdot\rangle$ from $\mathcal{V}\times\mathcal{V}$ to $\mathbb{C}$ satisfying the following conditions:

(a) For all $u, v, w \in \mathcal{V}$, $\langle u, v+w\rangle = \langle u, v\rangle + \langle u, w\rangle$.

(b) For all $z \in \mathbb{C}$ and $v, w \in \mathcal{V}$, $\langle zv, w\rangle = z^* \langle v, w\rangle$ and $\langle v, zw\rangle = z \langle v, w\rangle$.

(c) For all $v, w \in \mathcal{V}$, $\langle v, w\rangle = \langle w, v\rangle^*$.

(d) $\langle v, v\rangle \geq 0$ for all $v \in \mathcal{V}$, and $\langle v, v\rangle > 0$ if $v \neq 0$.[1]

The geometric structure due to the Hermitian product allows to define the magnitude of vectors which is important in the probabilistic interpretation of quantum mechanics (see Section 1.3). It also enables to quantify how close two state vectors are; see Section 5.4.

A.1.3 Basis

Definition 4 (*linear independence*) Let $\mathcal{V}$ be a vector space. The vectors $v_1, \cdots, v_n \in \mathcal{V}$ are said to be *linearly dependent* with each other if there exits a solution $z_1, \cdots, z_n \in \mathbb{C}$ to the equation

$$z_1 v_1 + \cdots + z_n v_n = 0\,. \tag{A.2}$$

If not, they are *linearly independent*.

Definition 5 (*basis*) Let $\mathcal{V}$ be a vector space. If every element of $\mathcal{V}$ is a linear combination of $v_1, \cdots, v_n$, then $v_1, \cdots, v_n$ are said to *span* (or *generate*) the vector space $\mathcal{V}$. The set $\{v_1, \cdots, v_n\} \subset \mathcal{V}$ is called a *basis* of $\mathcal{V}$ if $v_1, \cdots, v_n$ span $\mathcal{V}$ and are linearly independent. The number of elements in a basis of $\mathcal{V}$ is called the *dimension* of $\mathcal{V}$ and denoted by $\dim \mathcal{V}$.

Quantum states are described by a state vector in a Hilbert space. Hilbert space is a vector space, usually infinite dimensional, with additional analytic properties provided by the notion of completeness. However, as long as the dimension is finite, there is no distinction between Hilbert space and vector space. Unless mentioned otherwise explicitly, we assume that vector spaces are finite dimensional.

[1] This positive definiteness is included in the definition as it is required in most applications in quantum mechanics.

When a basis (recall Definition 5) of $\{v_1, v_2, \cdots, v_n\}$ spanning $\mathcal{V}$ satisfies

$$\langle v_i, v_j \rangle = \delta_{ij} \,, \tag{A.3}$$

it is called an *orthonormal basis*. For a finite dimensional vector space $\mathcal{V}$, one can always find an orthogonal basis as long as $\mathcal{V} \neq \{0\}$.

A choice of basis is arbitrary and one can change the basis to another: Suppose that $\mathcal{A} = \{v_1, v_2, \cdots, v_n\}$ and $\mathcal{B} = \{w_1, w_2, \cdots, w_n\}$ are two bases of the same vector space $\mathcal{V}$. As both $\mathcal{A}$ and $\mathcal{B}$ are bases, each vector $w_j \in \mathcal{B}$ is expanded in $v_i \in \mathcal{A}$ (and vice versa):

$$w_j = \sum_i v_i U_{ij} \,, \quad U_{ij} \in \mathbb{C} \,. \tag{A.4}$$

The matrix $U := [U_{ij}]$ composed of these coefficients characterizes the relation between the two bases, and must be invertible since the elements in each basis are linearly independent of each other. The relation is particularly simple when the bases are *orthonormal*: As $\mathcal{A}$ is orthonormal, the coefficients U_{ij} can be obtained by

$$U_{ij} = \langle v_i, w_j \rangle \,. \tag{A.5}$$

More importantly, one can show that U is a *unitary* matrix—see also Theorem 15.

A.1.4 Representations

Given a fixed basis $\{v_1, v_2, \cdots, v_n\}$ of a vector space $\mathcal{V}$, any vector $\alpha \in \mathcal{V}$ is *uniquely* specified by the coefficients $\alpha_j \in \mathbb{C}$ in the expansion

$$\alpha = v_1 \alpha_1 + v_2 \alpha_2 + \cdots v_n \alpha_n \,. \tag{A.6}$$

The column vector consisting of $\alpha_1, \ldots, \alpha_n$ is said to be the *representation* of the vector α in the basis, and denoted by

$$\alpha \doteq \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} . \tag{A.7}$$

When the basis $\{v_1, v_2, \cdots, v_n\}$ is *orthonormal*, the expansion coefficients α_j are obtained directly by means of the Hermitian product, $\alpha_j = \langle v_j, \alpha \rangle$. Hence, the vector is represented by the expansion

$$\alpha = \sum_j v_j \langle v_j, \alpha \rangle \tag{A.8}$$

or, equivalently, by the column vector

$$\alpha \doteq \begin{bmatrix} \langle v_1, \alpha \rangle \\ \langle v_2, \alpha \rangle \\ \vdots \\ \langle v_n, \alpha \rangle \end{bmatrix} . \tag{A.9}$$

Consider another vector $\beta \in \mathcal{V}$ and suppose that $\beta_j := \langle v_j, \beta \rangle$ is its representation in the same orthonormal basis. Then, the Hermitian product $\langle \alpha, \beta \rangle$ can be evaluated using their column-vector representations

$$\langle \alpha, \beta \rangle = \sum_j \alpha_j^* \beta_j = \begin{bmatrix} \alpha_1^* & \alpha_2^* & \cdots & \alpha_n^* \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} , \tag{A.10}$$

where we have used the identity $\langle \alpha, v_j \rangle = \langle v_j, \alpha \rangle^* = \alpha_j^*$.

Upon the change of basis, the represenations of vectors also change: Suppose that a vector $\alpha \in \mathcal{V}$ is represented by

$$\alpha \doteq \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{A.11}$$

in the basis $\{v_i\}$, and by

$$\alpha \doteq \begin{bmatrix} \alpha'_1 \\ \alpha'_2 \\ \vdots \\ \alpha'_n \end{bmatrix} \tag{A.12}$$

in the basis $\{w_j\}$. Obviously, the relation between the two representations is fixed by the relation between the two bases. Let us take a closer look at the relation when the two bases are orthonormal: We note from (A.8) that

$$\alpha'_k = \langle w_k, \alpha \rangle = \sum_{j=1}^{n} \langle w_k, v_j \rangle \langle v_j, \alpha \rangle = \sum_k U_{kj} \alpha_j \, , \tag{A.13}$$

where we have put $U_{kj} := \langle w_k, v_j \rangle$. We have seen in Eq. (A.5) that the matrix U is unitary. Therefore, we see that the representations in two different orthonormal bases are related by a unitary matrix as

$$\begin{bmatrix} \alpha'_1 \\ \alpha'_2 \\ \vdots \\ \alpha'_n \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ U_{21} & U_{22} & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n1} & U_{n2} & \cdots & U_{nn} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} . \tag{A.14}$$

A.2 Linear Operators

In quantum mechanics, the evolution of quantum states and the properties of physical quantities are described by linear operators. Linear operators are special kind of linear mappings.

A.2.1 Linear Maps

As already mentioned, the most important algebraic property of a vector space is the superposition. Therefore, the mapping preserving this property from one vector space to another plays an important role in the theory of linear algebra.

Definition 6 (*linear map*) Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces. A mapping (or simply map)

$$\hat{L} : \mathcal{V} \to \mathcal{W}\,, \quad v \mapsto \hat{L}v \tag{A.15}$$

is said to be *linear* if it satisfies the following two properties:

(a) For any $v, w \in \mathcal{V}$, we have $\hat{L}(v+w) = \hat{L}v + \hat{L}w$.

(b) For all $z \in \mathbb{C}$ and $v \in \mathcal{V}$ we have $\hat{L}(zv) = z(\hat{L}v)$.

When $\mathcal{V} = \mathcal{W}$, the map is called a *linear operator* on $\mathcal{V}$.

As a linear map preserves superposition, it is completely determined by specifying how it maps just the basis vectors. The following theorem just summarizes the property.

Theorem 7 Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces. Let $\{v_1, \cdots, v_n\}$ be a basis of $\mathcal{V}$, and $w_1, \cdots, w_n \in \mathcal{W}$ be arbitrary vectors—not to be necessarily distinctive nor to form a basis. Then there exists a *unique* linear map $\hat{L} : \mathcal{V} \to \mathcal{W}$ such that $w_j = \hat{L}v_j$ for all $j = 1, \ldots, n$.

(proof) Define a map $\hat{A} : \mathcal{V} \to \mathcal{W}$ by the associations

$$\hat{A}v_j \mapsto w_j \tag{†1}$$

and

$$\hat{A}(v_1 z_1 + \cdots v_n z_n) = w_1 z_1 + \cdots w_n z_n \tag{†2}$$

for all $z_1, \ldots, z_n \in \mathbb{C}$. Clearly $\hat{A}$ is linear, and we have shown that there exists a linear map satisfying the required condition. Now suppose that two linear maps $\hat{A}$ and $\hat{B}$ satisfy the condition. Let $v = v_1 z_1 + \cdots v_n z_n \in \mathcal{V}$ with $z_j \in \mathbb{C}$. Note that

$$\hat{B}v = (\hat{B}v_1)z_1 + \cdots + (\hat{B}v_n)z_n = w_1 z_1 + \cdots + w_n z_n = \hat{A}v\,. \tag{†3}$$

As v is arbitrary, we conclude that $\hat{A} = \hat{B}$.

A.2.2 Representations

As asserted by Theorem 7, a linear map $\hat{L} : \mathcal{V} \to \mathcal{W}$ is completely determined by specifying how it maps each element of a bsis $\{v_j : j = 1, \ldots, n\}$ of $\mathcal{V}$. Expanding the result $\hat{L}v_j$ in a basis $\{w_i : i = 1, \ldots, m\}$ of $\mathcal{W}$ as

$$\hat{L}v_j = \sum_i w_i L_{ij} \,, \tag{A.16}$$

we can equivalently say that $\hat{L}$ is *uniquely* specified by the coefficients $L_{ij} \in \mathbb{C}$. The $m \times n$ matrix composed of the coefficients is said to be the matrix representation of $\hat{L}$ in the basess $\{v_j\}$ and $\{w_i\}$, and is denoted by

$$\hat{L} \doteq \begin{bmatrix} L_{11} & L_{12} & \cdots \\ L_{21} & L_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} . \tag{A.17}$$

When the bases $\{v_j\}$ and $\{w_i\}$ are *orthonormal*, the matrix elements L_{ij} can be obtained by means of the Hermitian product, $L_{ij} = \langle w_i, \hat{L}v_j \rangle$, and hence

$$\hat{L}v_j = \sum_i w_i \langle w_i, \hat{L}v_j \rangle \,. \tag{A.18}$$

In quantum mechanics, one has to calculate frequently the matrix representations of linear maps in orthonormal bases, and the procedure is summarized in the following table:

$$\hat{L} \doteq \begin{array}{c|c|ccc} & & v_1 & v_2 & \cdots \\ \hline w_1 & & \langle w_1, \hat{L}v_1 \rangle & \langle w_1, \hat{L}v_2 \rangle & \cdots \\ w_2 & & \langle w_2, \hat{L}v_1 \rangle & \langle w_2, \hat{L}v_2 \rangle & \cdots \\ \vdots & & \vdots & \vdots & \ddots \end{array} \tag{A.19}$$

When $\mathcal{V} = \mathcal{W}$, a linear operator is represented by a square matrix. Let $\{v_i\}$ and $\{w_j\}$ are two different orthonormal bases of $\mathcal{V}$. As they are both orthonormal, there must be a unitary operator $\hat{U}$ such that

$$w_j = \hat{U}v_j = \sum_i v_i U_{ij} \,, \tag{A.20}$$

where $U_{ij} := \langle v_i, w_j \rangle$ is a unitary matrix—see Eq. (A.4). Suppose that L_{ij} be the matrix representation of a linear operator $\hat{L}$ in the basis $\{v_i\}$. What is the matrix representation L'_{ij} of $\hat{L}$ in the new basis $\{w_j\}$? As $\{w_j\}$ is orthonormal, the matrix representation is given by

$$L'_{ij} = \langle w_i, \hat{L}w_j \rangle = \sum_{kl} U^*_{ik} \langle v_i, \hat{L}v_l \rangle U_{lj} = \sum_{kl} U^*_{ik} L_{il} U_{lj} \,, \tag{A.21}$$

that is, the matrix representations in two different bases are related with each other by

$$L' = U^\dagger L U \,. \tag{A.22}$$

A.2.3 Hermitian Conjugate of Operators

On a vector space equipped with a Hermitian product, given a linear operator $\hat{L}$ one can define another linear operator $\hat{L}^\dagger$ naturally related to $\hat{L}$. Hermitian conjugates of operators greatly simplify the evaluations of operator-related expressions and spectral analysis of them.

Theorem 8 (*Hermitian conjugate*) Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces over $\mathbb{C}$, equipped with Hermitian products each. Let $\hat{L} : \mathcal{V} \to \mathcal{W}$ be a linear map. Then, the following statements hold:

(a) There exists a *unique* linear map $\hat{L}^\dagger : \mathcal{W} \to \mathcal{V}$ such that

$$\langle w, \hat{L}v \rangle_{\mathcal{W}} = \langle \hat{L}^\dagger w, v \rangle_{\mathcal{V}} \tag{A.23}$$

for all $v \in \mathcal{V}$ and $w \in \mathcal{W}$.

(b) $(\hat{L}^\dagger)^\dagger$ exists as well and is identital to $\hat{L}$, $(\hat{L}^\dagger)^\dagger = \hat{L}$.

$\hat{L}^\dagger$ is called the *Hermitian conjugate* of $\hat{L}$.

As the matrix representation of a linear map is unique, one can also define the Hermitian conjugate in terms of the matrix representation. Let $\hat{L} : \mathcal{V} \to \mathcal{W}$ be represented by

$$\hat{L}v_j = \sum_i w_i L_{ij} \,. \tag{A.24}$$

Then, $\hat{L}^\dagger : \mathcal{W} \to \mathcal{V}$ is defined by

$$\hat{L}^\dagger w_j = \sum_i v_i L^*_{ji} \,. \tag{A.25}$$

That is, the matrix representation of $\hat{L}^\dagger$ is the conjugate-transpose of the matrix representation of $\hat{L}$. For finite-dimensional vector spaces, the two definitions are equivalent.

For an operator on a vector space, its Hermitian conjugate also acts on the same vector space, and enables to characterize the operator itself.

Definition 9 (*normal operator*) A linear operator $\hat{L}$ on a vector space $\mathcal{V}$ is said to be *normal* if $[\hat{L}^\dagger, \hat{L}] = 0$.

The two most important examples of normal operators are Hermitian operators and unitary operators.

In quantum mechanics, the linear operator representing a physical quantity should be Hermitian:

Definition 10 (*Hermitian operator*) A linear operator $\hat{H}$ on a vector space is called *Hermitian* if

$$\langle \hat{H}v, w\rangle = \langle v, \hat{H}w\rangle \ , \quad \forall v, w \in \mathcal{V} \, . \tag{A.26}$$

There is a simple test for a Hermitian operator on a finite dimensional vector space:

Theorem 11 Let $\mathcal{V}$ be a vector space. An operator $\hat{H}$ on $\mathcal{V}$ is *Hermitian* if and only if $\langle v, \hat{H}v\rangle \in \mathbb{R}$ for all $v \in \mathcal{V}$.

In quantum mechanics, operators usually describe the changes of state of a system by means of transformation of vectors. However, there is a special example where the operator itself describes the "state" of the system. That is, the density operator describes the mixed state of the system. For the proper statistical interpretation of the mixed state, density operators are required to satisfy certain properties. They are Hermitian and positive among others properties. It motivates the following definition.

Definition 12 (*positive operator*) A *Hermitian* operator $\hat{H}$ on a vector space $\mathcal{V}$ is said to be *positive* (or more specifically, *positive definite*) if $\langle v, \hat{H}v\rangle > 0$ for all $v \in \mathcal{V}$ $(v \neq 0)$. A positive operator is denoted as $\hat{H} > 0$. It is said to be *positive semidefinite* or *non-negative* if $\langle v, \hat{H}v\rangle \geq 0$ for all $v \in \mathcal{V}$ $(v \neq 0)$. It is denoted as $\hat{H} \geq 0$.

Another kind of operators one can encounter very frequently in quantum mechanics is unitary operators, *norm-preserving and invertible* linear maps.

Definition 13 Let $\mathcal{V}$ be a vector space $\mathcal{V}$ equipped with a Hermitian product. A linear operator $\hat{U}$ is said to be *unitary* wnen it maps $\mathcal{V}$ onto the whole of $\mathcal{V}$ and preserve the norm. That is, $\hat{U}\mathcal{V} = \mathcal{V}$ and $\langle \hat{U}v, \hat{U}v\rangle = \langle v, v\rangle$ for all $v \in \mathcal{V}$.

A unitary operator is charactierized by the fact that its Hermitian conjugate is identical to its inverse.

Theorem 14 If $\hat{U}$ is a unitary operator on $\mathcal{V}$, then

$$\hat{U}^\dagger \hat{U} = \hat{U}\hat{U}^\dagger = 1 \, . \tag{A.27}$$

In fact, Eq. (A.27) can be used as an alternative definition of a unitary operator.

The unique linear map in Theorem 7 becomes a unitary operator when the image vectors form another orthonormal basis of the same vector space.

Theorem 15 Let $\mathcal{V}$ be a vector space. Let $\{v_1, \ldots, v_n\}$ and $\{w_1, \ldots, w_n\}$ be *orthonormal* bases of $\mathcal{V}$. Then there exists a *unique* unitary operator $\hat{U}$ on $\mathcal{V}$ such that $w_j = \hat{U}v_j$ for all $j = 1, \ldots, n$.

We have already seen in Eq. (A.4) that two orthonormal bases are related by a unitary matrix. Theorem 15 just asserts it again. Indeed, if U is the matrix representation of $\hat{U}$ in the basis $\{v_i\}$, then

$$w_j = \hat{U} v_j = \sum_i v_i U_{ij} \,. \tag{A.28}$$

Theorem 15 is even more general and hold for any orthonormal subsets:

Theorem 16 Let $\mathcal{V}$ is a Hilbert space equipped with a *positive-definite* Hermitian product $\langle \cdot, \cdot \rangle$, and $\mathcal{U} \subset \mathcal{V}$ a subspace. Suppose $\hat{U} : \mathcal{U} \to \mathcal{V}$ is a linear operator which preserves the Hermitian product. That is, for any $u, u' \in \mathcal{U}$,

$$\langle \hat{U} u, \hat{U} u' \rangle = \langle u, u' \rangle \,. \tag{A.29}$$

Show that there exists a unitary operator $\hat{V} : \mathcal{V} \to \mathcal{V}$ which extents $\hat{U}$. That is, $\hat{V} u = \hat{U} u$ for all $u \in \mathcal{U}$ and $\hat{V}$ is defined on the entire space $\mathcal{V}$.

An immediate consequence of Example 16 is that for any pair of vectors v and w one can always find a unitary operator $\hat{U}$ such that $w = \hat{U} v$.

A.3 Dirac's Bra-Ket Notation

For a given vector space $\mathcal{V}$, one can construct another special vector space $\mathcal{V}^*$ associated with $\mathcal{V}$, consisting of all linear mappings from $\mathcal{V}$ to $\mathbb{C}$, $\mathcal{V}^* := \{\phi : \mathcal{V} \to \mathbb{C}\}$. It is called the *dual space* of $\mathcal{V}$. With a fixed vector $v \in \mathcal{V}$ and the Hermitian product, one can define a linear mapping $\phi_v : \mathcal{V} \to \mathbb{C}$ by the relation $\phi_v(w) := \langle v, w \rangle$. Certainly, ϕ_v is an element of $\mathcal{V}^*$. This way, by choosing different vectors from $\mathcal{V}$, one can define a particular kind of linear mappings belonging to $\mathcal{V}^*$. Now the key observation is that in fact, any linear mapping in $\mathcal{V}^*$ is of this kind. That is, there is a one-to-one correspondence $v \leftrightarrow \phi_v$ between $\mathcal{V}$ and $\mathcal{V}^*$. ϕ_v is called the *dual vector* of v.

In Dirac's bra-ket notation, the dual ϕ_v is denoted by $\langle v|$ whereas the native vector v is denoted by $|v\rangle$; hence the name. It is just a simple notational change. However, it simplifies most evaluations in quantum mechanics so greatly that it is widely used. ***Throughout the book, we will almost always be using the bra-ket notation.***

When $\{|\alpha_1\rangle, \cdots, |\alpha_n\rangle\}$ is an orthonormal basis of $\mathcal{V}$, $\{\langle\alpha_1|, \ldots, \langle\alpha_n|\}$ is also an orthonormal basis of $\mathcal{V}^*$ and called the *dual basis* of the former. More importantly, the two bases satisfy the relation

$$\langle \alpha_i | \alpha_j \rangle = \delta_{ij} \,. \tag{A.30}$$

Suppose that a vector $|v\rangle \in \mathcal{V}$ is expanded as

$$|v\rangle = \sum_j |\alpha_j\rangle\, v_j\,, \quad v_j \in \mathbb{C}\,. \tag{A.31}$$

Then, its dual vector $\langle v|$ is given by

$$\langle v| = \sum_j v_j^*\, \langle \alpha_j|\,. \tag{A.32}$$

Armed with the basic properties of the bra-ket notation, now consider a combination of the form $|v\rangle\,\langle v'|$, where $|v\rangle, |v'\rangle \in \mathcal{V}$: We regard it as an operator on $\mathcal{V}$ defined by the association

$$|v\rangle\,\langle v'| : |u\rangle \mapsto |v\rangle\,\langle v'|u\rangle \tag{A.33}$$

for all $|u\rangle \in \mathcal{V}$. A simple inspection—see Definition 8—shows that its Hermitian conjugate $(|v\rangle\,\langle v'|)^\dagger$ is just given by

$$(|v\rangle\,\langle v'|)^\dagger = |v'\rangle\,\langle v|\,. \tag{A.34}$$

If one constructs $|\alpha_i\rangle\,\langle\alpha_j|$ out of a basis $\{|\alpha_j\rangle\}$ of $\mathcal{V}$, then any linear map on $\mathcal{V}$ can be expressed in terms of them—they form a basis of the vector space $\mathcal{L}(\mathcal{V})$ of linear operators (Appendix B.1). In particular, if the basis $\{|\alpha_j\rangle\}$ is *orthonormal*, then a linear operator $\hat{L}$ on $\mathcal{V}$ with the matrix representation L in the same basis is equivalent to

$$\hat{L} = \sum_{ij} |\alpha_i\rangle\, L_{ij}\, \langle\alpha_j|\,, \tag{A.35}$$

and, accordingly, its Hermitian conjugate $\hat{L}^\dagger$ to

$$\hat{L}^\dagger = \sum_{ij} |\alpha_i\rangle\, L_{ji}^*\, \langle\alpha_j|\,. \tag{A.36}$$

Both expansions can be verified by evaluating the matrix elements in the basis and applying the orthogonality relation (A.30). An interesting result is that for *any* orthonormal basis $\{|\alpha_j\rangle\}$ of $\mathcal{V}$, the linear combination

$$\hat{I} = \sum_j |\alpha_j\rangle\,\langle\alpha_j| \tag{A.37}$$

is equal to the identity operator on $\mathcal{V}$. It is called the *completeness relation*. The orthogonality relation (A.30) and the completeness relation (A.37), which are mutually complementary, together empower the bra-ket notation.

Consider a system of two qubits. The associated Hilbert space is spanned by the standard basis.

```
In[•]:= bs = Basis[2]
Out[•]= {|0, 0⟩, |0, 1⟩, |1, 0⟩, |1, 1⟩}
```

Consider a Hermitian operator, physically, corresponding to the Heisenberg exchange interaction between two S=1/2 spins.

```
In[•]:= op = Pauli[1, 1] + Pauli[2, 2] + Pauli[3, 3]
Out[•]= σ^x ⊗ σ^x + σ^y ⊗ σ^y + σ^z ⊗ σ^z
```

This shows the matrix representation of the operator.

```
In[•]:= mat = Matrix[op];
        mat // MatrixForm
Out[•]//MatrixForm=
        ⎛1  0   0  0⎞
        ⎜0  -1  2  0⎟
        ⎜0  2   -1 0⎟
        ⎝0  0   0  1⎠
```

This is an expansion of the operator in the bra-ket notation.

```
In[•]:= op2 = MultiplyDot[bs, mat, Dagger[bs]]
Out[•]= |0, 0⟩ ⟨0, 0| - |0, 1⟩ ⟨0, 1| + 2 |0, 1⟩ ⟨1, 0| +
        2 |1, 0⟩ ⟨0, 1| - |1, 0⟩ ⟨1, 0| + |1, 1⟩ ⟨1, 1|
```

Verify the above expansion by evaluating the matrix representation.

```
In[•]:= mat2 = Outer[Multiply, Dagger[bs], op2 ** bs];
        mat2 // MatrixForm
Out[•]//MatrixForm=
        ⎛1  0   0  0⎞
        ⎜0  -1  2  0⎟
        ⎜0  2   -1 0⎟
        ⎝0  0   0  1⎠
```

More generally, given two vector spaces $\mathcal{V}$ and $\mathcal{W}$, the combination $|w\rangle\langle v|$ with for $|v\rangle \in \mathcal{V}$ and $|w\rangle \in \mathcal{W}$ is a linear map $\mathcal{V} \to \mathcal{W}$ with an association similar to that in Eq. (A.33). Its Hermitian conjugate, $(|w\rangle\langle v|)^\dagger : \mathcal{W} \to \mathcal{V}$, is given by $(|w\rangle\langle v|)^\dagger = |v\rangle\langle w|$.

To illustrate the power of the bra-ket notation, let us consider a linear map $\hat{L} : \mathcal{V} \to \mathcal{W}$ and examine $\hat{L}|\alpha_j\rangle$: Let $\{|\alpha_j\rangle : j = 1, \ldots, m\}$ and $\{|\beta_k\rangle : k = 1, \ldots, n\}$ be orthonormal bases of $\mathcal{V}$ and $\mathcal{W}$, respectively. As $\hat{L}|\alpha_j\rangle$ is an element of $\mathcal{W}$, it should not be affected by the identity operator $\hat{I}_\mathcal{W}$ on $\mathcal{W}$, $\hat{L}|\alpha_j\rangle = \hat{I}_\mathcal{W}\hat{L}|\alpha_j\rangle$. Using the completeness relation (A.37) (replacing $|\alpha_j\rangle$ with $|\beta_k\rangle$), one can get

$$\hat{L}|\alpha_j\rangle = \hat{I}_\mathcal{W}\hat{L}|\alpha_j\rangle = \sum_k |\beta_k\rangle\langle\beta_k|\hat{L}|\alpha_j\rangle . \tag{A.38}$$

Noting that $L_{kj} = \langle\beta_k|\hat{L}|\alpha_j\rangle$ is the matrix elements of the representation of $\hat{L}$ in the given bases [see Eq. (A.18)], one recovers the defining relation [see Eq. (A.16)]

$$\hat{L}|\alpha_j\rangle = \sum_k |\beta_k\rangle L_{kj} \tag{A.39}$$

for the matrix representation of $\hat{L}$. Given the matrix representation L_{kj}, one can further expand $\hat{L}$ in terms of the bra-ket notation as

$$\hat{L} = \sum_{kj} |\beta_k\rangle L_{kj} \langle\alpha_j| \ . \tag{A.40}$$

Once expanded in the bra-ket notation, its Hermitian conjugate $\hat{L}^\dagger : \mathcal{W} \to \mathcal{V}$ reads as

$$\hat{L}^\dagger = \sum_{kj} |\alpha_j\rangle L^*_{kj} \langle\beta_k| \ . \tag{A.41}$$

A.4 Spectral Theorems

- At the moment, this section is very rough and incomplete.

The eigenvalues and eigenvectors of normal operators—see Definition 9—exhibit particulary useful properties. They are frequently used in quantum mechanics and simplifies many calculations and analyses. Here we summarize the properties of eigenvalues and eigenvectors of normal operators, especially, Hermitian and unitary operators. Although we will be focusing on the spectral properties, we will also discuss some other related properties.

A.4.1 Spectral Decomposition

The following theorems summarize the properties of the eigenvectors and eigenvalues of normal operators, especially, Hermitian, positive, and unitary operators:

Theorem 17 Let $\hat{A}$ be a *normal operator*—see Definition 9—on a vector space $\mathcal{V}$.

(a) Eigenstates of $\hat{A}$ belonging to distinct eigenvalues are orthogonal to each other.

(b) The set of all eigenvectors of $\hat{A}$ spans $\mathcal{V}$.

Theorem 18 (a) A Hermitian operator $\hat{H}$ is normal. That is, eigenvectors belonging to different eigenvalues are orthogonal to each other.

(b) Every eigenvalues of a Hermitian operator is real.

Theorem 19 Let $\hat{H}$ be a Hermitian operator on a vector space. Then,

(a) $\hat{H}$ is positive if and only if every eigenvalue of it is positive;

(b) $\hat{H}$ is positive-semidefinite if and only if the eigenvalues are non-negative.

Theorem 20 Let $\hat{U}$ be a unitary operator on a vector space $\mathcal{V}$. Then, every eigenvalue of $\hat{U}$ is of the form $e^{i\phi}$ with $\phi \in \mathbb{R}$.

Theorem 17 enables to expand a normal operator in terms of its eigenvectors and eigenvalues using Dirac's bra-ket notation: Suppose that a normal operator $\hat{A}$ on a vector space $\mathcal{V}$ has eigenvectors $|a\rangle$ and the corresponding eigenvalues a. If some eigenvalues are degenerate, that is, there are more than one linearly independent eigenvectors belonging to the same eigenvalue, we choose mutually orthogonal eigenvectors—this is always possible. Then, the normalized eigenvectors form an orthonormal basis, which is called the *eigenbasis* from $\hat{A}$ of the vector space $\mathcal{V}$. Then the matrix representation of $\hat{A}$ in the eigenbasis from itself must be diagonal with the diagonal elements given by the eigenvalues. Hence, in Dirac's bra-ket notation, $\hat{A}$ can be expanded as

$$\hat{A} = \sum_a |a\rangle\, a\, \langle a| \ , \tag{A.42}$$

where the sum is over all the eigenvalues of $\hat{A}$. The expansion is called the *spectral decomposition* of $\hat{A}$.

Sometimes, it is useful to normalize the eigenvectors $|a\rangle$ of a *positive operator* (see Definition 12 and Theorem 19) by their own (positive) eigenvalues a so that $\langle a|a\rangle = a$. In this case, the spectral decomposition of a positive operator $\hat{A}$ is given by

$$\hat{A} = \sum_a |a\rangle\, \langle a| \,, \tag{A.43}$$

This form of the spectral decomposition for a positive operator should not be confused with the completeness relation (A.37), where an orthonormal basis is used. For a *positive semidefinite operator*, there only appear eigenvectors with positive eigenvalues in the summation in (A.43)—the eigenvectors with zero eigenvalue are dropped automatically.

A.4.2 Functions of Operators

The spectral decomposition provides a convenient way to define *functions of a normal operator*. Let $f : \mathcal{D} \to \mathbb{C}$ be a function of complex variable defined in a domain $\mathcal{D} \subset \mathbb{C}$. Suppose that $\hat{A}$ be a normal operator with all eigenvalues $a \in \mathcal{D}$. Then the function $f(\hat{A})$ of the operator $\hat{A}$—another operator on the same vector space derived from $\hat{A}$—is defined by [to be compared with (A.42)]

$$f(\hat{A}) := \sum_a |a\rangle\, f(a)\, \langle a| \ . \tag{A.44}$$

Surprisingly, most students try to define a function of an operator by means of Taylor series expansion which involves multiple powers of the matrix. In most cases, however, it is very difficult to convince oneself whether the series is actually converging or not. Even if it does, it is tremendously difficult to figure out the behavior of the resulting operator, not to speak of the evaluation of the series itself. In contrast, the definition in (A.44) is well-defined as long as $f(z)$ of complex

variable z is well defined, and often provides clear physical meaning of the resulting operator $f(\hat{A})$. Above all, the evaluation is straightforward and, in many cases, simple. The definition applies only to normal operators. However, it is not a serious restriction since in most physics applications it is normal operators that we want to transform by means of already existing functions.

Consider again a Hermitian operator describing the Heisenberg exchange interaction between two S=1/2 spins.

```
In[•]:= opH = -J * (Pauli[1, 1] + Pauli[2, 2] + Pauli[3, 3])
```

$$\text{Out[•]= } -J\left(\sigma^x\otimes\sigma^x+\sigma^y\otimes\sigma^y+\sigma^z\otimes\sigma^z\right)$$

We want to consider functions of operators, for example, **opH** in particular. To do it, it is most efficient to proceed with the spectral decomposition of the operator.

```
In[•]:= {val, vec} = ProperSystem[opH]
```

$$\text{Out[•]= } \left\{\{3J,-J,-J,-J\},\left\{-|0,1\rangle+|1,0\rangle,|1,1\rangle,|0,1\rangle+|1,0\rangle,|0,0\rangle\right\}\right\}$$

The eigenvectors are orthogonal, but not properly normalized.

```
In[•]:= Outer[Multiply, Dagger[vec], vec] // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix}2&0&0&0\\0&1&0&0\\0&0&2&0\\0&0&0&1\end{pmatrix}$$

Normalize them, and check again.

```
In[•]:= nvec = vec / Sqrt[{2, 1, 2, 1}]
        Outer[Multiply, Dagger[nvec], nvec] // MatrixForm
```

$$\text{Out[•]= } \left\{\frac{-|0,1\rangle+|1,0\rangle}{\sqrt{2}},|1,1\rangle,\frac{|0,1\rangle+|1,0\rangle}{\sqrt{2}},|0,0\rangle\right\}$$

Out[•]//MatrixForm=

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

Suppose we want to get the exponential function of **opH**. For example, the time-evolution operator is given by

```
In[•]:= opU = MultiplyExp[-I t opH]
```

$$\text{Out[•]= } e^{iJt\left(\sigma^x\otimes\sigma^x+\sigma^y\otimes\sigma^y+\sigma^z\otimes\sigma^z\right)}$$

This uses the spectral decomposition.

```
In[•]:= newU = Total@Multiply[nvec, Exp[-I t val], Dagger[nvec]]
```

$$\begin{aligned}\text{Out[•]= } & e^{iJt}|0,0\rangle\langle 0,0|+\frac{1}{2}e^{iJt}|0,1\rangle\langle 0,1|+\\ & \frac{1}{2}e^{-3iJt}|0,1\rangle\langle 0,1|+\frac{1}{2}e^{iJt}|0,1\rangle\langle 1,0|-\\ & \frac{1}{2}e^{-3iJt}|0,1\rangle\langle 1,0|+\frac{1}{2}e^{iJt}|1,0\rangle\langle 0,1|-\frac{1}{2}e^{-3iJt}|1,0\rangle\langle 0,1|+\\ & \frac{1}{2}e^{iJt}|1,0\rangle\langle 1,0|+\frac{1}{2}e^{-3iJt}|1,0\rangle\langle 1,0|+e^{iJt}|1,1\rangle\langle 1,1|\end{aligned}$$

This converts the bra-ket expression into a form in terms of the Pauli operators.

```
In[•]:= newU2 = Elaborate@ExpressionFor@Matrix[newU]
```

$$\text{Out[•]}= \frac{1}{4} e^{-3 i J t} \left(1 + 3 e^{4 i J t}\right) \sigma^0 \otimes \sigma^0 + \frac{1}{4} e^{-3 i J t} \left(-1 + e^{4 i J t}\right) \sigma^x \otimes \sigma^x + \frac{1}{4} e^{-3 i J t} \left(-1 + e^{4 i J t}\right) \sigma^y \otimes \sigma^y + \frac{1}{4} e^{-3 i J t} \left(-1 + e^{4 i J t}\right) \sigma^z \otimes \sigma^z$$

In many cases, MultiplyExp can be further evaluated by means of Elaborate.

```
In[•]:= opU2 = Elaborate@Elaborate[opU]
```

$$\text{Out[•]}= \frac{1}{4} e^{-3 i J t} \left(1 + 3 e^{4 i J t}\right) \sigma^0 \otimes \sigma^0 + \frac{1}{4} e^{-3 i J t} \left(-1 + e^{4 i J t}\right) \sigma^x \otimes \sigma^x + \frac{1}{4} e^{-3 i J t} \left(-1 + e^{4 i J t}\right) \sigma^y \otimes \sigma^y + \frac{1}{4} e^{-3 i J t} \left(-1 + e^{4 i J t}\right) \sigma^z \otimes \sigma^z$$

A.5 Tensor-Product Spaces

When there are more than one systems, each system is associated with a different vector space. Even for a single system, independent degrees of freedom (such as external and internal degrees of freedom) are associated with different vector spaces. Then the vector space of the total system should be constructed by the vector spaces associated with the individual systems or degrees of freedom. Mathematically, such a construction is called the tensor product of the constituent vector spaces.

A.5.1 Vectors in a Product Space

Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces, and $\{|v_i\rangle\}$ and $\{|w_j\rangle\}$ their respective bases. The tensor product $\mathcal{V} \otimes \mathcal{W}$ is a vector space spanned by the basis

$$\{|v_i\rangle \otimes |w_j\rangle : i = 1, \ldots, \dim \mathcal{V};\ j = 1, \ldots, \dim \mathcal{W}\} \tag{A.45}$$

We call it the *standard basis* of $\mathcal{V} \otimes \mathcal{W}$. Obviously, the dimension of $\mathcal{V} \otimes \mathcal{W}$ is given by $\dim(\mathcal{V} \otimes \mathcal{W}) = (\dim \mathcal{V}) \times (\dim \mathcal{W})$. The symbol '$\otimes$' in the basis states separates $|v_i\rangle$ and $|w_j\rangle$ from each other clearly indicating which vector space they are from. In many cases, when there is no risk of confusion, it is dropped and the product states are simply written as $|v_i\rangle\,|w_j\rangle$ or even $|v_i w_j\rangle$.

Here is the standard basis (product basis) for a (unlabelled) two-qubit system.

```
In[•]:= bs = Basis[2];
        bs // LogicalForm
Out[•]= {|0, 0⟩, |0, 1⟩, |1, 0⟩, |1, 1⟩}
```

The Hermitian products $\langle\cdot,\cdot\rangle_{\mathcal{V}}$ and $\langle\cdot,\cdot\rangle_{\mathcal{W}}$ in the corresponding spaces are inherited to the tensor-product space to give the standard Hermitian product

$$(\langle v_i| \otimes \langle w_j|)(|v_k\rangle \otimes |w_l\rangle) = \langle v_i|v_k\rangle_{\mathcal{V}} \langle w_j|w_l\rangle_{\mathcal{W}} . \tag{A.46}$$

Expanded in the standard basis, a vector

$$|\Psi\rangle = \sum_{ij} |v_i\rangle \otimes |w_j\rangle \, \Psi_{ij} \in \mathcal{V} \times \mathcal{W} \tag{A.47}$$

involves $M \times N$ terms in general, where $M := \dim \mathcal{V}$ and $N := \mathcal{W}$. It can be reduced to a form with much less terms. To see this, rewrite the matrix Ψ consisting of the expansion coefficients Ψ_{ij} by means of the singular-value decomposition

$$\Psi = U \Sigma V^{\dagger} , \tag{A.48}$$

where U is a $M \times M$ unitary matrix, V a $N \times N$ matrix, and Σ a $M \times N$ diagonal matrix. The diagonal elements s_j of Σ are all non-negative, and the number R of non-zero elements cannot be greater than $\min(M, N)$. Putting (A.48) back into (A.47) leads to the so-called *Schmidt decomposition*

$$|\Psi\rangle = \sum_{j=1}^{R} |\alpha_j\rangle \otimes |\beta_j\rangle \, s_j , \quad |\alpha_j\rangle := \sum_i |v_i\rangle \, U_{ij} , \quad |\beta_j\rangle := \sum_i |w_i\rangle \, V_{ij}^{*} . \tag{A.49}$$

Note that $\langle\alpha_i|\alpha_j\rangle = \delta_{ij}$ and $\langle\beta_i|\beta_j\rangle = \delta_{ij}$ because U and V are unitary, and that $\sum_{j=1}^{R} s_j^2 = 1$ $(0 < s_j < 1)$ if $|\Psi\rangle$ is normalized.

The number R is called the *Schmidt rank* or *Schmidt number* of the vector $|\Psi\rangle$. When $R = 1$, $|\Psi\rangle$ is factorized as $|\Psi\rangle = |\alpha_1\rangle \otimes |\beta_1\rangle$, and is said to be *separable*. Otherwise, it cannot be factorized and it is called an *entangled vector*. The Schmidt decomposition is a convenient method to test whether a vector is separable or entangled.

Consider an arbitrary vector in the tensor-product space.

```
In[•]:= cc = Re@RandomVector[4];
        vec = bs.cc;
        vec // LogicalForm
Out[•]= -0.392437 |0, 0⟩ + 0.309225 |0, 1⟩ - 0.352869 |1, 0⟩ + 0.733405 |1, 1⟩
```

This is its Schmidt decomposition and shows that the state vector is entangled.

```
In[•]:= {ww, uu, vv} = SchmidtDecomposition[vec, {1}, {2}];
        ww // Normal
        uu
        vv
Out[•]= {0.935711, 0.190978}
Out[•]= {0.504017 |0⟩ + 0.863694 |1⟩, -0.863694 |0⟩ + 0.504017 |1⟩}
Out[•]= {-0.537096 |0⟩ + 0.843521 |1⟩, 0.843521 |0⟩ + 0.537096 |1⟩}
```

`SchmidtForm` presents the Schmidt decomposition in a more intuitively-appealing form. For a thorough analysis of the result, use `SchmidtDecomposition`.

```
In[•]:= new = SchmidtForm[vec, {1}, {2}]
Out[•]= 0.190978 (-0.863694 |0⟩ + 0.504017 |1⟩) ⊗ (0.843521 |0⟩ + 0.537096 |1⟩) +
        0.935711 (0.504017 |0⟩ + 0.863694 |1⟩) ⊗ (-0.537096 |0⟩ + 0.843521 |1⟩)
```

Check whether the two vectors are the same or not.

```
In[•]:= vec - ReleaseTimes[new] // Garner // Chop
Out[•]= 0
```

A.5.2 Operators on a Product Space

Let $\hat{A}$ and $\hat{B}$ be linear operators on $\mathcal{V}$ and $\mathcal{W}$, respectively. Then the *tensor-product operator* $\hat{A} \otimes \hat{B}$ is a linear operator on the tensor-product space $\mathcal{V} \otimes \mathcal{W}$ defined by the association

$$(\hat{A} \otimes \hat{B})(|v_i\rangle \otimes |w_j\rangle) = (\hat{A}\,|v_i\rangle) \otimes (\hat{B}\,|w_j\rangle). \tag{A.50}$$

Suppose that $\hat{A}$ and $\hat{B}$ are represented by matrices A and B, respectively, i.e.,

$$\hat{A}\,|v_j\rangle = \sum_i |v_i\rangle\, A_{ij}\,, \quad \hat{B}\,|v_j\rangle = \sum_i |w_i\rangle\, B_{ij}\,. \tag{A.51}$$

Then it follows from (A.50) that

$$(\hat{A} \otimes \hat{B})(|v_i\rangle \otimes |w_j\rangle) = \left(\sum_k |v_k\rangle\, A_{ki}\right) \otimes \left(\sum_l |w_l\rangle\, B_{lj}\right) = \sum_{kl} |v_k\rangle \otimes |w_l\rangle\, A_{ki} B_{lj}. \tag{A.52}$$

It implies that the matrix representation of $\hat{A} \otimes \hat{B}$ in the standard product basis is given by the *direct product* $A \otimes B$ of the two matrices.

In general, a linear operator $\hat{C}$ on $\mathcal{V} \otimes \mathcal{W}$ is not a single product but a sum of such products. How many terms are there? Suppose that the matrix $C_{ij;kl}$ is the matrix representation of $\hat{C}$ in the standard product basis,

$$\hat{C}\,|v_k w_l\rangle = \sum_{ij} |v_i v_j\rangle\, C_{ij;kl}\,, \tag{A.53}$$

or equivalently,

$$\hat{C} = \sum_{ij;kl} |v_i w_j\rangle \langle v_k w_l|\; C_{ij;kl} = \sum_{ij;kl} |v_i\rangle \langle v_k| \otimes |w_j\rangle \langle w_l|\; C_{ij;kl}\,. \tag{A.54}$$

The $M^2 \times N^2$ matrix $G_{ik;jl} := C_{ij;kl}$ with collective indices (ik) and (jl) has a singular value decomposition

$$G_{ik;jl} = \sum_\mu U_{ik;\mu} \gamma_\mu V^\dagger_{\mu;jl}\,, \quad \gamma_\mu \geq 0. \tag{A.55}$$

Defining

$$\hat{A}_\mu := \sum_{ik} |v_i\rangle \langle v_k| U_{ik;\mu} , \quad \hat{B}_\mu := \sum_{jl} |w_j\rangle \langle w_l| V^*_{jl;\mu} , \tag{A.56}$$

leads to the expression

$$\hat{C} = \sum_\mu \hat{A}_\mu \otimes \hat{B}_\mu \gamma_\mu , \tag{A.57}$$

which is in direct analogy with the Schmidt decomposition (A.49) of a vector in the tensor-product space. Here the number of non-vanishing singular values γ_μ is less than or equal to $\min(M^2, N^2)$.

Appendix B

Superoperators

- March 20, 2021 (v1.7)

A superoperator is a linear operator acting on a vector space of linear operators. As the concept of vectors is completely general, at a first glance there seems to be no reason why one should reserve a distinctive name for such operators and devote additional discussions. A considerable amount of interest in superoperators came with the booming of quantum information theory in the 1990s when it became clear that superoperators are important in the study of entanglement. Since then, mathematical theories on superoperators have been developed at a notably fast pace and applied to a wide range of subjects in quantum computation and quantum information. In this appendix, we briefly survey the properties of superoperators and provide some mathematical tools for the studies of entanglement and decoherence (see Chapter 5).

B.1 Operators as Vectors

The addition of two operators acting on a vector space as well as the multiplication of an operator by a scalar are defined in a natural and straightforward way. That is, operators on a vector space form a vector space themselves. Vector space of operators is not merely a mathematical generalization, but has an important physical relevance. In quantum physics, a mixed state, a statistical mixture of pure quantum states, is described by a so-called density operator.

There is another rather mathematically motivated and yet physically important fact that makes regarding operators as vectors very useful: Any unitary operator $\hat{U}$ can be written in the form $\hat{U} = \exp(i\hat{H})$, where $\hat{H}$ is an Hermitian operator. To describe any physical process, one has to deal with unitary operators. Because of the defining constraint, $\hat{U}^\dagger\hat{U} = \hat{I}$, it is often difficult to directly handle unitary operators. In most case, it is much more convenient and easier to handle Hermitian operators and consider the exponential function of them. As there is no constraint—apart from the rather trivial Hermiticity condition—for Hermitian operators, it is

natural to express them as linear combinations of some basis elements. It makes the handling of physically relevant operators much more tractable.

In this appendix, we will discuss the general structure of the vector space of all linear operators on a vector space. Before we discuss more general vector spaces of linear operators, let us first consider vector spaces of matrices.

Exercise 7 (*matrices as vectors*) Consider the set $\mathcal{M}_n$ of all $n \times n$ complex matrices.

(a) Show that $\mathcal{M}_n$ is a vector space.

(b) What is the dimension of $\mathcal{M}_n$.

(c) Define a *Hermitian product* in $\mathcal{M}_n$.

(d) Construct an *orthogonal basis* of $\mathcal{M}_n$.

Here is an even more specific example.

Example 8 (*Pauli decomposition*) Consider $\mathcal{M}_2$.

(a) Show that the four Pauli matrices $\hat{\sigma}^\mu$ ($\mu = 0, 1, 2, 3$) form an orthogonal basis.

(b) Given an arbitrary matrix $L \in \mathcal{M}_2$, expand it in terms of the Pauli matrices. That is, find the most general form of L in terms of the Pauli matrices.

(c) Find the most general form of a Hermitian matrix $H \in \mathcal{M}_2$.

(d) Find the most general form of a unitary matrix $U \in \mathcal{M}_2$.

Let us demonstrate that any 2×2 matrix can be written as a linear superposition of the Pauli matrices. Consider an arbitrary 2×2 matrix.

```
In[•]:= L = {{1, 2 I},
            {-I, 3}};
        L // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 1 & 2\,i \\ -i & 3 \end{pmatrix}$$

ExpressionFor converts a matrix into an operator expression in terms of the Pauli operators -- the Pauli operators are the operator forms of the Pauli matrices.

```
In[•]:= op = ExpressionFor[L]
        Elaborate[op]
```

Out[•]= $2\,\sigma^0 - \sigma^z + 2\,i\,\sigma^+ - i\,\sigma^-$

Out[•]= $2\,\sigma^0 + \frac{i\,\sigma^x}{2} - \frac{3\,\sigma^y}{2} - \sigma^z$

The symbols σ^μ in the above are the displayed form of Pauli.

```
In[•]:= InputForm[op]
```

Out[•]//InputForm=

```
2*Pauli[0] - Pauli[3] + (2*I)*Pauli[4] - I*Pauli[5]
```

ThePauli is the matrix form of Pauli. The following statement reconstructs the original matrix.

```
In[•]:= new = 2 ThePauli[0] + (I / 2) ThePauli[1] - (3 / 2) ThePauli[2] - ThePauli[3];
        new // MatrixForm
Out[•]//MatrixForm=
        ( 1   2 i )
        ( -i   3  )
```

The conversion of Pauli to the corresponding matrix -- ThePauli -- can be achieved by simply using Matrix.

```
In[•]:= new2 = Matrix[op];
        new2 // MatrixForm
Out[•]//MatrixForm=
        ( 1   2 i )
        ( -i   3  )
```

Let us analyse the above demonstration in more detail. Here are the Pauli matrices. They form a basis of $\mathcal{M}_2$.

```
In[•]:= bs = ThePauli /@ {0, 1, 2, 3};
        MatrixForm /@ bs
Out[•]= {( 1  0 ), ( 0  1 ), ( 0  -i ), ( 1   0 )}
        ( 0  1 )  ( 1  0 )  ( i   0 )  ( 0  -1 )
```

The function PauliDecompose returns the expansion coefficients in the Pauli basis.

```
In[•]:= cc = PauliDecompose[L]
Out[•]= {2, i/2, -3/2, -1}
```

Indeed, the coefficients reconstructs the original matrix.

```
In[•]:= new = cc.bs;
        new // MatrixForm
Out[•]//MatrixForm=
        ( 1   2 i )
        ( -i   3  )

In[•]:= L - new // Chop
Out[•]= {{0, 0}, {0, 0}}
```

Let us further consider a 2×2 Hermitian matrix.

```
In[•]:= H = RandomHermitian[];
        H // MatrixForm
Out[•]//MatrixForm=
        (   0.558658 + 0. i      -0.89873 - 0.66087 i )
        ( -0.89873 + 0.66087 i      0.120512 + 0. i   )

In[•]:= H - Topple[H] // Chop // MatrixForm
Out[•]//MatrixForm=
        ( 0  0 )
        ( 0  0 )
```

It is noted that the expansion coefficients are all real.

```
In[•]:= cc = PauliDecompose[H] // Chop
Out[•]= {0.339585, -0.89873, 0.66087, 0.219073}
```

Finally, consider a 2×2 unitary matrix.

```
In[•]:= U = RandomUnitary[];
        U // MatrixForm
Out[•]//MatrixForm=
        ( -0.35467 + 0.843542 i   -0.196382 - 0.352251 i )
        ( -0.116783 + 0.386016 i   0.526328 + 0.748553 i )

In[•]:= Topple[U].U // Chop // MatrixForm
Out[•]//MatrixForm=
        ( 1.  0 )
        ( 0  1. )
```

One can see that the column vector of the expansion coefficients is normalized.

```
In[•]:= cc = PauliDecompose[U]
Out[•]= {0.085829 + 0.796047 i, -0.156582 + 0.0168825 i,
         0.369134 - 0.0397996 i, -0.440499 + 0.0474942 i}

In[•]:= Conjugate[cc].cc // Chop
Out[•]= 1.
```

In the above demonstration, we have used the Pauli operators for *unlabelled* qubits. One could use the Pauli operators for qubits with labels. Let us consider a system of two qubits, which are denoted by the symbol S.

```
Let[Qubit, S]
```

Consider an arbitrary 2×2 matrix.

```
In[•]:= mat = RandomInteger[{-3, 3}, {2, 2}];
        mat // MatrixForm
Out[•]//MatrixForm=
        ( 1   0 )
        ( 1  -1 )
```

This converts the matrix into an operator expression in terms of the Pauli operators on the labelled qubits. Here S[μ] corresponds to Pauli[μ] acting on the the qubit `S[None]`.

```
In[•]:= op = Elaborate@ExpressionFor[mat, S[None]]
Out[•]= S^x/2 - (i S^y)/2 + S^z
```

The operator expression can be converted back to a matrix by using Matrix.

```
In[•]:= new = Matrix[op];
        new // MatrixForm
Out[•]//MatrixForm=
        ( 1   0 )
        ( 1  -1 )
```

Definition 21 (*vector space of linear maps*) Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces. Let $\mathcal{L}(\mathcal{V}, \mathcal{W})$ be the set of all linear maps $\hat{L} : \mathcal{V} \to \mathcal{W}$. Equip it with a natural multiplication of linear map $\hat{L}$ by scalars $x \in \mathbb{C}$ as

$$(x\hat{L})\,|v\rangle := x(\hat{L}\,|v\rangle) \tag{B.1}$$

for all $|v\rangle \in \mathcal{V}$. Also define the sum of two linear maps by

$$(\hat{L} + \hat{M})\,|v\rangle := \hat{L}\,|v\rangle + \hat{M}\,|v\rangle \tag{B.2}$$

for all $|v\rangle \in \mathcal{V}$. Then the set $\mathcal{L}(\mathcal{V}, \mathcal{W})$ forms a vector space. When $\mathcal{V} = \mathcal{W}$, $\mathcal{L}(\mathcal{V}) \equiv \mathcal{L}(\mathcal{V}, \mathcal{V})$ is the vector space of all linear *operators* on $\mathcal{V}$.

Let $\{|v_1\rangle, \ldots, |v_m\rangle\}$ and $\{|w_1\rangle, \ldots, |w_n\rangle\}$ be bases of $\mathcal{V}$ and $\mathcal{W}$, respectively. A natural choice for basis of $\mathcal{L}(\mathcal{V}, \mathcal{W})$ is

$$\{|w_k\rangle \langle v_j| : j = 1, \ldots, m;\ k = 1, \ldots, n\}. \tag{B.3}$$

$\mathcal{L}(\mathcal{V}, \mathcal{W})$ also needs a Hermitian product. In the same spirit as the Frobenius inner product of matrices, a natural choice of the Hermitian product in $\mathcal{L}(\mathcal{V}, \mathcal{W})$ inherited from the Hermitian products in $\mathcal{V}$ and $\mathcal{W}$ is that

$$\langle \hat{L}, \hat{M} \rangle := \operatorname{Tr} \hat{L}^\dagger \hat{M} = \sum_j \langle v_j, \hat{L}^\dagger \hat{M} v_j \rangle_{\mathcal{V}} = \sum_j \langle \hat{L} v_j, \hat{M} v_j \rangle_{\mathcal{W}} . \tag{B.4}$$

It is called the *trace Hermitian product* or simply *trace product*. With respect to this Hermitian product, the basis in Eq. (B.3) is orthonormal.

For the vector space $\mathcal{L}(\mathcal{V})$ of all *operators* on $\mathcal{V}$, equipped with the trace Hermitian product analogous to (B.4), another choice of basis other than the standard basis (B.3) is widely used. It is to pick the identity operator $\hat{I}$ as an element of the basis. Then every other element in the basis must be *traceless*. For example, let $\mathcal{S}$ be a two-dimensional Hilbert space, and in $\mathcal{L}(\mathcal{S})$ the four Pauli operators form such a basis, $\{\hat{I} \equiv \hat{S}^0, \hat{S}^x, \hat{S}^y, \hat{S}^z\}$. Obviously, the non-identity three Pauli operators are traceless, $\operatorname{Tr} \hat{S}^\mu = 0$ $(\mu = x, y, z)$. Any operator $\hat{A} \in \mathcal{L}(\mathcal{S})$ is expanded in the four Pauli operators

$$\hat{A} = \hat{S}^0 \alpha_0 + \hat{S}^x \alpha_x + \hat{S}^y \alpha_y + \hat{S}^z \alpha_z \quad (\alpha_\mu \in \mathbb{C}). \tag{B.5}$$

The expansion coefficients can be obtained using the orthogonality of the basis and the trace Hermitian product,

$$\alpha_\mu = \frac{1}{2} \operatorname{Tr} \hat{S}^x \hat{A} \tag{B.6}$$

(recall that the Pauli operators are all Hermitian). We have already observed this fact in Exercise 8 using the matrix form of the Pauli operators.

Example 9 Consider the Hilbert space $\mathcal{S} \otimes \mathcal{S}$ associated with a system of two qubits. Show that the products of the Pauli operators

$$\{\hat{S}_1^\mu \otimes \hat{S}_2^\nu\} \tag{B.7}$$

form an orthogonal basis of $\mathcal{L}(\mathcal{S} \otimes \mathcal{S})$.

Solution: Consider an arbitrary 4×4 matrix.

```
In[•]:= mat = RandomInteger[{-3, 3}, {4, 4}];
        mat // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -2 & -1 & -1 & 1 \\ 1 & 0 & 1 & -1 \\ 2 & 2 & 0 & 2 \\ -2 & -3 & 0 & -3 \end{pmatrix}$$

This converts the matrix into an operator expression in terms of the products of the Pauli operators, which are represented by `Pauli[μ,ν,...]`.

```
In[•]:= op = Elaborate@ExpressionFor[mat]
```

Out[•]=

$$-\frac{5}{4}\sigma^0\otimes\sigma^0+\frac{\sigma^0\otimes\sigma^x}{2}+\frac{\sigma^0\otimes\sigma^z}{4}-\frac{3\,\sigma^x\otimes\sigma^0}{4}+\frac{\sigma^x\otimes\sigma^x}{2}+\mathrm{i}\,\sigma^x\otimes\sigma^y+\frac{5\,\sigma^x\otimes\sigma^z}{4}-\frac{1}{4}\mathrm{i}\,\sigma^y\otimes\sigma^0+$$
$$\frac{1}{2}\mathrm{i}\,\sigma^y\otimes\sigma^x+\sigma^y\otimes\sigma^y-\frac{5}{4}\mathrm{i}\,\sigma^y\otimes\sigma^z+\frac{\sigma^z\otimes\sigma^0}{4}-\frac{\sigma^z\otimes\sigma^x}{2}-\mathrm{i}\,\sigma^z\otimes\sigma^y-\frac{5\,\sigma^z\otimes\sigma^z}{4}$$

This converts the operator expression back into the original matrix.

```
In[•]:= new = Matrix[op];
        new // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -2 & -1 & -1 & 1 \\ 1 & 0 & 1 & -1 \\ 2 & 2 & 0 & 2 \\ -2 & -3 & 0 & -3 \end{pmatrix}$$

In the above demonstration, we have used the Pauli operators for *unlabelled* qubits. One could use the Pauli operators for qubits with labels. Let us consider a system of two qubits, which are denoted by the symbol S and the flavor indices.

```
Let[Qubit, S]
```

This converts the matrix into an operator expression in terms of the Pauli operators on the labelled qubits. Here S[1,μ] corresponds to Pauli[μ,0] acting on the first qubit and S[2,μ] to Pauli[0,μ] on the second qubit.

```
In[•]:= op = ExpressionFor[mat, S[{1, 2}, None]]
        Elaborate[op]
```

Out[•]=

$$-\frac{5}{4}-\frac{5}{4}S_1^zS_2^z-\frac{3}{2}S_1^zS_2^++\frac{1}{2}S_1^zS_2^-+S_1^+S_2^++S_1^+S_2^-+$$
$$\frac{5}{2}S_1^-S_2^z+2\,S_1^-S_2^+-2\,S_1^-S_2^-+\frac{S_1^z}{4}-S_1^+-\frac{S_1^-}{2}+\frac{S_2^z}{4}+\frac{S_2^+}{2}+\frac{S_2^-}{2}$$

Out[•]=

$$-\frac{5}{4}+\frac{1}{2}S_1^xS_2^x+\mathrm{i}\,S_1^xS_2^y+\frac{5}{4}S_1^xS_2^z+\frac{1}{2}\mathrm{i}\,S_1^yS_2^x+S_1^yS_2^y-$$
$$\frac{5}{4}\mathrm{i}\,S_1^yS_2^z-\frac{1}{2}S_1^zS_2^x-\mathrm{i}\,S_1^zS_2^y-\frac{5}{4}S_1^zS_2^z-\frac{3\,S_1^x}{4}-\frac{\mathrm{i}\,S_1^y}{4}+\frac{S_1^z}{4}+\frac{S_2^x}{2}+\frac{S_2^z}{4}$$

The operator expression can be converted back to a matrix by using Matrix.

```
In[•]:= new = Matrix[op];
        new // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -2 & -1 & -1 & 1 \\ 1 & 0 & 1 & -1 \\ 2 & 2 & 0 & 2 \\ -2 & -3 & 0 & -3 \end{pmatrix}$$

B.2 Superoperators

As the operators on vector spaces are vectors themselves, one can consider a linear map $\mathscr{F} : \mathscr{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ from operators on $\mathcal{V}$ to those on $\mathcal{W}$. We call it a *super-mapping* or *supermap* to distinguish it from one between simple vectors. Physically, supermaps are most relevant when input operators represent mixed states, that is, when they are density operators.

In many cases, the input and output spaces are identical, $\mathcal{V} = \mathcal{W}$. In such a case, $\mathscr{F}$ itself is an operator—an operator on operators—and is called a *superoperator* on $\mathcal{V}$. superoperators are useful to mathematically describe the evolution of open quantum systems, i.e., the systems interacting with other surrounding systems.

B.2.1 Matrix Representation

How can a supermap be characterized? Recall that a linear map of simple vectors is characterized by its matrix representation. Upon a choice of bases $\{|v_j\rangle\}$ and $\{|w_i\rangle\}$ of $\mathcal{V}$ and $\mathcal{W}$, respectively, $\hat{L} : \mathcal{V} \to \mathcal{W}$ is completely specified by

$$\hat{L}\,|v_j\rangle = \sum_i |w_i\rangle\, L_{ij}\,. \tag{B.8}$$

For a supermap $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$, the involved spaces $\mathcal{L}(\mathcal{V})$ and $\mathcal{L}(\mathcal{W})$ have additional algebraic structures, and there are several ways to characterize it at different levels. One straightforward way to characterize a supermap is to take a plain analogy of the above matrix representation. Recall that $|v_k\rangle\langle v_l|$ and $|w_i\rangle\langle w_j|$ form the standard bases of $\mathcal{L}(\mathcal{V})$ and $\mathcal{L}(\mathcal{W})$, respectively. For each $|v_k\rangle\langle v_l|$,, $\mathscr{F}(|v_k\rangle\langle v_l|)$ belongs to $\mathcal{L}(\mathcal{W})$ and is expanded in the standard basis $\{|w_i\rangle\langle w_j|\}$ [see Eq. (B.3)] as (notice the order of indices in $C_{ik;jl}$)

$$\mathscr{F}(|v_k\rangle\langle v_l|) = \sum_{ij} |w_i\rangle\langle w_j|\, C_{ik;jl}\,, \quad C_{ik;jl} \in \mathbb{C}\,. \tag{B.9}$$

Here the matrix C—regarding (ik) and (jl) as collective indices—is called the *Choi matrix* associated with the supermap $\mathscr{F}$, and completely characterizes the supermap $\mathscr{F}$. For an arbitrary linear operator $\hat{\rho} := \sum_{kl} |v_k\rangle\langle v_l|\,\rho_{kl} \in \mathcal{L}(\mathcal{V})$, its image through $\mathscr{F}$ is given by

$$\hat{\sigma} := \mathscr{F}(\hat{\rho}) = \sum_{kl} \mathscr{F}(|v_k\rangle\langle v_l|)\rho_{kl} = \sum_{ij}\sum_{kl} |w_i\rangle\langle w_j|\, C_{ik;jl}\,\rho_{kl}\,. \tag{B.10}$$

This implies that the components of $\hat{\sigma}$ and $\hat{\rho}$ are related to each other by the Choi matrix as

$$\sigma_{ij} = \sum_{kl} C_{ik;jl}\,\rho_{kl}. \tag{B.11}$$

Let us take a few examples: First consider a supermap of the form

$$\mathscr{F}(\hat{\rho}) = \hat{A}\hat{\rho}\hat{B}^\dagger\,, \tag{B.12}$$

where $\hat{A}, \hat{B} \in \mathcal{L}(\mathcal{V}, \mathcal{W})$. Then the Choi matrix of $\mathscr{F}$ is given by

$$C_{ij;kl} = A_{ij}B^*_{kl}\,. \tag{B.13}$$

From this identity, it immediately follows that the Choi matrix for the supermap $\mathscr{F}(\hat{\rho}) = -i[\hat{H}, \hat{\rho}]$ is given by

$$C_{ij;kl} = -i(H_{ij}\delta_{kl} - \delta_{ij}H^*_{kl})\,. \tag{B.14}$$

B.2.2 Operator-Sum Representation

Another method to characterize a supermap is the so-called operator-sum representation and turns out to be extremely useful in many areas of physics, including quantum information theory and quantum statistical mechanics. Putting the identity $\rho_{kl} = \langle v_k|\hat{\rho}|v_l\rangle$ back into (B.10), one gets

$$\mathscr{F}(\hat{\rho}) = \sum_{ij}\sum_{kl} |w_i\rangle\langle v_k|\hat{\rho}|v_l\rangle\langle w_j|\, C_{ik;jl}\,. \tag{B.15}$$

Now identify $\hat{E}_{ik} := |w_i\rangle\langle v_k|$ as a linear map from $\mathcal{V}$ to $\mathcal{W}$, $\hat{E}_{ik} \in \mathcal{L}(\mathcal{V}, \mathcal{W})$. Similarly, $|v_l\rangle\langle w_j| \in \mathcal{L}(\mathcal{W}, \mathcal{V})$ and it is identical to $\hat{E}^\dagger_{jl}$. Hence

$$\mathscr{F}(\hat{\rho}) = \sum_{ij}\sum_{kl} \hat{E}_{ik}\hat{\rho}\hat{E}^\dagger_{jl}C_{ik;jl}\,. \tag{B.16$'$}$$

With the notation of collective indices $\mu \equiv (ik)$ and $\nu \equiv (jl)$, the supermap $\mathscr{F}$ takes the operator-sum representation

$$\mathscr{F}(\hat{\rho}) = \sum_{\mu=1}^{MN}\sum_{\nu=1}^{MN} \hat{E}_\mu\hat{\rho}\hat{E}^\dagger_\nu C_{\mu\nu}\,, \tag{B.16}$$

where $M := \dim\mathcal{V}$ and $N := \dim\mathcal{W}$. Diagrammatically, it is depicted as

$$\begin{array}{ccc} \mathcal{V} & \xleftarrow{\hat{E}^\dagger_\nu} & \mathcal{W} \\ \big\downarrow{\scriptstyle \hat{\rho}} & & \big\downarrow{\scriptstyle \mathscr{E}(\hat{\rho})} \\ \mathcal{V} & \xrightarrow{\hat{E}_\mu} & \mathcal{W} \end{array} \tag{B.17}$$

In Eqs. (B.16) and (B.16$'$), a standard basis $\{\hat{E}_\mu\}$ has been chosen in $\mathcal{L}(\mathcal{V}, \mathcal{W})$. But one can choose any basis, which leads to the following theorem.

Theorem 22 Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces. If $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ is a supermap, then there exist $\hat{F}_\mu \in \mathcal{L}(\mathcal{V}, \mathcal{W})$ such that

$$\mathscr{F}(\hat{\rho}) = \sum_{\mu=1}^{MN} \hat{F}_\mu \hat{\rho} \hat{F}_\nu^\dagger \, C_{\mu\nu} \,, \quad C_{\mu\nu} \in \mathbb{C} \,. \tag{B.18}$$

We are often interested in mapping density operators—not just any operators. In this case, the relevant supermaps are required to preserve the properties of density operators—density operators are *Hermitian* and in particular *positive*. The condition to preserve Hermiticity simplifies the representation (B.16) further.

Theorem 23 Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces, equipped with Hermitian products. Let $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ be a supermap. If $\mathscr{F}(\hat{\rho})$ is Hermitian for every Hermitian $\hat{\rho} \in \mathcal{L}(\mathcal{V})$, then there exist $\hat{F}_\mu \in \mathcal{L}(\mathcal{V}, \mathcal{W})$ such that

$$\mathscr{F}(\hat{\rho}) = \sum_{\mu=1}^{MN} \epsilon_\mu \hat{F}_\mu \hat{\rho} \hat{F}_\mu^\dagger \,, \tag{B.19}$$

where $\epsilon_\mu = \pm 1$ and the linear maps $\hat{F}_\mu$ are orthogonal to each other, $\mathrm{Tr}\, \hat{F}_\mu^\dagger \hat{F}_\nu = 0$ for $\mu \neq \nu$.

In the representation (B.19), all numerical factors have been absorbed into the operators $\hat{F}_\mu$ leaving only possibly negative signs in ϵ_μ. An immediate question is what condition a supermap should satisfy to have $\epsilon_\mu = 1$ for all μ? Would the condition to preserve positivity be sufficient to guarantee it? Unfortunately, the positivity-preserving condition does not bring any meaningful simplification, and a much stronger condition is required:

Definition 24 (*completely positive supermap*) Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces and $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ a supermap. $\mathscr{F}$ is said to be *completely positive* if $\mathscr{F} \otimes \mathscr{I} : \mathcal{L}(\mathcal{V} \otimes \mathcal{E}) \to \mathcal{L}(\mathcal{W} \otimes \mathcal{E})$ is positive[1] for any vector space $\mathcal{E}$, where $\mathscr{I}$ denotes the identity superoperator on $\mathcal{L}(\mathcal{E})$.

Physically, the vector space $\mathcal{E}$ is associated with an environment—see Section 5.1. The operator-sum representation in (B.18) or (B.19) is further simplified for completely positive supermaps. The following example exhibits the motivation.

Exercise 10 For any linear maps $\hat{F}_\mu \in \mathcal{L}(\mathcal{V}, \mathcal{W})$, the supermap $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ defined by

$$\mathscr{F}(\hat{\rho}) := \sum_\mu \hat{F}_\mu \hat{\rho} \hat{F}_\mu^\dagger \tag{B.20}$$

is completely positive.

[1] Here "positive" actually means "positivity-preserving".

Note that the linear maps $\hat{F}_\mu$ in (B.20) are completely arbitrary. They do not have to be orthogonal to each other, $\mathrm{Tr}\,\hat{F}_\mu^\dagger \hat{F}_\nu \neq 0$, nor to span the space $\mathcal{L}(\mathcal{V},\mathcal{W})$. The following theorem confirms that any supermap takes the above form in Eq. (B.20). In fact, for a given supermap, one can find a more compact and refined linear maps to represent it with.

Theorem 25 (*Kraus representation theorem*) Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces, and $\mathscr{F} : \mathcal{L}(\mathcal{V}) \to \mathcal{L}(\mathcal{W})$ be a supermap. Then the following statement are equivalent:

(a) $\mathscr{F}$ is completely positive.

(b) For any $\hat{\rho} \in \mathcal{L}(\mathcal{V})$, the effect $\mathscr{F}(\hat{\rho})$ can be written as

$$\mathscr{F}(\hat{\rho}) = \sum_{\mu=0}^{m-1} \hat{F}_\mu \hat{\rho} \hat{F}_\mu^\dagger , \tag{B.21}$$

where $m \leq (\dim \mathcal{V})(\dim \mathcal{W})$ and $\hat{F}_\mu : \mathcal{V} \to \mathcal{W}$ are *mutually orthogonal* linear maps—$\mathrm{Tr}\,\hat{F}_\mu^\dagger \hat{F}_\nu = 0$ for all $\mu \neq \nu$.

(c) For any $\hat{\rho} \in \mathcal{L}(\mathcal{V})$, the effect $\mathscr{F}(\hat{\rho})$ can be written as a finite sum of the form

$$\mathscr{F}(\hat{\rho}) = \sum_{\mu} \hat{F}_\mu \hat{\rho} \hat{F}_\mu^\dagger , \tag{B.22}$$

where $\hat{F}_\mu : \mathcal{V} \to \mathcal{W}$ are (arbitrary) linear maps.

The expressions (B.21) and (B.22) are called the *Kraus operator-sum representation* or simply the *Kraus representation* of the completely positive supermap $\mathscr{F}$. The linear maps $\hat{F}_\mu$ are called the *Kraus elements* or the *Kraus maps* (the *Kraus operators* when $\mathcal{V} = \mathcal{W}$). The *orthogonal* Kraus elements in Eq. (B.21) are optimal in the sense that the sum has the least possible number of terms.

It is fairly obvious that a supermap expressed in the form (B.21) or (B.22) is completely positive. The converse can be proved by starting from (B.16′).

Exercise 11 Using the representation in (B.16′) and requiring the positivity of $(\mathscr{F} \otimes \mathscr{J})(|\Phi\rangle\langle\Phi|)$ with

$$|\Phi\rangle := \sum_j |v_j\rangle \otimes |v_j\rangle \in \mathcal{V} \otimes \mathcal{V}, \tag{B.23}$$

prove that a completely positive map has the Kraus representation of the form (B.21).

B.2.3 Choi Isomorphism

A less widely known yet intriguing method to characterize a supermap is provided by the *Choi isomorphism* (also known as Jamiolkowski, Choi-Jamiolkowski or Jamiolkowski-Choi isomorphism).[2]

Before we discuss the Choi isomormphism of supermaps, let us first examine the same isomorphism of linear maps. Let $\hat{A} : \mathcal{V} \to \mathcal{W}$ be a linear map. Recall that it is completely characterized by the $n \times m$ matrix A such that

$$\hat{A} = \sum_{kj} |w_k\rangle A_{kj} \langle v_j| , \tag{B.24}$$

where $\{v_j\}$ and $|w_k\rangle$ are bases of $\mathcal{V}$ and $\mathcal{W}$, respectively. Now note that the same matrix defines a vector

$$|A\rangle := \sum_{kj} |w_k\rangle \otimes |v_j\rangle A_{kj} \tag{B.25}$$

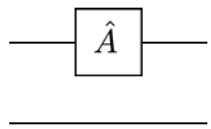
in the tensor-product space $\mathcal{W} \otimes \mathcal{V}$. The correspondence $\hat{A} \leftrightarrow |A\rangle$ turns out to be an isomorphism between $\mathcal{L}(\mathcal{V}, \mathcal{W})$ and $\mathcal{W} \otimes \mathcal{V}$. The isomorphism is called the Choi isomorphism and $|A\rangle$ is called the *Choi vector* associated with the linear map $\hat{A}$. Looking almost trivial at a first glance, the isomorphism brings about several interesting things. To see it, consider a maximally entangled state

$$|\Phi\rangle := \sum_{k} |v_k\rangle \otimes |v_k\rangle \in \mathcal{V} \otimes \mathcal{V} . \tag{B.26}$$

in the tensor-product space $\mathcal{V} \otimes \mathcal{V}$. First, observe that the Choi vector $|A\rangle$ of a linear map $\hat{A}$ is given by

$$|A\rangle = (\hat{A} \otimes \hat{I}) |\Phi\rangle . \tag{B.27}$$

This is depicted in the following quantum circuit model

$|A\rangle =$ $|\Phi\rangle$ 

(B.28)

The isomorphism preserves the Hermitian products in $\mathcal{L}(\mathcal{V}, \mathcal{W})$ and $\mathcal{W} \otimes \mathcal{V}$, that is, $\mathrm{Tr}\, \hat{A}\hat{B} = \langle A|B\rangle$ for all linear maps $\hat{A}$ and $\hat{B}$. Further, for an arbitrary state $|\psi\rangle = \sum_j |v_j\rangle \psi_j \in \mathcal{V}$, define its conjugate state by

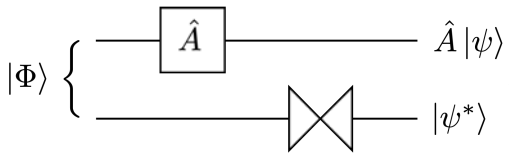
$$|\psi^*\rangle := \sum_{j} |v_j\rangle \psi_j^* . \tag{B.29}$$

Then, it follows that

$$\hat{A} |\psi\rangle = \langle \psi^*| (\hat{A} \otimes \hat{I}) |\Phi\rangle = \langle \psi^*|A\rangle , \tag{B.30}$$

[2] There are subtle but important differences between the Choi and Jamiolkowski isomorphism; see Jiang *et al.* (2013).

where the Hermitian product on the right-hand side is applied partially and only on $\mathcal{V}$—the remaining part is a vector belonging to $\mathcal{W}$. This is depicted in a quantum circuit model as



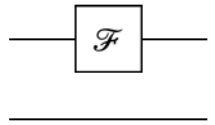
(B.31)

where the quantum circuit element —⋈— represents the projection onto the state specified at the output port. Interestingly, the result is not affected whether the projection onto $|\psi^*\rangle$ is made before or after the operation $\hat{A}$. This does not violate any physiccal principle as the two parts in $\mathcal{V}\otimes\mathcal{V}$ are separated spacelike.

Now let us turn to the Choi isomorphism between supermaps and operators: Consider again the maximally entangled state in Eq. (B.26) and operate an extended supermap $\mathscr{F}\otimes\mathcal{I}:\mathcal{L}(\mathcal{V}\otimes\mathcal{V})\to\mathcal{L}(\mathcal{W}\otimes\mathcal{V})$ on $|\Phi\rangle\langle\Phi|$ to get

$$\begin{aligned}\hat{C}_{\mathscr{F}} := (\mathscr{F}\otimes\mathscr{I})(|\Phi\rangle\langle\Phi|) &= \sum_{kl}\mathscr{F}(|v_k\rangle\langle v_l|)\otimes|v_k\rangle\langle v_l| \\ &= \sum_{ij}\sum_{kl}|w_i v_k\rangle\langle w_j v_l|\,C_{ik;jl}\,, \end{aligned} \tag{B.32}$$

where C is the Choi matrix of $\mathscr{F}$; see Eq. (B.9). This is depicted in a quantum circuit model as

$$\hat{C}_{\mathscr{F}} = |\Phi\rangle \left\{ \right.$$


(B.33)

Clearly $\hat{C}_{\mathscr{F}}$ is an operator (not a superoperator) on $\mathcal{W}\otimes\mathcal{V}$ and the Choi matrix C is nothing but its matrix representation in the standard tensor-product basis. Hence $\hat{C}_{\mathscr{F}}$ is called the *Choi operator* associated with $\mathscr{F}$. It turns out that the correspondence $\mathscr{F}\leftrightarrow\hat{C}_{\mathscr{F}}$ by means of (B.32) is one-to-one and an isomorphism.[3] For any state $|\psi\rangle\in\mathcal{V}$, the effect $\mathscr{F}(|\psi\rangle\langle\psi|)$ of supermap $\mathscr{F}$ on the pure state can be obtained by means of the conjugate state $|\psi^*\rangle$ [see Eq. (B.29)] as

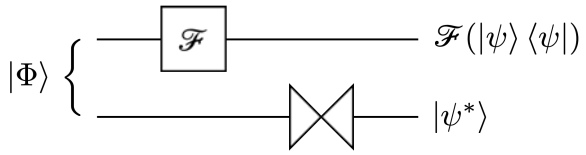
$$\mathscr{F}(|\psi\rangle\langle\psi|) = \langle\psi^*|\,\hat{C}_{\mathscr{F}}\,|\psi^*\rangle\,, \tag{B.34}$$

or, more generally, for any $\hat{\rho}=\sum_{ij}|v_i\rangle\langle v_j|\,\rho_{ij}$

$$\mathscr{F}(\hat{\rho}) = \operatorname*{Tr}_{\mathcal{V}}\hat{\rho}^*\hat{C}_{\mathscr{F}}\,, \tag{B.35}$$

[3]Here we have just defined an association $\mathscr{F}\mapsto\hat{C}_{\mathscr{F}}$. Given an operator $\hat{C}\in\mathcal{L}(\mathcal{W}\otimes\mathcal{V})$, one can also find the corresponding supermap $\mathscr{F}_{\hat{C}}$, that is, the association $\hat{C}\to\mathscr{F}_{\hat{C}}$ in the reverse direction; see Størmer (2013).

where $\hat{\rho}^* := \sum_{ij} |v_i\rangle \langle v_j| \rho_{ij}^*$. This has been depicted in a quantum circuit model



(B.36)

Further, taking the matrix representation of each entity in the relation (B.35) confirms the linear transformation rule (B.11) between the matrix elements of $\hat{\rho}$ and $\hat{\sigma} := \mathscr{F}(\hat{\rho})$ The transformation rule in (B.11) is another illustration of the Choi isomorphism.

The Choi operator plays a key role in the *gate teleportation* protocol, and it provides an interesting proof of the Kraus-representation theorem (see Section 5.1).

B.3 Partial Trace

Tensor product (see Appendix A.5) extends vectors and operators. Partial trace is effectively an inverse procedure and reduces operators on a tensor-product space to one of component spaces. Consider an operator $\hat{C}$ on a tensor product space $\mathcal{V} \otimes \mathcal{W}$. The partial trace over the space $\mathcal{W}$ is defined by

$$\hat{A} = \operatorname*{Tr}_{\mathcal{W}} \hat{C} = \sum_j \langle w_j| \hat{C} |w_j\rangle \tag{B.37}$$

The procedure is said to *trace out* the vector space $\mathcal{W}$, and the resulting operator $\hat{A}$ is called a *reduced operator* of $\hat{C}$. Given the matrix representation $C_{ij;kl}$ of $\hat{C}$ in the standard product basis

$$\hat{C} |v_k w_l\rangle = \sum_{ij} |v_i w_j\rangle C_{ij;kl}, \tag{B.38}$$

one can obtain the matrix representation of the reduced operator $\hat{A}$ in the basis $\{|v_i\rangle\}$ by

$$A_{ik} = \sum_j C_{ij;kj}\,. \tag{B.39}$$

Consider an operator A on a three-qubit Hilbert space. Suppose that an 8x matrix A is its matrix representation.

```
In[•]:= A = Re@RandomMatrix[8];
        A[[ ;; 5, ;; 5]] // MatrixForm
```

Out[•]//MatrixForm=

0.748769	0.633463	0.637068	0.831058	0.0608231
-0.15509	-0.511914	-0.768065	-0.994727	0.973921
-0.772355	0.33461	-0.861444	0.0559188	0.0583494
-0.235148	-0.959995	0.673298	-0.728021	-0.65737
-0.605041	0.584033	0.906122	-0.787569	-0.509827

This is the reduced matrix after tracing out the second and third qubits.

```
In[•]:= redA1 = PartialTrace[A, {2, 3}];
        redA1 // MatrixForm
Out[•]//MatrixForm=
        ( -1.35261   -2.02621 )
        ( -0.528031  0.439414 )
```

This traces out the second qubit.

```
In[•]:= redA2 = PartialTrace[A, {2}];
        redA2 // MatrixForm
Out[•]//MatrixForm=
        ( -0.446217  -0.0737435  -0.0971918   1.01527  )
        ( -1.01302    0.155287   -0.224515   -1.16846  )
        ( 0.331291    0.216007    0.112729    1.26334  )
        ( 0.736839    0.0580864  -0.329636   -0.0437415 )
```

Example 12 Show that partial trace is a *completely positive* supermap (see Definition 24).

One can prove it simply by constructing an operator-sum representation as in Theorem 25. Consider the partial trace over the subspace $\mathcal{W}$. According to the definition of the partial trace, we write

$$\operatorname*{Tr}_{\mathcal{W}} \hat{C} = \sum_{ijk} |v_i\rangle \langle v_i w_j| \hat{C} |v_k w_j\rangle \langle v_k| \tag{B.40}$$

It means that defining $\hat{F}_j = \sum_i |v_i\rangle \langle v_i w_j|$ the partial trace can be expressed as

$$\operatorname*{Tr}_{\mathcal{W}} \hat{C} = \sum_j \hat{F}_j \hat{C} \hat{F}_j^\dagger \,, \tag{B.41}$$

which proves the claim.

B.4 Partial Transposition

We conclude this appendix with a rather unusual mathematical tool—the *partial transposition*. As the name suggests, it applies the matrix transposition to the part of the matrix representation of an operator which corresponds to a subsystem. The resulting matrix gives a new operator associated with it. Roughly speaking, the partial transformation would correspond to a time-reversal transformation only on the subsystem.[4]

Like the (full) transposition, the partial transposition *cannot* be a *linear* supermap. Nevertheless, the partial transposition has attracted considerable attention thanks to the seminal work on the separability test of mixed states of a composite quantum system by Peres (1996) and Horodecki *et al.* (1996). Since

[4] Rigorously speaking, this statement is wrong because no anti-unitary transformation such as time reversal can be applied on a subpart of the system.

then, it has been widely used for the study of the structure of the tensor-product space of composite systems with respect to various entanglement properties.

Consider an operator $\hat{A}$ on a tensor product space $\mathcal{V} \otimes \mathcal{W}$ with the matrix representation

$$\hat{A} = \sum_{ij;kl} |v_i w_j\rangle \langle v_k w_l| A_{ij;kl} \tag{B.42}$$

in the standard tensor-product basis, $\{|v_i w_j\rangle \equiv |v_i\rangle \otimes |w_j\rangle\}$. The *partial transpose* of $\hat{A}^{T_\mathcal{W}}$ with respect to the subspace $\mathcal{W}$ is defined by

$$\hat{A}^{T_\mathcal{W}} := \sum_{ij;kl} |v_i w_j\rangle \langle v_k w_l| A_{il;kj} \,. \tag{B.43}$$

In a fixed basis, the partial transposition is entirely defined by the matrix representations. For the above case,

$$A^{T_\mathcal{W}}_{ij;kl} = A_{il;kj} \,. \tag{B.44}$$

It is important to remember that the partial transposition is basis-dependent.

Consider a system consisting of two subsystem with Hilbert-space dimensions 2 and 3, respectively. A and B are matrix representations of operators on two respective systems.

```
In[•]:= Let[Species, a, b]
        A = Array[a, {2, 2}];
        A // MatrixForm
        B = Array[b, {3, 3}];
        B // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

Let us take a special case -- an operator with a single term of tensor product.

```
In[•]:= matAB = CircleTimes[A, B];
        matAB // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} a_{1,1} b_{1,1} & a_{1,1} b_{1,2} & a_{1,1} b_{1,3} & a_{1,2} b_{1,1} & a_{1,2} b_{1,2} & a_{1,2} b_{1,3} \\ a_{1,1} b_{2,1} & a_{1,1} b_{2,2} & a_{1,1} b_{2,3} & a_{1,2} b_{2,1} & a_{1,2} b_{2,2} & a_{1,2} b_{2,3} \\ a_{1,1} b_{3,1} & a_{1,1} b_{3,2} & a_{1,1} b_{3,3} & a_{1,2} b_{3,1} & a_{1,2} b_{3,2} & a_{1,2} b_{3,3} \\ a_{2,1} b_{1,1} & a_{2,1} b_{1,2} & a_{2,1} b_{1,3} & a_{2,2} b_{1,1} & a_{2,2} b_{1,2} & a_{2,2} b_{1,3} \\ a_{2,1} b_{2,1} & a_{2,1} b_{2,2} & a_{2,1} b_{2,3} & a_{2,2} b_{2,1} & a_{2,2} b_{2,2} & a_{2,2} b_{2,3} \\ a_{2,1} b_{3,1} & a_{2,1} b_{3,2} & a_{2,1} b_{3,3} & a_{2,2} b_{3,1} & a_{2,2} b_{3,2} & a_{2,2} b_{3,3} \end{pmatrix}$$

This is the partial trace of matAB with respect to the second subsystem.

```
In[•]:= new = PartialTranspose[matAB, {2, 3}, {2}];
        new // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} a_{1,1}\,b_{1,1} & a_{1,1}\,b_{2,1} & a_{1,1}\,b_{3,1} & a_{1,2}\,b_{1,1} & a_{1,2}\,b_{2,1} & a_{1,2}\,b_{3,1} \\ a_{1,1}\,b_{1,2} & a_{1,1}\,b_{2,2} & a_{1,1}\,b_{3,2} & a_{1,2}\,b_{1,2} & a_{1,2}\,b_{2,2} & a_{1,2}\,b_{3,2} \\ a_{1,1}\,b_{1,3} & a_{1,1}\,b_{2,3} & a_{1,1}\,b_{3,3} & a_{1,2}\,b_{1,3} & a_{1,2}\,b_{2,3} & a_{1,2}\,b_{3,3} \\ a_{2,1}\,b_{1,1} & a_{2,1}\,b_{2,1} & a_{2,1}\,b_{3,1} & a_{2,2}\,b_{1,1} & a_{2,2}\,b_{2,1} & a_{2,2}\,b_{3,1} \\ a_{2,1}\,b_{1,2} & a_{2,1}\,b_{2,2} & a_{2,1}\,b_{3,2} & a_{2,2}\,b_{1,2} & a_{2,2}\,b_{2,2} & a_{2,2}\,b_{3,2} \\ a_{2,1}\,b_{1,3} & a_{2,1}\,b_{2,3} & a_{2,1}\,b_{3,3} & a_{2,2}\,b_{1,3} & a_{2,2}\,b_{2,3} & a_{2,2}\,b_{3,3} \end{pmatrix}$$

Take close look at individual blocks.

```
In[•]:= new[[1 ;; 3, 1 ;; 3]] // MatrixForm
        new[[1 ;; 3, 1 + 3 ;; 3 + 3]] // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} a_{1,1}\,b_{1,1} & a_{1,1}\,b_{2,1} & a_{1,1}\,b_{3,1} \\ a_{1,1}\,b_{1,2} & a_{1,1}\,b_{2,2} & a_{1,1}\,b_{3,2} \\ a_{1,1}\,b_{1,3} & a_{1,1}\,b_{2,3} & a_{1,1}\,b_{3,3} \end{pmatrix}$$

Out[•]//MatrixForm=

$$\begin{pmatrix} a_{1,2}\,b_{1,1} & a_{1,2}\,b_{2,1} & a_{1,2}\,b_{3,1} \\ a_{1,2}\,b_{1,2} & a_{1,2}\,b_{2,2} & a_{1,2}\,b_{3,2} \\ a_{1,2}\,b_{1,3} & a_{1,2}\,b_{2,3} & a_{1,2}\,b_{3,3} \end{pmatrix}$$

This is the partial trace with respect to the first subsystem.

```
In[•]:= new = PartialTranspose[matAB, {2, 3}, {1}];
        new // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} a_{1,1}\,b_{1,1} & a_{1,1}\,b_{1,2} & a_{1,1}\,b_{1,3} & a_{2,1}\,b_{1,1} & a_{2,1}\,b_{1,2} & a_{2,1}\,b_{1,3} \\ a_{1,1}\,b_{2,1} & a_{1,1}\,b_{2,2} & a_{1,1}\,b_{2,3} & a_{2,1}\,b_{2,1} & a_{2,1}\,b_{2,2} & a_{2,1}\,b_{2,3} \\ a_{1,1}\,b_{3,1} & a_{1,1}\,b_{3,2} & a_{1,1}\,b_{3,3} & a_{2,1}\,b_{3,1} & a_{2,1}\,b_{3,2} & a_{2,1}\,b_{3,3} \\ a_{1,2}\,b_{1,1} & a_{1,2}\,b_{1,2} & a_{1,2}\,b_{1,3} & a_{2,2}\,b_{1,1} & a_{2,2}\,b_{1,2} & a_{2,2}\,b_{1,3} \\ a_{1,2}\,b_{2,1} & a_{1,2}\,b_{2,2} & a_{1,2}\,b_{2,3} & a_{2,2}\,b_{2,1} & a_{2,2}\,b_{2,2} & a_{2,2}\,b_{2,3} \\ a_{1,2}\,b_{3,1} & a_{1,2}\,b_{3,2} & a_{1,2}\,b_{3,3} & a_{2,2}\,b_{3,1} & a_{2,2}\,b_{3,2} & a_{2,2}\,b_{3,3} \end{pmatrix}$$

Appendix C

Mathematica Application Q3

- March 6, 2021 (v1.4)

Q3 is a Mathematica application to help study quantum information processing, quantum many-body systems, and quantum spin systems. It provides various tools and utilities for symbolic and numerical calculations in these areas of quantum physics.

Q3 consists of several packages at different levels. Quisso, Fock, and Wigner are the three main packages, and they are devoted to the simulation of quantum information processing, quantum many-body systems, and quantum spin systems, respectively. They are based on two other lower-level packages, Pauli and Cauchy. Pauli itself provides useful tools to handle Pauli operators directly, but it also defines programming structures and objects for the aforementioned three and other higher-level packages. Cauchy, at the lowest level, defines the programming structure of the whole application. But it can also be used individually to facilitate complex analysis.

Q3 is distributed through the GitHub respository:

https://github.com/quantum-mob/Q3App

C.1 Installation

Q3 provides two installation methods: The first is based on the paclet system that has recently been introduced by Wolfram Research. It is not only fully automatic but also convenient to get updates later on. The other traditional method is to download and copy the files to a proper folder – just the traditional method. Take a look at the *installation guide* at:

https://github.com/quantum-mob/Q3App/blob/main/INSTALL.md

C.2 Quick Start

Once the application is installed, put

```
"Q3" or "Q3/guide/Q3"
```

in the search field of the Wolfram Language Documentation Center (Mathematica help window) to get detailed technical information about the application.

Note that after installing the application, the first time you search a keyword in Wolfram Language Documentation Center (help window), Mathematica builds the search index of the new documentation files. It can take a few seconds to minutes depending on your computer. It happens only once (everytime you update the application), though.

Bibliography

Aharonov, Y. & J. Anandan, Phys. Rev. Lett. **58 (16)**, 1593 (1987). "Phase Change During a Cyclic Quantum Evolution". DOI: 10.1103/PhysRevLett.58.1593

Anandan, J., Physics Letters A **133 (4-5)**, 171 (1988). "Non-adiabatic non-abelian geometric phase". DOI: 10.1016/0375-9601(88)91010-9

Aspect, A., P. Grangier, & G. Roger, Phys. Rev. Lett. **47 (7)**, 460 (1981). "Experimental Tests of Realistic Local Theories via Bell's Theorem".

Barenco, A., C. H. Bennett, R. Cleve, *et al.*, Physical Review A **52 (5)**, 3457 (1995). "Elementary gates for quantum computation". DOI: 10.1103/physreva.52.3457 arXiv:quant-ph/9503016

Bell, J. S., Rev. Mod. Phys. **38 (3)**, 447 (1966). "On the Problem of Hidden Variables in Quantum Mechanics".

Bennett, C. H. & S. J. Wiesner, Phys. Rev. Lett. **69 (20)**, 2881 (1992). "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states".

Bergou, J. A., U. Herzog, & M. Hillery, "Discrimination of Quantum States," in Paris & Rehacek (2004), Chap. 11, pp. 417–465. DOI: 10.1007/978-3-540-44481-7_11

Bernstein, E. & U. Vazirani, "Quantum Complexity Theory," in *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing* (ACM Press, New York, 1993), pp. 11–20.

Bernstein, E. & U. Vazirani, SIAM Journal on Computing **26 (5)**, 1411 (1997). "Quantum Complexity Theory". DOI: 10.1137/s0097539796300921

Berry, M. V., Proc. R. Soc. London A **392**, 45 (1984). "Quantal Phase Factors Accompanying Adiabatic Changes".

Blum, K., *Density Matrix Theory and Applications*, Vol. 64 of *Springer Series on Atomic, Optical, and Plasma Physics* (Springer Berlin Heidelberg, 2012), 3rd ed., ISBN 978-3-642-20560-6.

Born, M., Z. Phys. **37 (12)**, 863 (1926). "Zur Quantenmechanik der Stoßvorgänge".

Breuer, H.-P. & F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, New York, 2002).

Caves, C. M., Phys. Rev. D **23 (8)**, 1693 (1981). "Quantum-mechanical noise in an interferometer".

Chefles, A., "Quantum States: Discrimination and Classical Information Transmission.A Review of Experimental Progress," in Paris & Rehacek (2004), Chap. 12, pp. 467–511. DOI: 10.1007/978-3-540-44481-7_12

Chiaverini, J., Science **308 (5724)**, 997 (2005). "Implementation of the Semiclassical Quantum Fourier Transform in a Scalable System". DOI: 10.1126/science.1110335

Choi, M.-S., J. Phys.: Condens. Matt. **15 (46)**, 7823 (2003). "Geometric Quantum Computation in Solid-State Qubits". arXiv:quant-ph/0111019

Cleve, R., A. Ekert, C. Macchiavello, & M. Mosca, Proceedings of the Royal Society A **454 (1969)**, 339 (1998). "Quantum algorithms revisited". DOI: 10.1098/rspa.1998.0164 arXiv:quant-ph/9708016

Deutsch, D., Proc. R. Soc. London A **400**, 97 (1985). "Quantum theory, the Church-Turing principle and the universal quantum computer". DOI: 10.1098/rspa.1985.0070

Deutsch, D. & R. Jozsa, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **439 (1907)**, 553 (1992). "Rapid Solution of Problems by Quantum Computation". DOI: 10.1098/rspa.1992.0167

DiVincenzo, D. P., Fortschr. Phys. **48**, 771 (2000). "The Physical Implementation of Quantum Computation". DOI: 10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E arXiv:quant-ph/0002077

Dum, R., A. S. Parkins, P. Zoller, & C. W. Gardiner, Phys. Rev. A **46 (7)**, 4382 (1992). "Monte Carlo simulation of master equations in quantum optics for vacuum, thermal, and squeezed reservoirs".

Einstein, A., B. Podolsky, & N. Rosen, Phys. Rev. **47**, 777 (1935). "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?"

Giovannetti, V., S. Lloyd, & L. Maccone, Physical Review Letters **96 (1)**, 010401 (2006). "Quantum Metrology". DOI: 10.1103/PhysRevLett.96.010401 arXiv:quant-ph/0509179

Griffiths, R. B. & C.-S. Niu, Physical Review Letters **76 (17)**, 3228 (1996). "Semiclassical Fourier Transform for Quantum Computation". DOI: 10.1103/physrevlett.76.3228 arXiv:quant-ph/9511007

Hardy, L., Phys. Rev. Lett. **68 (20)**, 2981 (1992). "Quantum Mechanics, Local Realistic Theories, and Lorentz-Invariant Realistic Theories".

Higgins, B. L., D. W. Berry, S. D. Bartlett, H. M. Wiseman, & G. J. Pryde, Nature **450 (7168)**, 393 (2007). "Entanglement-free Heisenberg-limited phase estimation". DOI: 10.1038/nature06257 arXiv:0709.2996

Horodecki, M., P. Horodecki, & R. Horodecki, Phys. Lett. A **223 (1)**, 1 (1996). "Separability of mixed states: necessary and sufficient conditions". DOI: 10.1016/0375-9601(95)00930-2

Jiang, M., S. Luo, & S. Fu, Physical Review A **87 (2)** (2013). "Channel-state duality". DOI: 10.1103/physreva.87.022310

Kitaev, A. Y., Electronic Colloquium on Computational Complexity **3**, 3 (1996). "Quantum measurements and the Abelian Stabilizer Problem". arXiv:quant-ph/9511026

Kitaev, A. Y., Russian Mathematical Surveys **52 (6)**, 1191 (1997). "Quantum computations: algorithms and error correction".

Lang, S., *Introduction to Linear Algebra*, Undergraduate Texts in Mathematics (Springer New York, New York, 1986), 2nd ed., ISBN 9781461210702. DOI: 10.1007/978-1-4612-1070-2

Lang, S., *Linear Algebra* (Springer, Berlin, 1987), 3rd ed., ISBN 978-1-4757-1949-9. DOI: 10.1007/978-1-4757-1949-9

Loss, D. & D. P. DiVincenzo, Phys. Rev. A **57 (1)**, 120 (1998). "Quantum comutation with quantum dots".

Lundeen, J. S., B. Sutherland, A. Patel, C. Stewart, & C. Bamber, Nature **474 (7350)**, 188 (2011). "Direct measurement of the quantum wavefunction". DOI: 10.1038/nature10120

Nakazato, H., Y. Hida, K. Yuasa, B. Militello, A. Napoli, & A. Messina, Physical Review A **74 (6)**, 062113 (2006). "Solution of the Lindblad equation in the Kraus representation". DOI: 10.1103/physreva.74.062113 arXiv:quant-ph/0606193

Nielsen, M. & I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, New York, 2011), 10th anniversary ed., ISBN 978-1107002173.

Paris, M. & J. Rehacek, eds., *Quantum State Estimation*, Vol. 649 of *Lecture Notes in Physics* (Springer Berlin Heidelberg, Berlin, 2004), ISBN 9783540444817. DOI: 10.1007/b98673

Peres, A., Phys. Rev. Lett. **77 (8)**, 1413 (1996). "Separability Criterion for Density Matrices". DOI: 10.1103/PhysRevLett.77.1413 arXiv:quant-ph/9604005

Plenio, M. B. & P. L. Knight, Rev. Mod. Phys. **70 (1)**, 101 (1998). "The quantum-jump approach to dissipative dynamics in quantum optics".

Raussendorf, R. & H. J. Briegel, Phys. Rev. Lett. **86 (22)**, 5188 (2001). "A One-Way Quantum Computer".

Raussendorf, R., D. Browne, & H. Briegel, Journal of Modern Optics **49 (8)**, 1299 (2002). "The one-way quantum computer–a non-network model of quantum computation". DOI: 10.1080/09500340110107487 arXiv:quant-ph/0108118

Raussendorf, R., D. E. Browne, & H. J. Briegel, Phys. Rev. A **68 (2)**, 022312 (2003). "Measurement-based quantum computation on cluster states". DOI: 10.1103/PhysRevA.68.022312 arXiv:quant-ph/0301052

Shor, P. W., "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, USA, 1994), SFCS '94, pp. 124–134. DOI: 10.1109/SFCS.1994.365700

Shor, P. W., SIAM Journal on Computing **26 (5)**, 1484 (1997). "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". arXiv:quant-ph/9508027

Simon, D. R., SIAM Journal on Computing **26 (5)**, 1474 (1997). "On the Power of Quantum Computation". DOI: 10.1137/s0097539796298637

Sjöqvist, E., D. M. Tong, L. Mauritz Andersson, B. Hessmo, M. Johansson, & K. Singh, New Journal of Physics **14 (10)**, 103035 (2012). "Non-adiabatic holonomic quantum computation". DOI: 10.1088/1367-2630/14/10/103035 arXiv:1107.5127

Smolin, J. A. & D. P. DiVincenzo, Phys. Rev. A **53 (4)**, 2855 (1996). "Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate". DOI: 10.1103/PhysRevA.53.2855

Størmer, E., *Positive Linear Maps of Operator Algebras* (Springer, Berlin, 2013), ISBN 9783642343698. DOI: 10.1007/978-3-642-34369-8

Vallone, G. & D. Dequal, Physical Review Letters **116 (4)**, 040502 (2016). "Strong Measurements Give a Better Direct Measurement of the Quantum Wave Function". DOI: 10.1103/physrevlett.116.040502 arXiv:1504.06551

Wilczek, F. & A. Zee, Phys. Rev. Lett. **52 (24)**, 2111 (1984). "Appearance of Gauge Structure in Simple Dynamical Systems". DOI: 10.1103/PhysRevLett.52.2111

Wilmut, I., A. E. Schnieke, J. McWhir, A. J. Kind, & K. H. S. Campbell, Nature **385 (6619)**, 810 (1997). "Viable offspring derived from fetal and adult mammalian cells".

Wooters, W. K. & W. H. Zurek, Nature **299**, 802 (1982). "A single quantum cannot be cloned".

Zanardi, P. & M. Rasetti, Phys. Lett. A **264 (2-3)**, 94 (1999). "Holonomic quantum computation". DOI: 10.1016/S0375-9601(99)00803-8 arXiv:quant-ph/9904011

Zurek, W. H., Phys. Today **44 (10)**, 36 (1991). "Decoherence and the transition from quantum to classical".

Zurek, W. H., Nature **404**, 130 (2000). "Quantum cloning: Schrodinger's sheep". DOI: 10.1038/35004684

Zurek, W. H., Los Alamos Science **27**, 2 (2002). "Decoherence and the Transition from Quantum to Classical: Revisited".

Index

amplitude damping, 183
ancillary qubit, 76

Bell basis, *see also* Bell states, 125
Bell measurement, 85, 125
Bell states, 125
Bell's inequality, 124
Bell's test, 124
Bernstein-Vazirani algorithm, 127, 136
birthday paradox, 137
bit flip, 40
bitwise AND, 74
Bloch sphere, 18, 19
Bloch vector, 18
bra-ket notation, 38
bright state, 106

channel-state duality, 172
Choi isomorphism, 171, 172, 227
Choi matrix, 171, 221, *see also* Choi operator
Choi operator, 171, 192, 226
Choi vector, 173
classical communication channel, 125
closed system, 10, 163, 164
cluster state, 87, 107, 115
CNOT, 48, 79, 125
    controlled-NOT gate, 48
    multi-qubit controlled-NOT, 75
CNOT gate, 56, 100, 107
complementarity principle, 9
completely positive supermap, 164, 165, 171, 172, 223, 224
completeness relation, 31, 32, 208
compuational basis, *see also* logical basis
control qubit, 49
controlled unitary gate, 152, 153
controlled-U gate, 48, 161
    multi-qubit controlled-U gate, 62, 73
convex linear, 165
cyclic evolution, 104, 106
CZ, 54
    controlled-Z gate, 54
CZ gate, 100, 101, 118

damping operator, 180, 184, 190
dark state, 106, 118
decoherence, 165
density matrix, 15
density operator, 15, 163, 164, 171, 172, 186, 215
dephasing, 176
depolarizing process, 183
Deutsch-Jozsa algorithm, 127
    Deutsch-Jozsa problem, 137
discrete Fourier transform, 141, 142
discrete logarithm, 141
DiVincenzo criteria, 90

effective Hamiltonian, 180, 184, 190
elementary qunatum logic gates, 38
Elements, 9
entangled state, 13, 15, 50
entanglement, 125, 131
entanglement fidelity, 172
environment, 163
Euclid of Alexandria, 9
Euler rotation, 47, 111
    Euler angles, 47
exclusive OR, 51

factorization algorithm, 121, 137, 152, 160
fidelity, 172
flux quantization, 89
Fredkin gate, 79

gate teleportation, 173
gauge transformation, 105
generalized interaction picture, 190
geometric phase, 102
graph state, *see also* cluster state, 107, 114
Gray code, 62, 73
  Gray code sequence, 74

Hadamard gate, 42, 86, 87, 100, 125
  Hadamard matrix, 42
Hardy's test, 124
Heisenberg exchange interaction, 99
Heisenberg limit, 31
Hermitian operator, 164, 223
hidden subgroup problem, 141, 152, 158–160
hiddne subgroup problem, 141
Hilbert space, 10, 90

inertial force, 95
inertial frame, 94
initialization, 91
inverse quantum Fourier transform
  quantum Fourier transform, 147
irreversible population loss, 180
Ising exchange ineraction, 101
Ising exchange interaction, 100, 101

Josephson inductance, 89

kinetic inductance, 89
Kraus elements, 177, 182, 192, 224
  orthogonal Kraus elements, 224
Kraus maps, *see also* Kraus elements
Kraus operator-sum representation, *see also* Kraus representation
Kraus operators, *see also* Kraus elements
Kraus representation, 166, 224

Larmor precession, 93
Lindblad basis, 186
Lindblad equation, 180, 185
Lindblad generator, 180, 181, 187
Lindblad operators, 180, 181
logical basis, 11

Markov approximation, 179
Markov assumption, 181
maximally entangled, 87
maximally entangled state, 171
measurement, 37, 91
measurement operators, 32
measurement-based quantum computation, 107
mixed state, 15, 176
modular exponentiation, 158, 160
modular multiplcation, 154
modular multiplication, 160
momentum basis, 157

Newton's laws of motion, 9
no-cloning theorem, 124
non-Abelian gauge potential, 105
non-Hermitian Hamiltonian, 180, 190
non-inertial effect, 95, 104
non-negative operator, *see also* positive semidefinite operator
non-selective measurement, 32
nonlocality, 122
normal operator, 207

one-way quantum computation, *see also* measurement-based quantum computation
open quantum system, 163, 164
operation time, 93
operator-sum representation, 171, 222, 223
order-finding algorithm, 160
  order-finding problem, 141, 154, 160
orthonormal basis, 208

parallel transport, 105
path ordering, 105

Pauli gates, *see also* Pauli operators
Pauli operator, 118
Pauli operators, 38, 87, 91, 125
    Pauli X, 38
    Pauli Y, 40
    Pauli Z, 39
period-finding algorithm, 160
phase damping, 177, 183
phase flip, 40
phase gate, 86
pinciple of deffered measurement, 148
planar exchange interaction, *see also* XY exchange interaction
position basis, 157
positive definite operator, *see also* positive operator
positive operator, 164, 172, 203, 207, 208, 223
positive semidefinite operator, 166, 203, 207, 208
postulates of quantum mechanics, 9
projection operator, 172, 226

quantum entanglement, 15
quantum channel, 165
quantum circuit model
    quantum circuit diagram, 37
quantum communication, 124
quantum computer, 89
    quantum computer architecture, 89
quantum decoherence, *see also* decoherence
quantum efficiency, 91
quantum entanglement, 31, 53
quantum entangler circuit, 50
quantum Fourier transform, 143, 153, 154, 156, 157
quantum gate teleportation, 173, 192
quantum information theory, 222
quantum jump approach, 180, 190
quantum jump operators, *see also* Lindblad operators, *see also* Lindblad operators
quantum logic gate, 48, 89
quantum logic gate operation, 37
quantum Markovian dynamics, 180
quantum master equation, *see also* Lindblad equation, 185
quantum non-demolition measurement, 116
quantum operation, 32, 164, 170, 192
quantum oracle, 128, 134, 138, 161
quantum parallelism, 127
quantum phase estimation, 141, 152, 158, 160
quantum register, 43, 51
quantum state, 10
quantum statistical mechanics, 222
quantum teleportation, 15, 53, 122, 124
qubit, 37, 90
    quantum bit, 37
qunatum phase estimation, 152

Rabi oscillation, 95, 104
    Rabi frequency, 95
reduced density matrix, 87
reference space, 171
resonance, 95
rorating-wave approximation, 98
rotating frame, 94
    Hamiltonian in the rotating frame, 95
    time-evolution operator in the rotating frame, 95
rotation, 46
rotation operator, 93

scalable system, 90
Schmidt decomposition, 13
Schmidt rank, 22
selective measurement, 32
separable state, 13
Shor, Peter W., 121
Simon's algorithm, 127, 137
    Simon's problem, 137
special theory of relativity, 124
spectral decomposition, 166, 208
spin-boson model, 116

standard quantum limit, 31
state vector, 10
statistical ensemble, 15
statistical mixture, 215
super-mapping, *see also* supermap
superdense coding, 122
supermap, 164, 170, 221, 223
superoperator, 180, 215, 221
SWAP, 55, 79
SWAP gate, 56, 99, 100, 118
    $\sqrt{\text{SWAP}}$ gate, 57, 100, 118

target qubit, 49
tensor-product basis, 13
tensor-product space, 13
time ordering, 104
Toffoli gate, 77, 79
trace Hermitian product, 166, 219
trace product, *see also* trace Hermitian product
translational freedom of Lindblad operators, 181
two-leve unitary transformation, 81
two-level unitary transformation, 65, 68, 75
    two-level unitary matrix, 71, 73

unconditional security, 124
unitary freedom of Kraus elements, 174, 181
unitary freedom of Lindblad operators, 181, 182
unitary group, 38
unitary matrix, 38
universal quantum computation, 38, 65, 73
    universal set of quantum gate operations, 91
    universal set of quantum logic gates, 81
universal set of classical logic gates, 81

vector space of linear maps, 218
vector space of linear operators, 19
von Neumann scheme of measurement, 156

wave-particle duality, 9

XOR, *see also* exclusive OR
XY exchange interaction, 100