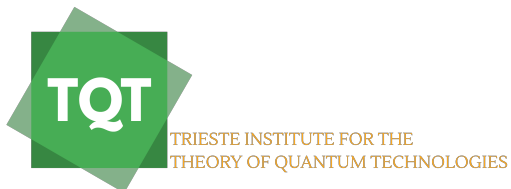




Introduction to Quantum Computation and Information
- Lecture Notes -

Giuseppe E. Santoro
SISSA, Trieste

Academic year 2023-2024
- work in progress -
Printed on March 25, 2024



Preface

These are the lecture notes prepared for a course held at SISSA, starting in the spring of 2021.

The main references I have consulted are: The book by David Mermin, *Quantum Computer Science* [1], the book by Benenti, Casati, Rossini and Strini *Principles of Quantum Computation and Information* [2], the book by Nielsen and Chuang *Quantum Computation and Quantum Information* [3], Scott Aaronson's [lecture notes](#), John Preskill's [course](#) at CalTech, and Stefano Olivares' [lecture notes](#), all available on the web (click on the link).

All you need in this life is ignorance and confidence; then success is sure.

Mark Twain

These are my principles, and if you don't like them... well, I have others.

Groucho Marx

Contents

1. Introduction	7
1.1. Turing machines and classical computation	9
1.1.1. Computability and decidability	13
1.2. Probability theory and Quantum Mechanics	15
1.2.1. Interference in a Mach-Zehnder interferometer	16
1.2.2. Wheeler’s delayed-choice experiment	20
1.2.3. Which-way experiments and the delayed-choice quantum eraser	21
1.3. Concluding remarks	24
1.4. Hands-on: EPR-type calculations with entangled particles	24
2. Classical gates and elements of classical computation	29
2.1. Classical bits, probability distributions and Stochastic Matrices	29
2.2. More than one Cbit: tensor products	31
2.3. More on Cbit operations: connection to digital computer operations	33
2.4. Reversible extensions of Boolean functions	38
2.5. Elementary logic gates	40
2.6. A simple algorithm: adding numbers	41
2.7. Universal classical gates	42
2.8. Universality vs Efficiency: Tractable vs Intractable problems	45
2.9. Boolean Satisfiability	47
3. Quantum gates and elements of quantum computation	49
3.1. Computational states and superpositions: the Hilbert space	50
3.2. Unitary operators associated to function evaluation	51
3.3. Pauli operators and associated single-Qbit unitary gates	53
3.4. The Hadamard gate \mathbf{H}	55
3.4.1. Using only Hadamard and rotations around the z-axis	57
3.5. Drawing quantum circuits	58
3.6. Two-Qbit states and gates	60
3.6.1. Bell measurements	62
3.7. NMR-like Hamiltonian model for 1- and 2-Qbit gates	63
3.8. A variety of 2-Qbit and multi-Qbit unitary gates.	64
3.8.1. Multi-Qbit unitary gates	65
3.9. Universal quantum gates	66
3.10. Examples of function evaluation with a QC	66
3.10.1. The quantum adder	69
3.11. No-cloning theorem	70
3.12. The Deutsch’s problem	71
3.12.1. An interesting “variant” of Deutsch’s problem, and some general remarks on the role of additional Qbits.	74
3.13. The Bernstein-Vazirani problem	75
3.14. Teleportation	78
3.15. Hands-on: state preparation, control- \mathbf{U} and Toffoli gates	80
3.15.1. Representing a general 2-Qbit state	80

3.15.2. Constructing control-unitary operators	81
3.15.3. Constructing the Toffoli gate out of cNOTs	82
4. Grover searching with a quantum computer	85
4.1. The Grover iteration	86
4.2. How to construct the kinetic term \mathbf{K}	89
4.3. Generalisation to the case of several solutions	91
4.4. Connection to p-spin models and to QAOA	92
5. Quantum Fourier Transform	95
5.1. The Quantum Fourier Transform circuit	98
5.2. Period-finding	102
5.3. Factoring and cryptography	108
5.3.1. Modular arithmetics: some tools.	109
5.3.2. RSA public-key cryptography	111
5.3.3. Breaking RSA through period-finding	113
5.3.4. Period-finding and factoring	114
5.3.5. Implementing modular exponentials on a Quantum Computer	116
5.4. Phase estimation protocol	118
5.5. Finding eigenstates and eigenvalues of an Hamiltonian	120
6. Quantum cryptography	125
6.1. To be sure: the Vernam cypher	125
6.2. Implementing Qbits with photon polarisation	130
6.3. Exploiting the special nature of Quantum Randomness	134
6.3.1. The BB84 protocol	135
6.3.2. Important details	136
6.4. Exploiting quantum correlations due to entanglement	136
6.4.1. CHSH version of Bell's inequalities	137
6.4.2. The E91 protocol	139
7. Hardware implementations of Quantum Computers	141
7.1. DiVincenzo criteria	142
7.2. A few tools: LC circuits, Josephson's Junctions, SQUIDS	142
7.2.1. From BCS to the Josephson junction	143
7.3. The superconducting Qbits platforms	149
7.3.1. Charge Qbits: The Cooper pair box	149
7.3.2. The transmon	151
7.4. Variants of JJ Qbits	154
7.5. Manipulating and coupling superconducting QBits	155
7.5.1. Manipulating single Qbits	155
7.6. What can go wrong: the sources of dissipation and decoherence	155
7.7. Circuit QED	155
8. Density matrices	157
8.1. The density matrix for a pure state	157
8.2. The density matrix for a mixed state	159
8.3. Spectral properties of $\hat{\rho}$ and ambiguity on the ensemble originating $\hat{\rho}$	160
8.4. Density matrices after measurements	162
8.5. Density matrices in statistical mechanics	164
8.6. Density matrices by tracing out an environment	165
8.7. Schmidt decomposition	166
8.7.1. The singular value decomposition (SVD)	169

8.8. Convex nature of density matrices	170
8.9. The spin-1/2 case and the Bloch sphere	172
9. Open Quantum Systems and Quantum Maps	175
9.1. Kraus representation of the dynamics	175
9.2. Quantum measurements and POVM	178
9.2.1. von Neumann projective measurements	178
9.2.2. Generalised quantum measurements	180
9.2.3. Ambiguity in the preparation of a post-measurement state	182
9.2.4. The von Neumann protocol	184
9.2.5. POVM and summary of quantum measurement	186
9.3. Inverting Kraus: how to “invent” unitaries	187
9.4. Quantum maps	188
9.5. Ambiguity of the Kraus representation and purification	190
9.6. Composition laws of Quantum Maps	192
9.7. Useful examples of single-Qbit maps	194
9.7.1. Phase damping (or dephasing)	194
9.7.2. Amplitude damping (or relaxation)	196
9.7.3. Depolarising channel	199
10. Open Quantum Systems and Lindblad Quantum Master Equation	201
10.1. The Markovian condition	201
10.2. The Lindblad construction	203
11. Introduction to quantum error correction	207
11.1. Classical error correction and Shannon’s theorem	207
11.2. Quantum error correction: the simple case of bit flips	210
11.3. Measuring error syndromes: general idea	212
11.4. More general errors: error digitisation	214
11.5. The five-Qbit encoding	217
11.6. General criteria for quantum error correction	220
11.6.1. Content of the QEC criterion and the quantum Hamming bound	223
11.6.2. Digitization of quantum noise: again	224
11.7. The stabilizers and the Pauli group	224
11.8. Unitary transformations and the Clifford group	226
11.9. Stabilizer codes	228
11.9.1. Error correction for stabilizer codes	229
11.9.2. Syndrome detection for stabilizer codes	231
11.10. The Toric code	232
11.10.1. The toric code ground states	233
I. Appendices	237
A. Simple tools from arithmetics	239
A.1. The Euclid algorithm for the greatest common divisor	239
A.2. Finding the multiplicative inverse in modular arithmetics	240
A.3. The probability of two random integers being co-prime	241
B. Uniaxial birefringence	243
B.1. The wave-plate geometry	246
B.2. The double-refraction geometry	247
B.3. Quantum optics single-Qbit gates with photon polarisation	249

B.4. Hands-on: Peres' problems with calcite crystals	250
C. Superconductivity	251
C.1. The BCS problem	251
C.2. The Josephson effect	255
C.3. The Ginsburg-Landau description	258
C.4. Quantum interference of two JJ: The dc-SQUID	261
D. Quantum master equations	263
D.1. A general framework: system plus environment	263
D.2. The Bloch-Redfield quantum master equation	266
D.3. The secular approximation and the Lindblad form	268
D.3.1. Rotating-wave (or secular) approximation	269
D.3.2. The Lindblad form	271
D.3.3. Non-degenerate spectrum and population dynamics	271
D.4. Application to a two-level system	272
D.4.1. Lindblad form for the two-level system	274
D.4.2. Decoherence and relaxation towards equilibrium	275
E. Classical and Quantum Error Correction	277
E.1. Linear codes in classical error correction	277
E.1.1. Errors induced by the communication channel.	280
E.1.2. More about decoding: cosets and syndromes.	282
E.1.3. The binary Hamming code	284
E.1.4. The probability of error	286
E.1.5. Shannon's theorem: the existence of good codes	288
E.1.6. Dual codes	288
E.1.7. Construction of new codes from old ones	290
E.1.8. General properties of linear codes	290
E.2. Quantum codes	291
E.2.1. Calderbank-Shor-Steane (CSS) quantum codes	291
E.2.2. The CSS codes seen as stabilizers codes	294
E.3. Pauli group and stabilizers reloaded	296
E.3.1. Measurements in the Stabilizer formalism	301
E.3.2. The construction of logical \mathbf{X} and \mathbf{Z} for stabilizer codes	302
E.4. The Gottesman-Knill theorem	304

1. Introduction

Digital computers have created the “information age” in which we live. True that the basic building block of current digital computers is ultimately a quantum device — the transistor. But such smallish devices — by now having certainly less than 10^{11} atoms — are used classically: they conduct current if a certain bias voltage is applied, and this is a bit 1, or not, and this is a bit 0. Hence, each transistor acts as a two-state classical device, capable of coding a 0 or a 1. With n transistors, we can code the state of an n -bit classical computer as a string of n binary digits, formally $\{0, 1\}^n$: e.g., 01001100 is one of the 2^8 possible states of an $n = 8$ bit classical register.

Can quantum mechanics have an impact on the way we process and transmit information? You should not think of inevitably quantum effects when the size of each transistor gets smaller and smaller: this is only a nuisance, in some sense. We are asking a much more ambitious question: can we base a computing device entirely on the law of quantum mechanics (QM)? Would that be good, allowing us to solve some problems that are difficult to solve using classical digital computers?

One of the distinctive features of QM is the fact that one can form **superposition of states**. For a single spin-1/2, not only we can have states $|\uparrow\rangle$ and $|\downarrow\rangle$, which we could easily identify with the two states $|0\rangle = |\uparrow\rangle$ and $|1\rangle = |\downarrow\rangle$ needed by a digital device, but we can also construct spin states in any direction $\mathbf{n} = (\sin\theta \cos\phi, \sin\theta \sin\phi, \cos\theta)$:

$$|+, \mathbf{n}\rangle = \cos\frac{\theta}{2}|\uparrow\rangle + e^{i\phi}\sin\frac{\theta}{2}|\downarrow\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle. \quad (1.1)$$

So, the information encoded into a single such state is equivalent to dealing with \mathbb{C} rather than with the set $\{0, 1\}$. With n spins: \mathbb{C}^{2^n-1} — discounting for normalisation and an overall phase in front — rather than just the classical states in $\{0, 1\}^n$.

But, as Spider-Man would say, with great power comes great responsibility. One of the nice things about the small-but-not-too-small transistors of our digital devices is their relative insensitivity to thermal fluctuations or small electrical noise: it takes some effort to turn a bit 0 into a bit 1. Bit flip errors are important, of course, in classical communication through noisy channels, and this is what Shannon understood and taught us in 1948: classical information theory was born at that time. But if our “bits” become quantum — Qbits, from now on — they encode, surely, much more information, but they are much more prone to errors of all sorts: not only bit flips, but also small phase errors might accumulate during computation, and lead to disaster. Hence the absolute necessity of a Quantum Error Correction — a non-trivial fact, by itself, because QM tells us that we mess up our states, collapsing them if we make measurements to discover errors — in a fully scalable quantum computer.

Another feature of QM which is important is **entanglement**. Even far-away systems can be in states that are not simple product states, showing therefore hidden correlations that defeat our physical intuition based on *local realism*. In the prototypical re-interpretation of the EPR paradox given by Bohm, a two-particle spin-singlet state shared by two far-away stations A and B would look like:

$$|\psi_{\text{ent}}\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|\uparrow\rangle_A \otimes |\downarrow\rangle_B - |\downarrow\rangle_A \otimes |\uparrow\rangle_B\right). \quad (1.2)$$

Upon measuring the spin along the z-axis, if A finds it to be \uparrow , then the state of the system *collapses* and B would necessarily get \downarrow if measuring the spin along the same z-axis. But this is not the strange

side of the story.¹ What is strange is what happens when measurements in *different spin directions* are performed. This leads to violations of Bell's inequalities, which any theory based on local realism would obey. The final section of this introductory chapter is intended to provide you with a way of revising your QM, in case you might need, to appreciate a few of these remarkable QM facts. Experiments with entangled photons [4, 5] have confirmed that Nature behaves in such a quite weird manner.

Question: Superposition and entanglement are enough?

Is that all? These two ingredients, superposition of states and entanglement, are enough to lead to a new paradigm of computation which is indeed more powerful than the classical one behind our current digital computers?

Surprisingly, the answer to this question is negative. A famous result, known as Gottesman-Knill theorem [6, 7] shows that there are highly entangled and non-trivial quantum operations which can still be simulated efficiently on a classical computer.

❶

Gottesman-Knill theorem. More precisely, if the quantum algorithm is based on:

- 1) Preparation of states on the so-called computational basis. ^a
- 2) Application of any sequence of quantum gates based on single-Qbit Hadamard and Phase gate, plus a two-Qbit controlled-NOT gate. ^b
- 3) von-Neumann projective measurements in the computational basis.

then, a classical algorithm [7] which is polynomial in n can simulate it. The interesting fact, which is the reason why we still do Quantum Computation, is that these gates are *not universal*: there are possibly interesting Quantum Algorithm which do involve other gates that cannot be approximated by simply using Hadamard, Phase, and controlled-NOT. The addition of the **T**-gate, which is such that $\mathbf{T}^2 = \mathbf{S}$, would make this set universal, and, as far as we know, most likely impossible to simulate classically in an efficient way.

^aThe computational basis of an n -Qbit system is simply the tensor product basis $|\sigma_n\rangle \otimes \dots \otimes |\sigma_2\rangle \otimes |\sigma_1\rangle$ with $\sigma_j = \uparrow$ or \downarrow . In terms of Boolean variables $x_j = 0, 1$, we would write it as $|x_n\rangle \otimes \dots \otimes |x_2\rangle \otimes |x_1\rangle$.

^bThe Hadamard gate **H** is a single Qbit gate that implements $\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, hence it is represented on the computational basis by the following 2×2 matrix:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The phase gates **S** is a single-Qbit gate that acts by putting a phase factor i when acting on $|1\rangle$, hence it is represented by

$$\mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

The controlled-NOT two-Qbit gate implements on the computational basis the classical operation $\mathbf{C}_{12}|x_2\rangle \otimes |x_1\rangle = |x_1 + x_2, \text{ mod } 2\rangle \otimes |x_1\rangle$.

Nevertheless, while much has still to be learned about the possible power of a quantum paradigm of computation, there are great expectations in the news on what a Quantum Computer would allow us to do. To get a feeling for what we could dream to do with a Quantum Computer, let me briefly discuss, very superficially, what we can do with a classical computer.

¹Think of a photocopy of both sides of a coin, which is cut in two: one side is sent to you, the other to a friend of yours in Japan. Until the envelopes are closed, each of you might have a Tail or a Head. As soon as you receive your envelope, and see for instance a Tail, you might say that the figure that your friend receives instantaneously collapses into a Head. This is clear nonsense: there is nothing strange in such classical correlations.

1.1. Turing machines and classical computation

We start with the idea of an *algorithm* — a set of instructions to carry out a given task —, a notion known since ancient times (thinks of Euclid’s algorithm for the greatest common division) but made precise only surprising late, in the 1930s. Before that, mathematicians, including Hilbert in 1900, gave for granted that an algorithm exists, although possibly very difficult to find, for every mathematical task. We concentrate here on two kinds of tasks: *decision problems* and *computational problems*. Computational problem can be thought as devising an algorithm to calculate a function $f(\mathbf{x}^{\text{in}}) = \mathbf{y}^{\text{out}}$. Decision problems are, in some sense, a particular case, where the output \mathbf{y}^{out} is binary: True or False.

Decision problems. A classical important decision problem is to decide if a given statement, assuming a set of axioms, is True (hence a Theorem) or False. You can think of very easy decision problems, as we now exemplify.

- 1) Consider the set of strings $L = \{0, \dots, 0 \text{ } m \text{ times}\}$ made of m repeated 0s. Devise an algorithm that accepts the string if $m = 2^n$ with $n \geq 0$, a multiple of two, rejecting it otherwise. This is example 3.7 in Sipser [8].
- 2) Consider strings of the form $L = \{\underline{t} \# \underline{t} \mid \underline{t} \in \{0, 1\}^*\}$ where $\{0, 1\}^*$ denotes the set of Boolean strings of any length. Devise an algorithm that accepts strings of this form, while rejecting any other string build on the same set of symbols $\{0, 1, \#\}$. This is example 3.9 in Sipser [8].
- 3) **Element distinctness.** Consider strings of the form

$$L = \{\underline{t}_1 \# \underline{t}_2 \# \dots \# \underline{t}_n \mid \underline{t}_i \in \{0, 1\}^* \text{ with } \underline{t}_i \neq \underline{t}_j \text{ for } i \neq j\} .$$

Devise an algorithm that accepts strings of this form, i.e, all \underline{t}_i are *different*, while rejects any other string build on the same set of symbols $\{0, 1, \#\}$. This is example 3.12 in Sipser [8].

i

Existence of integral roots of integer polynomials. Consider polynomials $p(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^n$ with integer coefficients. Given the coefficients, devise an algorithm that tests if the polynomial has *integral roots*, i.e., $p(\mathbf{x}) = 0$ with $\mathbf{x} \in \mathbb{Z}^n$. This is a famous decision problem: Hilbert’s 10th problem in his famous address at the International Congress of Mathematicians, held in Paris in 1900.

For a polynomial of a single variable $p(x) = c_n x^n + \dots c_1 x + c_0$ an integral root $x = x_0$ is bound to be $|x_0| < (n + 1)|c_{\text{max}}/c_n|$, where c_{max} is the maximum coefficient, and an algorithm that tests if such an integral root exists can be easily devised: test all integers $0, \pm 1, \pm 2, \dots$ up to the previous bound. A mathematician, Yuri Matijasevič, has shown in 1970 that no such algorithm exists for a general polynomial of many variables.

To do that, however, we need a precise formal definition of “*what an algorithm is*”, which came only in the 1930s, through the work of Alan Turing — who introduced the concept of Turing Machine — and Alonso Church — who described as computable those functions that one can formally describe as *recursive* and invented the *lambda-calculus*. These two approaches have been shown (by Turing) to be computationally equivalent, leading to the following:

1

The Church-Turing thesis.

“The class of all functions computable by means of an algorithm is equivalent to the class of all functions computable by a Turing machine”.^a Very informally: Every computing device can be simulated on a *Turing machine* to any desired precision.

^aAs such, this might be regarded as the “definition of an algorithm”, more than a theorem of mathematics. In other words, the concept of a “function computable by means of an algorithm” is too vague, if I do not define what an algorithm is: hence, it is in some sense impossible to *prove* the Thesis.

A Turing Machine. Let me spend a few words on the concept of a Turing Machine (TM), a subject of classical computation, which would take an entire course on its own. If you want to learn more about classical computation, consult the beautiful book by M. Sipser [8], which I will here follow. A TM is an abstract general model of a classical computing machine, of which our current digital computers are the most known and relevant hardware implementation: both have equivalent computational power.

1

A Turing Machine. A TM, schematically illustrated in Fig. 1.1, is specified by:

Tape) A semi-infinite Tape, made of cells containing symbols taken from a given tape alphabet \mathcal{T} . A special symbol \sqcup (blank) is used to signal, for instance, the end of the meaningful cells of the tape.

Input alphabet) A finite set of input symbols \mathcal{A} , not containing \sqcup , which is a subset of the tape symbols, $\mathcal{A} \subset \mathcal{T}$.

States) A finite set of states $\mathcal{S} = \{s_0, s_1, \dots, s_n, s_{\text{accept}}, s_{\text{reject}}\}$, which includes a *starting state* s_0 , as well as two special states, s_{accept} and s_{reject} , where the TM accepts or rejects the input string and **halts**. Each state of the TM determines a certain behaviour of the machine, as described below.

Head) At step k of the computation, with the TM in state $s^k \in \mathcal{S}$, the Head reads a certain symbol t_j from the j th cell of the Tape. Following that, according to a well-defined set of transition rules (see below), it moves the TM to a (possibly new) state s^{k+1} , writes a new symbol \tilde{t}_j on cell j , and moves by one cell left (L) or right (R) along the tape.

Transition rules) The **code** governing the TM is a set of transition rules of the type $(s, t) \rightarrow (\tilde{s}, \tilde{t}, R/L)$, where $s \in \mathcal{S}$ is a state of the TM at step k (hence $s = s^k$), and t is the tape symbol at the cell where the Head is located, $\tilde{s} = s^{k+1}$ is the new state over which the TM switches (including $\tilde{s} = s$), \tilde{t} the new tape symbol written on the current cell, and L/R tells the Head to move to the left (L) or to the right (R).

To illustrate the code you need to write for the 1st simple decision problem above, recognising if $L = \{0, \dots, 0 \text{ } m \text{ times}\}$ has a power of two, $m = 2^n$, number of 0s, consider the following pseudo-code.

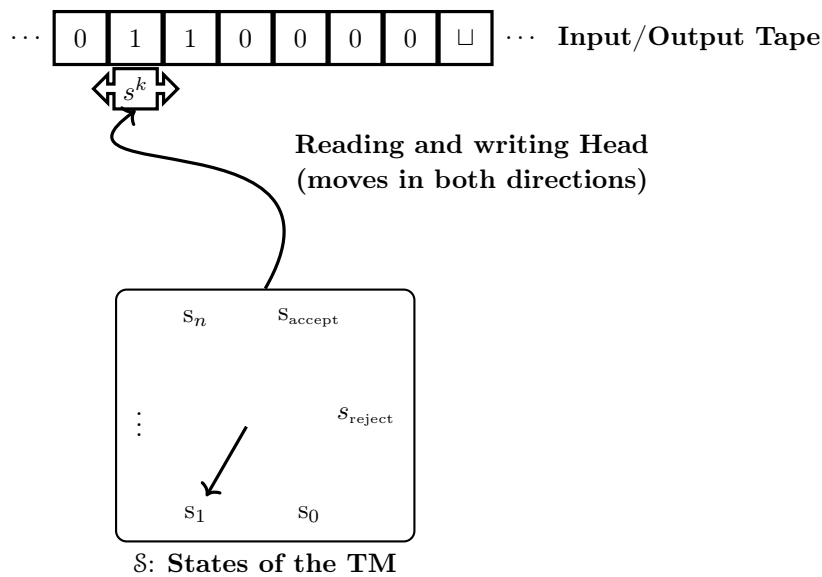


Figure 1.1.: Representation of Turing Machine. The tape alphabet is here $\mathcal{T} = \{0, 1, \sqcup\}$. The machine at time-step k is in state $s^k = s_1$ and is reading a cell with a 1. \sqcup signals the end of the tape.

i

Pseudo-code.

1. Sweep the tape left to right, crossing off (changing 0 with a x) every other 0.
2. If in 1. the Tape contained a single 0, accept
3. If in 1. the Tape contained an *odd* number of 0s, greater than one, reject.
4. Return the Head to the initial position of the tape.
5. Go to step 1. and repeat.

The rationale is that Step 1. is a process of division by 2 of the 0s, where half of the 0s are changed into a new tape symbol x, while half remain 0s. Step 4. allows moving the Head back to the leftmost position (which you should mark by substituting the 0 in the initial cell with a \sqcup), to repeat the process of division by 2 of the 0s. If the number of 0s at a certain stage is larger than 1 but *odd*, then the original number of 0s was not a multiple of 2: for instance, from $m = 6$, you would reduce to 3, and therefore eventually reject the string. Figure 1.2 below shows the input string and tape after a full sweep to the right, for $m = 6$.

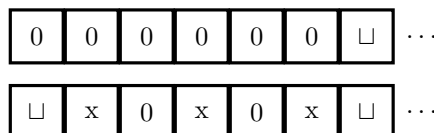


Figure 1.2.: Top: The starting input configuration for a string with $m = 6$ 0s. Below: The tape configuration after a first sweep of the string.

More in detail, we define a TM where $\mathcal{A} = \{0\}$ is the input alphabet, $\mathcal{T} = \{0, x, \sqcup\}$ the tape alphabet, including the symbol x to substitute every other 0. By thinking a bit ², one comes out with a set of transition rules between the $5 + 2$ states that you need to have for the machine to work. The full code (transition rules) can be represented by the graph in Fig. 1.3.

²This is the non-trivial part of the design of the actual “code” of the TM.

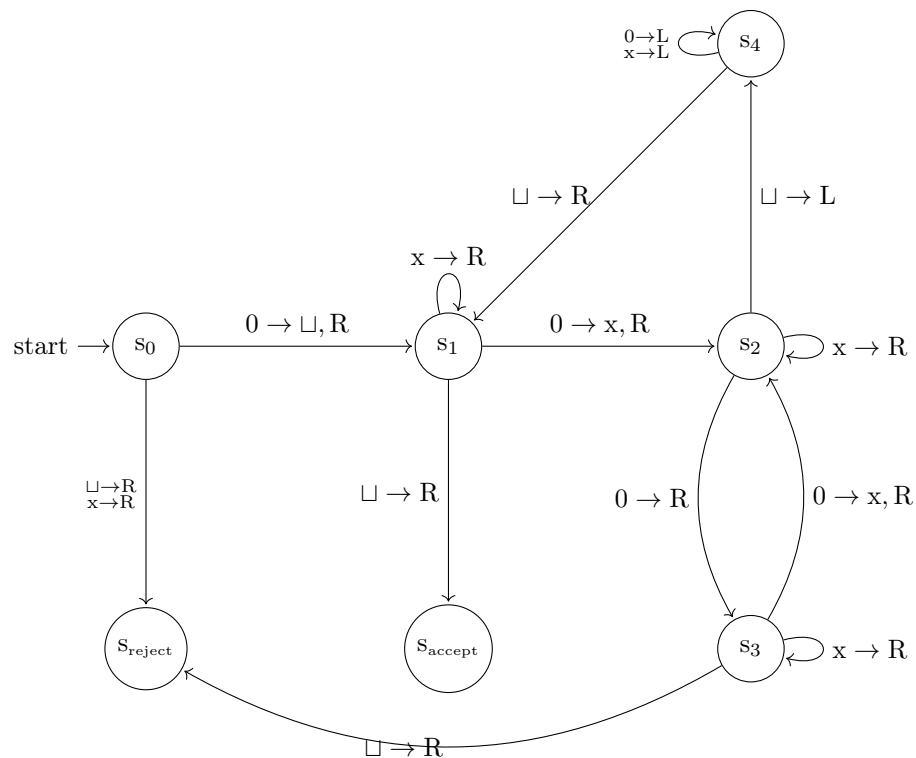


Figure 1.3.: A Turing Machine code, showing the transition map for deciding if the number of 0s is a multiple of 2. Verify that you get Fig. 1.2 after a full sweep of the tape from left to right.

Computational problems. Of course, a TM can also be used for computational problems. In that case, the two states s_{accept} and s_{reject} might be substituted by a single state s_{halt} , where the TM halts. The output of the calculation, however, should be written, in this case, on the Tape.

Let us illustrate the simplest example of a TM computation: summing two integers. For the sake of simplicity in the design of the TM, we will use here a *very inefficient* way of coding the integers on the Tape: a *unary representation*, where each integer is represented by as many 1s as the integer itself: 5, for instance, is written as 11111. To signal the beginning of an integer we use the symbol $\#$. The alphabet is here $\mathcal{A} = \{1, \#\}$, while the tape alphabet is $\mathcal{T} = \{1, \#, \sqcup\}$.

Suppose we want to sum $2 + 3$. We write on the tape the symbols $\#11\#111\sqcup$, see Fig. 1.4 (top). The TM starts in state s_0 and, upon reading the first $\#$ moves to state s_1 . In state s_1 , the machine reads a 1, does not modify it, and moves to the right, until a new $\#$ (signalling the second integer) is met. At this point, the machine switches to state s_2 , whose duty is to change the $1 \rightarrow \#$, move to the left and switch to a state s_3 which does precisely the opposite: changes $\# \rightarrow 1$ and moves to the right. The combined effect of this action is to have now the tape in configuration $\#111\#11\sqcup$, with a 1 that was moved to the left of the second $\#$. Proceeding in this way, the machine switches between s_2 and s_3 to keep moving all the 1s to the left of the second $\#$, until it stops, entering the state s_{halt} , when no 1s are left. The machine ends the calculation with the tape in configuration $\#11111\#1\sqcup$, see Fig. 1.4 (bottom), which contains the desired unary representation of $5 = 2 + 3$.

The full code (transition rules) for a TM performing the sum of integers in unary representation is represented by the graph in Fig. 1.5.

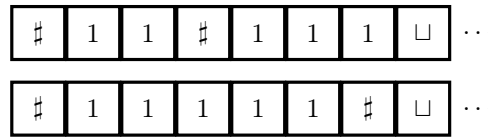


Figure 1.4.: Top: The starting input configuration of the Tape when summing $2 + 3$. Below: The tape configuration when the TM halts, where you read the integer 5.

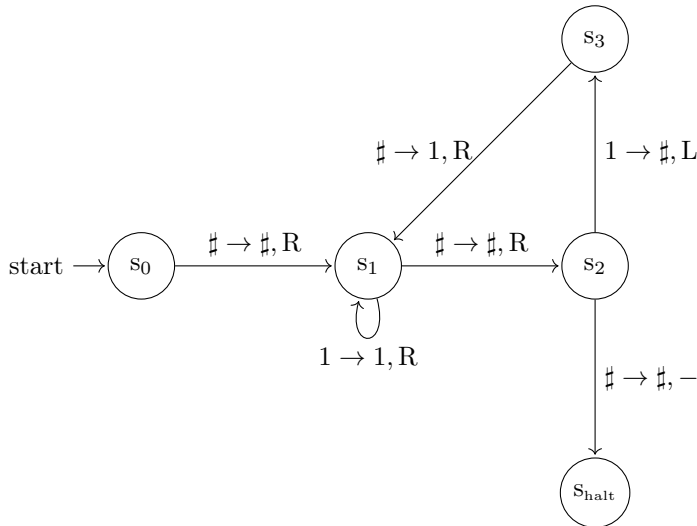


Figure 1.5: A Turing Machine code, showing the transition map for summing two integers in unary representation.

1.1.1. Computability and decidability

The statement of the Church-Turing thesis — with the allusion to “computable by an algorithm” — effectively implies that there are problems that are “non-computable”.

For the case of decision problems, one would call them **undecidable** problems. This happens because the possible outcome of running a TM on a given input string $\underline{t}^{\text{input}}$ of the tape can be **accept**, **reject**, or **loop and never halt**, with any simple or complex behaviour which visits the possible states $s_j \in \mathcal{S}$ but never reaches one of the two halting states s_{accept} or s_{reject} . We saw before simple decision problems which were decidable, but also a remarkable problem of mathematics — Hilbert’s 10th problem — which is **undecidable** for polynomials of more than one variable.



A no-go result. In essence, no TM can be devised such that, upon coding in the input tape the integer coefficients of an arbitrary polynomial of many variables, the TM decides if the polynomial has integer roots or not.

A very famous undecidable problem was introduced by Turing since the beginning of his theory, the **halting problem**: deciding if, given a TM operating on an input, the machine will eventually halt, or rather loop forever.

Some of these undecidable problems might surprise you, because they are inside the realm of the quantum theory of many-body systems, like the undecidability of the *spectral gap problem*: given the Hamiltonian of a quantum many-body system — say, a translationally invariant spin Hamiltonian on a square lattice — decide from the knowledge of the interaction coefficients if the model is gapped, with a unique ground state, or rather gapless (in the thermodynamic limit). ³

³See Cubitt *et al.*, Nature **528**, 207 (2015). The precise statement is the following. The *spectral gap problem* — i.e., deciding with an algorithm if a given quantum Hamiltonian has a gapless spectrum, or is gapped with a unique

The formulation of the Church-Turing thesis does not put limits to the *running time* of the algorithm, or to the *memory* used. As such, the distinction between computable and non-computable is too coarse. A computer scientist would ask questions about the *efficiency* with which one can calculate a “computable function”, meaning whether the length (or memory usage) of the computation would scale polynomially, or super-polynomially, with the length of the input.

There are computable but notoriously difficult problems in all disciplines, from computer science to physics. For instance, in physics, simulating a quantum many-body system. A **classification of computational efficiency** can only make sense if it is independent of the computing device you use. This has led to the following “extended” or “strong” or “quantitative” version of the Church-Turing thesis:

❶

Extended Church-Turing thesis. Any “physical” computing device can be simulated by a Turing machine in a number of steps polynomial in the “resources” used by the computing device.

The implication is that if a problem cannot be solved with polynomial “resources” on a Turing machine, then it has no efficient solution on any other “physical” machine. Notice the word “physical”, which means “*which can be built and made to work*”. Notice also the other clause alluding to the “resources”. For *digital devices* by “resources” one means **time** (or computational steps), and **space** (or memory used). For *analog devices*, there is an additional “resource”, the **precision**.

Proposals of analogue devices which violate the extended Church-Turing thesis, seemingly solving in polynomial time hard problems of computer science, have so far been based on exponentially precise parts or involved an exponentially large energy cost. See Ref. [9][Sec. 1] for a useful introduction to these points.

Question: Why quantum?

This leads to the crucial question: would a computing device based on the laws of Quantum Mechanics be a counter-example to the extended Church-Turing thesis?

One could argue positively on that, following Feynman. After all, Nature is quantum, and, for instance, a quantum device might be able to *simulate* efficiently quantum many-body problems.

This sub-field of the “Quantum technology” endeavour is known as *Quantum Simulators*. Physicists have now built physical quantum devices on which they have a remarkable degree of control. For instance, assemblies of **Rydberg atoms** — e.g., ^{87}Rb — held in desired positions \mathbf{R}_i by optical tweezers, see Refs. [11–13] — which are essentially described by a quantum Hamiltonian of the form:

$$\hat{H}_{\text{Rydberg}} = \frac{1}{4} \sum_{i < j} V_{ij} (1 - \hat{\sigma}_i^z) (1 - \hat{\sigma}_j^z) + \frac{\hbar\Omega}{2} \sum_i \hat{\sigma}_i^x - \frac{\hbar\Delta}{2} \sum_i (1 - \hat{\sigma}_i^z). \quad (1.3)$$

Here Ω is the so-called *Rabi frequency* (or coupling) between the atom in the ground state, $|g\rangle = |\uparrow\rangle$ with a spin-1/2 mapping, and the atom in the highly excited Rydberg state $|r\rangle = |\downarrow\rangle$, induced by a two-photon optical transition. $\hbar\Delta$ is the so-called *detuning* of the laser frequencies in the Rydberg two-photon transition, and $V_{ij} \sim C_6/|\mathbf{R}_i - \mathbf{R}_j|^6$ is the (van der Waals) interaction between Rydberg excited atoms.

ground state — is *algorithmically undecidable*, in the same sense in which the halting problem for a Turing Machine is undecidable. This means that there cannot be an algorithm that, given in input a description of the local interactions, determines whether the corresponding quantum Hamiltonian is gapless or gapped.

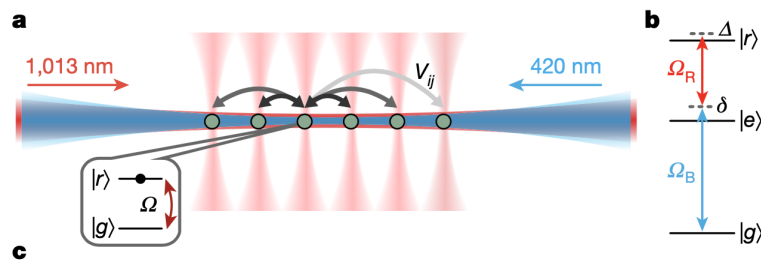


Figure 1.6.: Rydberg atoms setup. Figure extracted (including part of the caption) from Fig.1 of Ref. [10]. a) Individual ^{87}Rb atoms are trapped using optical tweezers (vertical red beams). Coherent interactions V_{ij} between the atoms are enabled by exciting them (horizontal blue and red beams) to a Rydberg state with strength Ω and detuning Δ . b) A two-photon process couples the ground state $|g\rangle = |5S_{1/2}, F = 2, m_F = -2\rangle$ to the Rydberg state $|r\rangle = |70S_{1/2}, J = 1/2, m_J = -1/2\rangle$ via an intermediate state $|e\rangle = |6P_{3/2}, F = 3, m_F = -3\rangle$ with detuning δ , using circularly polarized 420-nm and 1,013-nm lasers.

But there are issues about the *precision* which should inevitably be posed, as some of the ingredients of such a quantum device are *analogue*: amplitudes of quantum states, which one would like to manipulate and control, but protect from external noise, etc.

i **Quantum technologies.** Quantum Mechanics might help us not only in **simulating** quantum systems, and possibly in **computing** more efficiently problems that are classically hard, but also in other important applications like **building sensors** — *Quantum sensors* — or having **secure communications** — *Quantum cryptography*.

Let us stop here these general considerations, and briefly review a few basic facts of QM which are useful to have in mind.

1.2. Probability theory and Quantum Mechanics

We choose to examine a phenomenon that is impossible, absolutely impossible, to explain in any classical way, and which has in it the heart of quantum mechanics. In reality, it contains the only mystery. We cannot make the mystery go away by “explaining” how it works. We will just tell you how it works. In telling you how it works we will have told you about the basic peculiarities of all quantum mechanics.

Richard Feynman, Lectures on Physics, Vol. III

You shoot photons (or electrons, or any quantum particle), even *one-by-one*, towards a wall with two narrow slits, and observe interference fringes on the screen that sits behind the wall, after many individual events are collected.

Where each photon (or particle) lands on the screen is *probabilistic*. But this is *not* the strange side of the story. You might always think that there are unobserved (hidden) variables that, if known, would make the dynamics totally “deterministic”. Figure 1.7 alludes to such a mental picture, where you should imagine that the world we observe is only “one of the walls” of a billiard, where a chaotic but deterministic motion occurs.

What *is strange* is a non-monotone behaviour of the probability \mathbb{P} of observing a particle hitting a certain region of the screen. Classically, you would think that closing slit 2, and leaving only slit 1

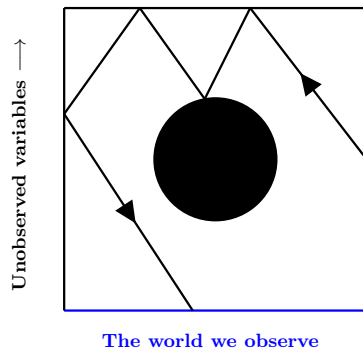


Figure 1.7: A Sinai billiard, made of a square or rectangle in which a circular hole is cut in the center. A classical particle bounces elastically from all the walls in the billiard. As proved by Sinai, the resulting classical motion is ergodic, indeed *mixing*. Hence, the location in which the particle hits one of the walls is chaotic.

open, the probability \mathbb{P}_1 of observing events in the same region would be *smaller*: $\mathbb{P}_1 < \mathbb{P}$. But this is not true: even places with $\mathbb{P} \approx 0$ can get a finite, appreciable, \mathbb{P}_1 if you close slit 1. Indeed, as you know, probabilities in QM are calculated starting from complex amplitudes:

$$\mathbb{P} = |\psi_1 + \psi_2|^2. \quad (1.4)$$

Interference is the key. You might have, say, $\psi_1 = \frac{1}{\sqrt{2}}$ and $\psi_2 = -\frac{1}{\sqrt{2}}$, resulting in $\mathbb{P} = 0$ but $\mathbb{P}_1 = |\psi_1|^2 = \frac{1}{2}$. We will see shortly an explicit demonstration of this weird fact in discussing the Mach-Zehnder interferometer.

More generally, in a classical world, you would think that if there are two possible ways in which something (an “event”) can happen, with probabilities \mathbb{P}_1 and \mathbb{P}_2 , then the total probability is:

$$\mathbb{P} = \mathbb{P}_1 + \mathbb{P}_2. \quad (1.5)$$

This is *false*, in the microscopic realm of QM. But if you think of *measuring* which slit the photon (or particle) went through — a so-called *which-way* experiment —, then you would *destroy interference*.



Decoherence. Our description of the world reverts to classical probabilities when systems are considered to be coupled to an environment, a highly non-trivial phenomenon known as *decoherence*. We will have more to say about it when discussing open quantum system dynamics.

Let us discuss these facts in more detail.

1.2.1. Interference in a Mach-Zehnder interferometer

To whet your appetite about weird QM effects, let us consider the following *Mach-Zehnder Interferometer* (MZI) setting shown in Fig. 1.8(a). It could view it as a simplified version of a double-slit experiment, where the continuum of possible paths and possible hits on the screen is substituted by only two paths and two detectors. No interference will lead to the two detectors having equal counts, interference to an imbalance in the counts.

A *single photon*⁴ of wavevector, \mathbf{k}_1 moving along the x-direction is sent onto the first 50-50 beam-splitter (BS_1), a very common device in all Quantum Optics labs, made of a carefully crafted half-reflecting surface. After BS_1 , there is a 50-50 probability that the photon is transmitted (T), keeping

⁴Since photon sources were invented by A. Aspect in 1985, they are nowadays routinely available. They should be distinguished from strongly attenuated photon sources. If a beam is strongly attenuated so that the average number of photons is very small, say $\langle n \rangle = \frac{1}{100}$, then 99% of the time there is no photon, in 1% of the cases there is one photon, but, with a Poisson’s distribution, there could also be 2, 3, etc. photons, and the coincidence counts would reveal the subtle difference with a true single-photon source. See A. Aspect’s public lecture upon receiving the N. Bohr Gold Medal 2013, available on [YouTube](#). Incidentally, the photon polarisation is assumed to be conserved, and hence neglected in the following discussion.

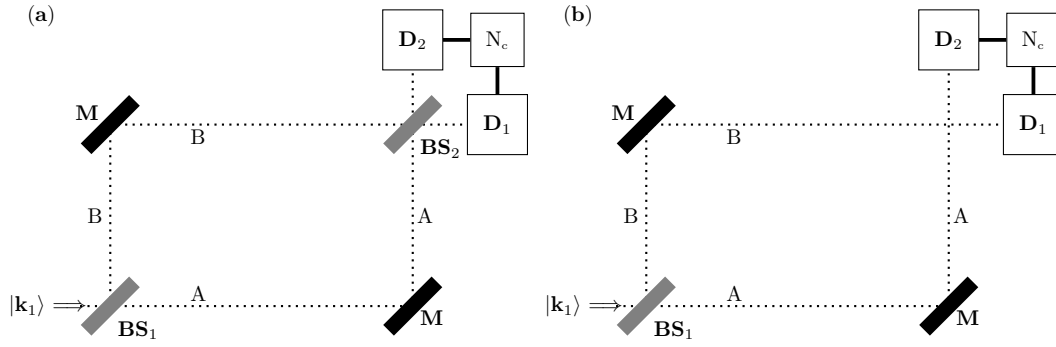


Figure 1.8.: (a) A Mach-Zehnder interferometer, with two beam-splitters (BS), two mirrors (M) and two final detectors (D). After BS₁, there is a 50-50 probability that the photon is transmitted (T), keeping its \mathbf{k}_1 and going into path A of the MZI, or reflected (R) into a state of wave-vector \mathbf{k}_2 along the y-direction, hence going into path B. If paths A and B have exactly equal lengths, all photons go into detector D₁. Coincidences are never registered. (b) Same as (a), but without the second beam-splitter BS₂. The apparatus now behaves as a *which-way* particle detector, with photons ending up in either one of the two detectors with equal probability, but *never in coincidence*, as one can verify with a coincidence counter.

its \mathbf{k}_1 and going into path A of the MZI, or reflected (R) into a state of wave-vector \mathbf{k}_2 along the y-direction, hence going into path B. We will denote such states as $|\mathbf{k}_1\rangle$ and $|\mathbf{k}_2\rangle$: they will form the basis of our simple 2×2 calculations. The phase accumulated by these states will not be explicitly included, because the length of the two arms A and B of the interferometer is assumed to be identical. Notice how we are effectively using our classical “image” of a “photon going through a path”, typical of “particle” way of looking at the photon.

Quantum Optics teaches us that such a 50-50 beam-splitter can be described by the following unitary matrix in the basis $\{|\mathbf{k}_1\rangle, |\mathbf{k}_2\rangle\}$ of x- and y-directed wave-vector states:

$$\mathbf{U}_{\text{BS}_1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & ie^{i\varphi_{\text{BS}}} \\ ie^{-i\varphi_{\text{BS}}} & 1 \end{pmatrix}. \quad (1.6)$$

As you see from element 21 of the matrix, the reflected photon gets an extra phase $ie^{-i\varphi_{\text{BS}}}$, where φ_{BS} depends on the reflecting coating. Unitarity, however, constraints the element 12 to be $ie^{i\varphi_{\text{BS}}}$. A common choice is to assume $\varphi_{\text{BS}} = 0$, as we will do. The two mirrors (assumed identical) simply reflect momenta, and can be described by the unitary matrix

$$\mathbf{U}_{\text{M}} = \begin{pmatrix} 0 & e^{i\varphi_{\text{M}}} \\ e^{i\varphi_{\text{M}}} & 0 \end{pmatrix}. \quad (1.7)$$

Here one could take $\varphi_{\text{M}} = \pi$ (the reflection is associated with a change of sign), but this phase will play no role in our discussion, and we will leave it generic.

i

No BS₂: which-way detector operation. In absence of the second beam splitter, our description of the MZI — see Fig. 1.8(b) — would not be an interferometer at all. It is leading to a photon *either* going along A, ending in D₂, *or* going along B, ending in D₁, like a “*particle*”. If you measure the *coincidence counts* of the two detectors you find 0, compatibly with noise.

The quantum calculation easily confirms this. The initial state being $|\psi_{\text{in}}\rangle = |\mathbf{k}_1\rangle \rightarrow (1, 0)^T$, we get a final state, in absence of BS₂:

$$|\psi_{\text{fin-no BS}_2}\rangle = \mathbf{U}_{\text{M}}\mathbf{U}_{\text{BS}_1}|\psi_{\text{in}}\rangle = \frac{1}{\sqrt{2}}e^{i\varphi_{\text{M}}} \begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}e^{i\varphi_{\text{M}}} \begin{pmatrix} i \\ 1 \end{pmatrix}. \quad (1.8)$$

Calculating the probability that the photon ends up in either one of the two detectors is now easy. Recalling that $|\mathbf{k}_2\rangle \rightarrow (0, 1)^T$, we get:

$$\mathbb{P}_{D_1} = |\langle \mathbf{k}_1 | \psi_{\text{fin-no BS}_2} \rangle|^2 = \frac{1}{2} \quad \text{and} \quad \mathbb{P}_{D_2} = |\langle \mathbf{k}_2 | \psi_{\text{fin-no BS}_2} \rangle|^2 = \frac{1}{2} .$$

Essentially, the photon is detected in D_1 and D_2 with a 50-50 probability. But never, experimentally, in D_1 and in D_2 .

Now we add the second beam-splitter, operating the MZI like a real interferometer, see again Fig. 1.8(a). The final state predicted by QM is:

$$|\psi_{\text{fin}}\rangle = \mathbf{U}_{\text{BS}_2} \mathbf{U}_M \mathbf{U}_{\text{BS}_1} |\psi_{\text{in}}\rangle = \frac{1}{2} e^{i\varphi_M} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = e^{i\varphi_M} \begin{pmatrix} i \\ 0 \end{pmatrix} . \quad (1.9)$$

The probability of observing the photon in either of the two detectors is strongly modified:

$$\mathbb{P}_{D_1} = |\langle \mathbf{k}_1 | \psi_{\text{fin}} \rangle|^2 = 1 \quad \text{and} \quad \mathbb{P}_{D_2} = |\langle \mathbf{k}_2 | \psi_{\text{fin}} \rangle|^2 = 0 . \quad (1.10)$$

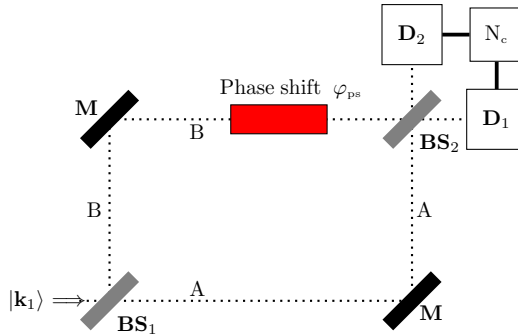


Figure 1.9: A Mach-Zehnder interferometer, with a phase-shifter inserted along arm B and acting on $|\mathbf{k}_1\rangle$ photons. Physically, it is enough that the photon passes through a piece of optical fibre of length L , which has an index of refraction $n > 1$, leading to a controllable phase delay with respect to the other path, where no fibre is present. Coincidences are never registered.

i **Interference in MZI.** With a perfect MZI, all photons entering the MZI along the x-arm, end at detector D_1 . If a phase-shifter, provoking a phase-delay φ_{ps} , is inserted along arm B, see Fig. 1.9, then you can easily show — as you learn by doing Exercise 1.1 — that

$$\mathbb{P}_{D_1} = |\langle \mathbf{k}_1 | \psi_{\text{fin}} \rangle|^2 = \cos^2 \frac{\varphi_{\text{ps}}}{2} \quad \text{and} \quad \mathbb{P}_{D_2} = |\langle \mathbf{k}_2 | \psi_{\text{fin}} \rangle|^2 = \sin^2 \frac{\varphi_{\text{ps}}}{2} . \quad (1.11)$$

Exercise 1.1. Assuming that the phase-shifter is modelled with a unitary matrix:

$$\mathbf{U}_{\text{ps}} = \begin{pmatrix} e^{i\varphi_{\text{ps}}} & 0 \\ 0 & 1 \end{pmatrix} ,$$

affecting only $|\mathbf{k}_1\rangle$ photons, show that:

$$|\psi_{\text{fin}}\rangle = \mathbf{U}_{\text{BS}_2} \mathbf{U}_{\text{ps}} \mathbf{U}_M \mathbf{U}_{\text{BS}_1} |\psi_{\text{in}}\rangle ,$$

is such that the probabilities in Eq. (1.11) are obtained. Check also what would be the role of a possible phase φ_{BS} appearing in the beam-splitter unitary matrix.

Let us now see how we would describe the same process in terms of classical probabilities. ⁵ The phase shifter — just a piece of optical fibre inserted in arm B — should make absolutely no difference

⁵NB: The word “classical” should create no confusion. We do not mean “in terms of *classical electromagnetic fields*”, which are waves, and do show interference!

at the probability level: if a photon has gone through arm B, it will enter and exit the optical fibre: that's it. A 50-50 beam splitter would be described by a transition matrix of the form:

$$\mathbf{T}_{\text{BS}} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad (1.12)$$

a *stochastic matrix* (SM) typical of Markov chains and classical master equations. The mirrors would be simply described by a trivial SM of the form:

$$\mathbf{M} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Starting from a probability vector $\mathbf{P}_{\text{in}} = (1, 0)^T$, in absence of the second beam-splitter I would expect:

$$\mathbf{P}_{\text{fin-no BS}_2} = \mathbf{M}\mathbf{T}_{\text{BS}_1}\mathbf{P}_{\text{in}} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix},$$

the same conclusion that QM reaches. The big difference with the QM calculation comes when one includes the second beam-splitter. Classically:

$$\mathbf{P}_{\text{fin}} = \mathbf{T}_{\text{BS}_2}\mathbf{M}\mathbf{T}_{\text{BS}_1}\mathbf{P}_{\text{in}} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}. \quad (1.13)$$

As perhaps expected, this classical conclusion is identical to that obtained from a single BS and radically different from that of QM.

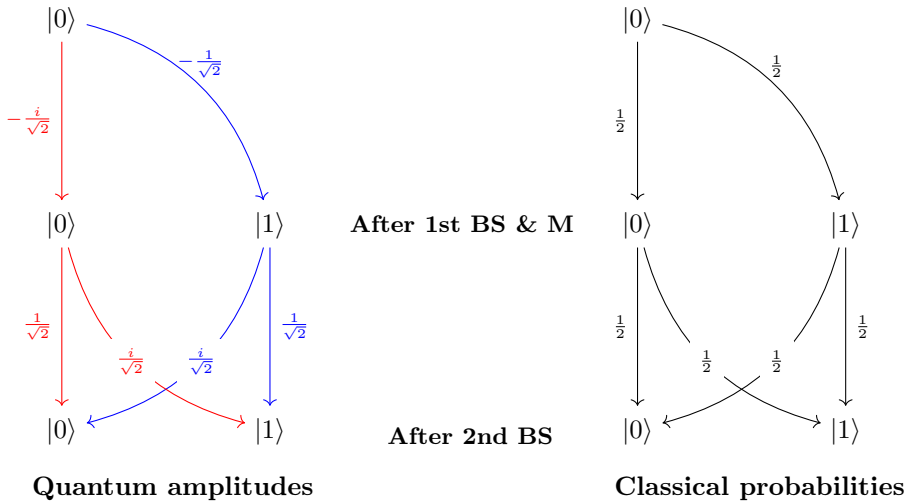


Figure 1.10.: Left: The amplitudes in the two paths of the interferometer. Here $|0\rangle$ is a shorthand for $|\mathbf{k}_1\rangle$, a photon travelling along x, and $|1\rangle$ stands for $|\mathbf{k}_2\rangle$, a photon travelling along y. The phase $e^{i\pi} = -1$ due to the mirror is included. Right: The corresponding classical probabilities.

Figure 1.10 shows the two paths connecting the initial state $|\mathbf{k}_1\rangle = |0\rangle$ to each of the two detector final states: D_1 , associated to $|\mathbf{k}_1\rangle = |0\rangle$, and D_2 , associated to $|\mathbf{k}_2\rangle = |1\rangle$. Here $|0\rangle$ and $|1\rangle$ are shorthands for the two basis states used in our calculations, where all identical accumulated phases are consistently neglected. The left part shows the QM calculation, with the individual amplitudes making up $\mathcal{A}_{|0\rangle \rightarrow |0\rangle}$ and $\mathcal{A}_{|0\rangle \rightarrow |1\rangle}$. Here blue denotes arm A and red is associated with arm B. One gets:

$$\mathcal{A}_{|0\rangle \rightarrow |0\rangle} = \left(\frac{-i}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}}\right) + \left(\frac{-1}{\sqrt{2}}\right)\left(\frac{i}{\sqrt{2}}\right) = -i \quad \text{and} \quad \mathcal{A}_{|0\rangle \rightarrow |1\rangle} = \left(-\frac{i}{\sqrt{2}}\right)\left(\frac{i}{\sqrt{2}}\right) + \left(-\frac{1}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}}\right) = 0,$$

from which Eq. (1.10) for $\mathbb{P}_{D_1} = |\mathcal{A}_{|0\rangle \rightarrow |0\rangle}|^2$ and $\mathbb{P}_{D_2} = |\mathcal{A}_{|0\rangle \rightarrow |1\rangle}|^2$ follows. The right part of Fig. 1.10 shows the corresponding classical probability calculation, where along different paths probabilities are summed: both paths contribute equally to the final probability.



Blocking one path. It is worth to remark that while

$$\mathbf{P}_{D_2} = \mathbf{P}_{D_2|\text{path A}} + \mathbf{P}_{D_2|\text{path B}} = \frac{1}{4} + \frac{1}{4} = \frac{1}{2},$$

hence if you block path B the probability of getting a photon in D_2 is reduced to $\mathbf{P}_{D_2|\text{path A}} = \frac{1}{4}$, the same blocking of path B provokes a remarkable *increase* of the quantum probability \mathbb{P}_{D_2} , from 0 to $\frac{1}{4}$.

1.2.2. Wheeler's delayed-choice experiment

As you saw from the previous calculation, an almost identical apparatus, except for the presence or absence of the second beam-splitter, acts in two complementary ways: either as a which-way particle detector (BS_2 absent) or as an interferometer (BS_2 present). The word “complementary” is used on purpose. This is, essentially, *Bohr's complementarity principle*: A quantum object, depending on the experimental apparatus, *either* shows particle-like aspects, *or* wave-like aspects, but not both. Since you cannot do two radically different experiments at the same time, there is no source of inconsistency.

But here comes a bright idea by J.A. Wheeler: *What if the experimentalist decides to insert or not insert the second beam-splitter when the photon has already entered the apparatus, passing the first beam-splitter?* The experimentalist, in other words, has *delayed the choice* of what type of experiment — which-way particle detector *or* interferometry — to perform. How? By a very fast electronic device known as EOM (Electro-Optical modulator), which essentially can act as a beam-splitter or not depending on a certain voltage V which the system *randomly chooses* when the photon is already in the middle of the 48 meters long ⁶ arms of the MZI. Fig. 1.11 shows the apparatus used in 2007 by the Aspect's group to perform such a Wheeler's delayed-choice experiment.

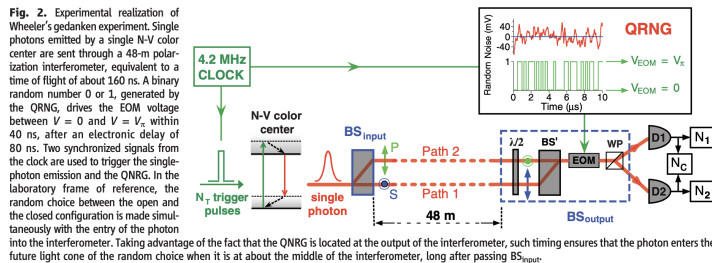


Fig. 2. Experimental realization of Wheeler's gedanken experiment. Single photons emitted by a single N-V color center are sent through a 48-m polarization interferometer, equivalent to a time of flight of about 160 ns. A binary random number 0 or 1, generated by the QRNG, drives the EOM voltage between $V = 0$ and $V = V_s$ within 40 ns, after an electronic delay of 80 ns. Two synchronized signals from the clock are used to trigger the single-photon emission and the QRNG. In the laboratory frame of reference, the random choice between the open and the closed configuration is made simultaneously with the entry of the photon into the interferometer. Taking advantage of the fact that the QRNG is located at the output of the interferometer, such timing ensures that the photon enters the future light cone of the random choice when it is at about the middle of the interferometer, long after passing BS_{input} .

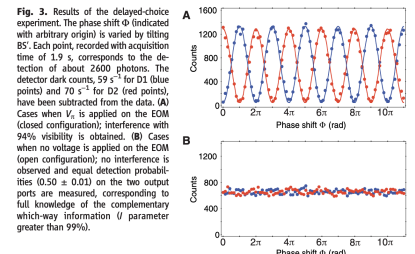


Fig. 3. Results of the delayed-choice experiment. The phase shift Φ (indicated with arbitrary origin) is varied by tilting BS_2 . Each point, recorded with acquisition time of 1.9 s, corresponds to the detection of about 2600 photons. The detector dark counts, 59 ± 2 for D_1 (blue points) and 70 ± 1 for D_2 (red points), have been subtracted from the data. (A) Cases when V_s is applied on the EOM (closed configuration): interference with 94% visibility is obtained. (B) Cases when no voltage is applied on the EOM (open configuration): no interference is observed and equal detection probabilities (0.50 ± 0.01) on the two output ports are measured, corresponding to full knowledge of the complementary which-way information (V parameter greater than 99%).

Figure 1.11.: Left: Figure 2 from Ref. [14], with a sketch of the experimental apparatus. Right: Figure 3 from the same paper, showing the interference for the counts made “in coincidence” with the presence of the second beam-splitter, and no interference otherwise.

Needless to say, QM works perfectly. Our classical mental pictures are simply unable to catch up with the QM rules-of-the-game. It is not that the photon passes BS_1 , “sees” that you want to do a “particle experiment” and therefore acts like a particle, or vice-versa, seeing that you want to perform an interferometric experiment, it behaves as a wave. You do not play tricks to nature by delaying your choices on “what operators you apply to the state”. The photon simply follows the strange rules of QM, that govern the object — the quantum state $|\psi\rangle$ — containing all the statistical information on the experiment you perform: if you insert the BS_2 along the way, in the middle of the arm of the MZI, the state $|\psi\rangle$ will be acted-upon by \mathbf{U}_{BS_2} and the game is over: calculate probabilities, and you get the correct answers. If you do not insert the BS_2 , the state is not acted-upon by \mathbf{U}_{BS_2} , and the resulting probabilities are again correct. No dirty tricks are allowed.

⁶It takes $t = 160\text{ns}$ for a photon to travel 48 meters: fast enough electronics can easily cope with that.

1.2.3. Which-way experiments and the delayed-choice quantum eraser

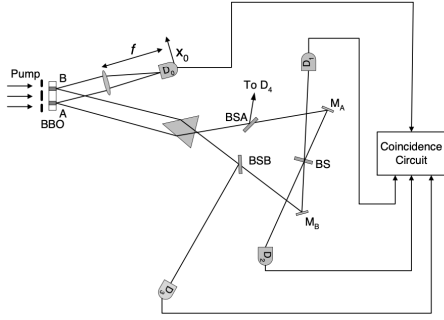


FIG. 2. Schematic of the actual experimental setup. The pump laser beam is divided by a double slit and forms two regions A and B inside the BBO crystal. A pair of signal-idler photons is then generated from either the A or the B region. The “delayed choice” to observe either wave or particle behavior of the signal photon is made randomly by the idler photon about 7.7 ns after the detection of the signal photon.

Figure 1.12: Sketch of the experimental apparatus used in Ref. [15], in the spirit of a Young’s double-slit apparatus, with single photons generating, by a non-linear Quantum Optics trick known as *Spontaneous Parametric Down Conversion* a pair of polarisation-entangled photons, one sent to a screen (the *signal* photon), and one (the *idler* photon) sent to a complex apparatus used as a “path-detector”. A very ingenious way of using entanglement to try and trick Nature, getting interference on a screen while trying to also get (or erase) which-way information.

Let us now see what happens if we try to measure the path that the photon has gone, with a “*which-way*” measuring device. This time I will discuss a *Gedanken*⁷ experiment, modelling the problem as suggested in a recent paper by Qureshi, Ref. [16]. The actual experiment was performed back in 2000 by the group of M. Scully, see Ref. [15], and is sketched in Fig. 1.12. I will not describe the experiment, which is a bit involved because of the many mirrors/beam-splitters/detectors.

So, let us model the problem as sketched in Fig. 1.13.

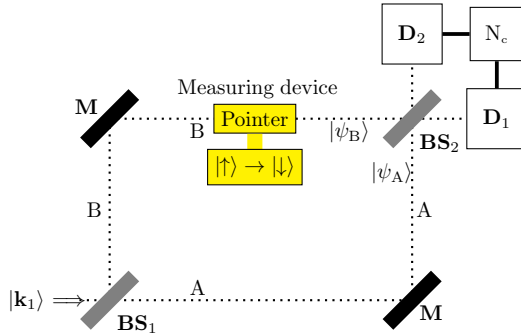


Figure 1.13: A Mach-Zehnder interferometer, with a spin-1/2 measuring device (pointer). The pointer, normally in state $|\uparrow\rangle_p$, flips $|\uparrow\rangle_p \rightarrow |\downarrow\rangle_p$, if a photon goes through path B. Here $|\psi_A\rangle = (0, -1)^T$ and $|\psi_B\rangle = (-i, 0)^T$ are the (normalised) states before BS_2 “in the two paths”.

We place along path B a *measuring device* consisting of a single spin-1/2. Such a device is sometimes called a *pointer*. The pointer is normally in the state $|\uparrow\rangle_p$, but flips into state $|\downarrow\rangle_p$ if the photon goes through path B. The Hilbert space of the problem includes now the pointer space, with the usual tensor product. The initial state of the photon+pointer is now $|\Psi_{\text{in}}\rangle = |\mathbf{k}_1\rangle \otimes |\uparrow\rangle_p$.

Just *before* the second BS, in absence of pointer, the photon state would be, as predicted by Eq. (1.8) which we summarise here, taking $\varphi_M = \pi$:

$$|\psi_{\text{fin-no } BS_2}\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} -i \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|\psi_A\rangle + |\psi_B\rangle), \quad (1.14)$$

where $|\psi_A\rangle = (0, -1)^T$ and $|\psi_B\rangle = (-i, 0)^T$ are (normalised) states “in the two arms” (neglecting as usual the the equal-phase accumulated along the paths). In presence of the pointer, the state of the system before the second BS, but *after the photon has possibly interacted with the pointer*, would be:

$$|\Psi_{\text{before } BS_2}\rangle = \frac{1}{\sqrt{2}} (|\psi_A\rangle \otimes |\uparrow\rangle_p + |\psi_B\rangle \otimes |\downarrow\rangle_p).$$

⁷But see a very recent proposal by Qureshi, arXiv:2010.00049, with a realisable experiment based on a simple enough MZI device, using polarisation-entangled photons.

As you see, the state is *no longer separable*: interaction with the pointer has produced *entanglement* between the photon and the pointer. Now we let the second beam-splitter act, getting the final state:

$$\begin{aligned}
|\Psi_{\text{fin}}\rangle &= \mathbf{U}_{\text{BS}_2} \otimes \mathbf{1}_p |\Psi_{\text{before BS}_2}\rangle = \frac{1}{\sqrt{2}} \left(\mathbf{U}_{\text{BS}_2} |\psi_A\rangle \otimes |\uparrow\rangle_p + \mathbf{U}_{\text{BS}_2} |\psi_B\rangle \otimes |\downarrow\rangle_p \right) \\
&= \frac{-i}{\sqrt{2}} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} \otimes |\uparrow\rangle_p + \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} \otimes |\downarrow\rangle_p \right) \\
&= \frac{-i}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \frac{|\uparrow\rangle_p + |\downarrow\rangle_p}{\sqrt{2}} - i \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \frac{|\uparrow\rangle_p - |\downarrow\rangle_p}{\sqrt{2}} \right). \quad (1.15)
\end{aligned}$$

The two totally equivalent ways of writing the final state involve different combinations of states. Recall that we might call $|D_1\rangle = |\mathbf{k}_1\rangle = (1, 0)^T$ and $|D_2\rangle = |\mathbf{k}_2\rangle = (0, 1)^T$ the final states in which the photon enters one or the other detector. In terms of these, one might define:

$$|\psi_{\pm}\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm i \end{pmatrix} = \frac{1}{\sqrt{2}} (|D_1\rangle \pm i|D_2\rangle),$$

to be photon states which are in *superposition* in the two detectors. Obviously, concerning the pointer states, we have that

$$|\pm\rangle_p = \frac{1}{\sqrt{2}} (|\uparrow\rangle_p \pm |\downarrow\rangle_p)$$

are pointer spin states in the +x and -x direction. Armed with these shorthands, we can rewrite the final state as:

$$|\Psi_{\text{fin}}\rangle = \frac{-i}{\sqrt{2}} (|\psi_-\rangle \otimes |\uparrow\rangle_p + |\psi_+\rangle \otimes |\downarrow\rangle_p) = \frac{-i}{\sqrt{2}} (|D_1\rangle \otimes |+\rangle_p + |D_2\rangle \otimes |-\rangle_p). \quad (1.16)$$

These two alternative expressions are both very useful. The first tells us that *if I measure the pointer along the z-axis* and find it, say, in state $|\uparrow\rangle_p$, then the overall state collapses to

$$|\Psi_{\text{fin}}\rangle \xrightarrow{\text{measured } \uparrow_p} |\psi_-\rangle \otimes |\uparrow\rangle_p,$$

which implies that at the detectors I would measure:

$$\mathbb{P}_{D_1|\uparrow_p} = \mathbb{P}_{D_2|\uparrow_p} = \frac{1}{2},$$

and the same for measurement of $|\downarrow\rangle_p$.

❶

Measuring the pointer. So, getting information on the pointer state, to discriminate the path taken by the photon, completely *washes-out interference*.

But, what about if we *first detect* the photon, say in D_1 , without measuring the pointer? Then, the second expression in Eq. (1.16) would tell us that the state now collapses to:

$$|\Psi_{\text{fin}}\rangle \xrightarrow{\text{measured } D_1} |D_1\rangle \otimes |+\rangle_p.$$

A *subsequent* measurement of the pointer, to try and infer the path taken by the photon revealed in D_1 , would, however, give us:

$$\mathbb{P}_{\uparrow_p|D_1} = \mathbb{P}_{\downarrow_p|D_1} = \frac{1}{2},$$

and the same for the detection in D_2 .

i **Detecting the photon.** So, upon detecting the photon, I have *lost any information on the path* taken by that photon.

One further noteworthy point. Suppose we completely *disregard the pointer*, measuring the detector hits with the usual von Neumann projectors. For instance, we would get:

$$\mathbb{P}_{D_1} = \langle \Psi_{\text{fin}} | (|D_1\rangle\langle D_1| \otimes \mathbf{1}_p) | \Psi_{\text{fin}} \rangle = \frac{1}{2},$$

and similarly for \mathbb{P}_{D_2} . Hence, the interaction with the pointer, even if you completely disregard the pointer and *do not measure it*, completely washes-out interference. And notice that, as soon as the photon hits the detector, as seen above, *which-way information* is lost. Nevertheless, *interference is also lost*. Quite noteworthy.

i **The interaction with the pointer.** The *entanglement* with the pointer — a which-way detector — suffices to *destroy interference*.

Two very final remarks, a consequence of the second form of Eq. (1.16), which possibly demystify some common lore about *delayed-choice quantum eraser* experiments.

Quantum Eraser) Suppose I measure the pointer in the x-basis in spin space, which means that I *erase* the which-way information. Then, if I find the pointer in $|+\rangle_p$, the state of the photon collapses to $|D_1\rangle$. Hence, all events in which $|+\rangle_p$ is measured *do show interference*, as those lead to photons going to D_1 , as in absence of any pointer. Similarly, if I measure the pointer in $|-\rangle_p$, still erasing the which-way information, all particles go to detector D_2 , showing again interference, but of *opposite sign*. Taken together, however, interference is washed-out, consistently with the fact that no interference is seen when the pointer is ignored.

i **Coincidence measurements.** So, interference would only be seen in *coincidence measurements* with the pointer measured along the x-direction, to *erase which-way information*.

Delayed-choice Quantum Eraser) We now play a *delayed-choice* trick: we *first detect* the photon, say we find it in D_1 , and only *afterwards* we decide what to measure for the pointer.⁸ This is what QM predicts: Which-way information is erased, as we said previously, by the very fact of detecting the photon. Some information remains on “how which-way information is erased” in the pointer apparatus: the pointer is for sure in state $|+\rangle_p$, if the photon was detected in D_1 . You could verify this experimentally by measuring the pointer along x, *after* registering the photon in “coincidence”: indeed, if photons come at a sufficiently low rate, even if the pointer is measured “after detecting of the photon”⁹ you can uniquely correlate photon detection and pointer measurement. If you decide to measure the pointer in the z-direction, you find it $|\uparrow\rangle_p$ and $|\downarrow\rangle_p$ with a 50-50 probability, as previously discussed.

⁸Again, the scandal is only in our mental picture: there is no “retro-causal” behaviour, as sometimes alluded at, with profound philosophical implications about the Universe, and all such non-sense, as often discussed in the hundredths of videos on Youtube on this subject.

⁹Obviously in a way that is compatible with special relativity, i.e., in *all reference frames* you would say that the pointer is measured in the *future cone* of the photon detection.

1.3. Concluding remarks

These considerations bring us to a few final remarks. In a classical world, if an event is random, there is nothing you can do to make randomness disappear. On the contrary, the probabilistic aspect of *measurement* in QM brings in some very peculiar randomness in the outcomes of the measurement which, however, *depends on the choice of the operator you measure* and the associated basis of states. It can well happen that by an appropriate change of basis — hence by applying a suitable unitary operator to the states — a superposition state is transformed into a state with a *definite answer*. The Mach-Zehnder example just discussed shows this fact in a rather clear way.

This feature is at the root of the speed-up that QM allows, for instance, in Shor's Quantum Fourier Transform algorithm, leading, among other things, to a very efficient *period finding* algorithm, which in turn would allow breaking the current RSA public-key crypto-system.

1.4. Hands-on: EPR-type calculations with entangled particles

And now is your turn to try and revisit your QM with a few simple but very instructive calculations.

We consider the setup of the Einstein-Podolsky-Rosen (EPR) *Gedanken* experiment, in the formulation given by D. Bohm. A *spin zero* particle decays by emitting two particles with opposite momenta and opposite spin, which fly away from the emitting source towards two very far away from experimental stations A and B . In each station, there is a Stern-Gerlach (SG) apparatus which can be rotated in a direction, \mathbf{n}_A and \mathbf{n}_B , respectively, which can be modified at will. The two particles, labelled by A and B do not interact after the decay while flying towards the corresponding experimental station. Their total spin being $S = 0$, the state predicted by quantum mechanics would be the entangled pure state:

$$|\psi_{\text{ent}}\rangle_{AB} = \frac{1}{\sqrt{2}} \left(|+\rangle_A \otimes |-\rangle_B - |-\rangle_A \otimes |+\rangle_B \right). \quad (1.17)$$

where $|+\rangle = |\uparrow\rangle$ and $|-\rangle = |\downarrow\rangle$.

Exercise 1.2. (Singlet states.)

Show that, consistently with *rotational invariance*, the singlet state can be equivalently written with spin-states pointing into an arbitrary direction \mathbf{n} :

$$|\psi_{\text{ent}}\rangle_{AB} = \frac{1}{\sqrt{2}} \left(|\uparrow\rangle_A \otimes |\downarrow\rangle_B - |\downarrow\rangle_A \otimes |\uparrow\rangle_B \right) = \frac{1}{\sqrt{2}} \left(|+\rangle_A \otimes |-\rangle_B - |-\rangle_A \otimes |+\rangle_B \right). \quad (1.18)$$

Suppose that now A measures the spin along direction \mathbf{n}_A — hence, technically, the operator $\hat{A} = \mathbf{n}_A \cdot \hat{\boldsymbol{\sigma}}_A$ — getting the eigenvalue $a = +1$, with probability $\frac{1}{2}$, before B does his measurement.¹⁰ According to QM, the state collapses to:

$$|\psi_{\text{ent}}\rangle_{AB} \xrightarrow{\text{collapse}} |+\rangle_A \otimes |-\rangle_B, \quad (1.19)$$

where you should observe that B gets a spin state in a direction determined by the measurement in A . It turns out that it is useful to describe such a projective measurement and collapse in terms of density matrices and projectors.

¹⁰Obviously, “before” has a unique meaning only if the measurement that B performs is in the future cone of the measurement of A .

i

Von Neumann projective measurement. According to von Neumann, if

$$\hat{\Pi}_a^{\hat{A}} = \frac{1 + a\mathbf{n}_A \cdot \hat{\boldsymbol{\sigma}}_A}{2} \otimes \mathbf{1}_B \quad (1.20)$$

is the projector associated to \hat{A} — measuring the spin state with eigenvalue $a = \pm 1$ along direction \mathbf{n}_A — at station A , and $\hat{\rho}$ is an *arbitrary initial state*, then the probability of measuring a for the measurement at A , which we denote by $\text{Prob}(\mathcal{A}_a|\hat{\rho})$, and the final collapsed state $\hat{\rho}_a$ are:

$$\text{Prob}(\mathcal{A}_a|\hat{\rho}) = \text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\rho}) \quad \text{and} \quad \hat{\rho} \xrightarrow{\text{collapse}} \hat{\rho}_a \equiv \frac{\hat{\Pi}_a^{\hat{A}}\hat{\rho}\hat{\Pi}_a^{\hat{A}}}{\text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\rho}\hat{\Pi}_a^{\hat{A}})} = \frac{\hat{\Pi}_a^{\hat{A}}\hat{\rho}\hat{\Pi}_a^{\hat{A}}}{\text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\rho})}, \quad (1.21)$$

where we used, in the denominator, the cyclic property of the trace, and the fact that $(\hat{\Pi}_a^{\hat{A}})^2 = \hat{\Pi}_a^{\hat{A}}$.

Exercise 1.3. Verify that this indeed leads to $\text{Prob}(\mathcal{A}_a|\hat{\rho}) = \frac{1}{2}$ and to the collapsed state in Eq. (1.19).

Now, *after* A has measured a and the state has collapsed to $\hat{\rho}_a$, B measures the spin along a direction \mathbf{n}_B that he chooses, hence the operator $\hat{B} = \mathbf{n}_B \cdot \hat{\boldsymbol{\sigma}}_B$. As usual, get armed with the projector:

$$\hat{\Pi}_b^{\hat{B}} = \mathbf{1}_A \otimes \frac{1 + b\mathbf{n}_B \cdot \hat{\boldsymbol{\sigma}}_B}{2} \quad (1.22)$$

Exercise 1.4. (B measures after A.)

The probability that B measures the eigenvalue b for the spin along direction \mathbf{n}_B on the collapsed state $\hat{\rho}_a$ is given, following the von Neumann prescription, by:

$$\text{Prob}(\mathcal{B}_b|\hat{\rho}_a) = \text{Tr}(\hat{\Pi}_b^{\hat{B}}\hat{\rho}_a) = \frac{\text{Tr}(\hat{\Pi}_b^{\hat{B}}\hat{\Pi}_a^{\hat{A}}\hat{\rho}\hat{\Pi}_a^{\hat{A}})}{\text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\rho})} = \frac{\text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\Pi}_b^{\hat{B}}\hat{\Pi}_a^{\hat{A}}\hat{\rho})}{\text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\rho})}. \quad (1.23)$$

Now use Bayes' theorem for conditional probabilities

$$\text{Prob}(B \wedge A|X) = \text{Prob}(B|A \wedge X) \text{Prob}(A|X), \quad (1.24)$$

interpreting X as the state $\hat{\rho}$ originally prepared, and $A \wedge X$ as the state $\hat{\rho}_a$ after collapse, upon measuring \hat{A} on $\hat{\rho}$. Then show that:

$$\text{Prob}(\mathcal{B}_b \overset{\leftarrow}{\wedge} \mathcal{A}_a|\hat{\rho}) = \text{Prob}(\mathcal{B}_b|\hat{\rho}_a) \text{Prob}(\mathcal{A}_a|\hat{\rho}) = \text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\Pi}_b^{\hat{B}}\hat{\Pi}_a^{\hat{A}}\hat{\rho}), \quad (1.25)$$

where, in principle, $\mathcal{B}_b \overset{\leftarrow}{\wedge} \mathcal{A}_a$ reminds us that the measurement of B occurs *after* that of A .

Exercise 1.5. (The order of the measurements doesn't matter.)

Finally show that, since the two projectors acts on *different spaces*, hence commute, the order in which the two measurements are performed is actually irrelevant, and:

$$\text{Prob}(\mathcal{B}_b \overset{\leftarrow}{\wedge} \mathcal{A}_a|\hat{\rho}) = \text{Prob}(\mathcal{A}_a \overset{\leftarrow}{\wedge} \mathcal{B}_b|\hat{\rho}) \stackrel{\text{def}}{=} \text{Prob}(\mathcal{A}_a \wedge \mathcal{B}_b|\hat{\rho}) = \text{Tr}(\hat{\Pi}_a^{\hat{A}}\hat{\Pi}_b^{\hat{B}}\hat{\rho}). \quad (1.26)$$

Calculate this probability for the entangled state $\hat{\rho}_{\text{ent}} = |\psi_{\text{ent}}\rangle_{\text{AB}}\langle\psi_{\text{ent}}|$. Without loss of generality you can take $a = +1$ and $b = +1$, since you can always change the sign by inverting the corresponding direction \mathbf{n}_A or \mathbf{n}_B . Show that:

$$\text{Prob}(\mathcal{A}_+ \wedge \mathcal{B}_+|\hat{\rho}) = \frac{1 - \mathbf{n}_A \cdot \mathbf{n}_B}{4}. \quad (1.27)$$

1 **Rotational invariance.** Notice how the result of the calculation depends only on the scalar product $\mathbf{n}_A \cdot \mathbf{n}_B$ between the two measurement directions \mathbf{n}_A and \mathbf{n}_B at the two stations. This is evidently a consequence of the *rotational invariance* of the singlet entangled state.

Exercise 1.6. (Measurements on a mixed state.)

Contrast the previous result with that obtained by assuming a mixed state of the form:

$$\hat{\rho}_{\text{mix}} = \frac{1}{2} |\psi_{\uparrow\downarrow}\rangle\langle\psi_{\uparrow\downarrow}| + \frac{1}{2} |\psi_{\downarrow\uparrow}\rangle\langle\psi_{\downarrow\uparrow}| \quad (1.28)$$

where $|\psi_{\uparrow\downarrow}\rangle = |+, \mathbf{z}\rangle_A \otimes |-, \mathbf{z}\rangle_B$ and $|\psi_{\downarrow\uparrow}\rangle = |-, \mathbf{z}\rangle_A \otimes |+, \mathbf{z}\rangle_B$. Calculate $\text{Prob}(\mathcal{A}_+ \wedge \mathcal{B}_+ | \hat{\rho}_{\text{mix}})$.

We will now denote by Z_A the event in which A gets the eigenvalue $+1$ by measuring $\mathbf{z} \cdot \hat{\sigma}_A = \hat{\sigma}_A^z$. Similarly, Θ_A denotes the event in which A finds the eigenvalue $+1$ by measuring $\mathbf{n}_\theta \cdot \hat{\sigma}_A$, with $\mathbf{n}_\theta = \mathbf{z} \cos \theta + \mathbf{x} \sin \theta$, and X_A the event in which A gets the eigenvalue $+1$ by measuring $\mathbf{x} \cdot \hat{\sigma}_A = \hat{\sigma}_A^x$. Similar definitions apply for B .

Exercise 1.7. (Joint probabilities on the entangled and on the mixed state.)

Show that:

$$\text{Prob}(Z_A \wedge \Theta_B | \hat{\rho}_{\text{ent}}) = \frac{1}{4}(1 - \cos \theta) \quad \text{and} \quad \text{Prob}(\Theta_A \wedge X_B | \hat{\rho}_{\text{ent}}) = \frac{1}{4}(1 - \sin \theta). \quad (1.29)$$

Contrast these results with those calculated on the mixed state, where you should show that:

$$\text{Prob}(Z_A \wedge \Theta_B | \hat{\rho}_{\text{mix}}) = \frac{1}{4}(1 - \cos \theta) \quad \text{and} \quad \text{Prob}(\Theta_A \wedge X_B | \hat{\rho}_{\text{mix}}) = \frac{1}{4}. \quad (1.30)$$

2 **Rotational invariance, again.** Observe how the result for the entangled state is perfectly consistent with rotational invariance, since when $\theta \rightarrow \frac{\pi}{2} - \theta$, then $\cos \theta \rightarrow \sin \theta$, in Eq. (1.29). Remarkably, the mixed state is *not* rotationally invariant.

Evidently, the only *non-classical* measurement is that of $\Theta_A \wedge X_B$. How should we pin down, unambiguously, that QM is right in predicting that the state is entangled and not mixed? More generally, how would a “classical world” work for these measurements? In a classical world, since the outcomes of the measurements *along the same direction* in the two stations are perfectly anti-correlated, one would be inclined to assume that, for instance, $P(\Theta_A \wedge X_B) = P(\Theta_A \wedge \bar{X}_A)$. Notice that, in QM, it would not be possible to measure the spin along \mathbf{n}_θ and the spin along $-\mathbf{x}$ in the same experiment performed by A , because spin operators along different directions do not commute! But in a classical world it is quite reasonable to assume that in all events in which B has found $+1$ when measuring the spin in the $+\mathbf{x}$ direction, A would have a spin in the $-\mathbf{x}$ direction, even if that was not observed at all.

Now comes a very simple theorem of “classical logic”, to rescue us.

i

Bell's inequality. Given three events A, B, and C, the number of times in which A *and* not-C = \bar{C} occurs — which we denote by $N(A \wedge \bar{C})$ — is not larger than the sum of the number of times in which $A \wedge \bar{B}$ occurs plus those in which $B \wedge \bar{C}$ occurs. In formulas:

$$N(A \wedge \bar{C}) \leq N(A \wedge \bar{B}) + N(B \wedge \bar{C}) . \quad (1.31)$$

By rescaling by the total number of events, a similar inequality works for the corresponding probabilities:

$$P(A \wedge \bar{C}) \leq P(A \wedge \bar{B}) + P(B \wedge \bar{C}) . \quad (1.32)$$

To exemplify. In a large crowd of people assembled in a room, let us consider these three properties:

$$\begin{cases} A = \text{being taller than 165 cm} \\ B = \text{do not wear a pullover} \\ C = \text{do not wear jeans} \end{cases}$$

Then, strange but true:

$$N(\text{taller than 165} \wedge \text{with jeans}) \leq N(\text{taller than 165} \wedge \text{with pullover}) + N(\text{no pullover} \wedge \text{with jeans}) .$$

The proof of this inequality is really simple:

$$\begin{aligned} N(A \wedge \bar{C}) &= N(A \wedge \bar{B} \wedge \bar{C}) + N(A \wedge B \wedge \bar{C}) \\ &\leq N(A \wedge \bar{B} \wedge \bar{C}) + N(A \wedge B \wedge \bar{C}) + N(A \wedge \bar{B} \wedge C) + N(\bar{A} \wedge B \wedge \bar{C}) \\ &= \underbrace{N(A \wedge \bar{B} \wedge \bar{C}) + N(A \wedge \bar{B} \wedge C)}_{N(A \wedge \bar{B})} + \underbrace{N(A \wedge B \wedge \bar{C}) + N(\bar{A} \wedge B \wedge \bar{C})}_{N(B \wedge \bar{C})} \\ &= N(A \wedge \bar{B}) + N(B \wedge \bar{C}) . \end{aligned} \quad (1.33)$$

A few comments, to stress the crucial ingredients behind the theorem. In the first equality we added B, which, even if *unobserved*, must, in our classical world, be either B or \bar{B} . Next, we transformed equality into inequality by adding positive terms and rearranging things. Finally, we used again, twice, the trick that if both C and \bar{C} appear, then you can eliminate C and the same for A. Evidently, the inequality assumes that:

Tertium non datur) Classical logic works: either A or \bar{A} is true, but *not both*.

Reality without observation) The *reality* of objects or of their properties *exists independently of our observation*.

Now we return to the EPR experiment. We interpret $A \mapsto Z_A$, $B \mapsto \Theta_A$ and $C \mapsto X_A$, and we *assume* that $P(Z_A \wedge \bar{X}_A) \mapsto P(Z_A \wedge X_B)$, $P(Z_A \wedge \bar{\Theta}_A) \mapsto P(Z_A \wedge \Theta_B)$, and $P(\Theta_A \wedge \bar{X}_A) \mapsto P(\Theta_A \wedge X_B)$.

Exercise 1.8. (Inequality for mixed and entangled states.)

Show that for the mixed state, the inequality is perfectly satisfied:

$$\text{Prob}(Z_A \wedge X_B | \hat{\rho}_{\text{mix}}) \leq \text{Prob}(Z_A \wedge \Theta_B | \hat{\rho}_{\text{mix}}) + \text{Prob}(\Theta_A \wedge X_B | \hat{\rho}_{\text{mix}}) . \quad (1.34)$$

On the contrary, show that for the entangled state the inequality is violated for all $\theta \in (0, \frac{\pi}{2})$:

$$\text{Prob}(Z_A \wedge X_B | \hat{\rho}_{\text{ent}}) \geq \text{Prob}(Z_A \wedge \Theta_B | \hat{\rho}_{\text{ent}}) + \text{Prob}(\Theta_A \wedge X_B | \hat{\rho}_{\text{ent}}) . \quad (1.35)$$

What is the value of θ for which the violation is maximal?

1 **Classical vs Quantum correlations.** The result of these series of exercises shows that mixed states encode essentially *classical correlations*, while entangled states have genuinely *quantum correlations* which escape any classical explanation.

2. Classical gates and elements of classical computation

I review here some elementary facts about classical computation with logic gates. I start with an introduction to classical gates, with some excursions into Pauli matrices and quantum mechanics. The main references are the book by Mermin [1], and the [lecture notes](#) by Aaronson.

2.1. Classical bits, probability distributions and Stochastic Matrices

Consider first a single bit, with a basis of states which we start denoting in a “quantum-like” way, as $\{|0\rangle, |1\rangle\}$, where, beware, superpositions are *not allowed*.

Returning to our discussion of probabilities and stochastic matrices of Sec. 1.2, let us look at transition matrices in classical probability more closely. A probability distribution for a single classical bit — from now on a “Cbit” — is described by:

$$\mathbf{p} = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \quad \text{with} \quad p_0, p_1 \geq 0, \quad p_0 + p_1 = 1.$$

A 2×2 *transition matrix* would tell us the *conditional probability* of moving between the states in a single step:

$$\mathbf{T} = \begin{pmatrix} P(0|0) & P(0|1) \\ P(1|0) & P(1|1) \end{pmatrix}.$$

Here are a few examples, some of which we have already encountered.

Bit flips) A transition matrix that flips the state is:

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mapsto \hat{\sigma}^x, \quad (2.1)$$

where the right-hand side highlights the immediate connection with one of the Pauli matrices. Recall that this is precisely what the mirror \mathbf{M} does in the classical discussion of the Mach-Zehnder interferometer in Sec. 1.2.1. By applying \mathbf{X} :

$$\mathbf{p}_{\text{fin}} = \mathbf{X} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_0 \end{pmatrix}.$$

Hence, the probability that the Cbit is in $|0\rangle$ is equal to the probability that it *was* in $|1\rangle$.

Fair coin-tossing) Consider:

$$\mathbf{T} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \quad (2.2)$$

It is a kind of fair coin-tossing: independently of the probabilities p_0 and p_1 we had — for instance, those with which the bit was generated — *after* the transformation \mathbf{T} we have:

$$\mathbf{p}_{\text{fin}} = \mathbf{T} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}.$$

You recognise our classical version of a beam-splitter in Sec. 1.2.1.

Erase) Consider now the transition matrix:

$$\mathbf{E}_{\text{rase}} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} .$$

It describes the *erasing* of the Cbit, by which the state ends up being $|0\rangle$ independently of the input:

$$\mathbf{p}_{\text{fin}} = \mathbf{E}_{\text{rase}} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} .$$

Another example) Suppose we generate a Cbit transformation in the following way. If we get 1, we generate a new fair random bit, $\frac{1}{2} - \frac{1}{2}$ distributed; If we get 0, we flip it to 1. The transformation matrix is evidently:

$$\mathbf{T} = \begin{pmatrix} 0 & \frac{1}{2} \\ 1 & \frac{1}{2} \end{pmatrix} ,$$

where the two columns encode precisely the prose I have written.

Stochastic matrices. In general, an $N \times N$ stochastic matrix (SM) \mathbf{T} must obey the following conditions:

$$\mathbf{1) } T_{ij} \in \mathbb{R} \quad \mathbf{2) } T_{ij} \geq 0 \quad \mathbf{3) } \sum_i T_{ij} = 1 \quad \forall j . \quad (2.3)$$

In words: a *real and non-negative* matrix with the elements in each *column* summing to 1.

All these requirements are crucial for \mathbf{T} to be a legitimate transition matrix for a probability distribution. In particular **3)** is crucial for probability conservation.

Indeed, suppose that \mathbf{P}_{in} is a *classical pure state*, i.e., a probability distribution all concentrated into a point, Krönecker-like:

$$\mathbf{P}_{\text{in}} = \mathbf{P}_{\text{in}}^{(j)} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \leftarrow j^{\text{th}} \text{ position} . \quad (2.4)$$

Then, you immediately deduce that $\mathbf{P}_{\text{fin}} = \mathbf{T}\mathbf{P}_{\text{in}}^{(j)}$ is such that $(\mathbf{P}_{\text{fin}})_i = T_{ij}$, and therefore $\sum_i (\mathbf{P}_{\text{fin}})_i = 1$ indeed requires condition **3)**.

Unique decomposition of \mathbf{P}_{in} in classical pure states. Observe that the decomposition of an arbitrary \mathbf{P}_{in} into *classical pure states* is *unique* — unlike, as we shall discuss later on, the decomposition of quantum mixed states $\hat{\rho}$ into quantum pure states. Hence, by linearity, probability conservation follows from the SM nature of \mathbf{T} quite generally.

A noteworthy example of a stochastic matrix is a *permutation matrix* \mathbf{P} , which has exactly a *single* 1 in each column and each row, all the other elements being zero. We will encounter permutation matrices later on when discussing *swaps* of Cbits.

2.2. More than one Cbit: tensor products

Let's examine more closely the case of $n = 2$ Cbits. The possible configuration space is now $2^2 = 4$ -dimensional: $\{0,1\}^2$. Let us insist on our quantum-like notation. We will count the Cbits starting from 0 — so, Cbit 0 and Cbit 1 — and order the elements in the configuration space of two Cbits with the convention that Cbit 0 stays to the *right* of Cbit 1 and increases faster, as follows:

$$\{ |0\rangle|0\rangle \equiv |00\rangle, |0\rangle|1\rangle \equiv |01\rangle, |1\rangle|0\rangle \equiv |10\rangle, |1\rangle|1\rangle \equiv |11\rangle \},$$

where we will often take the liberty, to spare typing, of joining together the two Cbits inside the same “ket”.

As you see, this convention is very natural when thinking in terms of *binary strings*: indeed, you automatically read the binary strings associated with the numbers from 0 to $2^n - 1 = 3$.

$$\{ |0\rangle_2 = |00\rangle, |1\rangle_2 = |01\rangle, |2\rangle_2 = |10\rangle, |3\rangle_2 = |11\rangle \}.$$



Warning: Observe that to uniquely specify the state with the equivalent “integer notation”, I need to specify the number of Cbits, hence the subscript 2 in the “kets”. Indeed, the state associated with the integer 2 for 4 Cbits would be written as: $|2\rangle_4 = |0010\rangle$.

Now, let us suppose that Cbit-0 has a probability distribution $\mathbf{q} = (q_0, q_1)^T$, and Cbit-1 has a probability distribution $\mathbf{p} = (p_0, p_1)^T$, and the two Cbits are *uncorrelated*. How should I describe a probability distribution in the 4-dimensional space we just wrote? It turns out that mathematics gives us the tool: just the *tensor product* of the two vectors \mathbf{p} and \mathbf{q} . More in detail:

$$\underbrace{\mathbf{p}}_{\text{Cbit 1}} \otimes \underbrace{\mathbf{q}}_{\text{Cbit 0}} = \begin{pmatrix} p_0 q_0 \\ p_0 q_1 \\ p_1 q_0 \\ p_1 q_1 \end{pmatrix}. \quad (2.5)$$



The structure of the tensor product. Carefully observe the structure of such a tensor product, which is essentially the same for higher-dimensional vectors. In particular, the central expression is written in “block-form”, with the vector \mathbf{q} not spelt out in components. In the final expression, all components are spelt out, and the final vector is 4-dimensional, as it should. Notice, however, how the components of the vector \mathbf{q} , related to Cbit 0, advance faster. Notice, finally, that the final 4-dimensional vector is fully *factorised* in the two probability distribution, as indeed appropriate for *uncorrelated distribution*.

Not all probability distributions are factorised in this way. For instance:

$$\begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \end{pmatrix},$$

is a legitimate probability distribution which, however, is *not separable*, as you should try to prove: hence the Cbits are *correlated*.¹

¹Notice that, with small variants, you get uncorrelated distributions. For instance, take $\mathbf{p} = (\frac{1}{2}, \frac{1}{2})^T$ and $\mathbf{q} = (1, 0)^T$.

In a similar manner, we can consider tensor products of *operations* on the bits. Given a 2-dim matrix \mathbf{A} that we want to apply to Cbit-1, and a 2-dim matrix \mathbf{B} to apply to Cbit-0, the $\mathbf{A} \otimes \mathbf{B}$ is defined as:

$$\mathbf{A} \otimes \mathbf{B} = \left(\begin{array}{c|c} a_{00}\mathbf{B} & a_{01}\mathbf{B} \\ \hline a_{10}\mathbf{B} & a_{11}\mathbf{B} \end{array} \right) = \left(\begin{array}{cc|cc} a_{00}b_{00} & a_{00}b_{01} & a_{01}b_{00} & a_{01}b_{01} \\ a_{00}b_{10} & a_{00}b_{11} & a_{01}b_{10} & a_{01}b_{11} \\ \hline a_{10}b_{00} & a_{10}b_{01} & a_{11}b_{00} & a_{11}b_{01} \\ a_{10}b_{10} & a_{10}b_{11} & a_{11}b_{10} & a_{11}b_{11} \end{array} \right). \quad (2.6)$$

Again, carefully observe the structure, in particular the middle “block-form” which is much easier to read, and generalised straightforwardly to large matrices.

i **Python kron.** Python does these products for you in one line. Simply call the function `numpy.kron(A,B)` — which stands for Krönecker product. You can even do that by using sparse matrices, with `scipy.kron`. This is a very handy tool to write a python code that diagonalises a small spin-1/2 chain model in just a few lines of code, starting from the Pauli matrices, and using `kron` to define their tensor product with identities.

Exercise 2.1. (Tensor product operations applied to separable states.) Show that:

$$\mathbf{A} \otimes \mathbf{B} (\mathbf{p} \otimes \mathbf{q}) = (\mathbf{A}\mathbf{p}) \otimes (\mathbf{B}\mathbf{q}).$$

Hence, tensor products of operations act *independently* on a two-Cbit separable state.

Let us now consider probably the most important 2-bit operation — or gate — in the entire course: the control-NOT or cNOT. We will indicate it as:

$$\mathbf{C}_{10} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right) = \left(\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right)_1 \otimes \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right)_0 + \left(\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array} \right)_1 \otimes \left(\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right)_0. \quad (2.7)$$

You should try to reproduce the 4×4 expression, as an exercise to train your tensor-product abilities. Notice the extra subscript indices added to the tensor products on the right-hand side, that might be omitted at this stage: they refer to the Cbit on which the 2×2 matrix operates. Incidentally, \mathbf{C}_{10} is a SM, but the very fact that we have an expression involving a *sum of two tensor-products* implies that, when acting on a separable state, this operation will lead to a *non-separable (correlated) state*. Try to apply it to the separable state $(\frac{1}{2}, 0, \frac{1}{2}, 0)^T$ and see what you get.

i **Projectors on Cbit states.** It is useful to introduce the following two 2×2 operations:

$$\mathbf{N}_0 = \left(\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right) \quad \text{and} \quad \mathbf{N}_1 = \left(\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array} \right) \quad (2.8)$$

which act precisely as *projectors* would do in QM:

$$\mathbf{N}_0^2 = \mathbf{N}_0 \quad \mathbf{N}_1^2 = \mathbf{N}_1 \quad \mathbf{N}_0\mathbf{N}_1 = \mathbf{N}_1\mathbf{N}_0 = \mathbf{0} \quad \mathbf{N}_0 + \mathbf{N}_1 = \mathbf{1}. \quad (2.9)$$

Then:

$$\mathbf{p} \otimes \mathbf{q} = \left(\begin{array}{c} \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \end{array} \right).$$



Notation. The notation used by Mermin [1] is $\mathbf{N}_1 \rightarrow \mathbf{n}$ and $\mathbf{N}_0 \rightarrow \tilde{\mathbf{n}}$, useful to spare a subscript, which you can later use for site indices, but slightly confusing for me. The drawback of my notation is that I will have to use parenthesis before a site-index subscript, e.g., in $(\mathbf{N}_0)_1$, which means the \mathbf{N}_0 projector on Cbit 1. Choose your own.

Armed with this two projectors, we re-examine our \mathbf{C}_{10} expression and we write it as follows.



The \mathbf{C}_{10} , a cNOT with 1 as control bit, and 0 as target bit.

$$\mathbf{C}_{10} = (\mathbf{N}_0)_1 \otimes \mathbf{1}_0 + (\mathbf{N}_1)_1 \otimes \mathbf{X}_0 = (\mathbf{N}_0)_1 + (\mathbf{N}_1)_1 \mathbf{X}_0, \quad (2.10)$$

where you should observe that second shorter expression is unambiguous, if you use Cbit-indices as subscripts, even omitting identities and explicit tensor product signs. Notice the great readability: The first term tells us that “if Cbit 1 is in $|0\rangle$, then \mathbf{C}_{10} acts as the identity”, doing nothing. The second term tells us that “if Cbit-1 is in $|1\rangle$, then \mathbf{C}_{10} *flips* the target Cbit-0, using \mathbf{X}_0 . To put it with different words: *The control-bit is unchanged. The target-bit is flipped if the control-bit is in 1.*

Before proceeding, let us pause for a second to notice that, although not directly involved in a classical computation (because of a conspicuous minus sign), there is another Pauli matrix which plays a role in the game.



The \mathbf{Z} -Pauli matrix. Similarly to Eq. (2.1), we introduce:

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \mapsto \hat{\sigma}^z. \quad (2.11)$$

This has the nice feature that projectors \mathbf{N}_0 and \mathbf{N}_1 have the familiar QM expression:

$$\mathbf{N}_0 = \frac{1}{2}(\mathbf{1} + \mathbf{Z}) \mapsto \hat{\Pi}_\uparrow \quad \text{and} \quad \mathbf{N}_1 = \frac{1}{2}(\mathbf{1} - \mathbf{Z}) \mapsto \hat{\Pi}_\downarrow. \quad (2.12)$$

Hence the *natural* identification we will later do is that, in terms of spin-1/2 states: $|0\rangle \mapsto |\uparrow\rangle$, and $|1\rangle \mapsto |\downarrow\rangle$. Do not even dare to change this convention: your algebra will be scrambled.

Using \mathbf{Z} we can equivalently re-express the cNOT as follows:

$$\mathbf{C}_{10} = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_1 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_1 \mathbf{X}_0, \quad (2.13)$$

where we have omitted identity and tensor-product signs: the bit-indices make the operation unambiguous.

2.3. More on Cbit operations: connection to digital computer operations

Let us start generalising our notation. If I have n Cbits, I can write integers from 0 up to $2^n - 1 = N - 1$, with $N = 2^n$.

1

n -Cbit configurations. The space of classical Cbit configurations will be denoted as:

$$|x\rangle_n = |x_{n-1}\rangle|x_{n-2}\rangle \cdots |x_1\rangle|x_0\rangle \equiv |x_{n-1}x_{n-2} \cdots x_1x_0\rangle \quad (2.14)$$

where:

$$x = \sum_{j=0}^{n-1} x_j 2^j \quad \text{with} \quad x_j = 0, 1 \quad (2.15)$$

We will use this identification of integers $0 \leq x \leq 2^n - 1$ with their corresponding binary string $x \mapsto (x_{n-1}x_{n-2} \cdots x_1x_0) \in \{0, 1\}^n$ very often in the following. Familiarise with it.

Quite amusingly, this notation naturally leads to *tensor products*. Let us review this for $n = 1, 2, 3$. For $n = 1$:

$$|0\rangle_1 \equiv |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle_1 \equiv |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (2.16)$$

where the 2-dim column vectors on the RHS are the standard column vectors for the chosen basis. Incidentally, this is how you would write the corresponding $\hat{\sigma}^z$ spinors.

For $n = 2$:

$$|0\rangle_2 = |00\rangle \equiv |0\rangle|0\rangle \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |1\rangle_2 = |01\rangle \equiv |0\rangle|1\rangle \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|2\rangle_2 = |10\rangle \equiv |1\rangle|0\rangle \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |3\rangle_2 = |11\rangle \equiv |1\rangle|1\rangle \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

where you observe that the RHS 4-dim column has exactly a single 1 in the position indicated by the integer, starting from 0. This is not a coincidence. If you want a further example for $n = 3$ Cbits:

$$|5\rangle_3 = |101\rangle = |1\rangle|0\rangle|1\rangle \mapsto \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

hence a 1 in the 5th-position, assuming that the column arrays are numbered starting from 0, as python or C++ would do! This works beautifully for any n .

i

Configurations as classical pure states. You can show that:

$$|x\rangle_n = |x_{n-1}x_{n-2}\cdots x_1x_0\rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \leftarrow \text{a single 1 at position } x \quad (2.17)$$

with the ultra-strong *warning* that *superpositions are not allowed*. This is a “must” for a classical Cbit configuration. Probability distributions in this N -dimensional space — although not representing computer states — can formally, and uniquely, be decomposed as:

$$\mathbf{p} = (p_0, \cdots p_{N-1})^T = \sum_{x=0}^{N-1} p_x |x\rangle_n \quad \text{with } p_x \geq 0 \text{ and } \sum_x p_x = 1. \quad (2.18)$$

The normalisation condition can be also re-written as $\|\mathbf{p}\|_1 = 1$, where you should observe the presence of the so-called 1-norm, as opposed to the standard 2-norm of QM.

X again.) Let us return to bit operations. Consider again \mathbf{X} , the bit flip. For a single Cbit:

$$\mathbf{X}|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle,$$

and similarly $\mathbf{X}|1\rangle = |0\rangle$.

i

Bit flip alias NOT gate. If $x = 0, 1$, then we denote $\bar{x} = 1 - x$, hence $\bar{0} = 1$ and $\bar{1} = 0$. You recognise the NOT of Boolean logic, and this is indeed an alternative name for the \mathbf{X} gate. More generally, denote by $\mathbf{X}_j \equiv \mathbf{1} \otimes \cdots \otimes \mathbf{1} \otimes \mathbf{X} \otimes \mathbf{1} \cdots \otimes \mathbf{1}$, where \mathbf{X} sits at position j , so as to be free from the horribly long sequence of identities and tensor products. Then:

$$\mathbf{X}_j |x_{n-1}x_{n-2}\cdots x_j \cdots x_1x_0\rangle = |x_{n-1}x_{n-2}\cdots \bar{x}_j \cdots x_1x_0\rangle. \quad (2.19)$$

Another useful connection of the \mathbf{X} gate is with addition (mod 2), denoted by \oplus , or the Boolean logic **XOR**. Indeed, if $x = 0, 1$:

$$x = x \oplus 0 \quad \text{and} \quad \bar{x} = x \oplus 1 \quad \implies \quad \mathbf{X}|x\rangle_1 = |x \oplus 1\rangle_1, \quad (2.20)$$

a relationship that will turn out useful later on.

Reversible versus irreversible gates.) \mathbf{X}_j and $\mathbf{1}_j$ exhaust the *reversible* single-Cbit gates. By reversible we mean that you can invert the operation, so that to each output corresponds a unique input. Indeed, since

$$\mathbf{X}^2 = \mathbf{1} \quad \implies \quad \mathbf{X}^{-1} = \mathbf{X}.$$

Any other single-Cbit operation is not reversible. For instance, the

$$\mathbf{E}_{\text{rase}} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

is a SM, but not invertible, as it sends any input into the output 0. It is used to erase a memory bit, putting it into a standard state 0. By the **Landauer principle**, you might recall that such *irreversibility* is associated to *dissipation*: the entropy of the memory is reduced, but the entropy of the *environment* has to increase in such a way that $\Delta S^{\text{tot}} \geq 0$, by the 2nd principle of Thermodynamics. More to our point, since the Schrödinger dynamics is *unitary* and hence reversible, in thinking about carrying out calculations with a Quantum Computer, we will look for *reversible gates*, staying away from irreversibility. More about this later on.

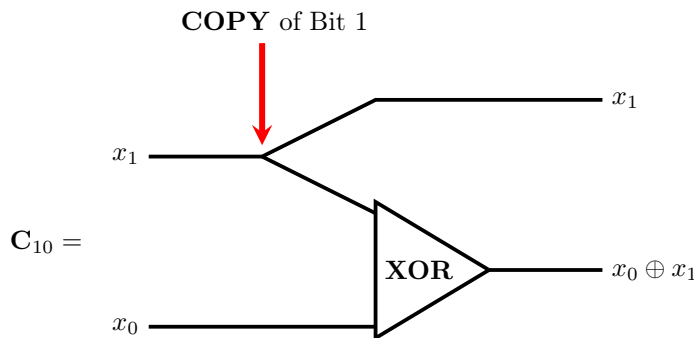


Figure 2.1: A schematic representation of \mathbf{C}_{10} , involving a **COPY** and a **XOR**.

\mathbf{C}_{10} , XOR, and COPY.) Let us revisit the \mathbf{C}_{10} cNOT gate again. On the computational basis of 2-Cbits we might write:

$$\mathbf{C}_{10}|x_1x_0\rangle_2 = \mathbf{C}_{10}|x_1\rangle|x_0\rangle = |x_1\rangle|x_0 \oplus x_1\rangle, \quad (2.21)$$

where you would recall that $x_0 \oplus x_1 = x_0$ for $x_1 = 0$ (identity), while $x_0 \oplus x_1 = \bar{x}_0$ for $x_1 = 1$ (bit flip). This is represented in Fig. 2.1. Observe that the **XOR** operation

$$\mathbf{XOR}|x_1\rangle|x_0\rangle = |x_1 \oplus x_0\rangle = |x_1 + x_0 \pmod{2}\rangle,$$

by itself, as any function $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ in *not reversible*. But if you supplement it with a **COPY** operation

$$\mathbf{COPY}|x\rangle = |x\rangle|x\rangle \quad \text{where} \quad \mathbf{COPY} : \{0, 1\} \rightarrow \{0, 1\}^2,$$

then the **XOR** is made *invertible*: it is the cNOT operation \mathbf{C}_{10} . Incidentally, the cNOT operates like a copy when acting on the target 0:

$$\mathbf{C}_{10}|x_1\rangle|0\rangle = |x_1\rangle|0 \oplus x_1\rangle = |x_1\rangle|x_1\rangle = \mathbf{COPY}|x_1\rangle.$$

The SWAP gate.) Consider the gate that swaps two bits:

$$\mathbf{S}_{10}|x_1x_0\rangle = |x_0x_1\rangle.$$

This time the two bits appear *symmetrically* in the operation, hence $\mathbf{S}_{10} = \mathbf{S}_{01}$. A 4×4 matrix representation of \mathbf{S}_{10} on the computational basis is:

$$\mathbf{S}_{10} = \mathbf{S}_{01} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right). \quad (2.22)$$

Hence, once again, like for \mathbf{C}_{10} , a particular SM which is also a *permutation matrix*, not surprisingly, perhaps. Notice that the way we have represented it in block-form does not do justice to the basic fact that the whole non-trivial action occurs in the central 2×2 block (recall that $00 \rightarrow 00$ and $11 \rightarrow 11$, while $01 \rightarrow 10$ and $10 \rightarrow 01$: this is the way you write the matrix representation right away). Unlike Eq. (2.7), where we decomposed the matrix into a sum of two tensor products, we prefer to suffer a

bit more now and free ourselves from the nightmare of explicit tensor products in favour of a much cleaner and simple tool: the algebra of Pauli matrices, which you learned in QM.

Let's start. You recall that Pauli matrices anti-commute: $\mathbf{Z}\mathbf{X} = -\mathbf{X}\mathbf{Z}$. This fact, together with the expressions for the projectors \mathbf{N}_0 and \mathbf{N}_1 in terms of \mathbf{Z} , see Eq. (2.12), immediately leads to:

$$\mathbf{X}\mathbf{N}_0 = \mathbf{N}_1\mathbf{X} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \mapsto \hat{\sigma}^- \quad \text{and} \quad \mathbf{X}\mathbf{N}_1 = \mathbf{N}_0\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mapsto \hat{\sigma}^+, \quad (2.23)$$

where, incidentally, we recognise the familiar $\hat{\sigma}^\pm$. Now we look at Eq. (2.22), and we recognise the following 4 tensor-product ingredients (try yourself):

$$\mathbf{S}_{10} = \underbrace{(\mathbf{N}_0)_1 \otimes (\mathbf{N}_0)_0}_{\text{element 11}} + \underbrace{(\mathbf{N}_1)_1 \otimes (\mathbf{N}_1)_0}_{\text{element 44}} + \underbrace{(\mathbf{X}\mathbf{N}_1)_1 \otimes (\mathbf{X}\mathbf{N}_0)_0}_{\text{element 23}} + \underbrace{(\mathbf{X}\mathbf{N}_0)_1 \otimes (\mathbf{X}\mathbf{N}_1)_0}_{\text{element 32}}.$$

Now we substitute the projectors with \mathbf{Z} , see Eq. (2.12), and recall that $\mathbf{X}\mathbf{Z} = -i\mathbf{Y}$, where $\mathbf{Y} = \hat{\sigma}^y$, obtaining (after cancellations):

$$\begin{aligned} \mathbf{S}_{10} &= \frac{1}{4}(\mathbf{1} + \mathbf{Z})_1 \otimes (\mathbf{1} + \mathbf{Z})_0 + \frac{1}{4}(\mathbf{1} - \mathbf{Z})_1 \otimes (\mathbf{1} - \mathbf{Z})_0 + \\ &+ \frac{1}{4}(\mathbf{X}(\mathbf{1} - \mathbf{Z}))_1 \otimes (\mathbf{X}(\mathbf{1} + \mathbf{Z}))_0 + \frac{1}{4}(\mathbf{X}(\mathbf{1} + \mathbf{Z}))_1 \otimes (\mathbf{X}(\mathbf{1} - \mathbf{Z}))_0 \\ &= \frac{1}{2}\mathbf{1}_1 \otimes \mathbf{1}_0 + \frac{1}{2}\mathbf{Z}_1 \otimes \mathbf{Z}_0 + \frac{1}{2}\mathbf{X}_1 \otimes \mathbf{X}_0 - \frac{1}{2} \underbrace{(\mathbf{X}\mathbf{Z})_1 \otimes (\mathbf{X}\mathbf{Z})_0}_{-\mathbf{Y}_1 \otimes \mathbf{Y}_0} \\ &= \frac{1}{2}\mathbf{1}_1 \otimes \mathbf{1}_0 + \frac{1}{2}\mathbf{X}_1 \otimes \mathbf{X}_0 + \frac{1}{2}\mathbf{Y}_1 \otimes \mathbf{Y}_0 + \frac{1}{2}\mathbf{Z}_1 \otimes \mathbf{Z}_0 \\ &= \frac{1}{2}(\mathbf{1} + \hat{\sigma}_1 \cdot \hat{\sigma}_0). \end{aligned} \quad (2.24)$$

Notice that \mathbf{Y} has appeared. Also, notice that in the last expression we got rid of the tensor products: the $\mathbf{1}$ is acting on the 4-dimensional space, and $\hat{\sigma}_1 \cdot \hat{\sigma}_0$ means precisely the same thing as:

$$\hat{\sigma}_1 \cdot \hat{\sigma}_0 \equiv \mathbf{X}_1 \otimes \mathbf{X}_0 + \mathbf{Y}_1 \otimes \mathbf{Y}_0 + \mathbf{Z}_1 \otimes \mathbf{Z}_0.$$

i

Heisenberg model. Those of you who have studied the antiferromagnetic Heisenberg model will recognise its basic 2-site building block. The ground state of such a combination of Pauli matrices would be the *singlet state*, but this is a different story. Remember that we are doing classical computation so far: The SWAP gate is a classical gate! While \mathbf{X} is a legitimate 1-Cbit gate, the \mathbf{Z} and even less so the “complex” \mathbf{Y} are *not* classical 1-Cbit gates. Nevertheless, they are extremely useful in the algebra, to construct the \mathbf{C}_{10} and \mathbf{S}_{10} gates.

Sites indices are enough.) We are now grown-up people, and we can get rid of the tensor products, which can be sometimes very annoying. Imagine that you want to write the SWAP gate, but now for two generic bits i and j of an n -Cbits state:

$$\mathbf{S}_{ij}|x_{n-1}\rangle \cdots |x_i\rangle \cdots |x_j\rangle \cdots |x_0\rangle = |x_{n-1}\rangle \cdots |x_j\rangle \cdots |x_i\rangle \cdots |x_0\rangle.$$

The matrix representing such an object would be an awfully large $2^n \times 2^n$ permutation matrix with just a few 1s out of the diagonal: the rest is an identity. But now we know that this is simply:

$$\mathbf{S}_{ij} = \frac{1}{2}(\mathbf{1} + \hat{\sigma}_i \cdot \hat{\sigma}_j) = \mathbf{S}_{ji},$$

where all identities have been omitted, and the resulting expression is totally unambiguous. Recall, incidentally, that Pauli matrices operating on different sites *commute*, and that the meaning, for instance, of $\mathbf{X}_0\mathbf{X}_3$ is simply:

$$\mathbf{X}_0\mathbf{X}_3 \mapsto \mathbf{1}_{n-1} \otimes \cdots \otimes \mathbf{1}_4 \otimes \mathbf{X}_3 \otimes \mathbf{1}_2 \otimes \mathbf{1}_1 \otimes \mathbf{X}_0 .$$

Quite a sparing of typing. With this freedom we can now write a cNOT gate operating with an arbitrary *control-bit* i , and *target-bit* j .

1 The C_{ij} cNOT, with i as a control-bit, and j as a target-bit.

$$C_{ij} = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_i + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_i\mathbf{X}_j , \tag{2.25}$$

which reads in a quite transparent way: if Cbit i , the *control-Cbit*, is in state $|0\rangle = |\uparrow\rangle$ then do nothing; if the control-Cbit i is in state $|1\rangle = |\downarrow\rangle$, then flip (with \mathbf{X}_j) the *target-Cbit* j . The control-bit never changes.

2.4. Reversible extensions of Boolean functions

In a classical context, reversibility is not required. The reason why we start considering this issue here is twofold. First, and most importantly, we will later need reversibility in the quantum case. Second, irreversibility is associated to dissipation, according to the *Landauer's principle*. For instance, every time you erase a memory bit, you have to release/waste at least $k_B T \log 2$ of heat in the environment, to compensate for the reduced memory entropy. Estimates can be made that the current level of energy dissipation in our digital computers is orders of magnitude higher than such a Landauer's bound. ² The reason for this was pointed out by Bennett, and is not difficult to appreciate: to change a wire voltage, one must dump it to ground through a resistance. Figure

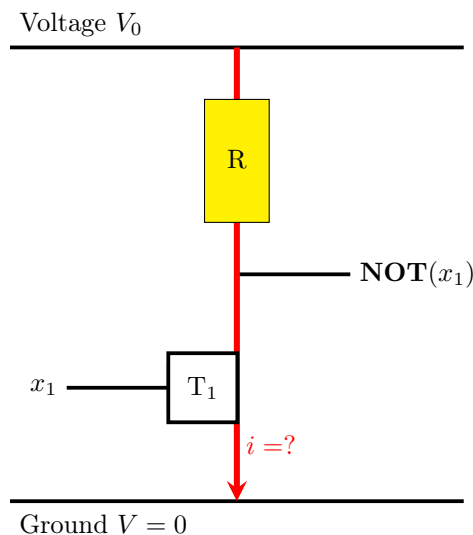


Figure 2.2: A transistor implementation of a **NOT** gate. The boolean logic is implemented by associating a reference voltage V_0 to logic value 1 = TRUE, and voltage $V = 0$ to the logic value 0 = FALSE. The transistor T_1 “conducts” if its gate terminal is at voltage V_0 , while it is “closed” at gate voltage $V = 0$. If $x_1 = 1$ then T_1 “conducts” hence a current $i > 0$ passes through the resistor R , and the gate $\mathbf{NOT}(x_1)$ is at ground, hence $\mathbf{NOT}(x_1) = 0$. If $x_1 = 0$ then T_1 does not conduct, no current passes thorough R and the gate $\mathbf{NOT}(x_1) = 1$ (its voltage is V_0).

2.2 shows an electronic device that implements the **NOT** gate classically, with a Transistor and a Resistor. Our digital computers work by implementing the boolean logic by associating a reference voltage V_0 to logic value 1 = TRUE, and a voltage $V = 0$ to the logic value 0 = FALSE. The physics

²Roughly, 10^{12} bit erasures in a second, imply a minimum dissipation, by Landauer's principle, of $\sim 3 \times 10^{-9}$ W. This is roughly 10^{11} times smaller than the actual energy consumption of our current desktops.

of transistors is such that the transistor “conducts” if its gate terminal is at voltage V_0 , while it is “closed” (like an infinite resistance) at gate voltage $V = 0$. It is clear that to change the value of the bit x_1 from $1 \rightarrow 0$, you need to have a current flowing through the resistor, hence provoking some Joule’s heating. So, lots of energy is wasted through Joules’ heating in resistance elements. Ideally, a reversible classical computation, although more demanding in terms of bits used, as we shall see, would not suffer from such a waste of energy.

So, let us start from the reversibility issue: the construction we present here for the classical case will be used, *verbatim*, in the quantum case to construct a unitary operator \mathbf{U}_f that encodes any Boolean function f .

Any algorithm or computation in a classical digital computer amounts to computing a Boolean function:

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m \quad \text{with} \quad f(\underline{x}) = \underline{y},$$

where \underline{x} is an input binary string of n bits, and \underline{y} the output of the computation, a binary string with m bits. Such a computation is *not in general reversible*. Think of a memory erase, for instance. More generally, if $n > m$, since we are dealing with finite spaces, there must be more than one input \underline{x} for the same output \underline{y} , as we shall soon see for the elementary functions of Boolean logic.

i

Notation. From now on we will often (but not always) get rid of the underline, and simply identify a binary string \underline{x} with the corresponding integer x :

$$\underline{x} = (x_{n-1}, \dots, x_0) \longleftrightarrow x = \sum_{j=0}^{n-1} x_j 2^j \quad \text{with} \quad x_j = 0, 1. \quad (2.26)$$

Let us illustrate the case of $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ which corresponds to the logical **AND**:

$$f_{\wedge}(x_1, x_0) = x_1 \wedge x_0 = x_1 x_0.$$

As you see, $f_{\wedge} = 0$ for three input pairs, 00, 10, and 01. To make it reversible, we would need to extend the function in such a way that “it keeps track of the input variable”. Naively, one would be tempted to define:

$$\tilde{f}_{\wedge}(x_1, x_0) = (x_1, x_0, f_{\wedge}(x_1, x_0)),$$

but this also cannot be invertible, since now the image is a space larger than the domain. The way out is to introduce an *ancillary variable* y in the input string as well, defining a $\tilde{f}_{\wedge}(x_1, x_0, y)$ which now sends $\{0, 1\}^3 \rightarrow \{0, 1\}^3$. But how exactly y should enter in the output of the function? Let’s be slightly more general, and imagine that $f : \{0, 1\}^n \rightarrow \{0, 1\}$ has input on a n -bit binary string, $f(\underline{x})$. We extend it as follows:

$$\tilde{f} : \{0, 1\}^{n+1} \rightarrow \{0, 1\}^{n+1} \quad \text{with} \quad \tilde{f}(\underline{x}, y) = (\underline{x}, y \oplus f(\underline{x})). \quad (2.27)$$

To show that this works, suppose that you have two different inputs \underline{x} and \underline{x}' which led originally to the same output $z = f(\underline{x}) = f(\underline{x}')$. Now we have:

$$\begin{cases} \tilde{f}(\underline{x}, 0) = (\underline{x}, z) & \tilde{f}(\underline{x}', 0) = (\underline{x}', z) \\ \tilde{f}(\underline{x}, 1) = (\underline{x}, \bar{z}) & \tilde{f}(\underline{x}', 1) = (\underline{x}', \bar{z}) \end{cases}, \quad (2.28)$$

hence, different inputs are associated to different outputs, and \tilde{f} is one-to-one. Notice the crucial role played by the ancillary bit involved in the **XOR** in the output: $y \oplus f(\underline{x})$.

1 \tilde{f} properties.)

1) \tilde{f} coincides with its inverse, since we can show that $\tilde{f} \circ \tilde{f} = \text{id} \implies \tilde{f}^{-1} = \tilde{f}$. To show this, observe that:

$$\tilde{f}(\tilde{f}(\underline{x}, y)) = \tilde{f}(\underline{x}, y \oplus f(\underline{x})) = (\underline{x}, y \oplus f(\underline{x}) \oplus f(\underline{x})) = (\underline{x}, y) .$$

2) Very useful is the fact that:

$$\tilde{f}(\underline{x}, y = 0) = (\underline{x}, f(\underline{x})) , \tag{2.29}$$

hence the value of the function $f(\underline{x})$ is directly extracted from setting the ancillary bit $y = 0$.

This trick works as a standard extension for functions with m output variables.

1 **Invertible extension.** For any $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, you can extend it in such a way that it is invertible by introducing m ancillary bits \underline{y} , one for each output variable, defining:

$$\tilde{f} : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^{n+m} \quad \text{with} \quad \tilde{f}(\underline{x}, \underline{y}) = (\underline{x}, \underline{y} \oplus f(\underline{x})) , \tag{2.30}$$

where \oplus on the RHS indicates an m -bitwise \oplus .

Question: is $\underline{y} \oplus f(\underline{x})$ the same thing as $y + f(x) \pmod{2^m}$, as for a single bit?

2.5. Elementary logic gates

Let us now review some elementary Boolean logic functions.








GATE	CIRCUIT SYMBOL	TRUTH TABLE	GATE	CIRCUIT SYMBOL	TRUTH TABLE																				
NOT The output is 1 when the input is 0 and 0 when the input is 1.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	Input	Output	0	1	1	0	NAND The output is 0 only when both inputs are 1, otherwise the output is 1.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>1</td> </tr> <tr> <td>0 1</td> <td>1</td> </tr> <tr> <td>1 0</td> <td>1</td> </tr> <tr> <td>1 1</td> <td>0</td> </tr> </tbody> </table>	Input	Output	0 0	1	0 1	1	1 0	1	1 1	0				
Input	Output																								
0	1																								
1	0																								
Input	Output																								
0 0	1																								
0 1	1																								
1 0	1																								
1 1	0																								
AND The output is 1 only when both inputs are 1, otherwise the output is 0.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> </tr> <tr> <td>0 1</td> <td>0</td> </tr> <tr> <td>1 0</td> <td>0</td> </tr> <tr> <td>1 1</td> <td>1</td> </tr> </tbody> </table>	Input	Output	0 0	0	0 1	0	1 0	0	1 1	1	NOR The output is 1 only when both inputs are 0, otherwise the output is 0.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>1</td> </tr> <tr> <td>0 1</td> <td>0</td> </tr> <tr> <td>1 0</td> <td>0</td> </tr> <tr> <td>1 1</td> <td>0</td> </tr> </tbody> </table>	Input	Output	0 0	1	0 1	0	1 0	0	1 1	0
Input	Output																								
0 0	0																								
0 1	0																								
1 0	0																								
1 1	1																								
Input	Output																								
0 0	1																								
0 1	0																								
1 0	0																								
1 1	0																								
OR The output is 0 only when both inputs are 0, otherwise the output is 1.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> </tr> <tr> <td>0 1</td> <td>1</td> </tr> <tr> <td>1 0</td> <td>1</td> </tr> <tr> <td>1 1</td> <td>1</td> </tr> </tbody> </table>	Input	Output	0 0	0	0 1	1	1 0	1	1 1	1	XOR The output is 1 only when the two inputs have different value, otherwise the output is 0.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> </tr> <tr> <td>0 1</td> <td>1</td> </tr> <tr> <td>1 0</td> <td>1</td> </tr> <tr> <td>1 1</td> <td>0</td> </tr> </tbody> </table>	Input	Output	0 0	0	0 1	1	1 0	1	1 1	0
Input	Output																								
0 0	0																								
0 1	1																								
1 0	1																								
1 1	1																								
Input	Output																								
0 0	0																								
0 1	1																								
1 0	1																								
1 1	0																								
			XNOR The output is 1 only when the two inputs have the same value, otherwise the output is 0.		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>1</td> </tr> <tr> <td>0 1</td> <td>0</td> </tr> <tr> <td>1 0</td> <td>0</td> </tr> <tr> <td>1 1</td> <td>1</td> </tr> </tbody> </table>	Input	Output	0 0	1	0 1	0	1 0	0	1 1	1										
Input	Output																								
0 0	1																								
0 1	0																								
1 0	0																								
1 1	1																								

Figure 2.3.: Illustration of classical gates. Figure taken from Ref. [17][Fig. 8].

1-bit For $f : \{0, 1\} \rightarrow \{0, 1\}$ there are only 4 functions. The identity, the NOT, the Erase and its negation:

$$\text{id}(x) = x \quad \mathbf{NOT}(x) = \bar{x} = 1 - x \quad \mathbf{E}_{\text{rase}}(x) = 0 \quad \mathbf{NOT}(\mathbf{E}_{\text{rase}}(x)) = 1 ,$$

the first two being the only reversible 1-bit functions. The connection to Logic is through the usual identification $0 = \text{FALSE}$, $1 = \text{TRUE}$.

2-bit For $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ there are only 16 different functions, a handful of which are really useful.

x	y	$\mathbf{AND}(x, y)$	$\mathbf{OR}(x, y)$	$\mathbf{XOR}(x, y)$	$\mathbf{NAND}(x, y)$	$\mathbf{NOR}(x, y)$	$\mathbf{E}_{\text{rase}}(x, y)$	x	\dots
0	0	0	0	0	1	1	0	0	
0	1	0	1	1	1	0	0	0	
1	0	0	1	1	1	0	0	1	
1	1	1	1	0	0	0	0	1	

All of them are *not invertible*. Simple formulas can be written for the most important logic functions. Calling the two binary arguments x and y , for simplicity, and recalling that $\mathbf{NOT}(x) = \bar{x} = 1 - x$, we have:

$$\left\{ \begin{array}{l} \mathbf{AND}(x, y) = x \wedge y = xy \\ \mathbf{OR}(x, y) = x \vee y = x + y - xy \\ \mathbf{XOR}(x, y) = x \oplus y = x + y \pmod{2} \\ \mathbf{NAND}(x, y) = \mathbf{NOT}(x \wedge y) = \mathbf{NOT}(xy) = 1 - xy \\ \mathbf{NOR}(x, y) = \mathbf{NOT}(x \vee y) = \mathbf{NOT}(x + y - xy) = 1 - x - y + xy \end{array} \right. . \quad (2.31)$$

Recall also that **AND**, **OR**, and **NOT** are related by De Morgan's identities:

$$\left\{ \begin{array}{l} \mathbf{NAND}(x, y) = \mathbf{NOT}(x \wedge y) = \bar{x} \vee \bar{y} \\ \mathbf{NOR}(x, y) = \mathbf{NOT}(x \vee y) = \bar{x} \wedge \bar{y} \end{array} \right. . \quad (2.32)$$

Notice also that **XOR** can be constructed from **AND**, **OR**, and **NOT**:

$$\mathbf{XOR}(x, y) = (x \vee y) \wedge (\bar{x} \vee \bar{y}) \quad (2.33)$$

2.6. A simple algorithm: adding numbers

Suppose you want to write an algorithm to add two integers, represented by two binary strings $x = (x_{n-1}, \dots, x_0)$ and $y = (y_{m-1}, \dots, y_0)$ — assume without loss of generality $n \geq m$ — to produce their sum s which has $n + 1$ bits:

$$s = x + y = (s_n, \dots, s_0) .$$

To do that, we use the elementary one-bit addition rule: $0 + 0 = 0$, $0 + 1 = 1 + 0 = 1$, but $1 + 1 = 0$ with a carry-over $c = 1$. This means:

$$x + y = x \oplus y \quad \text{with} \quad c = x \wedge y .$$

For instance, to sum $15 + 7$ we have:

$$\begin{array}{rcccccc} 1 & 1 & 1 & 1 & 0 & \longleftarrow & c_i \text{ carry-overs} \\ \hline & 1 & 1 & 1 & 1 & + & \longleftarrow & x = 15 \\ & & 0 & 1 & 1 & 1 & = & \longleftarrow & y = 7 \\ \hline 1 & 0 & 1 & 1 & 0 & & \longleftarrow & s = 22 \end{array} .$$

The algorithm is easy to implement step by step. Here $n > m$ and we set $y_{i>m} = 0$. At step $i = 0$, $c_0 = 0$ and we have:

$$\text{Step } 0 : \quad c_0 = 0 \implies \begin{cases} s_0 = x_0 \oplus y_0 \\ c_1 = x_0 \wedge y_0 \end{cases} .$$

At step $i > 0$ the carry-over c_{i+1} has to be calculated more carefully, because the previous carry-over c_i enters. Indeed $c_{i+1} = 1$ in two cases: **a)** $(x_i = y_i = 1)$ or **b)** $(x_i \oplus y_i = 1 \text{ and } c_i = 1)$. (Case **b**) occurs for bit 3 in the previous example.) Hence the generic step i requires:

$$\text{Step } i \leq n - 1 : \quad c_i \implies \begin{cases} s_i = x_i \oplus y_i \oplus c_i \\ c_{i+1} = (x_i \wedge y_i) \vee ((x_i \oplus y_i) \wedge c_i) \end{cases}$$

Finally, at step $i = n$ we simply put $s_n = c_n$ and we are done.

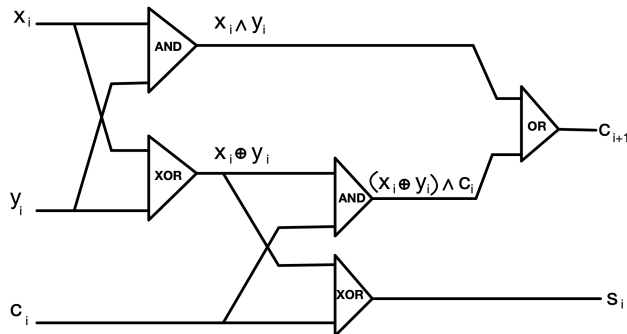


Figure 2.4: The classical circuit performing addition of two bits x_i and y_i with a carry-over c_i . Here $s_i = x_i \oplus y_i \oplus c_i$ with $c_{i+1} = (x_i \wedge y_i) \vee ((x_i \oplus y_i) \wedge c_i)$ as a carry-over for next step.

As you see, the algorithm requires, at each step, 2 **XOR**, 2 **AND**, 1 **OR**, and a certain number of **COPY**, to replicate variables for use in the various gates. Recall also that the **XOR** can be written in terms of **AND**, **OR**, and **NOT**.

2.7. Universal classical gates

The previous simple example brings to mind two questions:

Question:

Q1) Given a generic computation, encoded by $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, what are the elementary gates that guarantee that I can construct an algorithm to calculate f ?

Q2) What is the *minimal set* of gates that I need to have in order to construct such an f ?

Q1) will have a simple answer: it is enough to have **AND**, **OR**, **NOT**, and **COPY** to calculate any function f . These gates can therefore be regarded as *universal*.

Q2) has to do with the minimal number of gates that I should be prepared to implement in the digital hardware. We will see that **NAND** and **COPY** are sufficient to reproduce any gate. The importance of **NAND** is appreciated when you realise that it is rather easy to fabricate using two transistors in series, as illustrated in Fig. 2.5.

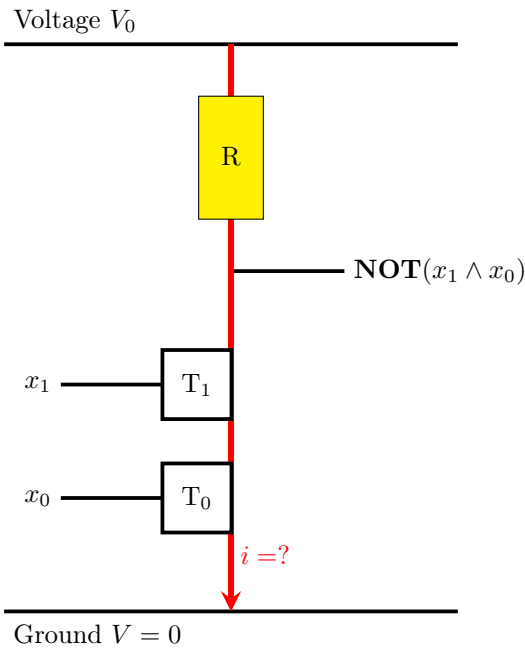


Figure 2.5: A **NAND** gate made with two transistors in series. Recall that $\text{NAND}(x_1, x_0) = \text{NOT}(x_1 \wedge x_0)$. The voltage at the **NAND** node is $+V_0$ (**NAND** = 1) if no current is flowing ($i = 0$). The voltage drops to 0 (**NAND** = 0) when current flows ($i > 0$). This in turns requires that *both* transistors conduct: the voltages applied at x_1 and x_0 must be $+V_0$, hence $x_1 = x_0 = 1$. Observe the presence of the resistor, which will inevitably bring *Joules' heating* when the **NAND** = 0.

We tackle **Q1**) by an explicit construction. One should keep in mind that this construction is, by no means, the best algorithm you can come about to calculate f . It simply shows that you *can* calculate f using only **AND**, **OR**, **NOT**, and **COPY**. The construction is very simple. First of all, it is sufficient to consider $m = 1$, since an m -valued function is built from the m components functions $f_1 \cdots f_m : \{0, 1\}^n \rightarrow \{0, 1\}$.

I have 2^n possible input strings, which we denote by $\underline{x}^{(0)} = (0, 0 \cdots, 0)$, up to $\underline{x}^{(2^n-1)} = (1, 1, \cdots, 1)$. Imagine you construct a *table* for the function f as follows: ³

	x_{n-1}	\cdots	x_1	x_0	f	
$\underline{x}^{(0)}$	\rightarrow 0	0	\cdots	0	0	$\leftarrow a_0$
$\underline{x}^{(1)}$	\rightarrow 0	0	\cdots	0	1	$\leftarrow a_1$
$\underline{x}^{(2)}$	\rightarrow 0	0	\cdots	1	0	$\leftarrow a_2$
\vdots			\cdots		\vdots	\vdots
$\underline{x}^{(J)}$			\cdots		1	$\leftarrow a_J = f(\underline{x}^{(J)})$
\vdots			\cdots		\vdots	\vdots
$\underline{x}^{(2^n-1)}$	\rightarrow 1	1	\cdots	1	1	$\leftarrow a_{2^n-1}$

The values that that function f attains on any input $\underline{x}^{(J)}$, $f(\underline{x}^{(J)}) = a_J$, are encoded in a 2^n -dimensional binary string $(a_{2^n-1}, \cdots, a_1, a_0)$.

³The notation here is a bit baroque, and could be simplified, at the price of some possible ambiguity. For instance, the index $J = 0, \cdots, 2^n - 1$ is simply the integer x associated to a binary string \underline{x} . Hence I could denote $\underline{x}^J \rightarrow x$, $a_J \rightarrow a_x$, and even introduce the Krönercker as $K_x(x') = \delta_{x,x'}$ and go on writing:

$$f(x') = \sum_{x=0}^{2^n-1} a_x K_x(x'),$$

instead of Eq. (2.35). The reason why I didn't do that, is because of the possible ambiguity with the bit variables x_j . No notation is perfect. Choose what you prefer.

1 **How many different f ?** Evidently, since each $a_J = 0, 1$, there are 2^{2^n} different functions f that you can write. The function $f = 0$ erases all bits, and will be considered separately. Obviously, the Erase of a bit x_j can be written in terms of **AND**(x_j, \bar{x}_j), hence by using **COPY**, **NOT** and **AND**.

Among these, there are 2^n very special “pure functions”, the exact analogues of the Kronecker- δ : a function K_J which is always 0, except for a *single* 1 on input $\underline{x}^{(J)}$, with $J = 0 \cdots 2^n - 1$:

$$K_J(\underline{x}) = \delta_{\underline{x}, \underline{x}^{(J)}} = \begin{cases} 1 & \text{for } \underline{x} = \underline{x}^{(J)} \\ 0 & \text{otherwise} \end{cases} . \quad (2.34)$$

Evidently, an arbitrary function f can be decomposed as a “sum over the pure (Kronecker) components”:

$$f(\underline{x}) = \sum_{J=0}^{2^n-1} a_J K_J(\underline{x}) , \quad (2.35)$$

where, in reality, the number of terms in the sum is equal to the number of elements 1 in the binary-string \underline{a} :

$$K_f = \sum_{J=0}^{2^n-1} a_J = \text{Number of non-zero elements in } f \quad \text{with} \quad 0 \leq K_f \leq 2^n . \quad (2.36)$$

Suppose that $K_f > 0$, i.e., we are not considering the function $f = 0$. Now denote by J_k , for $k = 1 \cdots K_f$, the terms where $a_{J_k} = 1$, all other a_J vanishing. Totally equivalently to writing Eq. (2.35), we can express f through K_f logical **OR** of the appropriate “Kronecker”, as follows:

$$f(\underline{x}) = K_{J_1}(\underline{x}) \vee K_{J_2}(\underline{x}) \vee \cdots \vee K_{J_{K_f}}(\underline{x}) . \quad (2.37)$$

This expression requires $K_f - 1$ logical **OR**, and K_f **COPY**, to replicate the input in each of the $K_{J_k}(\underline{x})$. To conclude, we need to see how we can calculate each pure component $K_{J_k}(\underline{x})$. This is a very simple Boolean satisfiability problem: you can write $K_{J_k}(\underline{x})$ using n **AND** for the variables x_i , possibly negated, using **NOT**. This form of the function f is known as *disjunctive normal form*: a disjunction (**OR**) of formulas written, in turn, in terms of conjunction (**AND**) of literals.

As an illustration, for $n = 5$, $K_{13}(\underline{x})$ is given by:

$$13 = (01101) \quad \implies \quad K_{13}(\underline{x}) = \bar{x}_4 \wedge x_3 \wedge x_2 \wedge \bar{x}_1 \wedge x_0 ,$$

since you are guaranteed that you will get 0 unless the bits x_i satisfy the formula encoded by K_{13} . This concludes our proof: **OR**, **AND**, **NOT** and **COPY** are enough to construct any f .

Warning: The proof has some conceptual interest, but is somewhat devoid of practical applications. It does not show at all how to construct an *efficient* algorithm to calculate f . Quite the opposite, you are supposed to know the whole *table* for f , and in general K_f can be as large as $2^n - 1$, hence an exponential complexity emerges from the construction. Imagine, to exemplify the point, the function $f_{\text{add}}(x, y) = s$ which adds two integers x and y , producing the sum s . If x has n bits, and y has $m \leq n$ bits, then s has $n + 1$ bits, hence we have $f_{\text{add}} : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^{n+1}$. Now, you do not certainly construct the *table* for f_{add} to proceed: you should know the answer, for all x and y , to do that. You rather find a smart iterative way of obtaining the bit-string for s bit-by-bit, as discussed before, with a number of operations that scales as $O(n)$, the length of the input.

Question Q2) Let us discuss now what is the *minimal* set of gates necessary to implement any digital calculation. Here things are quite simple:

1: OR) De Morgan's identities imply that **OR** can be obtained from **NOT** and **AND**.

2: NOT) Interestingly **NOT** can be obtained from **NAND** and **COPY**. If $x = 0, 1$ is a bit:

$$\text{COPY}(x) = (x, x) \rightarrow \text{NAND}(x, x) = 1 - x^2 = 1 - x = \bar{x} = \text{NOT}(x).$$

3: AND) I leave to you, as an exercise, to show that you can construct **AND** from **NAND** and **COPY**. Hence, using De Morgan, you can construct also **OR** from **NAND** and **COPY**.

i

Minimal set of universal gates. This shows that **NAND** and **COPY** constitute a minimal set of universal gates, in terms of which you can express any other gate, in particular **OR**, **AND**, and **NOT**, and therefore express any function f .

2.8. Universality vs Efficiency: Tractable vs Intractable problems

As already discussed, our proof of universality of the classical gates tells us nothing about the minimal number of gates needed to calculate f , an information in turn connected to the running TIME and memory SPACE needed in the calculation. Recall that, the number of **OR** of Kröner terms K_f being $K_f \leq 2^n$, this implies, potentially, an exponential (in n) *complexity*.

But sometimes much easier algorithms exist. For instance, as discussed, $f_{\text{add}}(x, y)$ uses a number of operations of $O(n)$, the length of the input. Another simple function to calculate is $f_{\text{mul}}(x, y) = xy$: in elementary school we learned an algorithm that requires $t_{\text{mul}} = O(n^2)$.

i

You never know ...) In 1971, Schönhage and Strassen found an algorithm to multiply two numbers based on Fast Fourier Transform (FFT) that scales as $O(n(\log n)(\log \log n))$ and Fürer in 2007 found a novel method with a slightly better asymptotic behaviour. This is just to remark that is never guaranteed that someone will not come out, in a given problem, with a much better algorithm! And since we mentioned FFT, this is another interesting case: the Discrete Fourier Transform of a set of N data has a complexity which scales, superficially, as N^2 . But then, in 1965, Cooley and Tukey (re)-discovered the FFT algorithm, which apparently also Gauss had used in some unpublished astronomical work of his. And this led to an algorithm, FFT, that scales as $N \log N$, and is nowadays recognised as one of the 10 most important algorithms of the last century.

Both addition and multiplication are very simple algorithms, with complexity that scales *polynomially* with n : improving on the power of $\text{poly}(n)$ is always possible by smarter algorithms yet to come.

i

Classically intractable problems. There are problems for which no polynomial algorithm has ever been found, and people suspect *it does not exist* (but, as we said, “you never know ...”). Such problems are known as “*classically intractable*”.

Sometimes the difficulty is associated to “inverting” a simple operation. For instance, we know that $z = f_{\text{mul}}(x, y) = xy$ is simple. But if I give you a very large integer z , and I ask you to find its factors x and y , *if they exist?* In one version of this *integer factorisation* problem, you could formulate a (simpler) *decision problem*:

1 Primality decision problem.) Give z , an n -bit integer, is z a *prime number*? More formally, you are asked to write an algorithm to calculate the following function:

$$f_{\text{primality}}(z) = \begin{cases} 1 \text{ (TRUE)} & \text{if } z \text{ is prime} \\ 0 \text{ (FALSE)} & \text{otherwise} \end{cases} .$$

With the table-idea used in the universality proof, you could think of having something like this: ⁴

		x_{n-1}	\dots	x_1	x_0	$f_{\text{primality}}$
$\underline{x}^{(0)}$	→	0	\dots	0	0	0
$\underline{x}^{(1)}$	→	0	\dots	0	1	0
$\underline{x}^{(2)}$	→	0	\dots	1	0	1
$\underline{x}^{(3)}$	→	0	\dots	1	1	1
$\underline{x}^{(4)}$	→	0	\dots	1	0	0
$\underline{x}^{(5)}$	→	0	\dots	1	1	1
\vdots					\vdots	\vdots
$\underline{x}^{(J)}$	→	\dots				?
\vdots					\vdots	\vdots
$\underline{x}^{(2^n-1)}$	→	1	\dots	1	1	?

Simple algorithms like the *Sieve of Erathosthenes* would typically require checking for division by integers up to \sqrt{z} , and with z as large as $N = 2^n$, this definitely implies an algorithm which is **super-polynomial** in n , the number of bits.

But in 2002-2004 Agrawal, Kayal & Saxena (AKS) invented an algorithm for testing primality which scales as $O(n^{12})$, later refined (also by others) to $O(n^6)$. So, the *primality decision problem* is, in the end, *polynomial* in n , the number of bits of the integer.

But what about the problem of *integer factorisation*? If an AKS primality test tells you that z is composite and not prime, you would like to go ahead and find its factors! Here the problem is much harder. You are asked to construct a function $f(z)$ which returns 1 if z is prime, and, say, the smaller of its factors if z is not prime. The best known algorithm, so far, requires:

$$t_{\text{factorisation}} \sim \exp\left(O\left(n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}\right)\right) .$$

The largest “semi-prime” (i.e., product of two primes) yet (as of Feb. 2020) factored is an 829-bit number with 250 decimal digits. ⁵ People believe that with ~ 1024 -bit integers finding factors is essentially impossible by classical computers. And this is a crucial ingredient in the public-key cryptography based on the RSA algorithm, of which we will have more to say later on.

1 Shor’s algorithm. Probably the most remarkable application of Quantum Computing to date is Shor’s algorithm for period-finding (and factorisation) which scales as $t_{\text{Shor}} \sim O(n^2(\log n)(\log \log n))$. This finding, which we will discuss in some detail, gave an enormous boost to the Quantum Computation idea.

⁴Incidentally, the ? in the primality of $2^n - 1$ is an interesting problem. Prime numbers of this kind are known as “Mersenne primes”. Sometimes $2^n - 1$ is prime, like 7 or 31, sometimes not, like 15.

⁵It required ~ 2700 core-years of computing with INTEL XEON Gold 6130 at 2.1 GHz.

2.9. Boolean Satisfiability

A typical objection to the importance of factorising integers is that, if you find an algorithm to do that, breaking therefore RSA public-key cryptography, people would simply change the cryptographic algorithm, turning to a different classically intractable problem. This of course does no justice to Shor's remarkable achievement, which should not be regarded simply as an algorithm for breaking RSA, but has profound and important potential applications in various problems.

Nevertheless, if you want to mention a classically intractable problem for the importance of which people would never raise any objection of any kind, this is probably **Boolean Satisfiability**, or SAT, for short. The importance of such a problem is multifold. In theoretical Computer Science, it was the first problem proven — **Cook's theorem** — to be **NP-complete**. On the application side, it is a typical problem for which people at Boeing or Lockheed Martin would pay you a lot, if you give them a very good algorithm. The reason is that they need efficient algorithms to test if the many pieces of their complicated machines all satisfy the appropriate safety tests, otherwise the airplane would, for instance, blow-up.

Let me explain what a SAT problem is. You consider the usual n -bit binary strings \underline{x} . You take k -bits and form a so-called *clause* $C(x_{j_1}, \dots, x_{j_k})$ depending on such k variables only, by using logical **OR** and possible **NOT** of the variables. To illustrate this, with the important case of $k = 3$, you take, for instance:

$$C(x_{j_1}, x_{j_2}, x_{j_3}) = \zeta_{j_1} \vee \zeta_{j_2} \vee \zeta_{j_3} \quad \text{with} \quad \zeta_j = x_j \text{ or } \bar{x}_j .$$

Next, you consider m such k -clauses, C_1, C_2, \dots, C_m , each depending from its k variables — possibly/usually shared by some of the clauses —, and you take the **AND** of them, forming the following Boolean formula:

$$f_{k\text{-SAT}}(\underline{x}) = C_1 \wedge C_2 \wedge \dots \wedge C_m , \quad (2.38)$$

where I have omitted indicating the variables in each clause to avoid a messy multi-index notation. This is known as *conjunctive normal form*, and the resulting Boolean formula is known as a k -SAT formula. The k -SAT decision problem consists in “deciding” if an assignment \underline{x} exists which makes the given formula TRUE. The corresponding optimisation problem is to actually *find* one or more satisfactory assignments, if they exist, or, otherwise, find one that minimises the number of clauses which are not satisfied. As you see, the logical **AND** pose conflicting requirements — or constraints — on the variables involved in the various clauses. If the number of such clauses/constraints is very small, many satisfying assignments exist, and the problem is generally easy to solve. When the ratio

$$\alpha = \frac{m}{n} ,$$

of clauses-to-bits increases, the problem starts to be harder.

i

2-SAT is easy, k -SAT with $k \geq 3$ intractable. It turns out that the problem is still polynomial for $k = 2$, but starts being classically intractable for $k \geq 3$.

Many studies have been devoted to the $k = 3$ case, particularly in the case in which the 3 variables in each clause are chosen randomly, the so-called Random 3-SAT, for which a well studied phase diagram — through spin-glass techniques of Statistical Physics — as a function of α is known. In particular, for $\alpha > \alpha_c \approx 4.267$ — the so-called UNSAT phase — no satisfying assignment is found, for large n . Within the SAT phase with $\alpha < \alpha_c$, there is a window $4.15 \approx \alpha_G < \alpha < \alpha_c \approx 4.267$ where satisfying assignment exist, with probability 1 for large n , but finding them is extremely hard. If you want to know more about these beautiful applications of Statistical Physics to Information Science, you should read the book by Mézard & Montanari, *Information, physics, and computation* [18].

Interestingly, the algorithms that people used for solving k -SAT problems, before the statistical physicists introduced quite powerful cavity-method-based algorithms, were incredibly simple. They are known as GSAT and WalkSAT, and are closely related. Here is how WalkSAT works.

1**WalkSAT.**

- 1) Choose a random initial string x . If it satisfies the formula, you are done. Otherwise go to 2).
- 2) Pick at random a clause among those that are *unsatisfied*.
- 3) Flip a variable within that clause. The way you pick the variable to be flipped is:
 1. either randomly, with some probability q ;
 2. or, with probability $1 - q$, you flip the variable that will result in the *fewest* previously satisfied clauses becoming unsatisfied.
- 4) Repeat until a solution is found or a maximum number of iterations is reached.

3. Quantum gates and elements of quantum computation

Here I will start discussing Quantum Computation (QC), including quantum gates and the most elementary quantum algorithms. As previously announced, QC is potentially richer than Classical Computation because of the possibility of forming superpositions of states, and of using *unitary operations*, a much larger set than classical operations on Cbits. To exploit such a richness I must be able to *create* such superposition states out of standard product states, and build any possible unitary in a standard way, so that one can implement it on a suitable hardware.

The starting point are two classical Cbit states $\{|0\rangle, |1\rangle\}$ — previously prematurely written with the ket-notation — which we now upgrade to the two orthogonal basis states of a single spin-1/2 Hilbert space: ¹

$$\{|0\rangle = |\uparrow\rangle, |1\rangle = |\downarrow\rangle\},$$

the so-called *computational basis* of the Qbit. The states $|\psi\rangle_1$ of the single-Qbit Hilbert space \mathcal{H}_1 are obtained as complex normalised *superpositions* of such two states:

$$|\psi\rangle_1 = z_0|0\rangle + z_1|1\rangle \quad \text{with} \quad |z_0|^2 + |z_1|^2 = 1.$$

Up to an overall phase factor in front, you can identify them with the spin states $|+, \mathbf{n}\rangle$ in Eq. (1.1), taking $z_0 = \cos(\theta/2)$ and $z_1 = e^{i\phi} \sin(\theta/2)$.

Before moving to the many-Qbit Hilbert space, let me make a few comments on the nature of such a “spin-1/2” in practical implementations. You should really not think that we can do quantum computation by working with the spin-1/2 degree-of-freedom of the electrons in a piece of matter, as we would have no control of them. So, depending on the *hardware* on which we would base our Quantum Computer, the nature of “the spin-1/2” Qbit changes. Here are a few of the proposals that have been investigated to date. We will return to these issues later on, in a series of lectures given by Prof. Rosario Fazio.

NMR) This was an early proposal, based on the vast abilities that physicists had demonstrated, since the 1950’s, in controlling nuclear spins in molecules. This amazing control has led to the remarkable success of NMR as a sophisticated medical imaging technique. Although the level of control of a few nuclear-spin/Qbits in a molecule is impressive, the technique is not *scalable*: if you need to increase the number of Qbits you need to change molecule.

Two-level atoms) Quantum optics and the control of atom-photon interactions have advanced tremendously since the middle 1980’s. The two levels which make the Qbit are simply two levels of an atom, the ground state and some excited state, which are controlled and manipulated with the use of coherent radiation. Trapping atoms in optical lattices has created a sufficiently scalable platform for the coherent manipulation of many atoms.

¹This identification is almost mandatory. Think of the mess that would come out from the opposite convention $\{|0\rangle = |\downarrow\rangle, |1\rangle = |\uparrow\rangle\}$, when I start writing the “Pauli-Z” matrix as:

$$\mathbf{Z} = \begin{pmatrix} -1 & 0 \\ 0 & +1 \end{pmatrix}.$$

Polarisation of photons) An all-optical implementation of the Qbit exploits the two polarisation states of photons. Beam-splitter, mirrors, polarisers, non-linear crystals generating entangled photons and all these machinery allow a great flexibility, which enjoys also from the fact that photons suffer very little from interactions with external agents that could lead to loss of coherence. See, for instance, the [Xanadu](#) website.

Trapped ions) Similar to the coherent control of atoms, but with ions, for instance $^{40}\text{Ca}^+$, which are then trapped with radio-frequency traps ([Paul traps](#)) and manipulated with coherent radiation. Blatt's group experimental expertise at U. of Innsbruck has lead — through their spin-off [Alpine Quantum Technology](#) — to a commercial general purpose Quantum Computer based on trapped ions.

Rydberg atoms) Atoms that can stay in long-lived highly excited Rydberg states are natural candidates for two-level atoms. This is also a very promising platform, especially for *Quantum Simulators*.

Superconducting Qbits) This is the platform which has more overlap with traditional solid-state systems. It exploits the fact that with small superconducting Josephson junctions one can create tiny loops — like in a [SQUID](#) — where current can circulate in two opposite directions, and the two states can be coherently manipulated. In some sense, this is a mesoscopic, rather than microscopic, implementation of a Qbit, but the ability to maintain coherence in such systems is remarkable.

3.1. Computational states and superpositions: the Hilbert space

Recall that, for n Cbits, the space of the 2^n classical configurations was denoted as:

$$|x\rangle_n = |x_{n-1}\rangle|x_{n-2}\rangle \cdots |x_1\rangle|x_0\rangle \equiv |x_{n-1}x_{n-2} \cdots x_1x_0\rangle = |\underline{x}\rangle_n \quad (3.1)$$

where:

$$x = \sum_{j=0}^{n-1} x_j 2^j \quad \text{with} \quad x_j = 0, 1. \quad (3.2)$$

As repeatedly mentioned we will often identify $0 \leq x \leq 2^n - 1$ with their corresponding binary string $x \mapsto \underline{x} = (x_{n-1}x_{n-2} \cdots x_1x_0) \in \{0, 1\}^n$.

①

The n -Qbit Hilbert space. This is, essentially, the basis set of the n -Qbit Hilbert space

$$\mathcal{H}_n = \mathcal{H}_1 \otimes \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_1 \stackrel{\text{def}}{=} \mathcal{H}_1^{\otimes n},$$

obtained as *tensor* product of the single-Qbit Hilbert space \mathcal{H}_1 . The most general state of \mathcal{H}_n will be a complex normalised superposition of computational basis (product) states:

$$|\psi\rangle_n = \sum_{x_{n-1}=0}^1 \cdots \sum_{x_0=0}^1 \psi_{x_{n-1} \cdots x_1 x_0} |x_{n-1} \cdots x_1 x_0\rangle = \sum_{x=0}^{2^n-1} \psi_x |x\rangle_n \quad (3.3)$$

with the normalisation condition reading:

$$\sum_{x=0}^{2^n-1} |\psi_x|^2 = 1. \quad (3.4)$$

This allows to identify quantum states with elements of \mathbb{C}^{2^n-1} , the wave-function amplitudes $\psi_{\underline{x}} = \psi_x$, the “-1” discounting from an overall normalisation and global phase. In some sense, the Quantum Computation scheme is a “simplified” finite-dimensional version of a general QM setting. You can proceed by just studying complex linear algebra with a few extra rules (Quantum Measurements), which is indeed Mermin’s approach in his superbe account of the subject for computer scientists [1].



Notation. A word on the many equivalent notations we will use. Sometimes, we will associate an explicit number to the various kets appearing in $|x\rangle_n$. For instance:

$$|01 \cdots 100\rangle = |0\rangle_{n-1} |1\rangle_{n-2} \cdots |1\rangle_2 |0\rangle_1 |0\rangle_0 = |0\rangle_{n-1} \otimes |1\rangle_{n-2} \otimes \cdots \otimes |1\rangle_2 \otimes |0\rangle_1 \otimes |0\rangle_0$$

is a product state (the \otimes will be often omitted) in which the Qbit-0 is in state $|0\rangle$, Qbit-1 in state $|0\rangle$, Qbit-2 in state $|1\rangle$, etc. Notice that we typically count the n Qbits from 0 to $n-1$, and we order them from right to left. This notation has sometimes a potential clash with $|\psi\rangle_n$, which might indicate a generic n -Qbit state, but notice the **roman** n . For instance, you should remark the difference between $|0 \cdots 0\rangle_n \equiv |0\rangle_n$, a n -Qbit product state with all Qbits in state $|0\rangle$, from the single-Qbit state $|0\rangle_m$, where the m -th Qbit is in state $|0\rangle$.

3.2. Unitary operators associated to function evaluation

Let us recall the result we got in Sec. 2.4. For any $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, you can extend it in such a way that it is invertible by introducing m ancillary bits \underline{y} , one for each output variable, defining:

$$\tilde{f} : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^{n+m} \quad \text{with} \quad \tilde{f}(\underline{x}, \underline{y}) = (\underline{x}, \underline{y} \oplus f(\underline{x})), \quad (3.5)$$

where \oplus on the RHS indicates an m -bitwise \oplus . As you recall, such \tilde{f} is self-inverse and gives the function $f(\underline{x})$ when the ancillary bits are all set to 0, $\tilde{f}(\underline{x}, \underline{0}) = (\underline{x}, f(\underline{x}))$.

The extension of such an invertible function \tilde{f} to a unitary operator \mathbf{U}_f on $\mathcal{H}_n \otimes \mathcal{H}_m$ is very simple. Define \mathbf{U}_f on the computational basis vectors in the obvious way:

$$\mathbf{U}_f(|\underline{x}\rangle_n \otimes |\underline{y}\rangle_m) = |\underline{x}\rangle_n \otimes |\underline{y} \oplus f(\underline{x})\rangle_m. \quad (3.6)$$

Next, define the linear extension of such \mathbf{U}_f to any state $|\psi\rangle_n \in \mathcal{H}_n$ as follows:

$$\begin{aligned} \mathbf{U}_f(|\psi\rangle_n \otimes |\underline{y}\rangle_m) &= \mathbf{U}_f\left(\sum_{x=0}^{2^n-1} \psi_x |\underline{x}\rangle_n \otimes |\underline{y}\rangle_m\right) = \sum_{x=0}^{2^n-1} \psi_x \mathbf{U}_f(|\underline{x}\rangle_n \otimes |\underline{y}\rangle_m) \\ &= \sum_{x=0}^{2^n-1} \psi_x |\underline{x}\rangle_n \otimes |\underline{y} \oplus f(\underline{x})\rangle_m. \end{aligned} \quad (3.7)$$



Unitarity of the extension. It is very simple to prove that this linear extension defines a *unitary* operator. The reason for that is simple:

- 1) The 2^{n+m} basis states $|\underline{x}\rangle_n \otimes |\underline{y}\rangle_m$ are *orthogonal*.
- 2) The map \tilde{f} being invertible, it maps distinct inputs into distinct output elements, which are still orthogonal computational basis elements, just permuted.
- 3) Any transformation that maps an orthonormal basis into another orthonormal basis is *unitary*, as you recall from QM.

Observe that we have not, so far, taken linear combinations of the ancillary bits $|y\rangle_m$. Sometimes, later on, it will prove convenient to do so. For the time being observe that when operating on $|y\rangle_m = |0\rangle_m$ we get:

$$\mathbf{U}_f\left(|\psi\rangle_n \otimes |0\rangle_m\right) = \sum_{x=0}^{2^n-1} \psi_x |\underline{x}\rangle_n \otimes |f(\underline{x})\rangle_m, \quad (3.8)$$

which has a very deceptive *quantum parallelism* appearance: in one application of \mathbf{U}_f to a superposition input state, we get a superposition of all possible output states where $|\underline{x}\rangle_n \otimes |f(\underline{x})\rangle_m$ appears for *all* \underline{x} .

The reason why such quantum parallelism is *totally useless* has to do with the nature of the quantum states and to the rules of Quantum Measurements. Recall that $|\psi\rangle_n$ encodes all the statistical information on the measurements that I might decide to perform on the quantum state, but *I am not allowed* to think that I can read *all coefficients* of such a state in a single shot. Quite the opposite, to learn $|\psi\rangle_n$ I have to do measurements. Measurements in the computational basis (essentially, measurements of commuting products of \mathbf{Z} -operators for each of the n spins), according to von Neumann, will result in getting, each time, a collapse into *one of the eigenstates* $|\underline{x}\rangle_n \otimes |f(\underline{x})\rangle_m$, with a probability $\text{Prob}(\underline{x}|\psi) = |\psi_{\underline{x}}|^2$. Hence, not only I get, upon measurement, only *one component* of the final superposition, but I am not able to predict which one will come out! The intrinsic randomness of the Quantum Measurement makes the parallelism, as such, utterly useless.

❶

Smart transformations needed. As we learned in the introduction — recall the role of the output beam-splitter in the Mach-Zehnder interferometer — the randomness intrinsic in QM is very special, unlike classical randomness. If I am able to devise some transformation — effectively changing the basis of states on which I measure — by applying suitable gates to the input and/or output Qbits, in such a way that the final superposition is transformed into a *certain outcome*, or one in which I can read the answer easily, then the quantum parallelism turns into an amazingly effective tool. This is the goal of any smart QC algorithm.

Having done the general theory, let us turn to one instructive example. Consider the Boolean functions $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ which we have already examined in Chap. 2, and which we repeat here in slightly more systematic fashion. As you know, there are $2^{2^2} = 16$ of them. Here they are:

x_1	x_0	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	\dots
0	0	0	0	0	0	0	0	0	0	1	\dots
0	1	0	0	0	0	1	1	1	1	0	\dots
1	0	0	0	1	1	0	0	1	1	0	\dots
1	1	0	1	0	1	0	1	0	1	0	\dots

We stopped at f_8 because $f_{15-i} = \mathbf{NOT}(f_i)$. You recognise most of them: $f_0 = \mathbf{E}_{\text{rase}}$, $f_1 = f_{\wedge} = \mathbf{AND}$, $f_3 = x_1$, $f_5 = x_0$, $f_6 = \mathbf{XOR}$, $f_7 = \mathbf{OR}$, $f_8 = \mathbf{NOR}$, etc. Let us consider the **AND**, and write explicitly its reversible extension. Since $f_{\wedge}(x_1, x_0) = x_1 \wedge x_0 = x_1 x_0$ we conclude that:

❶

Toffoli gate.

$$\tilde{f}_{\wedge}(x_1, x_0, y) = (x_1, x_0, y \oplus f_{\wedge}(x_1, x_0)) = (x_1, x_0, y \oplus x_1 x_0). \quad (3.9)$$

Such a *doubly-controlled* 3-bit reversible classical gate — the reversible extension of the **AND** — is known as *Toffoli gate*. It is of some relevance in classical computation, where you can prove that it belongs to the minimal set of universal gates for reversible classical computation: 2-bit gates are not enough. ^a

^aAs you would learn from studying Sec. 3.15.3 a Toffoli gate can be re-expressed in QC in terms of 6 cNOTs plus single-Qbit unitaries. But this is not possible in CC, i.e., by using only classical gates.

To get a more standard ordering of the bits, we redefine $x_1 \rightarrow x_2$, $x_0 \rightarrow x_1$, $y \rightarrow x_0$, and write:

$$\tilde{f}_\wedge(x_2, x_1, x_0) = (x_2, x_1, x_0 \oplus x_2x_1).$$

The corresponding unitary operator is defined on the basis as:

$$\mathbf{U}_{f_\wedge}(|x_2\rangle|x_1\rangle|x_0\rangle) = |x_2\rangle|x_1\rangle|x_0 \oplus x_2x_1\rangle,$$

and then linearly (and unitarily) extended. Recall that the \oplus does the same action as a **NOT**, when $x_2x_1 = 1$, otherwise you get $x_0 \oplus 0 = x_0$, hence the identity.

We will use a notation for $\mathbf{U}_{f_\wedge} = \mathbf{C}_{21,0}$ which will be later useful when constructing other control-gates in QC. It is the generalisation of the \mathbf{C}_{10} , where now the two Qbits before the comma, 2 and 1, are the two control-Qbits, and 0, after the comma, is the target-Qbit. One might even write an explicit expression in terms of projectors and \mathbf{X} , as follows:

$$\mathbf{C}_{21,0} = (\mathbf{N}_0)_2(\mathbf{N}_0)_1\mathbf{1}_0 + (\mathbf{N}_0)_2(\mathbf{N}_1)_1\mathbf{1}_0 + (\mathbf{N}_1)_2(\mathbf{N}_0)_1\mathbf{1}_0 + (\mathbf{N}_1)_2(\mathbf{N}_1)_1\mathbf{X}_0.$$

Even after expressing the projectors in terms of \mathbf{Z} , we still have a gate in which operators have to be applied to *three Qbits*, something that is not ideal in hardware implementations. We will later on explore, see Sec. 3.15.3, how to express Toffoli by using 6 (cNOT) \mathbf{C}_{ij} 2-Qbit gates applied appropriately one after the other. This will be just a demonstration of a general principle:

i **Universal quantum gates.** Any unitary can be decomposed in terms of single-Qbit unitaries — Hadamards and rotations around the z-axis are enough — and \mathbf{C}_{ij} gates, which are therefore universal gates for QC.

I will not prove this theorem, see [2] or [19]. More about this later on, when we will investigate quantum gates more systematically.

3.3. Pauli operators and associated single-Qbit unitary gates

Let us review some basic properties of the Pauli operators, and the associated spin-rotation (unitary) matrices. Here I count on your previous QM studies.

As you know, the 3 Pauli matrices form the basis for the traceless Hermitian 2×2 matrices, and, supplemented by the identity, the 4 of them form a basis of all Hermitian 2×2 matrices. Hence I can write the most general Hermitean 2×2 matrix as:

$$a_0\mathbf{1} + \mathbf{a} \cdot \hat{\boldsymbol{\sigma}} \quad \text{with} \quad a_0 \in \mathbb{R}, \mathbf{a} \in \mathbb{R}^3. \quad (3.10)$$

As a consequence, I can write the most general *unitary* 2×2 matrix by exponentiating this, with an imaginary i :

$$\mathbf{U} = e^{-i(a_0\mathbf{1} + \mathbf{a} \cdot \hat{\boldsymbol{\sigma}})} = e^{-ia_0} e^{-i\mathbf{a} \cdot \hat{\boldsymbol{\sigma}}}.$$

Now we express the vector \mathbf{a} as $\mathbf{a} = |\mathbf{a}|\mathbf{n}$ where $\mathbf{n} = \mathbf{a}/|\mathbf{a}|$ is the associated *versor*. Hence:

$$\begin{aligned} \mathbf{U} = e^{-ia_0} e^{-i|\mathbf{a}|\mathbf{n} \cdot \hat{\boldsymbol{\sigma}}} &= e^{-ia_0} \left(\mathbf{1} \cos |\mathbf{a}| - i(\mathbf{n} \cdot \hat{\boldsymbol{\sigma}}) \sin |\mathbf{a}| \right) \\ &= e^{-ia_0} \left(\mathbf{1} \cos \theta - i(\mathbf{n} \cdot \hat{\boldsymbol{\sigma}}) \sin \theta \right) \\ &= e^{-ia_0} \underbrace{\left(\mathbf{1} \cos \frac{\gamma}{2} - i(\mathbf{n} \cdot \hat{\boldsymbol{\sigma}}) \sin \frac{\gamma}{2} \right)}_{\stackrel{\text{def}}{=} \mathbf{U}_\mathbf{n}(\gamma)}, \end{aligned} \quad (3.11)$$

where we expanded the exponential, and used the property that $(\mathbf{n} \cdot \hat{\sigma})^2 = \mathbf{1}$ when $|\mathbf{n}| = 1$, hence $(\mathbf{n} \cdot \hat{\sigma})^{2k+1} = (\mathbf{n} \cdot \hat{\sigma})$ and $(\mathbf{n} \cdot \hat{\sigma})^{2k} = \mathbf{1}$. In the second expression, we used the fact that $|\mathbf{a}|$ enters periodic trigonometric functions so that we can restrict $|\mathbf{a}| = \theta \in [0, 2\pi)$ without loss of generality. The third and final expression, where we posed $2\theta = \gamma$, hence with $\gamma \in [0, 4\pi)$, is probably very well-known to you from spin-physics: it represents a *rotation in spin-space* along direction \mathbf{n} by an angle γ , which will denote by $\mathbf{U}_{\mathbf{n}}(\gamma)$ from now on. And, as you remember, there is this curious fact of spin-1/2 that a rotation by $\gamma = 4\pi$ is needed to get the identity. Get to work yourself, now, to discover some of the particular unitaries that you can write by choosing different \mathbf{n} and θ .

Exercise 3.1. (Pauli and Hadamard as rotations by π in spin-space.)

Obviously, for $\theta = 0$, you have that $\mathbf{U} = e^{-ia_0}\mathbf{1}$, a diagonal phase factor. Show that by eating-up factors of i with the choice of $a_0 = -\frac{\pi}{2}$, and setting $\theta = \frac{\pi}{2}$, or $\gamma = \pi$, you can get

$$\mathbf{n} = (1, 0, 0) \rightarrow \mathbf{U} = \mathbf{X} \quad \mathbf{n} = (0, 1, 0) \rightarrow \mathbf{U} = \mathbf{Y} \quad \mathbf{n} = (0, 0, 1) \rightarrow \mathbf{U} = \mathbf{Z}, \quad (3.12)$$

and

$$\mathbf{n} = \left(\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right) \rightarrow \mathbf{U} = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) \stackrel{\text{def}}{=} \mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (3.13)$$

The last matrix obtained, the so-called *Hadamard* \mathbf{H} , is so useful that deserves a section by itself, see Sec. 3.4. Remember that all of these matrices represent rotations by $\gamma = \pi$ in spin space, in the direction associated to \mathbf{n} , with an extra factor $e^{-ia_0} = i$. Since $\mathbf{X}^2 = \mathbf{Y}^2 = \mathbf{Z}^2 = \mathbf{1}$, you immediately deduce that not only the Pauli matrices are unitary (as well as Hermitean), but they coincide with their inverse:

$$\mathbf{X}^{-1} = \mathbf{X}^\dagger = \mathbf{X} \quad \mathbf{Y}^{-1} = \mathbf{Y}^\dagger = \mathbf{Y} \quad \mathbf{Z}^{-1} = \mathbf{Z}^\dagger = \mathbf{Z}. \quad (3.14)$$

By using the fact that $\mathbf{XZ} = -\mathbf{ZX}$ — or, equivalently, that $\mathbf{ZXZ} = -\mathbf{X}$, which fits well with the π -rotation story — show that:

$$\mathbf{H}^2 = \mathbf{1} \quad \implies \quad \mathbf{H}^{-1} = \mathbf{H}^\dagger = \mathbf{H}. \quad (3.15)$$

Show also that, as a nice rotation by π around $\mathbf{n} = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$, \mathbf{H} rotates \mathbf{Z} into \mathbf{X} and viceversa:

$$\mathbf{HZH} = \mathbf{X} \quad \text{and} \quad \mathbf{HXH} = \mathbf{Z}. \quad (3.16)$$



Warning: Don't be surprised that I told you that “the Pauli matrices represent rotations by π ” in the direction associated to the Cartesian versor \mathbf{n} , and that, for instance, $\mathbf{X}^2 = \mathbf{1}$, while you know that you need a $\gamma = 4\pi$ rotation, and not 2π , to get the identity! Indeed, it is the factor i coming from e^{-ia_0} that fixes everything.

Exercise 3.2. (Rotations around the y-axis.)

There is another very interesting particular case of \mathbf{U} that you can find. Take $\mathbf{n} = (0, 1, 0)$ and $a_0 = 0$, but now leave $\theta = \gamma/2$ arbitrary, to get:

$$\mathbf{U} = e^{-ia_0} e^{-i\theta\hat{\sigma}^y} = \mathbf{1} \cos \theta - i\hat{\sigma}^y \sin \theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \stackrel{\text{def}}{=} \mathbf{R}_\theta \equiv \mathbf{U}_y(2\theta), \quad (3.17)$$

a very familiar rotation matrix \mathbf{R}_θ , which is now also regarded as a rotation by an angle $\gamma = 2\theta$ around the y -direction in spin-space, $\mathbf{U}_y(2\theta)$. More about this later on.

3.4. The Hadamard gate \mathbf{H}

This section highlights two basic sides of the Hadamard transformation, both immensely useful in Quantum Computation. The first is that it creates very simple superposition states, aligned along the x-direction in spin space, starting from the computational states $|0\rangle = |\uparrow\rangle$ and $|1\rangle = |\downarrow\rangle$, which can be regarded as spin states in the z-direction. The second is that it serves as a generator of powerful identities between different gates.

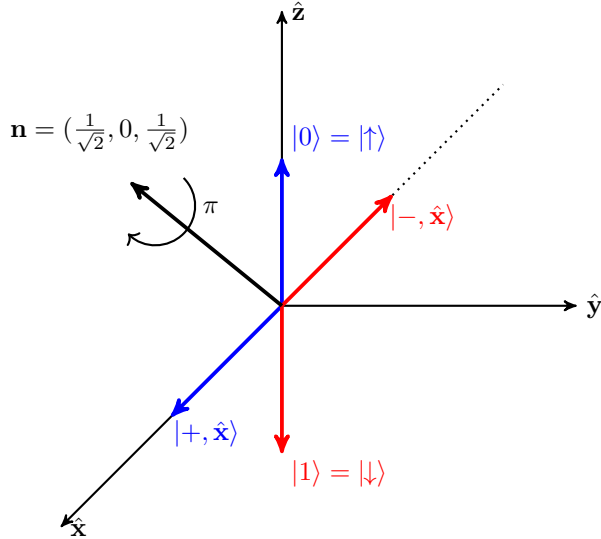


Figure 3.1: The Hadamard transformation \mathbf{H} regarded as a rotation by π around the axis $\mathbf{n} = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$, which transforms \mathbf{Z} eigenstates into \mathbf{X} eigenstates, and viceversa. As we will later discuss, this is the Bloch sphere picture for spin-1/2 states.

Let us start from the rotation of computational states, a kind of anticipation of the future QC story. Observe that, while, as you know $|0\rangle \mapsto |\uparrow\rangle$, by applying \mathbf{H} you get:

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \mapsto |+, \mathbf{x}\rangle. \quad (3.18)$$

Similarly, while $|1\rangle \mapsto |\downarrow\rangle$, by applying \mathbf{H} you get:

$$\mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \mapsto |-, \mathbf{x}\rangle. \quad (3.19)$$

This “rotation by π ” interpretation, illustrated in Fig. 3.1, leads to the very useful relations, that we have already found, which I repeat here:

$$\mathbf{H}\mathbf{Z}\mathbf{H} = \mathbf{X} \quad \text{and} \quad \mathbf{H}\mathbf{X}\mathbf{H} = \mathbf{Z}. \quad (3.20)$$

We will use them shortly to deduce important identities between gates.

But before doing that, let me remark that there is an alternative way to picture the rotation of \mathbf{Z} eigenstates into \mathbf{X} eigenstates, which is slightly more comfortable for our classical intuition of orthogonality and elementary rotations. This is shown in Fig. 3.2. Notice the nice aspect of it. The two orthogonal states $|0\rangle = |\uparrow\rangle$ and $|1\rangle = |\downarrow\rangle$ are now shown as orthogonal, and the same occurs for the $|\pm, \mathbf{x}\rangle$ states. And the rotation that brings one to the other is $\mathbf{R}_{\theta=\frac{\pi}{4}}$, which can be equivalently regarded as a rotation by $\frac{\pi}{2}$ around the y-axis in spin space, $\mathbf{U}_y(\gamma = 2\theta = \frac{\pi}{2})$. Notice also how we immediately appreciate that you need a $\gamma = 4\pi$ rotation in spin-space to return to the original state, since a $\gamma = 2\pi$ corresponds to a $\mathbf{R}_{\theta=\pi}$ rotation, which leads to a change of sign. The only drawback of this picture, is that it works for x-z spin-states only, the real combinations, and y-states are missing. On the contrary, the Bloch-sphere more traditional viewpoint of Fig. 3.1 allows y-states to be drawn, but orthogonal spin-states are shown as antipodean points on the sphere. Choose what you like most, but get used, in principle, to both.

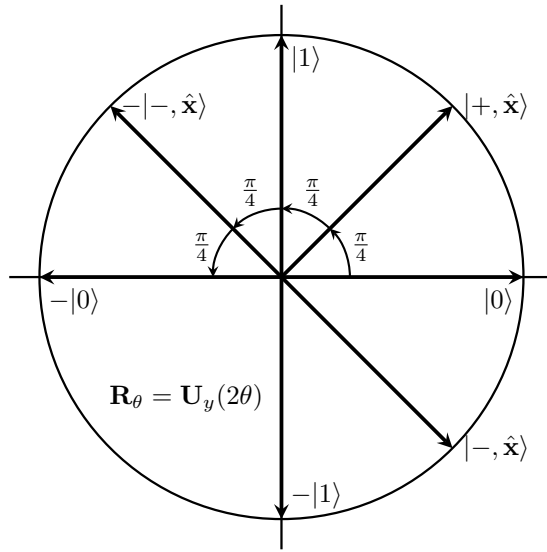


Figure 3.2: A representation of spin-states in the x - and z -direction that shows, more conventionally, the orthogonal nature of the two eigenstates of \mathbf{Z} and \mathbf{X} , as well as the fact that a $\theta = \pi$ rotation \mathbf{R}_π , corresponding to a $\gamma = 2\pi$ rotation in spin space, leads to a change of sign.

Now we return to \mathbf{C}_{10} , which I report below for convenience, adding an $\mathbf{1}_0$ for clarity:

$$\mathbf{C}_{10} = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_1 \mathbf{1}_0 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_1 \mathbf{X}_0 . \quad (3.21)$$

Application of \mathbf{H}_0 on both sides. Let us start applying \mathbf{H}_0 on both sides. Since $\mathbf{H}_0^2 = \mathbf{1}_0$ the first term is unchanged. But the second term, since $\mathbf{H}_0 \mathbf{X}_0 \mathbf{H}_0 = \mathbf{Z}_0$, leads to:

$$\begin{aligned} \mathbf{H}_0 \mathbf{C}_{10} \mathbf{H}_0 &= \frac{1}{2}(\mathbf{1} + \mathbf{Z})_1 \mathbf{1}_0 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_1 \mathbf{H}_0 \mathbf{X}_0 \mathbf{H}_0 \\ &= \frac{1}{2}(\mathbf{1} + \mathbf{Z})_1 \mathbf{1}_0 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_1 \mathbf{Z}_0 \stackrel{\text{def}}{=} \mathbf{C}_{10}^{\mathbf{Z}} . \end{aligned} \quad (3.22)$$

This is *not* a gate of relevance for Classical Computation, but will be useful later on: it is a control- \mathbf{Z} gate. We will encounter later a whole family of control-unitary, or control- \mathbf{U} , operators: we leave it for the time being, except for observing that the control- \mathbf{Z} gate is symmetric in the two-bit indices, since:

$$\mathbf{C}_{10}^{\mathbf{Z}} = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_1 \mathbf{1}_0 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_1 \mathbf{Z}_0 = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_0 \mathbf{1}_1 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_0 \mathbf{Z}_1 \stackrel{\text{def}}{=} \mathbf{C}_{01}^{\mathbf{Z}} ,$$

as you can immediately verify by arranging terms.

Application of $\mathbf{H}_1 \mathbf{H}_0$ on both sides. Let us insist on applying \mathbf{H} , this time $\mathbf{H}_1 \mathbf{H}_0$ on both sides. We get:

$$\begin{aligned} \mathbf{H}_1 \mathbf{H}_0 \mathbf{C}_{10} \mathbf{H}_1 \mathbf{H}_0 &= \frac{1}{2} \mathbf{H}_1 (\mathbf{1} + \mathbf{Z})_1 \mathbf{H}_1 \mathbf{1}_0 + \frac{1}{2} \mathbf{H}_1 (\mathbf{1} - \mathbf{Z})_1 \mathbf{H}_1 \mathbf{H}_0 \mathbf{X}_0 \mathbf{H}_0 \\ &= \frac{1}{2} (\mathbf{1} + \mathbf{X})_1 \mathbf{1}_0 + \frac{1}{2} (\mathbf{1} - \mathbf{X})_1 \mathbf{Z}_0 \\ &= \frac{1}{2} (\mathbf{1} + \mathbf{Z})_0 \mathbf{1}_1 + \frac{1}{2} (\mathbf{1} - \mathbf{Z})_0 \mathbf{X}_1 \stackrel{\text{def}}{=} \mathbf{C}_{01} , \end{aligned} \quad (3.23)$$

i.e., a cNOT with the role of the bits exchanged: bit-0 is now the control, bit-1 the target. Entirely similar algebra for general ij shows that:

$$\mathbf{H}_i \mathbf{H}_j \mathbf{C}_{ij} \mathbf{H}_i \mathbf{H}_j = \mathbf{C}_{ji} . \quad (3.24)$$

This is very useful.

i **Order of commuting operators is irrelevant.** Notice as, on several occasions, the site-index notation has freed us from putting operators in the conventional order as appropriate for a tensor product. As you know, for instance, $\mathbf{Z}_0\mathbf{X}_1$ stands, as a matrix in the 4-dim space, for $\mathbf{X}_1 \otimes \mathbf{Z}_0$, and no possible confusion arises.

3.4.1. Using only Hadamard and rotations around the z-axis

Now we would like to do something similar to what you do when you express ordinary rotations in three-dimensions in terms of Euler angles and rotations about particular axes. More precisely, we will use only rotations around the z-axis and Hadamard transformations.

Notice that:

$$\mathbf{U}_{\mathbf{z}}(\gamma) = \mathbf{1} \cos \frac{\gamma}{2} - i\mathbf{Z} \sin \frac{\gamma}{2} = \begin{pmatrix} e^{-i\frac{\gamma}{2}} & 0 \\ 0 & e^{i\frac{\gamma}{2}} \end{pmatrix}.$$

It is convenient to multiply by an overall phase factor $e^{i\frac{\gamma}{2}}$ and define the standard rotation matrix around the z-axis as follows:

i **The phase gate.**

$$\mathbf{R}_{\mathbf{z}}(\gamma) = e^{i\frac{\gamma}{2}} \mathbf{U}_{\mathbf{z}}(\gamma) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\gamma} \end{pmatrix}, \quad (3.25)$$

which leaves $|0\rangle$ unchanged, and adds a phase γ to $|1\rangle$.

We have already discussed rotations around the y-axis in Sec. 3.3: we will not repeat the discussion here. We recall instead the all-important Hadamard transformation, a rotation by π around $\mathbf{n}_{\mathbf{H}} = \frac{1}{\sqrt{2}}(\mathbf{x} + \mathbf{z})$.

i **The Hadamard.**

$$\mathbf{H} = e^{i\frac{\pi}{2}} \mathbf{U}_{\mathbf{n}_{\mathbf{H}}}(\pi) = \mathbf{n}_{\mathbf{H}} \cdot \hat{\boldsymbol{\sigma}} = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (3.26)$$

Observe that, since $\mathbf{HZH} = \mathbf{X}$ you immediately deduce that:

$$\mathbf{HR}_{\mathbf{z}}(\theta)\mathbf{H} = \mathbf{R}_{\mathbf{x}}(\theta) = e^{i\frac{\theta}{2}} \mathbf{U}_{\mathbf{x}}(\theta) = e^{i\frac{\theta}{2}} (\mathbf{1} \cos \frac{\theta}{2} - i\mathbf{X} \sin \frac{\theta}{2}) = e^{i\frac{\theta}{2}} \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}. \quad (3.27)$$

Next we consider a phase-gate with angle $\frac{\pi}{2} + \phi$:

$$\mathbf{R}_{\mathbf{z}}(\frac{\pi}{2} + \phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i(\frac{\pi}{2} + \phi)} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & ie^{i\phi} \end{pmatrix}. \quad (3.28)$$

Applied successively:

$$\mathbf{R}_{\mathbf{z}}(\frac{\pi}{2} + \phi)\mathbf{HR}_{\mathbf{z}}(\theta)\mathbf{H} = e^{i\frac{\theta}{2}} \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & ie^{i\phi} \cos \frac{\theta}{2} \end{pmatrix}. \quad (3.29)$$

This implies:

1 Rotating spin-states. You obtain spin eigenstates in arbitrary direction \mathbf{n} by starting from the computational basis and applying \mathbf{R}_z and \mathbf{H} :

$$\begin{cases} \mathbf{R}_z(\frac{\pi}{2} + \phi)\mathbf{H}\mathbf{R}_z(\theta)\mathbf{H}|0\rangle = e^{i\frac{\theta}{2}} \left(\cos \frac{\theta}{2}|0\rangle + e^{i\phi} \sin \frac{\theta}{2}|1\rangle \right) & = e^{i\frac{\theta}{2}}|+, \mathbf{n}\rangle \\ \mathbf{R}_z(\frac{\pi}{2} + \phi)\mathbf{H}\mathbf{R}_z(\theta)\mathbf{H}|1\rangle = ie^{i\frac{\theta}{2}} \left(-\sin \frac{\theta}{2}|0\rangle + e^{i\phi} \cos \frac{\theta}{2}|1\rangle \right) & = ie^{i\frac{\theta}{2}}|-, \mathbf{n}\rangle \end{cases} \quad (3.30)$$

4 Why not 4 parameters? You might be worried that we used just *two* parameters in the previous story. The most general unitary obviously involves *four* parameters (see also below). One is an overall phase e^{-ia_0} , that we might omit indicating below. The other one is, in principle, an extra phase that you can put in the two orthogonal states $|\pm, \mathbf{n}\rangle$, by considering

$$\mathbf{U} = \mathbf{R}_z(\frac{\pi}{2} + \phi)\mathbf{H}\mathbf{R}_z(\theta)\mathbf{H}\mathbf{R}_z(-\frac{\pi}{2} + \lambda)$$

which sends:

$$\begin{cases} |0\rangle \rightarrow \mathbf{U}|0\rangle = e^{i\frac{\theta}{2}} \left(\cos \frac{\theta}{2}|0\rangle + e^{i\phi} \sin \frac{\theta}{2}|1\rangle \right) & = e^{i\frac{\theta}{2}}|+, \mathbf{n}\rangle \\ |1\rangle \rightarrow \mathbf{U}|1\rangle = e^{i(\frac{\theta}{2} + \lambda)} \left(-\sin \frac{\theta}{2}|0\rangle + e^{i\phi} \cos \frac{\theta}{2}|1\rangle \right) & = e^{i(\frac{\theta}{2} + \lambda)}|-, \mathbf{n}\rangle \end{cases} \quad (3.31)$$

Exercise 3.3. Show that if \mathbf{n}_1 and \mathbf{n}_2 are two directions parameterised by spherical angles (θ_1, ϕ_1) and (θ_2, ϕ_2) , then the unitary operator that rotates $|+, \mathbf{n}_1\rangle$ into $|+, \mathbf{n}_2\rangle$, is given by:

$$|+, \mathbf{n}_2\rangle = \mathbf{U}|+, \mathbf{n}_1\rangle \quad \text{with} \quad \mathbf{U} = \mathbf{R}_z(\frac{\pi}{2} + \phi_2)\mathbf{H}\mathbf{R}_z(\theta_2 - \theta_1)\mathbf{H}\mathbf{R}_z(-\frac{\pi}{2} - \phi_1). \quad (3.32)$$

This concludes our effort concerning single-Qbit unitaries, showing that an arbitrary rotation of spin-states can be expressed in terms of phase gates $\mathbf{R}_z(\gamma)$ and Hadamard \mathbf{H} . Before ending, we mention a few useful gates, all related to the phase-gate.

1 $\frac{\pi}{4}$ and $\frac{\pi}{2}$ phase gates. The **T**-gate, or $\frac{\pi}{4}$ -gate, is defined by:

$$\mathbf{T} = \mathbf{R}_z(\frac{\pi}{4}) = e^{i\frac{\pi}{8}}\mathbf{U}_z(\frac{\pi}{4}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \quad \Longrightarrow \quad \mathbf{T}^4 = \mathbf{Z}. \quad (3.33)$$

The **S**-gate, or $\frac{\pi}{2}$ -gate, is defined by:

$$\mathbf{S} = \mathbf{R}_z(\frac{\pi}{2}) = \mathbf{T}^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad \Longrightarrow \quad \mathbf{S} = \sqrt{\mathbf{Z}}. \quad (3.34)$$

3.5. Drawing quantum circuits

Before moving on, let us review \mathbf{C}_{10} again, and introduce the *circuit notation* to draw gates in QC.

The circuit notation is explained in Fig. 3.4 where you should notice the *left-to-right convention*, while the standard convention in writing the corresponding equations — as well as the standard for linear algebra — is right-to-left.

GATE	CIRCUIT REPRESENTATION	MATRIX REPRESENTATION	TRUTH TABLE	BLOCH SPHERE	GATE	CIRCUIT REPRESENTATION	MATRIX REPRESENTATION	TRUTH TABLE	BLOCH SPHERE
I Identity-gate: no rotation is performed.		$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & 0\rangle \\ 1\rangle & 1\rangle \end{matrix}$		S gate: rotates the qubit state by $\frac{\pi}{2}$ radians (90°) about the z-axis.		$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & 0\rangle \\ 1\rangle & e^{i\pi/2} 1\rangle \end{matrix}$	
X gate: rotates the qubit state by π radians (180°) about the x-axis.		$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & 1\rangle \\ 1\rangle & 0\rangle \end{matrix}$		T gate: rotates the qubit state by $\frac{\pi}{4}$ radians (45°) about the z-axis.		$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & 0\rangle \\ 1\rangle & e^{i\pi/4} 1\rangle \end{matrix}$	
Y gate: rotates the qubit state by π radians (180°) about the y-axis.		$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & i 1\rangle \\ 1\rangle & -i 0\rangle \end{matrix}$		H gate: rotates the qubit state by π radians (180°) about an axis diagonal in the x-z plane. This is equivalent to an X-gate followed by a $\frac{\pi}{2}$ rotation about the y-axis.		$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & \frac{1}{\sqrt{2}}(0\rangle + 1\rangle) \\ 1\rangle & \frac{1}{\sqrt{2}}(0\rangle - 1\rangle) \end{matrix}$	
Z gate: rotates the qubit state by π radians (180°) about the z-axis.		$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Input Output $\begin{matrix} 0\rangle & 0\rangle \\ 1\rangle & - 1\rangle \end{matrix}$						

Figure 3.3.: Illustration of single QBit quantum gates. Figure taken from Ref. [17][Fig. 9].

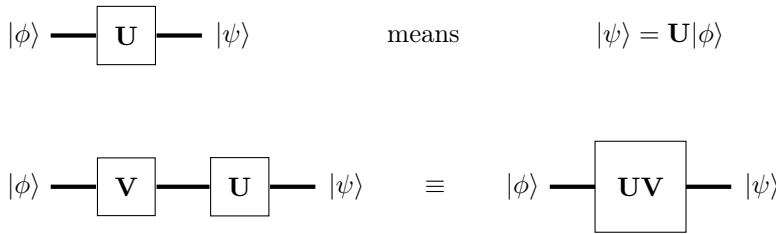


Figure 3.4: Top: The left-to-right convention in drawing a unitary U applied to a state $|\phi\rangle$, resulting in a state $|\psi\rangle = U|\phi\rangle$. Bottom: When joining unitaries into a single square, you have to reverse their order.

The C_{10} and its variants. Figure 3.5 shows two possible ways of drawing the C_{10} gate acting on 2-Qbit computational basis states $|x_1x_0\rangle$. The top drawing highlights the connection to the logical XOR

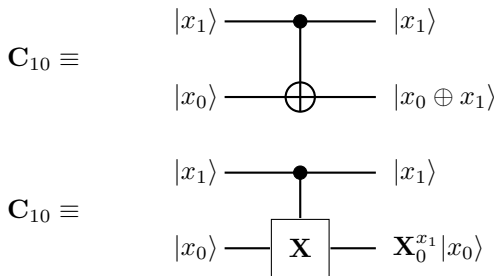


Figure 3.5: Two alternative drawing conventions for the cNOT gate C_{10} . Top: This highlights the arithmetic nature related to the XOR gate, with the target sent to $|x_0 \oplus x_1\rangle$. Bottom: This highlights the control-NOT nature, where $X_0^{x_1}$ means 1_0 for $x_1 = 0$ and X_0 for $x_1 = 1$. The ordering convention of Qbits is from 0 to 1, bottom-to-top. The solid circle indicated that Qbit-1 is the control Qbit. The linear (unitary) extension to arbitrary 2-Qbits states is the usual one.

and arithmetic \oplus , while the bottom drawing highlights the connection to a control-X, or control-NOT. The solid circle on Qbit-line 1 means that Qbit-1 is the control-Qbit, which is unchanged.

Observe that there is a closely connected variant of C_{10} which acts non-trivially on the target when the control-bit is $|0\rangle$ rather than $|1\rangle$. We will denote as $C_{\bar{1}0}$ and reads:

$$C_{\bar{1}0} = (N_1)_1 \otimes 1_0 + (N_0)_1 \otimes X_0 = (N_1)_1 + (N_0)_1 X_0 . \tag{3.35}$$

Figure 3.6 shows a circuit representation of $C_{\bar{1}0}$, where the empty circle indicates the reversed control. Since X is such that $XXZ = -Z$, hence $XN_0X = N_1$ and $XN_1X = N_0$, you can immediately conclude this two variants are related as follows:

$$C_{\bar{1}0} = X_1 C_{10} X_1 , \tag{3.36}$$

which is the second form shown in Fig. 3.6.

Recall, finally, that you can exchange control and target using Hadamards:

$$H_1 H_0 C_{10} H_1 H_0 = C_{01} . \tag{3.37}$$

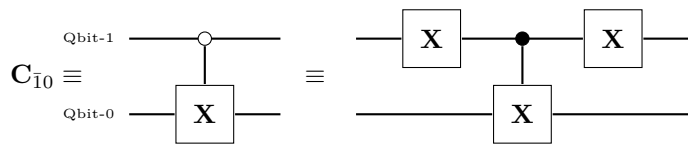


Figure 3.6: The C_{10} variant of C_{10} . The control now acts non-trivially only when the control-Qbit 1 is in state $|0\rangle$, as denoted by the empty circle. On the right we used Eq. (3.36).

This identity is illustrated in Fig. 3.7, together with a second form which follows from observing that $HXH = Z$.

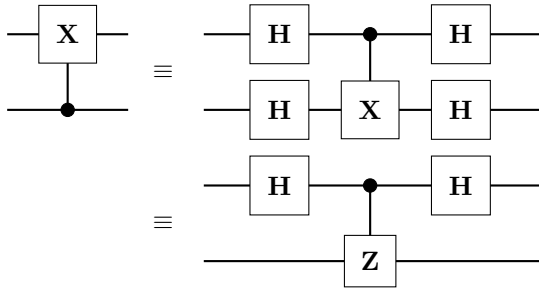


Figure 3.7: The identity in Eq. (3.37), illustrating how to exchange control- and target-Qbit by a sandwich with H on both lines. The second form (below) comes from observing that $HXH = Z$.

Finally, to practice with multiply-controlled gates, we show the Toffoli gate in Fig. 3.8.

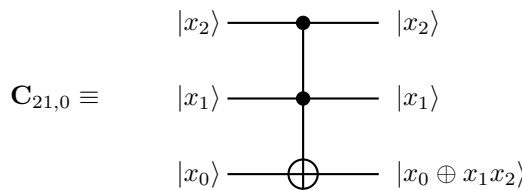


Figure 3.8: The Toffoli gate, the 3-bit reversible extension of the 2-bit logical **AND**. Qbits 1 and 2 are control-Qbits, Qbit 0 is the target. Here the action on the computational basis is shown. The linear (unitary) extension is defined as usual.

3.6. Two-Qbit states and gates

When you start considering 2-Qbit states, you realise immediately the role of entanglement. The most general product state of 2-Qbit might be written as

$$|\psi_{\text{prod}}\rangle = U_1 \otimes U_0 |00\rangle \longleftrightarrow |+, \mathbf{n}_1\rangle \otimes |+, \mathbf{n}_0\rangle,$$

requiring essentially $2 \times 2 = 4$ real parameters to be specified, up to a non-interesting global phase. On the contrary, the most general normalised superposition state for two Qbits depends on $2 \times 2^2 - 1 - 1 = 6$ real parameters, again up to a global phase. The situation becomes exponentially amplified for general n .

1 Exponential growth of entanglement complexity. A separable n -Qbit state depends only on $2n$ \mathbb{R} -parameters, or n \mathbb{C} -parameters, while a general (normalised) superposition state in the 2^n -dimensional Hilbert space of n -Qbits has $(2^n - 1)$ \mathbb{C} -parameters, up to an overall phase.

Returning to our $n = 2$ problem, I can presumably make reference to your knowledge of 2-site spin states, with their total spin eigenstates $|\psi_{S,M}\rangle$, triplet $S = 1$ and singlet $S = 0$. Rewritten in our

language they would be the following 4 states:

$$\begin{cases} |\psi_{1,+1}\rangle = |\uparrow\rangle_1|\uparrow\rangle_2 = |00\rangle \\ |\psi_{1,0}\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle_1|\downarrow\rangle_0 + |\downarrow\rangle_1|\uparrow\rangle_0) = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\psi_{1,-1}\rangle = |\downarrow\rangle_1|\downarrow\rangle_2 = |11\rangle \\ |\psi_{0,0}\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle_1|\downarrow\rangle_0 - |\downarrow\rangle_1|\uparrow\rangle_0) = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{cases} . \quad (3.38)$$

Notice that the two states with $M = \pm 1$ are indeed *product states*. If you do not care much about working with proper spin eigenstates, you can work with appropriate *entangled* combinations of them, as we will:

$$\begin{cases} |\psi_{1,+}\rangle = \frac{1}{\sqrt{2}}(|\psi_{1,+1}\rangle + |\psi_{1,-1}\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\psi_{1,-}\rangle = \frac{1}{\sqrt{2}}(|\psi_{1,+1}\rangle - |\psi_{1,-1}\rangle) = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \end{cases} . \quad (3.39)$$

These 4 entangled combinations form the so-called *Bell basis*: you could write any states of $\mathcal{H}_1 \otimes \mathcal{H}_2$ in terms of them, instead of working with the 4 computational basis states $\{|x_1x_0\rangle\}$.

i

The Bell basis.

$$\begin{cases} |\psi_{1,+}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\beta_{00}\rangle \\ |\psi_{1,0}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = |\beta_{01}\rangle \\ |\psi_{1,-}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\beta_{10}\rangle \\ |\psi_{0,0}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = |\beta_{11}\rangle \end{cases} , \quad (3.40)$$

where the RHS introduced an alternative notation — the β reminding us of “Bell” — explained below.

The notation $|\beta_{x_1x_0}\rangle$ will be now explained. Let us start from $|\beta_{00}\rangle$. It is simple to verify that you obtain it as:

$$\mathbf{C}_{10}\mathbf{H}_1|00\rangle = \mathbf{C}_{10}\frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1)|0\rangle_0 = \frac{1}{\sqrt{2}}((|0\rangle_1|0\rangle_0 + |1\rangle_1|1\rangle_0) = |\beta_{00}\rangle ,$$

which clearly demonstrates the power of \mathbf{C}_{10} in creating entanglement when working on superposition states of the control Qbit, in turn created by \mathbf{H}_1 .

Exercise 3.4. Show that:

$$|\beta_{x_1x_0}\rangle = \mathbf{C}_{10}\mathbf{H}_1|x_1x_0\rangle . \quad (3.41)$$

By using that $|x_1x_0\rangle = \mathbf{X}_1^{x_1}\mathbf{X}_0^{x_0}|00\rangle$, that $\mathbf{H}_1\mathbf{X}_1 = \mathbf{Z}_1\mathbf{H}_1$, and both \mathbf{Z}_1 and \mathbf{X}_1 commute with \mathbf{C}_{10} show that you can also rewrite the previous expression as:

$$|\beta_{x_1x_0}\rangle = \mathbf{C}_{10}\mathbf{H}_1|x_1x_0\rangle = \mathbf{C}_{10}\mathbf{H}_1\mathbf{X}_1^{x_1}\mathbf{X}_0^{x_0}|00\rangle = \mathbf{Z}_1^{x_1}\mathbf{X}_0^{x_0}\mathbf{C}_{10}\mathbf{H}_1|00\rangle . \quad (3.42)$$

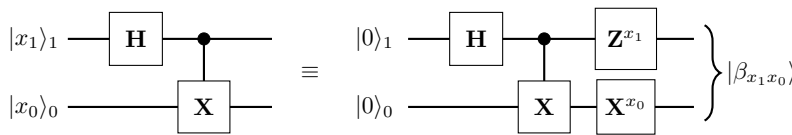


Figure 3.9: The circuit behind Eq. (3.42), generating the four Bell’s states.

Interestingly, this is the general structure of 2-Qbit states, as you will learn in Sec. 3.15.1. More precisely, you can show that you can generally represent any 2-Qbit state as:

$$|\Psi\rangle = z_{00}|00\rangle + z_{01}|01\rangle + z_{10}|10\rangle + z_{11}|11\rangle = \mathbf{U}_1\mathbf{V}_0\mathbf{C}_{10}\mathbf{W}_1|00\rangle , \quad (3.43)$$

where \mathbf{U}_1 , \mathbf{V}_0 and \mathbf{W}_1 are suitable unitaries operating on the respective Qbits. This shows that to construct an arbitrary *entangled* 2-Qbit state you can use just 1-Qbit unitaries — hence in the end \mathbf{R}_z and \mathbf{H} , as see previously — together with the cNOT 2-Qbit gate \mathbf{C}_{10} , which therefore starts playing a leading role in the story.

3.6.1. Bell measurements

The first form of Eq. (3.41) is very useful to answer to the following question. Suppose I give you a Bell state $|\Psi_{\text{Bell}}\rangle$ but I do not tell you which one. How would you recognise it?

◆ **No point in doing measurements right away.** As we have already stressed, and will repeatedly do, there is no point in trying to “learn” a quantum state by doing measurements! You would just get (randomly) collapsed components with certain probabilities. What you always want to do is to find an appropriate *transformation* to apply *before* measurement, in such a way that the outcome becomes *certain*. Recall the Mach-Zehnder interferometer example.

Here the unitary transformation you need to do before measurement is suggested by Eq. (3.41).

Exercise 3.5. [Inverting the Bell’s construction.] After proving that $\mathbf{C}_{10}^2 = \mathbf{1}$, and using that $\mathbf{H}_1^2 = \mathbf{1}_1$, show that:

$$|x_1 x_0\rangle = \mathbf{H}_1 \mathbf{C}_{10} |\beta_{x_1 x_0}\rangle. \quad (3.44)$$

Eq. (3.44) tells us that *after* having applied $\mathbf{H}_1 \mathbf{C}_{10}$ you would obtain a single product state, and not a superposition. Hence, a measurement in the computational basis — measuring commuting $\hat{\sigma}^z$ on the two Qbits — would immediately tell you which Bell’s state you were given, since you know x_1 and x_0 .

● **Bell’s measurement.** This mechanism is known as *Bell measurement* and is the basic building block of the **Quantum Teleportation** protocol, which will be discussed in Sec. 3.14.

At this point I want to pose for a while, to satisfy a legitimate curiosity.

Question:

How would we implement such 1-Qbit unitaries — for instance \mathbf{R}_z and \mathbf{H} — and the 2-Qbit cNOT \mathbf{C}_{10} in the actual hardware of a Quantum Computer?

The question is a crucial one, especially in the light of the fact which we will later show: Not only you can represent 2-Qbit states and transformations using 1-Qbit unitaries and the cNOT \mathbf{C}_{10} , but an arbitrary n-Qbit unitary transformation can always be decomposed in terms of such ingredients, which act therefore as *universal gates*.

The answer to such a question really *depends* on the hardware on which you implement a QC. But for any hardware, is essentially one of the first questions that those who build the QC ask themselves. I will here provide — for the purpose of a mere illustration of principle — a demonstration of how such a question might be answer in an NMR setting, where you act and control true spin-1/2 objects which respond to static and time-dependent magnetic fields.

3.7. NMR-like Hamiltonian model for 1- and 2-Qbit gates

Consider a single spin-1/2 in a *circularly polarised* in-plane magnetic field $\mathbf{B}_\perp(t)$, together with a longitudinal (usually larger) B_z -field. This is a standard set-up when discussing NMR. We take, for a single spin-1/2, the Hamiltonian to be: ²

$$\hat{H} = -\mu B_z \hat{\sigma}^z - \mu B_\perp (\hat{\sigma}^x \cos(\omega t) + \hat{\sigma}^y \sin(\omega t)) . \quad (3.45)$$

This model allows for an *exact solution* when you move to a *rotating frame*. A similar model with a *linearly polarised* in the x -direction, $\mathbf{B}_\perp = (B_\perp, 0, 0) \cos(\omega t)$, would allow only approximate solutions.

Exercise 3.6. [Exact solution in a circularly polarised field.] By making a transformation to a rotating frame moving with the in-plane field, hence transforming the problem into a time-independent one (but be careful to the fact that the transformation being time-dependent, you have to account for the appropriate derivative in the Schrödinger equation), show that the spin state evolves as $|\psi(t)\rangle = \hat{U}(t)|\psi(0)\rangle$ with a unitary evolution operator

$$\hat{U}(t) = e^{-i\frac{\omega_0 t}{2} \hat{\sigma}^z} e^{-i\frac{(\omega_0 - \omega)t}{2} \hat{\sigma}^z} e^{-i\frac{\mu B_\perp t}{\hbar} \hat{\sigma}^x} , \quad (3.46)$$

and

$$\omega_0 = \frac{2\mu B_z}{\hbar}$$

is the Zeeman splitting frequency in the B_z longitudinal field.

The *resonance condition* $\omega = \omega_0$ simplifies the result to:

$$\hat{U}_{\text{res}}(t) = e^{-i\frac{\omega_0 t}{2} \hat{\sigma}^z} e^{-i\frac{\Omega_{\text{Rabi}} t}{2} \hat{\sigma}^x} \quad \text{where} \quad \Omega_{\text{Rabi}} = \frac{2\mu B_\perp}{\hbar} , \quad (3.47)$$

is the so-called *Rabi frequency*, which you control by the in-plane magnetic field strength.

Exercise 3.7. [Parameter choice for X and H.] Determine what choice of parameters you need to make for the operator $\hat{U}(t)$, acting for a time $t = \tau$, to implement:

- 1) The **X** (or **NOT**) unitary.
- 2) The **H** Hadamard.

[Hint: while **X** can be done at resonance, **H** requires having $\omega \neq \omega_0$...]

To construct the cNOT \mathbf{C}_{10} we need two *interacting* spins. We label the target spin by 0, and the control by 1, as usual. At this point we might proceed in two ways.

First route. We write a 2-Qbit Hamiltonian with an Ising interaction, of the form:

$$\hat{H} = -\mu_0 B_0 \hat{\sigma}_0^z - \mu_1 B_1 \hat{\sigma}_1^z + J \hat{\sigma}_0^z \hat{\sigma}_1^z - \mu_0 B_\perp (\hat{\sigma}_0^x \cos(\omega t) + \hat{\sigma}_0^y \sin(\omega t)) , \quad (3.48)$$

where, notice, the in-plane field is applied only to the target Qbit-0. Observing that there is no term that could change the spin of the control Qbit-1, you can simplify the problem by working with two different single-spin Hamiltonians for Qbit-0. For $\hat{\sigma}_1^z \rightarrow \pm 1$ we have:

$$\hat{H}_\pm = \mp \mu_1 B_1 \mathbf{1}_0 - (\mu_0 B_0 \mp J) \hat{\sigma}_0^z - \mu_0 B_\perp (\hat{\sigma}_0^x \cos(\omega t) + \hat{\sigma}_0^y \sin(\omega t)) . \quad (3.49)$$

²For electrons $\mu = -\mu_B$, the Bohr magneton. Nuclei have a positive μ which is however, due to their large mass, much smaller than a μ_B .

Python exercise 3.1. Now you observe that \hat{H}_{\pm} have different longitudinal fields ($\mu_0 B_0 \mp J$), hence different resonance conditions: you can make the resonance effective in turning the Qbit-0 when Qbit-1 is in state $|1\rangle$ (i.e., for \hat{H}_-) and not for the Qbit-1 in state $|0\rangle$. Analyse this numerically, for instance with a python notebook.

Alternative route. The alternative is to notice that $\mathbf{C}_{10} = \mathbf{H}_0 \mathbf{C}_{10}^Z \mathbf{H}_0$, where the symmetric \mathbf{C}_{10}^Z control-gate reads:

$$\mathbf{C}_{10}^Z = \frac{1}{2}(\mathbf{1} + \mathbf{Z}_1 + \mathbf{Z}_0 - \mathbf{Z}_1 \mathbf{Z}_0) . \quad (3.50)$$

So, up to Hadamards acting on the target Qbit-0, you can concentrate your effort in building a Hamiltonian implementing \mathbf{C}_{10}^Z . Notice that this operator is *diagonal*, i.e., it provokes no transitions: it simply brings controlled-phase changes to the state.

One useful thing to notice is that $(\mathbf{C}_{10}^Z)^2 = \mathbf{1}$, hence, with algebra totally identical to that used for the exponential of Pauli matrices, you can show that:

$$e^{i\mathbf{C}_{10}^Z \theta} = \cos \theta + i\mathbf{C}_{10}^Z \sin \theta . \quad (3.51)$$

From this you deduce that:

$$\mathbf{C}_{10}^Z = e^{-i\frac{\pi}{4}} e^{i\frac{\pi}{4}(\mathbf{Z}_1 + \mathbf{Z}_0 - \mathbf{Z}_1 \mathbf{Z}_0)} . \quad (3.52)$$

It is now clear that a *time-independent* Hamiltonian of the form:

$$\hat{H} = -\mu_0 B_0 \hat{\sigma}_0^z - \mu_1 B_1 \hat{\sigma}_1^z + J \hat{\sigma}_0^z \hat{\sigma}_1^z ,$$

will do the job, if acting for a controlled time t .

If you want to avoid Ising interactions, because they are hard to implement, you can also proceed with fully-rotational-invariant Heisenberg interactions: read Appendix H of Mermin's book [1] to learn how to do that.

3.8. A variety of 2-Qbit and multi-Qbit unitary gates.

Let us resume our journey into 2-Qbit unitary gates. We have already discussed \mathbf{C}_{10} and $\mathbf{C}_{\bar{1}0}$ and the interchanged control versions \mathbf{C}_{01} and $\mathbf{C}_{\bar{0}1}$ where the target is Qbit-1 (the second element) and the control is Qbit-0 in its $|1\rangle_0$ state or in its $|0\rangle_0$ (the $\mathbf{C}_{\bar{0}1}$ -version). One can also show that the SWAP (symmetric) gate can be written as:

$$\mathbf{S}_{10} = \mathbf{S}_{01} = \mathbf{C}_{10} \mathbf{C}_{01} \mathbf{C}_{10} . \quad (3.53)$$

Perhaps more intuitive is the fact that a SWAP transforms \mathbf{C}_{10} into \mathbf{C}_{01} and viceversa:

$$\mathbf{C}_{01} = \mathbf{S}_{10} \mathbf{C}_{10} \mathbf{S}_{10} . \quad (3.54)$$

Exercise 3.8. Prove Eq. (3.53) and (3.54). [Hint: An operator way of proving Eq. (3.53) is to write the control in terms of projectors \mathbf{N}_0 and \mathbf{N}_1 , substitute, and do the algebra.]

Control-phase gates. We have seen already \mathbf{C}_{10}^Z . One can generalise it to an arbitrary phase γ by defining:

$$\mathbf{C}_{10}^\gamma = (\mathbf{N}_0)_1 \otimes \mathbf{1}_0 + (\mathbf{N}_1)_1 \otimes (\mathbf{R}_z(\gamma))_0 \quad (3.55)$$

where the phase-gate appears:

$$\mathbf{R}_z(\gamma) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\gamma} \end{pmatrix}.$$

Similar definitions can be given for \mathbf{C}_{10}^γ , \mathbf{C}_{01}^γ , and \mathbf{C}_{01}^γ .

i

\mathbf{C}_{10}^π **control-phase gate.** Observe that:

$$\mathbf{C}_{10}^\pi = \mathbf{C}_{10}^{\mathbf{Z}}. \quad (3.56)$$

This operator is sometimes denoted as $\mathbf{C}_{10}^{\text{MINUS}}$ and could be much simpler to implement in the hardware than \mathbf{C}_{10} . As mentioned previously, it is connected to \mathbf{C}_{10} by a pair of \mathbf{H}_0 :

$$\mathbf{C}_{10} = \mathbf{H}_0 \mathbf{C}_{10}^\pi \mathbf{H}_0. \quad (3.57)$$

General control-U gates. Given a general 1-Qbit unitary \mathbf{U} , you can define the obvious generalisation of \mathbf{C}_{10} and its variants as follows:

$$\mathbf{C}_{10}^{\mathbf{U}} = (\mathbf{N}_0)_1 \otimes \mathbf{1}_0 + (\mathbf{N}_1)_1 \otimes \mathbf{U}_0 \quad (3.58)$$

For $\mathbf{U} = \mathbf{X}$ we recover the \mathbf{C}_{10} — the *default* control- \mathbf{U} —, while for $\mathbf{U} = \mathbf{Z}$ we have the \mathbf{C}_{10}^π .

Exercise 3.9. Show that $\mathbf{C}_{10}^{\mathbf{U}}$ can be written in terms of \mathbf{C}_{10} and 1-Qbit unitaries.

Finally, observe the useful fact that, in the computation basis you can write:

$$\mathbf{C}_{10}^{\mathbf{U}} |x_1 x_0\rangle = \mathbf{U}_0^{x_1} |x_1 x_0\rangle, \quad (3.59)$$

with the usual understanding that $\mathbf{U}_0^0 = \mathbf{1}_0$.

3.8.1. Multi-Qbit unitary gates

When the number of Qbits $n > 2$, it turns out to be important to be able to implement operations on a target-Qbit that depend on two or more control-Qbits. Multiply-controlled unitary-gates can be defined in similar ways, but are increasing hard to construct directly. An example is given by the Toffoli gate, the doubly-controlled-**NOT**, the reversible extension of the logical **AND** we have already encountered: read Sec. 3.15.3 to learn how to write it in terms of 6 cNOTs.

Other examples will be seen later on, for instance when discussing the Grover algorithm. For instance, Fig. 3.10 illustrate the multi-controlled \mathbf{Z} gate which is useful in the context of Grover's searching for the construction of the crucial operator \mathbf{K} , the universal “kinetic energy”. These multi-controlled gates, with the addition of ancillary gates, and by paying a poly(n) number of gates, can be transformed into circuits with at most *doubly-controlled* gates.

Multiply-controlled unitary gates acting on a single target Qbit are in principle defined in a very similar way. For instance, $\mathbf{C}_{6320,4}^{\mathbf{U}}$ would be a gate acting with \mathbf{U} on target-Qbit 4, controlled by Qbit-6,3,0 (in the standard way), and by Qbit-2 in the reversed way. On the computational basis it would act as:

$$\mathbf{C}_{6320,4}^{\mathbf{U}} |x_{n-1} \cdots x_6, x_5, x_4, x_3, x_2, x_1, x_0\rangle = \mathbf{U}_4^{x_6 x_3 \bar{x}_2 x_0} |x_{n-1} \cdots x_6, x_5, x_4, x_3, x_2, x_1, x_0\rangle. \quad (3.60)$$

I could also express it in terms of projectors \mathbf{N}_0 and \mathbf{N}_1 , but the real question is how to “simplify” it, reducing it, with tricks similar to those used for the Toffoli and for the other multiply-controlled gates, eventually to single-Qbit unitaries and cNOT \mathbf{C}_{ij} gates. This can be done, in principle.

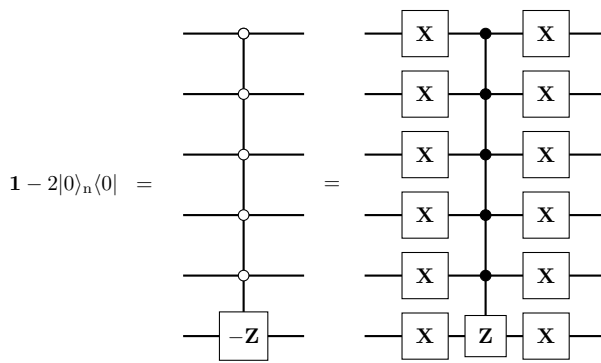


Figure 3.10: The circuit for $1 - 2|0\rangle_n\langle 0|$. When the controls are all in $|0\rangle$, the lower $-Z_0$ changes the sign of state $|0\rangle_0$; otherwise, it acts like an identity. The form on the right, with addition of X gates to the left and right of each control bit transforms the controls to the standard case (action when the control is $|1\rangle$), and Z into $-Z$, since $XZX = -Z$.

3.9. Universal quantum gates

Recall that, for classical computation of a Boolean function, **NAND** and **COPY** are universal gates. Recall also that the issue of *efficiency* has nothing to do with that of *universality*.

For the 2-Qbit case we already showed that C_{10} and 1-Qbit unitaries — hence H and $R_z(\gamma)$, ultimately — are sufficient to express any 2-Qbit unitary, and construct any 2-Qbit state. The same result holds for the n-Qbit case.

1 **Universality of cNOT and 1-Qbit unitaries.** Any n-Qbit unitary transformation U can be reduced to 1-Qbit unitaries and C_{ij} cNOT-gates, in a number that, in principle, is of order $O(n^2 4^n)$.

Details of this construction are given for instance in Ref. [2][Sec. 3.7], or in Ref. [19]. Notice, however, that the exponential number of gates involved in this construction is not good news. Efficient algorithms will have to use smart tricks to employ only a poly(n) number of gates.

1 **Which unitaries can be efficiently computed?** A fundamental and open problem of QC is to identify special classes of n-Qbit unitaries U that can be implemented, in the circuit model, using a poly(n) number of elementary gates.

3.10. Examples of function evaluation with a QC

This section simply illustrates, through a few additional examples taken from Ref. [2], the general theory presented in Sec. 3.2. We return to the $n = 2$ bit functions $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ which we already looked at, which we rewrite here for convenience:

x_1	x_0	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	\dots
0	0	0	0	0	0	0	0	0	0	1	\dots
0	1	0	0	0	0	1	1	1	1	0	\dots
1	0	0	0	1	1	0	0	1	1	0	\dots
1	1	0	1	0	1	0	1	0	1	0	\dots

where, as you recall, $f_0 = E_{\text{rase}}$, $f_1 = f_\wedge = \mathbf{AND}$, $f_3 = x_1$, $f_5 = x_0$, $f_6 = \mathbf{XOR}$, $f_7 = \mathbf{OR}$, $f_8 = \mathbf{NOR}$, etc. We already considered the **AND**, and wrote explicitly its reversible extension,

$$\tilde{f}_\wedge(x_1, x_0, y) = (x_1, x_0, y \oplus f_\wedge(x_1, x_0)) = (x_1, x_0, y \oplus x_1 x_0)$$

corresponding to the *Toffoli gate*, whose action on the computational basis, ordered as $|x_1\rangle_2|x_0\rangle_1|y\rangle_0$, reads:

$$C_{21,0}|x_1, x_0, y\rangle = X_0^{x_1x_0}|x_1, x_0, y\rangle. \tag{3.61}$$

Exercise 3.10. 1) Show that the circuit for $f_2 = x_1 \wedge \bar{x}_0$, with the usual shift in the Qbit numbering such that the ancilla y is Qbit-0, is $C_{2\bar{1},0}$ and that:

$$C_{2\bar{1},0}|x_1x_0\rangle \otimes |0\rangle = |x_1x_0\rangle \otimes |x_1 \wedge \bar{x}_0\rangle.$$

2) Similarly show that $f_4 \rightarrow C_{2\bar{1},0}$ and $f_8 \rightarrow C_{2\bar{1},0}$.

Reversible extension of the XOR. Let us consider the case of the **XOR**: $f_6 = x_0 \oplus x_1$. To encode \tilde{f}_6

$$\tilde{f}_6(x_1, x_0, y = 0) = (x_1, x_0, x_0 \oplus x_1),$$

we need to proceed in two steps:

Step 1) We put x_0 in Qbit-0 by a C_{10} operating with $|y = 0\rangle_0$ as a target (ancilla) input, which acts like a **COPY** of bit-1 into bit-0. ³ At this point we have the ancilla in $|x_0\rangle_0$.

Step 2) We next operate with $C_{2,0}$, which now does the \oplus (which you recall is essentially the **X** or **NOT**).

Figure 3.11 illustrates the circuit, showing its action in the computational basis: as you see, we have to use *two* cNOT gates here.

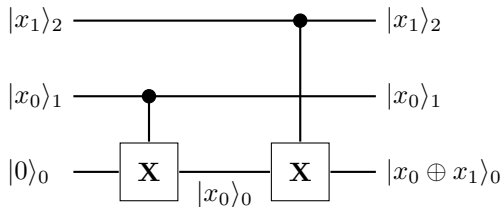


Figure 3.11: The quantum circuit to represent the reversible extension of $f_6 = \mathbf{XOR}$, with its action on the computational basis. The ancilla y is here Qbit-0, initially set to $|0\rangle_0$.

Exercise 3.11. Draw the circuit for $f_3 = x_1$, $f_5 = x_0$ and $f_0 = 0$.

An example of a f with 3 arguments. To get more feelings about how many quantum gates a function evaluation might require, consider the $f : \{0, 1\}^3 \rightarrow \{0, 1\}$ so defined:

x_2	x_1	x_0	f	
0	0	0	0	
0	0	1	1	$\leftarrow \underline{x}^{(1)} = (0, 0, 1)$
0	1	0	0	
0	1	1	0	
1	0	0	1	$\leftarrow \underline{x}^{(4)} = (1, 0, 0)$
1	0	1	1	$\leftarrow \underline{x}^{(5)} = (1, 0, 1)$
1	1	0	0	
1	1	1	0	

³This is not a violation of the famous *no-cloning theorem*, as we will discuss later on in Sec. 3.11.

Here, on the RHS we have highlighted the only input strings on which the function is 1. If you recall the discussion we have in Sec. 2.7, this preludes to the fact that we decompose f into its Krönercker components.

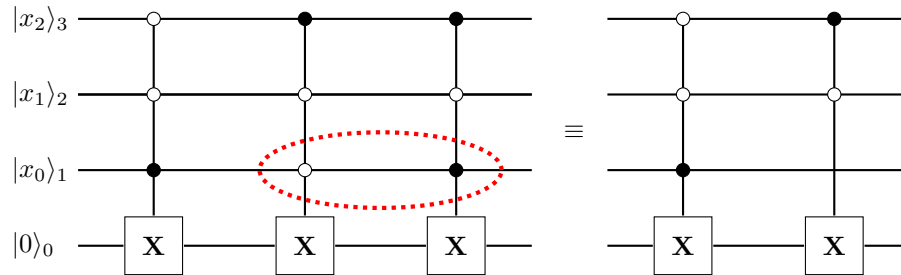


Figure 3.12.: The quantum circuit to represent the function f in Eq. (3.62). The ancilla y is here Qbit-0, initially set to $|0\rangle_0$. Notice, however, that the names of the variables have not been shifted here. The dotted ellipse on the left shows the nodes that simplify because $x_0 + \bar{x}_0 = 1$.

1 The Krönercker components. The basic idea is that $C_{3\bar{2}1,0}$ will take care of giving you $f = 1$ on the input $\underline{x}^{(1)}$, and only on that input! Similarly, $C_{3\bar{2}\bar{1},0}$ does the job on $\underline{x}^{(4)}$ and $C_{3\bar{2}1,0}$ on $\underline{x}^{(5)}$. You realise that, in general, such a circuit coding of a function f involves an exponentially large number of fully-controlled cNOTs, as Eq. (2.36) for K_f , the number of Krönercker components in the truth table of an f , immediately tells you.

In the present case, some of the fully-controlled gates simplify, as we highlight with the ellipse in the left part of Fig. 3.12. Indeed, when the same control line, here that of Qbit-1, has the control in both possibilities (the first acts for $x_0 = 0$, the second for $x_0 = 1$), all the other controls being the same, than, since $x_0 + \bar{x}_0 = 1$ you can actually simplify the circuit by eliminating the two controls from line 1, leaving in the end the circuit on the right, involving only two doubly-controlled cNOTs,

1 Designing optimised circuits. The design of optimised circuits is a basic problem of computation. Simplification rules are given in Lee *et al.* (1999).

A final example: evaluating $f(x) = x^2$. As a final example consider evaluating $f(x) = x^2$ for $x = 0, 1, 2, 3$ (a 2-bit integer). The output x^2 is in $[0, 9]$, hence requires $m = 4$ bits (and ancillas, for a reversible circuit). More generally, for an input $x \in [0, N - 1]$ we need $n = \lceil \log_2 N \rceil + 1$ bits, while $x^2 \in [0, (N - 1)^2]$ requires $m = \lceil \log_2 (N - 1)^2 \rceil$ bits, hence the reversible circuit would have $n + m$ bits: $\tilde{f} : \{0, 1\}^{n+m} \rightarrow \{0, 1\}^{n+m}$. Here is the table for f :

x_1	x_0	x	x^2	f
0	0	0	0	0000
0	1	1	1	0001
1	0	2	4	0100
1	1	3	9	1001

We have $m = 4$ ancillas, y_0, y_1, y_2, y_3 , initially set to 0, each taking care of the corresponding bit in the output. With the Krönercker components trick, acting in parallel for each of the output bits, we would have the circuit shown in Fig. 3.13, where the color-codes suggest which ancilla is taking care of which bit.

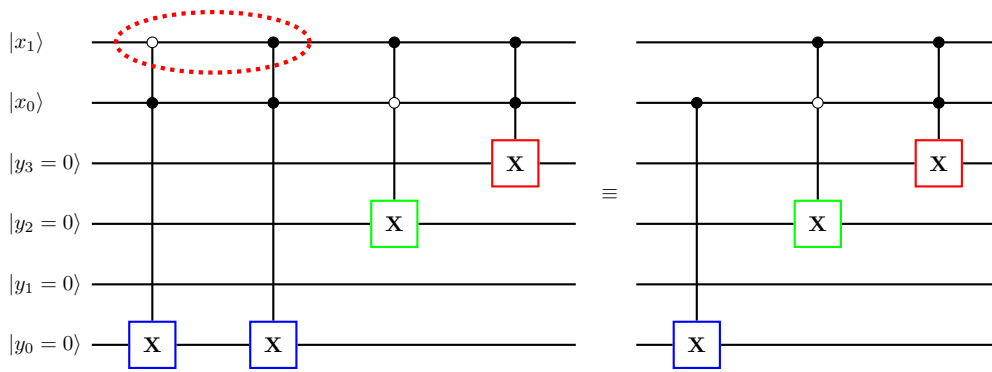


Figure 3.13.: The circuit for a reversible computation of $f : \{0, 1\}^2 \rightarrow \{0, 1\}^4$ where $f(x) = x^2$. Notice the simplification alluded at by the dotted ellipse.

◆ **Warning:** In general, evaluating a function can require many gates, and the goal is to find cases where, with smart superposition tricks, we can use much less gates than the corresponding classical computation, by “concentrating the answer”. We will see examples of this phenomenon later on.

3.10.1. The quantum adder

The examples given above are for illustration-of-principle only: you should not think of the Kröneckers δ trick as a standard tool to construct a quantum algorithm: quite the opposite. The reason is that you need to construct first the computational basis TRUTH table of your computation, which is in general non-trivial (and part of the reason to have a routine is to avoid constructing such an object case-by-case).

To better understand this point, imagine that you want to construct a quantum algorithm to add numbers, $s = x + y$. As already discussed for the classical case, see Sec. 2.6, you do not construct truth tables, but rather:

- 1) If c_i is the carry-over from previous step, construct a circuit for the bitwise sum

$$s_i = c_i \oplus x_i \oplus y_i .$$

- 2) Construct a circuit for the next step carry-over

$$c_{i+1} = (x_i \wedge y_i) \vee ((x_i \oplus y_i) \wedge c_i) = (x_i \wedge y_i) \vee (c_i \wedge x_i) \vee (c_i \wedge y_i) .$$

Step 1) involves the standard cNOT, and you can show that it involves the circuit shown in

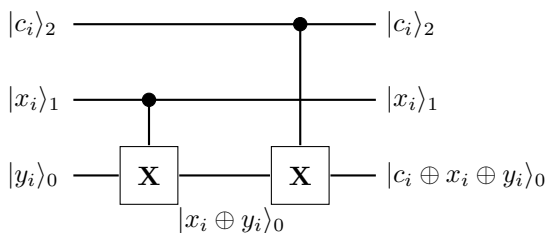


Figure 3.14: The quantum circuit to represent the bitwise sum $s_i = c_i \oplus x_i \oplus y_i$. Notice that it is precisely the reversible XOR circuit of Fig. 3.11, where the previous ancilla y , at Qbit-0, is set to $|y_i>_0$.

Step 2) involves the non-reversible OR and AND, which need to be made reversible with the ancilla trick.

Exercise 3.12. 1) Write the quantum circuit for c_{i+1} . 2) To practice, build the full circuit to sum an $n = 2$ -bit x and an $m = 2$ -bit y .

[Ans: Ref. [2][Fig.3.21, 3.22]

3.11. No-cloning theorem

We saw that the **COPY** gate was useful in classical computation (CC): it belongs to the universal set of gates. As a matter of fact, we already noticed that in principle, it can be realised by a cNOT:

$$\mathbf{C}_{10}|x_1\rangle_1 \otimes |0\rangle_0 = |x_1\rangle_1 \otimes |x_1\rangle_0 .$$

While this is a waste of resources in CC, because simpler strategies can be adopted in the digital hardware to implement **COPY**, it suggests the *wrong* impression that \mathbf{C}_{10} acts as a **quantum-COPY** machine. This is wrong: no such machine can exist, as we will now discuss.

The point is simple: while, as the previous expression for \mathbf{C}_{10} in the computational basis shows, certainly I can copy $|0\rangle$ and $|1\rangle$ and any computational basis state, I cannot copy unknown arbitrary superposition of computational basis states. We will prove this *no-go theorem* — known as *no-cloning theorem* — in three different ways.

i **No-cloning theorem.** A unitary operator \mathbf{U}_{copy} which operates the copy of an arbitrary n -bit quantum state $|\psi\rangle_n$ cannot exist.

$$\mathbf{U}_{\text{copy}}|\psi\rangle_n \otimes |0\rangle_n = |\psi\rangle_n \otimes |\psi\rangle_n \quad \forall \psi \quad (3.63)$$

Proof 1 (with emphasis on orthogonality). Take two states $|\psi\rangle_n$ and $|\phi\rangle_n$ and assume that

$$\mathbf{U}_{\text{copy}}|\psi\rangle_n \otimes |0\rangle_n = |\psi\rangle_n \otimes |\psi\rangle_n \quad \text{and} \quad \mathbf{U}_{\text{copy}}|\phi\rangle_n \otimes |0\rangle_n = |\phi\rangle_n \otimes |\phi\rangle_n .$$

Next, consider the scalar product $\langle\phi|\psi\rangle_n$. We have:

$$\begin{aligned} \langle\phi|\psi\rangle_n &= \langle 0| \otimes \langle\phi|\psi\rangle_n \otimes |0\rangle_n \\ &= \langle 0| \otimes \langle\phi|\mathbf{U}_{\text{copy}}^\dagger \mathbf{U}_{\text{copy}}|\psi\rangle_n \otimes |0\rangle_n \\ &= \langle\phi| \otimes \langle\phi|\psi\rangle_n \otimes |\psi\rangle_n \equiv \langle\phi|\psi\rangle_n^2 , \end{aligned} \quad (3.64)$$

where in the first line we have inserted $1 = \langle 0|0\rangle_n$ for the copy Qbit registers, and in the second we inserted an identity $\mathbf{U}_{\text{copy}}^\dagger \mathbf{U}_{\text{copy}} = \mathbf{1}$. So, this limits the pair of states $|\psi\rangle_n$ and $|\phi\rangle_n$ to be such that $\langle\phi|\psi\rangle_n = 0$ — **orthogonal** states —, or $\langle\phi|\psi\rangle_n = 1$ — **identical** states.

i **Non-orthogonal states cannot be copied.** This implies that I can construct such a unitary operator to copy all the states of, for instance, the computational basis — a multiple set of cNOT gates would do that, as discussed — or of any other basis, but I cannot use it to copy arbitrary non-orthogonal states.

Proof 2 (with emphasis on linearity). Take again two states $|\psi\rangle_n$ and $|\phi\rangle_n$. When acting on a linear superposition $\alpha|\psi\rangle_n + \beta|\phi\rangle_n$, \mathbf{U}_{copy} should do the following:

$$\begin{aligned} \mathbf{U}_{\text{copy}}(\alpha|\psi\rangle_n + \beta|\phi\rangle_n) \otimes |0\rangle_n &= \alpha\mathbf{U}_{\text{copy}}|\psi\rangle_n \otimes |0\rangle_n + \beta\mathbf{U}_{\text{copy}}|\phi\rangle_n \otimes |0\rangle_n \\ &= \alpha|\psi\rangle_n \otimes |\psi\rangle_n + \beta|\phi\rangle_n \otimes |\phi\rangle_n . \end{aligned} \quad (3.65)$$

But, on the other hand, by its very definition of copy-operator, \mathbf{U}_{copy} should do this:

$$\begin{aligned} \mathbf{U}_{\text{copy}}(\alpha|\psi\rangle_n + \beta|\phi\rangle_n) \otimes |0\rangle_n &= (\alpha|\psi\rangle_n + \beta|\phi\rangle_n) \otimes (\alpha|\psi\rangle_n + \beta|\phi\rangle_n) \\ &= \alpha^2|\psi\rangle_n \otimes |\psi\rangle_n + \beta^2|\phi\rangle_n \otimes |\phi\rangle_n + \alpha\beta(|\psi\rangle_n \otimes |\phi\rangle_n + |\phi\rangle_n \otimes |\psi\rangle_n) . \end{aligned} \quad (3.66)$$

The equality of Eqs. (3.65)-(3.66) requires in turn:

$$\begin{cases} \alpha\beta = 0 \\ \alpha^2 = \alpha \\ \beta^2 = \beta \end{cases} \iff \begin{cases} \alpha = 1 \\ \beta = 0 \end{cases} \text{ OR } \begin{cases} \alpha = 0 \\ \beta = 1 \end{cases} . \quad (3.67)$$

Exercise 3.13. Apply \mathbf{C}_{10} to a state $|+\rangle_1 \otimes |0\rangle_0 = \mathbf{H}_1|00\rangle$. What is the resulting state? Observe how different it is from the non-entangled (product) state $|+\rangle_1 \otimes |+\rangle_0$.

i

A possible objection. To these proofs one might object that, perhaps, the quantum-copy-machine involves another (larger) Hilbert space \mathcal{H}_C , and that the initial state is $|\psi\rangle_n \otimes |0\rangle_n \otimes |\mathcal{C}_{\text{in}}\rangle_C$ and the unitary works as follows:

$$\mathbf{U}_{\text{copy}}|\psi\rangle_n \otimes |0\rangle_n \otimes |\mathcal{C}_{\text{in}}\rangle_C = |\psi\rangle_n \otimes |\psi\rangle_n \otimes |\mathcal{C}_{\psi}\rangle_C \quad \forall \psi , \quad (3.68)$$

with a final state $|\mathcal{C}_{\psi}\rangle_C$ which depends on the state ψ .

Exercise 3.14. Repeat the linearity-based proof in this more general setting, showing that the no-cloning theorem still holds.

Proof 3 (with emphasis on measurement). Suppose, for concreteness of argumentation, that the state that we plan to copy is a single Qbit state

$$|\psi\rangle = |+, \mathbf{n}\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle .$$

The Measurement Postulate of QM tells us that with a *single measurement* I can only obtain one of the eigenvalues of the operator which is being measured, with an ensuing *collapse* of the state. For instance, with a measurement performed on $|+, \mathbf{n}\rangle$ in the computational basis (i.e., measuring $\frac{1}{2}(1 - \hat{\sigma}^z)$) I get 0 or 1, with probabilities $\mathbb{P}_0 = \cos^2 \frac{\theta}{2}$ and $\mathbb{P}_1 = \sin^2 \frac{\theta}{2}$, collapsing the state to $|0\rangle$ or $|1\rangle$, respectively. Usually, one imagines of repeating the measurement over-and-over again on an *ensemble of identically prepared states* $|\psi\rangle$, which requires following a certain recipe of experimental preparation procedure to get $|\psi\rangle$: this should *not be confused with quantum-cloning*.⁴

Now, *by contradiction*: Suppose that quantum-cloning is possible. Then, the experimentalist could construct a very complex measurement apparatus that *includes the quantum-cloning-machine*, measure on these copies of $|\psi\rangle$ the average spin $\langle \psi | \hat{\mathbf{S}} | \psi \rangle = \frac{1}{2}(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, and therefore get full information on the state $|\psi\rangle$, i.e., the values of θ and ϕ , from a *single preparation* of $|\psi\rangle$. This violates the Measurement Postulate.

For useful remarks on how cloning would impact on “faster than light” communication in Bell’s pair EPR experiments, see [2][Sec. 5.2.1]. A useful reading is also [2][Sec. 5.2.2-5.2.4].

3.12. The Deutsch’s problem

So far we have illustrated how to write quantum circuit to make computations that a classical digital computer would do even more efficiently, at least from the point of view of the memory involved: recall

⁴“Quantum-cloning” is about applying a unitary operator to make identical copies of an unknown state $|\psi\rangle$.

that reversibility requests using ancillary bits in the computation. Now we come to illustrating the first problem — admittedly very simple and academic — where you can show a definite *quantum speedup* of the QC. Here is the problem, known as *Deutsch’s problem*.

Consider the 4 single-bit function $f : \{0, 1\} \rightarrow \{0, 1\}$, which I rewrite here for convenience:

x_0	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

You recognise that $f_0(x_0) = \mathbf{E}_{\text{rase}}(x_0) = 0$, $f_1(x_0) = x_0$, $f_2(x_0) = \bar{x}_0 = \mathbf{NOT}(x_0)$, $f_3(x_0) = \mathbf{NOT}(f_0) = 1$. We make all of them reversible with the usual extension procedure: $\tilde{f}(x_0, y) = (x_0, y \oplus f(x_0))$. The corresponding circuits are shown in Fig. 3.15.

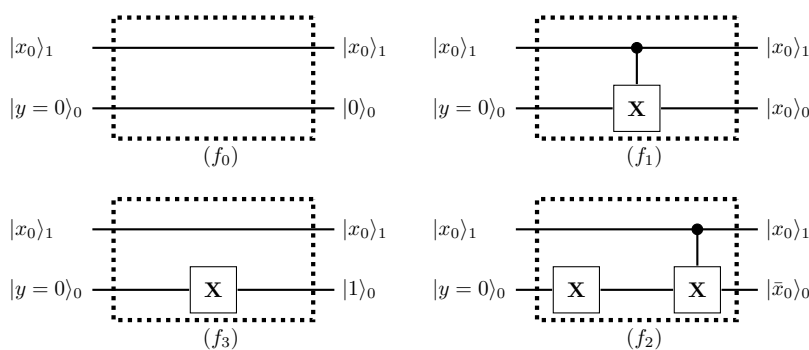


Figure 3.15: The four circuits for the reversible computation of the single-bit functions. The red dotted rectangles allude to the fact that we should start thinking to such circuits as a “black box”, of which we do not know the details. **Constant** functions are on the left, **non-constant** on the right.

Suppose — this is the rule of the game — that \mathbf{U}_f is coded into some “black box” — often called a *oracle*, because I can interrogate it, getting an answer, but without further information on the inner mechanism — and I can call the black box to know if the f coded by \mathbf{U}_f is *constant* (like f_0 or f_3) or *not-constant* (like f_1 of f_2).

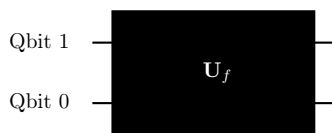


Figure 3.16: The “black box” encoding \mathbf{U}_f , of which we do not know the constructive details.

In a classical computation, I would need to invoke the oracle \mathbf{U}_f *twice*, calculating:

$$\mathbf{U}_f|0\rangle_1|0\rangle_0 = |0\rangle_1|f(0)\rangle_0 \quad \text{and} \quad \mathbf{U}_f|1\rangle_1|0\rangle_0 = |1\rangle_1|f(1)\rangle_0 ,$$

to know the function and hence answer the question. On a quantum computer, we can exploit *superpositions* — but we have to do this wisely — and answer with a *single call* of the oracle \mathbf{U}_f .



Naif application of superpositions. The naif application of superpositions offers no advantage. If you start from $|\psi_{\text{in}}\rangle = \mathbf{H}_1|0\rangle_1|0\rangle_0 = \frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1)|0\rangle_0$, you would get:

$$\mathbf{U}_f|\psi_{\text{in}}\rangle = \mathbf{U}_f\mathbf{H}_1|0\rangle_1|0\rangle_0 = \frac{1}{\sqrt{2}}(|0\rangle_1|f(0)\rangle_0 + |1\rangle_1|f(1)\rangle_0) ,$$

which suffers from the general problem. It is a *fake quantum parallelism*: when I make a measurement, I either get $|0\rangle_1$ or $|1\rangle_1$, with probability $\frac{1}{2}$, and I have no control of what I get! So, no advantage whatsoever on the classical computation.

Notice, however, the magic that occurs if we start from:

$$|\psi_{\text{in}}\rangle = \mathbf{H}_1 \mathbf{H}_0 \mathbf{X}_0 |0\rangle_1 |0\rangle_0 = \mathbf{H}_1 \mathbf{H}_0 |0\rangle_1 |1\rangle_0 = \frac{1}{2} (|00\rangle + |10\rangle - |01\rangle - |11\rangle).$$

Then, by linearity:

$$\begin{aligned} \mathbf{U}_f |\psi_{\text{in}}\rangle &= \frac{1}{2} (\mathbf{U}_f |00\rangle + \mathbf{U}_f |10\rangle - \mathbf{U}_f |01\rangle - \mathbf{U}_f |11\rangle) \\ &= \frac{1}{2} (|0\rangle_1 |f(0)\rangle_0 + |1\rangle_1 |f(1)\rangle_0 - |0\rangle_1 |\overline{f(0)}\rangle_0 - |1\rangle_1 |\overline{f(1)}\rangle_0). \end{aligned} \quad (3.69)$$

Now you observe that if $f(0) = f(1)$ (no matter the values), then:

$$\mathbf{U}_f |\psi_{\text{in}}\rangle = \frac{1}{2} (|0\rangle_1 + |1\rangle_1) (|f(0)\rangle_0 - |\overline{f(0)}\rangle_0) \quad \text{if} \quad f(0) = f(1).$$

Otherwise, for $f(0) \neq f(1)$, since $\overline{f(0)} = f(1)$ and $\overline{f(1)} = f(0)$:

$$\mathbf{U}_f |\psi_{\text{in}}\rangle = \frac{1}{2} (|0\rangle_1 - |1\rangle_1) (|f(0)\rangle_0 - |\overline{f(0)}\rangle_0) \quad \text{if} \quad f(0) \neq f(1).$$

Hence, if you act with a final Hadamard on Qbit-1, after acting with \mathbf{U}_f , you get:

$$\mathbf{H}_1 \mathbf{U}_f |\psi_{\text{in}}\rangle = \begin{cases} \frac{1}{\sqrt{2}} |0\rangle_1 (|f(0)\rangle_0 - |\overline{f(0)}\rangle_0) & \text{for} \quad f(0) = f(1) \\ \frac{1}{\sqrt{2}} |1\rangle_1 (|f(0)\rangle_0 - |\overline{f(0)}\rangle_0) & \text{for} \quad f(0) \neq f(1) \end{cases}. \quad (3.70)$$

You see the miracle, similar to the phenomenon observed when acting with the final beam-splitter in the Mach-Zehnder interferometer discussed in the Introduction: a superposition has been transformed into a *certain outcome*. By measuring Qbit-1 in the computational basis, if I find 0, I know that $f(0) = f(1)$, hence the function is constant; if I find 1, then $f(0) \neq f(1)$.



What we gain and what we pay. By measuring the output register Qbit-0 we do not learn anything useful on the actual value, say, of $f(0)$. Indeed, the result is equally likely to give a collapse to $|f(0)\rangle_0$ or to $|\overline{f(0)}\rangle_0$. To exemplify, suppose I got 0 in measuring Qbit-1, so $f(0) = f(1)$. Then, if $f(0) = f(1) = 0$ (as for f_0) the collapsed state is

$$|0\rangle_1 \frac{1}{\sqrt{2}} (|0\rangle_0 - |1\rangle_0).$$

If $f(0) = f(1) = 1$ (as for f_3), the collapsed state is:

$$|0\rangle_1 \frac{1}{\sqrt{2}} (|1\rangle_0 - |0\rangle_0),$$

which differs from the previous by an unobservable *minus sign*. Thus, the price paid for being able to answer with a single call of \mathbf{U}_f is that we actually do not know the value of $f(0)$ or $f(1)$. What a QC can give us is the ability to discriminate between constant and non-constant f with a single call of \mathbf{U}_f : no classical computation can do that.

If we want to give to the treatment of the two Qbits a more symmetric look, it is appropriate to add a final \mathbf{H}_0 . We can easily show that this leads to:

$$\mathbf{H}_1 \mathbf{H}_0 \mathbf{U}_f |\psi_{\text{in}}\rangle = \mathbf{H}_1 \mathbf{H}_0 \mathbf{U}_f \mathbf{H}_1 \mathbf{H}_0 |0\rangle_1 |1\rangle_0 = \begin{cases} (-1)^{f(0)} |0\rangle_1 |1\rangle_0 & \text{for} \quad f(0) = f(1) \\ (-1)^{f(0)} |1\rangle_1 |1\rangle_0 & \text{for} \quad f(0) \neq f(1) \end{cases}. \quad (3.71)$$

It is instructive to see what happens to the circuits “inside” the black-box \mathbf{U}_f for the cases shown in Fig. 3.15, when the $\mathbf{H}_1 \mathbf{H}_0 \mathbf{U}_f \mathbf{H}_1 \mathbf{H}_0 |0\rangle_1 |1\rangle_0$ is considered. The results — obtained by using simple identities like $\mathbf{H}^2 = \mathbf{1}$, $\mathbf{H}\mathbf{X}\mathbf{H} = \mathbf{Z}$, and $\mathbf{H}_1 \mathbf{H}_0 \mathbf{C}_{10} \mathbf{H}_1 \mathbf{H}_0 = \mathbf{C}_{01}$ — are shown in Fig. 3.17.

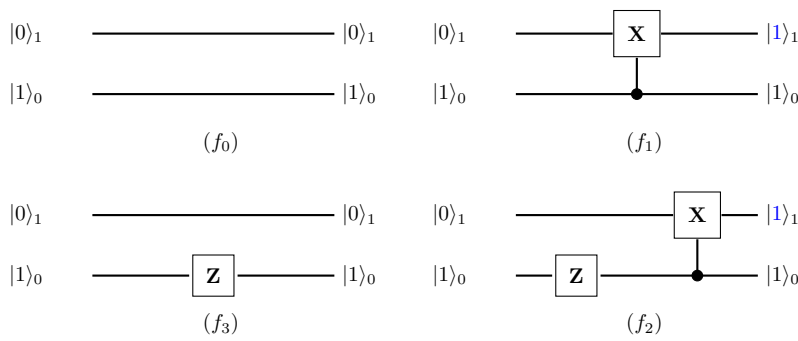


Figure 3.17: The four circuits for the reversible computation of the single-bit functions, in the version for $\mathbf{H}_1\mathbf{H}_0\mathbf{U}_f\mathbf{H}_1\mathbf{H}_0$. Notice that change in Qbit-1, from $|0\rangle_1 \rightarrow |1\rangle_1$, for the two non-constant functions, on the right.

The Deutsch-Jozsa problem. An n -Qbit variant of the problem is the following. A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is either *constant* or *balanced* — i.e., on half of the input strings $f(x) = 0$, on the other half $f(x) = 1$. One can discriminate between these two situations with a single call of an oracle. See Ref. [2][Sec. 4.1.1]

3.12.1. An interesting “variant” of Deutsch’s problem, and some general remarks on the role of additional Qbits.

As formulated, the problem is eminently academic: nobody would have constructed a QC to solve it. Still academic, but less trivial is the following formulation. Take $f(x) : \{0, 1\} \rightarrow \{0, 1\}^m$ where $f(x) = \sqrt{2+x}$, written as an m -bit binary of the form $1.y_1y_2y_3 \cdots y_m$, where now increasing bits refer to less significant binary digits. Now I consider $n < m$ but very large, like $n = 10^6$, and I ask something about the n -th binary digit of f . More precisely, I define $f_n : \{0, 1\} \rightarrow \{0, 1\}$ so defined:

$$f_n(x) = n\text{-th binary digit of } \sqrt{2+x} .$$

The Deutsch’s problem for f_n would then be: is $f_n(0) = f_n(1)$? In other words, we are asking if the n -th digit of $\sqrt{2}$ and $\sqrt{3}$ coincide or not. Now, to calculate anything about f_n , you certainly need many more than two Qbits, $|x\rangle_1|y\rangle_0$, as done so far.

Question:

What would these Qbits do during the Deutsch’s problem computation, as described so far? Would they become entangled with the output register, ruining the magic we saw before?

The question can be formulated in more general terms, and is useful to do so, because it applies equally well to the general scheme for \mathbf{U}_f proposed in Sec. 3.2. So, let us suppose that we implement a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ with \mathbf{U}_f , which on the computational basis would read:

$$\mathbf{U}_f|\underline{x}\rangle_n \otimes |\underline{y}\rangle_m = |\underline{x}\rangle_n \otimes |\underline{y} \oplus f(\underline{x})\rangle_m .$$

This is represented by the circuit in Fig. 3.18.

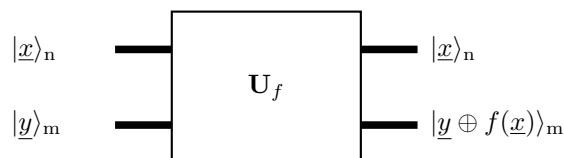


Figure 3.18: A \mathbf{U}_f acting on an n -Qbit input register $|\underline{x}\rangle_n$ and an m -Qbit output register $|\underline{y}\rangle_m$, denoted by thick lines.

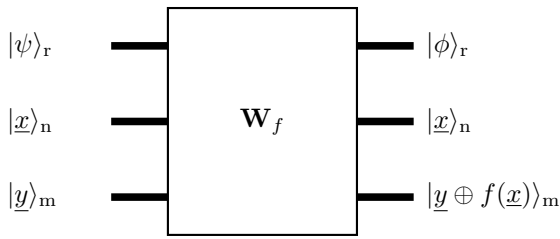


Figure 3.19: A more realistic unitary \mathbf{W}_f acting on the n -Qbit input register $|\underline{x}\rangle_n$, the m -Qbit output register $|\underline{y}\rangle_m$, and the r -Qbit register of additional Qbits involved in the computation.

Imagine, now, that the computation involves many other Qbits, say r of them. A more realistic way of thinking at the calculation would be therefore:



Warning: A crucial requirements for this more general \mathbf{W}_f to do its proper job is that $|\psi\rangle_r$ and $|\phi\rangle_r$ are *pure states* independent of the initial content of $|\underline{x}\rangle_n|\underline{y}\rangle_m$, so that, when restricted to the lower $(n + m)$ Qbits, the resulting transformation is the unitary \mathbf{U}_f we want.

For instance, if we can arrange things such that $|\psi\rangle_r = |0\rangle_r$, a standard fixed reference state for the extra Qbits, and $|\phi\rangle_r = |0\rangle_r$ as well, so that we return to the initial reference state, then $\langle 0|\mathbf{W}_f|0\rangle_r = \mathbf{U}_f$ does the desired job on the lower $(n + m)$ Qbits. Figure 3.20 illustrates a possible way of doing this trick.

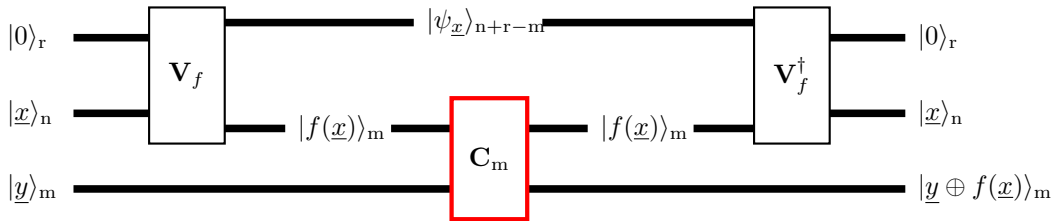


Figure 3.20.: A fully reversible computation with extra Qbits initially set in the standard reference state $|0\rangle_r$. The unitary \mathbf{V}_f prepares $|f(\underline{x})\rangle_m$ while leaving $(n + r - m)$ -Qbits in a pure state $\psi_{\underline{x}}$ which *depends* on the initial input \underline{x} . This state is however sent back to $|0\rangle_r$ by the final \mathbf{V}_f^\dagger . In the lower part, the box labelled \mathbf{C}_m is simply a multi-bit version of a standard \mathbf{C}_{10} cNOT, which transfers the m -th control Qbit of $|f(\underline{x})\rangle_m$ in the corresponding target Qbit of the output register $|\underline{y} \oplus f(\underline{x})\rangle_m$.



Warning: For the Deutsch’s problem, the 2-to-1 speed-up of QC with respect to CC would be wasted by this necessity of applying \mathbf{V}_f^\dagger after \mathbf{V}_f to return to the original reference state $|0\rangle_r$. Nevertheless, problems with a larger quantum speed-up would still maintain the advantage of QC over CC.

3.13. The Bernstein-Vazirani problem

Here is another academic problem of no intrinsic arithmetic interest which, once again, shows a quantum speed-up: this time from n classical calls to 1 quantum call. The problem is the following. Given an assigned but *unknown* n -bit integer $a \in \{0, 1, \dots, 2^n - 1\}$, hence an associated binary string $\underline{a} = (a_{n-1}, \dots, a_0)$ with $a_j = 0, 1$, and given an n -bit integer $x \longleftrightarrow \underline{x} = (x_{n-1}, \dots, x_0)$ we define the bitwise (mod 2) scalar product of \underline{a} and \underline{x} to be the function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ given by:

$$f(\underline{x}) = \underline{a} \cdot \underline{x} \stackrel{\text{def}}{=} a_0x_0 \oplus a_1x_1 \oplus \dots \oplus a_{n-1}x_{n-1} . \tag{3.72}$$

1 **Example with $n = 5$.** For $\underline{a} = (1, 0, 1, 1, 0) \equiv 22$, and $\underline{x} = (0, 1, 1, 1, 1) \equiv 15$ then

$$\underline{a} \cdot \underline{x} = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1 \oplus 1 = 0 .$$

Question:

Suppose you want to discover the unknown value of \underline{a} by calling $f(\underline{x})$ repeatedly. How many calls would you need?

With a classical computer, I would need n calls. The best strategy is to call $f(\underline{x})$ for $x = 2^j$ which corresponds to $\underline{x}^{(j)} = (0 \dots 0 1_j 0 \dots 0)$, where 1_j means “a 1 at position j ”. Indeed in this case:

$$f(\underline{x}^{(j)}) = a_j 1 = a_j .$$

Here is, for instance, a circuit for $f(\underline{x})$, assuming that $n = 5$ with $\underline{a} = (1, 0, 1, 1, 0)$.

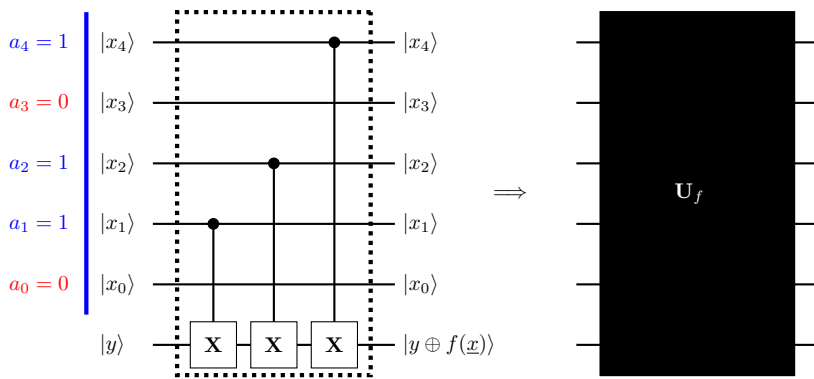


Figure 3.21: The circuit for evaluating the Bernstein-Vazirani $f(x) = \underline{a} \cdot \underline{x}$ for $n = 5$ with $\underline{a} = (1, 0, 1, 1, 0)$, as you immediately verify by working on computational states of the form $\underline{x}^{(j)}$. The right-hand-side reminds us that this should be considered as a “black box” (oracle) where the positions of the control Qbits in the various cNOT is unknown.

Let us now observe what happens if we use Hadamards before and after U_f . We do that directly at the circuit level, for the same example shown in Fig. 3.21. Recall the identity in Eq. (3.24), which we repeat here for convenience:

$$\mathbf{H}_i \mathbf{H}_j \mathbf{C}_{ij} \mathbf{H}_i \mathbf{H}_j = \mathbf{C}_{ji} . \tag{3.73}$$

This is expressed by the circuit identity in Fig. 3.22.

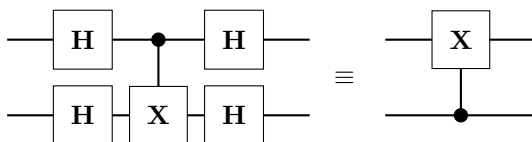


Figure 3.22: The identity in Eq. (3.73).

Applying this identity repeatedly, we therefore get the circuit shown in Fig. 3.23. Notice how, starting from the initial state $|\psi_0\rangle = |0 \dots 0\rangle_n \otimes |y = 1\rangle$, we get directly the value of \underline{a} in the final values of the input registers, with a *single* application of U_f .

One of the ingredients that helps in getting the solution is the following. When the ancilla Qbit is in the state $\mathbf{H}|1\rangle$, the application of U_f amounts to encoding the value of $f(x)$, for *any* function

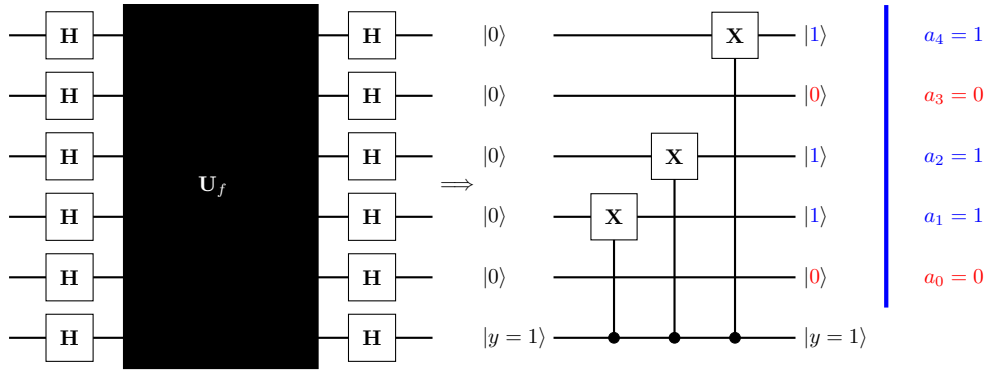


Figure 3.23.: The circuit for the Bernstein-Vazirani problem when Hadamards are used before and after U_f . The value of the unknown \underline{a} is read directly from the input register at the end of the transformation.

$f : \{0, 1\}^n \rightarrow \{0, 1\}$, into an overall sign:

$$\begin{aligned}
 U_f |x\rangle_n \otimes (\mathbf{H}|1\rangle) &= U_f |x\rangle_n \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
 &= \frac{1}{\sqrt{2}} |x\rangle_n \otimes (|f(x)\rangle - |\overline{f(x)}\rangle) = (-1)^{f(x)} |x\rangle_n \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
 &= (-1)^{f(x)} |x\rangle_n \otimes (\mathbf{H}|1\rangle). \tag{3.74}
 \end{aligned}$$

This transformation will be also useful in the Grover’s search problem we will later discuss.

The following exercise will guide you in discovering why the special nature of the Bernstein-Vazirani $f(x)$ leads to the solution we found diagrammatically.

Exercise 3.15. [The Bernstein-Vazirani problem.]

1) Show that, for a single Qbit:

$$\mathbf{H}|x\rangle_1 = \frac{1}{\sqrt{2}}(|0\rangle_1 + (-1)^x |1\rangle_1) = \frac{1}{\sqrt{2}} \sum_{k=0}^1 (-1)^{kx} |k\rangle_1.$$

2) Generalising to n Qbits, show that:

$$\mathbf{H}^{\otimes n} |x\rangle_n = \mathbf{H}_{n-1} \cdots \mathbf{H}_0 |x_{n-1}\rangle \cdots |x_0\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} (-1)^{k \cdot x} |k\rangle_n.$$

[Hint: $(-1)^{\sum_j k_j x_j} = (-1)^{k \cdot x}$ in terms of the bitwise- (mod 2) scalar product.]

3) Prove that:

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{a \cdot x + k \cdot x} = \frac{1}{2^n} \prod_{j=0}^{n-1} \left(\sum_{x_j=0}^1 (-1)^{(a_j + k_j) x_j} \right) = \delta_{\underline{a}, \underline{k}}. \tag{3.75}$$

4) Finally show that:

$$\mathbf{H}^{\otimes(n+1)} U_f \mathbf{H}^{\otimes(n+1)} |0\rangle_n \otimes |1\rangle = |\underline{a}\rangle_n |1\rangle. \tag{3.76}$$

1

Simon's problem. The problem has to do with a two-to-one function $f : \{0, 1\}^n \rightarrow \{0, 1\}^{n-1}$ which is periodic under bitwise modulo-2 addition, i.e., such that:

$$f(x \oplus a) = f(x) .$$

The goal is to find the n-bit integer “period” a . Here QC leads to an exponential speed-up: $O(n)$ calls are enough, against the $O(2^{n/2})$ calls needed in CC. Since this is a less interesting version of Shor's period-finding algorithm, where the periodicity is with respect to ordinary addition, we will not discuss it here. If you are interested, see Mermin [1][Sec. 2.5].

3.14. Teleportation

Suppose that two parties, A and B, share a Bell state, say $|\beta_{00}\rangle$. Assuming that A has Qbit 1, while B has Qbit 0, we write:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|0\rangle_1 \otimes |0\rangle_0 + |1\rangle_1 \otimes |1\rangle_0) = \mathbf{C}_{10}\mathbf{H}_1|0\rangle_1 \otimes |0\rangle_0 . \quad (3.77)$$

Assume that A also holds a Qbit 2 in the (unknown) state $|\psi\rangle_2$. The overall state is therefore:

$$|\Psi_{\text{in}}\rangle = \frac{1}{\sqrt{2}}|\psi\rangle_2 \otimes (|0\rangle_1 \otimes |0\rangle_0 + |1\rangle_1 \otimes |1\rangle_0) = \mathbf{C}_{10}\mathbf{H}_1|\psi\rangle_2 \otimes |0\rangle_1 \otimes |0\rangle_0 . \quad (3.78)$$

1

The goal of teleportation. The goal is to devise a protocol by which the unknown state $|\psi\rangle_2 = z_0|0\rangle_2 + z_1|1\rangle_2$ of A is **transferred** to B, as $|\psi\rangle_0 = z_0|0\rangle_0 + z_1|1\rangle_0$. How to do that?

Here is the protocol to be used. A first applies \mathbf{C}_{21} , and then \mathbf{H}_2 to $|\Psi_{\text{in}}\rangle$. The state is then transformed as follows:

$$\begin{aligned} \mathbf{H}_2\mathbf{C}_{21}|\Psi_{\text{in}}\rangle &= \mathbf{H}_2\mathbf{C}_{21}\frac{1}{\sqrt{2}}\left(z_0|0\rangle_2 \otimes (|0\rangle_1 \otimes |0\rangle_0 + |1\rangle_1 \otimes |1\rangle_0) + z_1|1\rangle_2 \otimes (|0\rangle_1 \otimes |0\rangle_0 + |1\rangle_1 \otimes |1\rangle_0)\right) \\ &= \frac{1}{\sqrt{2}}\mathbf{H}_2\left(z_0|0\rangle_2 \otimes (|0\rangle_1 \otimes |0\rangle_0 + |1\rangle_1 \otimes |1\rangle_0) + z_1|1\rangle_2 \otimes (|1\rangle_1 \otimes |0\rangle_0 + |0\rangle_1 \otimes |1\rangle_0)\right) \\ &= \frac{1}{2}\left(z_0(|0\rangle_2 + |1\rangle_2) \otimes (|0\rangle_1 \otimes |0\rangle_0 + |1\rangle_1 \otimes |1\rangle_0) \right. \\ &\quad \left. + z_1(|0\rangle_2 - |1\rangle_2) \otimes (|1\rangle_1 \otimes |0\rangle_0 + |0\rangle_1 \otimes |1\rangle_0)\right) \\ &= \frac{1}{2}|0\rangle_2 \otimes |0\rangle_1 \otimes (z_0|0\rangle_0 + z_1|1\rangle_0) + \frac{1}{2}|1\rangle_2 \otimes |0\rangle_1 \otimes (z_0|0\rangle_0 - z_1|1\rangle_0) \\ &\quad + \frac{1}{2}|0\rangle_2 \otimes |1\rangle_1 \otimes (z_0|1\rangle_0 + z_1|0\rangle_0) + \frac{1}{2}|1\rangle_2 \otimes |1\rangle_1 \otimes (z_0|1\rangle_0 - z_1|0\rangle_0) \\ &= \frac{1}{2}|0\rangle_2 \otimes |0\rangle_1 \otimes |\psi\rangle_0 + \frac{1}{2}|1\rangle_2 \otimes |0\rangle_1 \otimes \mathbf{Z}_0|\psi\rangle_0 \\ &\quad + \frac{1}{2}|0\rangle_2 \otimes |1\rangle_1 \otimes \mathbf{X}_0|\psi\rangle_0 + \frac{1}{2}|1\rangle_2 \otimes |1\rangle_1 \otimes \mathbf{X}_0\mathbf{Z}_0|\psi\rangle_0 . \end{aligned} \quad (3.79)$$

Following that, A measures the two Qbits (2 and 1) in the computational basis. If the result of the measurement is (00) , the state is collapsed to:

$$\mathbf{H}_2\mathbf{C}_{21}|\Psi_{\text{in}}\rangle \xrightarrow{\text{A measures } (00)} |0\rangle_2 \otimes |0\rangle_1 \otimes |\psi\rangle_0 ,$$

hence B has directly the state $|\psi\rangle_0$. Similarly, for the measurement outcome (10):

$$\mathbf{H}_2\mathbf{C}_{21}|\Psi_{\text{in}}\rangle \xrightarrow{\text{A measures (10)}} |1\rangle_2 \otimes |0\rangle_1 \otimes \mathbf{Z}_0|\psi\rangle_0 ,$$

hence B has the state $\mathbf{Z}_0|\psi\rangle_0$. A communicates the outcome of the measurement, and B can apply \mathbf{Z}_0 to the state in his possession, reconstructing once again $|\psi\rangle_0$. For the measurement outcome (01):

$$\mathbf{H}_2\mathbf{C}_{21}|\Psi_{\text{in}}\rangle \xrightarrow{\text{A measures (01)}} |0\rangle_2 \otimes |1\rangle_1 \otimes \mathbf{X}_0|\psi\rangle_0 ,$$

hence B has the state $\mathbf{X}_0|\psi\rangle_0$, and needs to apply \mathbf{X}_0 to the state to reconstruct $|\psi\rangle_0$. Finally, for the measurement outcome (11):

$$\mathbf{H}_2\mathbf{C}_{21}|\Psi_{\text{in}}\rangle \xrightarrow{\text{A measures (11)}} |1\rangle_2 \otimes |1\rangle_1 \otimes \mathbf{X}_0\mathbf{Z}_0|\psi\rangle_0 ,$$

hence B needs to apply $\mathbf{Z}_0\mathbf{X}_0$ to the state to reconstruct $|\psi\rangle_0$. Figure 3.24 illustrates the protocol we have discussed.

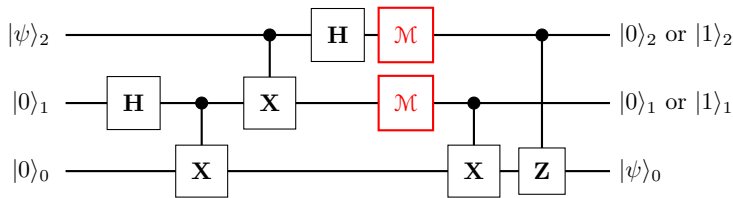


Figure 3.24: The circuit for the quantum teleportation protocol discussed in the text. Assuming that the two parties are far away, the control-gates applied in the final state of the protocol should be intended to follow an explicit *classical* communication of the measurement outcome from A to B.

Warning: You should notice a few facts.

- 1) The original state $|\psi\rangle_2$ has been destroyed, hence this is not a cloning machine, obviously;
- 2) The price paid for transferring the state was that the original entanglement between Qbit-1 and 0 has been destroyed;
- 3) Finally, A had to send to B *two classical bits of information*, the results of the measurements of Qbits 2 and 1, in order for B to apply the appropriate gates and recover $|\psi\rangle_0$.

Interestingly, if the various Qbits are not really belonging to two separate parties A and B, but are part of the same quantum hardware, then the teleportation protocol can be implemented in a totally automatic fashion, as illustrated in Fig. 3.25.

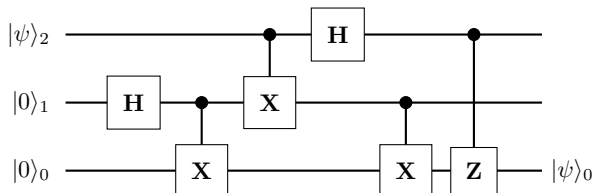


Figure 3.25: The automated version of the circuit for the quantum teleportation protocol, without the measurements.

Finally, since the control-gates by definition do not change the state of the control-Qbits, we can eventually perform a measurement of Qbit-1 and 2, obtaining exactly the same result as in the original protocol, where the measurements were performed *before* the control-gates, in turn activated by B *after* being informed by A about the outcome of the measurements. This last possibility is shown in Fig. 3.26.

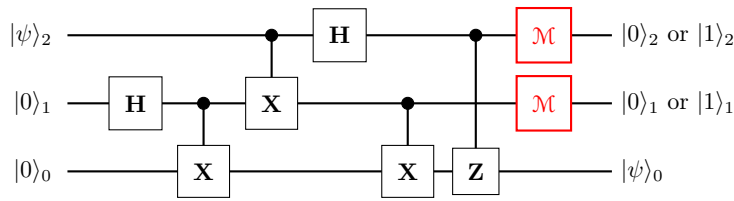


Figure 3.26: The automated version of the circuit for the quantum teleportation protocol, with the measurements performed at the very end.

3.15. Hands-on: state preparation, control-U and Toffoli gates

The goal of this section is to let you practice. You will first learn, in Sec. 3.15.1, how to construct general 2-Qbit states, generalising the Bell’s state construction of Sec. 3.6.



Exponential complexity of quantum state preparation. A warning is appropriate here. Constructing a desired n-Qbit quantum state is a generally very complex task, of exponential difficulty, essentially equivalent to representing a general n-Qbit unitary U in terms of single-Qbit unitaries and cNOTs. Ref. [2][Sec. 3.7.1] shows the 3-Qbit case, if you are interested.

The standard preparation, which is simple and immensely useful, is the democratic superposition of all computational states, obtained by applying n Hadamards:

$$|+\rangle_n = \mathbf{H}^{\otimes n} |0\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n . \tag{3.80}$$

Next, you will learn, in Sec. 3.15.2-3.15.3, how to construct a general control- U gate, and also a doubly-controlled-NOT (the Toffoli gate) out of standard cNOTs and single-Qbit rotations.

These are the typical instructive-but-boring applications which a teacher would never like to lecture on: instructive if you do them yourself, boring otherwise.

3.15.1. Representing a general 2-Qbit state

Consider a general 2-Qbit state, which we write as:

$$\begin{aligned} |\Psi\rangle &= z_{00}|00\rangle + z_{01}|01\rangle + z_{10}|10\rangle + z_{11}|11\rangle \\ &= |0\rangle_1 \otimes \underbrace{(z_{00}|0\rangle_0 + z_{01}|1\rangle_0)}_{|\psi'\rangle_0} + |1\rangle_1 \otimes \underbrace{(z_{10}|0\rangle_0 + z_{11}|1\rangle_0)}_{|\phi'\rangle_0} \\ &= |0\rangle_1 \otimes |\psi'\rangle_0 + |1\rangle_1 \otimes |\phi'\rangle_0 , \end{aligned} \tag{3.81}$$

where we have rewritten $|\Psi\rangle$ as an entangled superposition of the two computational states of Qbit-1 with two states of Qbit-0, $|\psi'\rangle_0$ and $|\phi'\rangle_0$, which are generally *not orthogonal*.

Exercise 3.16. Show that with a suitable unitary U_1 acting on Qbit-1 only, you can always write:

$$U_1^\dagger \otimes \mathbf{1}_0 |\Psi\rangle = a_0 |0\rangle_1 \otimes |\psi\rangle_0 + b_0 |1\rangle_1 \otimes |\phi\rangle_0 \tag{3.82}$$

where now $|\psi\rangle_0$ and $|\phi\rangle_0$ are *orthogonal*, $\langle \psi | \phi \rangle_0 = 0$, and *normalised*.

Hint: Write a general 2×2 unitary matrix U_1^\dagger on the computational basis $\{|0\rangle_1, |1\rangle_1\}$ as:

$$U_1^\dagger = \begin{pmatrix} u & -v^* \\ v & u^* \end{pmatrix} \quad \text{with} \quad |u|^2 + |v|^2 = 1 ,$$

and show that you can impose orthogonality on the Qbit-0 states by solving a quadratic equation for u/v^* .

Being $|\psi\rangle_0$ and $|\phi\rangle_0$ now orthogonal and normalised, a *unitary* transformation \mathbf{V}_0 acting on Qbit-0 must exist which transforms the standard computational basis $\{|0\rangle_0, |1\rangle_0\}$ into $\{|\psi\rangle_0, |\phi\rangle_0\}$.

Exercise 3.17. By applying \mathbf{U}_1 to both sides of Eq. (3.82), show that we can write:

$$|\Psi\rangle = \mathbf{U}_1 \mathbf{V}_0 \mathbf{C}_{10} (a_0|0\rangle_1 + b_0|1\rangle_1) \otimes |0\rangle_0. \quad (3.83)$$

Now you observe that the state $|\Psi\rangle$ is normalised, and all the operators acting are unitaries, hence the state $a_0|0\rangle_1 + b_0|1\rangle_1$ is also normalised, which means that a third rotation \mathbf{W}_1 must exist, for Qbit-1, such that:

$$a_0|0\rangle_1 + b_0|1\rangle_1 = \mathbf{W}_1|0\rangle_1.$$

Hence, finally:

i

Representing $|\Psi\rangle$ with 1-Qbit unitaries and \mathbf{C}_{10} . We have shown that, for a general 2-Qbit state $|\Psi\rangle$, three 1-Qbit unitaries \mathbf{U}_1 , \mathbf{V}_0 and \mathbf{W}_1 must exist such that:

$$|\Psi\rangle = \mathbf{U}_1 \mathbf{V}_0 \mathbf{C}_{10} \mathbf{W}_1 |00\rangle. \quad (3.84)$$

The corresponding circuit is show in Fig. 3.27. This shows the central role played by \mathbf{C}_{10} in creating entanglement.

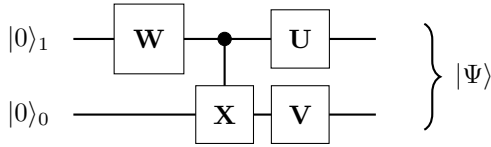


Figure 3.27: The circuit representing Eq. (3.84), generalising the Bell's states construction of Fig. 3.9.

3.15.2. Constructing control-unitary operators

Consider two operators $\mathbf{A} = \mathbf{a} \cdot \hat{\sigma}$ and $\mathbf{B} = \mathbf{b} \cdot \hat{\sigma}$, with $|\mathbf{a}| = |\mathbf{b}| = 1$: essentially, two spin operators in directions \mathbf{a} and \mathbf{b} . As you already know, it is simple to show that:

$$\mathbf{A}^2 = \mathbf{B}^2 = \mathbf{1}.$$

This means that not only \mathbf{A} and \mathbf{B} are Hermitian, but they are also unitary, since $\mathbf{A}^{-1} = \mathbf{A}^\dagger = \mathbf{A}$, and $\mathbf{B}^{-1} = \mathbf{B}^\dagger = \mathbf{B}$. Assuming that $\mathbf{b} \neq \mathbf{a}$, you can always define a unit vector \mathbf{m} orthogonal to both:

$$\mathbf{a} \times \mathbf{b} = \mathbf{m} \sin \theta,$$

where θ is the angle from \mathbf{a} to \mathbf{b} , hence $\mathbf{a} \cdot \mathbf{b} = \cos \theta$.

Exercise 3.18. 1) Show that by considering the spin rotation by θ around the axis \mathbf{m} :

$$\mathbf{U}_{\mathbf{m}}(\theta) = e^{-i\frac{\theta}{2}\mathbf{m}\cdot\hat{\sigma}} = \mathbf{1} \cos \frac{\theta}{2} - i(\mathbf{m} \cdot \hat{\sigma}) \sin \frac{\theta}{2},$$

you can “rotate \mathbf{A} into \mathbf{B} ”:

$$\mathbf{U}_{\mathbf{m}}(\theta)(\mathbf{a} \cdot \hat{\sigma})\mathbf{U}_{\mathbf{m}}^\dagger(\theta) = (\mathbf{b} \cdot \hat{\sigma}). \quad (3.85)$$

In particular, taking $\mathbf{b} = \mathbf{x}$, this shows that you can rotate any \mathbf{A} into a $\hat{\sigma}^x$.

2) Show that you can similarly “rotate $\mathbf{C}_{10}^{\mathbf{A}}$ into $\mathbf{C}_{10}^{\mathbf{B}}$ ” by $\mathbf{U}_0 = \mathbf{1}_1 \otimes (\mathbf{U}_{\mathbf{m}}(\theta))_0$:

$$\mathbf{U}_0 \mathbf{C}_{10}^{\mathbf{A}} \mathbf{U}_0^\dagger = \mathbf{C}_{10}^{\mathbf{B}}. \quad (3.86)$$

For $\mathbf{b} = \mathbf{x}$, this shows that we can write $\mathbf{U}_0 \mathbf{C}_{10}^{\mathbf{A}} \mathbf{U}_0^\dagger = \mathbf{C}_{10}$.

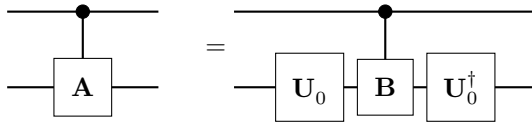


Figure 3.28: The circuit representing Eq. (3.86). When $\mathbf{B} = \mathbf{X}$ we have the relationship between any controlled spin operator $\mathbf{A} = \mathbf{a} \cdot \hat{\sigma}$ and the standard \mathbf{C}_{10} .

Consider now \mathbf{C}_{10}^U with a *general* 1-Qbit \mathbf{U} , which we can always write as:

$$\mathbf{U} = e^{i\alpha} e^{i\frac{\gamma}{2} \mathbf{n} \cdot \hat{\sigma}} = e^{i\alpha} \left(\mathbf{1} \cos \frac{\gamma}{2} + i(\mathbf{n} \cdot \hat{\sigma}) \sin \frac{\gamma}{2} \right).$$

Exercise 3.19. Show that by taking two vectors \mathbf{a}_1 and \mathbf{a}_2 in the plane orthogonal to \mathbf{n} , with an angle $\frac{\gamma}{2}$ between \mathbf{a}_1 and \mathbf{a}_2 , you can write:

$$\mathbf{U} = e^{i\alpha} \left(\mathbf{1} \cos \frac{\gamma}{2} + i(\mathbf{n} \cdot \hat{\sigma}) \sin \frac{\gamma}{2} \right) = e^{i\alpha} (\mathbf{a}_1 \cdot \hat{\sigma})(\mathbf{a}_2 \cdot \hat{\sigma}). \tag{3.87}$$

[Hint: Use the Pauli identity: $(\mathbf{a}_1 \cdot \hat{\sigma})(\mathbf{a}_2 \cdot \hat{\sigma}) = (\mathbf{a}_1 \cdot \mathbf{a}_2)\mathbf{1} + i(\mathbf{a}_1 \times \mathbf{a}_2) \cdot \hat{\sigma}$.

Exercise 3.20. Define now a (\mathbf{m}_1, θ_1) to “rotate \mathbf{a}_1 into \mathbf{x} ”, and a similar (\mathbf{m}_2, θ_2) which rotates \mathbf{a}_2 into \mathbf{x} . Show that:

$$\mathbf{U} = e^{i\alpha} \mathbf{U}_{\mathbf{m}_1}^\dagger(\theta_1) \hat{\sigma}^x \mathbf{U}_{\mathbf{m}_1}(\theta_1) \mathbf{U}_{\mathbf{m}_2}^\dagger(\theta_2) \hat{\sigma}^x \mathbf{U}_{\mathbf{m}_2}(\theta_2). \tag{3.88}$$

Consider now the phase-gate $\mathbf{R}_z(\alpha) = e^{i\alpha N_1}$: as you recall, it is the unitary operator adding a phase α when the Qbit is in state $|1\rangle$. Define as a shorthand $\mathbf{U}_{\mathbf{m}_1}(\theta_1) = \mathbf{U}_1$ and $\mathbf{U}_{\mathbf{m}_2}(\theta_2) = \mathbf{U}_2$.

Exercise 3.21. Show that the circuit for \mathbf{C}_{10}^U is that shown in Fig. 3.29.

Summary

Summary of control-U. We have shown that a \mathbf{C}_{10}^A , when $\mathbf{A}^2 = \mathbf{1}$, can be written in terms of a *one* cNOT supplemented by single-Qbit rotations. A general \mathbf{C}_{10}^U needs *two* cNOTs, supplemented by single-Qbit rotations and a phase-gate $\mathbf{R}_z(\alpha)$.

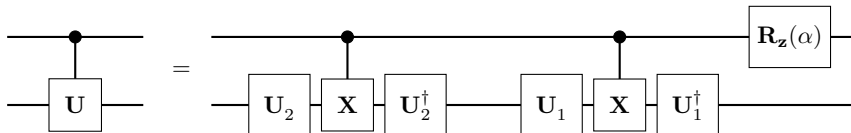


Figure 3.29: The circuit representing Eq. (3.88), in terms of the standard \mathbf{C}_{10} .

3.15.3. Constructing the Toffoli gate out of cNOTs

The goal here is to construct the Toffoli gate $\mathbf{C}_{21,0}^X$ in terms of *six* standard cNOT gates, plus single-Qbit unitaries.

Consider, as before, two spin operators $\mathbf{A} = \mathbf{a} \cdot \hat{\sigma}$ and $\mathbf{B} = \mathbf{b} \cdot \hat{\sigma}$, with $|\mathbf{a}| = |\mathbf{b}| = 1$, such that $\mathbf{A}^2 = \mathbf{B}^2 = \mathbf{1}$.

Exercise 3.22. Show that:

$$\mathbf{C}_{10}^B \mathbf{C}_{20}^A \mathbf{C}_{10}^B \mathbf{C}_{20}^A = \mathbf{C}_{21,0}^{(BA)^2}. \tag{3.89}$$

Hint: Work on the computational basis $|x_2\rangle|x_1\rangle|x_0\rangle$ and apply the operators on the LHS. For instance:

$$\mathbf{C}_{20}^A |x_2\rangle|x_1\rangle(\mathbf{A}^{x_2}|x_0\rangle).$$

Exercise 3.23. Take now, for instance, $\mathbf{b} = \mathbf{z}$ and $\mathbf{a} = \frac{1}{\sqrt{2}}(\mathbf{z} - \mathbf{y})$. Verify that:

$$\mathbf{BA} = \cos \frac{\pi}{4} + i\hat{\sigma}^x \sin \frac{\pi}{4} \quad \implies \quad (\mathbf{BA})^2 = i\hat{\sigma}^x . \quad (3.90)$$

Hence, by applying the previous exercise, we see that:

$$\mathbf{C}_{10}^{\mathbf{B}} \mathbf{C}_{20}^{\mathbf{A}} \mathbf{C}_{10}^{\mathbf{B}} \mathbf{C}_{20}^{\mathbf{A}} = \mathbf{C}_{21,0}^{\mathbf{X}} .$$

We still need to get rid of the phase due to the imaginary i , which we can do by using a phase gate unitary $\mathbf{U} = \mathbf{R}_{\mathbf{z}}(-\frac{\pi}{2})$. This has to be done, however, in a controlled fashion: it has to act *only* when Qbit-2 is in $|1\rangle_2$. Since $\mathbf{R}_{\mathbf{z}}(-\frac{\pi}{2})$ already acts non-trivially only on $|1\rangle$, we can include a $\mathbf{C}_{21}^{\mathbf{U}}$ which will apply the right phase only when needed. Hence:

$$\mathbf{C}_{21}^{\mathbf{U}} \mathbf{C}_{10}^{\mathbf{B}} \mathbf{C}_{20}^{\mathbf{A}} \mathbf{C}_{10}^{\mathbf{B}} \mathbf{C}_{20}^{\mathbf{A}} = \mathbf{C}_{21,0}^{\mathbf{X}} , \quad (3.91)$$

represented by the circuit in Fig. 3.30. Notice that the seemingly trivial controlled-phase gate, since $\mathbf{U} = \mathbf{R}_{\mathbf{z}}(-\frac{\pi}{2})$ is *not* a Pauli-unitary like \mathbf{A} and \mathbf{B} , might require *two extra* cNOTs and further single-Qbit rotations to be implemented. Each $\mathbf{C}_{10}^{\mathbf{B}}$ and $\mathbf{C}_{20}^{\mathbf{A}}$, on the other hand, require a single cNOT, plus single-Qbit unitaries. This brings, in principle, the total count of cNOTs necessary to *fix*, for each Toffoli gate.

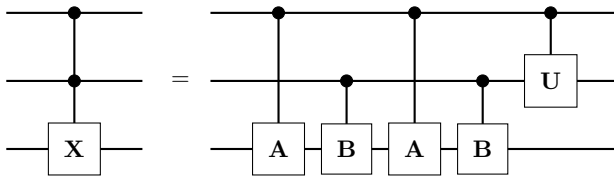


Figure 3.30: The circuit representing Eq. (3.91), expressing the Toffoli gate $\mathbf{C}_{21,0}^{\mathbf{X}}$ in terms of singly-controlled gates. Here $\mathbf{A} = \mathbf{a} \cdot \hat{\sigma}$, with $\mathbf{a} = \frac{1}{\sqrt{2}}(\mathbf{z} - \mathbf{y})$, and $\mathbf{B} = \mathbf{b} \cdot \hat{\sigma} = \mathbf{Z}$ are Pauli unitaries (each requiring a single cNOT), while the phase-gate $\mathbf{U} = \mathbf{R}_{\mathbf{z}}(-\frac{\pi}{2})$ is not a Pauli unitary, see comments in the text.



Controlled-phase gates vs general control-U gates. It might appear that a simple controlled-phase gate is as complex as a general control-U gate, requiring necessarily two cNOTs. This is a too swift view. Indeed, controlled-phase gates are much simpler objects, requiring certainly 2-Qbit interactions, but not provoking Qbit transitions: they can be implemented with $\hat{\sigma}_j^z \hat{\sigma}_j^z$ interactions. Quoting from Mermin’s book [1][pag. 62]:

If quantum computation ever becomes a working technology, it might well be easier to construct controlled-phase gates as fundamental gates in their own right — pieces of 2-Qbit hardware as basic as cNOT gates.

4. Grover searching with a quantum computer

Quantum mechanics helps in searching for a needle in a haystack.
 Lov K. Grover, Phys. Rev. Lett. **79**, 325 (1997).

Suppose you are given a strange Boolean function which is almost constant, except for a single integer input value a where it does something different. To be definite, imagine an $f : \{0, 1\}^n \rightarrow \{0, 1\}$ so defined:

$$\text{Standard: } f(x) = \begin{cases} 1 \text{ (TRUE)} & \text{for } x = a \\ 0 \text{ (FALSE)} & \text{for } x \neq a \end{cases} \quad \text{Golf-yard minimum: } f(x) = \begin{cases} 0 & \text{for } x = a \\ 1 & \text{for } x \neq a \end{cases}, \quad (4.1)$$

with $x = \sum_{i=0}^{n-1} x_i 2^i$, as usual.

I have given the problem in two equivalent formulations. The first formulation (standard) imagines a single “satisfying assignment” a which makes a certain condition TRUE (=1), while all other assignments $x \neq a$ lead to a FALSE (=0). The second formulation is more in the framework of “searching for a minimum in an energy landscape”, with the important caveat that the landscape is totally flat, except for a single minimum somewhere, a kind of “golf-yard” situation: you get no clue on where that minimum might be from looking at the “gradient”.

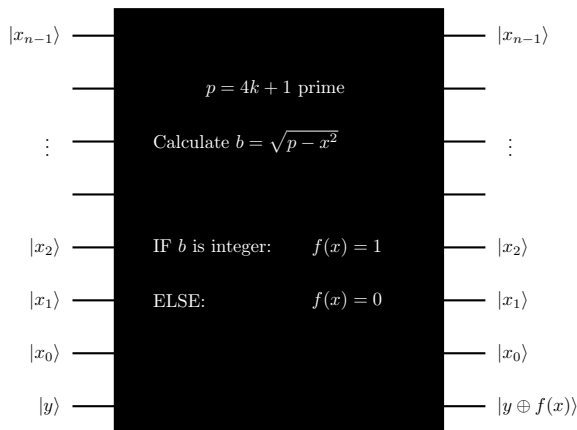


Figure 4.1: The “black box” that outputs $f(a) = 1$ (TRUE) when a second integer b exists such that $a^2 + b^2 = p = 4k + 1$, with p prime, and $f(x) = 0$ (FALSE) otherwise.

To illustrate the first formulation, here is a simple arithmetic example presented by Mermin. An *odd* integer p which is a sum of two squared positive integers $p = a^2 + b^2$ is necessarily of the form $p = 4k + 1$.¹ However, not all integers of the form $p = 4k + 1$ might be expressible as $p = a^2 + b^2$: perfect squares of odd integers, are among them.² A fairly elementary theorem of number theory guarantees that if $p = 4k + 1$ is *prime*, then there is a *unique* way of writing it as $p = a^2 + b^2$. Examples, with small numbers: $5 = 4 + 1$, $13 = 9 + 4$, $17 = 16 + 1$, $29 = 25 + 4$, $37 = 36 + 1$, $41 = 25 + 16$, $53 = 49 + 4$, $61 = 36 + 25$, and so on.

¹First, a and b cannot be both odd, as otherwise the sum of their squares is even. Next, suppose $a = (2l + 1)$ and $b = 2m$. Then $p = a^2 + b^2 = 4(l^2 + l + m^2) + 1 = 4k + 1$.

²For instance, a square of a single odd integer, $p = (2k + 1)^2$, is such that $p = 4k(k + 1) + 1$.

Now suppose you have a prime p of the form $4k + 1$ which is very very large, say $p < 2N^2 = 2^{2n+1}$, with $N = 2^n$ very large. Then, a simple-minded-approach of looking for the a and b such that $p = a^2 + b^2$ would be to loop over all integers $x = 1 \cdots N = 2^n$, and see if $b = \sqrt{p - x^2}$ is integer or not. If so, you are done. As an alternative route, select a random x repeatedly, and “hope for the best”. Figure 4.1 illustrate a “black-box” routine, already in the format proper for a QC call, which would restitute the value $f(x)$ according to the strategy described. As already discussed, alternative name for such a “black-box” is “*oracle*”: you can invoke it to get answers, but there is no way of learning “how and why”. Used “classically”, as a classical computer routine, this would imply checking an exponentially large number $\sim N$ of integers.

And here I cannot refrain from quoting two paragraphs from Mermin’s book, because his prose is perfect and should not contaminated by a paraphrase in my broken English.

Mathematically well-informed friends tell me that for this particular example there are ways to proceed with a classical computer that are much more efficient than random testing, but the quantum algorithm to be described below enables even mathematical ignoramuses, equipped with a quantum computer, to do better than random testing by a factor of $1/\sqrt{N}$. And Grover’s algorithm will provide this speed-up on arbitrary problems.

Alternatively, the black box could contain Qbits that have been loaded with a body of data — for example alphabetically ordered names with phone numbers — and one might be looking for the name that went with a particular number. It is with this kind of application in mind that Grover’s neat trick has been called searching a database. Using as precious a resource as Qbits, however, merely to store classical information would be insanely extravagant, given our current or even our currently foreseeable ability to manufacture Qbits. Finding a unique solution — or one of a small number of solutions, as described in Section 4.3 — to a tough mathematical puzzle seems a more promising application.

N. David Mermin, *Quantum Computer Science*, Chapter 4

I now move to explaining how you would use the “black-box” encoding f in a smart way, discovered by Grover [20].

4.1. The Grover iteration

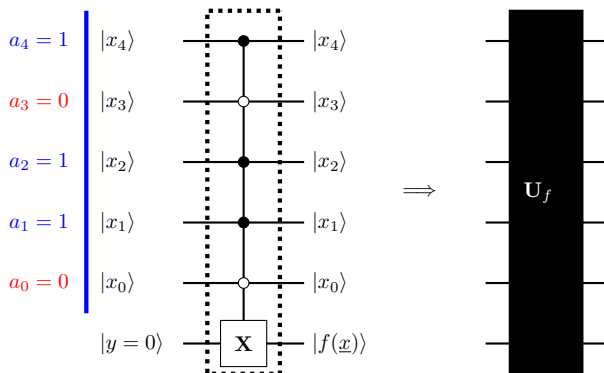


Figure 4.2: Left: A “Quantum Circuit” which would output $f(a) = 1$ and $f(x) = 0$ for all other x , here with $n = 5$ bits and $\underline{a} = (1, 0, 1, 1, 0)$. With addition of a \mathbf{X} on the output register (lower line) to the right of the first \mathbf{X} , it becomes a circuit for the second “incarnation” of the Grover search: finding a minimum $f(a) = 0$ in an otherwise totally flat “energy landscape” $f(x) = 1$. Needless to say, a circuit like this is just for illustration: to design it, I have to know the solution a .

The Bernstein-Vazirani (BV) example could be solved by devising a transformation, with appropriate gates, such that the output of the “oracle” would give you directly the a you are searching. Unfortunately, such an approach does not work here. Imagine that you would be able to “look inside” the black-box, discovering that it looks like depicted in Fig. 4.2. What would you do with such a fully-controlled gate? Nothing useful. Needless to say, to “*design*” such a circuit, you would have to know the solution beforehand. So, let us abandon the idea of inventing tricks similar to the BV case.

The only useful transformation we can apply is that of working with superpositions of the ancilla (output) register, more precisely start with $\mathbf{H}|1\rangle$ instead of $|y=0\rangle$. Indeed, as already discussed in the context of the Deutsch's problem, see Eq. (3.74), this would give:

$$\begin{aligned} \mathbf{U}_f|x\rangle_n \otimes (\mathbf{H}|1\rangle) &= \frac{1}{\sqrt{2}} \left(\mathbf{U}_f|x\rangle_n \otimes |0\rangle - \mathbf{U}_f|x\rangle_n \otimes |1\rangle \right) = \frac{1}{\sqrt{2}} \left(|x\rangle_n \otimes |f(x)\rangle - |x\rangle_n \otimes |\overline{f(x)}\rangle \right) \\ &= (-1)^{f(x)}|x\rangle_n \otimes (\mathbf{H}|1\rangle), \end{aligned} \quad (4.2)$$

as you can readily verify for both formulations of the problem, standard and golf-yard. So, upon postulating that the ancilla register is set forever in $\mathbf{H}|1\rangle$ you can completely neglect it and consider only the restriction \mathbf{V} of \mathbf{U}_f to the physical register:

$$\mathbf{V}|x\rangle_n = (-1)^{f(x)}|x\rangle_n, \quad (4.3)$$

a clear *unitary* operator that encodes the value of the function into a *sign* multiplying the computational basis states. One can write an explicit “formal” expression for \mathbf{V} as follows:

$$\text{Standard: } \mathbf{V} = \mathbf{1} - 2|a\rangle_n\langle a| \quad \text{Golf-yard minimum: } \mathbf{V} = -\mathbf{1} + 2|a\rangle_n\langle a|. \quad (4.4)$$

Grover's idea was to invent a sort of “kinetic energy” (unitary) operator \mathbf{K} which would allow exploring appropriately the Hilbert space of the problem. For that, he introduced the following:

$$\text{Standard: } \mathbf{K} = 2|+\rangle_n\langle +| - \mathbf{1} \quad \text{Golf-yard minimum: } \mathbf{K} = \mathbf{1} - 2|+\rangle_n\langle +|. \quad (4.5)$$

Notice that the two minus signs cancel, and the product \mathbf{KV} is identical in the two cases.



Universality of \mathbf{K} . It is worth stressing, once again, that you should regard \mathbf{V} as an “unknown” unitary depending on the projector operator $\hat{P}_a = |a\rangle\langle a|$ on an “unknown” state $|a\rangle$. On the contrary, \mathbf{K} is totally universal: it does not know anything about the unknown state $|a\rangle$.

The initial state is $|\psi_0\rangle = |+\rangle_n$, which is very simple to prepare: $|\psi_0\rangle = |+\rangle_n = \mathbf{H}^{\otimes n}|0\rangle_n$.

Next, let us see what we get by “applying the oracle” \mathbf{V} . We do that calculation in the “golf-yard” case. Recalling that, no matter what $|a\rangle_n$ is, we have that $\langle a|+\rangle_n = \frac{1}{\sqrt{N}}$, we get:

$$\mathbf{V}|\psi_0\rangle = \left(-\mathbf{1} + 2|a\rangle_n\langle a| \right) |+\rangle_n = \frac{2}{\sqrt{N}}|a\rangle_n - |+\rangle_n.$$

Now we apply \mathbf{K} and we get:

$$|\psi_1\rangle_n = \mathbf{KV}|\psi_0\rangle = \left(\mathbf{1} - 2|+\rangle_n\langle +| \right) \left(\frac{2}{\sqrt{N}}|a\rangle_n - |+\rangle_n \right) = \frac{2}{\sqrt{N}}|a\rangle_n + \left(1 - \frac{4}{N} \right) |+\rangle_n.$$

As you see, we keep obtaining combinations of $|a\rangle_n$ and $|+\rangle_n$, which are however *not orthogonal*, since $\langle a|+\rangle_n = \frac{1}{\sqrt{N}}$. The goal would be to apply \mathbf{KV} repeatedly:

$$|\psi_q\rangle_n = \mathbf{KV}|\psi_{q-1}\rangle_n = (\mathbf{KV})^q|\psi_0\rangle_n \quad (\text{Grover algorithm}). \quad (4.6)$$

To make the algebra simpler, we better work in a basis of orthogonal states $\{|a\rangle_n, |a^\perp\rangle_n\}$, where the unitary matrix \mathbf{KV} must look like a 2×2 *rotation matrix*, thus making further applications of \mathbf{KV} completely straightforward. To get $|a^\perp\rangle_n$ we use a Gram-Schmidt orthogonalisation to subtract to $|+\rangle_n$ its component along $|a\rangle_n$:

$$|a^\perp\rangle_n = \alpha \left(|+\rangle_n - |a\rangle_n\langle a|+\rangle_n \right),$$

with $\alpha = \pm\sqrt{\frac{N}{N-1}}$ to normalise the state. We get: ³

$$|a^\perp\rangle_n = \frac{1}{\sqrt{N-1}}|a\rangle_n - \sqrt{\frac{N}{N-1}}|+\rangle_n \implies |+\rangle_n = \frac{1}{\sqrt{N}}|a\rangle_n - \sqrt{\frac{N-1}{N}}|a^\perp\rangle_n. \quad (4.7)$$

Very simple algebra is needed to show that the matrix representing \mathbf{KV} in the basis $\{|a\rangle_n, |a^\perp\rangle_n\}$ is a 2×2 orthogonal matrix: ⁴

$$\mathbf{KV} \implies \mathbf{R}_{\theta_R} = \begin{pmatrix} 1 - \frac{2}{N} & -2\frac{\sqrt{N-1}}{N} \\ 2\frac{\sqrt{N-1}}{N} & 1 - \frac{2}{N} \end{pmatrix}, \quad (4.8)$$

where the rotation angle is

$$\theta_R = \arcsin 2\frac{\sqrt{N-1}}{N} \approx \frac{2}{\sqrt{N}}. \quad (4.9)$$

The approximation in the previous equation is entirely appropriate in the limit of a very large $N = 2^n$.

Exercise 4.1. Verify the algebra behind Eq. (4.8), by applying \mathbf{KV} to $|a\rangle_n$ and to $|a^\perp\rangle_n$.

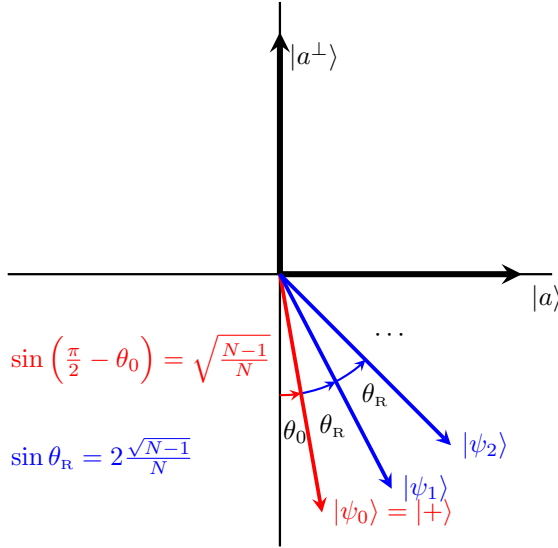


Figure 4.3: The Grover iteration in the plane $|a\rangle - |a^\perp\rangle$ (both unknown, beforehand) where the iterations induced by the unitary operator \mathbf{KV} induce rotations. The starting point $|\psi_0\rangle = |+\rangle_n$ is nearly orthogonal to the wanted $|a\rangle$. Each application of \mathbf{KV} rotates the vector by an angle $\theta_R \approx 2/\sqrt{N}$. Hence of order $(\pi/4)\sqrt{N}$ applications of \mathbf{KV} are needed to rotate the initial vector by nearly $\frac{\pi}{2}$.

The algorithm in Eq. (4.6), consisting of repeated applications of \mathbf{KV} to the initial state $|\psi_0\rangle = |+\rangle_n$ is illustrated in Fig. 4.3. The initial state, as per Eq. (4.7), forms an angle θ_0 with the $-|a^\perp\rangle_n$ direction, where:

$$\sin\left(\frac{\pi}{2} - \theta_0\right) = \sqrt{\frac{N-1}{N}} \implies \theta_0 = \arcsin \frac{1}{\sqrt{N}} \approx \frac{1}{\sqrt{N}}. \quad (4.10)$$

The number N_{Grover} of applications of the operator \mathbf{KV} to the initial state that one needs, in order to come as close as possible to a rotation by $\frac{\pi}{2} - \theta_0$, is:

$$N_{\text{Grover}} = \text{nint}\left(\frac{\frac{\pi}{2} - \theta_0}{\theta_R}\right) \approx \text{nint}\left(\frac{\pi}{4}\sqrt{N}\right). \quad (4.11)$$

³We select $\alpha = -\sqrt{\frac{N}{N-1}}$ so that the initial state $|+\rangle_n$ is nearly opposite to $|a^\perp\rangle_n$ only for the purpose of drawing Fig. 4.3.

⁴Show that:

$$\begin{aligned} \mathbf{KV}|a\rangle_n &= \left(1 - \frac{2}{N}\right)|a\rangle_n + 2\frac{\sqrt{N-1}}{N}|a^\perp\rangle_n \\ \mathbf{KV}|a^\perp\rangle_n &= -2\frac{\sqrt{N-1}}{N}|a\rangle_n + \left(1 - \frac{2}{N}\right)|a^\perp\rangle_n. \end{aligned}$$

Here, once again, the final approximation applies in the limit of a large N , where $\theta_0 \approx \frac{1}{\sqrt{N}}$ can be neglected and $\theta_r \approx \frac{2}{\sqrt{N}}$. Remarkably: $O(\sqrt{N})$ applications of \mathbf{KV} — “call of the oracle” \mathbf{V} , followed by the universal kinetic term \mathbf{K} — are needed to arrive *very close* to the wanted unknown state $|a\rangle_n$.

i **Very close?** Notice that, in general, $|\psi_q\rangle_n$ with $q = N_{\text{Grover}}$ does not coincide precisely with $|a\rangle_n$. However, by measuring the final state on the computational basis, the probability of getting $|a\rangle_n$ is overwhelmingly higher than that of any other computational state. As we will see later also for the Shor’s algorithm, Quantum Computation algorithms are often *probabilistic*.

Exercise 4.2. Consider the Grover iteration for $n = 2$ ($N = 4$). Show that the initial angle with $-|a^\perp\rangle$ is $\theta_0 = \pi/6$ and that a single application of \mathbf{KV} is enough to reach $|a\rangle$ exactly.

4.2. How to construct the kinetic term \mathbf{K}

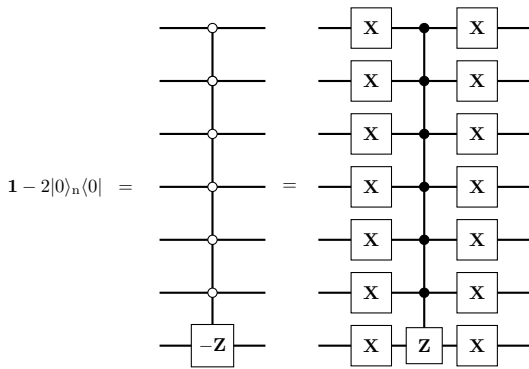


Figure 4.4: The circuit for $\mathbf{1} - 2|0\rangle_n\langle 0|$ when $n = 7$. When the controls are all in $|0\rangle$, the lower $-\mathbf{Z}_0$ changes the sign of state $|0\rangle_0$; otherwise, it acts like an identity. The form on the right, with addition of \mathbf{X} gates to the left and right of each control bit transforms the controls to the standard case (action when the control is $|1\rangle$), and \mathbf{Z}_0 into $-\mathbf{Z}_0$ in the lower line (QBit-0), since $\mathbf{XZ}\mathbf{X} = -\mathbf{Z}$.

Recall that $\mathbf{K} = \mathbf{1} - 2|+\rangle_n\langle +|$ and that an Hadamard rotates Z to X ($\mathbf{HZH} = \mathbf{X}$, and viceversa, since $\mathbf{H}^2 = \mathbf{1}$) hence $\mathbf{H}|0\rangle = |+\rangle$. This implies that:

$$\mathbf{K} = \mathbf{1} - 2|+\rangle_n\langle +| = \mathbf{H}^{\otimes n} (\mathbf{1} - 2|0\rangle_n\langle 0|) \mathbf{H}^{\otimes n} . \tag{4.12}$$

So, it is enough the construct a circuit for $\mathbf{1} - 2|0\rangle_n\langle 0|$ — incidentally, this coincides with a circuit for $-\mathbf{V}$ when $a = 0$ — see also Fig. 4.2 — which is as general a number as any other a if you do not assume to know it — to construct \mathbf{K} .

To construct $\mathbf{1} - 2|0\rangle_n\langle 0|$ you would need the fully-controlled- \mathbf{Z}_0 gate shown in the middle-section of the right part of Fig. (4.4). Now, fully-controlled gates are bad, because they are difficult to manufacture in any hardware. What people know how to fabricate is, usually, single-Qbit gates, control-NOT and doubly-controlled-NOT (Toffoli gates). With the use of Hadamards, you can therefore construct, for instance, doubly-controlled- \mathbf{Z} gates. So the idea now is how to manage, at the cost of adding extra gates, the fully-controlled- \mathbf{Z} .

Step 1) The first step is explained in Fig. 4.5: with the addition of a *single ancillary* Qbit (top line), and paying 4 times more gates, we reduce ourselves to considering nearly half-controlled-gates. The circuit equality in Fig. 4.5 can be checked on the computational basis. For that purpose, observe that if *any* of the control Qbits on the left (QBits 1 to 6 in Fig. 4.5) is in state $|0\rangle$, then, on the right, either the \mathbf{Z} gate is transformed into an identity (for control-QBit 1 to 3) or the \mathbf{X} gate is transformed into an identity (for control-QBit 4 to 6). In both cases, the remaining gates, even if present, act twice, hence they are equivalent to an identity, as you can easily verify. On the contrary, if *all* the control

QBits (1 to 6) are in state $|1\rangle$, then all the \mathbf{Z} and \mathbf{X} are present, but the presence of controls on the ancilla (QBit-7) intertwined with \mathbf{X} is such that the lower \mathbf{Z} gate acts *only once*, as explained in Fig. 4.6, hence reproducing the correct behaviour of the left circuit.

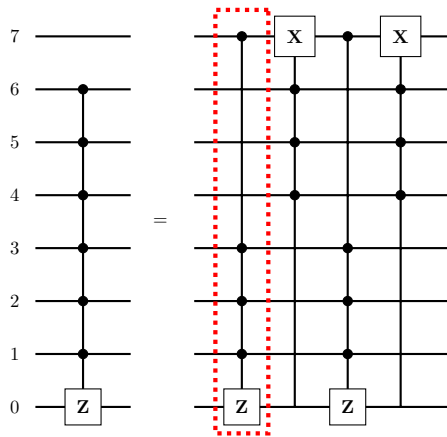


Figure 4.5: A fully-controlled- \mathbf{Z} is transformed, by addition of a single ancillary Qbit (Qbit-7, top line), into a circuit with 4 times more gates but with (nearly) half control-Qbits. To check the equality of the two circuits, it is enough to verify it on the computational basis states. For this, observe that if *any* of the control Qbits on the left (QBits 1 to 6) is in state $|0\rangle$, then, on the right, either the \mathbf{Z} gate is transformed into an identity (for control-QBit 1 to 3) or the \mathbf{X} gate is transformed into an identity (for control-QBit 4 to 6). In both cases, the remaining gates, even if present, act twice, hence they are equivalent to an identity. If all the control QBits (1 to 6) are in state $|1\rangle$, then all the \mathbf{Z} and \mathbf{X} are present, but the presence of controls on the ancilla (QBit-7) intertwined with \mathbf{X} is such that the lower \mathbf{Z} gate acts *only once*, see Fig. 4.6, reproducing the correct behaviour of the left circuit. The part of the circuit highlighted by the dotted rectangle is further analysed in Fig. 4.7 below.

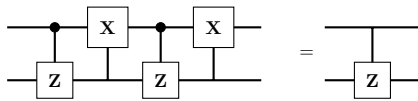


Figure 4.6: A simple identity useful in the analysis of Fig. 4.5. The two \mathbf{X} gates on the top line always leave the initial computational state of the line unchanged. If the initial state of the top line is $|0\rangle$, then the first control- \mathbf{Z} does not act, while the second acts (because of the \mathbf{X} on its immediate left flipping the QBit to $|1\rangle$). Viceversa, if the initial state of the top line is $|1\rangle$, then the first control- \mathbf{Z} acts, while the second doesn't.

Step 2) Next, consider each of the nearly half-controlled \mathbf{Z} or \mathbf{X} gates present in Fig. 4.5, for instance the half-controlled \mathbf{Z} gates highlighted by the dashed rectangle. There are lines that do nothing,⁵ and can be used as ancillary Qbits to set-up further transformations, as long as their computational state is unchanged.

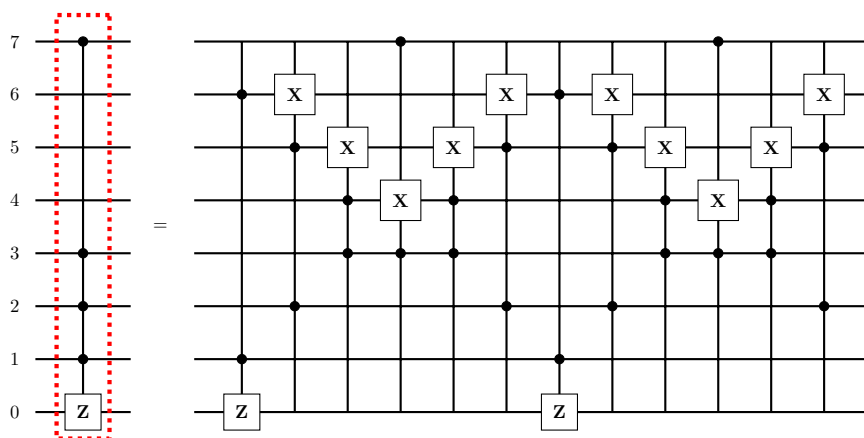


Figure 4.7: A partially-controlled \mathbf{Z} , with enough free Qbit lines (playing the role of ancillas) can be transformed, by adding extra gates, into a circuit which uses doubly-controlled \mathbf{Z} or \mathbf{X} . A similar trick can be used for partially-controlled \mathbf{X} gates, since $\mathbf{X} = \mathbf{H}\mathbf{Z}\mathbf{H}$.

Figure 4.7 indeed shows that, with a sufficient number of “lines not involved in controls”, and with a further increase in the number of gates, one can work with at-most **doubly-controlled gates**.

⁵The idea is to have enough of them ($\geq m - 3$, if m is the total number of the other lines) so as to set-up further transformations, as explained in Fig. 4.7.

Verifying the circuit identity shown in Fig. 4.7 is boring but straightforward. Indeed, observe that if any of the control Qbits on the left is in state $|0\rangle$, then two of doubly-controlled gates involving that Qbit are transformed into identities, while the remaining doubly-controlled gates appear *squared*, hence produce an overall identity.⁶ If, on the other hand, *all* the control QBits are in state $|1\rangle$, then all the gates are present, and the circuit can be rewritten as shown in Fig. 4.8.

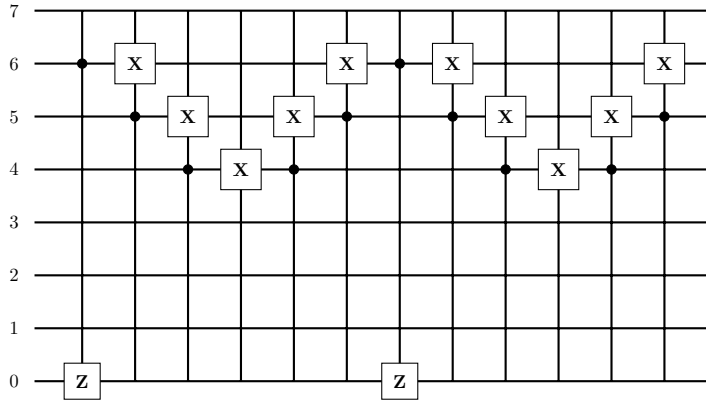


Figure 4.8: The circuit on the right of Fig. 4.7 when all control-Qbits 1, 2, 3, 7 are in state $|1\rangle$, leaving all the gates and remaining controls on the free lines 4, 5, 6.

The crucial identity to be used is now shown in Fig. 4.9. By using this, you transform the circuit in Fig. 4.8 into one that, by further using the identity in Fig. 4.6, is totally equivalent to the action of a *single Z* gate on the lower line.

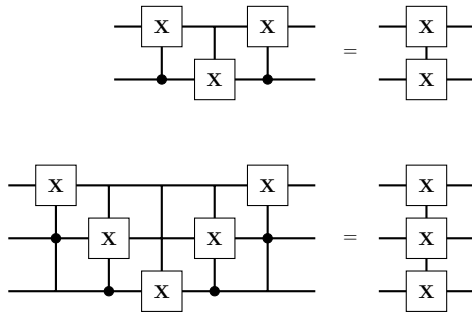


Figure 4.9: Two simple identities useful in the analysis of Fig. 4.8. The top identity, used repeatedly twice, produces the lower identity, which in the end simplifies the circuit in Fig. 4.8 to something which, using the identity in Fig. 4.6 is totally equivalent to the action of a *single Z* gate on the lower line.

Exercise 4.3. Verify the circuit identities shown in Figs. 4.4-4.9.

4.3. Generalisation to the case of several solutions

The algorithm generalises easily to the case where $M > 1$ solutions exist:

$$\text{Golf-yard minimum: } f(x) = \begin{cases} 0 & \text{for } x = a^{(1)}, \dots, a^{(M)} \\ 1 & \text{otherwise} \end{cases}, \quad (4.13)$$

The idea is to substitute the role of $|a\rangle_n$ in the previous algorithm, with the symmetric combination of the searched states:

$$|a\rangle_n \quad \longrightarrow \quad |\psi_{\text{sym}}\rangle_n = \frac{1}{\sqrt{M}} \sum_{m=1}^M |a^{(m)}\rangle_n. \quad (4.14)$$

⁶Observe that for the square of the Toffoli gate we have $C_{21,0}^2 = 1$, and similarly for doubly-controlled **Z** gates.

Exercise 4.4. Verify how M enters in the Grover algorithm. To do that, calculate the overlap $\langle \psi_{\text{sym}} | + \rangle_n$, and the orthogonal combination $|\psi_{\text{sym}}^\perp\rangle_n$. The outcome should be that the number of iterations decreases as \sqrt{M} .



Unknown M . Notice that if the number of solutions M is unknown, the optimal number of applications of the algorithm is not completely determined. For strategies to mitigate this drawback, read Sec. 4.4 in Mermin's book.

4.4. Connection to p-spin models and to QAOA

Consider a (classical) Ising model with Hamiltonian:

$$\hat{H}_z^{(2)} = -nJ(\hat{m}_z^2 - 1) \quad \text{with} \quad \hat{m}_z = \frac{1}{n} \sum_{j=1}^n \hat{\sigma}_j^z.$$

The classical ground states, with energy zero, would be associated to two states, with all spins up $|\uparrow \cdots \uparrow\rangle_n = |0 \cdots 0\rangle_n$, or down $|\downarrow \cdots \downarrow\rangle_n = |1 \cdots 1\rangle_n$. All other states, with intermediate magnetisation $-1 < m_z < 1$, have a positive energy, and are degenerate, up to the largest-energy states, with $m_z = 0$ and energy $E = nJ$, which are massively degenerate: $\binom{n}{n/2}$. Upon expanding the square of the average magnetisation \hat{m}_z , one would obtain:

$$\hat{H}_z^{(2)} = -\frac{2J}{n} \sum_{j_1 < j_2} \hat{\sigma}_{j_1}^z \hat{\sigma}_{j_2}^z + (n-1)J,$$

hence an all-to-all ferromagnetic interaction.

A generalisation of this model is a p-spin all-to-all Ising ferromagnet, with Hamiltonian:

$$\hat{H}_z^{(p)} = -nJ(\hat{m}_z^p - 1).$$

Observe that, for odd $p = 2k + 1$, the classical ground state $|\uparrow \cdots \uparrow\rangle_n = |0 \cdots 0\rangle_n$ would be *non-degenerate*.⁷



The Grover limit. Observe also that, in the limit $p = 2k + 1 \rightarrow \infty$ — since $|x|^{2k+1} \rightarrow 0$ for $|x| < 1$ — all other states different from $|h_1 \cdots h_n\rangle$ end-up being degenerate, with energy $E = nJ$. Hence, setting $J = 1/n$, we get precisely the Grover limit:

$$\mathbf{V} = \lim_{k \rightarrow \infty} e^{-i\pi \hat{H}_z^{(2k+1)}}. \quad (4.16)$$

An interesting branch of the story has to do with alternative techniques for constructing quantum states by repeatedly applying unitaries to a simple initial state. One such technique is known as *Quantum Approximate Optimization Algorithm* (QAOA) [21]. Its connection with Quantum Annealing (QA), *alias* Adiabatic Quantum Computation (AQC), has been recently studied in our group at

⁷Interestingly, any *arbitrary spin-configuration* could be promoted to be the unique classical ground state of such a model. It is enough to re-define the Hamiltonian, for any wanted target ground state $|h_1 \cdots h_n\rangle$ with $h_j = \pm 1$, as follows:

$$\hat{H}_z^{(p)} = -nJ \left(\left(\frac{1}{n} \sum_{j=1}^n h_j \hat{\sigma}_j^z \right)^p - 1 \right). \quad (4.15)$$

SISSA [22]. In these approaches, the kinetic energy term typically takes a much simpler form: a simple transverse field term, acting independently on each Qbit. The unitary that one would apply is therefore of the form:

$$e^{-i\gamma\hat{H}_x} \quad \text{with} \quad \hat{H}_x = \sum_{j=1}^n \hat{\sigma}_j^x .$$

In QAOA one writes an *Ansatz* for the states, written iteratively as:

$$|\psi_m\rangle = e^{-i\gamma_m^x \hat{H}_x} e^{-i\gamma_m^z \hat{H}_z} |\psi_{m-1}\rangle \quad \text{with} \quad |\psi_0\rangle = |+\rangle_n . \quad (4.17)$$

The *Ansatz* $|\psi_P(\gamma)\rangle$, assuming we apply the algorithm from $m = 1$ to $m = P$, depends on $2P$ parameters

$$\gamma = (\gamma_1^z, \dots, \gamma_P^z, \gamma_1^x, \dots, \gamma_P^x) ,$$

and reads, explicitly, as follows:

$$|\psi_P(\gamma)\rangle = e^{-i\gamma_P^x \hat{H}_x} e^{-i\gamma_P^z \hat{H}_z} \dots e^{-i\gamma_2^x \hat{H}_x} e^{-i\gamma_2^z \hat{H}_z} e^{-i\gamma_1^x \hat{H}_x} e^{-i\gamma_1^z \hat{H}_z} |+\rangle_n . \quad (4.18)$$

In principle, one should optimise the choice of the parameters, for instance by a classical minimisation algorithm: the variational principle of Quantum Mechanics is the basic driving principle behind such a choice.

Ref. [23] has studied the application of QAOA to the Grover problem. Apparently, taking $\gamma_p^z = \pi$ and $\gamma_p^x = \pi/n$ gives the optimal QAOA *Ansatz*, and P , the number of iterations of the algorithm, would scale in a near-optimal fashion, as $P \sim \frac{\pi}{2\sqrt{2}} \sqrt{N}$. See also Ref. [24] for an application of QAOA to the problem of a fully-connected p-spin Ising ferromagnet.

5. Quantum Fourier Transform

I present here the Quantum Fourier Transform algorithm introduced by P. Shor in 1994, see Ref. [9] for a longer account. The presentation is largely based on the book by Mermin [1], with a few applications, notably to the phase estimation protocol, Sec. 5.4, and to eigenvalue determination, Sec. 5.5, for which I have consulted Ref. [2].

Let us recall the Discrete Fourier Transform (DFT), also known to you from tight-binding problems with periodic boundary conditions in many condensed matter courses.

i

Discrete Fourier Transform. Given an array of N complex numbers u_j , with $j = 0 \cdots N - 1$ — hence an element in \mathbb{C}^N — we define its DFT as the following array of \mathbb{C}^N :

$$\tilde{u}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-2\pi i k j / N} u_j \quad \text{with} \quad k = 0 \cdots N - 1. \quad (5.1)$$

If $N = N_{\text{Hilbert}} = 2^n$, as appropriate for the Hilbert space dimension for n Qbits, we will redefine $j \rightarrow x = 0 \cdots 2^n - 1$ so as to adhere to the standard state representation used so far whereby we identify n -bits binary strings $\underline{x} = (x_{n-1}, \cdots, x_0) \in \{0, 1\}^n$ with their corresponding integer:

$$\underline{x} \longleftrightarrow x = \sum_{j=0}^{n-1} x_j 2^j.$$

We will therefore rewrite the DFT and its inverse for $N = 2^n$ as:

$$\left\{ \begin{array}{l} \tilde{u}_k = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{-2\pi i k x / 2^n} u_x \quad \text{with} \quad k = 0 \cdots 2^n - 1 \\ u_x = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k x / 2^n} \tilde{u}_k \quad \text{with} \quad x = 0 \cdots 2^n - 1 \end{array} \right. . \quad (5.2)$$

i

Fast Fourier Transform (FFT). In a simple minded approach, the DFT would seem to require $O(N^2) = O(2^{2n})$ operations. But a fantastic algorithm, the Fast Fourier Transform, was (re)-discovered in 1965 by Cooley and Tukey. ^a FFT requires $O(N \log N) = O(n 2^n)$ operations, and has been indicated as one of the Top 10 Algorithms of the 20th century by IEEE, with immensely useful applications in many fields. To mention one, mp3 compression of music would not be possible without FFT: recall that a standard audio-CD uses a digital sampling rate of 44 kHz, and a 16-bit resolution. With a mere $N = 2^{16} = 65536$ you would appreciate the enormous difference between $O(N^2)$ and $O(N \log N)$. For a description of the crucial idea behind FFT, see *Numerical Recipes'* book.

^aApparently, Gauss already used it in 1805 in some unpublished astronomical work. Danielson & Lanczos, in 1942, also introduced a form of FFT. And many others over the decades.

Now we move to our usual QC setting in which we consider n Qbits and the computation basis $|x\rangle_n = |x_{n-1}\rangle \cdots |x_0\rangle$. We will need some dummy-labelled computational basis states — in no way different from the $|x\rangle_n$ — which we will denote by $|k\rangle_n = |k_{n-1}\rangle \cdots |k_0\rangle$, where as usual $k = \sum_{j=0}^{n-1} k_j 2^j \longleftrightarrow \underline{k} = (k_{n-1}, \dots, k_0) \in \{0, 1\}^n$.

❶

Quantum Fourier Transform (QFT). We define the Quantum Fourier Transform as:

$$\mathbf{U}_{\text{QFT}}|k\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} |x\rangle_n, \quad (5.3)$$

where, we stress, $|k\rangle_n = |k_{n-1}\rangle \cdots |k_0\rangle$ represent standard computational basis states. We will soon show that \mathbf{U}_{QFT} is a *unitary operator* in the n -Qbit Hilbert space. The great discovery by P. Shor [9] is that one can execute it on a QC with $O(n^2)$ quantum gates.

Before entering into details, let us remark a very close connection with DFT. Suppose that you consider a superposition state $|\psi_{\tilde{u}}\rangle$:

$$|\psi_{\tilde{u}}\rangle_n = \sum_{k=0}^{2^n-1} \tilde{u}_k |k\rangle_n, \quad (5.4)$$

and you apply \mathbf{U}_{QFT} to it. You get:

$$\begin{aligned} \mathbf{U}_{\text{QFT}}|\psi_{\tilde{u}}\rangle_n &= \sum_{k=0}^{2^n-1} \tilde{u}_k \mathbf{U}_{\text{QFT}}|k\rangle_n \\ &= \sum_{k=0}^{2^n-1} \tilde{u}_k \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} |x\rangle_n = \sum_{x=0}^{2^n-1} \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \tilde{u}_k e^{2\pi i k x / 2^n} \right) |x\rangle_n \\ &= \sum_{x=0}^{2^n-1} u_x |x\rangle_n \stackrel{\text{def}}{=} |\psi_u\rangle_n \end{aligned} \quad (5.5)$$

where the coefficients are given by u_x , the inverse-DFT of \tilde{u}_k .

❷

Warning: The analogy with our physicists way of picturing the back-and-forth switch from “real” to “momentum” space should not be pushed too far, as Mermin correctly warns its physicist reader. As he puts it: “*The number x is the integer represented by the state $|x\rangle_n$; it is not the position of anything. Changing x to $x + 1$ induces an arithmetically natural but physical quite unnatural transformation on the computational basis-states [...].*”^a It bears no resemblance to anything that could be associated with a spatial translation in the physical space of Qbits. Granted all that, still, the analogy will prove useful in the following, as you will see.

^aNotice that indeed $|7\rangle_5 = |00111\rangle$ while $|8\rangle_5 = |01000\rangle$, involving a quite non-local flips of “spins” in the computational basis.

One of the goals of this chapter is to show that there is a Quantum Algorithm, due to Shor, for executing \mathbf{U}_{QFT} on a QC with $O(n^2)$ quantum gates, compared to the $O(n^{2^n})$ operations of FFT. Be aware, however, that executing efficiently \mathbf{U}_{QFT} on a QC does not mean that we can use a QC to calculate the FFT \tilde{u}_k of a signal u_x — or viceversa — in $O(n^2)$ operations! Recall that, as discussed several times, a state $|\psi_u\rangle$ is an *abstract object*, a kind of “black box” from the practical viewpoint: you are not allowed to learn all its coefficients u_x . When you perform a *measurement* on it, QM, with von Neumann, teaches us that you provoke a *collapse* of the state into one of its components $|x\rangle_n$, with a probability $|u_x|^2$. So, indeed you do not “learn” the inverse-DFT u_x of the input state,

but rather a *random* single-component piece of it. So, QFT is *not a replacement* for FFT in practical applications. But, there are cases in which a smart use of QFT within a QC framework might make great use of superposition and interference, with an incredible speedup with respect to anything a classical computer could do.

Relationship with Hadamards. I also want to remark the connection of QFT with the Hadamard transformation. For $n = 1$ Qbit, the two coincide:

$$\mathbf{H}|k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^k|1\rangle) = \frac{1}{\sqrt{2}} \sum_{x=0}^1 e^{i\pi x k} |x\rangle. \quad (5.6)$$

For n -Qbits, the two are very different. The n -Qbit Hadamard is a tensor product of *uncorrelated* single-Qbit unitaries:

$$\begin{aligned} \mathbf{H}^{\otimes n}|k\rangle_n &= \frac{1}{\sqrt{2^n}} \sum_{x_{n-1}=0}^1 \cdots \sum_{x_0=0}^1 e^{i\pi \sum_j x_j k_j} |x_{n-1}\rangle \cdots |x_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{i\pi x \cdot k} |x\rangle_n \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle_{n-1} + e^{\pi i k_{n-1}} |1\rangle_{n-1} \right) \otimes \cdots \otimes \left(|0\rangle_0 + e^{\pi i k_0} |1\rangle_0 \right), \end{aligned} \quad (5.7)$$

where $e^{i\pi \sum_j x_j k_j} = e^{i\pi x \cdot k}$, the bitwise (mod 2) product of two integers we have encountered before in the Bernstein-Vazirani problem. On the contrary, $\mathbf{U}_{\text{QFT}}|k\rangle_n$ is a more complex ¹ superposition of the various Qbits which, as we shall soon see, can be remarkably decomposed into 1- and 2-Qbit unitaries.

A closer look at the QFT state. Consider again our expression for the QFT state:

$$\begin{aligned} \mathbf{U}_{\text{QFT}}|k\rangle_n &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} |x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} |x_{n-1}\rangle \cdots |x_0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x_{n-1}=0}^1 \cdots \sum_{x_0=0}^1 e^{2\pi i k (x_0/2^n + x_1/2^{n-1} + \cdots + x_{n-1}/2)} |x_{n-1}\rangle \cdots |x_0\rangle \\ &= \frac{1}{\sqrt{2^n}} \left(\sum_{x_{n-1}=0}^1 e^{2\pi i k x_{n-1}/2} |x_{n-1}\rangle \right) \otimes \cdots \otimes \left(\sum_{x_0=0}^1 e^{2\pi i k x_0/2^n} |x_0\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle_{n-1} + e^{\pi i k} |1\rangle_{n-1} \right) \otimes \cdots \otimes \left(|0\rangle_0 + e^{\pi i k / 2^{n-1}} |1\rangle_0 \right) \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle_{n-1} + e^{\pi i k_0} |1\rangle_{n-1} \right) \otimes \cdots \otimes \left(|0\rangle_0 + e^{\pi i (k_{n-1} + \cdots + \frac{1}{2^{n-2}} k_1 + \frac{1}{2^{n-1}} k_0)} |1\rangle_0 \right), \end{aligned} \quad (5.8)$$

where, in the last step, we used $k = 2^{n-1}k_{n-1} + \cdots + 2k_1 + k_0$ and dropped multiples of $2\pi i$.

1 Faster oscillations and more-significant bits. The most significant bits in a binary bit string are those to the *left*. Observe that here the most significant Qbit, the $(n-1)$ -th, oscillates the most as k increases, changing sign for k odd/even, since it depends on the least significant bit k_0 of k . On the contrary, the least significant Qbit-0 has a phase that does only a “single turn” of the complex unit circle as k grows from $k = 0$ towards $2^n - 1$.

¹Technically, diagonal Qbit interactions, hence controlled-phase gates, are involved. In principle, these controlled-phase gates could be realised without employing cNOTs, the basic entanglement creators, but this depends on the hardware implementation.



Interactions and entanglement. By looking at the form of the state in Eq. (5.8), you might say that this is still a *product state*.^a True, but this is so only when \mathbf{U}_{QFT} is applied to a computational basis state $|k\rangle_n$. More generally, the controlled-phase gates present in \mathbf{U}_{QFT} are due to *interactions between the qubits* and *do create entanglement* when \mathbf{U}_{QFT} is applied on more general product states, made of superpositions of computational states. Unlike the simple $\mathbf{H}^{\otimes n}|k\rangle_n$, where the phase of each single j th Qbit is governed only by the value of the corresponding k_j , here diagonal interactions bring in phase factors, in each Qbit, which depend on *all* the k_j values, in principle.

^aI must thank Pietro Torta for raising this issue.

Exercise 5.1. Show that if \mathbf{C}_{10}^θ is a phase-controlled gate, then the state $\mathbf{C}_{10}^\theta \mathbf{H}_1 |x_1\rangle \otimes |x_0\rangle$ is a product state, while $\mathbf{C}_{10}^\theta \mathbf{H}_1 |x_1\rangle \otimes |\psi_0\rangle$, with $|\psi_0\rangle = z_0|0\rangle + z_1|1\rangle$, is entangled.



Fourier basis. You might regards the state

$$|\tilde{\psi}_k\rangle_n \stackrel{\text{def}}{=} \mathbf{U}_{\text{QFT}} |k\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} |x\rangle_n \quad (5.9)$$

as the *Fourier basis state* associated to “ k ”, as long as you refrain from thinking that the $|k\rangle_n$ on the left-hand-site (LHS) are any different from dummy-labelled configurational states, conveniently denoted with k . In some sense, still, k plays the role of “momentum” in the superposition on the RHS, so that

$$|\tilde{\psi}_{k=0}\rangle_n = \mathbf{U}_{\text{QFT}} |0\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n \equiv \mathbf{H}^{\otimes n} |0\rangle_n$$

is indeed the “maximally delocalized” state in the computational basis. That Eq.(5.9) provides a legitimate change of orthogonal basis follows from the fact, which we will show in several ways, that \mathbf{U}_{QFT} is *unitary*.

Exercise 5.2. Verify that the Fourier-basis states in Eq. (5.9) form indeed an orthonormal basis set:

$$\langle \tilde{\psi}_{k'} | \tilde{\psi}_k \rangle_n = \delta_{k'k} \quad (5.10)$$

From this, you would immediately conclude that \mathbf{U}_{QFT} is unitary. Why?

Hint: Use that:

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} = \delta_{k,0} .$$

5.1. The Quantum Fourier Transform circuit

Our goal will be to express \mathbf{U}_{QFT} in a way that is *independent* of the basis used, and expressed in terms of the usual gates: in particular, as we shall see, Hadamard single-Qbits, and control-phase two-Qbit gates.

To achieve that, it is useful to introduce the following *unitary diagonal operator* \mathcal{Z} on the computation basis $|x\rangle_n$:

$$\mathcal{Z}|x\rangle_n = e^{2\pi i x / 2^n} |x\rangle_n , \quad (5.11)$$

which attaches to each state $|x\rangle_n$ the corresponding 2^n -root of unity in complex plane $e^{2\pi ix/2^n}$. The matrix representation for \mathcal{Z} on the computational basis is obviously $\text{diag}(e^{2\pi ix/2^n})$. \mathcal{Z} coincides, for $n = 1$, with the Pauli- \mathbf{Z} operator, hence the name. Clearly, for any integer k we can easily define \mathcal{Z}^k :

$$\mathcal{Z}^k|x\rangle_n = e^{2\pi ikx/2^n}|x\rangle_n, \quad (5.12)$$

with $\mathcal{Z}^{2^n} = \mathcal{Z}^0 = \mathbf{1}$.² Now observe that:

$$\mathcal{Z}^k \mathbf{H}^{\otimes n} |0\rangle_n = \mathcal{Z}^k \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi ikx/2^n} |x\rangle_n \stackrel{\text{def}}{=} \mathbf{U}_{\text{QFT}}|k\rangle_n, \quad (5.13)$$

which shows in one-shot that \mathbf{U}_{QFT} is *unitary*, since \mathcal{Z} and $\mathbf{H}^{\otimes n}$ are so.

The goal is now to represent the k -dependent operator $\mathcal{Z}^k \mathbf{H}^{\otimes n}$ in terms of a circuit with k -independent gates acting on the computational basis, in such a way that we can get an operator identity. An important identity which we will use is the following.

i

The projector on state $|1\rangle$.

If $(\mathbf{N}_1)_j = \frac{1}{2}(\mathbf{1} - \mathbf{Z})_j$ is the projector on the state $|1\rangle$ of Qbit- j , then

$$(\mathbf{N}_1)_j|x_j\rangle = x_j|x_j\rangle \quad \text{with} \quad x_j = 0, 1 \quad \implies \quad e^{i\alpha x_j}|x_j\rangle = e^{i\alpha(\mathbf{N}_1)_j}|x_j\rangle. \quad (5.14)$$

Following Mermin, let us consider the $n = 4$ -Qbit case, for clarity. Here $|x\rangle_4 = |x_3\rangle|x_2\rangle|x_1\rangle|x_0\rangle$ and $x = 8x_3 + 4x_2 + 2x_1 + x_0$, with $x_j = 0, 1$. From the fact that $\mathcal{Z}|x\rangle_4 = e^{2\pi ix/2^4}|x\rangle_4$ we deduce that:

$$\mathcal{Z}|x\rangle_4 = e^{\frac{\pi i}{8}(8(\mathbf{N}_1)_3 + 4(\mathbf{N}_1)_2 + 2(\mathbf{N}_1)_1 + (\mathbf{N}_1)_0)}|x\rangle_4, \quad (5.15)$$

which can now be regarded as an operator identity for \mathcal{Z} , since it holds for all computational basis states $|x\rangle_4$. Take now the usual binary expression for $k = 8k_3 + 4k_2 + 2k_1 + k_0$, and calculate:

$$\mathcal{Z}^k = e^{\frac{\pi i}{8}(8k_3 + 4k_2 + 2k_1 + k_0)}(8(\mathbf{N}_1)_3 + 4(\mathbf{N}_1)_2 + 2(\mathbf{N}_1)_1 + (\mathbf{N}_1)_0). \quad (5.16)$$

Since $e^{2\pi ip(\mathbf{N}_1)_j} = \mathbf{1}_j$ for any integer p , it follows that all the terms appearing in the last expression for \mathcal{Z}^k which happen to involve multiples of $2\pi i$ can be safely dropped. Keeping only the non-trivial terms we get:

$$\mathcal{Z}^k = e^{\pi i(k_0(\mathbf{N}_1)_3 + (k_1 + \frac{1}{2}k_0)(\mathbf{N}_1)_2 + (k_2 + \frac{1}{2}k_1 + \frac{1}{4}k_0)(\mathbf{N}_1)_1 + (k_3 + \frac{1}{2}k_2 + \frac{1}{4}k_1 + \frac{1}{8}k_0)(\mathbf{N}_1)_0)}, \quad (5.17)$$

where we highlighted in **blue** the terms with integer coefficients, which will soon play a particular role.

Next we need to consider $\mathcal{Z}^k \mathbf{H}^{\otimes n} |0\rangle_n$ where $n = 4$ copies of \mathbf{H} sit to the right of \mathcal{Z}^k . Notice that, for a *single* Qbit

$$e^{\pi ik\mathbf{N}_1} \mathbf{H}|0\rangle = \mathbf{H}|k\rangle, \quad (5.18)$$

an equality which is trivial for $k = 0$ and simple for $k = 1$.³ This means that we can transfer the coefficients k into the states to the right of \mathbf{H} , while \mathbf{N}_1 disappears from the left of \mathbf{H} . Consider now the **blue** terms with integer coefficients in the last expression for \mathcal{Z}^k . When their action is considered, using Eq. (5.18) for each Qbit, we get:

$$\begin{aligned} e^{\pi i(k_0(\mathbf{N}_1)_3 + k_1(\mathbf{N}_1)_2 + k_2(\mathbf{N}_1)_1 + k_3(\mathbf{N}_1)_0)} \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{H}_0 |0\rangle_3 |0\rangle_2 |0\rangle_1 |0\rangle_0 = \\ = \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{H}_0 |k_0\rangle_3 |k_1\rangle_2 |k_2\rangle_1 |k_3\rangle_0, \end{aligned} \quad (5.19)$$

²Incidentally, one can also define $\mathcal{Z}^{-k} = (\mathcal{Z}^k)^* = (\mathcal{Z}^k)^\dagger$, so that $\mathcal{Z}^{-k}\mathcal{Z}^k = \mathbf{1}$ and, more generally, $\mathcal{Z}^{k_1}\mathcal{Z}^{k_2} = \mathcal{Z}^{k_1+k_2}$.

³Indeed:

$$e^{\pi i\mathbf{N}_1} \mathbf{H}|0\rangle = e^{\pi i\mathbf{N}_1} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \mathbf{H}|1\rangle.$$

where we explicitly indicated the standard ordering of the computational basis — from right to left upon increasing the Qbit-index — to remark that the way the various k_j appear in the states is just *reversed*: k_0 goes into $|\cdot\rangle_3$, k_1 into $|\cdot\rangle_2$, k_2 into $|\cdot\rangle_1$, k_3 into $|\cdot\rangle_0$. No mystery! The operators that enforce this fact are indeed $(\mathbf{N}_1)_3$ which multiplies k_0 , $(\mathbf{N}_1)_2$ which multiplies k_1 , $(\mathbf{N}_1)_1$ which multiplies k_2 , etc. More generally, the integer coefficient terms will always have the structure $k_{n-1-j}(\mathbf{N}_1)_j$. If you revisit Eq. (5.8) you will see the connection.

We are not done yet, as we still have to consider the terms with powers of 2 in the denominator. Collecting the expressions we have so far we get:

$$\mathcal{Z}^k \mathbf{H}^{\otimes 4} |0\rangle_4 = e^{\pi i \left(\frac{1}{2} k_0 (\mathbf{N}_1)_2 + \left(\frac{1}{2} k_1 + \frac{1}{4} k_0 \right) (\mathbf{N}_1)_1 + \left(\frac{1}{2} k_2 + \frac{1}{4} k_1 + \frac{1}{8} k_0 \right) (\mathbf{N}_1)_0 \right)} \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{H}_0 |k_0\rangle_3 |k_1\rangle_2 |k_2\rangle_1 |k_3\rangle_0 .$$

Now, the $\mathbf{H}_{j'}$ commute with the $(\mathbf{N}_1)_j$ for $j' \neq j$, hence we can *move* the various $(\mathbf{N}_1)_j$ so that they appear *immediately to the left* of the corresponding \mathbf{H}_j . Omitting the colours, we have:

$$\begin{aligned} \mathbf{U}_{\text{QFT}} |k\rangle_4 &= \mathcal{Z}^k \mathbf{H}^{\otimes 4} |0\rangle_4 \\ &= \mathbf{H}_3 e^{\pi i \frac{1}{2} k_0 (\mathbf{N}_1)_2} \mathbf{H}_2 e^{\pi i \left(\frac{1}{2} k_1 + \frac{1}{4} k_0 \right) (\mathbf{N}_1)_1} \mathbf{H}_1 e^{\pi i \left(\frac{1}{2} k_2 + \frac{1}{4} k_1 + \frac{1}{8} k_0 \right) (\mathbf{N}_1)_0} \mathbf{H}_0 |k_0\rangle_3 |k_1\rangle_2 |k_2\rangle_1 |k_3\rangle_0 \end{aligned} \quad (5.20)$$

Observe now that $|k_0\rangle_3$ is eigenstate of $(\mathbf{N}_1)_3$ with eigenvalue $k_0 = 0, 1$, and k_0 appears in the exponentials safely to the *right* of \mathbf{H}_3 , which would ruin our trick. Hence, we can substitute $k_0 \rightarrow (\mathbf{N}_1)_3$ anywhere in the exponentials:

$$e^{i k_0 \mathbf{A}} |k_0\rangle_3 = e^{i (\mathbf{N}_1)_3 \mathbf{A}} |k_0\rangle_3 , \quad (5.21)$$

where \mathbf{A} is an arbitrary operator. The same trick applies to $k_1 \rightarrow (\mathbf{N}_1)_2$, and $k_2 \rightarrow (\mathbf{N}_1)_1$. As a result of that, the explicit dependence on the k_j in the exponentials is now transformed into a dependence on the *operators* $(\mathbf{N}_1)_{3-j}$. To simplify our writing, we define the following phase-control operators, symmetric under exchange $j \leftrightarrow j'$:

$$\mathbf{V}_{jj'} = e^{\pi i (\mathbf{N}_1)_j (\mathbf{N}_1)_{j'} / 2^{j-j'}} = \mathbf{V}_{j'j} \quad (5.22)$$

and finally rewrite Eq. (5.20) (adding unnecessary parentheses) as:

$$\begin{aligned} \mathcal{Z}^k \mathbf{H}^{\otimes 4} |0\rangle_4 = \mathbf{U}_{\text{QFT}} |k\rangle_4 &= \mathbf{U}_{\text{QFT}} |k_3\rangle_3 |k_2\rangle_2 |k_1\rangle_1 |k_0\rangle_0 = \\ &= \mathbf{H}_3 (\mathbf{V}_{32} \mathbf{H}_2) (\mathbf{V}_{31} \mathbf{V}_{21} \mathbf{H}_1) (\mathbf{V}_{30} \mathbf{V}_{20} \mathbf{V}_{10} \mathbf{H}_0) |k_0\rangle_3 |k_1\rangle_2 |k_2\rangle_1 |k_3\rangle_0 . \end{aligned}$$

The final touch is obtained by defining the bit-reversal operator \mathbf{P} which performs the permutation $3210 \rightarrow 0123$:

$$\mathbf{P} |k_3\rangle_3 |k_2\rangle_2 |k_1\rangle_1 |k_0\rangle_0 = |k_0\rangle_3 |k_1\rangle_2 |k_2\rangle_1 |k_3\rangle_0 , \quad (5.23)$$

where $\mathbf{P}^2 = \mathbf{1}$. \mathbf{P} is an obvious unitary (permutation-type) operator, which, when appearing to the right of all the \mathbf{H}_j and $\mathbf{V}_{jj'}$ takes due care of the reverse ordering of the labels in the states. ⁴

Hence, finally, since the equality holds for any computational basis state $|k\rangle$, the states can be omitted altogether, and we can write our circuit for the \mathbf{U}_{QFT} operator:

❶

The QFT circuit. For the $n = 4$ case we arrived at:

$$\mathbf{U}_{\text{QFT}} = \mathbf{H}_3 (\mathbf{V}_{32} \mathbf{H}_2) (\mathbf{V}_{31} \mathbf{V}_{21} \mathbf{H}_1) (\mathbf{V}_{30} \mathbf{V}_{20} \mathbf{V}_{10} \mathbf{H}_0) \mathbf{P} . \quad (5.24)$$

which generalised in a rather obvious way to larger n . Notice that the number of 1- and 2-Qbit gates grows as n^2 .

⁴Mermin suggests that the permutation \mathbf{P} can be constructed out of cNOT gates and one additional Qbit, initially in the state $|0\rangle$. Try. The alternative would be to write \mathbf{P} into a number of two-bit *swap gates*.

Exercise 5.3. By using the fact that $\mathbf{P}^\dagger \mathbf{P} = \mathbf{1}$, and the definition of \mathbf{P} as a permutation of qubits, show that

$$\mathbf{U}_{\text{QFT}}^\dagger = \mathbf{H}_3(\mathbf{V}_{32}^\dagger \mathbf{H}_2)(\mathbf{V}_{31}^\dagger \mathbf{V}_{21}^\dagger \mathbf{H}_1)(\mathbf{V}_{30}^\dagger \mathbf{V}_{20}^\dagger \mathbf{V}_{10}^\dagger \mathbf{H}_0)\mathbf{P}, \quad (5.25)$$

i.e., the same form as \mathbf{U}_{QFT} with $\mathbf{V}_{jj'} \rightarrow \mathbf{V}_{jj'}^\dagger = \mathbf{V}_{jj'}^*$, hence with opposite phases.

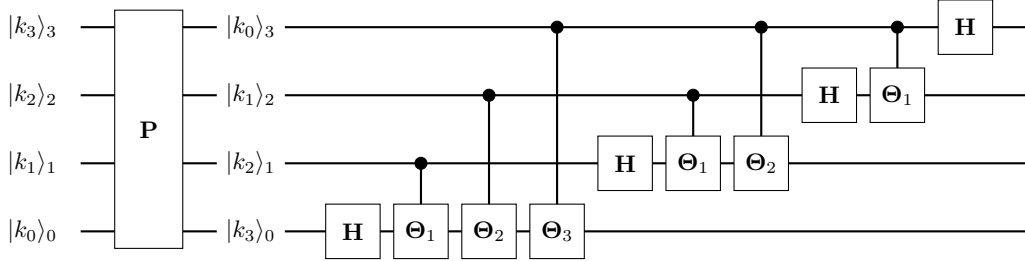


Figure 5.1.: The circuit for \mathbf{U}_{QFT} . The site-indices of the \mathbf{H} is deduced by the line on which they act, which should be read by the state kets at the extreme left, which comply with one of the conventions on bit ordering, to which we adhere, following Mermin: Qbit numbers increase from bottom (least significant bit) to top (most significant bit).

Figure 5.1 shows the circuit corresponding to \mathbf{U}_{QFT} , where you should recall that circuits are left-to-right, while operator action in our equations are right-to-left. Take a moment to trace back the operators appearing in Eq.(5.24). You see the appearance of gates denoted by Θ_l , which require a few extra remarks. Recall that a general control-unitary would read: ⁵

$$\mathbf{C}_{jj'}^{\mathbf{U}} = (\mathbf{N}_0)_j(\mathbf{1})_{j'} + (\mathbf{N}_1)_j(\mathbf{U})_{j'} \quad (5.26)$$

For the present case, the relevant unitary is the phase gate

$$\mathbf{R}_{\mathbf{z}}(\theta_l = \frac{\pi}{2^l}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta_l} \end{pmatrix} = e^{\pi i(\mathbf{N}_1)/2^l} \stackrel{\text{def}}{=} \Theta_l \quad \text{with} \quad l = |j - j'|. \quad (5.27)$$

Hence we can also write:

$$\mathbf{C}_{jj'}^{\Theta_l} = (\mathbf{N}_0)_j(\mathbf{1})_{j'} + (\mathbf{N}_1)_j e^{\pi i(\mathbf{N}_1)_{j'}/2^l} = e^{\pi i(\mathbf{N}_1)_j(\mathbf{N}_1)_{j'}/2^l} = \mathbf{V}_{jj'}, \quad (5.28)$$

where the last expression shows more clearly the symmetric nature of such control- \mathbf{U} gate.

The version of the circuit shown in Fig. 5.2 follows from observing that the $\mathbf{V}_{jj'}$ operators are indeed symmetric under exchange of the control and target bit.

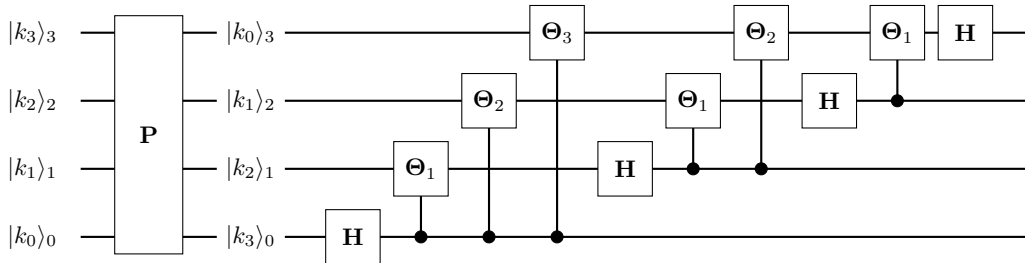


Figure 5.2.: The circuit for \mathbf{U}_{QFT} . This circuit follows from that in Fig. 5.1 upon observing that the control- $\Theta_{l=|j-j'|}$ operators $\mathbf{V}_{jj'}$ are symmetric under exchange of the control and target Qbits.

This last form, with controls following the \mathbf{H} gates, is particularly useful. Indeed, suppose that \mathbf{U}_{QFT} is the *last stage* of a QC, after which we are going to perform a measurement of all Qbits on

⁵Recall that $\mathbf{N}_0 = \frac{1}{2}(\mathbf{1} + \mathbf{Z})$ and $\mathbf{N}_1 = \frac{1}{2}(\mathbf{1} - \mathbf{Z})$ are the projectors on the state $|0\rangle$ and $|1\rangle$, respectively.

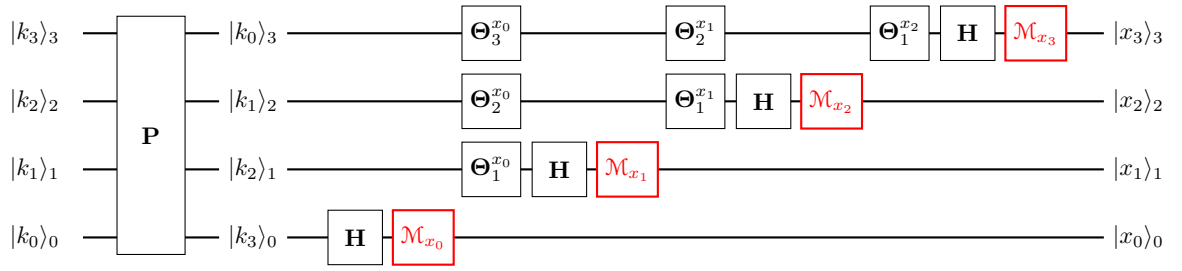


Figure 5.3.: The circuit for U_{QFT} , when measurements of the Qbits on the computational basis are performed *immediately after* the action of H_j on Qbit- j . Notice that the gates denoted by \mathcal{M}_{x_j} are not standard unitary gates: they denote *measurements*, collapsing the Qbit on state $|x_j\rangle_j$.

the computational basis. Then, since the control Qbits are never changed, we might perform the measurement of Qbit-0 *immediately after* H_0 on the lower-line of Fig. 5.2. This provokes a collapse of Qbit-0 into the state $|x_0\rangle_0$ and brings, as a side benefit, that all the subsequent control- U that depend on Qbit-0 transform into *single-Qbit gates* U^{x_0} , i.e., $\mathbf{1}$ if $x_0 = 0$ and \mathbf{U} if $x_0 = 1$ [25].⁶ As you recall, a general control- U — even for an innocent $\mathbf{U} = \Theta_l$ — requires 2 cNOTs and single-Qbit rotations to be implemented in the hardware, while a single-Qbit phase gate is much simpler to implement. The same arguments holds for all other Qbits, if they are measured *immediately after* the action of the corresponding H_j . The circuit that represents U_{QFT} when these measurements are performed is shown in Fig. 5.3.

A final, non-standard, but more symmetric form of the circuit for U_{QFT} is shown in Fig. 5.4. Notice how the Qbit lines are now not straight, and the crossing points between line j and j' are occupied by (symmetric) $V_{jj'}$ gates, whose effect, recall, depends only on $l = |j - j'|$. Notice also how this form of the circuit adheres very closely to Eq (5.24), which we report here for convenience:

$$U_{\text{QFT}} = \mathbf{H}_3(\mathbf{V}_{32}\mathbf{H}_2)(\mathbf{V}_{31}\mathbf{V}_{21}\mathbf{H}_1)(\mathbf{V}_{30}\mathbf{V}_{20}\mathbf{V}_{10}\mathbf{H}_0)\mathbf{P}. \tag{5.29}$$

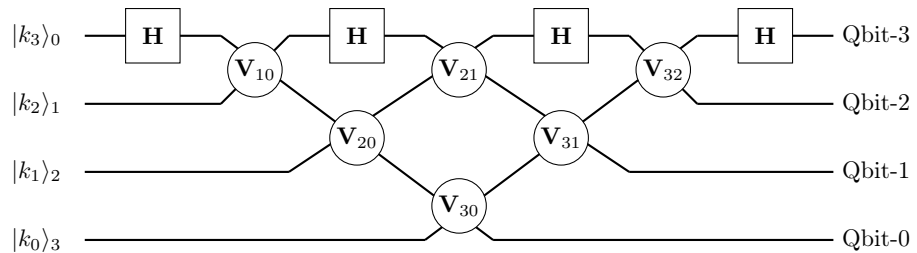


Figure 5.4.: An unconventional form of the circuit for U_{QFT} based on Ref. [25][Fig. 1]. Notice how well it adheres to Eq. (5.24), read from right-to-left.

5.2. Period-finding

Suppose you have an n -bit function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ which is periodic in its n -bit integer domain⁷ $x = 0 \dots 2^n - 1$. Let r be a period of f . So, if $f(\tilde{x}) = \tilde{f}$, where $\tilde{x} \geq 0$ is the smallest integer

⁶Mermin [1][Sec. 3.9] calls this the “*Griffiths-Niu trick*”.

⁷This might seem a rather peculiar choice. But any bounded function in a bounded real domain can be always appropriately rescaled to such a form: after all, we are simply stating that we use a finite precision, in terms of bits, to represent numbers, as a digital computer does.

such that the value \tilde{f} is attained — smaller than the period, $\tilde{x} < r$ —, then:

$$\tilde{f} = f(\tilde{x}) = f(\tilde{x} + r) = f(\tilde{x} + 2r) = \dots = f(\tilde{x} + (P - 1)r).$$

Here P is the smallest integer such that $\tilde{x} + Pr > 2^n$, i.e., outside the domain. So P — which depends on \tilde{x} and hence on \tilde{f} — is the number of “periods seen” inside the n -bit integer domain:

$$P = \left\lfloor \frac{2^n}{r} \right\rfloor \quad \text{or} \quad P = \left\lceil \frac{2^n}{r} \right\rceil + 1. \quad (5.30)$$

We will later learn how to make sure that P is sufficiently large, by having a sufficiently “large domain”. The goal of this section is to show how to use QFT to learn the value of the period r .

Suppose we have a unitary \mathbf{U}_f to code f on a QC. ⁸ As usual, to guarantee that the transformation is invertible (and unitary) we supplement the input $|x\rangle_n$ register with an ancillary output register $|y\rangle_m$ and define:

$$\mathbf{U}_f |x\rangle_n \otimes |y\rangle_m = |x\rangle_n \otimes |y \oplus f(x)\rangle_m \quad \implies \quad \mathbf{U}_f |x\rangle_n \otimes |0\rangle_m = |x\rangle_n \otimes |f(x)\rangle_m. \quad (5.31)$$

We use the Hadamard-trick to calculate \mathbf{U}_f on a uniform superposition of input states:

$$|\Psi\rangle_{n+m} = \mathbf{U}_f \left(\mathbf{H}^{\otimes n} |0\rangle_n \right) \otimes |0\rangle_m = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n \otimes |f(x)\rangle_m. \quad (5.32)$$

Now, suppose you perform a projective measurement on the output m -Qbit register, obtaining a value \tilde{f} for the function (a random one, from the image points). The collapsed state will then read:

$$|\Psi_{\tilde{f}}\rangle_{n+m} = \underbrace{\frac{1}{\sqrt{P}} \sum_{p=0}^{P-1} |\tilde{x} + pr\rangle_n}_{|\psi_{\tilde{f}}\rangle_n} \otimes |\tilde{f}\rangle_m = |\psi_{\tilde{f}}\rangle_n \otimes |\tilde{f}\rangle_m. \quad (5.33)$$



Do not measure too soon. There is no point in measuring now the input register, because we would obtain, with equal probability $1/P$ one of the points $\tilde{x} + pr$ — a random one —, from which I would not be able to extract r . And, since no-cloning is possible, repeating the preparation of $|\Psi\rangle_{n+m}$ would also not help: I would obtain a *different value* of \tilde{f} . As usual.

So, let us keep the state intact, and apply an n -Qbit QFT to the input register state $|\psi_{\tilde{f}}\rangle_n$ (notice that the state $|\Psi_{\tilde{f}}\rangle_{n+m}$ is a product, with $|\tilde{f}\rangle_m$ playing no role):

$$\begin{aligned} \mathbf{U}_{\text{QFT}} |\psi_{\tilde{f}}\rangle_n &= \frac{1}{\sqrt{P}} \sum_{p=0}^{P-1} \mathbf{U}_{\text{QFT}} |\tilde{x} + pr\rangle_n = \frac{1}{\sqrt{P}} \sum_{p=0}^{P-1} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k(\tilde{x} + pr)/2^n} |k\rangle_n \\ &= \sum_{k=0}^{2^n-1} e^{2\pi i k \tilde{x} / 2^n} \left(\frac{1}{\sqrt{2^n P}} \sum_{p=0}^{P-1} e^{2\pi i k pr / 2^n} \right) |k\rangle_n. \end{aligned} \quad (5.34)$$

The probability of measuring now, *after* QFT, one of the computational states $|k\rangle_n$ is totally independent of the overall phase $e^{2\pi i k \tilde{x} / 2^n}$ and given by:

$$\mathbb{P}(k) = \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i k pr / 2^n} \right|^2. \quad (5.35)$$

You notice that when $k \sim \frac{2^n}{r}$, or a multiple of it, the phases in the exponential conspire to give *constructive interference*. Observe, however, that $\frac{2^n}{r}$ is in general not an integer, unless we are so lucky that r is a power of 2.

⁸For the application to RSA this would be a modular exponential subroutine, see Sec. 5.3.5.

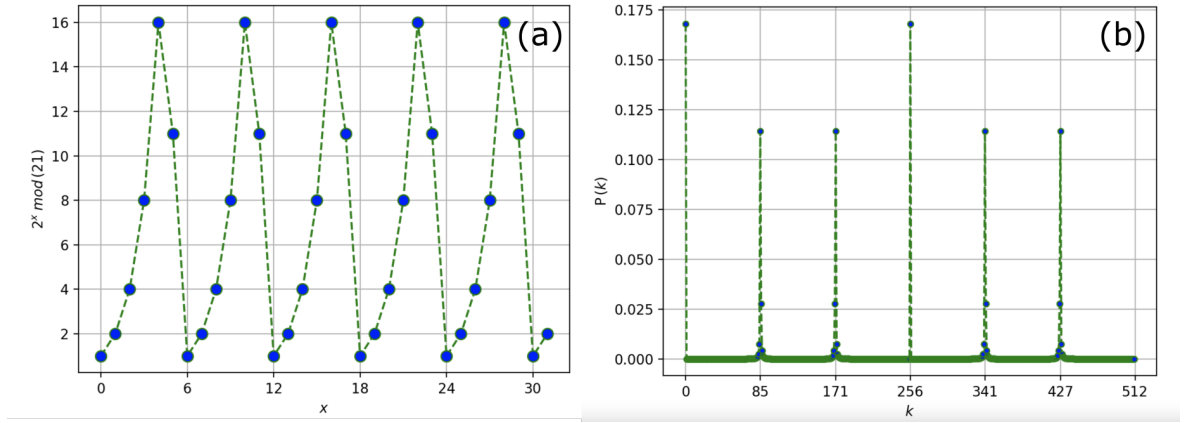


Figure 5.5.: (a) The function $f(x) = 2^x \pmod{21}$ in the domain $x = 0, \dots, 31$. Notice the period $r = 6$. (b) The probability $\mathbb{P}(k)$ in Eq. (5.36). Notice the six peaks at the grid-points k_s . The peak at $k_0 = 0$ and $k_3 = 256$ are sharp, Krönecker-like, because the corresponding $\delta_s = 0$. The other peaks show a small width.

To help you visualise things, let us consider a specific example:

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m \quad \text{with} \quad f(x) = 2^x \pmod{21} .$$

Since $N = 21 < 2^5 = 32$, we can take $m = 5$. As for the domain of x , let us take $n = 9$, hence $x = 0, \dots, 2^9 - 1 = 511$. By experimenting a little, you discover that the only image points of $f(x)$ are $\tilde{f} = \{1, 2, 4, 8, 16, 11\}$, obtained for $x = 0, 1, 2, 3, 4, 5$, and then $f(x)$ repeats periodically with a period $r = 6$, since $2^6 = 64 \equiv 1 \pmod{21}$. We will discuss more about modular arithmetics in Sec. 5.3.1. Figure 5.5(a) shows a plot of $f(x)$ in a restricted domain $x = 0, \dots, 31$. You get $\tilde{f} = 2$, for instance, at the first point $\tilde{x} = 1$. There are $P = 86$ points inside the full $n = 9$ -bit domain, $\tilde{x} + pr$, with $p = 0, \dots, P - 1$, the last one being at $x = 1 + 85 \times 6 = 511$. For $\tilde{f} = 2$, hence $P = 86$, the probability $\mathbb{P}(k)$ would be given by:

$$\mathbb{P}(k) = \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i k p r / 2^n} \right|^2 = \frac{1}{512 \times 86} \left| \sum_{p=0}^{85} e^{2\pi i k p 6 / 512} \right|^2 . \quad (5.36)$$

Figure 5.5(b) shows a plot of $\mathbb{P}(k)$. Notice the 6 sharp peaks.

As anticipated, $\frac{2^n}{r} = \frac{512}{6} = 85 + \frac{1}{3}$ is not an integer. Consider therefore the integer k -grid defined through the *nearest integers*

$$k_s = \text{nint} \left(s \frac{2^n}{r} \right) = s \frac{2^n}{r} + \delta_s \quad \text{with} \quad |\delta_s| \leq \frac{1}{2} , \quad (5.37)$$

where, to stay within the n -bit domain, we take $s = 0, \dots, r - 1$. In the previous example: the grid points are $k_{s=0, \dots, 5} = \{0, 85, 171, 256, 341, 427\}$, and $\delta_{s=0, \dots, 5} = \{0, -\frac{1}{3}, \frac{1}{3}, 0, -\frac{1}{3}, \frac{1}{3}\}$. Observe how the grid points match very well with the peaks in Fig. 5.5(b).

Now that the intuition is guaranteed, let us return general. Evaluating $\mathbb{P}(k)$ at one of such r grid points we find, through a simple geometric series:

$$\begin{aligned} \mathbb{P}(k_s) &= \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i (s + \delta_s r / 2^n) p} \right|^2 = \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i p \theta_s} \right|^2 = \frac{1}{2^n P} \left| \frac{e^{2\pi i P \theta_s} - 1}{e^{2\pi i \theta_s} - 1} \right|^2 \\ &= \frac{1}{2^n P} \frac{\sin^2(\pi P \theta_s)}{\sin^2(\pi \theta_s)} \quad \text{where} \quad \theta_s = \delta_s \frac{r}{2^n} \implies |\theta_s| \leq \frac{r}{2^{n+1}} . \end{aligned} \quad (5.38)$$

Recall now that P is “within an integer interval” from $2^n/r$, see Eq. (5.30), hence:

$$\frac{2^n}{r} - \frac{1}{2} \leq P \leq \frac{2^n}{r} + \frac{1}{2} \implies \left| \frac{Pr}{2^n} - 1 \right| \leq \frac{r}{2^{n+1}} . \quad (5.39)$$

Let us now assume, as in the example above, that the period r is sufficiently small with respect to 2^n so that a large number P of periods occurs within the domain.

i

Assumptions on r and n . Take $N < 2^m = M$ and $r < N$, as will be appropriate when r is the period of $a^x \pmod{N}$.

1) A first possible choice is to take n such that $2^n > N^2$. In this case:

$$2^n > N^2 \quad \Longrightarrow \quad \frac{|\delta_s|}{2^n} \leq \frac{1}{2^{n+1}} \leq \frac{1}{2N^2}, \quad (5.40)$$

a condition which will later prove very useful. To further ensure that $|\theta_s|$ is sufficiently small, so as to proceed with our numerical estimate of $\mathbb{P}(k_s)$, we need that N is sufficiently large, for instance $2N^2 > 2^n > N^2$. If this is the case:

$$2N^2 > 2^n > N^2 \quad \Longrightarrow \quad |\theta_s| = \frac{|\delta_s|r}{2^n} \leq \frac{r}{2N^2} < \frac{r}{2^n} < \frac{N}{2^n} < \frac{1}{\sqrt{2^n}}. \quad (5.41)$$

2) Alternatively: take $n \geq 2m$. Then, automatically, $2^n > N^2$, hence Eq. (5.40) holds, and we would have:

$$\frac{2^n}{r} \geq \frac{2^{2m}}{r} = \frac{M^2}{r} > M \quad \Longrightarrow \quad \frac{r}{2^n} < \frac{1}{M} = \frac{1}{2^m} \quad \Longrightarrow \quad |\theta_s| \leq \frac{1}{2^{m+1}}. \quad (5.42)$$

With such assumptions — satisfied in the previous example — all θ_s are small, indeed *very small*, in more relevant applications. It is then safe to approximate $\sin^2(\pi\theta_s) \approx (\pi\theta_s)^2$ in the denominator of Eq. (5.38) and to set $P \approx 2^n/r$ and $P\theta_s \approx \delta_s$, obtaining:

$$\mathbb{P}(k_s) \approx \frac{1}{2^n P} \frac{\sin^2(\pi P\theta_s)}{(\pi\theta_s)^2} \approx \frac{P}{2^n} \frac{\sin^2(\pi\delta_s)}{(\pi\delta_s)^2} \approx \frac{1}{r} \frac{\sin^2(\pi\delta_s)}{(\pi\delta_s)^2} \geq \frac{1}{r} \frac{4}{\pi^2}. \quad (5.43)$$

The last inequality follows from $\pi|\delta_s| \leq \frac{\pi}{2}$ and the fact that $|\sin \pi\delta_s| \geq 2|\delta_s|$. Observe also that $s = 0$ is special, since $k_{s=0} = 0$, $\delta_{s=0} = 0$, and therefore $\mathbb{P}(0) = \frac{P}{2^n} \approx \frac{1}{r}$. Check that these estimates apply to the previous example.

i

How small is such a probability? The estimates and bounds we have just derived

$$\mathbb{P}(0) \approx \frac{1}{r} \quad \text{and} \quad \mathbb{P}(k_s > 0) \geq \frac{1}{r} \frac{4}{\pi^2}, \quad (5.44)$$

might not look good enough if r is a large number. But recall that *any* k_s — and there are r such grid value points — has a probability of being found in the measurement, and *all but* $k_s = 0$ are useful to infer the values of r , as we will soon discuss. For the time being, be reassured that the probability that I find *any of the* $k_s > 0$ is much more encouraging:

$$\sum_{s=1}^{r-1} \mathbb{P}(k_s) \geq \frac{r-1}{r} \frac{4}{\pi^2} \approx \frac{r-1}{r} 0.4. \quad (5.45)$$

And, remember, this is only a lower bound, obtained by taking strictly the grid-points k_s : values of k very close to k_s would also have a reasonable probability of being “observed”.

Suppose that the measurement has produced a value of $k = k_s$:

$$k_s = \text{nint} \left(s \frac{2^n}{r} \right) = s \frac{2^n}{r} + \delta_s \quad \text{with} \quad |\delta_s| \leq \frac{1}{2}.$$

If we get $k_s = 0$, no information on r is available: bad luck, I have to start again. But if we get any of the $k_s > 0$ values, then we know that:

$$\frac{k_s}{2^n} = \frac{s}{r} + \frac{\delta_s}{2^n},$$

where the LHS is known, but the RHS (containing r) not. Recall however that, with the assumptions on the number of bits used, see Eq. (5.40), we have:

$$\frac{|\delta_s|}{2^n} \leq \frac{1}{2N^2} \implies \left| \frac{k_s}{2^n} - \frac{s}{r} \right| = \frac{|\delta_s|}{2^n} \leq \frac{1}{2N^2}.$$

Question: Can we extract s/r in an unambiguous way?

The question is if two *inequivalent* fractions s_1/r_1 and s_2/r_2 , with $r_1 < r_2 < N$, could both be compatible with the bounds we have derived. The answer is no, for a simple reason. Indeed:

$$\left| \frac{s_1}{r_1} - \frac{s_2}{r_2} \right| = \frac{|s_1 r_2 - s_2 r_1|}{r_1 r_2} \geq \frac{1}{r_1 r_2} \geq \frac{1}{N^2}, \quad (5.46)$$

unless $s_1 r_2 - s_2 r_1 = 0$, which means that the two fractions are really equivalent, up to common factors. But since the “measured” k_s , see Eq. (5.40), is such that

$$\left| \frac{k_s}{2^n} - \frac{s}{r} \right| \leq \frac{1}{2N^2} \implies \frac{s_1}{r_1} \text{ can be inferred in an unambiguous way.}$$

How? With a **continued fraction** expansion of $\frac{k_s}{2^n}$, which will give $\frac{s_1}{r_1}$ reduced to lowest terms (no common factors). An example will be worked out in Sec. 5.3.4. ^a

^aIf you want to go more in depth, I suggest you to look at Appendix K of Mermin’s book.

We are almost done. Nobody guarantees that the fraction $\frac{s_1}{r_1}$, without common factors, which we unambiguously extracted from the “measured” k_s , provides the period r we are looking for, because r might be a *multiple* of r_1 . But it is a simple matter to verify with a classical computer if r_1 is indeed a period, or which multiple of it we need to consider. So, the problem is essentially solved with a little “detective work”, as Mermin puts it. A similar story applies when the measured k is *close* to a grid-point k_s , still within the same peak of $\mathbb{P}(k)$.



To whet your appetite. Periodic functions might not be as simple as the one considered in the previous example. For instance, $f(x) = 2^x \pmod{N}$ for $N = 35$ or $N = 77$ have a rather wild-looking appearance, as you see from Fig. 5.6. With large numbers, and with appropriate requirements on N being a product of two large primes p and q , finding the period r of a function like $f(x) = b^x \pmod{N}$ might be discouragingly difficult. But it is extremely relevant, because *if you could find that period r* , then you would be able to find the prime factors of N , or, from the practical point of view, break the RSA public-key cryptographic system. More about this in Sec. 5.3.

The unimportance of small unitary phase-errors. Suppose I want to apply the QFT algorithm for $n = 1024$ bit integers. I would in principle need to consider phase-gates, in the QFT circuit, with $l = |j - j'|$ as large as n , hence have a control of phase gates with angles $\theta_l = \pi/2^l \sim \pi/2^n \approx 10^{-308}$. It is obvious that such *precision* is an issue: there is no hope of doing that in practice. More generally,

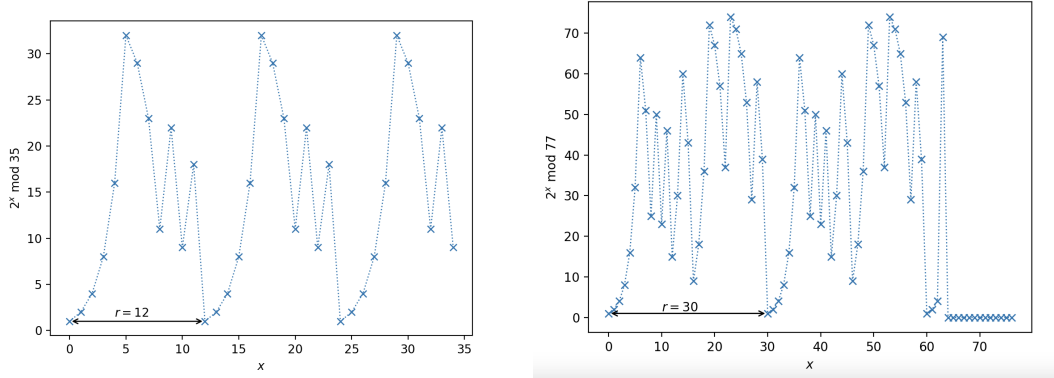


Figure 5.6.: Left: The function $f(x) = 2^x \pmod N$ for $N = 5 \times 7 = 35$, where a period $r = 12$ is clear. Right: Same for $N = 7 \times 11 = 77$ (observe how python loses precision for high x) where $r = 30$ emerges.

the experimentalist’s ability in controlling the angles θ_l must account for inevitable imprecision, to some degree. This means that the phases that are encoded in the unitary QFT are affected by “unitary errors”, due to imprecisions in the gate construction.

Question:

Do inevitable unitary errors in the construction of the gates spoil the period-finding algorithm we have discussed?

The answer is no, and deserves a discussion. Indeed, the construction of a unitary such as \mathbf{U}_{QFT} is akin to some *analog* quantum devices, with continuous variables that can suffer from errors. Nevertheless, the *quantum measurement is digital*: when we measure, in the computational basis, we get 0s and 1s. This digital robustness of measurements saves the day, as we shall see. The probability of getting some measured values of k suffers very little from even relatively large unitary errors.

To show this, let us assume that the QFT is affected by phase errors $\varphi(k, x)$, as follows:

$$\mathbf{U}_{\text{QFT}}^{(\varphi)} |x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k x / 2^n} e^{i\varphi(k, x)} |k\rangle_n . \tag{5.47}$$

We assume the errors to be uniformly bound, $|\varphi(k, x)| < \epsilon \ll 1$, by some small quantity ϵ . By applying such an imperfect QFT to the collapsed n -Qbit input register state, upon measuring \tilde{f} , we would get, see Eq. (5.34):

$$\mathbf{U}_{\text{QFT}}^{(\varphi)} |\psi_{\tilde{f}}\rangle_n = \sum_{k=0}^{2^n-1} e^{2\pi i k \tilde{x} / 2^n} \left(\frac{1}{\sqrt{2^n P}} \sum_{p=0}^{P-1} e^{2\pi i k p r / 2^n} e^{i\varphi(k, \tilde{x} + p r)} \right) |k\rangle_n . \tag{5.48}$$

Notice that now \tilde{x} enters in $\varphi(k, \tilde{x} + p r)$, and not simply as an overall phase factor. ⁹ Still, the probability of measuring k afterwards is:

$$\mathbb{P}_\varphi(k) = \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i k p r / 2^n} e^{i\varphi(k, \tilde{x} + p r)} \right|^2 \approx \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i k p r / 2^n} \left(1 + i\varphi(k, \tilde{x} + p r) \right) \right|^2 ,$$

⁹Hence, the probability of measuring a value of k is in some sense “conditioned on \tilde{f} ”: if we repeat the protocol, we would get a different \tilde{f} , and \tilde{x} . But, after a moment’s reflection, you realise that the same was true in the ideal case, since also in that case the number of periods P seen would depend on \tilde{f} .

where in the second expression we expanded the exponential, assuming $|\varphi(k, x)| \ll 1$. When calculated on the usual grid of $k_s = s \frac{2^n}{r} + \delta_s$ with $s = 0 \cdots r - 1$, see Eq. (5.37), this would give:

$$\begin{aligned} \mathbb{P}_\varphi(k_s) &\approx \frac{1}{2^n P} \left| \sum_{p=0}^{P-1} e^{2\pi i \delta_s p r / 2^n} (1 + i \varphi_{s,p}) \right|^2 \\ &\approx \mathbb{P}(k_s) + \frac{2}{2^n P} \operatorname{Im} \left[\left(\sum_{p=0}^{P-1} e^{-2\pi i \delta_s p r / 2^n} \varphi_{s,p} \right) \left(\sum_{p'=0}^{P-1} e^{2\pi i \delta_s p' r / 2^n} \right) \right], \end{aligned} \quad (5.49)$$

where $\varphi_{s,p} = \varphi(k_s, \tilde{x} + pr)$, and we have further expanded the squared modulus to linear order in $\varphi_{s,p}$. Hence, we can estimate the difference in the probability of the outcome k_s as:

$$\begin{aligned} \left| \mathbb{P}(k_s) - \mathbb{P}_\varphi(k_s) \right| &\approx \frac{2}{2^n P} \left| \operatorname{Im} \left[\left(\sum_{p=0}^{P-1} e^{-2\pi i \delta_s p r / 2^n} \varphi_{s,p} \right) \left(\sum_{p'=0}^{P-1} e^{2\pi i \delta_s p' r / 2^n} \right) \right] \right| \\ &\leq \frac{2}{2^n P} \left| \left(\sum_{p=0}^{P-1} e^{-2\pi i \delta_s p r / 2^n} \varphi_{s,p} \right) \right| \left| \left(\sum_{p'=0}^{P-1} e^{2\pi i \delta_s p' r / 2^n} \right) \right| \\ &\leq \frac{2}{2^n} \left(\sum_{p=0}^{P-1} |\varphi_{s,p}| \right) \leq \frac{2}{2^n} P \epsilon \approx \frac{2}{r} \epsilon, \end{aligned} \quad (5.50)$$

where we used $P \approx 2^n / r$. Recall now, see Eq. (5.45), that:

$$\sum_{s=1}^{r-1} \mathbb{P}(k_s) \geq \frac{r-1}{r} \frac{4}{\pi^2} \approx \frac{r-1}{r} 0.4. \quad (5.51)$$

So, if we want a probability of any of such special values that is at worst, say, 1% from the expected value ≈ 0.4 , it is enough to require that ϵ is at most

$$\epsilon_{\max} = \frac{0.4}{200} = \frac{1}{500}. \quad (5.52)$$

This estimate is reassuring: we do not need a precision of 300 decimal digits in the way we construct our phase-gates! Even more. Suppose that, for instance, we completely neglect to apply control- Θ_l gates when l exceeds some value l_{\max} . Suppose that Θ_l gates with $l > l_{\max}$ are neglected: since they apply repeatedly, but at most n times, this might lead to a phase error as large as $\varphi = n\pi/2^l$. To be safe, we would need to take:

$$l_{\max} > \frac{\log(n\pi/\epsilon_{\max})}{\log 2} \approx 21 \quad \text{for} \quad n = 1024.$$



Info: As an extra bonus, with such truncated l_{\max} , the number of gates in the QFT circuit would scale $O(n)$ rather than $O(n)^2$.

5.3. Factoring and cryptography

Before we enter into the details of the public-key cryptographic system introduced by Rivest, Shamir and Adleman (RSA), we need a few number-theory preliminaries. Even before that, I suggest you to watch the following [Veritasium channel YouTube video](#), quite well made, and with good animations. It is definitely worth the 20 minutes you invest.

5.3.1. Modular arithmetics: some tools.

In $(\text{mod } N)$ arithmetics integers differing by a multiple of N are identified, like in a clock, so that we are left with a representation of each integer in the set $\{0, 1, \dots, N - 1\}$. We will denote by $a \equiv b \pmod{N}$ whenever $a = b + mN$ for some integer m . Addition $(\text{mod } N)$ is relatively simple. Multiplication $(\text{mod } N)$ is more involved, and we concentrate on that.

Let us start defining an important subset of $\{0, 1, \dots, N - 1\}$:

$$G_N = \{\forall n \text{ with } 1 \leq n < N \text{ such that } \gcd(N, n) = 1\}.$$

In words, G_N is the set of all positive integers $n < N$ which are *co-prime* with N , i.e., they share no common factors with N . As a simple example, for $N = 15$:

$$G_{15} = \{1, 2, 4, 7, 8, 11, 13, 14\} \quad \text{with} \quad |G_{15}| = 8,$$

where $|G_N|$ will from now on denote the *number of elements* in G_N . Incidentally, this number is the so-called *Euler's totient function* $\varphi(N) = |G_N|$.

If $a, b \in G_N$, then $ab \pmod{N} \in G_N$, since ab cannot have any common factors with N (a common factor must divide either a or b , as you can immediately show). Hence G_N is *closed* under multiplication $(\text{mod } N)$. Next you can easily show that $ab \equiv ac \pmod{N} \implies b \equiv c \pmod{N}$. So, multiplication by $a \in G_N$ takes distinct members into distinct members. Hence all elements of the type ag with $g \in G_N$ are distinct, and must represent a permutation of G_N . Since $1 \in G_N$, an element d must exist such that $ad \equiv 1 \pmod{N}$, which we also denote by $d = a^{-1}$, the *multiplicative inverse* of a in G_N . This concludes the proof that G_N is indeed a group under multiplication $(\text{mod } N)$.

i

The order of an element a . Take an element $a \in G_N$ and take successive powers: $a^2 \pmod{N}$, $a^3 \pmod{N} \dots$. These powers for a while are all different (recall the “distinct members” story) until, at a certain stage, you must get back to 1 (because $|G_N|$ is finite), and the cycle would start again. The smallest r such that

$$a^r \equiv 1 \pmod{N},$$

is known as the *order* of a in G_N . It is the number of elements of the *cyclic subgroup* $\{1, a, a^2 \dots a^{r-1}\}$ generated from repeated multiplications by $a \pmod{N}$. **Lagrange theorem** guarantees that the number of elements in a subgroup must be a *divisor* of $|G|$. Hence in particular, the order r of each element a must be a divisor of $|G_N|$.

If you want to practice, try to calculate the order of all elements in G_{15} . You should find:

a	1	2	4	7	8	11	13	14
r	1	4	2	4	4	2	4	2

and you notice that 2 and 4 are divisors of $|G_{15}| = 8$.

Exercise 5.4. Take $N = 21$. Determine G_{21} and the order of every element $a \in G_{21}$.

Prime numbers are special in this context, since $G_p = \{1, 2, 3, \dots, p - 1\}$ and $|G_p| = p - 1$: no number less than p can share factors with a prime number p . Take now any $a \in G_p$ and consider its order r , i.e., the integer r such that $a^r \equiv 1 \pmod{p}$. Since r has to divide $|G_p| = p - 1$, then $p - 1$ is certainly a multiple of r , and therefore $a^{p-1} \equiv 1 \pmod{p}$, a relation that is known as Fermat Little Theorem.

1 Fermat little theorem. Take any $a \in \mathbb{Z}$ not divisible by p (including $a \notin G_p$). Since you can always find a representative $a' \equiv a \pmod{p}$ in G_p , you conclude that:

$$a^{p-1} \equiv 1 \pmod{p} \quad \forall a \neq mp. \quad (5.53)$$

Notice that $a = mp$, a multiple of p , can be identified with $a \equiv 0 \pmod{p}$, and 0 does not belong to G_p .

Next, take *two distinct* primes p and q . Consider any integer a which is *not divisible by p and q* , i.e., such that neither p nor q belong to the unique factorisation of a into primes. Certainly a^k for any k cannot be divisible by p , hence also a^{q-1} is not divisible by p . Therefore, by Fermat Little Theorem:

$$(a^{q-1})^{p-1} \equiv 1 \pmod{p},$$

which you can interpret as telling us that $a^{(p-1)(q-1)} = 1 + m_1p$, for some integer m_1 . Symmetrically, a^{p-1} is not divisible by q , and

$$(a^{p-1})^{q-1} \equiv 1 \pmod{q},$$

can be interpreted as telling us that $a^{(p-1)(q-1)} = 1 + m_2q$, for some other integer m_2 . Since p and q are two distinct primes, the last two equations, together, imply that

$$a^{(p-1)(q-1)} = 1 + mpq,$$

for some integer m . Hence we conclude that: ¹⁰

1 Euler theorem. Let p and q be two distinct primes, and take any $a \in \mathbb{Z}$ not divisible by p and q . Then:

$$a^{(p-1)(q-1)} \equiv 1 \pmod{pq}. \quad (5.54)$$

Remarkably, $|G_{pq}| = |G_p||G_q| = (p-1)(q-1)$: Euler's totient function is multiplicative.

Example.

Take $p = 7$ and $q = 11$, so that $pq = 77$ and $(p-1)(q-1) = 60$. Then take $a = 2$. So, $2^{60} - 1$ (a number of order 10^{18}) must be divisible by 77. With smaller numbers: $p = 5$ and $q = 7$, hence $pq = 35$ and $(p-1)(q-1) = 24$. Then $2^{24} \equiv 1 \pmod{35}$. Indeed $2^{24} = 1 + 479349 \times 35$. Both cases are behind the functions plotted in Fig. 5.6.

There is an interesting consequence of Euler's theorem which is particularly useful in RSA, because it allows to eliminate the restrictions on a . Take *any* integer s . Then $a^{s(p-1)(q-1)} \equiv 1 \pmod{pq}$. Multiply both terms by a and you arrive at:

¹⁰Here is an alternative proof of Euler's theorem. Since a is not divisible by p or q , then a has no common factors with pq , hence $a \in G_{pq}$. What is $|G_{pq}|$? One can show that the Euler totient function is multiplicative, hence $|G_{pq}| = |G_p||G_q| = (p-1)(q-1)$. For an elementary proof of this particular case ($N = pq$, the product of two primes), you can argue as follows. There are $pq - 1$ integers $< pq$. Among them $p - 1$ multiples of q and other distinct $q - 1$ multiples of p . Hence $pq - 1 - (p - 1) - (q - 1) = (p - 1)(q - 1)$ are the integers less than pq which have no common factors with p and q . Having established that $|G_{pq}| = (p - 1)(q - 1)$, by Lagrange theorem the order k of any $a \in G_{pq}$ must be a divisor of $(p - 1)(q - 1)$. Since $a^k \equiv 1 \pmod{pq}$, and as a consequence also $a^{(p-1)(q-1)} \equiv 1 \pmod{pq}$.

i **A crucial consequence of Euler's theorem.** Given two primes p and q , then for any $a \in \mathbb{Z}$ and any integer s

$$a^{1+s(p-1)(q-1)} \equiv a \pmod{pq}. \quad (5.55)$$

Remarkably, the previous relationship holds even if a is *divisible* by p or q .¹¹ This is very important in the RSA application, because a will be the integer-coded message, which you certainly do not write by checking that it is co-prime with pq .

Now a few consequences on which RSA is based. Take an integer c with no common factors with $(p-1)(q-1)$, hence $c \in G_{(p-1)(q-1)}$. The multiplicative inverse of c is guaranteed to exist:

$$\exists d \in G_{(p-1)(q-1)} \implies cd \equiv 1 \pmod{(p-1)(q-1)}.$$

The mnemonic is that c will be used to *code* messages, while d is used to *decode* them. The last equation implies that an integer s exists such that

$$cd = 1 + s(p-1)(q-1).$$

Hence, as a consequence of Eq. (5.55), you deduce that for *any* a :

$$a^{cd} = a^{1+s(p-1)(q-1)} \equiv a \pmod{pq}.$$

So, remarkably:

i **Inverse RSA relations.** If d is the multiplicative inverse of c in $G_{(p-1)(q-1)}$, then, for *any* a :

$$\text{If } b \equiv a^c \pmod{pq} \implies b^d \equiv a^{cd} \equiv a \pmod{pq}. \quad (5.56)$$

Details of how this is used for actual coding/encoding in RSA will be discussed in the next section. A number $N = pq$ which is the product of two distinct primes p and q is known as *semi-prime*. These numbers are important in cryptography for a reason that we now explore.

5.3.2. RSA public-key cryptography

Bertuccio (**B** from now on) wants to receive messages from Agata (**A** in the following) in such a way that nobody can read it. To do so, **B** picks two large primes p and q and calculates the very large semi-prime $N = pq$. With p and q of the order of 250-decimals, N will be of order 500-decimals, hence of order $500 \times \log_2(10) \sim 1660$ bits. Then **B** chooses a large “encoding” number c which has *no common factors with* $(p-1)(q-1)$, hence $c \in G_{(p-1)(q-1)}$. Then:

$$\mathbf{B} \text{ makes } N \text{ and } c \text{ public.} \quad (5.57)$$

¹¹Observe that Eq. (5.55) is trivially satisfied if $a = mpq$, with m an integer. Suppose now that a is divisible by just one of the two primes, say q but not p . Then $a = mq$. By assumption a is not divisible by p , so that no power of a is divisible by p . Therefore, Fermat little theorem tells us that

$$\left(a^{s(q-1)}\right)^{p-1} = 1 + np \quad \text{for some integer } n.$$

On multiplying both sides by a you deduce that $a^{1+s(q-1)(p-1)} = a + anp = a + (nm)pq$, which amounts to saying that $a^{1+s(p-1)(q-1)} \equiv a \pmod{pq}$.

①

Choosing numbers without common factors. You might ask how **B** chooses c , making sure that it has no factors in common with $(p-1)(q-1)$. Superficially, this seems quite hard. But is indeed quite simple for two reasons:

1. As a consequence of a beautiful result by Euler — Euler’s Basel problem —, see App. A.3, the probability that two random numbers n and m have no common factors is:

$$\text{Prob}_{\text{gcd}(n,m)=1} = \frac{6}{\pi^2} \approx 0.6079 .$$

Hence, a random choice is quite likely to succeed.

2. You can always *check* that your choice was right, by checking that $\text{gcd}(c, (p-1)(q-1)) = 1$, discarding c otherwise. Calculating the greatest-common-divisor of two numbers can be done *very efficiently* with Euclid’s algorithm, which requires a number of operations of at most 5 times the number of decimal digits of the smallest of the two numbers. See App. A.1 for details.

To encode a message for **B**, using the N and c that **B** made public, **A** does the following.

①

Coding part of RSA.

- C1)** The message is e.g. ASCII-translated into an integer with *fewer* than N digits. Longer messages can be chopped into pieces, each with $< N$ digits. For instance “*Nel mezzo del cammin di nostra vita*” would translate into the following 92-digits-long number, that I show here inserting spaces between each ASCII-character:

$$a = 78\ 101\ 108\ 32\ 109\ 101\ 122\ 122\ 111\ 32\ 100\ 101\ 108\ 32\ 99\ 97\ 109\ 109\ 105\ 110\ 32\ 100\ 105\ 32\ 110\ 111\ 115\ 116\ 114\ 97\ 32\ 118\ 105\ 116\ 97$$

- C2)** **A** calculates an integer b through a $(\text{mod } N)$ -exponential with the public c as exponent:

$$b \equiv a^c \pmod{N} \quad \implies \quad b = \text{encoded message} ,$$

and sends it to **B** on a public channel. This modular exponential with very large numbers might seem a difficult task, but is indeed quite simple, as we will see in Sec. 5.3.5.

Now **B** receives the encoded message b . To decode it, this is what he does.

①

Decoding part of RSA.

- D1)** **B** knows d , the multiplicative inverse of c in $G_{(p-1)(q-1)}$, because he knows the *separate prime factors* p and q of $N = pq$, hence also $(p-1)(q-1)$. Finding d such that $dc \equiv 1 \pmod{(p-1)(q-1)}$, even with these large numbers, can be done quite efficiently using the same Euclid’s algorithm he had used to check that $\text{gcd}(c, (p-1)(q-1)) = 1$. See App. A.2.

- D2)** With d in his hands, the so-called *private key*, **B** can exploit Eq. (5.56) and perform an inverse $(\text{mod } N)$ -exponential:

$$b^d \equiv a \pmod{N} \quad \implies \quad a = \text{decoded message} .$$

Nobody else would be able to perform such an inverse operation described in **D2)**, without knowing the separate factors p and q of N , which allow **B** to calculate its private key d . *Unless you have an*

efficient period-finding machine, using which you can efficiently create a *clone* d' of the private key d . And this leads us to the last section of our story.

But, before ending this section, a small curiosity inspired by Chap. 7 of Simon Singh's *The Code Book*, which I strongly suggest you to read.

Authentication. **B** receives an email message. It is a love letter “from **A**”, or, more properly, from “someone who pretends to be **A**”.

Question: Who are you?

How can **B** be sure that the message *really* comes from **A** and not from an impostor? This is the problem that has originated the need for *certified email* and *digital signatures*. In traditional letters there is an ink signature, tragically missing in our emails.

The solution to the problem comes from an idea of Diffie and Hellmann. **A** starts encrypting the message with **A**'s *private key* $d_{\mathbf{A}}$:

$$b_1 = a^{d_{\mathbf{A}}} \pmod{N_{\mathbf{A}}} .$$

As such, this is a very weak encryption: anybody can decrypt the message by using **A**'s public key $c_{\mathbf{A}}$, performing $b_1^{c_{\mathbf{A}}} \pmod{N_{\mathbf{A}}} = a$. But **A** adds one further encryption step: b_1 is further encrypted with **B**'s public key c :

$$b = b_1^c = \left(a^{d_{\mathbf{A}}} \pmod{N_{\mathbf{A}}} \right)^c \pmod{N} .$$

When **B** receives the encrypted message b , he decrypts it with its private key d , obtaining

$$b_1 = b^d \pmod{N} = a^{d_{\mathbf{A}}} \pmod{N_{\mathbf{A}}} .$$

Knowing that the message is — officially — a “certified email from **A**”, **B** knows that it should be further decrypted with **A**'s public key $c_{\mathbf{A}}$, obtaining finally a . If the message a is readable — it is a love letter —, **B** is sure that it was encrypted with **A**'s private key. If it is meaningless garbage, it was not written by **A**. Quite simple and effective. This comes almost for free in a system based on a separation of public and private keys. Much more difficult is to do similar games with traditional symmetric cyphers, where the sharing of an identical key is essential for decoding.

5.3.3. Breaking RSA through period-finding

Suppose I am not **B**, say I am Ernesto (**E** from now on) but I still want to decode the encoded message b that **A** has sent to **B** on the public channel. Since **E** doesn't know the private key d , he does the following. **E** knows, as everybody, $N = pq$, but not the separate factors p and q . He takes the encoded message b and finds its *order* r in G_N , i.e., the smallest integer r such that:

$$b^r \equiv 1 \pmod{N} . \tag{5.58}$$

This, as you see, requires finding the *period* r of the function $f(x) = b^x \pmod{N}$. So, what?

1 **Period-finding as a decoding tool.** Remarkably r coincides with the order of a in G_N , i.e.,

$$b^r \equiv 1 \pmod{N} \quad \Longrightarrow \quad a^r \equiv 1 \pmod{N} \quad (5.59)$$

In words: the encoded message b and the original message a have the *same order* r in G_N . Why? Because the subgroup generated by a contains $a^c = b$, and therefore it contains the subgroup generated by b as well. Viceversa, by the RSA-inverse relations, since $b^d \equiv a$, the subgroup generated by b contains a . Hence the two subgroups generated by a and b are *identical*, and they *share the same order* r .

This finding is a crucial ingredient for **E**, who is willing to decode b : he has found the *common order* r of b and a in G_N . Now:

- 1) Recall that **B** had chosen c to have *no factors in common* with $(p-1)(q-1)$ (unknown to anybody but **B**).
- 2) Recall also that, by Lagrange theorem, r — the order of b and a in G_N —, must be a divisor of $|G_N| = (p-1)(q-1)$. Hence c can have *no factors in common with* r as well:

$$\gcd(c, r) = 1 \quad \Longrightarrow \quad \exists c' \equiv c \pmod{r} \quad \text{and} \quad c' \in G_r.$$

Consider now G_r . Recall that c is public, but r is **E**'s ingenious discovery, thanks to the period-finding machine. What we just showed allows us to conclude that a representative $c' \equiv c \pmod{r}$ must exist in G_r : **E** can find c' because he knows r , and c is public.

- 3) In turn, c' will have a multiplicative inverse d' in G_r .

$$cd' \equiv c'd' \equiv 1 \pmod{r} \quad \Longrightarrow \quad cd' = 1 + mr,$$

and **E** can find d' *efficiently* by working with the ingredients of Euclid's algorithm applied to $\gcd(c, r) = 1$, see App. A.2.

1 **E has a clone of the private key.** d' will be a **E**'s *clone* of **B**'s private key d . Indeed:

$$b^{d'} \pmod{N} \equiv (a^c)^{d'} \equiv a^{cd'} \equiv a^{1+mr} \equiv a(a^r)^m \equiv a \pmod{N}, \quad (5.60)$$

where we used the fact that $a^r \equiv 1 \pmod{N}$, and all the equivalences are \pmod{N} .

5.3.4. Period-finding and factoring

The great usefulness of period-finding — in turn a gift of QFT on a QC — goes beyond the application to breaking RSA, which we have just described. We will see other applications later on. But, staying close to the current RSA application, you could, with a little extra work, use it to factor N . We will show this for the RSA-relevant case in which N is a composite semi-prime, made of two prime factors: $N = pq$. Relevant details for this part of the story are given in Ref. [1][Sec. 3.10 & App. M]. Here is the recipe.

- 1) N is given, but p and q are unknown. Take a , a random number co-prime with N , i.e., such that $\gcd(a, N) = 1$. The probability that a random a is co-prime with N is quite large — see App. A.3 —, but the probability that a is a multiple of p or q is minuscule: and in case this is not so, the same Euclid's algorithm that you would use to check that $\gcd(a, N) = 1$, would immediately signal p or q as a common factor of a and N , and you would be done. So, let us assume that $a \in G_N$.

- 2) Define now $f(x) = a^x \pmod{N}$ and find — using the period-finding quantum subroutine — the period r of $f(x)$ (which is also the order of a in G_N), as the smallest integer r for which:

$$a^r \equiv 1 \pmod{N}.$$

- 3: **Luck #1)** The first piece of good luck occurs *if r is even*. If so,

$$a^r - 1 \equiv 0 \pmod{N} \quad \implies \quad a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) = mN, \quad (5.61)$$

for some integer m . Since $a^{r/2} \not\equiv 1 \pmod{N}$ — recall that r is the *smallest* integer such that $a^r \equiv 1 \pmod{N}$ — then certainly:

$$a^{r/2} - 1 \not\equiv 0 \pmod{N} \quad \implies \quad a^{r/2} - 1 \text{ is not a multiple of } N. \quad (5.62)$$

This in turn implies that p and q cannot *both* appear as factors of $a^{r/2} - 1$.

- 4: **Luck #2)** The second piece of good luck occurs if:

$$a^{r/2} + 1 \not\equiv 0 \pmod{N} \quad \implies \quad a^{r/2} + 1 \text{ is not a multiple of } N. \quad (5.63)$$

Then p and q do not *both* appear as factors of $a^{r/2} + 1$.

- 5) Eq. (5.61), however, tells us that $(a^{r/2} - 1)(a^{r/2} + 1)$ *must be a multiple of N* , hence contains both p and q as factors — recall that $N = pq$ — while neither of its two factors do contain both p and q at the same time. This can only be realised if p , say, divides $a^{r/2} - 1$, while q divides $a^{r/2} + 1$. Hence:

$$\begin{cases} p = \gcd(a^{r/2} - 1, N) \\ q = \gcd(a^{r/2} + 1, N) \end{cases}. \quad (5.64)$$

i

The likelihood of being lucky. Read Mermin's [1][App. M] if you want to learn that the probability that a random $a \in G_N$ has an order r which is *even* (luck #1), while still $a^{r/2} + 1 \not\equiv 0 \pmod{N}$ (luck #2), is at least $\frac{1}{2}$: remarkably large!

A worked-out example: $f(x) = 2^x \pmod{21}$. Let us return back to the example considered in Sec. 5.2, which I briefly recap in the light of what we learned on modular arithmetics. Take $N = 21$, and $G_{21} = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$, with $|G_{21}| = 12$. Take now $a = 2$, which is co-prime with $N = 21$, $\gcd(a, N) = 1$. You can verify directly that the order of $a = 2$ is $r = 6$, a divisor of $|G_{21}| = 12$: $2^6 \equiv 1 \pmod{21}$. Consider now $f(x) = 2^x \pmod{21}$. The image points \tilde{f} are: $\{1, 2, 4, 8, 16, 11\}$. Since $N = 21 < 2^5 = 32 = M$, it is enough to take $m = 5$ bits in the output register. We can take $n = 9$ in the input register, since $N^2 = 441 < 2^n = 512 < 2N^2 = 882$, which is enough. As we said, upon measuring $\tilde{f} = 2$ in the output register, the collapsed state would be:

$$|\psi_{\tilde{f}=2}\rangle_9 = \frac{1}{\sqrt{P}} \left(\sum_{p=0}^{P-1} |1 + 6p\rangle_9 \right) \quad \text{with} \quad P = 86,$$

i.e., $P = 86$ periods are seen, for $\tilde{f} = 2$, in the domain we have chosen. ¹²

Observe the peaks in Fig. 5.5(b), close to the integer grid-points $k_s = \{0, 85, 171, 256, 341, 427\}$. Now suppose that, by measuring the input register you find the value $k = 86$, just above the second

¹²Higher values of \tilde{f} , would show only $P = 85$ periods.

grid point $k_1 = 85$, but still inside the second peak of $\mathbb{P}(k)$. Your “measurement” would give you $\frac{k}{2^n} = \frac{86}{512}$. To infer $\frac{s}{r}$, we write the **continued fraction** for $\frac{86}{512}$:

$$\frac{86}{512} = 0 + \frac{1}{5 + \frac{1}{1 + \frac{1}{20 + \frac{1}{2}}}}$$

where on the RHS you see a standard notation used for continued fractions:

$$[a_0; a_1, \dots, a_p] = a_0 + \frac{1}{a_1 + \frac{1}{\dots + \frac{1}{a_p}}} \tag{5.65}$$

The so-called q -convergents, with $0 \leq q \leq p$, are fractions obtained by truncating the continued original fraction: $[a_0; a_1, \dots, a_q]$. In the present case:

$$[0; 5] = \frac{1}{5} \quad [0; 5, 1] = \frac{1}{6} \quad [0; 5, 1, 20] = \frac{21}{125}$$

Since $\frac{s_1}{r_1}$ has to have a denominator $r_1 < N = 21$, we take $\frac{1}{6} = \frac{s}{r}$ as our best approximation, which directly gives $r = 6$, the period of $f(x)$.

Now we see if we are lucky enough that we can determine the prime factors of $N = 21$. Observe that the order $r = 6$ is an *even number*: the first piece of luck. Next $a^{r/2} + 1 = 9 \not\equiv 0 \pmod{N}$: the second piece of luck. Indeed:

$$p = \gcd(2^{r/2} - 1, 21) = 7 \quad \text{and} \quad q = \gcd(2^{r/2} + 1, 21) = 3,$$

give us the two prime factors of $N = 21$.

Exercise 5.5. Consider the case in which the measurement gives $k = 170$, close to the third grid point. By using python, or a **continued fraction calculator**, calculate the best rational approximation $\frac{s_1}{r_1}$ to $\frac{170}{512}$ with $r_1 < N = 21$. What is the value of r_1 ? What do you need to get the period r ?

5.3.5. Implementing modular exponentials on a Quantum Computer

How difficult is to compute $f(x) = b^x \pmod{N}$ with x an n -bit integer? You might be worried that this is exceedingly complicated. Let us start thinking to a classical algorithm. The naive way of calculating the function at a required value of $\tilde{x} < 2^n - 1$, would be to loop over $x = 0, \dots, \tilde{x}$ performing modular multiplications by b :

$$f(0) = 1 \rightarrow f(1) = b \rightarrow f(2) \equiv b^2 \pmod{N} \rightarrow f(3) = b^3 \pmod{N} \dots$$

Question:

Is there a way of calculating $f(\tilde{x})$ for a large value \tilde{x} without having to calculate recursively, in a loop over x , the (exponentially) many smaller values of $x < \tilde{x}$?

After thinking a while, it becomes clear that the basic building blocks from which I can reconstruct *any* $f(x)$ are the n powers-of-two values $x = 2^j$ with $j = 0 \dots n - 1$. The basic reason is that $f(x + y) = f(x)f(y)$ and that any integer x can be written — recall the binary expansion — as $x = \sum_{j=0}^{n-1} x_j 2^j$. Indeed:

$$b^{x+y} \pmod{N} = b^x b^y \pmod{N} = \left(b^x \pmod{N} \right) \left(b^y \pmod{N} \right) \pmod{N},$$

where the last step follows from the general property of multiplication modulo- N : if a and b are two (possibly very large) integers, then

$$c = ab \pmod N = a'b' \pmod N,$$

where a' and b' are the representatives of a and b in G_N .¹³ Now, suppose I have calculated a table of n numbers (all $< N$):

$j \rightarrow$	0	1	2	3	...	$n-1$
$t_j \rightarrow$	b	$b^2 \pmod N$	$b^{2^2} \pmod N$	$b^{2^3} \pmod N$...	$b^{2^{n-1}} \pmod N$

which can be readily constructed by using a subroutine that performs the *square of the input* (mod N):

$$t_0 = b \rightarrow \underbrace{b^2 \pmod N}_{t_1} \rightarrow \underbrace{(b^2)^2 \pmod N}_{t_2} \rightarrow \underbrace{(b^4)^2 \pmod N}_{t_3} \dots$$

Next I consider *any* integer x , I write it in binary form, $x = \sum_{j=0}^{n-1} x_j 2^j$ with $x_j = 0, 1$, and I calculate:

$$f(x) \equiv b^x \equiv b^{\sum_{j=0}^{n-1} x_j 2^j} \pmod N \equiv \prod_{j=0}^{n-1} \left[(b^{2^j})^{x_j} \pmod N \right] = \prod_{j=0}^{n-1} t_j^{x_j} \pmod N.$$

Classically, I would therefore create a look-up table ($j, t_j = f(2^j)$), which I can keep in the memory of my computer, and the calculate $f(x)$ as needed using this table.

Python exercise 5.1. Apply this method to implement a numerically stable way of calculating $2^x \pmod{77}$ and $3^x \pmod{77}$, improving the quality of Figs. 5.6.

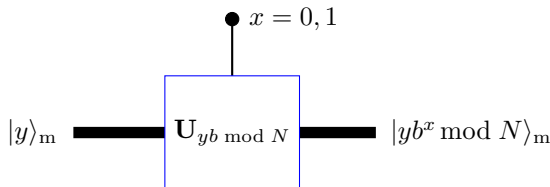


Figure 5.7: The control-U gate that performs multiplication $y \rightarrow yb \pmod N$ on a m -Qbit register.

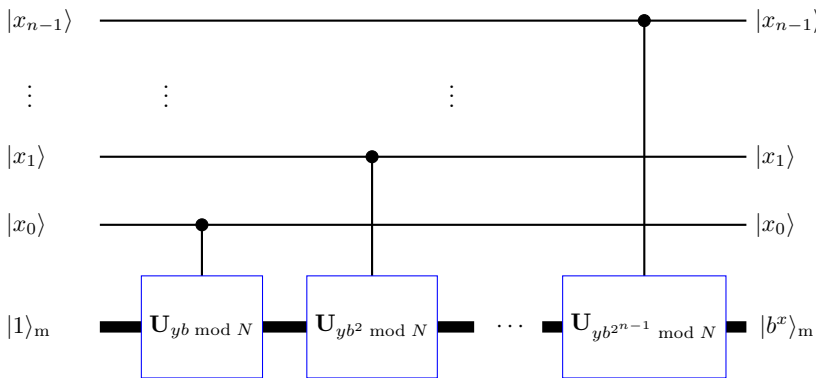


Figure 5.8: Sketch of a quantum circuit for modular exponentiation $x \rightarrow b^x \pmod N$ of an n -Qbit input register $|x>_n$, using the power-of-2 classical strategy. For the many missing ingredients, see Ref. [26].

On a Quantum Computer the philosophy is different. Qbits are precious and delicate, and you do not store look-up tables: Qbits in a Quantum-Drive would modify their state in quite a short time, due to decoherence and dissipation caused by inevitable interactions with the environment. Rather,

¹³Indeed, $a = a' + m_1N$ and $b = b' + m_2N$, hence:

$$ab = (a' + m_1N)(b' + m_2N) = a'b' + (m_1b' + m_2a' + m_1m_2N)N.$$

you implement the function on a Quantum Circuit, and you use the QFT period-finding procedure to find the period. This comes with a “single” application of the unitary circuit. To design a circuit to calculate $f(x)$ you use an idea similar to the power-of-2 strategy described above, but now with control- \mathbf{U} gates doing the work. The details, however, are rather intricate — ancillary work Qbits, used reversibly, and other routines for modular additions are needed — and pertain to quantum software design: see Ref. [26]. Figures 5.7-5.8 show a sketch of the idea.

5.4. Phase estimation protocol

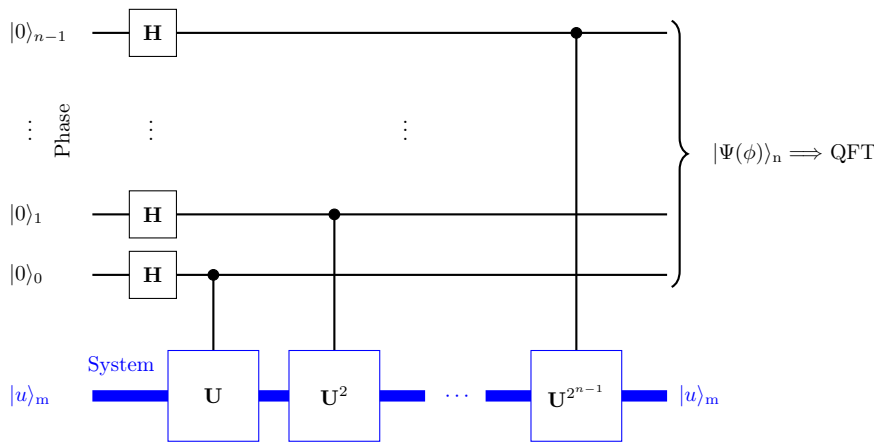


Figure 5.9: The quantum circuit behind the phase estimation protocol. Although the top lines act as control-Qbits, the final superposition state $|\Psi(\phi)\rangle_n$, thanks to the Hadamards, knows about the phase ϕ of \mathbf{U} . Nothing interesting occurs in the lower registry, where the eigenstate $|u\rangle_m$, supposedly known, is unchanged.

Here is another interesting application of QFT. Suppose someone has a way of implementing an unknown m -Qbit black-box unitary \mathbf{U} , and also provides me one of its eigenstates $|u\rangle_m$. The problem is to devise a circuit to estimate the phase $2\pi\phi$, with $\phi \in [0, 1)$, of the eigenvalue of \mathbf{U} associated to $|u\rangle_m$:

$$\mathbf{U}|u\rangle_m = e^{2\pi i\phi}|u\rangle_m .$$

With n -bits integers, I can always write the phase ϕ as:

$$\phi = \frac{a}{2^n} + \delta_\phi \quad \text{where } |\delta_\phi| \leq \frac{1}{2^{n+1}} \text{ and } a = \sum_{j=0}^{n-1} a_j 2^j \text{ with } a_j = 0, 1 . \quad (5.66)$$

By employing an idea reminiscent of that used in the modular-exponentials of Sec. 5.3.5, we now consider \mathbf{U}^{2^j} with $j = 0, \dots, n - 1$, which acts as:

$$\mathbf{U}^{2^j}|u\rangle_m = e^{2\pi i(\phi 2^j)}|u\rangle_m .$$

I can in principle assume that this is “calculable” as a power of the black-box \mathbf{U} which is provided.

Example. The T-gate.

Consider the single-Qbit T-gate, whose action on the state $|1\rangle$ is $\mathbf{T}|1\rangle = e^{i\frac{\pi}{4}}|1\rangle$ (while $\mathbf{T}|0\rangle = |0\rangle$): hence $\mathbf{T}^4 = \mathbf{Z}$. So, here $m = 1$, $\mathbf{U} = \mathbf{T}$, and $|u\rangle_m = |1\rangle$. In this case $\phi = \frac{1}{8} = 0.001$, where the last expression should be intended in binary notation, since $\phi = \frac{1}{2^3}$. More details are given in the [Qiskit textbook](#).

I set an n -Qbit register $|x\rangle_n$ which will eventually encode a state $|\Psi(\phi)\rangle_n$ having ϕ in its belly. For that purpose, I devise a control- \mathbf{U}^{2^j} gate with the control-Qbit being the computational basis j^{th} -Qbit and the unitary \mathbf{U}^{2^j} acting on the $|u\rangle_m$ “target” register:

$$c_j\text{-}\mathbf{U}^{2^j}|x_j\rangle_j \otimes |u\rangle_m = |x_j\rangle_j \otimes \left(e^{2\pi i(\phi^{2^j})x_j} |u\rangle_m \right) = \left(e^{2\pi i(\phi^{2^j})x_j} |x_j\rangle_j \right) \otimes |u\rangle_m .$$



Important: This gate does nothing if $x_j = 0$, while it adds a phase $e^{2\pi i(\phi^{2^j})}$ to the state $|u\rangle_m$ if $x_j = 1$. As such, it is a legitimate phase-control gate, which does not change the computational basis state $|x_j\rangle_j$, and with $|u\rangle_m$ as target. Nevertheless, the added phase can be seen as “borrowed” by the computation state $|x_j\rangle_j$ — recall that $|u\rangle_m$ is an eigenstate of \mathbf{U} , and therefore it is not “essentially changed” by this phase —, as highlighted by the second form in the last expression. And this leads to profound consequences if I act on a *superposition* of computational states:

$$c_j\text{-}\mathbf{U}^{2^j}(\mathbf{H}_j|0\rangle_j) \otimes |u\rangle_m = \frac{1}{\sqrt{2}} \left(|0\rangle_j + e^{2\pi i(\phi^{2^j})} |1\rangle_j \right) \otimes |u\rangle_m . \quad (5.67)$$

Now we are ready to proceed. We implement the quantum circuit shown in Fig. 5.9. Starting from $|0\rangle_n$, with the usual Hadamard-trick we arrive at the following final state of the n -Qbit register:

$$\begin{aligned} |\Psi(\phi)\rangle_n &= \frac{1}{\sqrt{2^n}} \left(|0\rangle_{n-1} + e^{2\pi i(\phi^{2^{n-1}})} |1\rangle_{n-1} \right) \otimes \cdots \otimes \left(|0\rangle_1 + e^{2\pi i\phi^{2^1}} |1\rangle_1 \right) \otimes \left(|0\rangle_0 + e^{2\pi i\phi} |1\rangle_0 \right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x_{n-1}=0}^1 \cdots \sum_{x_0=0}^1 e^{2\pi i\phi(x_{n-1}2^{n-1} + \cdots + x_02^0)} |x_{n-1}\rangle_{n-1} \otimes \cdots \otimes |x_0\rangle_0 \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i\phi x} |x\rangle_n . \end{aligned} \quad (5.68)$$

Observe how the control-gates did not create any entanglement between the $|x\rangle_n$ register and the “target” $|u\rangle_m$ register, but did indeed create non-trivial controlled-phase interactions *within* the $|x\rangle_n$ register, thanks to the Hadamard superposition.

As stressed in various occasions, nothing could be more pointless than performing a projective measurement on the computational basis on $|\Phi(\phi)\rangle_n$. But imagine feeding this state to an $\mathbf{U}_{\text{QFT}}^{-1}$ Quantum machine. You would then get:

$$\begin{aligned} \mathbf{U}_{\text{QFT}}^{-1}|\Psi(\phi)\rangle_n &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i\phi x} \mathbf{U}_{\text{QFT}}^{-1}|x\rangle_n = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i\phi x} e^{-2\pi i x k / 2^n} |k\rangle_n \\ &= \sum_{k=0}^{2^n-1} \left(\frac{1}{2^n} \sum_{x=0}^{2^n-1} e^{2\pi i\delta_\phi x} e^{2\pi i(a-k)x/2^n} \right) |k\rangle_n . \end{aligned} \quad (5.69)$$

where we used Eq. (5.66).

The probability of measuring, on the state $\mathbf{U}_{\text{QFT}}^{-1}|\Phi(\phi)\rangle_n$, the computational state $|k\rangle_n$ can be now easily extracted to be:

$$\mathbb{P}(k) = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} e^{2\pi i\delta_\phi x} e^{2\pi i(a-k)x/2^n} \right|^2 . \quad (5.70)$$

For $\delta_\phi=0$, i.e., if the n -bit representation of the phase ϕ is *exact*, then we would have a Krönecker-delta appearing in $\mathbb{P}(k)$:

$$\mathbb{P}(k)|_{\delta_\phi=0} = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} e^{2\pi i(a-k)x/2^n} \right|^2 = \delta_{a,k} , \quad (5.71)$$

i.e., the probability $\mathbb{P}(k)$ is perfectly peaked on the computation state representing the value $a = 2^n \phi$.

What about the more realistic case in which $\delta_\phi \neq 0$? Then the probability that I measure a k that ends-up being the correct a is:

$$\mathbb{P}(k = a) = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} e^{2\pi i \delta_\phi x} \right|^2 = \left| \frac{1}{2^n} \frac{\sin(\pi \delta_\phi 2^n)}{\sin(\pi \delta_\phi)} \right|^2. \tag{5.72}$$

If this gives you a *déjà vu* sensation, you are totally right: it is precisely the same algebra behind the period-finding strategy of Sec. 5.2. And indeed, since $|\delta_\phi| \leq \frac{1}{2^{n+1}}$, see Eq. (5.66), you can use the same approximations used for the period-finding estimates, arriving at the same bound found there:¹⁴

$$\mathbb{P}(k = a) = \left| \frac{1}{2^n} \frac{\sin(\pi \delta_\phi 2^n)}{\sin(\pi \delta_\phi)} \right|^2 \approx \frac{\sin^2(\pi 2^n \delta_\phi)}{(\pi 2^n \delta_\phi)^2} \geq \frac{4}{\pi^2} = 0.405 \dots \tag{5.73}$$

So, you have a good probability of getting the n -bit integer $a = \text{nint}(2^n \phi)$. And one can improve the bound in the probability by adding extra bits. As shown by Cleve *et al.* [27][App.C], the best n -bit approximation to ϕ is obtained with a probability $P_\epsilon > 1 - \epsilon$ provided you use a slightly larger register with $n' = n + \lceil \log_2(\frac{1}{2\epsilon} + \frac{1}{2}) \rceil$ Qbits.

Python exercise 5.2. Using Qiskit, apply the phase estimation protocol to determine the phase of the T-gate.

5.5. Finding eigenstates and eigenvalues of an Hamiltonian

The crucial input in the phase-estimation protocol is our ability to “prepare” $|u\rangle_m$, an eigenstate of U , and our ability to “code” U on a QC with a number of elementary gates that scales polynomially with m . Now I will show a beautiful illustration of a similar scheme for the determination of eigenstates

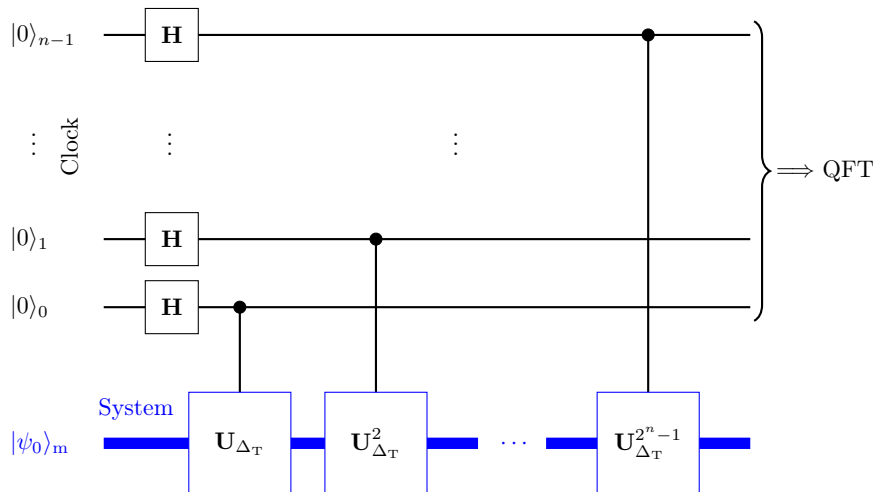


Figure 5.10: The circuit to determine the eigenvalues of a physical Hamiltonian. U_{Δ_T} represents here the evolution operator for a time-step Δ_T , which we assume to be able to represent efficiently.

and eigenvalues of a physical Hamiltonian \hat{H} .

To illustrate the ideas, I show how we could use a QC to solve the quantum problem for a single particle in one-dimension [2]. This is more an excuse, with a familiar illustration, than a necessity: finite-dimensional interacting spin local Hamiltonians would be an alternative, indeed even more appropriate, example. We will comment later on the generality of the approach.

¹⁴If you plot $f(x) = \frac{\sin^2 \pi x}{(\pi x)^2}$ versus x you realise that $f(x) \geq f(\frac{1}{2}) = \frac{4}{\pi^2}$ for $|x| \leq \frac{1}{2}$. Identify then $x = 2^n |\delta_\phi|$, and the result follows.

Suppose we have a quantum particle in the region $[-L, L]$, and we discretise this domain in $N = 2^m$ grid points at a distance ΔX . To do that, I insist in using the variable x to be the usual m -bit integer, $x = 0, \dots, 2^m - 1$, and I use x as a label of the grid points:

$$X_x = -L + x\Delta_X \quad \text{with} \quad x = 0, \dots, 2^m - 1 \quad \text{and} \quad \Delta_X = \frac{2L}{2^m - 1} .$$

I hope you can tolerate this strange notation, whose only useful ingredient is that it preserves the standard integer label x for the computation basis states $|x\rangle_m$ of an m -Qbit register.

The wave-function $\psi(X)$ of the quantum particle is now represented as:

$$|\psi\rangle_m = \sum_{x=0}^{2^m-1} \psi_x |x\rangle_m \quad \text{with} \quad \psi_x = \psi(X_x) .$$

i

A remarkable gift of a QC. Notice the remarkable fact that with an m -Qbit quantum register I allow an exponentially large number of grid point in the interval $[-L, L]$. A similar gift will soon occur for the time-evolution. Observe, however, that Mermin's caveat, suggested when discussing the QFT, applies: The "mathematically simple" operation $x \rightarrow x + 1$ (here neighboring points on the grid) translates into a physically quite complex (and non-local) operation on the Qbits representing the state $|x\rangle_m$. Hence, for instance, representing a discretised Laplacian on our QC would be a non-trivial issue. More about this in the final comments.

Now we consider the time-evolution operator over a fixed time interval Δ_T :

$$\mathbf{U}_{\Delta_T} = e^{-\frac{i}{\hbar} \hat{H} \Delta_T} , \quad (5.74)$$

where \hat{H} denotes the Hamiltonian acting on the 2^m -dimensional Hilbert space of the problem. As shown by Lloyd in a seminal paper on Science in 1996 [28], one can "simulate" \mathbf{U}_{Δ_T} efficiently on a QC for a large class of *physically significant Hamiltonians*. Lloyd's idea is based on the application of Suzuki-Trotter decompositions of \hat{H} . More precisely, if

$$\hat{H} = \sum_{\ell=1}^L \hat{H}_\ell \quad \text{with a } k\text{-local } \hat{H}_\ell ,$$

where k -local means that it involves interaction of at most k -Qbits, then the standard Lie-Trotter first-order decomposition

$$e^{-\frac{i}{\hbar} \hat{H} t} = \left(e^{-\frac{i}{\hbar} \hat{H}_1 \frac{t}{P}} \dots e^{-\frac{i}{\hbar} \hat{H}_L \frac{t}{P}} \right)^P + O\left(\frac{t^2}{P}\right) , \quad (5.75)$$

can be used to argue that one can simulate $e^{-\frac{i}{\hbar} \hat{H} t}$, for a given t , with an accuracy ε using a number of quantum gates which scales as $\sim PLN_k^2$, where $N_k = 2^k$ is the local Hilbert space dimension of the k -interacting Qbits. ¹⁵

¹⁵Each quantum gate must be devised so that the experimental error less than $\varepsilon/(PLN_k^2)$, in order to produce an overall error ε . The quadratic error in the first-order Lie-Trotter decomposition implies that the minimum number of steps needed to propagate the system for a time t with an accuracy ε scales as $P_{\min} \sim t^2/\varepsilon$, but each individual unitary term $e^{-\frac{i}{\hbar} \hat{H}_\ell t/P}$ is applied for a time t/P .

Example. Heisenberg model on a cubic lattice.

Consider, to exemplify, a (possibly anisotropic and disordered) Heisenberg model on a hyper-cubic lattice in d dimensions:

$$\hat{H} = \sum_{\langle \mathbf{x}, \mathbf{x}' \rangle} \sum_{\alpha=x,y,z} J_{\mathbf{x}, \mathbf{x}'}^{\alpha\alpha} \hat{\sigma}_{\mathbf{x}}^{\alpha} \hat{\sigma}_{\mathbf{x}'}^{\alpha}.$$

Then $L = 3dN_{\text{sites}}$, and each link ℓ refers to a physical link $\langle \mathbf{x}, \mathbf{x}' \rangle$ on the lattice and a choice of $\alpha = x, y, z$ in the two-site interaction term $\hat{\sigma}_{\mathbf{x}}^{\alpha} \hat{\sigma}_{\mathbf{x}'}^{\alpha}$. The two-Qbit gate that you need to implement must encode the 4×4 unitary matrix $\exp(ib\hat{\sigma}_{\mathbf{x}}^{\alpha} \hat{\sigma}_{\mathbf{x}'}^{\alpha})$ with an appropriate b .

1

Representing the Hamiltonian. For a particle in one-dimension, think of how you would implement the potential energy and the real-space-discretised Laplacian operator (kinetic energy) by using a Trotter decomposition of \hat{H} . What type of quantum gates would you need to represent \mathbf{U}_{Δ_T} ?

So, let us assume we have an efficient circuit for constructing \mathbf{U}_{Δ_T} acting on the m -Qbit state (target) register where we want to encode the wave-function $|\psi\rangle_m$. How would we prepare an initial state? This is a difficult task, in general. But, for our purpose, it is perfectly legitimate to think that you start from a state entirely localised at a single point on the grid, i.e., a single computational basis state:

$$|\psi_0\rangle_m = |x^{(0)}\rangle.$$

This choice is good to search for eigenstates that have a non-zero projection on $|\psi_0\rangle_m$.

Let us now discuss time and the Schrödinger dynamics. I want to propagate the evolution up to a final time T_{fin} and I set a grid of discrete times using n -bit integers $t = 0, 1, \dots, 2^n - 1$. Notice that t is here an integer label for the actual time, which is discretised as:

$$T_t = t\Delta_T \quad \text{with} \quad t = 0, \dots, 2^n - 1 \quad \text{and} \quad \Delta_T = \frac{T_{\text{fin}}}{2^n - 1}.$$

For reasons that will be hopefully clear soon, I will promote the n -bit integer t into an n -Qbit register with computational basis denoted by $|t\rangle_n$: we might call it the “clock” register. The “ $t = 0$ ” state would be associated to the $|0\rangle_n = |0\rangle_{n-1} \cdots |0\rangle_0$. We now set-up a series of control- \mathbf{U} gates, as pictorially denoted in Fig. 5.10. The first control- \mathbf{U} acts with a control on the clock Qbit-0, and target on the physical m -Qbit register:

$$c_0 - \mathbf{U}_{\Delta_T} |t_0\rangle_0 \otimes |\psi_0\rangle_m = |t_0\rangle_0 \otimes (\mathbf{U}_{\Delta_T}^{t_0} |\psi_0\rangle_m).$$

In words: it applies the evolution operator \mathbf{U}_{Δ_T} if the clock state is $|1\rangle_0$, while it is the identity if the clock is in state $|0\rangle_0$. It does not change the clock state (which acts as a control Qbit), but only the (target) physical register. As such, it is a perfectly legitimate control-unitary. Notice that, unlike the phase-estimation protocol, now this is not simply a phase that the target can “lend” to the other register. Notice also the magic that occurs if I apply such a control- \mathbf{U} to a *superposition* of clock states, through the usual Hadamard trick:

$$(c_0 - \mathbf{U}_{\Delta_T}) \mathbf{H}_0 |0\rangle_0 \otimes |\psi_0\rangle_m = \frac{1}{\sqrt{2}} \left(|0\rangle_0 \otimes |\psi_0\rangle_m + |1\rangle_0 \otimes \mathbf{U}_{\Delta_T} |\psi_0\rangle_m \right). \quad (5.76)$$

Entanglement between clock and physical registers has been generated. One more step, with the control acting on clock state $|t_1\rangle_1$, with a unitary $\mathbf{U}_{\Delta_T}^2$ will show the clear structure that emerges:

$$\begin{aligned} (c_1 - \mathbf{U}_{\Delta_T}^2) (c_0 - \mathbf{U}_{\Delta_T}) \mathbf{H}_1 \mathbf{H}_0 |0\rangle_1 |0\rangle_0 \otimes |\psi_0\rangle_m &= \frac{1}{\sqrt{2^2}} \left(|0\rangle_1 |0\rangle_0 \otimes |\psi_0\rangle_m + |0\rangle_1 |1\rangle_0 \otimes \mathbf{U}_{\Delta_T} |\psi_0\rangle_m \right. \\ &\quad \left. + |1\rangle_1 |0\rangle_0 \otimes \mathbf{U}_{\Delta_T}^2 |\psi_0\rangle_m + |1\rangle_1 |1\rangle_0 \otimes \mathbf{U}_{\Delta_T}^3 |\psi_0\rangle_m \right). \end{aligned}$$

The structure should be clear: each clock-state $|t\rangle_n$ is associated to a number of applications of \mathbf{U}_{Δ_T} as appropriate for the “integer time” t associated to the binary string (t_{n-1}, \dots, t_0) . The general expression for the final state generated after all the control-gates have acted, after Hadamards on the clock Qbits, as prescribed by Fig. 5.10, is therefore:

$$|\Psi_{\text{fin}}\rangle_{n+m} = \frac{1}{\sqrt{2^n}} \sum_{t=0}^{2^n-1} |t\rangle_n \otimes (\mathbf{U}_{\Delta_T}^t |\psi_0\rangle_m) . \quad (5.77)$$

Notice the very complex entangled superposition of states at all times, with the clock register states $|t\rangle_n$ selecting the appropriate “time-frame” for the system states. As usual, no point in measuring now!

Before applying the usual QFT machine, let us pause for a second and rewrite the final state by expanding the system states in terms of eigenfunctions of the Hamiltonian \hat{H} , with $\hat{H}|\phi_\alpha\rangle_m = E_\alpha|\phi_\alpha\rangle_m$. Without loss of generality will assume that all $E_\alpha \geq 0$. The initial state is expanded as:

$$|\psi_0\rangle_m = \sum_{\alpha=0}^{2^m-1} C_\alpha |\phi_\alpha\rangle_m ,$$

where $C_\alpha = \langle \phi_\alpha | \psi_0 \rangle_m$ are overlap coefficients. Hence the action of the evolution operator is now simple:

$$\mathbf{U}_{\Delta_T}^t |\psi_0\rangle_m = \sum_{\alpha=0}^{2^m-1} C_\alpha e^{-\frac{i\Delta_T E_\alpha t}{\hbar}} |\phi_\alpha\rangle_m .$$

Therefore, the final state can be written as:

$$|\Psi_{\text{fin}}\rangle_{n+m} = \frac{1}{\sqrt{2^n}} \sum_{t=0}^{2^n-1} \sum_{\alpha=0}^{2^m-1} C_\alpha e^{-\frac{i\Delta_T E_\alpha t}{\hbar}} |t\rangle_n \otimes |\phi_\alpha\rangle_m . \quad (5.78)$$

Now we apply \mathbf{U}_{QFT} to the clock-registers to move to the “Fourier basis”, while doing nothing to the physical state register. We get:

$$\begin{aligned} |\tilde{\Psi}_{\text{fin}}\rangle_{n+m} = \mathbf{U}_{\text{QFT}} \otimes \mathbf{1}_m |\Psi_{\text{fin}}\rangle_{n+m} &= \frac{1}{\sqrt{2^n}} \sum_{t=0}^{2^n-1} \sum_{\alpha=0}^{2^m-1} \frac{1}{\sqrt{2^n}} \sum_{f=0}^{2^n-1} C_\alpha e^{-\frac{i\Delta_T E_\alpha t}{\hbar}} e^{2\pi i f t / 2^n} |f\rangle_n \otimes |\phi_\alpha\rangle_m \\ &= \sum_{f=0}^{2^n-1} \sum_{\alpha=0}^{2^m-1} C_\alpha \left(\frac{1}{2^n} \sum_{t=0}^{2^n-1} e^{-\frac{i\Delta_T E_\alpha t}{\hbar}} e^{2\pi i f t / 2^n} \right) |f\rangle_n \otimes |\phi_\alpha\rangle_m \\ &= \sum_{f=0}^{2^n-1} \sum_{\alpha=0}^{2^m-1} C_\alpha A_{f,\alpha} |f\rangle_n \otimes |\phi_\alpha\rangle_m . \end{aligned} \quad (5.79)$$

Here, I have first reshuffled the various sums and finally highlighted an important factor

$$A_{f,\alpha} = \frac{1}{2^n} \sum_{t=0}^{2^n-1} e^{-\frac{i\Delta_T E_\alpha t}{\hbar}} e^{2\pi i f t / 2^n} = \frac{1}{2^n} \sum_{t=0}^{2^n-1} e^{2\pi i \left(\frac{f}{2^n} - \tilde{f}_\alpha \right) t} , \quad (5.80)$$

multiplying the amplitude of each of the product states participating into the final entangled superposition. This second expression for $A_{f,\alpha}$ is written in terms of dimensionless frequencies

$$0 \leq \tilde{f}_\alpha = \frac{E_\alpha \Delta_T}{2\pi \hbar} \leq 1 , \quad (5.81)$$

where the time-step Δ_T is assumed to be sufficiently small, so that even the largest rescaled frequency does not exceed 1. ¹⁶

¹⁶Observe that, most likely, the highest part of the spectrum is not nicely captured with a finite-dimensional truncation of the Hilbert space. As usual, the low-energy part of the spectrum is safer.

Suppose we now perform, *after* QFT, a projective measurement on the clock state $|f\rangle_n$. This measurement is associated to a projector $\hat{P}_f = |f\rangle_n \langle f| \otimes \mathbf{1}_m$. The probability of measuring the value f for the clock register is therefore:

$$\mathbb{P}(f) = \langle \tilde{\Psi}_{\text{fin}} | \hat{P}_f | \tilde{\Psi}_{\text{fin}} \rangle = \sum_{\alpha=0}^{2^m-1} |C_\alpha|^2 |A_{f,\alpha}|^2 = \sum_{\alpha=0}^{2^m-1} |C_\alpha|^2 \left| \frac{1}{2^n} \sum_{t=0}^{2^n-1} e^{2\pi i (\frac{f}{2^n} - \tilde{f}_\alpha) t} \right|^2. \quad (5.82)$$

The state of the system, upon measuring f , would collapse into:

$$|\tilde{\Psi}_f\rangle = \frac{1}{\sqrt{\mathbb{P}(f)}} |f\rangle_n \otimes \left(\sum_{\alpha=0}^{2^m-1} C_\alpha A_{f,\alpha} |\phi_\alpha\rangle_m \right).$$

The square-modulus of $A_{f,\alpha}$ in the expression for $\mathbb{P}(f)$ reveals the by-now familiar structure we have observed in the phase-estimate protocol and in the period-finding problem, with a few extra complications. By considering the usual nearest-integer trick of taking $s_\alpha = \text{nint}(2^n \tilde{f}_\alpha)$ we can write:

$$\tilde{f}_\alpha = \frac{s_\alpha}{2^n} + \delta_\alpha \quad \text{with} \quad 0 \leq s_\alpha < 2^n - 1 \quad \text{and} \quad |\delta_\alpha| \leq \frac{1}{2^{n+1}}. \quad (5.83)$$

Then:

$$A_{f,\alpha} = \frac{1}{2^n} \sum_{t=0}^{2^n-1} e^{2\pi i \frac{f - s_\alpha}{2^n} t} e^{-2\pi i \delta_\alpha t} \quad (5.84)$$

Suppose that our rescaling induced by Δ_T in Eq. (5.81) is such that $\tilde{f}_{\bar{\alpha}}$ has an exact n -bit binary representation, hence $\delta_{\bar{\alpha}} = 0$, for a specific eigenvalue associated to $\bar{\alpha}$. Then the geometric sum in Eq. (5.84) would precisely¹⁷ give a Krönecker-delta, and you would find that:

$$A_{f,\bar{\alpha}} = \delta_{f,s_{\bar{\alpha}}} \quad \implies \quad \mathbb{P}(f) = \delta_{f,s_{\bar{\alpha}}} |C_{\bar{\alpha}}|^2 + \sum_{\alpha \neq \bar{\alpha}} |C_\alpha|^2 |A_{f,\alpha}|^2.$$

Hence the value $f = s_{\bar{\alpha}}$ is obtained with a relatively high probability, dominated by the overlap $|C_{\bar{\alpha}}|^2$. Correspondingly, if $f = s_{\bar{\alpha}}$ is measured by the clock register, and you neglect that possibility that such an f -value was generated by a different $\alpha' \neq \bar{\alpha}$ — notice that $|A_{f=s_{\bar{\alpha}},\alpha'}|^2$ is very small, by destructive interference, but not necessarily zero — then the state has collapsed into:

$$|\tilde{\Psi}_{f=s_{\bar{\alpha}}}\rangle = |s_{\bar{\alpha}}\rangle_n \otimes |\phi_{\bar{\alpha}}\rangle_m.$$

This tells us that we have created a *filter* for the energy eigenstates.

¹⁷You are summing powers of all roots of unity. A simple geometric series calculation shows that for all integers a :

$$\frac{1}{2^n} \sum_{t=0}^{2^n-1} e^{2\pi i \frac{a}{2^n} t} = \delta_{a,0}.$$

6. Quantum cryptography

Cryptography is an old and fascinating subject. ¹ The public-key RSA cryptographic system currently used was a great discovery, based on a smart use of number theory. But the security of it relies on the believed classical intractability of integer factorisation, which you could break with a Quantum Computer. Even without a QC, RSA relies on the fact that no superior mathematical intelligence, within the enemy field, comes out with a *very very smart classical algorithm*, based on a yet-to-discover theorem, which would allow integer factorisation with polynomial resources. “You never know...”, remember?

We will see below that Quantum Mechanics provides an absolutely secure cryptography, allowing the practical implementation of the most secure of all cryptographic systems: the Vernam cypher.

6.1. To be sure: the Vernam cypher

In 1919, Gilbert Vernam, an AT&T Bell Labs engineer, patented a cryptosystem, known as **Vernam cypher**, an example of a **one-time pad** cypher based on the logic **XOR**. The essential aspects of this cypher can be summarised as follows. Agata (**A** from now on) wants to send a secrete message to Bertuccio (**B** from now on). To encode a message **A** does the following.

i

A: Coding.

- C1)** The plain-text message is e.g. ASCII-translated into a binary string of n bits. For instance “*Nel mezzo del cammin di nostra vita*”, a 35 characters string, including spaces, with 8-bits for each ASCII-character, would translate into the following $n = 280$ binary string:

```
a = 01001110011001010110110000100000011011010110010101111010011110100110111100100000110010001100101
01101100001000000110001101100001011011010110110101101001011011100010000011001000110100100100000
0110111001101111011100110110100011001001100001001000001110110011010010111010001100001
```

- C2)** **A** takes a coding-key $\underline{c} = (c_{n-1}, \dots, c_0)$ as long as the message, n -bits, and calculates the bitwise sum (mod 2), without carry-overs:

$$\underline{b} \equiv \underline{a} \oplus \underline{c} = (a_{n-1} \oplus c_{n-1}, \dots, a_0 \oplus c_0) \implies \underline{b} = \text{encoded message}$$

The result \underline{b} is then sent to **B** on a public channel, which we assume to be perfect, without errors.

¹Read the beautiful book by Simon Singh, *The Code book*, to know more about this. Chapter 7 of this book tells a very nice story: that of *Pretty Good Privacy* (PGP), a free software created by Phil Zimmermann, a computer scientist, in 1991. PGP allows to quickly encrypt and digitally sign messages on a personal computer, using RSA to encrypt a *short* key of a traditional symmetric cypher, used in turn to encrypt a much longer message. I suggest you to read it. It explains very nicely the idea of “digital signature”. Phil Zimmermann was subject to a grand-jury investigation in 1993: FBI was “unhappy” with PGP. The case against Zimmermann was dropped only in 1996.

1

B: Decoding.

D) **B** receives the encoded message \underline{b} . To decode it, **B** needs to possess the *same secret key* \underline{c} used by **A**. In other words, the “decoding-key” for the inverse transformation is $\underline{d} = \underline{c}$. Indeed:

$$\underline{b} \oplus \underline{c} = (\underline{a} \oplus \underline{c}) \oplus \underline{c} = \underline{a} \quad \Longrightarrow \quad \underline{a} = \text{decoded message}$$

since $c_j \oplus c_j = 0$ and \oplus is associative.

Re-using keys leads to disaster. Indeed if two messages $\underline{a}^{(1)}$ and $\underline{a}^{(2)}$ are coded with the same key, hence $\underline{b}^{(1)} \equiv \underline{a}^{(1)} \oplus \underline{c}$, and $\underline{b}^{(2)} \equiv \underline{a}^{(2)} \oplus \underline{c}$, then by bitwise summing the two encoded messages the key disappears:

$$\underline{b}^{(1)} \oplus \underline{b}^{(2)} = \underline{a}^{(1)} \oplus \underline{a}^{(2)},$$

and it would be simple for a crypto-analyst to recover the two original messages.

The secret of a secret key. Crucial for perfect secrecy is that the key \underline{c} chosen by **A** is *random*. Claude Shannon published in 1949 a paper entitled **Communication Theory of Secrecy Systems** where his studies on the **Mathematical Theory of Communication** are applied to what he called *theoretical secrecy*. Here is a very brief summary of the main concepts of relevance to us.

First, let us explain the main terms of the problem. A message M is a sequence of letters from an alphabet of symbols \mathcal{M} , for instance English, including all the symbols you need, punctuation, space, etc. Given a certain number of symbols $|\mathcal{M}|$, and a maximum length L_M of messages, you can in principle form a huge number of messages, $|\mathcal{M}|^{L_M}$, most of which meaningless. Some are meaningful, however, and you want to encrypt them. The message M should be encrypted into a cryptogram E by using a transformation T , also known as *key*, which is essentially a one-to-one function which you can invert to obtain the message back from the cryptogram:

$$E = TM \quad \xrightarrow{\text{decrypt}} \quad M = T^{-1}E.$$

The alphabet used in the cryptogram could be entirely different from \mathcal{M} , and it could include \mathcal{M} as a subset. Perhaps the simplest transformation key is the so-called Caesar’s cipher, a simple substitution cipher which works as follows. You take the message $M = m_1m_2m_3m_4 \dots$ and you apply to each letter a function f which “moves letters” by a fixed amount k , $f(m) = m + k \pmod{26}$, with the 26 letters of the alphabet enumerated as $0, \dots, 25$:

$$E = e_1e_2e_3e_4 \dots = f(m_1)f(m_2)f(m_3)f(m_4) \dots$$

For instance:

$$M = \text{avecaesarcastraparamus} \quad \xrightarrow{k=2} \quad \text{cxgecguctecuvtrcrtcowu}$$

More generally, the function f is only required to be *invertible*, and could be any permutation of the 26 letters. As you probably know, Caesar’s cipher, and its variants with permutations of letters, are very easy to break because they do not change the relative frequency of letters appearing. But smarter transformation keys have been invented, for instance the so-called **Vigenère cipher**, where the substitution key k_i changes with a given period d , so that $e_i = f_i(m_i) = m_i + k_i \pmod{26}$. For instance, with $d = 3$:

$$M = \text{avecaesarcastraparamus} \quad \xrightarrow{k=234} \quad \text{cyiediudvedwvuervdvcpyu}$$

The case in which k_i is *random* and has no period leads to the Vernam cipher, as we shall later discuss.

More generally, you could think of having a whole set of these transformation keys which you could use to encrypt your messages. Needless to say, there is no restriction on the alphabet used in the cryptograms, nor on the length of the cryptogram, which could differ from the length of the message: the only important requirement is that any transformation T you apply, should be invertible.

More formally, see sketch in Fig. 6.1, suppose that there is a finite number of messages ² $\{M_m, m =$

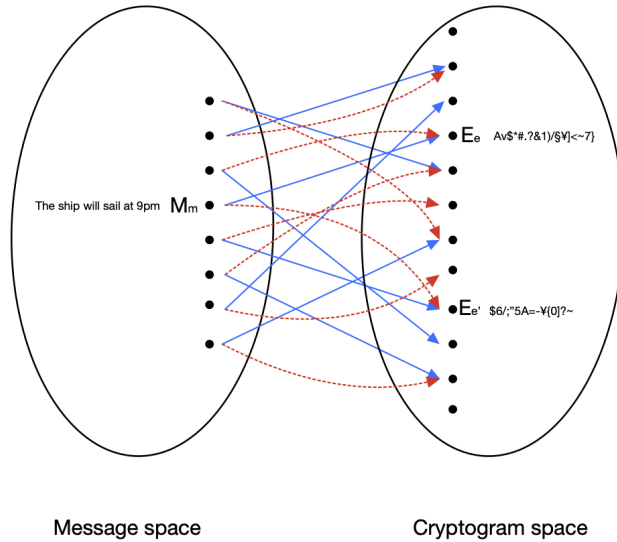


Figure 6.1: The spaces of messages and cryptograms in Shannon’s setting of a perfect secrecy system. The blue and dashed red arrows denote two different cryptographic keys (one-to-one transformations) from the space of messages to that of cryptograms. Messages and cryptograms can use a different alphabet of symbols. Observe that the second message from the top is mapped into the same cryptogram by the two different keys: we will see that this is not good for a perfectly secure system.

$1, N_M\}$, each with a certain *a priori* probability $\mathbb{P}(M_m)$. Let us agree to disregard all messages with zero probability, so that $\mathbb{P}(M_m) > 0$. The messages use letters from a given alphabet of symbols \mathcal{M} , for instance, English, or, after an ASCII-translation, binary: it doesn’t matter. Assume that the possible cryptograms also form a finite space $\{E_e, e = 1, N_E\}$, and use a possibly different alphabet of symbols. The space of “cryptographic keys” $\{K_k, k = 1, N_K\}$ is such that each K_k is associated to a *function* or *transformation* T_k , which maps *one-to-one* — for a unique decoding of a cryptogram — a message M_m into a cryptogram E_e :

$$E_e = T_k M_m \quad \xrightarrow{\text{decrypt}} \quad M_m = T_k^{-1} E_e .$$

We can associate a probability $\mathbb{P}(K_k)$ to the key K_k : the probability of using the transformation T_k . Clearly, in order for the transformations to be one-to-one, one must have at least as many cryptograms as messages: $N_E \geq N_M$.

Define now the conditional probability $\mathbb{P}(E_e|M_m)$ as the sum of all the key probabilities for the transformations leading from message M_m to cryptogram E_e :

$$\mathbb{P}(E_e|M_m) = \sum_k \mathbb{P}_{E_e=T_k M_m}(K_k) . \tag{6.1}$$

Obviously, the probability of a certain cryptogram E_e from *any message* is:

$$\mathbb{P}(E_e) = \sum_m \mathbb{P}(E_e|M_m) . \tag{6.2}$$

Now Bayes’ theorem allows us to calculate the so-called *a posteriori* probability of message M_m given that cryptogram E_e is intercepted (hence $\mathbb{P}(E_e) \neq 0$):

$$\mathbb{P}(M_m|E_e) = \frac{\mathbb{P}(E_e|M_m)\mathbb{P}(M_m)}{\mathbb{P}(E_e)} \quad \forall M_m \quad \text{provided } \mathbb{P}(E_e) \neq 0 . \tag{6.3}$$

²This is a simplifying assumption, since arbitrarily long messages would be *uncountably many* and there are technical difficulties in defining a probability for all subsets of an uncountable set. You need the theory of measure to do that.

Shannon defines the *perfect secrecy condition* to be given by the fact that

$$\mathbb{P}(M_m|E_e) = \mathbb{P}(M_m) \quad \forall M_m, \forall E_e \quad \text{independently of the value of } \mathbb{P}(M_m) . \quad (6.4)$$

In words, this tells us that the *a posteriori probability* that I would assign to the message M_m , given that the cryptogram E_e was intercepted, does not differ at all from the *a priori probability* of M_m : the information acquired by detecting E_e has not changed in anything the probability of the message originating it. Very reasonable. Now use Bayes' theorem, and arrive at:

Perfect secrecy condition.

$$\text{Perfect secrecy} \iff \mathbb{P}(E_e|M_m) = \mathbb{P}(E_e) \quad \forall M_m, \forall E_e \quad \text{with } \mathbb{P}(E_e) \neq 0 . \quad (6.5)$$

This is Theorem 6 in Shannon's paper. In words, the probability of every cryptogram used must be totally independent of the original message.

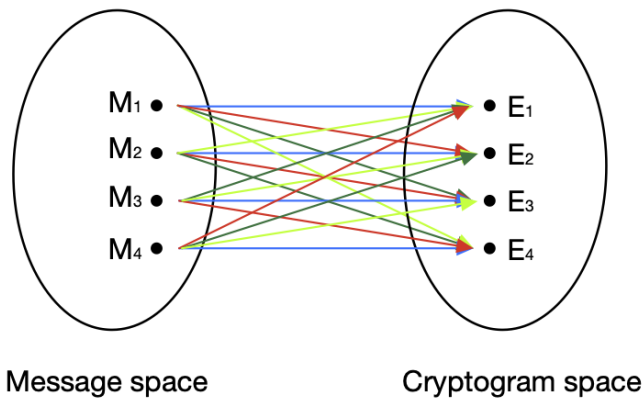


Figure 6.2: A perfect system with $n = 4$ messages. All key probabilities are identical, as well as the probabilities of the n cryptograms.

Notice that this leads to a *constructive recipe* for a perfect secrecy system. Indeed, observe that:

- 1) Eq. (6.2), where a sum over messages appears, combined with Eq. (6.5), dictates that *there must be only one message M_m leading to E_e* , i.e., a uniquely defined key $\bar{k}(m, e)$ must exist such that:

$$\mathbb{P}(E_e) = \mathbb{P}(E_e|M_m) = \mathbb{P}_{E_e = T_{\bar{k}}M_m}(K_{\bar{k}}) \quad (6.6)$$

- 2) Consider now all the E_e which are reached by a transformation, hence have $\mathbb{P}(E_e) > 0$: they must be, in number, $\geq N_M$, for the transformations to be one-to-one. Each message M_m — imagine fixing it — must lead to any of these different E_e with a unique and different key \bar{k} . Hence the number of keys must be greater than the number of messages, $N_K \geq N_M$.
- 3) Now imagine fixing a E_e with $\mathbb{P}(E_e) > 0$ and consider all different keys sending different messages M_m to that particular E_e : all those keys must have the *same probability*, by Eq. (6.6).
- 4) Perfect systems in which $n = N_M = N_K = N_E$ are such that all keys have probability $1/n$ and, likewise, all cryptograms have probability $1/n$:

$$\mathbb{P}(E_e) = \frac{1}{n} = \mathbb{P}(T_k) .$$

The probabilities of the n different messages *do not have to be equal*, of course. ³

³Notice that there might be more keys than messages, hence correspondingly more cryptograms than messages. This is possible, although unnecessary. What would you conclude about the probabilities of keys and cryptograms?

Fig. 6.2 realises a perfect secrecy system with $n = 4$.

Let us verify that Vernam’s cipher is indeed a perfect secrecy system. There are different ways to prove that. To be concrete, let the messages be written in ASCII-binary strings, as Dante’s “*Nel mezzo del cammin ...*”. Dante’s string \underline{a} is far from being a random string, of course. Incidentally, even a Sunday crypto-analyst would recognise the 6 occurrences of **00100000** — the ASCII binary for 32, the space — or the fact that most of the 8-long binary segments of the string start with **011**, consequence of the fact that most of the ASCII characters for ordinary text symbols are over 64. This is however not important. The bitwise (mod 2) sum $b_j = a_j \oplus c_j$ allows us to restrict our attention to a single bit j . Define the *marginal probability* for bit b_j as

$$\mathbb{P}(b_j) = \sum_{\text{all other bits } \neq j} \mathbb{P}(\underline{b}) ,$$

and similarly for $\mathbb{P}(a_j)$ and $\mathbb{P}(c_j)$. Consider now $\mathbb{P}(b_j)$. The **XOR** Vernam cypher implies that:

$$\begin{aligned} \mathbb{P}(b_j = 0) &= \mathbb{P}(a_j = 0 \wedge c_j = 0) + \mathbb{P}(a_j = 1 \wedge c_j = 1) \\ &= \mathbb{P}(a_j = 0)\mathbb{P}(c_j = 0) + \mathbb{P}(a_j = 1)\mathbb{P}(c_j = 1) \\ \mathbb{P}(b_j = 1) &= \mathbb{P}(a_j = 0 \wedge c_j = 1) + \mathbb{P}(a_j = 1 \wedge c_j = 0) \\ &= \mathbb{P}(a_j = 0)\mathbb{P}(c_j = 1) + \mathbb{P}(a_j = 1)\mathbb{P}(c_j = 0) , \end{aligned}$$

where we use that fact that the key and the message are *uncorrelated*, hence joint probabilities factorise. You can rewrite these in matrix form, abbreviating $p_0 = \mathbb{P}(c_j = 0)$ and $p_1 = \mathbb{P}(c_j = 1)$, as follows:

$$\begin{pmatrix} \mathbb{P}(b_j = 0) \\ \mathbb{P}(b_j = 1) \end{pmatrix} = \mathbf{T} \begin{pmatrix} \mathbb{P}(a_j = 0) \\ \mathbb{P}(a_j = 1) \end{pmatrix} \quad \text{with} \quad \mathbf{T} = \begin{pmatrix} p_0 & p_1 \\ p_1 & p_0 \end{pmatrix} . \quad (6.7)$$

Notice that \mathbf{T} is a *stochastic matrix* (SM). And you remember that a “fair coin flipping” SM is such that the outcome probability is $(\frac{1}{2}, \frac{1}{2})^T$ *independently* of the “incoming probability” of the source. Such a \mathbf{T} — recall our classical version of the “beam splitter” in Eq. (1.12), Sec. 1.2.1 — is given by:

$$\mathbf{T} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} ,$$

and is associated to a *random unbiased* key, where c_j is equally likely to be 0 or 1. The probability of bit b_j is, in such a case, guaranteed to be equally random and unbiased, independently of the original message bit a_j . For binary strings of length n the Shannon entropies of both the coding-keys \underline{c} and the cryptograms \underline{b} are the maximum possible: $n \log 2$.

An alternative equivalent formulation of the story invokes the concept of *mutual information*. The mutual information of two (discrete) random variables X and Y is defined as:

$$I(X; Y) = I(Y; X) = \sum_{x,y} \mathbb{P}(x, y) \log \frac{\mathbb{P}(x, y)}{\mathbb{P}(x)\mathbb{P}(y)} \stackrel{\text{def}}{=} D_{KL}(\mathbb{P}_{XY} || \mathbb{P}_X \otimes \mathbb{P}_Y) \geq 0 , \quad (6.8)$$

where the RHS defines the Kullback-Leibler divergence, and the non-negativity of $I(X; Y)$ follows from Jensen’s inequality. Very simple algebra shows that:

$$I(X; Y) = H(X) - H(X|Y) ,$$

where

$$H(X) = - \sum_x \mathbb{P}(x) \log \mathbb{P}(x) ,$$

is the Shannon entropy of the variable X , while $H(X|Y)$ is the conditional entropy:

$$H(X|Y) \stackrel{\text{def}}{=} \sum_y \mathbb{P}(y) H(X|Y = y) = - \sum_{x,y} \mathbb{P}(y) \mathbb{P}(x|y) \log \mathbb{P}(x|y) = - \sum_{x,y} \mathbb{P}(x, y) \log \frac{\mathbb{P}(x, y)}{\mathbb{P}(y)} ,$$

where, recall, $\mathbb{P}(x|y) = \mathbb{P}(x, y)/\mathbb{P}(y)$. For the present application, the fact that $H(E) = H(E|M)$, hence $I(E; M) = 0$, guarantees perfect secrecy.



Delicate points of Vernam's cypher. Two delicate points of the scheme emerge immediately.

Random numbers?) Who will give us a perfect source of random numbers? This problem has affected Monte Carlo computer simulations for decades, before people realised that the (pseudo)-random number generators used were perhaps not good enough. For cryptographic applications, the requirements on the quality of the (pseudo)-random number generators is even more demanding.

Key distribution?) The second delicate point of this one-pad cypher is that **B** has to *share* the secret key \underline{c} that **A** used in encoding. Hence, the problem of exchanging secret messages has been transformed into the problem of exchanging secret random keys, a kind of catch-22 dilemma.

Here again I cannot refrain from quoting a paragraph from Mermin's book.

The problem of exchanging such random strings in a secure way might appear to be identical to the original problem of exchanging meaningful messages in a secure way. But at this point quantum mechanics come to the rescue and provides an entirely secure means for exchanging identical sequences of random bits. Pause to savor this situation. Nobody has figured out how to exploit quantum mechanics to provide a secure means for directly exchanging meaningful messages. The secure exchange is possible only because the bit sequences are random. On the face of it one would think nothing could be more useless than such a transmission of noise. What is bizarre is that human ingenuity combined with human perversity has succeeded in inventing a context in which the need to hide information from a third party actually provides a purpose for such an otherwise useless exchange of random strings of bits.

N. David Mermin, *Quantum Computer Science*, Chapter 6, p. 139

6.2. Implementing Qbits with photon polarisation

The practical implementations of *random key distributions* exploit Qbits based on *photons*, more precisely on the two *polarisation states* of a photon. Let me remind you of this. A photon with wave-vector \mathbf{k} in the z direction, say, can have only *transverse polarisation* in the xy plane, depending on the way the “electric field” oscillates. Classical intuition on electromagnetic waves is vital for understanding things, although the photon is ultimately a quantum particle.

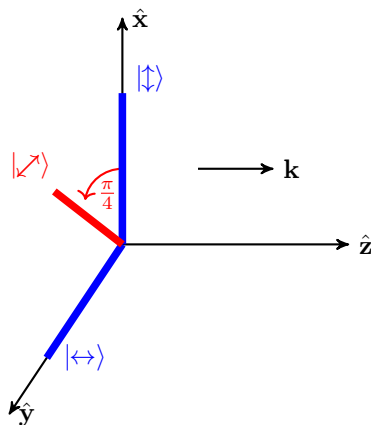


Figure 6.3: An electromagnetic plane-wave travelling along \mathbf{z} , with an electric field $\mathbf{E} = \text{Re} \epsilon e^{i(kz - \omega t)}$. Here $\epsilon = \hat{\mathbf{x}} = |\uparrow\downarrow\rangle$, and $\epsilon = \hat{\mathbf{y}} = |\leftrightarrow\rangle$ denote “vertical” and “horizontal” linear polarisations, corresponding to the $|\pm, \mathbf{z}\rangle$ spin eigenstates. $\epsilon = \frac{1}{\sqrt{2}}(\hat{\mathbf{x}} + \hat{\mathbf{y}}) = |\nearrow\rangle$ and (not shown) $\epsilon = \frac{1}{\sqrt{2}}(\hat{\mathbf{x}} - \hat{\mathbf{y}}) = |\searrow\rangle$ denote diagonal linear polarisations at $\pm 45^\circ$, corresponding to $|\pm, \mathbf{x}\rangle$ spin eigenstates. Circular polarisations $\epsilon_{\pm} = \frac{1}{\sqrt{2}}(\hat{\mathbf{x}} \pm i\hat{\mathbf{y}})$, denoted as $\epsilon_+ = |\odot\rangle$ and $\epsilon_- = |\ominus\rangle$, cannot be represented real polarisation vectors in this figure. They would correspond, in the spin language, to $|\pm, \mathbf{y}\rangle$ spin states.

A plane-wave electromagnetic field travelling along the $\hat{\mathbf{z}}$ direction would have an electric field given

by (taking a unit amplitude):

$$\begin{aligned} \mathbf{E} = \operatorname{Re} \epsilon_{\alpha,\delta} e^{i(kz-\omega t)} &= \operatorname{Re} \left(\overbrace{\hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} e^{i\delta} \sin \alpha}^{\epsilon_{\alpha,\delta}} \right) e^{i(kz-\omega t)} \\ &= \hat{\mathbf{x}} \cos \alpha \cos(kz - \omega t) + \hat{\mathbf{y}} \sin \alpha \cos(kz - \omega t + \delta), \end{aligned} \quad (6.9)$$

where the polarisation $\epsilon_{\alpha,\delta}$ has to be *transverse*, $\hat{\mathbf{z}} \cdot \epsilon_{\alpha,\delta} = 0$, so that $\nabla \cdot \mathbf{E} = 0$. Real polarisation vectors $\epsilon_{\alpha,\delta=0}$ correspond to electric fields that oscillate *linearly*, at an angle α with respect to the $\hat{\mathbf{x}}$ axis in Fig. 6.3. $\alpha = 0$ correspond to “vertical” polarisation, $\alpha = \frac{\pi}{2}$ to “horizontal” polarisation, $\alpha = \pm \frac{\pi}{4}$ to “diagonal” polarisations at $\pm 45^\circ$ from the $\hat{\mathbf{x}}$ -axis. Complex polarisation vectors with $\delta \neq 0$ lead to electric fields that move forward by making a spiral in the xy plane. They can be easily obtained by making the photon pass through a *wave-plate* made of a *birefringent* material like calcite or quartz, see App. B. Particularly noteworthy is the case of a circularly polarised wave, for which

$$\epsilon_{\pm} = \epsilon_{\frac{\pi}{4}, \pm \frac{\pi}{2}} = \frac{1}{\sqrt{2}}(\hat{\mathbf{x}} \pm i\hat{\mathbf{y}}),$$

obtained through a quarter-wave plate, see below.

There is a very direct correspondence between polarisation states of the “photon” — pictured, as before, through the corresponding \mathbf{E} -field — and spin-1/2 eigenstates. With our Qbit computational states we would write:

$$\mathbf{Z} \text{ - states: } \begin{cases} |0\rangle = |\uparrow\rangle \mapsto |\downarrow\rangle \\ |1\rangle = |\downarrow\rangle \mapsto |\leftrightarrow\rangle \end{cases} \quad \mathbf{X} \text{ - states: } \begin{cases} \mathbf{H}|0\rangle = |+, \mathbf{x}\rangle \mapsto |\nearrow\rangle \\ \mathbf{H}|1\rangle = |-, \mathbf{x}\rangle \mapsto |\searrow\rangle \end{cases}. \quad (6.10)$$

Circularly polarised photons would correspond to $|\pm, \mathbf{y}\rangle$ spin eigenstates, but will not be directly relevant for our discussion. The analogy is so complete that there are simple analogues of the Stern-Gerlach filter, indeed very simple to install on an optical table: optical polarisers. They are, essentially, polaroid lens: materials with an anisotropic chain-structure, so as to “absorb” photons whose polarisation corresponds to \mathbf{E} oscillating along the chain direction, while a photon would not be absorbed if its polarisation is orthogonal to the chain, the so-called *transmission axis*. If a photon of linear polarisation $\epsilon_{\alpha,0} = \hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha$, which we now denote as $|\epsilon_{\alpha,0}\rangle = \cos \alpha |\downarrow\rangle + \sin \alpha |\leftrightarrow\rangle$, travelling along the $\hat{\mathbf{z}}$ -axis, passes through a polariser with transmission axis along the $\hat{\mathbf{x}}$ direction, then it passes through, unaffected, with probability $\mathbb{P}_\alpha = \cos^2 \alpha$, and is absorbed with probability $\sin^2 \alpha$. The polariser then acts as a *measuring device* for linear polarisation, indeed a *filtering* measuring device. For instance, by sending a $|\mathbf{k} = k\hat{\mathbf{z}}, \downarrow\rangle$ photon into a polariser with transmission axis along $\epsilon_{\alpha,0}$ you would have a filtering measurement, and subsequent collapse, which might be described as:

$$|\mathbf{k} = k\hat{\mathbf{z}}, \downarrow\rangle \rightarrow \boxed{\text{Polariser transmission axis } \epsilon_{\alpha,0}} \begin{cases} \xrightarrow{\cos^2 \alpha} |\mathbf{k} = k\hat{\mathbf{z}}, \epsilon_{\alpha,0}\rangle \\ \xrightarrow{\sin^2 \alpha} \text{is absorbed.} \end{cases} \quad (6.11)$$

This is useful to *prepare* photons with a prescribed linear polarisation $\epsilon_{\alpha,0}$, see sketch in Fig. 6.4, even from an unpolarised source.

A second very useful piece of optical device is an optically anisotropic crystal like calcite or quartz, showing uniaxial birefringence. A birefringent crystal, see App. B for more details, shows a polarisation-dependent index of refraction. In a uniaxial birefringent crystal there is a particular *optic axis*, call it $\hat{\mathbf{y}}$, where the index of refraction is n_e , while in the other two directions, $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$, the index of refraction is n_o (the subscripts *e/o* stand for extra-ordinary/ordinary).⁴ At optical

⁴For calcite, CaCO_3 , $n_o = 1.658$ while $n_e = 1.486$. For quartz, SiO_2 , the difference is much smaller but in the opposite direction: $n_o = 1.544$ while $n_e = 1.553$.

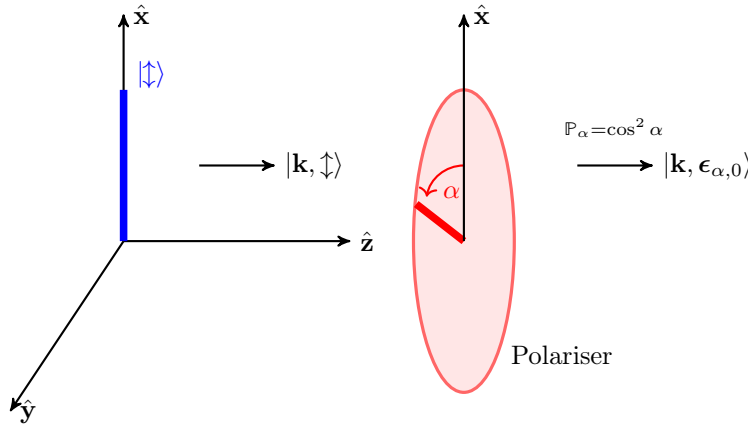


Figure 6.4: Preparation of a photon of given polarisation $|\epsilon_{\alpha,0}\rangle$ using a polaroid filter with transmission axis oriented along $\epsilon_{\alpha,0} = (\hat{x} \cos \alpha + \hat{y} \sin \alpha)$.

wavelengths you could take the dielectric function to be a constant, but anisotropic, matrix:

$$\bar{\epsilon} = \epsilon_0 \begin{pmatrix} n_o^2 & 0 & 0 \\ 0 & n_e^2 & 0 \\ 0 & 0 & n_o^2 \end{pmatrix}, \quad (6.12)$$

where ϵ_0 is the vacuum permittivity (I am using SI units). Depending on the relative orientation of the beam axis and of its polarisation with respect to the optic axis of the crystal, several modes of operations are possible.

One of the most interesting applications of uniaxial crystals is in a planar geometry where the wave enters the crystal orthogonally to the surface, and the optic axis is along the surface. This means that the wave suffers *no refraction*, and propagates along the \hat{z} axis. Fig. 6.5 illustrates the wave-plate geometry, with L the thickness of the crystal along the propagation direction. Solving Maxwell's

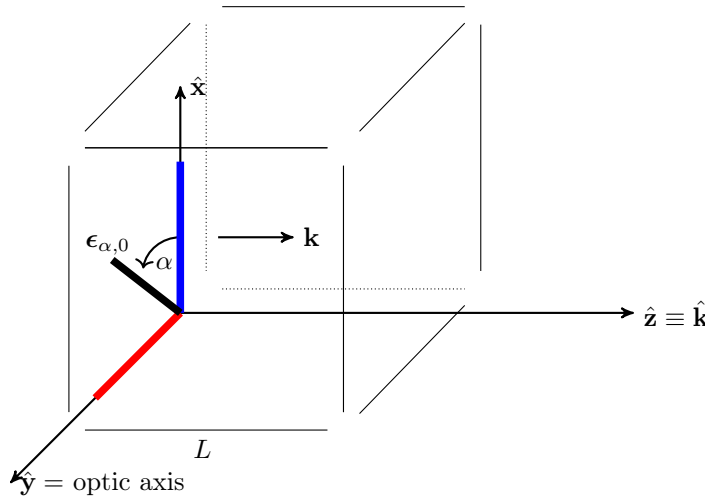


Figure 6.5: An electromagnetic plane-wave travelling along \hat{z} , with an incoming electric field $\mathbf{E} = \text{Re} \epsilon_{\alpha,0} e^{i(k_0 z - \omega t)}$ linearly polarised along $\epsilon_{\alpha,0} = \hat{x} \cos \alpha + \hat{y} \sin \alpha$, entering a uniaxial crystal of thickness L orthogonally to the surface where the optic axis lays. When exiting the polarisation is generally complex: $\epsilon_{\alpha,\delta} = \hat{x} \cos \alpha + \hat{y} e^{i\delta} \sin \alpha$, with $\delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0}$.

equations for a plane-wave in such a geometry is very simple, see App. B. The final polarisation is now generally *complex*

$$\epsilon_{\alpha,\delta} = \hat{x} \cos \alpha + \hat{y} e^{i\delta} \sin \alpha, \quad (6.13)$$

through a phase-factor $\delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0}$ which depends on the difference between the two refractive indices, and on the ratio between L , the thickness of the crystal, and λ_0 , the wave-length of the radiation in vacuum. Two cases are particular noteworthy. The first is known as *quarter-wave-plate*. In a quarter-wave-plate L is such that:

$$\delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0} = \pm \frac{\pi}{2}. \quad (6.14)$$

The \hat{x} and \hat{y} components of the field now advance out-of-phase by $\frac{\pi}{2}$. In the particularly important case in which the original polarisation was perfectly *diagonal*, $\alpha = \frac{\pi}{4}$, the exit polarisation is circular:

$$\epsilon_{\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{y}) \quad \rightarrow \quad \epsilon_{\frac{\pi}{4},\pm\frac{\pi}{2}} = \frac{1}{\sqrt{2}}(\hat{x} \pm i\hat{y}). \quad (6.15)$$

The second quite important case is that of a *half-wave-plate*. In a half-wave-plate L is twice as much as in the corresponding quarter-wave-plate:

$$\delta = 2\pi(n_e - n_o)\frac{L}{\lambda_0} = \pi. \quad (6.16)$$

The \hat{y} component is precisely reversed, hence $\alpha \rightarrow -\alpha$. In the particularly important case in which the original polarisation was perfectly *diagonal*, $\alpha = \frac{\pi}{4}$, the exit polarisation is anti-diagonal:

$$\epsilon_{\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{y}) \quad \rightarrow \quad \epsilon_{-\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} - \hat{y}). \quad (6.17)$$

It turns out that an appropriately oriented half-wave-plate, with its axis oriented as $\hat{x}_{\text{HWP}} = \hat{x} \cos \frac{\pi}{8} + \hat{y} \sin \frac{\pi}{8}$, plays the role of a Hadamard gate, see App. B for details, because it “mirrors” linear polarisations in the correct way.

The second important application of uniaxial birefringence is as an analogue of the *Stern-Gerlach* measurement apparatus for photon linear polarisation. By making the beam impinge at an angle θ_i with respect to the crystal surface normal, assuming the optic axis of the crystal to be parallel to the surface and the crystal to be thick enough, a classical beam of linear polarisation $\epsilon_{\alpha,0}$ is seen to be “split into two beams” of different polarisations \hat{x} and \hat{y} , due to the different Snell’s-law-induced refraction. See Figure 6.6. A single photon clearly is not “split”, but quantum mechanically goes *either* in the \hat{x} -polarised beam, with probability $\cos^2 \alpha$, *or* in the \hat{y} -polarised beam, with probability $\sin^2 \alpha$.

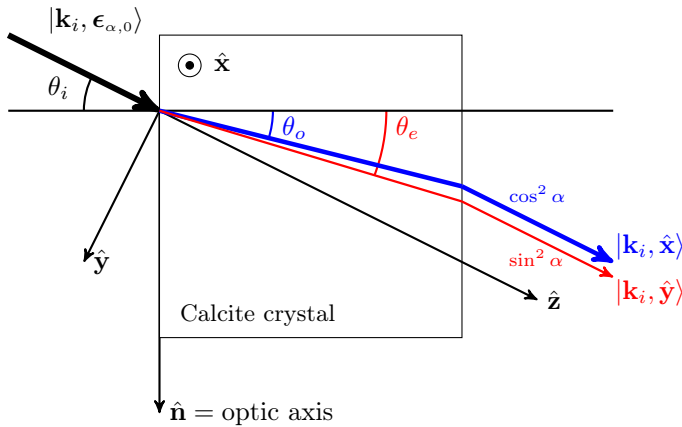


Figure 6.6: Polarisation-dependent refraction of a photon passing through a calcite crystal. We are assuming that the optic axis of the crystal is parallel to the crystal surface, as you would have for a wave-plate. The incoming momentum \mathbf{k}_i is now tilted at an angle θ_i with respect to the surface normal, provoking refraction of the incoming wave. Notice the orientation of the axes: the optic axis is now denoted by \hat{n} . The convention is identical to that of Ref. [29][Fig. 1.2].

So, summarising: we know how to prepare photon states in any of the states $|\uparrow\rangle$, $|\leftrightarrow\rangle$, $|\nearrow\rangle$ and $|\searrow\rangle$, and we even have Hadamard- \mathbf{H} gates, made of appropriately oriented half-wave-plates, to transform one polarisation into the other. This is enough to proceed in our discussion, with the extra remark that the momentum of the photon is actually not crucial in the story. Indeed, you can send the photons very far away, tens of kilometres (before they get absorbed), from the place where they were produced, through optical wave-guides. Sometimes even using the optical fibers that have been installed for commercial telecom applications. The polarisation of the photon is *preserved* inside the wave-guide, while the momentum obviously follows the complex path of the wave-guide. The only thing that the two experimentalists at far away stations have to agree is “what is the $|\uparrow\rangle$ direction” in their setup, and which individual photon they are talking about, i.e., they have to agree on the arrival time of the individual photons.

6.3. Exploiting the special nature of Quantum Randomness

Quantum measurements are as random as Nature can provide, according to QM. Still, the results depend on the basis that you use for measurements, an extra luxury which a classical random source would not provide. Preliminarily to discussing quantum key distribution protocols, let us comment on a few important points.

Suppose you want create a random string of bits, 0 and 1. How would you do that? There are classical methods, based on the electric noise in electronic devices, to do that.⁵ Let us see how one would do that with the randomness intrinsic in QM. You prepare photons all polarised at $\epsilon_{\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{y})$, for instance by making them pass through an appropriately oriented polariser. Then you measure the linear polarisation with a calcite crystal with optic axis oriented along \hat{y} . Photons will come out, randomly with probability $\frac{1}{2}$ and $\frac{1}{2}$, in either one of the two beams, polarised as $|\uparrow\rangle = |0\rangle$ or as $|\leftrightarrow\rangle = |1\rangle$. This is your random string of bits: computational Qbits in the standard \mathbf{Z} -basis.

It is *not* a good idea to send the Qbits so prepared to \mathbf{B} along the optical fiber: any measurement device of the \mathbf{Z} -type would be able to measure these Qbits, without collapsing them, because they are \mathbf{Z} -eigenstates.

But suppose that, randomly, \mathbf{A} applies a Hadamard transformation (\mathbf{H}) to the Qbits so prepared, transforming $|\uparrow\rangle \xrightarrow{\mathbf{H}} |\nearrow\rangle$ and $|\leftrightarrow\rangle \xrightarrow{\mathbf{H}} |\searrow\rangle$, before sending them through the optical fiber. Then, an intercepting device would not know if the Qbit is an eigenstate of \mathbf{Z} or an eigenstate of \mathbf{X} , because of the possible \mathbf{H} applied. Hence, the outcome of a measurement performed at the intercepting station would still be “0” or “1”, but the interceptor would have *no way* of knowing if this was because of a *certain outcome*, in case a matching measurement basis was used, or because of a random $\frac{1}{2} - \frac{1}{2}$ outcome followed by a *collapse*, in case the wrong measurement basis was used.

Changing preparation basis hides the information. So, by playing with the flexibility of applying unitaries and changing the preparation basis, \mathbf{A} can effectively “hide” the information on the random string of Qbits prepared.

Let us see more details of how you might do that, by reviewing the protocol invented by Bennett & Brassard [30] in 1984. See also Ref. [3][Sec.12.6.3]. Security of the BB84 protocol is discussed in Ref. [3][Sec.12.6.5].

Technical problems. I must warn you that the technicalities involved in practical Quantum Key Distributions are many: from the **photon sources** – usually not single-photon sources, but rather strongly attenuated laser pulses⁶ –, to **detection efficiency** of standard silicon photo-detectors, not to mention the necessity for **polarisation compensation** when transmitting photons via optical fibers. Alternatively to the polarization coding, techniques based on **phase coding**, using Mach-Zehnder interferometric techniques, see [2][Sec.10.4.3], are also used. All of these practicalities can in principle threaten the theoretical security of the protocol, and they have been addressed in many papers. I will skip these details, giving only the general idea of the protocol.

⁵See inset of Fig. 1.11, taken from Ref. [14].

⁶Single photon sources were invented by A. Aspect in 1985. They should be distinguished from strongly attenuated photon sources. If a beam is strongly attenuated so that the average number of photons is very small, say $\langle n \rangle = \frac{1}{100}$, then 99% of the time there is no photon, in 1% of the cases there is one photon, but, with a Poisson’s distribution, there could also be 2, 3, etc. photons, and the coincidence counts would reveal the subtle difference with a true single-photon source. See A. Aspect’s public lecture upon receiving the N. Bohr Gold Medal 2013, available on [YouTube](#).

6.3.1. The BB84 protocol

The first thing that **A** and **B** should agree is their “measurement” directions. Next, they have to set a common timing: photons have to be sent at rate such that **B** can distinguish them, and one has to have a clear way of establishing “which photon is which”, by a commonly established clock. Notice that some photons might be lost for different reasons (channel losses, interceptions), but if a clear timing is set, **B** will always know that “photon 2 did not arrive, but here is photon 3”.

A sends to **B** photon states which are chosen from the 4 possible preparations: **Z**-states, $|0\rangle$ or $|1\rangle$, or **X**-states, $\mathbf{H}|0\rangle$ or $\mathbf{H}|1\rangle$. Recall that one possibility is that **A** first prepares the random string by measuring along **Z** photons with polarization $|\nearrow\rangle$, and then randomly applies **H** to the **Z**-eigenstates obtained by the measurement, transforming some of the photons back to $|\nearrow\rangle$ or $|\searrow\rangle$. Upon receiving a photon, **B** can choose to directly measure the polarisation in the **Z**-direction, or rather in the **X**-direction. Here is a scheme of a possible sequence of photons prepared by **A** and sent to **B**.

Photon N. →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Quantum Transmitter														
A random bits:	0	1	1	0	1	1	0	0	1	0	1	1	0	0	1
A P-basis:	X	Z	X	Z	Z	Z	Z	Z	X	X	Z	X	X	X	Z
A P-state:	$ \nearrow\rangle$	$ \leftrightarrow\rangle$	$ \searrow\rangle$	$ \updownarrow\rangle$	$ \leftrightarrow\rangle$	$ \leftrightarrow\rangle$	$ \updownarrow\rangle$	$ \updownarrow\rangle$	$ \searrow\rangle$	$ \nearrow\rangle$	$ \leftrightarrow\rangle$	$ \searrow\rangle$	$ \nearrow\rangle$	$ \nearrow\rangle$	$ \leftrightarrow\rangle$
	Quantum Receiver														
B M-basis:	Z	X	X	Z	Z	X	X	Z	X	Z	X	X	X	X	Z
B M-eigenvalue:	1	nr	1	nr	1	0	0	0	nr	1	1	1	nr	0	1
	Classical (authenticated) communication														
B reports basis:	Z		X		Z	X	X	Z		Z	X	X		X	Z
A confirms basis:			ok		ok			ok			ok			ok	ok
Raw key:			1		1			0			1			0	1
B reveals bits:					1									0	
A confirms bits:					ok									ok	
	Final key → Information reconciliation & privacy amplification														
Secret bits:			1					0					1		1

A few comments and explanations about this table.

Q-T) The first 3 rows describe the state preparation in **A**'s lab. A random string of bits leads to a precise set of photon states (P-state) sent along the line, depending on the preparation basis (P-basis).

Q-R) The next 2 rows describe the reception of photons at **B**'s lab. At least with our current technology, **B** cannot store the photons received. Hence, a measurement should be done immediately, with a basis randomly decided by **B**, noting down measurement basis-type (M-basis) and eigenvalue result (M-eigenvalue). In the example, photons n. 2, 4, 9 and 13 were not received (nr) by **B**.

CC-1) By using a normal authenticated (so that **B** knows that is communicating with **A**) classical communication (CC), **B** reveals to **A** the measurement basis (M-basis) of all the photons received.

CC-2) **A** replies by stating which of the measurement-basis agree with the preparation-basis.

Raw key) At this point there is a substring of bits which “in principle” agree in the two labs, denoted as “Raw key”.

CC-3) For different reasons, there might have been errors (or interceptions). To gauge the statistics

of the possible errors, a fractions of \mathbf{B} 's bits measurements (M-eigenvalue) is communicated to \mathbf{A} .

CC-4) \mathbf{A} confirms (or not) that the measured eigenvalues indeed agree. This leads to the estimate of an error rate r .

Final key) The remaining bits (those not revealed) are secretly shared between \mathbf{A} and \mathbf{B} , but might contain some (estimated to be $\sim r$) fraction of errors. After that, a phase known as *information reconciliation* and *privacy amplification* — essentially, classical error correction protocols — follows.

Randomness of preparation is crucial: if \mathbf{E} knows that type- \mathbf{Z} is always prepared, in its possible variants 0 and 1, then \mathbf{E} measures in the \mathbf{Z} basis without disturbing (certain outcome) and sends to \mathbf{B} . This strategy is avoided if \mathbf{A} can randomly change the preparation basis.

The best \mathbf{E} can do is to measure, randomly, either type- \mathbf{Z} or type- \mathbf{X} . But this can be spotted by taking a fraction of the supposedly matched measurements and exchanging info on the state as well.

6.3.2. Important details

As we already remarked, there is a wealth of important technical details that you should take care of before the idea behind BB84 becomes a working piece of apparatus. This is the usual gap between physical ideas and engineering implementations. These details, important as they are, are not quantum. Some of them, listed below, pertain to the world of classical information theory, and must be applied to the raw key given by the BB84 algorithm.

Information reconciliation is essentially a classical error-correcting protocol which eliminates all the residual errors in the raw key, due to either experimental imperfections or to external attacks, i.e., measurements performed by third parties during the quantum transmission phase. The more the technology evolves, the more it is important to do this in a highly efficient manner. For our purposes, it suffices here to say that *parity checks* are often used. Read Ref. [3][Sec.12.6.2] for a general presentation, and Ref. [31] for a discussion of some technical points.

Privacy amplification wants to guarantee that the final key obtained — after correcting the errors in the raw key — is indeed secure. The problem is that during the error correction phase, which occurs on a public authenticad channel, information on the key — for instance, parity bits — might be intercepted by third parties. The goal is to effectively generate a new shorter key, out of the error-corrected one, for which third parties have effectively no information. This is done by applying algorithms known in mathematics and in computer science as **universal hash functions**. Again, Ref. [3][Sec.12.6.2] is a good starting point for a general introduction.

6.4. Exploiting quantum correlations due to entanglement

There is a second protocol, introduced in 1991 by Arthur Ekert [32] and known as E91, for Quantum Key Distribution. Unlike BB84, this protocol is more symmetric, and is based on entangled pairs of photons being sent to \mathbf{A} and \mathbf{B} , with polarisation measurements performed at the two stations.

The source emits pairs of entangled particles, one going to station \mathbf{A} , the other to station \mathbf{B} . In terms of spins, let us say that the particles are in a singlet state, and that:

$$|\psi_{\text{ent}}\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle_A \otimes |\downarrow\rangle_B - |\downarrow\rangle_A \otimes |\uparrow\rangle_B) = \frac{1}{\sqrt{2}}(|+, \mathbf{n}\rangle_A \otimes |-, \mathbf{n}\rangle_B - |-, \mathbf{n}\rangle_A \otimes |+, \mathbf{n}\rangle_B), \quad (6.18)$$

where the second expression follows from rotational invariance, and guarantees that we can consider equally well spin states in any direction \mathbf{n} , rather than those in the \mathbf{z} -direction. More practically, in a Quantum Optics lab, the source emits pairs of polarization-entangled photons, obtained by a non-linear process known as *spontaneous parametric down-conversion*. In terms of photon polarization states, you would write:

$$|\psi_{\text{ent}}\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B), \quad (6.19)$$

with the usual identification of $|0\rangle = |\uparrow\rangle = |V\rangle$ and $|1\rangle = |\leftrightarrow\rangle = |H\rangle$.

6.4.1. CHSH version of Bell's inequalities

Suppose at stations **A** and **B** the “spin” (or polarization) is measured in directions \mathbf{a} and \mathbf{b} , respectively, so that the relevant operator is $(\hat{\sigma}_A \cdot \mathbf{a}) \otimes (\hat{\sigma}_B \cdot \mathbf{b})$. It might be useful to refresh the calculations in Sec. 1.4, although I will rederive what we need.

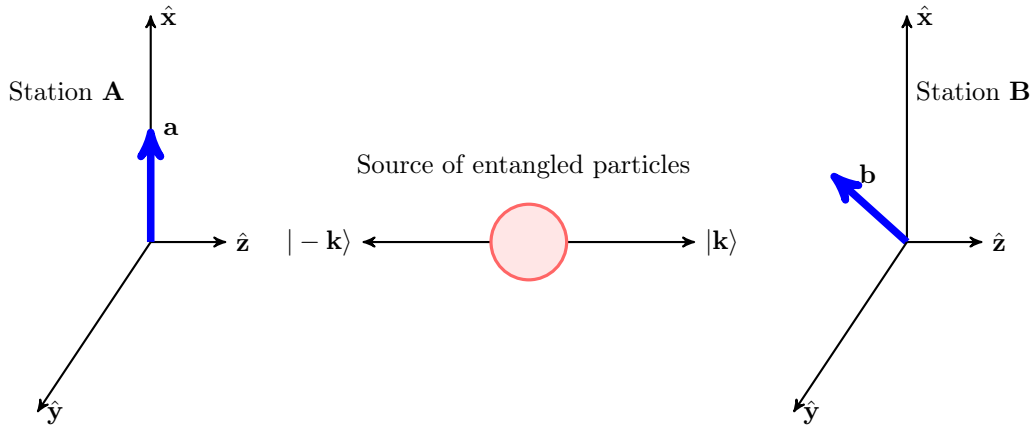


Figure 6.7.: Schematics of an EPR experiment. Two particles with opposite momenta and spin-entangled (or polarization-entangled, for photons) are sent to two far-away experimental stations **A** and **B** where their spin (or polarization) is measured into two different directions \mathbf{a} , at **A**, and \mathbf{b} , at **B**. The combined operator that is being measured is $(\hat{\sigma}_A \cdot \mathbf{a}) \otimes (\hat{\sigma}_B \cdot \mathbf{b})$.

As you recall, individual spin measurements in any directions will always give eigenvalues ± 1 , with a collapse of the state on the corresponding eigenstates. Hence, both **A** and **B** will obtain eigenvalues $\sigma_{\mathbf{a}} = \pm 1$ and $\sigma_{\mathbf{b}} = \pm 1$, collapsing the state $|\psi\rangle$ into $|\sigma_{\mathbf{a}}, \mathbf{a}\rangle \otimes |\sigma_{\mathbf{b}}, \mathbf{b}\rangle$, a product state which we will denote, for shortness, as $|\sigma_{\mathbf{a}}; \sigma_{\mathbf{b}}\rangle$:

$$|\psi\rangle \xrightarrow{\text{measure } \sigma_{\mathbf{a}}, \sigma_{\mathbf{b}} \text{ with } \mathbb{P}_{\sigma_{\mathbf{a}}, \sigma_{\mathbf{b}}}} |\sigma_{\mathbf{a}}, \mathbf{a}\rangle \otimes |\sigma_{\mathbf{b}}, \mathbf{b}\rangle. \quad (6.20)$$

According to the rules of QM, the probability of a joint measurement of $(\sigma_{\mathbf{a}}, \sigma_{\mathbf{b}})$ on the state $|\psi\rangle$ is given by $\mathbb{P}_{\sigma_{\mathbf{a}}, \sigma_{\mathbf{b}}} = |\langle \sigma_{\mathbf{a}}; \sigma_{\mathbf{b}} | \psi \rangle|^2$. The expectation value for the combined measurement of the spins in the two directions on a state $|\psi\rangle$ is given (by the usual trick of inserting identities) by:

$$\begin{aligned} E(\mathbf{a}, \mathbf{b}) &= \langle \psi | (\hat{\sigma}_A \cdot \mathbf{a}) \otimes (\hat{\sigma}_B \cdot \mathbf{b}) | \psi \rangle \\ &= \sum_{\sigma_{\mathbf{a}}, \sigma_{\mathbf{b}}} \langle \psi | \sigma_{\mathbf{a}}; \sigma_{\mathbf{b}} \rangle \langle \sigma_{\mathbf{a}}; \sigma_{\mathbf{b}} | (\hat{\sigma}_A \cdot \mathbf{a}) \otimes (\hat{\sigma}_B \cdot \mathbf{b}) | \sigma_{\mathbf{a}}; \sigma_{\mathbf{b}} \rangle \langle \sigma_{\mathbf{a}}; \sigma_{\mathbf{b}} | \psi \rangle \\ &= \sum_{\sigma_{\mathbf{a}}, \sigma_{\mathbf{b}}} \mathbb{P}_{\sigma_{\mathbf{a}}, \sigma_{\mathbf{b}}} \sigma_{\mathbf{a}} \sigma_{\mathbf{b}} = \mathbb{P}_{+,+} + \mathbb{P}_{-,-} - \mathbb{P}_{+,-} - \mathbb{P}_{-,+}. \end{aligned} \quad (6.21)$$

The last expression gives the experimental procedure to obtain the expectation value, by collecting the statistics of (infinitely many, in principle) measurements. This is true for any state $|\psi\rangle$.

On the other hand, by using for $|\psi\rangle$ the second expression for $|\psi_{\text{ent}}\rangle$ in Eq. (6.18), with $\mathbf{n} = \mathbf{a}$, and the fact that:

$$\langle \pm, \mathbf{a} | \hat{\sigma} \cdot \mathbf{b} | \pm, \mathbf{a} \rangle = \pm \mathbf{a} \cdot \mathbf{b} ,$$

it is very simple to show that for the entangled state:

$$E(\mathbf{a}, \mathbf{b}) = \langle \psi_{\text{ent}} | (\hat{\sigma}_A \cdot \mathbf{a}) \otimes (\hat{\sigma}_B \cdot \mathbf{b}) | \psi_{\text{ent}} \rangle = -\mathbf{a} \cdot \mathbf{b} . \quad (6.22)$$

The classical perspective. From a classical viewpoint, the measurement of an operator A is not deterministic simply because there might be *hidden variables*, collectively denoted by λ , which we do not control, leading to the result $A(\lambda)$. I invite you to carefully read the original paper, Ref. [33], where the physics is carefully discussed. Here, I simplify a bit the derivation, following Refs. [2, 3]. If $P(\lambda)$ denotes the probability distribution of the hidden variables λ , then you would predict a classical expectation value

$$E(A) = \int d\lambda P(\lambda) A(\lambda) .$$

Suppose now that the two operators measured at two different stations are spins along directions \mathbf{a} and \mathbf{b} , as before, and denote their measurements as $A_{\mathbf{a}}(\lambda)$ and $B_{\mathbf{b}}(\lambda)$. Clearly, $A_{\mathbf{a}}(\lambda) = \pm 1$ and $B_{\mathbf{b}}(\lambda) = \pm 1$. Then you would predict a classical expectation value:

$$E(\mathbf{a}, \mathbf{b}) = E(A_{\mathbf{a}} B_{\mathbf{b}}) = \int d\lambda P(\lambda) A_{\mathbf{a}}(\lambda) B_{\mathbf{b}}(\lambda) . \quad (6.23)$$

Consider now (omitting the variable λ , for brevity) measurements in different direction, more precisely:

$$\begin{aligned} A_{\mathbf{a}} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}'} - A_{\mathbf{a}} B_{\mathbf{b}'} &= \underbrace{(A_{\mathbf{a}} + A_{\mathbf{a}'})}_{\text{agree: } \pm 2 \quad \text{disagree: } 0} B_{\mathbf{b}} + \underbrace{(A_{\mathbf{a}'} - A_{\mathbf{a}})}_{\text{agree: } 0 \quad \text{disagree: } \pm 2} B_{\mathbf{b}'} \\ &= \pm 2 , \end{aligned} \quad (6.24)$$

where we used the fact that either $A_{\mathbf{a}} = A_{\mathbf{a}'}$ (they agree), hence $A_{\mathbf{a}'} - A_{\mathbf{a}} = 0$ and $A_{\mathbf{a}'} + A_{\mathbf{a}} = \pm 2$, or, viceversa, $A_{\mathbf{a}} = -A_{\mathbf{a}'}$ (they disagree), hence $A_{\mathbf{a}'} - A_{\mathbf{a}} = \pm 2$ and $A_{\mathbf{a}'} + A_{\mathbf{a}} = 0$. In all cases, the result is by inspection equal to ± 2 . Hence we can write the following inequality (omit again indicating the λ -dependence of the quantities inside the integral):

$$\begin{aligned} \left| E(A_{\mathbf{a}} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}'} - A_{\mathbf{a}} B_{\mathbf{b}'}) \right| &= \left| \int d\lambda P(\lambda) (A_{\mathbf{a}} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}'} - A_{\mathbf{a}} B_{\mathbf{b}'}) \right| \\ &\leq \int d\lambda P(\lambda) |A_{\mathbf{a}} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}} + A_{\mathbf{a}'} B_{\mathbf{b}'} - A_{\mathbf{a}} B_{\mathbf{b}'}| \\ &= 2 \int d\lambda P(\lambda) = 2 , \end{aligned} \quad (6.25)$$

where we used the integral inequality $|\int d\lambda f(\lambda)| \leq \int d\lambda |f(\lambda)|$, together with the normalization of $P(\lambda)$.

❶

The Clauser, Horne, Shimony and Holt (CHSH) inequality. We conclude that the quantity:

$$C(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') = E(\mathbf{a}, \mathbf{b}) + E(\mathbf{a}', \mathbf{b}) + E(\mathbf{a}', \mathbf{b}') - E(\mathbf{a}, \mathbf{b}') , \quad (6.26)$$

is such that it should obey, classically, the inequality:

$$|C(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}')| = |E(\mathbf{a}, \mathbf{b}) + E(\mathbf{a}', \mathbf{b}) + E(\mathbf{a}', \mathbf{b}') - E(\mathbf{a}, \mathbf{b}')| \leq 2 . \quad (6.27)$$

The remarkable fact is that there are directions $(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}')$ that *violate* this inequality, when expectation values are calculated quantum mechanically. For instance, take all vectors in the xy

plane, as $\mathbf{n} = \hat{\mathbf{x}} \cos \phi + \hat{\mathbf{y}} \sin \phi$ with: $\phi_{\mathbf{a}} = 0$, $\phi_{\mathbf{b}} = \frac{\pi}{4}$, $\phi_{\mathbf{a}'} = \frac{\pi}{2}$ and $\phi_{\mathbf{b}'} = \frac{3\pi}{4}$. Then, you immediately calculate:

$$C_{\text{quantum}}(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') = -\mathbf{a} \cdot (\mathbf{b} - \mathbf{b}') - \mathbf{a}' \cdot (\mathbf{b}' + \mathbf{b}) = -2\sqrt{2}, \quad (6.28)$$

which violates the inequality. Figure 6.8 illustrates this choice.

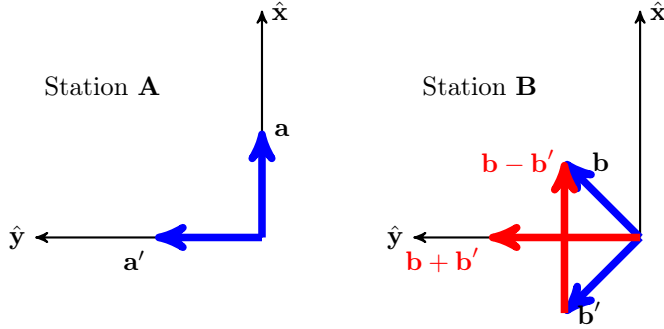


Figure 6.8: The choice of unit vectors \mathbf{a} , \mathbf{b} , \mathbf{a}' and \mathbf{b}' which leads to $C_{\text{quantum}}(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') = -2\sqrt{2}$, thus violating the CHSH inequality.

6.4.2. The E91 protocol

The source sends polarization-entangled pairs of photons to the two stations at **A** and **B**. At **A**, polarization measurements are performed in one of the three directions \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 , with $\phi_{\mathbf{a}_1} = 0$, $\phi_{\mathbf{a}_3} = \frac{\pi}{2}$ (as discussed before, for the CHSH inequality) and $\phi_{\mathbf{a}_2} = \frac{\pi}{4}$. At **B**, polarization measurements are performed in one of the three directions \mathbf{b}_1 , \mathbf{b}_2 , and \mathbf{b}_3 , with $\phi_{\mathbf{b}_1} = \frac{\pi}{4}$, $\phi_{\mathbf{b}_3} = \frac{3\pi}{4}$ (as discussed for the CHSH inequality) and $\phi_{\mathbf{b}_2} = \frac{\pi}{2}$. Notice that there are directions which are definitely different, but

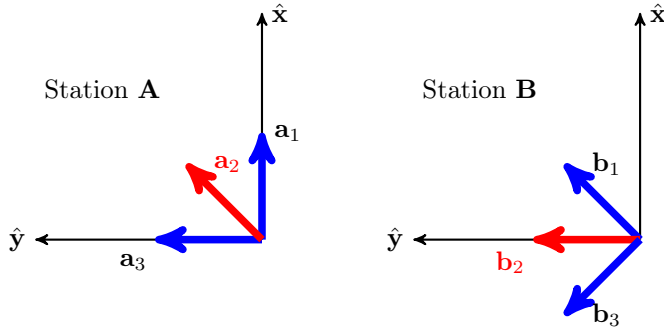


Figure 6.9: The choice of unit vectors \mathbf{a}_1 , \mathbf{a}_2 , \mathbf{a}_3 and \mathbf{b}_1 , \mathbf{b}_2 , \mathbf{b}_3 randomly used at the two stations as measuring directions. Notice that the blue vectors \mathbf{a}_1 , \mathbf{a}_3 and \mathbf{b}_1 , \mathbf{b}_3 are different, and coincide with the directions used in the CHSH inequality. The two new directions (in red) \mathbf{a}_2 and \mathbf{b}_2 introduce common measuring directions, since $\mathbf{a}_3 = \mathbf{b}_2$ and $\mathbf{a}_2 = \mathbf{b}_1$: measurement results obtained for those choices are perfectly correlated, and used as a secret shared raw key.

there are now also directions that match: indeed, $\mathbf{a}_2 = \mathbf{b}_1$ and $\mathbf{b}_2 = \mathbf{a}_3$. Hence, quantum mechanically, we are guaranteed that

$$E(\mathbf{a}_2, \mathbf{b}_1) = E(\mathbf{a}_3, \mathbf{b}_2) = -1, \quad (6.29)$$

i.e., the measurement performed at the two stations would be perfectly correlated: if **A** measures $+1$, then **B** measures -1 and viceversa. The protocol then proceeds as follows:

- 1) After measuring many times in one of the directions \mathbf{a}_j and \mathbf{b}_j , **A** and **B** communicate on a line the directions they have used.
- 2) After that, they communicate also the results of the measurements when the two directions are *different*. With these, they can both calculate and check that:

$$C_{\text{quantum}}(\mathbf{a}_1, \mathbf{b}_1, \mathbf{a}_3, \mathbf{b}_3) = -2\sqrt{2}. \quad (6.30)$$

- 3) The measurements along the same directions are not shared, but they are guaranteed to be perfectly correlated: hence, this is a shared secret raw key.

- 4) As for the BB84 protocol, this shared *raw key* is subject to the usual classical procedures of information reconciliation (i.e., error correction) and privacy amplification.

The technology to perform this experiment is very close to that employed by A. Aspect *et al.*, see Ref. [4], to test the violation of the CHSH inequality, and the strange non-local nature of Quantum Mechanics.

7. Hardware implementations of Quantum Computers



Warning: This chapter is still in a very preliminary version.

Depending on the *hardware* on which we would base our Quantum Computer, the nature of the “spin-1/2” Qbit changes. Correspondingly, the basic quantum gates that are easy to implement are also modified. The proposals that are currently investigated ¹ fall into three classes: 1) trapped atoms or ions, 2) photons, and 3) solid state QBits.

- 1. Atoms or Ions)** Quantum optics and the control of atom-photon interactions have advanced tremendously since the middle 1980’s. The two levels which make the Qbit are simply two levels of an atom or ion, the ground state and some excited state, which are selectively controlled and manipulated with the use of coherent radiation. It is mandatory that atoms or ions are kept fixed in space, by some form of trapping. Trapping atoms in optical lattices has created a sufficiently scalable platform for the coherent manipulation of many atoms. Atoms that can stay in long-lived highly excited **Rydberg states** are also natural candidates for two-level atoms. This is also a very promising platform, especially for *Quantum Simulators*. Finally, **ions**, for instance $^{40}\text{Ca}^+$, which are then trapped with radio-frequency traps (**Paul traps**) and manipulated with coherent radiation, are also currently used. Blatt’s group experimental expertise at U. of Innsbruck has lead — through their spin-off **Alpine Quantum Technology** — to a commercial general purpose Quantum Computer based on trapped ions.
- 2. Photons)** An all-optical implementation of the Qbit exploits the two polarisation states of photons. Beam-splitter, mirrors, polarisers, non-linear crystals generating entangled photons and all these machinery allow a great flexibility, which enjoys also from the fact that photons suffer very little from interactions with external agents that could lead to loss of coherence. See, for instance, the **Xanadu** website. Boson sampling (Aaronson) should be also mentioned.
- 3. Solid state Qbits)** This is the platform which has more overlap with traditional solid-state systems. Nano-structures have been considered since the 1990’s, due to technological advances in material fabrication. There is in principle an unlimited scalability, but the different QBits will never be totally identical, unlike atoms/ions or photons. Strong local correlation make the spectrum an-harmonic; maintaining coherence is a non-trivial achievement, due to the extensive wiring that these platforms involve. Currently, the most promising example in this class is that of **superconducting Qbits**.

In the following, we will concentrate on Superconducting QBits platforms. Particularly useful are two review papers: Refs. [17, 34]. Technical aspects of quantum circuitry are discussed in Ref. [35]. But before diving into the details, let us review some general ideas.

¹We do not discuss the early proposal of using the spin of molecules in an NMR setting, which is not scalable and abandoned.

7.1. DiVincenzo criteria

In 2000, David DiVincenzo formulated a few general criteria that a system should satisfy to qualify as a sensible Quantum Computation platform. They are known as **DiVincenzo criteria**.

- 1) You need a scalable physical system of QBits, i.e., quantum two-level systems. ²
- 2) The initial state $|\Psi_{\text{in}}\rangle$ should be easy to prepare.
- 3) You need long relevant decoherence times, so that the time-evolution is, as much as possible, unitary.
- 4) You need a “universal” set of quantum gates.
- 5) A QBit-specific measurement capability. One should be able to do **measurements** on the final state $|\Psi_{\text{fin}}\rangle$ reached, in order to “read” the answer of the Quantum Computation.

7.2. A few tools: LC circuits, Josephson’s Junctions, SQUIDS

To understand the dynamics of a superconducting QBit, we need to start from the very basic example of a linear LC resonator, made by a capacitor and an inductor in parallel. The energy accumulated in the capacitor is periodically passed to the inductor, back and forth, without loss, as we assume that the circuit has no resistance. Let $V(t)$ be the voltage across the capacitor, of capacitance C , and $I(t)$ the current flowing in the circuit, and through the inductor of inductance L . If $Q(t)$ is the charge in one of the plates of the capacitor, you know that $I = \dot{Q}$. If $\Phi(t)$ is the magnetic flux inside the inductor, then $\dot{\Phi} = V$, by Maxwell-Faraday’s law: the voltage across the inductor is associated to the flux derivative. Moreover, the capacitance governs the relationship between Q and V , and the inductance the relationship between I and Φ :

$$Q = VC \quad \text{and} \quad \Phi = LI . \quad (7.1)$$

We will now write everything in terms of the flux Φ , which we will regard as the basic “position variable”. You know that the charging energy of the capacitor is

$$K_C = \frac{1}{2C}Q^2 = \frac{1}{2}CV^2 = \frac{1}{2}C\dot{\Phi}^2 ,$$

which looks as a “kinetic energy” for a particle of mass C and velocity $\dot{\Phi}$. Similarly, the magnetic energy of the inductor is

$$U_L = \frac{1}{2}LI^2 = \frac{1}{2L}\Phi^2 ,$$

showing a “spring constant” $1/L$. To write the equations of motion, we start from the Lagrangian:

$$\mathcal{L}(\Phi, \dot{\Phi}) = K_C - U_L = \frac{1}{2}C\dot{\Phi}^2 - \frac{1}{2L}\Phi^2 . \quad (7.2)$$

The canonical momentum is calculated as

$$\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = C\dot{\Phi} = CV = Q . \quad (7.3)$$

²There are proposals based on three-level systems, or other choices, but we will be confined to two-level systems, the QBits. Obviously, this does not mean that the spectrum of the local degree of freedom is strictly made by two levels only: it is enough that two among the possible large number of levels are clearly identifiable (hence, you need an anharmonic spectrum) and there are tools to address them specifically, so that the other levels can be approximately ignored.

Hence, we write the Hamiltonian as:

$$H(\Phi, Q) = Q\dot{\Phi} - \mathcal{L} = \frac{1}{2C}Q^2 + \frac{1}{2L}\Phi^2. \quad (7.4)$$

This is obviously an harmonic oscillator. If you regard Q as the canonical momentum with C acting as a mass, and you write the potential energy as $\frac{1}{2}C\omega_r^2\Phi^2$, you immediately see that the resonance frequency is:

$$\omega_r = \frac{1}{\sqrt{LC}}. \quad (7.5)$$

To quantize it, you apply the standard procedure: regard the quantities as operators with $\hat{\Phi}$ and \hat{Q} obeying canonical commutation relations. It is convenient to rescale both \hat{Q} and $\hat{\Phi}$ to make them dimensionless: if e is the electronic charge, and h Plank's constant, the flux quantum is $\Phi_0 = h/(2e)$, and we set:

$$\hat{n} = \frac{\hat{Q}}{2e} \quad \hat{\phi} = 2\pi\frac{\hat{\Phi}}{\Phi_0} \quad \text{with} \quad \Phi_0 = \frac{h}{2e} \quad \implies \quad [\hat{\phi}, \hat{n}] = i. \quad (7.6)$$

The Hamiltonian then reads:

$$\hat{H} = \frac{1}{2C}\hat{Q}^2 + \frac{1}{2L}\hat{\Phi}^2 = 4E_C\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2, \quad (7.7)$$

where the energy constants E_C and E_L are:

$$E_C = \frac{e^2}{2C} \quad E_L = \frac{1}{L} \frac{\Phi_0^2}{(2\pi)^2}. \quad (7.8)$$

To express the Hamiltonian in terms of creation and destruction operators, it is useful to further rescale $\hat{\phi}$ and \hat{n} in such a way that the two terms of the Hamiltonian have a common form. More precisely, imagine rescaling:

$$\hat{\phi} = \phi_{zp}\hat{x} \quad \text{and} \quad \hat{n} = \frac{1}{\phi_{zp}}\hat{p},$$

where $[\hat{x}, \hat{p}] = i$, and choose the dimensionless constant ϕ_{zp} in such a way that:

$$4E_C\frac{1}{\phi_{zp}^2} = \frac{1}{2}E_L\phi_{zp}^2 \equiv \frac{1}{2}\hbar\omega_r. \quad (7.9)$$

This requires:

$$\phi_{zp}^2 = \sqrt{\frac{8E_C}{E_L}} \quad \implies \quad \hbar\omega_r = \sqrt{8E_LE_C} = \hbar\frac{1}{\sqrt{LC}}. \quad (7.10)$$

With this choice, and the standard representation $\hat{x} = \frac{1}{\sqrt{2}}(\hat{a} + \hat{a}^\dagger)$ and $\hat{p} = \frac{1}{\sqrt{2}i}(\hat{a} - \hat{a}^\dagger)$ we have all ingredients of our quantum harmonic oscillator.

i **A quantum LC circuit.** The harmonic oscillator Hamiltonian corresponding to the quantization of an LC circuit is given by:

$$\hat{H} = \frac{1}{2C}\hat{Q}^2 + \frac{1}{2L}\hat{\Phi}^2 = 4E_C\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2 = \frac{1}{2}\hbar\omega_r(\hat{p}^2 + \hat{x}^2) = \hbar\omega_r(\hat{a}^\dagger\hat{a} + \frac{1}{2}). \quad (7.11)$$

The problem with the spectrum of an harmonic oscillator is that there is no way to selectively address *two* states only, out of the infinite many. We will need some non-linearity, such that the gap between the lowest two states is different from the gap with the higher states. In the next section we will see that a Josephson junction provides precisely a controllable degree of non-linearity.

7.2.1. From BCS to the Josephson junction

In order to discuss the non-linearity induced by the Josephson effect, we need to discuss superconductivity first.

BCS superconductivity. Let me briefly recall the celebrated Bardeen-Cooper-Schrieffer solution of the superconducting state. Given a (mean-field) Hamiltonian of the form:

$$\hat{H}_{\text{BCS}} = \sum_{\mathbf{k}, \sigma} \xi_{\mathbf{k}} \hat{c}_{\mathbf{k}\sigma}^\dagger \hat{c}_{\mathbf{k}\sigma} - \sum_{\mathbf{k}} (\Delta_{\mathbf{k}} \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger + \Delta_{\mathbf{k}}^* \hat{c}_{-\mathbf{k}\downarrow} \hat{c}_{\mathbf{k}\uparrow}) \quad (7.12)$$

with $\Delta_{\mathbf{k}} = \Theta(\hbar\omega_D - \xi_{\mathbf{k}}) e^{i\varphi} \Delta$ and $\Delta > 0$, corresponding to an s-wave superconductor, its ground state is given by the BCS state:

$$|\Psi_{\text{BCS}}(\varphi)\rangle = \prod_{\mathbf{k}} \left(u_{\mathbf{k}} + e^{i\varphi} v_{\mathbf{k}} \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger \right) |0\rangle. \quad (7.13)$$

Here $u_{\mathbf{k}}$ and $v_{\mathbf{k}}$ are real:

$$\begin{cases} v_{\mathbf{k}}^2 &= \frac{1}{2} \left(1 - \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \right) \\ u_{\mathbf{k}}^2 &= \frac{1}{2} \left(1 + \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \right) \end{cases}, \quad (7.14)$$

with $u_{\mathbf{k}}^2 + v_{\mathbf{k}}^2 = 1$, and $E_{\mathbf{k}} = \sqrt{\xi_{\mathbf{k}}^2 + |\Delta_{\mathbf{k}}|^2}$ is the Bogoljubov quasi-particle energy. The phase φ plays a relatively minor role if you deal with a single superconductor, but will play a very important role in describing the Josephson tunnelling between *two superconductors* separated by a thin insulating (oxide) layer.

Evidently $|\Psi_{\text{BCS}}(\varphi)\rangle$ describe a superposition of states with all possible (even) fermion number. To simplify our writing, let us denote by $\hat{b}_{\mathbf{k}}^\dagger = \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger$ the operator that creates a pair of fermions in the Cooper-pair state ($\mathbf{k}\uparrow, -\mathbf{k}\downarrow$). By expanding the factor $\prod_{\mathbf{k}} (u_{\mathbf{k}} + e^{i\varphi} v_{\mathbf{k}} \hat{b}_{\mathbf{k}}^\dagger)$ you can write:

$$\begin{aligned} |\Psi_{\text{BCS}}(\varphi)\rangle &= \left(\prod_{\mathbf{k}} u_{\mathbf{k}} \right) \left(|0\rangle + e^{i\varphi} \sum_{\mathbf{k}_1} \frac{v_{\mathbf{k}_1}}{u_{\mathbf{k}_1}} \hat{b}_{\mathbf{k}_1}^\dagger |0\rangle + e^{2i\varphi} \sum_{(\mathbf{k}_1, \mathbf{k}_2)} \frac{v_{\mathbf{k}_1}}{u_{\mathbf{k}_1}} \frac{v_{\mathbf{k}_2}}{u_{\mathbf{k}_2}} \hat{b}_{\mathbf{k}_1}^\dagger \hat{b}_{\mathbf{k}_2}^\dagger |0\rangle + \dots \right. \\ &\quad \left. + e^{2ni\varphi} \sum_{(\mathbf{k}_1, \dots, \mathbf{k}_n)} \frac{v_{\mathbf{k}_1}}{u_{\mathbf{k}_1}} \dots \frac{v_{\mathbf{k}_n}}{u_{\mathbf{k}_n}} \hat{b}_{\mathbf{k}_1}^\dagger \dots \hat{b}_{\mathbf{k}_n}^\dagger |0\rangle + \dots \right) \\ &= \sum_{n=0}^{\infty} e^{in\varphi} A_n |\Psi_n\rangle \end{aligned} \quad (7.15)$$

where the notation $(\mathbf{k}_1, \dots, \mathbf{k}_n)$ means that the n -uple of wave-vectors should be included only once, and $|\Psi_n\rangle$ denotes a *normalised* state with exactly n Cooper pairs (hence $N = 2n$ fermions), appearing with (real) amplitude A_n but with an overall phase $e^{in\varphi}$. Normalisation of all states implies that the coefficients A_n^2 can be thought as a probability distribution of the various n in the BCS state:

$$\langle \Psi_{\text{BCS}} | \Psi_{\text{BCS}} \rangle = 1 \quad \implies \quad \sum_{n=0}^{\infty} A_n^2 = 1. \quad (7.16)$$

At this stage, φ could be used as a technical tool to single-out the various fixed particle number states. Indeed, by integrating of φ we get:

$$A_n |\Psi_n\rangle = \int_0^{2\pi} \frac{d\varphi}{2\pi} e^{-in\varphi} |\Psi_{\text{BCS}}(\varphi)\rangle. \quad (7.17)$$

The coefficients A_n^2 could also be explicitly calculated by an integral over φ , but this will not be relevant to our discussion. ³ What is relevant, is that in a macroscopic superconductor, A_n^2 is peaked

³One can verify that:

$$A_n^2 = \int_0^{2\pi} \frac{d\varphi}{2\pi} e^{-in\varphi} \prod_{\mathbf{k}} (u_{\mathbf{k}}^2 + e^{i\varphi} v_{\mathbf{k}}^2). \quad (7.18)$$

This gives, correctly, $A_0^2 = \prod_{\mathbf{k}} u_{\mathbf{k}}^2$, $A_1^2 = \sum_{\mathbf{k}_1} (\prod_{\mathbf{k} \neq \mathbf{k}_1} u_{\mathbf{k}}^2) v_{\mathbf{k}_1}^2$, etc.

at an n_0 which is *extensive*. If $\hat{N} = \sum_{\mathbf{k},\sigma} \hat{c}_{\mathbf{k}\sigma}^\dagger \hat{c}_{\mathbf{k}\sigma}$ denotes the total number of fermions operator, with average $N_0 = \langle \Psi_{\text{BCS}} | \hat{N} | \Psi_{\text{BCS}} \rangle$, you can write:

$$n_0 = \frac{1}{2} N_0 = \frac{1}{2} \langle \Psi_{\text{BCS}} | \hat{N} | \Psi_{\text{BCS}} \rangle = \sum_{\mathbf{k}} v_{\mathbf{k}}^2 = \text{Vol} \int \frac{d\mathbf{k}}{(2\pi)^3} v_{\mathbf{k}}^2. \quad (7.19)$$

Interestingly, the *width* of the distribution A_n^2 scales with $\sqrt{\text{Vol}}$, as you can show that:

$$\begin{aligned} (\Delta n)^2 &= \frac{1}{4} (\Delta N)^2 = \frac{1}{4} (\langle \Psi_{\text{BCS}} | \hat{N}^2 | \Psi_{\text{BCS}} \rangle - \langle \Psi_{\text{BCS}} | \hat{N} | \Psi_{\text{BCS}} \rangle^2) \\ &= \sum_{\mathbf{k}} u_{\mathbf{k}}^2 v_{\mathbf{k}}^2 = \text{Vol} \int \frac{d\mathbf{k}}{(2\pi)^3} u_{\mathbf{k}}^2 v_{\mathbf{k}}^2. \end{aligned} \quad (7.20)$$

Hence, for a macroscopic superconductor, it makes no difference if you calculate physical properties by using $|\Psi_{\text{BCS}}\rangle$, a state that is simple to work with, or rather the much more complicated state with fixed number of Cooper pairs $|\Psi_n\rangle$ with $n \sim n_0$. This is very similar to the grand-canonical description, as opposed to a canonical one, which become equivalent in the thermodynamic limit.

A Josephson junction. Two superconductors (S) separated by a thin insulating (I) barrier, in the SIS geometry sketched in Fig. 7.1, exhibit a remarkable phenomenon, discovered by Josephson in 1962 [36]. I will give here a phenomenological description. A more microscopic characterization is

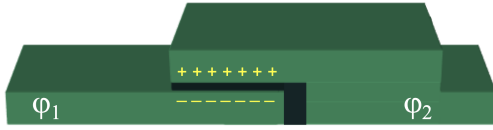


Figure 7.1: Sketch of a Josephson junction, with the two superconductors characterized by a phase $\varphi_{1,2}$, separated by a thin insulating layer. The charges highlight that a small enough junction should show Coulomb effects due to a finite capacitance.

given in App. C.2. Each superconductor is characterized by a macroscopic phase ⁴ — φ_1 and φ_2 for the two superconductors —, and the phase difference

$$\phi = \varphi_1 - \varphi_2, \quad (7.21)$$

is associated to a component of the tunnelling current going through the junction, indeed an *equilibrium current* ⁵ which is present also in the absence of any voltage drop between the two superconductors, and given by:

$$I = I_J \sin \phi. \quad (7.22)$$

This is known as **dc Josephson effect**.

A second ingredient discovered by Josephson is that, in the presence of a voltage drop V between the two superconductors, the phase difference ϕ will increase in time according to the relationship:

$$\dot{\phi} = \frac{2e}{\hbar} V, \quad (7.23)$$

which leads to an *oscillating* Josephson current:

$$I(t) = I_J \sin \left(\phi_0 + \frac{2eV}{\hbar} t \right). \quad (7.24)$$

⁴Technically, there is a complex order parameter related the anomalous average of two fermionic fields

$$\psi(\mathbf{x}) = \langle \hat{\Psi}_\downarrow(\mathbf{x}) \hat{\Psi}_\uparrow(\mathbf{x}) \rangle = |\psi(\mathbf{x})| e^{i\varphi(\mathbf{x})},$$

where $|\psi|$ is related to the superconducting gap, and φ the macroscopic phase.

⁵There is a piece of the equilibrium free-energy of the system which is $-E_J \cos \phi$, from which the dc-Josephson relations follows.

This is known as **ac Josephson effect**. The average dissipated power is zero.

Summarising, the Josephson junction (JJ) is a *non-linear element*: this will be important, when quantum effects are accounted for, in reducing its spectrum to a two-level system. Indeed, if the JJ is small, *quantum effects* start to be important. Across the junction, there would be a surface charge Q and small capacitance C , with an associated energy $Q^2/2C = CV^2/2$. Due to the presence of a voltage $V = Q/C$, using the Josephson relation $\dot{\phi} = (2e/\hbar)V$, you might anticipate a Lagrangian of the form:

$$\mathcal{L} = \frac{1}{2}C\left(\frac{\hbar\dot{\phi}}{2e}\right)^2 + E_J \cos \phi, \quad (7.25)$$

where the second piece is associated to the Josephson energy term $-E_J \cos \phi$. The canonical variable is $\Phi = \hbar\phi/(2e)$ and the associated canonical momentum would be

$$\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = C \frac{\hbar\dot{\phi}}{2e} = CV = Q.$$

Equivalently, we might write a Hamiltonian

$$H = \frac{Q^2}{2C} - E_J \cos \phi, \quad (7.26)$$

which looks like the Hamiltonian of a physical pendulum if one interprets ϕ as the angular coordinate and Q as the associated momentum.

The crucial step forward, for our purposes, is the suggestion that Q and $\hbar\phi/(2e)$ can be thought as canonically conjugate variables. Setting $\hat{Q} = (2e)\hat{n}$, with \hat{n} the integer charge imbalance, we would write

$$[\hat{\phi}, \hat{n}] = i, \quad (7.27)$$

and quantum effects might be important in the JJ Hamiltonian ⁶

$$\hat{H} = \frac{\hat{Q}^2}{2C} - E_J \cos \hat{\phi} = 4E_C \hat{n}^2 - E_J \cos \hat{\phi}, \quad (7.28)$$

where $E_C = e^2/(2C)$ is the Coulomb charging energy parameter, related to the junction capacitance. This suggestion was actively pursued in the 1980's: people were looking for signatures of quantum effects in an essentially macroscopic object, the JJ.

1

The JJ as a quantum pendulum. Summarising, the local degree of freedom of a JJ, after disregarding many of the ingredients in the mesoscopic solid-state system that composes it, can be regarded as a phase variable $\hat{\phi}$, and its canonically conjugate “momentum”, the integer charge imbalance \hat{n} , with $[\hat{\phi}, \hat{n}] = i\hbar$ and a quantum Hamiltonian which is essentially a quantum pendulum:

$$\hat{H}_{JJ} = 4E_C \hat{n}^2 - E_J \cos \hat{\phi}, \quad (7.29)$$

the (an-harmonic) spectrum of which depends on the ratio E_J/E_C . The different regimes in which JJ QBits have been considered in the various experiments depend, essentially, on the ratio E_J/E_C .

⁶The fact that there is a macroscopic phase ϕ is already a quantum phenomenon, clearly. But here we will be talking of quantum effects governing the dynamics of this macroscopic phase, a *secondary* quantum effect.

The role of magnetic fields and gauge invariance. So far, we did not consider the effect of external magnetic fields. This is best done through the Ginzburg-Landau theory [37, 38], which is briefly summarised in App. C. Here we simply stress the fact that most of the physics is dictated by **minimal coupling and gauge invariance**.

The microscopic form of the kinetic energy in presence of \mathbf{A} is given by:

$$\hat{H}_{\text{kin}} = \frac{1}{2m} \sum_{\sigma} \int d\mathbf{x} \left((-i\hbar\nabla + \frac{e}{c}\mathbf{A})\hat{\Psi}_{\sigma}(\mathbf{x}) \right)^{\dagger} \cdot \left((-i\hbar\nabla + \frac{e}{c}\mathbf{A})\hat{\Psi}_{\sigma}(\mathbf{x}) \right), \quad (7.30)$$

where $\hat{\Psi}_{\sigma}(\mathbf{x})$ is the field operator for the electrons. A gauge transformation $\mathbf{A} \rightarrow \mathbf{A} + \nabla\Lambda$ leaves the kinetic energy invariant provided we also transform the field operator as

$$\hat{\Psi}_{\sigma}(\mathbf{x}) \rightarrow \hat{\Psi}_{\sigma}(\mathbf{x}) e^{-i\frac{e}{\hbar c}\Lambda(\mathbf{x})}.$$

This implies that the local pair potential $\Delta(\mathbf{x})$, and therefore the local order parameter $\psi(\mathbf{x})$, containing *two* fermionic annihilation operators, should transform, after the gauge transformation, as:

$$\psi(\mathbf{x}) = \langle \hat{\Psi}_{\downarrow}(\mathbf{x})\hat{\Psi}_{\uparrow}(\mathbf{x}) \rangle = |\psi(\mathbf{x})|e^{i\varphi(\mathbf{x})} \xrightarrow{\mathbf{A} \rightarrow \mathbf{A} + \nabla\Lambda} \psi(\mathbf{x}) e^{-i\frac{2e}{\hbar c}\Lambda(\mathbf{x})} = \psi(\mathbf{x}) e^{-i\frac{2\pi}{\Phi_0}\Lambda(\mathbf{x})}, \quad (7.31)$$

where $\Phi_0 = hc/(2e)$ is the flux quantum. Hence, by gauge invariance, if $\mathbf{A} \rightarrow \mathbf{A} + \nabla\Lambda$, then you should change

$$\varphi(\mathbf{x}) \rightarrow \varphi(\mathbf{x}) - \frac{2\pi}{\Phi_0}\Lambda(\mathbf{x}). \quad (7.32)$$

These considerations suggest that the phase difference appearing in the JJ energy and current should be modified and made **gauge invariant** as follows:

$$\phi = \varphi_1 - \varphi_2 + \frac{2\pi}{\Phi_0} \int_{\text{link } 2 \rightarrow 1} \mathbf{A} \cdot d\mathbf{l}. \quad (7.33)$$

These gauge invariance considerations have multifold consequences. One of these, is the flux quantization inside a superconducting ring, briefly explained in App. C. Another consequence is a relationship between φ and \mathbf{A} inside a bulk superconductor. Indeed, starting from the GL kinetic energy density, you derive by the simple substitution $\psi(\mathbf{x}) = |\psi(\mathbf{x})|e^{i\varphi(\mathbf{x})}$:

$$\begin{aligned} \frac{1}{2m_*} \left| \left(-i\hbar\nabla + \frac{2e}{c}\mathbf{A} \right) \psi \right|^2 &= \frac{\hbar^2}{2m_*} (\nabla|\psi|)^2 + \frac{1}{2m_*} |\psi|^2 \left(\hbar\nabla\varphi + \frac{2e}{c}\mathbf{A} \right)^2 \\ &= \frac{\hbar^2}{2m_*} (\nabla|\psi|)^2 + \frac{1}{2} n_s m_* \mathbf{v}_s^2, \end{aligned} \quad (7.34)$$

where $n_s = |\psi|^2$ and $m_* \mathbf{v}_s = \nabla\varphi + \frac{2e}{c}\mathbf{A}$ are the superfluid density and velocity, respectively. The first term is the kinetic cost for changing the modulus of the order parameter, while the second piece is associated to the superfluid kinetic energy. This suggests that, inside a bulk superconductor, it is **energetically favourable** to have $\mathbf{v}_s \equiv 0$, hence:

$$\nabla\varphi = -\frac{2e}{\hbar c}\mathbf{A} = -\frac{\Phi_0}{2\pi}\mathbf{A} \quad (\mathbf{v}_s \equiv 0). \quad (7.35)$$

The final consequence of gauge invariance I need to mention is a remarkable phase-interference phenomenon, which leads to the physics of the dc-SQUID, an acronym for Superconducting QUantum Interference Device, which we now briefly consider.

The dc-SQUID. Consider a ring geometry with **two Josephson junctions**, A and B as sketched in Fig. 7.2, symmetrically placed and separating a first arm of the ring, with superconductor “1”, from the second arm, with superconductor “2”. The two superconductors are connected to leads and a

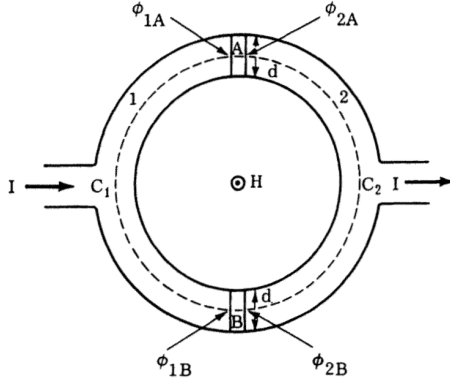


Figure 7.2: This is Fig. 7-7 in de Gennes' book [37]. In the text, we indicate ϕ_{1A} as φ_{1A} , and so on.

current I is driven through the circuit. In the center of the ring, there is an external magnetic field H , which can be changed. We want to show that the current flowing $I(H)$ is periodically modulated by the magnetic field H , in a way that closely resemble the interference effects in a double slit, or in the two arms of a Mach-Zehnder interferometer.

The first thing to notice is that the superconducting current flowing through the two JJ in parallel is simply the sum of the two currents along the branches:

$$I = I_J^A \sin \phi_A + I_J^B \sin \phi_B, \quad (7.36)$$

where ϕ_A and ϕ_B are the gauge-invariant phase differences at the two junctions.

The second thing to notice is that the phase in the superconductors must be single-valued. Hence, upon returning to the same point, the result must differ by a multiple of 2π . Therefore:

$$(\varphi_{1B} - \varphi_{1A}) + (\varphi_{2B} - \varphi_{1B}) + (\varphi_{2A} - \varphi_{2B}) + (\varphi_{1A} - \varphi_{2A}) = 2\pi n. \quad (7.37)$$

Third: the magnetic flux Φ trapped inside the ring can be calculated by integrating the vector potential along a close (anticlockwise) contour Γ made by the two pieces C_1 and C_2 inside the two superconductors, plus the two small links across the junctions.

$$\int_{C_1} \mathbf{A} \cdot d\mathbf{l} + \int_{\text{link B } 1 \rightarrow 2} \mathbf{A} \cdot d\mathbf{l} + \int_{C_2} \mathbf{A} \cdot d\mathbf{l} + \int_{\text{link A } 2 \rightarrow 1} \mathbf{A} \cdot d\mathbf{l} = \Phi. \quad (7.38)$$

If you imagine that the superconductors are larger than the penetration length, taking C_1 and C_2 in the bulk, the phase $\varphi(\mathbf{x})$ along C_1 and C_2 is related to \mathbf{A} :

$$\nabla \varphi = -\frac{2\pi}{\Phi_0} \mathbf{A}.$$

Integrating along C_1 and C_2 we get:

$$\varphi_{1B} - \varphi_{1A} = -\frac{2\pi}{\Phi_0} \int_{C_1} \mathbf{A} \cdot d\mathbf{l} \quad \text{and} \quad \varphi_{2A} - \varphi_{2B} = -\frac{2\pi}{\Phi_0} \int_{C_2} \mathbf{A} \cdot d\mathbf{l}. \quad (7.39)$$

Now, sum the two sides of Eq. (7.37) and Eq. (7.38) (multiplied by $2\pi/\Phi_0$):

$$(\varphi_{2B} - \varphi_{1B}) + \frac{2\pi}{\Phi_0} \int_{\text{link B } 1 \rightarrow 2} \mathbf{A} \cdot d\mathbf{l} + (\varphi_{1A} - \varphi_{2A}) + \frac{2\pi}{\Phi_0} \int_{\text{link A } 2 \rightarrow 1} \mathbf{A} \cdot d\mathbf{l} = 2\pi n + 2\pi \frac{\Phi}{\Phi_0},$$

where we made use of the cancellations provided by Eq. (7.39). The gauge invariant phase difference at junction A is

$$\phi_A = \varphi_{1A} - \varphi_{2A} + \frac{2\pi}{\Phi_0} \int_{\text{link A } 2 \rightarrow 1} \mathbf{A} \cdot d\mathbf{l}.$$

Similarly, the gauge invariant phase difference at junction B is

$$\phi_B = \varphi_{1B} - \varphi_{2B} + \frac{2\pi}{\Phi_0} \int_{\text{link B } 2 \rightarrow 1} \mathbf{A} \cdot d\mathbf{l} .$$

i **The relationship between the phases at the two junctions.** Therefore, we conclude that:

$$\phi_A - \phi_B = 2\pi n + 2\pi \frac{\Phi}{\Phi_0} . \quad (7.40)$$

Let us now parameterise ϕ_A and ϕ_B as follows:

$$\phi_A = \phi_+ + \pi \left(n + \frac{\Phi}{\Phi_0} \right) \quad \text{and} \quad \phi_B = \phi_+ - \left(n + \frac{\Phi}{\Phi_0} \right) , \quad (7.41)$$

with $\phi_+ = (\phi_A + \phi_B)/2$ the average phase difference. The total current is therefore predicted to be:

$$I = I_J^A \sin \left(\phi_+ + \pi \left(n + \frac{\Phi}{\Phi_0} \right) \right) + I_J^B \sin \left(\phi_+ - \pi \left(n + \frac{\Phi}{\Phi_0} \right) \right) . \quad (7.42)$$

For two **identical** junctions, $I_J^A = I_J^B = I_J$, simple trigonometry allows us to conclude that:

$$I = 2I_J \cos \left(\pi \left(n + \frac{\Phi}{\Phi_0} \right) \right) \sin \phi_+ . \quad (7.43)$$

Hence, effectively, the two parallel junctions act as a **single junction** with average phase difference ϕ_+ , and a critical current — hence Josephson energy constant — that can be **tuned by the magnetic flux** Φ :

$$I_J^{\text{eff}}(\Phi) = 2I_J \left| \cos \left(\pi \frac{\Phi}{\Phi_0} \right) \right| \quad \Longrightarrow \quad E_J^{\text{eff}}(\Phi) = \frac{\hbar}{2e} I_J^{\text{eff}}(\Phi) = 2E_J \left| \cos \left(\pi \frac{\Phi}{\Phi_0} \right) \right| . \quad (7.44)$$

Notice the constructive interference for all fluxes that are multiples of Φ_0 , while the interference is **destructive** for $\Phi = \Phi_0/2$. The case of two asymmetric JJ is briefly mentioned in App. C.4.

7.3. The superconducting Qbits platforms

7.3.1. Charge Qbits: The Cooper pair box

The setting of a Charge QBit — also known as Cooper pair box — is sketched in Fig. 7.3. Its effective Hamiltonian can be written as:

$$\hat{H}_{\text{CPB}} = 4E_C (\hat{n} - n_g)^2 - E_J \cos \hat{\phi} , \quad (7.45)$$

where $n_g = C_g U / (2e)$ is the average charge, divided by $2e$, that the gate at potential U induces in the small superconducting island ⁷, and $E_C = e^2 / (C_j + C_g)$ is the charging energy associated to a total capacitance $C = C_j + C_g$, where C_j is the JJ capacitance. In the charge QBits regime one assumes that

$$k_B T \ll E_J \ll E_C .$$

Let us treat E_J as a small perturbation. The charged states $|n\rangle$ can be seen, in terms of ϕ , as the

⁷One should regard this parameter n_g as a continuous variable, since typically one operates in a grand-canonical ensemble where the number of charge carriers is not fixed, but rather the chemical potential is controlled. On the contrary, the operator \hat{n} has an integer spectrum.

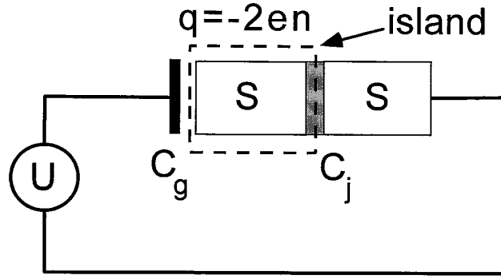


Figure 7.3: Sketch of a Cooper pair box. The Josephson junction is formed by a thin insulating layer between a bulk superconductor and a small superconducting island, connected (an open circuit) via a capacitor C_g to a gate voltage U which determines the average number $n_g = C_g U / (2e)$ of Cooper pairs in the island. C_j denotes the capacitance of the JJ, such that $E_C = e^2 / (2C)$, with $C = C_j + C_g$ the total capacitance. Figure taken from Ref. [39].

plane-wave eigenstates

$$\langle \phi | n \rangle = \frac{1}{\sqrt{2\pi}} e^{in\phi}.$$

The unperturbed spectrum of $4E_C(\hat{n} - n_g)^2$, when plotted versus n_g for different eigenvalues of \hat{n} forms a sequence of parabola centered at $n_g = 0, \pm 1, \pm 2, \dots$. The commutation relation between $\hat{\phi}$ and \hat{n} implies that $\cos \phi = (e^{i\phi} + e^{-i\phi})/2$ couples $|n\rangle$ to $|n+1\rangle$ and viceversa. Hence we can write:

$$\hat{H}_{\text{CPB}} = 4E_C \sum_{n=-\infty}^{+\infty} (n - n_g)^2 |n\rangle \langle n| - \frac{E_J}{2} \sum_{n=-\infty}^{+\infty} (|n+1\rangle \langle n| + |n\rangle \langle n+1|). \quad (7.46)$$

Let us suppose that $n_g \in [0, 1]$. By restricting the Hamiltonian to two lowest levels, corresponding to $n = 0$ and $n = 1$, we are eventually lead to a two-level system

$$\begin{aligned} \hat{H} &= 4E_C (n_g^2 |0\rangle \langle 0| + (1 - n_g)^2 |1\rangle \langle 1|) - \frac{E_J}{2} (|0\rangle \langle 1| + |1\rangle \langle 0|) \\ &= 2E_C (n_g^2 + (1 - n_g)^2) \mathbf{1} + 4E_C (n_g - \frac{1}{2}) \hat{\sigma}^z - \frac{E_J}{2} \hat{\sigma}^x, \end{aligned} \quad (7.47)$$

where the final form expresses the two-level system in terms of Pauli matrices. ⁸

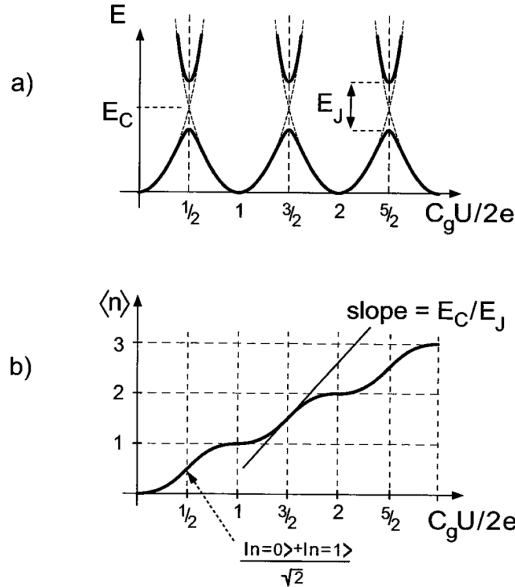


Figure 7.4: (a) The spectrum of the Cooper pair box lowest eigenvalues as a function of the gate-voltage-induced n_g , when $E_J \ll E_C$ so that the Josephson term is a small perturbation of the parabolic spectrum of $(\hat{n} - n_g)^2$. At $n_g = \pm 1/2$ the parabola with $n = 0$ crosses that for $n = \pm 1$, and degenerate first-order perturbation theory can be used to estimate a gap opening linear in E_J/E_C . (b) The theory predicted average number of Cooper pairs in the island. Figure taken from Ref. [39].

The voltage bias n_g plays a crucial role: by controlling it, you can easily implement various single-QBit unitaries. ⁹

⁸Use that $|0\rangle \langle 0| = (\mathbf{1} + \hat{\sigma}^z)/2$ and $|1\rangle \langle 1| = (\mathbf{1} - \hat{\sigma}^z)/2$, together with: $|0\rangle \langle 1| + |1\rangle \langle 0| \rightarrow \hat{\sigma}^x$.

⁹By sweeping n_g across the value $n_1 = 1/2$, with a certain speed, you effectively create a Landau-Zener dynamics in the two lowest eigenvalues.

To determine if a Cooper-pair box shows quantum effects or not, consider the average number of Cooper pairs $\langle \hat{n} \rangle$ as a function of n_g . For a classical model, one would predict a staircase at integer values of $\langle \hat{n} \rangle = 0, 1, 2, \dots$ with a rounding due to thermal effects.

Quantum mechanically, even at $T \rightarrow 0$, the ground state, for instance, for $n_g = \frac{1}{2}$ corresponds to a spin problem with $\hat{H} = -\frac{E_J}{2} \hat{\sigma}^x$. Hence, the ground state is $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, so that $\langle \hat{n} \rangle = \frac{1}{2}$.

More generally, one can show that at $T \rightarrow 0$, the form of $\langle \hat{n} \rangle$ is given in the Fig. 7.4(b), with a characteristic slope of the curve at half-integer n_g given by E_C/E_J . This is indeed experimentally shown in Ref. [39]. Signatures of quantum effects in a Cooper-pair box have been also revealed by Nakamura, in Ref. [40].

The problem with the Cooper-pair box Qbit is the sensitivity to *charge noise* which induces **fluctuations** in n_g , leading to short coherence times for the Qbit. To remedy to this, one needs to increase E_J/E_C .

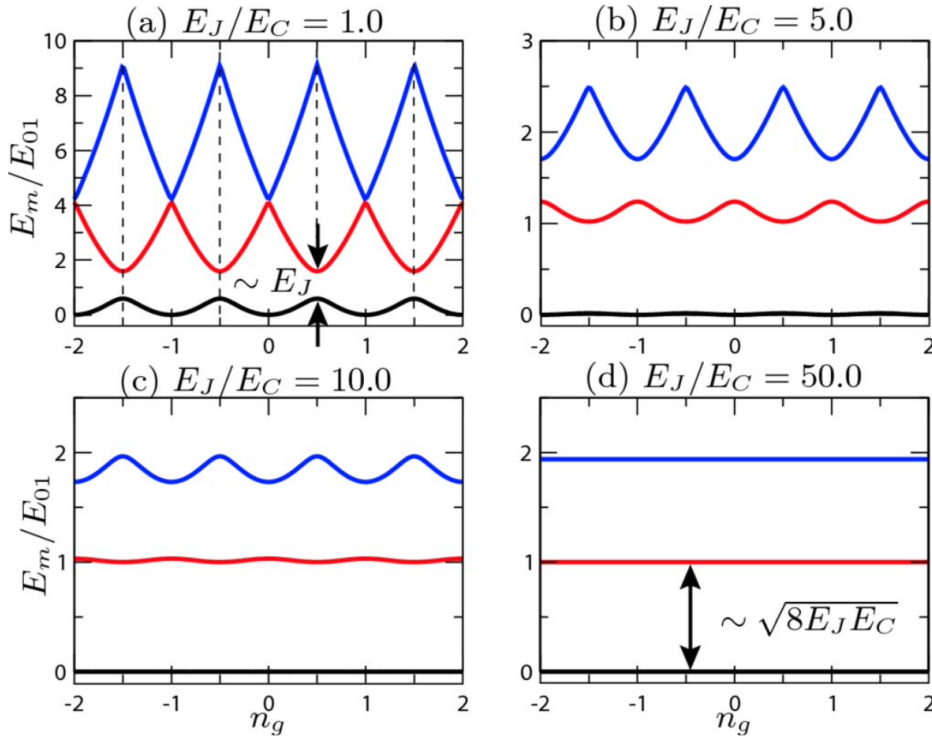


Figure 7.5.: The spectrum of JJ (the lowest 3 eigenvalues), as a function of the gate-voltage-induced n_g , for various regimes of E_J/E_C . This is Fig. 2 of Ref. [41].

Increasing E_J/E_C . Fig. 7.5 shows the spectrum of the JJ Hamiltonian — more precisely, the lowest three eigenvalues, as a function of the gate-voltage-induced n_g . One observes that the eigenvalues rapidly become insensitive to the value of n_g — indeed, exponentially so [41] — thus becoming relatively insensitive to charge noise.

7.3.2. The transmon

Let us consider \hat{H}_{JJ} , neglecting any bias term n_g :

$$\hat{H}_{JJ} = 4E_C \hat{n}^2 - E_J \cos \hat{\phi}, \quad (7.48)$$

in a regime in which

$$E_J \gg E_C \gg k_B T ,$$

for instance, with $E_J/E_C \sim 50$. In this regime the resulting QBit — after a suitable two-level system truncation — is usually called a **transmon**. Fig. 7.6 shows the spectrum of \hat{H}_{JJ} (on the right)

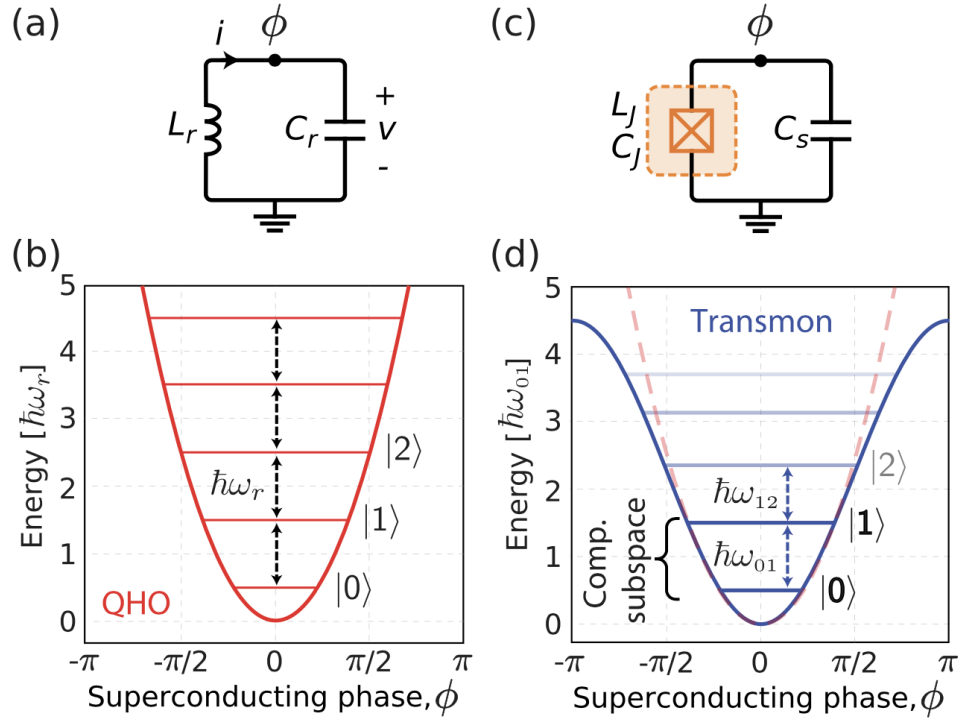


Figure 7.6.: (b) The spectrum of a quantum LC oscillator (a), compared to (d) the spectrum of a JJ (c) in the regime $E_C \ll E_J$. Figure taken from Ref. [17].

compared to the harmonic spectrum of a pure LC resonator (on the left). Notice how the $\cos \phi$ potential induces an **anharmonicity** in the spectrum, with the lowest spectral splitting $\hbar\omega_{01} = E_1 - E_0$ different from the next splitting $\hbar\omega_{12} = E_2 - E_1$. Figure 7.7 shows the same information as Fig. 7.6(d),

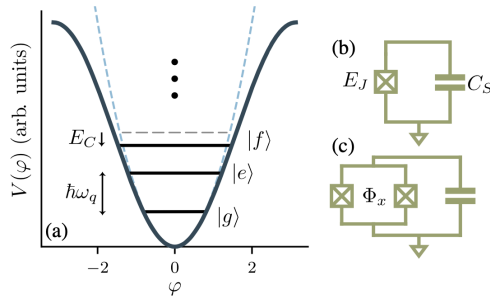


Figure 7.7: (a) The spectrum of a transmon QBit, as in Fig. 7.6(d), showing the anharmonicity E_C . (b) A capacitively shunted transmon. (c) A flux-tunable symmetric transmon. Figure taken from Ref. [34].

highlighting two facts:

- 1) While the computational subspace splitting is

$$\hbar\omega_q = \hbar\omega_{01} = \sqrt{8E_C E_J} - E_C ,$$

the anharmonicity is controlled by E_C :

$$\alpha = \hbar\omega_{12} - \hbar\omega_{01} = -E_C .$$

- 2) Panels (b) and (c) of Fig. 7.7 highlight the fact that the transmon can be either made by a single JJ (b) or rather by a symmetric or asymmetric SQUID (c) which has, therefore, a **flux-tunable** E_J .

Mapping to a Duffing oscillator

In order to map the JJ Hamiltonian into a non-linear oscillator problem, we start from the standard Taylor expansion of the $\cos \phi$, truncated to 4th order:

$$\cos \phi = 1 - \frac{1}{2}\phi^2 + \frac{1}{4!}\phi^4 + \dots$$

Let us now rewrite the conjugate variables $[\hat{\phi}, \hat{n}] = i$ in terms of oscillators in the standard way:

$$\hat{\phi} = \phi_{z\text{p}} \frac{1}{\sqrt{2}}(\hat{a} + \hat{a}^\dagger) \quad \text{and} \quad \hat{n} = \frac{1}{\phi_{z\text{p}}} \frac{1}{\sqrt{2}i}(\hat{a} - \hat{a}^\dagger),$$

where $[\hat{a}, \hat{a}^\dagger] = 1$, and choose the dimensionless quantity $\phi_{z\text{p}}$ in such a way that:

$$4E_C \frac{1}{\phi_{z\text{p}}^2} = \frac{1}{2}E_J \phi_{z\text{p}}^2 \equiv \frac{1}{2}\hbar\omega_0 \quad \Longrightarrow \quad \phi_{z\text{p}}^2 = \sqrt{\frac{8E_C}{E_J}} \quad \text{and} \quad \hbar\omega_0 = \sqrt{8E_J E_C}. \quad (7.49)$$

Neglecting constant terms, the Hamiltonian reduces, up to fourth order, to:

$$\begin{aligned} \hat{H}_{JJ} &\xrightarrow{\text{4th-order}} \hbar\omega_0 \hat{a}^\dagger \hat{a} - \frac{1}{24}E_J \frac{1}{4}\phi_{z\text{p}}^4 (\hat{a} + \hat{a}^\dagger)^4 \\ &= \hbar\omega_0 \hat{a}^\dagger \hat{a} - \frac{1}{12}E_C (\hat{a} + \hat{a}^\dagger)^4. \end{aligned} \quad (7.50)$$

Next, we consider only excitation-conserving terms, i.e., the 6 terms in the expansion of $(\hat{a} + \hat{a}^\dagger)^4$ which contain the same number of \hat{a} and \hat{a}^\dagger :

$$\begin{aligned} (\hat{a} + \hat{a}^\dagger)^4 &\xrightarrow{\text{exc-conserving}} \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} + \hat{a}^\dagger \hat{a} \hat{a}^\dagger \hat{a} + \hat{a} \hat{a}^\dagger \hat{a}^\dagger \hat{a} + \hat{a}^\dagger \hat{a} \hat{a} \hat{a}^\dagger + \hat{a} \hat{a} \hat{a}^\dagger \hat{a}^\dagger + \hat{a} \hat{a} \hat{a}^\dagger \hat{a}^\dagger \\ &= 3 + 12 \hat{a}^\dagger \hat{a} + 6 \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a}, \end{aligned} \quad (7.51)$$

where we made repeated use of the commutator, hence of $\hat{a} \hat{a}^\dagger = 1 + \hat{a}^\dagger \hat{a}$, to bring the \hat{a}^\dagger to the *left* of the \hat{a} (normal ordering). Neglecting constants we finally arrive at the so-called **Duffing oscillator**.

❶

Duffing oscillator.

$$\hat{H}_{JJ} \xrightarrow[\text{exc-conserving}]{\text{4th-order}} \hbar\omega_q \hat{a}^\dagger \hat{a} - \frac{1}{2}E_C \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a}, \quad (7.52)$$

where

$$\hbar\omega_q = \hbar\omega_0 - E_C = \sqrt{8E_J E_C} - E_C. \quad (7.53)$$

Using $\hat{a}|n\rangle = \sqrt{n}|n-1\rangle$ and $\hat{a}^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle$ it is simple to show that $\hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a}|n=2\rangle = 2|n=2\rangle$, hence the level splittings predicted are:

$$\hbar\omega_{01} = \hbar\omega_q = \sqrt{8E_J E_C} - E_C \quad \text{and} \quad \hbar\omega_{12} = \hbar\omega_{01} - E_C. \quad (7.54)$$

Mapping and reduction to a two-level system

The reduction to a two-level system (QBit) is totally obvious:

$$\hat{H}_{JJ} \xrightarrow[\text{exc-conserving}]{\text{4th-order}} \hbar\omega_q \hat{a}^\dagger \hat{a} - \frac{1}{2}E_C \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} \xrightarrow{\text{two-level system}} -\frac{1}{2}\hbar\omega_q \hat{\sigma}^z, \quad (7.55)$$

with the usual mapping $\hat{a}^\dagger \hat{a} = \frac{1}{2}(1 - \hat{\sigma}^z)$, restricted to the ground state $|0\rangle = |\uparrow\rangle$ and to the first excited state $|1\rangle = |\downarrow\rangle$, again, neglecting constants. This prescription allows mapping other off-diagonal terms as well:

$$-i(\hat{a} - \hat{a}^\dagger) \xrightarrow{\text{two-level system}} \hat{\sigma}^y \quad \text{and} \quad (\hat{a} + \hat{a}^\dagger) \xrightarrow{\text{two-level system}} \hat{\sigma}^x. \quad (7.56)$$

We will see that this turns out very useful when considering couplings introduced to manipulate each single QBit, or when **coupling between different QBits**, for instance by an inductive or capacitive coupling, see later on. The basic reason for this is that the original variables $\hat{\phi}$ and \hat{n} of each QBit are directly expressed in terms of \hat{a} and \hat{a}^\dagger :

$$\hat{\phi} = \frac{\phi_{z\text{p}}}{\sqrt{2}}(\hat{a} + \hat{a}^\dagger) \xrightarrow{\text{two-level system}} \frac{\phi_{z\text{p}}}{\sqrt{2}} \hat{\sigma}^x, \quad (7.57)$$

and similarly:

$$\hat{n} = \frac{1}{\phi_{z\text{p}}\sqrt{2}i}(\hat{a} - \hat{a}^\dagger) \xrightarrow{\text{two-level system}} \frac{1}{\sqrt{2}\phi_{z\text{p}}} \hat{\sigma}^y. \quad (7.58)$$

7.4. Variants of JJ Qbits

We have already discussed that a QBit in the regime in which $E_J/E_C \sim 50$ — a transmon — can be made **flux-tunable** by adopting a dc-SQUID geometry. Figure 7.8 shows on the left (a and b) the geometric arrangement and spectrum of the two lowest spectral splittings for a symmetric transmon — the two JJ are identical, $E_J^B = E_J^A$ —, while the right part (c and d) show an asymmetric transmon situation, with an asymmetry parameter $\gamma = 2.5$, i.e., $E_J^B = \gamma E_J^A$, see App. C.4, and Eq. C.78, for more details. Interestingly, many other geometric arrangements of many JJ have been explored, creating

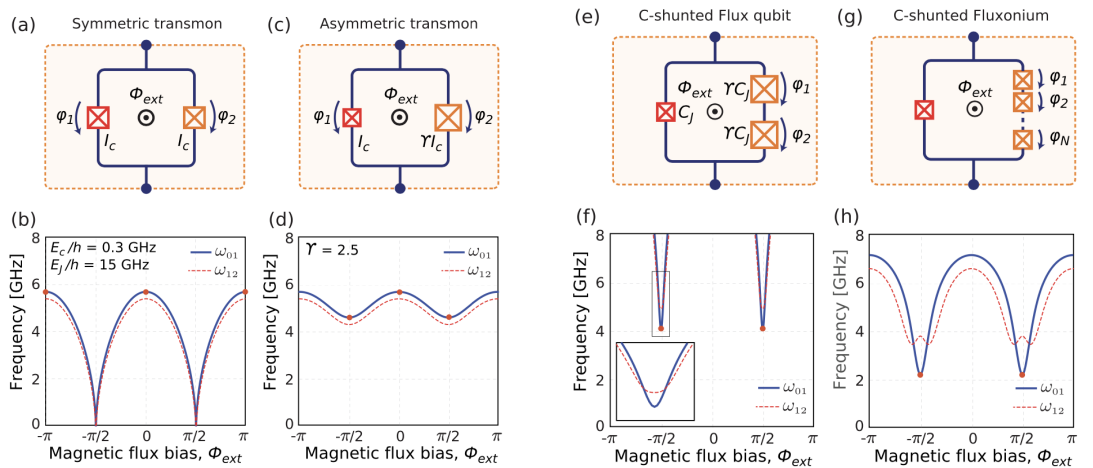


Figure 7.8.: (a-b) A symmetric flux-tunable transmon (a), and the behaviour of the two lowest spectral splittings $\hbar\omega_{01}$ and $\hbar\omega_{12}$ as a function of the magnetic flux Φ (b). (c-d) An asymmetric flux-tunable transmon, with asymmetry parameter $\gamma = 2.5$. (e-f) The Flux Qbit and (g-h) the Fluxonium, obtained by arranging more JJ of suitable parameters in a ring geometry. This is Fig. 2 of Ref. [17].

various sorts of QBits — Flux Qbit, Fluxonium, etc. —, depending on the JJ parameters, and leading to peculiar features of the spectrum as a function of the external magnetic flux. Figure 7.8(e-f,g-h) show two interesting cases which have been explored.

7.5. Manipulating and coupling superconducting QBits

A good source of material is Ref. [17], particularly Sec.II and IV. We want now to discuss how one can in principle implement single-QBit and two-QBit gates with JJ Qbits. For detailed discussions of convenient coupling schemes, see Refs. [42] and [43].

7.5.1. Manipulating single Qbits

Let us start discussing how you can manipulate the state of a single JJ transmon-like QBit. The idea is to couple the QBit to a microwave drive line, as sketched in Fig. 7.9. The Hamiltonian is

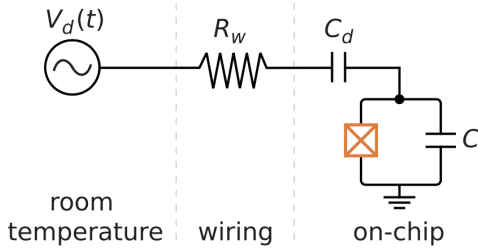


Figure 7.9: This is Fig. 12 from Ref. [17].

approximately given by: ¹⁰

$$\hat{H}(t) = \hat{H}_{JJ} + \frac{C_d}{C_{\text{tot}}} V_d(t) (2e) \hat{n}, \quad (7.59)$$

where $C_{\text{tot}} = C_d + C$ is the total capacitance. By truncating to the computational basis subspace, see Eq. (7.58), we can therefore write:

$$\hat{H}(t) = -\frac{\hbar\omega_q}{2} \hat{\sigma}^z + \Omega e V_d(t) \hat{\sigma}^y, \quad (7.60)$$

where the coupling constant Ω , see Eq. (7.49), is given by:

$$\Omega = \frac{C_d}{C_{\text{tot}}} \sqrt{2} \left(\frac{E_J}{8EC} \right)^{1/4}.$$

STILL UNDER CONSTRUCTION.

7.6. What can go wrong: the sources of dissipation and decoherence

UNDER CONSTRUCTION. A good source is Ref. [17].

7.7. Circuit QED

UNDER CONSTRUCTION. See Ref. [34].

¹⁰The quantization of the circuit requires some non-trivial steps, starting from classical Kirchoff's laws. It requires, among other things, a weak coupling assumption, $C_d \ll C$, and neglecting the resistance R_w . Starting from an LC circuit, one would write a total Lagrangian of the form:

$$\mathcal{L}(\Phi, \dot{\Phi}) = \frac{1}{2} C \dot{\Phi}^2 - \frac{1}{2L} \Phi^2 + \frac{1}{2} C_d (\dot{\Phi} - V_d(t))^2.$$

8. Density matrices

As you probably recall, in the Stern-Gerlach experiment, a beam of silver atoms ¹ passes through an appropriately designed magnet where a magnetic field gradient is present. Due to the magnetic field gradient, the center of mass of the atom is deflected, depending on the total spin of the atom, and two spots are revealed on the screen where the atoms impinge. Notice that the axis of the SG apparatus can be rotated, but the experimental outcome is that two spots with *equal abundance*, 50% and 50%, are registered no matter how you rotate the axis.

Let us ask ourselves the following question: what is the spin state $|\psi_{\text{spin}}\rangle$ of the silver atoms coming out of the oven before the Stern-Gerlach (SG) filter? Unfortunately, none of the spin states

$$|\psi_{\text{spin}}\rangle = z_+|+, \mathbf{z}\rangle + z_-|-, \mathbf{z}\rangle$$

describes the experiment. Indeed, due to normalisation $|z_+|^2 + |z_-|^2 = 1$ and with an appropriate choice of the overall phase, we can always rewrite such state as a spin pointing into an appropriate direction $\mathbf{n} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$:

$$|\psi_{\text{spin}}\rangle \equiv |+, \mathbf{n}\rangle = \cos \frac{\theta}{2} |+, \mathbf{z}\rangle + e^{i\phi} \sin \frac{\theta}{2} |-, \mathbf{z}\rangle .$$

This, in turn, would imply that by rotating the SG apparatus in the direction \mathbf{n} I should be able to measure 100% of atoms in one spot, which is, experimentally, not the case: no matter how the SG apparatus is rotated, you always get 50% of atoms in each of the two spots.

Question:

How to describe such a spin state entering the SG filter?

8.1. The density matrix for a pure state

Let us briefly recall the von Neumann postulates for a projective measurement in Quantum Mechanics. Consider a system in a state $|\psi\rangle$. You prepare a large ensemble of such pure states $|\psi\rangle$, and you measure the system observable A obtaining, according to von Neumann, one of the eigenvalues a of the associated Hermitean operator \hat{A} :

$$\hat{A} |\phi_{a,q}^A\rangle = a |\phi_{a,q}^A\rangle . \quad (8.1)$$

Here $\mathbf{q} = 1 \cdots d_a$ denotes the extra quantum numbers which distinguish the various eigenstates $|\phi_{a,q}^A\rangle$ associated to the same d_a -degenerate eigenvalue a . Let us denote by $\hat{\Pi}_a^A$ the projector on the subspace of states with eigenvalue a :

$$\hat{\Pi}_a^A = \sum_{\mathbf{q}=1}^{d_a} |\phi_{a,q}^A\rangle \langle \phi_{a,q}^A| . \quad (8.2)$$

¹Silver has an electronic configuration [Kr] $4d^{10}5s^1$, hence effectively $L = 0$ and $S = \frac{1}{2}$ due to the single unpaired electron in the $5s$ orbital.

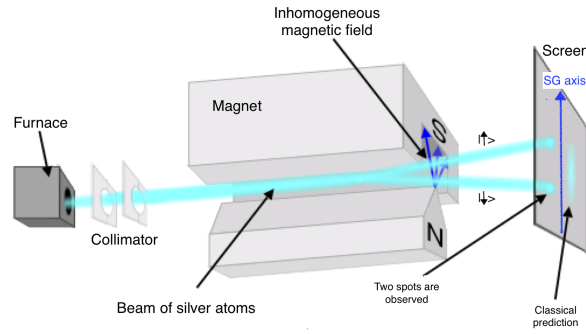


Figure 8.1.: Schematic representation of the Stern-Gerlach experiment. Two things are worth noticing: 1) The apparatus can be transformed into a *filter* by drawing a hole in the screen at the place of, say, the $|\uparrow\rangle$ spot; this guarantees that all atoms coming out from the hole are in the pure spin state $|\uparrow\rangle$. 2) The axis of the apparatus can be rotated to any direction \mathbf{n} , hence you can filter, for instance, spin states $|+\mathbf{n}\rangle$, i.e., eigenstates of $\mathbf{n} \cdot \hat{\mathbf{S}}$ with eigenvalue $+\frac{1}{2}$.

According to von Neumann, the probability P_a of measuring the eigenvalue a is:

$$\begin{aligned} P_a = \text{Prob}_A(a|\psi) &= \langle \psi | \hat{\Pi}_a^A | \psi \rangle = \langle \psi | \hat{\Pi}_a^A \hat{\Pi}_a^A | \psi \rangle = \| \hat{\Pi}_a^A | \psi \rangle \|^2 \\ &= \text{Tr} \left(\hat{\Pi}_a \hat{\Pi}_\psi \right) \end{aligned} \quad (8.3)$$

where the second form follows by a simple reshuffling of the terms ² and introduces the state projector $\hat{\Pi}_\psi = |\psi\rangle\langle\psi|$. According to the postulates, the pure state $|\psi\rangle$ collapses, after each measurement, to a new pure state $|\psi_a\rangle$:

$$|\psi\rangle \xrightarrow{a} |\psi_a\rangle = \frac{\hat{\Pi}_a^A |\psi\rangle}{\| \hat{\Pi}_a^A |\psi\rangle \|}. \quad (8.4)$$

The state is univocally assigned once we know the operator $\hat{\rho} = \hat{\Pi}_\psi$, because you can predict all probabilities of all possible measurement outcomes for any observable \hat{A} . Since the spectral representation of \hat{A} can be written as:

$$\hat{A} = \sum_a \sum_{q=1}^{d_a} a |\phi_{a,q}\rangle\langle\phi_{a,q}| = \sum_a a \hat{\Pi}_a^A \quad (8.5)$$

it immediately follows that the average of the measurements is given by:

$$\langle \hat{A} \rangle = \sum_a a \text{Prob}_A(a|\psi) = \langle \psi | \hat{A} | \psi \rangle = \text{Tr} \left(\hat{A} \hat{\Pi}_\psi \right).$$

As an example, consider that you measure $\hat{A} = \hat{S}^z$ by sending a silver atom through the SG apparatus with the SG axis aligned along z -direction. The possible outcomes are two, $a = \pm\frac{1}{2}$, and the associated projectors are:

$$\hat{\Pi}_\pm = |\pm, \mathbf{z}\rangle\langle\pm, \mathbf{z}| = \frac{1}{2}(1 \pm \hat{\sigma}^z).$$

The probabilities associated to the two outcomes, hence the abundance of atoms in the two spots, if the state entering the SG apparatus is $|\psi\rangle$, are

$$\text{Prob}_{S^z}(a = \pm\frac{1}{2}|\psi) = \text{Tr} \left(\hat{\Pi}_\pm \hat{\Pi}_\psi \right) = |\langle \pm, \mathbf{z} | \psi \rangle|^2.$$

²Indeed:

$$\text{Tr} \left(\hat{\Pi}_a \hat{\Pi}_\psi \right) = \sum_{a', q'} \langle \phi_{a', q'} | \hat{\Pi}_a \hat{\Pi}_\psi | \phi_{a', q'} \rangle = \sum_{q=1}^{d_a} \langle \phi_{a, q} | \psi \rangle \langle \psi | \phi_{a, q} \rangle = \langle \psi | \hat{\Pi}_a^A | \psi \rangle.$$

Given an arbitrary basis $|\phi_\alpha\rangle = |\alpha\rangle$ of the Hilbert space, we can write the matrix elements of $\hat{\rho}$ as:

$$\langle\alpha'|\hat{\rho}|\alpha\rangle = \langle\alpha'|\psi\rangle\langle\psi|\alpha\rangle = \psi_{\alpha'}\psi_\alpha^*.$$

In real space $|\alpha\rangle = |\mathbf{x}\rangle$, and $\rho(\mathbf{x}', \mathbf{x}) = \psi(\mathbf{x}')\psi^*(\mathbf{x})$. For a spin-1/2 state $|\psi_{\text{spin}}\rangle = z_+|+, \mathbf{z}\rangle + z_-|-, \mathbf{z}\rangle$, we have:

$$\hat{\rho} = |\psi_{\text{spin}}\rangle\langle\psi_{\text{spin}}| = |z_+|^2|+, \mathbf{z}\rangle\langle+, \mathbf{z}| + |z_-|^2|-, \mathbf{z}\rangle\langle-, \mathbf{z}| + z_+z_-^*|+, \mathbf{z}\rangle\langle-, \mathbf{z}| + z_-z_+^*|-, \mathbf{z}\rangle\langle+, \mathbf{z}|,$$

so that the matrix elements of $\hat{\rho}$ are:

$$\begin{aligned} (\hat{\rho}) &= \begin{pmatrix} |z_+|^2 & z_+z_-^* \\ z_-z_+^* & |z_-|^2 \end{pmatrix} = \begin{pmatrix} \cos^2\frac{\theta}{2} & \cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{-i\phi} \\ \cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{i\phi} & \sin^2\frac{\theta}{2} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 + \cos\theta & \sin\theta e^{-i\phi} \\ \sin\theta e^{i\phi} & 1 - \cos\theta \end{pmatrix} = \frac{1}{2}(1 + \mathbf{n} \cdot \hat{\boldsymbol{\sigma}}) \end{aligned} \quad (8.6)$$

where the last equality assumes that the spin is directed along \mathbf{n} , parameterised by θ and ϕ .³

i

Pure state density matrices. Notice the following properties of a *pure state density matrix*

$$\hat{\rho} = \hat{\Pi}_\psi:$$

(a) $\hat{\rho}$ is *Hermitean*: $\hat{\rho} = \hat{\rho}^\dagger$

(b) $\hat{\rho}$ is *non-negative*: for any state ϕ we have that $\langle\phi|\hat{\rho}|\phi\rangle = |\langle\phi|\psi\rangle|^2 \geq 0$

(c) $\text{Tr}(\hat{\rho}) = 1$

(d) $\hat{\rho}^2 = \hat{\Pi}_\psi^2 = \hat{\Pi}_\psi = \hat{\rho}$.

Property (c) is a consequence of the normalisation of ψ , while (d) follows from the properties of projectors.

The time-dependence of the pure-state $\hat{\rho} = \hat{\Pi}_\psi$ is inherited by the Schrödinger evolution of $|\psi(t)\rangle$:

$$\hat{\rho}(t) = \hat{\Pi}_{\psi(t)} = |\psi(t)\rangle\langle\psi(t)| = \hat{U}(t, 0)\hat{\rho}(0)\hat{U}^\dagger(t, 0). \quad (8.7)$$

i

von Neumann's equation. Equivalently, we can write this unitary evolution in the form a von-Neumann's equation:

$$i\hbar\frac{d}{dt}\hat{\rho}(t) = [\hat{H}(t), \hat{\rho}(t)]. \quad (8.8)$$

8.2. The density matrix for a mixed state

Suppose that the system we are describing can be found in a number of different pure states $|\psi_j\rangle$, $j = 1, \dots, N_E$, with a certain probability $p_j \geq 0$, with $\sum_j p_j = 1$. (We assume the pure states are *normalized* $\langle\psi_j|\psi_j\rangle = 1$, but we do not assume orthogonality at this stage.) This evidently reduces our knowledge/information of the state of the system, adding a further classical probabilistic aspect to specification of the state.

³Observe that all the possible choices of phases for the spinor eigenstates of $\mathbf{n} \cdot \hat{\boldsymbol{\sigma}}$ lead to the very same expression for the associated projector, since the overall phases cancel out.

What is the natural probabilistic way of calculating the probability of obtaining a upon measuring \hat{A} , or the mean value of \hat{A} ? The natural answer is to simply sum all quantum averages on pure states with their classical probability weights p_j , obtaining for instance:

$$\sum_{j=1}^{N_E} p_j \text{Prob}(a|\psi_j) = \sum_{j=1}^{N_E} p_j \text{Tr} \left(\hat{\Pi}_a \hat{\Pi}_{\psi_j} \right) = \text{Tr} \left(\hat{\Pi}_a \left(\sum_{j=1}^{N_E} p_j \hat{\Pi}_{\psi_j} \right) \right).$$

It is therefore clear that the density matrix describing such a state, called *mixed* state, as opposed to a *pure* state described by a single ψ , is:

$$\hat{\rho} = \sum_{j=1}^{N_E} p_j |\psi_j\rangle\langle\psi_j| = \sum_{j=1}^{N_E} p_j \hat{\Pi}_{\psi_j}. \quad (8.9)$$

In terms of $\hat{\rho}$, probabilities and averages are expressed as:

$$\begin{cases} \text{Prob}(a|\hat{\rho}) = \sum_{j=1}^{N_E} p_j \langle\psi_j|\hat{\Pi}_a|\psi_j\rangle = \text{Tr} \left(\hat{\Pi}_a \hat{\rho} \right) \\ \langle A \rangle = \sum_{j=1}^{N_E} p_j \text{Tr} \left(\hat{A} \hat{\Pi}_{\psi_j} \right) = \text{Tr} \left(\hat{A} \hat{\rho} \right) \end{cases}. \quad (8.10)$$



Warning: Observe that we have said nothing so far about the number of states N_E nor about the possible non-orthogonality of the different states $|\psi_j\rangle$ belonging to the ensemble $E = \{p_j, |\psi_j\rangle\}$. More about the ambiguity with which we can express density matrices in a short while.

Again, as for the pure-state case, the time-dependence of $\hat{\rho}$ is inherited by the Schrödinger evolution of the $|\psi_j(t)\rangle$

$$\hat{\rho}(t) = \sum_j p_j |\psi(t)\rangle\langle\psi(t)| = \hat{U}(t, 0) \hat{\rho}(0) \hat{U}^\dagger(t, 0), \quad (8.11)$$

hence the von Neumann's equation applies:

$$i\hbar \frac{d}{dt} \hat{\rho}(t) = [\hat{H}(t), \hat{\rho}(t)]. \quad (8.12)$$



Density matrices for mixed states. One can show that properties (a), (b), (c) given in the previous section are still obeyed by a mixed-state $\hat{\rho}$:

(a) $\hat{\rho}$ is *Hermitean*: $\hat{\rho} = \hat{\rho}^\dagger$

(b) $\hat{\rho}$ is *non-negative*: for any state ϕ we have that $\langle\phi|\hat{\rho}|\phi\rangle \geq 0$

(c) $\text{Tr}(\hat{\rho}) = 1$.

Notice that, in general, $\hat{\rho}^2 \neq \hat{\rho}$. More about this in the next section.

8.3. Spectral properties of $\hat{\rho}$ and ambiguity on the ensemble originating $\hat{\rho}$

So far we have not commented on the number N_E and on the possible non-orthogonality of the states $|\psi_j\rangle$ forming the ensemble $E = \{p_j, |\psi_j\rangle\}$ which generates $\hat{\rho}$.

Question:

Given a $\hat{\rho}$ satisfying the properties (a), (b), (c) listed above, can you reconstruct in a unique way an ensemble $E = \{p_j, |\psi_j\rangle\}$? The answer is, in general, *no*, and is also connected to a non-orthogonality issue, which we will briefly illustrate at the end of this section. More comments in Sec. 8.8 and Sec. 8.9.

To start with, suppose we are given a $\hat{\rho}$ which is positive, Hermitean and with unit trace. Then we can diagonalise it obtaining:

$$\hat{\rho} = \sum_{k=1}^{N_S} \lambda_k |\Lambda_k\rangle\langle\Lambda_k| \quad (8.13)$$

where $N_S \leq d$ (d being the dimension of the Hilbert space), $\lambda_k \in [0, 1]$ are the eigenvalues⁴ of $\hat{\rho}$ — assumed for instance to be ordered from the largest to the smallest, $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$ — and $|\Lambda_k\rangle$ the associated *orthogonal* eigenvectors. Since $\text{Tr}(\hat{\rho}) = 1$, we have:

$$\sum_{k=1}^{N_S} \lambda_k = 1. \quad (8.14)$$

This spectral decomposition is *unique* if the eigenvalues are non-degenerate. In this special eigenvector basis, it is immediate to deduce that:

$$\hat{\rho}^2 = \sum_{k=1}^{N_S} \lambda_k^2 |\Lambda_k\rangle\langle\Lambda_k| \implies \text{Tr}(\hat{\rho}^2) = \sum_{k=1}^{N_S} \lambda_k^2 \leq 1. \quad (8.15)$$

We clearly see that $\hat{\rho}^2 \neq \hat{\rho}$ unless $\lambda_k^2 = \lambda_k$. This, in turn, is possible — together with the requirement of unit trace — only if $N_S = 1$, i.e., a *single eigenvalue* $\lambda_1 = 1$ is non-vanishing, or, in other words, $\hat{\rho}$ is a *pure state*. Equivalently, we see that in general $\text{Tr}(\hat{\rho}^2) \leq 1$, the equality being obtained only if $\hat{\rho}$ is a *pure state*.

Interestingly, the unitary Schrödinger evolution is such that

$$\hat{\rho}^2(t) = \hat{U}(t, 0) \hat{\rho}^2(0) \hat{U}^\dagger(t, 0) \implies \text{Tr}(\hat{\rho}^2(t)) = \text{Tr}(\hat{\rho}^2(0)). \quad (8.16)$$

This shows that a pure-state remains pure through a Schrödinger evolution.

Let us now illustrate the non-orthogonality issue, which will be further developed in Sec. 8.8. Suppose we have a spin-1/2 system whose mixed state is represented by

$$\hat{\rho} = p_+ |+, \mathbf{z}\rangle\langle+, \mathbf{z}| + p_- |-, \mathbf{z}\rangle\langle-, \mathbf{z}| \quad (8.17)$$

where $|\pm, \mathbf{z}\rangle$ are the usual $\hat{\sigma}^z$ eigenstates, and $p_+ + p_- = 1$. Consider now the two states:

$$\begin{cases} |a\rangle = \sqrt{p_+} |+, \mathbf{z}\rangle + e^{i\phi} \sqrt{p_-} |-, \mathbf{z}\rangle \\ |b\rangle = \sqrt{p_+} |+, \mathbf{z}\rangle - e^{i\phi} \sqrt{p_-} |-, \mathbf{z}\rangle \end{cases}. \quad (8.18)$$

The density matrix associated to the ensemble of equal-weight of $|a\rangle\langle a|$ and $|b\rangle\langle b|$ is:

$$\hat{\rho}' = \frac{1}{2} |a\rangle\langle a| + \frac{1}{2} |b\rangle\langle b|.$$

⁴The eigenvalues λ_k are non-negative because of the positive nature of $\hat{\rho}$, since $\lambda_k = \langle\Lambda_k|\hat{\rho}|\Lambda_k\rangle \geq 0$.

A simple calculation shows that:

$$\begin{aligned}
\hat{\rho}' &= \frac{1}{2} \left(\sqrt{p_+} |+, \mathbf{z}\rangle + e^{i\phi} \sqrt{p_-} |-, \mathbf{z}\rangle \right) \left(\sqrt{p_+} \langle+, \mathbf{z}| + e^{-i\phi} \sqrt{p_-} \langle-, \mathbf{z}| \right) \\
&+ \frac{1}{2} \left(\sqrt{p_+} |+, \mathbf{z}\rangle - e^{i\phi} \sqrt{p_-} |-, \mathbf{z}\rangle \right) \left(\sqrt{p_+} \langle+, \mathbf{z}| - e^{-i\phi} \sqrt{p_-} \langle-, \mathbf{z}| \right) \\
&= \frac{1}{2} p_+ |+, \mathbf{z}\rangle \langle+, \mathbf{z}| + \frac{1}{2} p_- |-, \mathbf{z}\rangle \langle-, \mathbf{z}| + \frac{1}{2} \sqrt{p_+ p_-} \left(e^{-i\phi} |+, \mathbf{z}\rangle \langle-, \mathbf{z}| + e^{i\phi} |-, \mathbf{z}\rangle \langle+, \mathbf{z}| \right) \\
&+ \frac{1}{2} p_+ |+, \mathbf{z}\rangle \langle+, \mathbf{z}| + \frac{1}{2} p_- |-, \mathbf{z}\rangle \langle-, \mathbf{z}| - \frac{1}{2} \sqrt{p_+ p_-} \left(e^{-i\phi} |+, \mathbf{z}\rangle \langle-, \mathbf{z}| + e^{i\phi} |-, \mathbf{z}\rangle \langle+, \mathbf{z}| \right) \\
&= p_+ |+, \mathbf{z}\rangle \langle+, \mathbf{z}| + p_- |-, \mathbf{z}\rangle \langle-, \mathbf{z}| \equiv \hat{\rho}. \tag{8.19}
\end{aligned}$$

Evidently, the two different ensembles are associated to the *same* $\hat{\rho}$.

What is the crucial point behind this calculation? Notice that while $\langle a|a\rangle = 1$ and $\langle b|b\rangle = 1$, we have that:

$$\langle a|b\rangle = p_+ - p_- . \tag{8.20}$$

Hence, the two states are *non-orthogonal*, unless $p_+ = p_- = \frac{1}{2}$, in which case the density matrix is $\hat{\rho} = \frac{1}{2} \mathbb{1}$. The latter case is particularly interesting. You immediately notice that you can represent $\hat{\rho} = \frac{1}{2} \mathbb{1}$ in *infinitely many* ways as an equal-weight superposition of orthogonal projectors $|a\rangle\langle a|$ and $|b\rangle\langle b|$. Even more, you can show that for any \mathbf{n} :

$$\hat{\rho} = \frac{1}{2} \left(|+, \mathbf{z}\rangle \langle+, \mathbf{z}| + |-, \mathbf{z}\rangle \langle-, \mathbf{z}| \right) = \frac{1}{2} \left(|+, \mathbf{n}\rangle \langle+, \mathbf{n}| + |-, \mathbf{n}\rangle \langle-, \mathbf{n}| \right) \equiv \frac{1}{2} \mathbb{1} . \tag{8.21}$$

❶

The Stern-Gerlach state. This example shows the answer to the question we posed at the beginning: how to describe the spin state of the Silver atoms entering the SG apparatus out of the furnace. The answer is that the atoms are in the mixed state $\hat{\rho} = \frac{1}{2} \mathbb{1}$. Indeed, you can readily show that this guarantees a 50% abundance of atoms in each of the two spots, independently of the orientation of the SG apparatus.

8.4. Density matrices after measurements

I will now exemplify an “ensemble preparation” through the outcomes of a measurement. Recall that when you measure an observable A on a large ensembles of identical pure states $|\psi\rangle$ you obtain one of the eigenvalues a of the associated Hermitean operator \hat{A} , with probability

$$P_a = \text{Prob}_A(a|\psi) = \langle \psi | \hat{\Pi}_a^A | \psi \rangle = \langle \psi | \hat{\Pi}_a^A \hat{\Pi}_a^A | \psi \rangle = \| \hat{\Pi}_a^A | \psi \rangle \|^2 , \tag{8.22}$$

and the pure state $|\psi\rangle$ collapses, after each measurement, to a new pure state $|\psi_a\rangle$:

$$|\psi\rangle \xrightarrow{a} |\psi_a\rangle = \frac{\hat{\Pi}_a^A | \psi \rangle}{\| \hat{\Pi}_a^A | \psi \rangle \|} . \tag{8.23}$$

❶

Collapse after measurement. Summarising, in terms of pure state projectors, we might write the (collapsed) state after measuring the eigenvalue a as:

$$\hat{\rho}_{\text{in}} = |\psi\rangle\langle\psi| \xrightarrow{a} \hat{\rho}_a = |\psi_a\rangle\langle\psi_a| = \frac{\hat{\Pi}_a^A | \psi \rangle \langle \psi | \hat{\Pi}_a^A}{P_a} , \tag{8.24}$$

where

$$P_a = \| \hat{\Pi}_a^A | \psi \rangle \|^2 = \text{Tr} \left(\hat{\Pi}_a^A | \psi \rangle \langle \psi | \hat{\Pi}_a^A \right) = \text{Tr} \left(\hat{\Pi}_a^A \hat{\rho}_{\text{in}} \hat{\Pi}_a^A \right) . \tag{8.25}$$

Starting from a pure initial state $|\psi\rangle$, the ensemble of states obtained *after the measurement* — assuming we do not make any filtering selection of the states according to the outcome of the measurement, hence we disregard any information on the measured eigenvalue a — is given by a mixed state with an ensemble preparation $E = \{P_a, |\psi_a\rangle\}$, and is represented by a post-measurement density matrix

$$\hat{\rho}_{\text{p-m}} = \sum_a P_a |\psi_a\rangle\langle\psi_a| = \sum_a \mathcal{P}_a \frac{\hat{\Pi}_a^A \hat{\rho}_{\text{in}} \hat{\Pi}_a^A}{\mathcal{P}_a} = \sum_a \hat{\Pi}_a^A \hat{\rho}_{\text{in}} \hat{\Pi}_a^A. \quad (8.26)$$

If you recall that we can decompose any state ψ as:

$$|\psi\rangle = \sum_a \hat{\Pi}_a |\psi\rangle = \sum_a c_a |\psi_a\rangle, \quad (8.27)$$

where the amplitude coefficients $c_a = \langle\psi_a|\psi\rangle$ are such that:

$$|c_a|^2 = |\langle\psi_a|\psi\rangle|^2 = \|\hat{\Pi}_a^A |\psi\rangle\|^2 = \langle\psi|\hat{\Pi}_a^A \hat{\Pi}_a^A |\psi\rangle = \langle\psi|\hat{\Pi}_a^A |\psi\rangle \stackrel{\text{def}}{=} P_a, \quad (8.28)$$

you realise that the initial pure-state density matrix is given by:

$$\hat{\rho}_{\text{in}} = |\psi\rangle\langle\psi| = \sum_{a,a'} c_a c_{a'}^* |\psi_a\rangle\langle\psi_{a'}| = \sum_a P_a |\psi_a\rangle\langle\psi_a| + \sum_{a \neq a'} c_a c_{a'}^* |\psi_a\rangle\langle\psi_{a'}|. \quad (8.29)$$

Hence, in the process of measurement, the initial state off-diagonal elements, the so-called *coherences*, are killed, and only the diagonal elements P_a , the so-called *populations*, survive. Interference effects associated to the precise phase relationship contained in the amplitudes c_a are killed by the measurement.

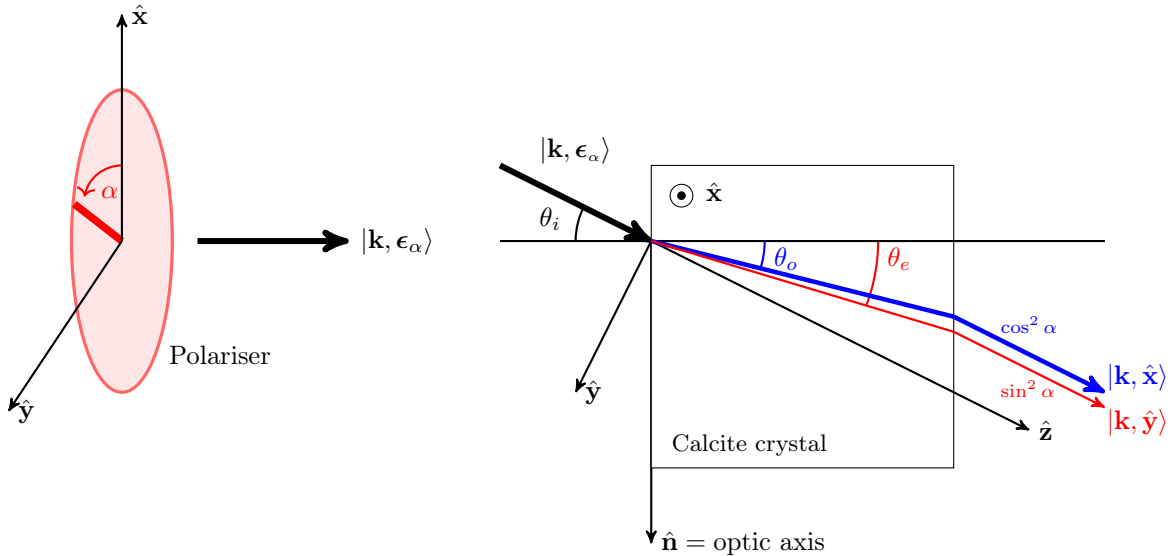


Figure 8.2.: Preparation of photons of given linear polarisation $|\epsilon_\alpha\rangle$ by using a polaroid filter with transmission axis oriented along $\epsilon_\alpha = (\hat{x} \cos \alpha + \hat{y} \sin \alpha)$. Next, the ensemble of identically prepared linearly polarised photons is sent into a calcite crystal to “measure” the polarisation along the two directions \hat{x} and \hat{y} , by exploiting the polarisation-dependent refraction of the photons. Notice the change of orientation of the axes in the right part of the figure: the optic axis of the calcite is now denoted by \hat{n} , and is assumed parallel to the crystal surface. The incoming momentum \mathbf{k} is tilted at an angle θ_i with respect to the surface normal, provoking refraction of the incoming photons.

To illustrate this idea of “ensemble preparation by measurement”, consider the experiment illustrated in Fig. 8.2. An ensemble of photons in a pure state of linear polarisation is prepared by a polaroid filter. The photons are then shined into a sufficiently thick calcite crystal which, due to uniaxial

birefringence, separates the two polarisation components, along $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, into two separate beams. The probability that a photon ends in one of the two beams is precisely that predicted by the von Neumann postulate. These are the two “collapsed pure states” obtained by the measurement. If we keep the two beams “separate”, we have prepared two separate ensembles of pure states. If we recombine, using appropriate wave-guides, the two beams into a single one, effectively ignoring the outcome of the experiment, the resulting preparation is a “post-measurement” mixed state with a $\cos^2 \alpha$ fraction of photons of polarisation $\hat{\mathbf{x}}$ and $\sin^2 \alpha$ of polarisation $\hat{\mathbf{y}}$.



Warning: Notice that while the initial state is pure, the post-measurement state has

$$\text{Tr}(\hat{\rho}_{p-m}) = \sum_a P_a^2 \leq 1 ,$$

hence it is in general a *mixed* state, unless only one eigenvalue a occurs, with probability $P_a = 1$. The process of *measurement cannot therefore be represented by a unitary evolution* operator, which would preserve the purity of the state.

8.5. Density matrices in statistical mechanics

Probably the most known example of a mixed state is the Gibbs ensemble. Suppose you have a quantum system which is enclosed into a thermostat which holds it at temperature T . This is the so-called *canonical ensemble* in quantum statistical mechanics (no exchange of particles is possible, but heat can flow in and out of the system to maintain the constant temperature). The quantum system is not described by a pure state, but rather by a density matrix whose spectral representation is

$$\hat{\rho}_{\text{Gibbs}} = \sum_n p_n |\psi_n\rangle \langle \psi_n| . \quad (8.30)$$

Here $p_n = e^{-\beta E_n} / Z$, $\beta = (k_B T)^{-1}$, is the Boltzmann weight of each state, $Z = \sum_n e^{-\beta E_n}$ is the partition function, and $\{|\psi_n\rangle\}$ denote the energy eigenstates of the system Hamiltonian \hat{H} (which form a basis of the Hilbert space of the system). It is very simple to show that you can also rewrite:

$$\hat{\rho}_{\text{Gibbs}} = \frac{1}{Z} e^{-\beta \hat{H}} . \quad (8.31)$$

As a simple application, consider a one-dimensional quantum oscillator of Hamiltonian

$$\hat{H} = \frac{\hbar\omega}{2} (\hat{p}^2 + \hat{x}^2) = \hbar\omega \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right) ,$$

\hat{p} and \hat{x} being, as usual, dimensionless momentum and coordinate, and \hat{a} and \hat{a}^\dagger the standard harmonic oscillator annihilation and creation operators. Assuming thermal equilibrium with a reservoir at temperature T , one can easily calculate many averages, as you will learn by doing the following exercise.

Exercise 8.1. Thermal averages for an harmonic oscillator.

- (1) Calculate the thermal average $\langle \hat{a}^\dagger \hat{a} \rangle = \text{Tr}(\hat{\rho}_{\text{Gibbs}} \hat{a}^\dagger \hat{a})$.
- (2) Calculate the average potential energy $\hbar\omega \hat{x}^2 / 2$ and the average kinetic energy $\hbar\omega \hat{p}^2 / 2$ and plot these quantities as a function of $k_B T / (\hbar\omega)$.
- (3) Calculate the specific heat $C_V = \partial E / \partial T$, E being the total internal energy, sum of kinetic and potential energy, and plot it as a function of $k_B T / (\hbar\omega)$.

(4) At what temperatures strong deviations from the expected classical result (state what it is that you expect, classically) are seen, due to quantum effects?

As a second example, consider a spin-1/2 in a magnetic field along the \mathbf{z} -direction. The Hamiltonian is:

$$\hat{H} = \mu_B B \hat{\sigma}^z \quad \Longrightarrow \quad \hat{\rho}_{\text{Gibbs}} = p_+ |+, \mathbf{z}\rangle \langle +, \mathbf{z}| + p_- |-, \mathbf{z}\rangle \langle -, \mathbf{z}|, \quad (8.32)$$

where

$$p_{\pm} = \frac{e^{\mp \beta \mu_B B}}{Z} \quad (8.33)$$

and $Z = e^{-\beta \mu_B B} + e^{-\beta \mu_B B} = 2 \cosh(\beta \mu_B B)$. Notice that we have a pure state only for $T = 0$ ($\beta = \infty$) since $p_- = 1$ and $p_+ = 0$. For all other temperatures T the state is mixed. The limit of $T \rightarrow \infty$ ($\beta = 0$) is represented by $\hat{\rho} = \frac{1}{2} \mathbb{1}$.

8.6. Density matrices by tracing out an environment

Suppose that you have a system \mathcal{A} , described by a Hilbert space \mathcal{H}_A with a basis set $\{|\alpha\rangle_A\}$, in interaction with an environment \mathcal{B} , described by a Hilbert space \mathcal{H}_B with basis set $\{|\beta\rangle_B\}$. The total wave-function $|\Psi\rangle$ of the combined system $\mathcal{A} + \mathcal{B}$ will live in the tensor product space $\mathcal{H}_A \otimes \mathcal{H}_B$, which by definition has a (product) basis set $\{|\alpha\rangle_A \otimes |\beta\rangle_B\}$. The total wave-function, therefore, is generally a combination of those basis states

$$|\Psi\rangle = \sum_{\alpha, \beta} \Psi_{\alpha, \beta} |\alpha\rangle_A \otimes |\beta\rangle_B, \quad (8.34)$$

with appropriate coefficients $\Psi_{\alpha, \beta}$. Notice that, in general, the state $|\Psi\rangle$ is not *separable*, i.e. it cannot be written as a single product of a state of A times a state of B (as the basis states are):

$$|\Psi\rangle \neq |a\rangle_A \otimes |b\rangle_B.$$

This occurs only for some special choices of the coefficients $\Psi_{\alpha, \beta}$, take for ⁵ instance $\Psi_{\alpha, \beta} = a_{\alpha} b_{\beta}$, with $a_{\alpha} = \langle \alpha | a \rangle$ and $b_{\beta} = \langle \beta | b \rangle$. Whenever $|\Psi\rangle$ is not separable, one says that it is *entangled*. A simple example of such a case is, for two spin-1/2 particles, a singlet state.

Suppose I want to calculate the expectation value of an operator \hat{A} which involves only system A . Then the result is (simple proof, which uses orthogonality of $|\beta\rangle_B$):

$$\langle \Psi | \hat{A} | \Psi \rangle = \sum_{\alpha, \alpha'} \left(\sum_{\beta} \Psi_{\alpha', \beta}^* \Psi_{\alpha, \beta} \right) \langle \alpha' | \hat{A} | \alpha \rangle, \quad (8.35)$$

which you immediately recognise it can be expressed as:

$$\langle \Psi | \hat{A} | \Psi \rangle = \text{Tr}(\hat{A} |\Psi\rangle \langle \Psi|) = \text{Tr}_A(\hat{A} \hat{\rho}_A),$$

where

$$\hat{\rho}_A = \text{Tr}_B(|\Psi\rangle \langle \Psi|) \quad \Longrightarrow \quad \langle \alpha | \hat{\rho}_A | \alpha' \rangle = \sum_{\beta} \Psi_{\alpha', \beta}^* \Psi_{\alpha, \beta}. \quad (8.36)$$

We will now prove that $\hat{\rho}_A$ is a (generally) mixed state for system A . To prove it, observe that it is a manifestly Hermitian matrix. Moreover, its trace is 1:

$$\text{Tr}_A(\hat{\rho}_A) = \sum_{\alpha \beta} \Psi_{\alpha, \beta}^* \Psi_{\alpha, \beta} = \sum_{\alpha \beta} |\Psi_{\alpha, \beta}|^2 = 1.$$

⁵A so-called rank-1 matrix.

It remains to show that $\hat{\rho}_A$ is a *positive* operator. Let $|\Lambda_k\rangle$ be the orthonormal basis (in the space \mathcal{H}_A) which diagonalises $\hat{\rho}_A$. Therefore, we can write its spectral decomposition as:

$$\hat{\rho}_A = \sum_{k=1}^{N_S} \lambda_k |\Lambda_k\rangle \langle \Lambda_k| ,$$

where the eigenvalues λ_k are real, and $\sum_k \lambda_k = 1$, since the trace has to be 1. We now show that the eigenvalues are non-negative: $\lambda_k \geq 0$. Indeed:

$$\begin{aligned} \lambda_k = \text{Tr}_A(|\Lambda_k\rangle \langle \Lambda_k| \hat{\rho}_A) &= \sum_{\alpha', \alpha} \langle \alpha' | \Lambda_k \rangle \langle \Lambda_k | \alpha \rangle \langle \alpha | \hat{\rho}_A | \alpha' \rangle \\ &= \sum_{\alpha', \alpha} \langle \alpha' | \Lambda_k \rangle \langle \Lambda_k | \alpha \rangle \sum_{\beta} \Psi_{\alpha', \beta}^* \Psi_{\alpha, \beta} \\ &= \sum_{\beta} \left(\sum_{\alpha} \langle \Lambda_k | \alpha \rangle \Psi_{\alpha, \beta} \right) \left(\sum_{\alpha'} \langle \Lambda_k | \alpha' \rangle^* \Psi_{\alpha', \beta}^* \right) \\ &= \sum_{\beta} \left| \sum_{\alpha} \langle \Lambda_k | \alpha \rangle \Psi_{\alpha, \beta} \right|^2 \geq 0 . \end{aligned} \quad (8.37)$$

As a particularly interesting example, consider two spin-1/2 systems in the spin-singlet (rotationally invariant) entangled state:

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left(|+, \mathbf{z}\rangle_A \otimes |-, \mathbf{z}\rangle_B - |-, \mathbf{z}\rangle_A \otimes |+, \mathbf{z}\rangle_B \right) . \quad (8.38)$$

Then, by tracing over B you easily find that:

$$\hat{\rho}_A = \text{Tr}_B(|\Psi\rangle \langle \Psi|) = \frac{1}{2} \left(|+, \mathbf{z}\rangle_A \langle +, \mathbf{z}| + |-, \mathbf{z}\rangle_A \langle -, \mathbf{z}| \right) . \quad (8.39)$$

Hence, the infinite temperature mixed state emerges from tracing over B the entangled singlet state of a pair of spins.



Info: This is the simplest example of a process known as *purification*. You regard a mixed state (in the last example, an infinite temperature state) as obtained from a pure state of a system+environment by *partial trace* of an environment.

8.7. Schmidt decomposition

This section is slightly more advanced and can be skipped on a first reading. The material is based on the lecture notes by Preskill.

Let us start again from a pure state of a composite system with Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$. We call $d_A = \dim(\mathcal{H}_A)$ and $d_B = \dim(\mathcal{H}_B)$ the dimensions of the two Hilbert spaces, which we assume to be *finite*, although they can be possibly very large (and in general different). A pure state of the composite system is written as:

$$|\Psi\rangle = \sum_{\alpha, \beta} \Psi_{\alpha, \beta} |\alpha\rangle_A \otimes |\beta\rangle_B , \quad (8.40)$$

with appropriate coefficients $\Psi_{\alpha, \beta}$, where $\{|\alpha\rangle\}$ and $\{|\beta\rangle\}$ are *arbitrary orthonormal* bases for \mathcal{H}_A and \mathcal{H}_B , respectively.

i **Info:** Notice that you can regard the coefficients $\Psi_{\alpha,\beta}$ specifying the pure state $|\Psi\rangle$ as a complex matrix of dimension $d_A \times d_B$.

Now we re-shuffle the expression for $|\Psi\rangle$ as follows:

$$|\Psi\rangle = \sum_{\alpha,\beta} \Psi_{\alpha,\beta} |\alpha\rangle_A \otimes |\beta\rangle_B = \sum_{\alpha} |\alpha\rangle_A \otimes \left(\sum_{\beta} \Psi_{\alpha,\beta} |\beta\rangle_B \right) = \sum_{\alpha} |\alpha\rangle_A \otimes |\tilde{\alpha}\rangle_B \quad (8.41)$$

where

$$|\tilde{\alpha}\rangle_B = \sum_{\beta} \Psi_{\alpha,\beta} |\beta\rangle_B. \quad (8.42)$$

Warning: In general the different states $|\tilde{\alpha}\rangle_B$, one for each label α in the basis $\{|\alpha\rangle_A\}$, are *non-orthogonal*.

Let us now take a partial trace over B , obtaining the (generally) mixed state $\hat{\rho}_A$:

$$\hat{\rho}_A = \text{Tr}_B(|\Psi\rangle\langle\Psi|) = \sum_{k=1}^{N_S} \lambda_k |\Lambda_k\rangle\langle\Lambda_k| \quad (8.43)$$

where we have written the spectral decomposition of $\hat{\rho}_A$, with $N_S \leq d_A$ the so-called *Schmidt number*. The states $|\Lambda_k\rangle$ are by definition *orthonormal* and can be *extended* to an orthonormal basis for \mathcal{H}_A if $N_S < d_A$.

Let us now switch to such a basis $\{|\Lambda_k\rangle_A\}$ for \mathcal{H}_A . We can rewrite the pure state $|\Psi\rangle$ as follows:

$$|\Psi\rangle = \sum_{k,\beta} C_{k,\beta} |\Lambda_k\rangle_A \otimes |\beta\rangle_B = \sum_{k=1}^{N_S} |\Lambda_k\rangle_A \otimes |\tilde{\Lambda}_k\rangle_B \quad (8.44)$$

where, as before:

$$|\tilde{\Lambda}_k\rangle_B = \sum_{\beta} C_{k,\beta} |\beta\rangle_B. \quad (8.45)$$

If we recalculate the partial trace over B we get:

$$\hat{\rho}_A = \text{Tr}_B(|\Psi\rangle\langle\Psi|) = \sum_{k=1}^{N_S} \sum_{k'=1}^{N_S} \text{Tr}_B(|\Lambda_k\rangle\langle\Lambda_{k'}| \otimes |\tilde{\Lambda}_k\rangle\langle\tilde{\Lambda}_{k'}|) = \sum_{k=1}^{N_S} \sum_{k'=1}^{N_S} \langle\tilde{\Lambda}_{k'}|\tilde{\Lambda}_k\rangle |\Lambda_k\rangle\langle\Lambda_{k'}| \quad (8.46)$$

where the last step follows from recognising a resolution of the identity in B :

$$\text{Tr}_B(|\tilde{\Lambda}_k\rangle\langle\tilde{\Lambda}_{k'}|) = \sum_{\beta} \langle\beta|\tilde{\Lambda}_k\rangle\langle\tilde{\Lambda}_{k'}|\beta\rangle = \langle\tilde{\Lambda}_{k'}|\left(\sum_{\beta} |\beta\rangle\langle\beta|\right)|\tilde{\Lambda}_k\rangle = \langle\tilde{\Lambda}_{k'}|\tilde{\Lambda}_k\rangle. \quad (8.47)$$

Now you compare Eq. (8.43) and (8.46) and conclude that

$$\langle\tilde{\Lambda}_{k'}|\tilde{\Lambda}_k\rangle = \lambda_k \delta_{k',k}, \quad (8.48)$$

i.e., the states $\{|\tilde{\Lambda}_k\rangle\}$ are mutually orthogonal but have a norm different from 1. We can easily make then an orthonormal set by defining:

$$|\hat{\Lambda}_k\rangle_B = \frac{1}{\sqrt{\lambda_k}} |\tilde{\Lambda}_k\rangle_B = \frac{1}{\sqrt{\lambda_k}} \sum_{\beta} C_{k,\beta} |\beta\rangle_B. \quad (8.49)$$

Once again, if necessary, they can be extended to a full orthonormal basis for \mathcal{H}_B .

1 **Schmidt decomposition.** This finally leads to the *Schmidt decomposition*:

$$\left\{ \begin{array}{l} |\Psi\rangle = \sum_{k=1}^{N_S} \sqrt{\lambda_k} |\Lambda_k\rangle_A \otimes |\widehat{\Lambda}_k\rangle_B \\ \hat{\rho}_A = \text{Tr}_B(|\Psi\rangle\langle\Psi|) = \sum_{k=1}^{N_S} \lambda_k |\Lambda_k\rangle\langle\Lambda_k| \\ \hat{\rho}_B = \text{Tr}_A(|\Psi\rangle\langle\Psi|) = \sum_{k=1}^{N_S} \lambda_k |\widehat{\Lambda}_k\rangle\langle\widehat{\Lambda}_k| \end{array} \right. \quad (8.50)$$

The expression for the state $|\Psi\rangle$ is particularly simple and symmetric in the appropriate bases for the two Hilbert spaces. In particular, the last expression shows that $\hat{\rho}_B$, obtained by partial trace over A , has precisely the same Schmidt number N_S and therefore:

$$N_S \leq \min(d_A, d_B) . \quad (8.51)$$



Warning: Notice that the Schmidt decomposition, and the construction of the bases $|\Lambda_k\rangle_A$ and $|\widehat{\Lambda}_k\rangle_B$ depend on the chosen state $|\Psi\rangle$. If you change $|\Psi\rangle$, the bases will change. A word on the notation is also useful. Technically, the states $|\widehat{\Lambda}_k\rangle_B$ live in a different Hilbert space than the corresponding $|\Lambda_k\rangle_A$, and they are made of different combination of basis states. Nevertheless, the notation stresses the fact that they are *related* in a precise way, which is described by Eqs. (8.44), (8.49).

Purification. If I know both $\hat{\rho}_A = \text{Tr}_B(|\Psi\rangle\langle\Psi|)$ and $\hat{\rho}_B = \text{Tr}_A(|\Psi\rangle\langle\Psi|)$, and their spectrum $\{\lambda_k\}$ (common to both) is *non-degenerate* for all the $\lambda_k > 0$, then there is a *unique* state $|\Psi\rangle$ which provides a purification of the two density matrices. Indeed, let $\hat{\rho}_A = \sum_{k=1}^{N_S} \lambda_k |\Lambda_k\rangle\langle\Lambda_k|$ be the spectral decomposition of $\hat{\rho}_A$ which uniquely (up to a phase) identifies an orthonormal basis $\{|\Lambda_k\rangle_A\}$ (because the spectrum is non-degenerate). Similarly, you can write in a unique way a spectral decomposition $\hat{\rho}_B = \sum_{k=1}^{N_S} \lambda_k |\widehat{\Lambda}_k\rangle\langle\widehat{\Lambda}_k|$, which identifies an orthonormal basis $\{|\widehat{\Lambda}_k\rangle_B\}$. Next you pair-up these basis states and write:

$$|\Psi\rangle = \sum_{k=1}^{N_S} \sqrt{\lambda_k} |\Lambda_k\rangle_A \otimes |\widehat{\Lambda}_k\rangle_B ,$$

as the unique (up to a phase) purification of both density matrices. If on the contrary, there are spectral degeneracies, then in general the purification can be done in a large number of ways, but the analysis is more complicated. The example of the infinite temperature spin-1/2 case given above illustrates this fact.

Entanglement entropy. Starting from a pure state $|\Psi\rangle$ of the combined system, and calculating the partial traces we obtain the two states $\hat{\rho}_A = \text{Tr}_B(|\Psi\rangle\langle\Psi|)$ and $\hat{\rho}_B = \text{Tr}_A(|\Psi\rangle\langle\Psi|)$. As said, although these density matrices live in Hilbert spaces that might have a very different dimensionality, they have the same spectrum of non-negative eigenvalues $\{\lambda_k\}$, which is known as *entanglement spectrum*. The reason for this name is the following. You can calculate the *entanglement entropy*

$$S_A \stackrel{\text{def}}{=} -\text{Tr}_A(\hat{\rho}_A \log \hat{\rho}_A) = -\sum_{k=1}^{N_S} \lambda_k \log \lambda_k , \quad (8.52)$$

which, incidentally, is such that $S_A = S_B$. Evidently $S_A = 0$ if and only if $\hat{\rho}_A$ is a pure state ($N_S = 1$ and $\lambda_1 = 1$), while $S_A > 0$ if $\hat{\rho}_A$ is a mixed state. The situation $S_A = 0$ is realised when the initial state $|\Psi\rangle$ is *separable*. Indeed, if $|\Psi\rangle = |a\rangle_A \otimes |b\rangle_B$ then $\hat{\rho}_A = |a\rangle\langle a|$ is a pure state, and the same for $\hat{\rho}_B = |b\rangle\langle b|$.

8.7.1. The singular value decomposition (SVD)

An important application of the Schmidt decomposition is a crucial tool of linear algebra: the Singular Value Decomposition (SVD) of a general $d_A \times d_B$ complex matrix $\Psi_{\alpha,\beta}$.

Let us recall the two alternative expressions for the same state $|\Psi\rangle$ obtained in the two different bases:

$$\begin{cases} |\Psi\rangle = \sum_{\alpha,\beta} \Psi_{\alpha,\beta} |\alpha\rangle_A \otimes |\beta\rangle_B \\ |\Psi\rangle = \sum_{k=1}^{N_S} \sqrt{\lambda_k} |\Lambda_k\rangle_A \otimes |\hat{\Lambda}_k\rangle_B \end{cases} \quad (8.53)$$

Now express the states $|\Lambda_k\rangle_A$ and $|\hat{\Lambda}_k\rangle_B$ in the (generic) original orthonormal bases $|\alpha\rangle_A$ and $|\beta\rangle_B$:

$$\begin{cases} |\Lambda_k\rangle_A = \sum_{\alpha} |\alpha\rangle_A \langle \alpha | \Lambda_k \rangle = \sum_{\alpha} |\alpha\rangle_A (\mathbf{U})_{\alpha,k} \\ |\hat{\Lambda}_k\rangle_B = \sum_{\beta} |\beta\rangle_B \langle \beta | \hat{\Lambda}_k \rangle = \sum_{\beta} |\beta\rangle_B (\mathbf{V}^T)_{k,\beta} \end{cases}, \quad (8.54)$$

where we defined the two unitary matrices that describe the change between the two orthonormal sets:

$$\begin{cases} (\mathbf{U})_{\alpha,k} = \langle \alpha | \Lambda_k \rangle \\ (\mathbf{V})_{\beta,k} = \langle \beta | \hat{\Lambda}_k \rangle = (\mathbf{V}^T)_{k,\beta} \end{cases} \quad (8.55)$$

Recall that, technically speaking, \mathbf{U} is a $d_A \times d_A$ matrix of which only the first N_S columns with $k = 1 \dots N_S$ will be important. Similarly, \mathbf{V}^T should be intended as a $d_B \times d_B$ matrix of which only the first N_S rows will be important.

Now we substitute these expressions in the Schmidt decomposition for $|\Psi\rangle$:

$$\begin{aligned} |\Psi\rangle &= \sum_{k=1}^{N_S} \sqrt{\lambda_k} |\Lambda_k\rangle_A \otimes |\hat{\Lambda}_k\rangle_B \\ &= \sum_{k=1}^{N_S} \sqrt{\lambda_k} \sum_{\alpha,\beta} (\mathbf{U})_{\alpha,k} (\mathbf{V}^T)_{k,\beta} |\alpha\rangle_A \otimes |\beta\rangle_B \\ &= \sum_{\alpha,\beta} \underbrace{\left(\sum_{k=1}^{N_S} (\mathbf{U})_{\alpha,k} \sqrt{\lambda_k} (\mathbf{V}^T)_{k,\beta} \right)}_{\Psi_{\alpha,\beta}} |\alpha\rangle_A \otimes |\beta\rangle_B. \end{aligned} \quad (8.56)$$

Hence we conclude that:

$$\Psi_{\alpha,\beta} = \sum_{k=1}^{N_S} (\mathbf{U})_{\alpha,k} \sqrt{\lambda_k} (\mathbf{V}^T)_{k,\beta}. \quad (8.57)$$

Translated into a matrix factorization tool, the so-called SVD, if Ψ denotes the original $d_A \times d_B$ complex matrix with elements $[\Psi]_{\alpha,\beta} = \Psi_{\alpha,\beta}$, and Σ a $d_A \times d_B$ matrix which is mostly made of 0 with the $N_S \times N_S$ diagonal block given by $\sqrt{\lambda_k}$, i.e., $[\Sigma]_{k,k} = \sqrt{\lambda_k}$, the so-called matrix of the *singular values*, then Eq. (8.57) implies that we can factorise the matrix Ψ as follows:

$$\Psi = \mathbf{U} \Sigma \mathbf{V}^T. \quad (8.58)$$

Let us specialise this result to a general operator \hat{A} acting on the finite-dimensional Hilbert space \mathcal{H} .

❶

SVD of an operator. Any linear operator \hat{A} , even if it is not diagonalisable, can be always factorised as:

$$\hat{A} = \hat{U} \hat{D}_+ \hat{V}^\dagger, \quad (8.59)$$

where \hat{U} and \hat{V} are unitary, while \hat{D}_+ is **diagonal with real and positive** entries, the singular values.

8.8. Convex nature of density matrices

If $\hat{\rho}_1$ and $\hat{\rho}_2$ are two legitimate density matrices, then you can show that the convex linear combination

$$\hat{\rho}(\lambda) = \lambda \hat{\rho}_1 + (1 - \lambda) \hat{\rho}_2 \quad \forall \lambda \in [0, 1] \text{ real} \quad (8.60)$$

is also a density matrix. Indeed the Hermitean nature is obvious since $\lambda \in \mathbb{R}$. The fact that $\text{Tr}(\hat{\rho}(\lambda)) = 1$ is clear, and the positivity

$$\langle \phi | \hat{\rho}(\lambda) | \phi \rangle = \lambda \langle \phi | \hat{\rho}_1 | \phi \rangle + (1 - \lambda) \langle \phi | \hat{\rho}_2 | \phi \rangle \geq 0,$$

is also rather obvious.

❶

The convex subset of density matrices. Hence, the set of the density matrices $\{\hat{\rho}\}$ is a *convex subset* of the *real* vector space of the $d \times d$ Hermitean matrices with unit trace, whose dimension (on \mathbb{R}) is $2 \frac{d(d-1)}{2} + (d-1) = d^2 - 1$. For $d = 2$, the dimension of such a space is $d^2 - 1 = 3$ and can be therefore parameterised by a real three-dimensional vector \mathbf{p} in terms of the three Pauli operators (which are traceless):

$$\hat{\rho} = \frac{1}{2} \mathbb{1} + \frac{1}{2} \mathbf{p} \cdot \hat{\boldsymbol{\sigma}}.$$

We will return to a more detailed study of the $d = 2$ (spin-1/2) case in Sec. 8.9, where we will also show that the convex set of spin-1/2 density matrices is given by the three-dimensional ball $|\mathbf{p}| \leq 1$, the so-called *Bloch sphere*.

❶

Operational significance. We have seen that the probability of outcome of the eigenvalue a upon measuring an arbitrary observable \hat{A} on a state $\hat{\rho}$ can be expressed as

$$\text{Prob}(a|\hat{\rho}) = \text{Tr}(\hat{\Pi}_a \hat{\rho}).$$

Suppose you have experimental methods to prepare two states $\hat{\rho}_1$ and $\hat{\rho}_2$ and perform measurements on them. Suppose you decide to perform such measurements with probability λ on $\hat{\rho}_1$ and probability $(1 - \lambda)$ on $\hat{\rho}_2$, obtaining a with probability $\lambda \text{Prob}(a|\hat{\rho}_1) + (1 - \lambda) \text{Prob}(a|\hat{\rho}_2)$. Evidently, the experimental outcomes are totally indistinguishable from performing measurements on the state $\hat{\rho}(\lambda) = \lambda \hat{\rho}_1 + (1 - \lambda) \hat{\rho}_2$, which would give $\text{Prob}(a|\hat{\rho}(\lambda)) = \text{Tr}(\hat{\Pi}_a \hat{\rho}(\lambda))$. And this is true for *any* outcome a of *any* physical observable \hat{A} .

We now explore a few general consequences of such a convex structure. The first consequence is a generalisation of Eq. (8.60). If $\hat{\rho}_1 \cdots \hat{\rho}_N$ are density matrices, then the general convex linear

combination:⁶

$$\hat{\rho} = \sum_{j=1}^N p_j \hat{\rho}_j \quad \forall p_j \in [0, 1] \text{ real and such that } \sum_{j=1}^N p_j = 1 \quad (8.61)$$

is also a density matrix.

Definition 1

We call *extremal state* a density matrix which *cannot* be represented as a convex linear combination.

One can show the following result:

Theorem 8.1. A state is *extremal* if and only if it is a *pure state*.

Proof. To prove that *not pure* \implies *not extremal*, simply recall that a mixed density matrix $\hat{\rho}$ admits a spectral decomposition of the form:

$$\hat{\rho} = \sum_k \lambda_k |\Lambda_k\rangle\langle\Lambda_k|,$$

which is precisely a convex linear combination of pure states $|\Lambda_k\rangle\langle\Lambda_k|$. To prove that *pure* \implies *extremal*, take a pure state $\hat{\rho} = |\psi\rangle\langle\psi|$ and consider any vector $|\psi_\perp\rangle$ which is orthogonal to $|\psi\rangle$, i.e., $\langle\psi_\perp|\psi\rangle = 0$. If $\hat{\rho} = \lambda\hat{\rho}_1 + (1-\lambda)\hat{\rho}_2$ with $\lambda \neq 0, 1$ then you can write:

$$0 = \langle\psi_\perp|\hat{\rho}|\psi_\perp\rangle = \lambda\langle\psi_\perp|\hat{\rho}_1|\psi_\perp\rangle + (1-\lambda)\langle\psi_\perp|\hat{\rho}_2|\psi_\perp\rangle,$$

which implies that

$$\langle\psi_\perp|\hat{\rho}_1|\psi_\perp\rangle = \langle\psi_\perp|\hat{\rho}_2|\psi_\perp\rangle = 0 \quad \forall \psi_\perp \text{ such that } \langle\psi_\perp|\psi\rangle = 0.$$

Hence you conclude that $\hat{\rho}_1 = \hat{\rho}_2 = |\psi\rangle\langle\psi| \equiv \hat{\rho}$. ■

i

Why a pure state is called pure. The fact that pure states cannot be represented as convex combinations of other states justifies, in view of the operational significance of the convex combination, the term *pure*.

Finally, there is a result that generalises our discussion about the non-uniqueness of the ensemble representation of mixed states:

Theorem 8.2. A general *mixed* state $\hat{\rho}$ can be realised in an *infinite* number of ways as a convex combination of (generally non-orthogonal) pure states.

Proof. Consider a general mixed state operationally realised by an ensemble $E = \{p_j, |\psi_j\rangle\}$:

$$\hat{\rho} = \sum_{j=1}^{N_E} p_j |\psi_j\rangle\langle\psi_j| = \sum_j |\tilde{\psi}_j\rangle\langle\tilde{\psi}_j|,$$

⁶The generalization can be proven by induction, by writing:

$$\hat{\rho} = p_N \hat{\rho}_N + (1 - p_N) \sum_{j=1}^{N-1} \frac{p_j}{1 - p_N} \hat{\rho}_j$$

and observing that the sum represents a convex combination of $N - 1$ terms.

where $|\tilde{\psi}_j\rangle = \sqrt{p_j}|\psi_j\rangle$ are *not-normalized* (and possibly non-orthogonal). Now define the transformation induced by a general $N_E \times N_E$ unitary matrix:

$$|\tilde{\psi}_j\rangle = \sum_{j'} \mathbf{U}_{j,j'} |\tilde{\phi}_{j'}\rangle \quad \implies \quad |\tilde{\phi}_j\rangle = \sum_{j'} \mathbf{U}_{j,j'}^\dagger |\tilde{\psi}_{j'}\rangle .$$

We easily get, using $\mathbf{U}^\dagger \mathbf{U} = \mathbb{1}$:

$$\hat{\rho} = \sum_j |\tilde{\psi}_j\rangle \langle \tilde{\psi}_j| = \sum_j \sum_{j',j''} \mathbf{U}_{j,j'} \mathbf{U}_{j,j''}^* |\tilde{\phi}_{j'}\rangle \langle \tilde{\phi}_{j''}| = \sum_{j',j''} \left(\sum_j \mathbf{U}_{j'',j}^\dagger \mathbf{U}_{j,j'} \right) |\tilde{\phi}_{j'}\rangle \langle \tilde{\phi}_{j''}| = \sum_{j'} |\tilde{\phi}_{j'}\rangle \langle \tilde{\phi}_{j'}| .$$

In a similar way one can show that $\langle \tilde{\phi}_j | \tilde{\phi}_j \rangle = \langle \tilde{\psi}_j | \tilde{\psi}_j \rangle$. Hence we can properly normalise the transformed states as $|\phi_j\rangle = \frac{1}{\sqrt{p_j}} |\tilde{\phi}_j\rangle$, obtaining:

$$\hat{\rho} = \sum_{j=1}^{N_E} p_j |\psi_j\rangle \langle \psi_j| \equiv \sum_{j=1}^{N_E} p_j |\phi_j\rangle \langle \phi_j| .$$

■

❗

Ambiguity of preparation. The fact that different ensemble preparations give rise to the *same* density matrix is peculiar of quantum states, and is in sharp contrast with classical probability theory. Classically, if there are d possible outcomes of an experiment, and (say, for $d = 3$) event 1 has probability 0.3, event 2 probability 0.6 and event 3 probability 0.1, then you can express the outcome of such an experiment in a *unique way* as a convex superpositions of the *extremal* distributions $P^{k\text{-ext}}$ in which event k is certain and the other events have 0 probability (the analogues of pure states), i.e. $P_j^{k\text{-ext}} = \delta_{j,k}$. In the example above:

$$P_j = 0.3 P_j^{1\text{-ext}} + 0.6 P_j^{2\text{-ext}} + 0.1 P_j^{3\text{-ext}} .$$

We will illustrate this highly ambiguous nature of mixed states with spin-1/2 examples in Sec. 8.9.

8.9. The spin-1/2 case and the Bloch sphere

We have already mentioned that for the spin-1/2 case ($d = 2$) the convex set of density matrices can be conveniently parameterised by by a real three-dimensional vector \mathbf{p} in terms of the three Pauli operators:

$$\hat{\rho}_{\mathbf{p}} = \frac{1}{2} \mathbb{1} + \frac{1}{2} \mathbf{p} \cdot \hat{\boldsymbol{\sigma}} . \quad (8.62)$$

Exercise 8.2. After showing that

$$(\mathbf{p} \cdot \hat{\boldsymbol{\sigma}})^2 = |\mathbf{p}|^2 \mathbb{1} ,$$

calculate $\hat{\rho}_{\mathbf{p}}^2$ and verify that

$$\hat{\rho}_{\mathbf{p}}^2 = \frac{1 + |\mathbf{p}|^2}{4} \mathbb{1} + \frac{1}{2} \mathbf{p} \cdot \hat{\boldsymbol{\sigma}} .$$

Deduce that $\text{Tr}(\hat{\rho}_{\mathbf{p}}^2) \leq 1$ if and only if $|\mathbf{p}| \leq 1$, and that $\hat{\rho}_{\mathbf{p}}^2 = \hat{\rho}_{\mathbf{p}}$ when $|\mathbf{p}| = 1$.

Hence the three-dimensional convex set of $d = 2$ density matrices can be visualised as a three-dimensional sphere ⁷ $|\mathbf{p}| \leq 1$, with pure states staying at the boundary $|\mathbf{p}| = 1$. This sphere is known as *Bloch sphere*, and the vector \mathbf{p} is known as *Bloch vector*.

⁷Strictly speaking a *ball*, whose boundary is the *sphere* $|\mathbf{p}| = 1$.

If $\mathbf{p} = \mathbf{n}$, where \mathbf{n} is a unit vector, then we recognise that $\hat{\rho}_{\mathbf{n}}$ is the projector on the spin state $|+, \mathbf{n}\rangle$

$$\hat{\rho}_{\mathbf{n}} = \frac{1}{2}\mathbb{1} + \frac{1}{2}\mathbf{n} \cdot \hat{\boldsymbol{\sigma}} \equiv \hat{\Pi}_{|+, \mathbf{n}\rangle}. \quad (8.63)$$

By varying \mathbf{n} on the unit sphere we realise all possible pure states.

i **States at the boundary of the convex set.** The fact that all states at the boundary of the convex set of density matrices are *pure* is very special of the $d = 2$ case. For higher d , pure states will always be at the boundary, but there are in general states at the boundary that are also mixed. Indeed, the boundary is realised by density matrices which have an eigenvalue *zero* (if you move to negative eigenvalue you would go out of the set), but having an eigenvalue zero is not enough to guarantee that the state is pure when $d > 2$. For $d = 2$, on the contrary, you are sure that $\lambda_1 = 1$ when $\lambda_2 = 0$.

Any mixed state has $|\mathbf{p}| < 1$, hence is *strictly inside* the Bloch sphere. Geometrically, it is quite intuitive that we can represent it in many (indeed, infinitely many) ways as convex combination of pure states, i.e. points on the boundary of the sphere. For instance, consider the infinite-temperature mixed state $\hat{\rho} = \frac{1}{2}\mathbb{1}$, which is associated to the origin $\mathbf{p} = \mathbf{0}$. Evidently, you can represent it as:

$$\frac{1}{2}\mathbb{1} = \frac{1}{2}\hat{\rho}_{\mathbf{n}} + \frac{1}{2}\hat{\rho}_{-\mathbf{n}} \quad \forall \mathbf{n}. \quad (8.64)$$

Notice that since $\langle -, \mathbf{n} | +, \mathbf{n} \rangle = 0$ we have realised $\hat{\rho}$ as an equal probability admixture of two *orthogonal* pure states.

i **A simple example of state tomography.** To measure the Bloch vector \mathbf{p} (also known as *polarization*) that a certain given density matrix $\hat{\rho}_{\mathbf{p}}$ has, imagine performing repeated measurements of the spin in direction \mathbf{n} , and calculate the average $\langle \mathbf{n} \cdot \hat{\boldsymbol{\sigma}} \rangle = \text{Tr}(\mathbf{n} \cdot \hat{\boldsymbol{\sigma}} \hat{\rho}_{\mathbf{p}})$. You easily calculate that

$$\text{Tr}(\mathbf{n} \cdot \hat{\boldsymbol{\sigma}} \hat{\rho}_{\mathbf{p}}) = \mathbf{n} \cdot \mathbf{p}. \quad (8.65)$$

Hence, by repeated measurements of the average spin in 3 orthogonal directions, for instance x, y, z , you fully determine the vector \mathbf{p} , completely determining the state $\hat{\rho}_{\mathbf{p}}$. This procedure is called *state tomography*.

Exercise 8.3. Show that

$$\frac{1}{2} \text{Tr}(\hat{\sigma}^\alpha \hat{\sigma}^\beta) = \delta_{\alpha, \beta}. \quad (8.66)$$

Using that, verify Eq. (8.65).

When $0 < |\mathbf{p}| < 1$, there are still infinitely many convex combinations, but most of them in terms of *non-orthogonal pure states*. For instance (and really without loss of generality), take $\mathbf{p} = (0, 0, p_z)$ with $0 < p_z < 1$:

$$\hat{\rho}_{\mathbf{p}} = \frac{1}{2}\mathbb{1} + \frac{p_z}{2}\hat{\sigma}^z. \quad (8.67)$$

The two eigenvalues are $\lambda_1 = (1 + p_z)/2$ and $\lambda_2 = (1 - p_z)/2$ and there is no spectral degeneracy. The spectral decomposition in terms of orthogonal states reads (as you can easily convince yourself):

$$\hat{\rho}_{\mathbf{p}} = \lambda_1 \hat{\rho}_{\mathbf{z}} + \lambda_2 \hat{\rho}_{-\mathbf{z}}.$$

But there are infinitely many convex decompositions in terms of pairs of non-orthogonal pure states. For instance, consider the circle obtained by intersection of the Bloch sphere with the plane at $z = p_z$, which can be parameterised by an angle ϕ as

$$\mathbf{n} = (\sqrt{1 - p_z^2} \cos \phi, \sqrt{1 - p_z^2} \sin \phi, p_z).$$

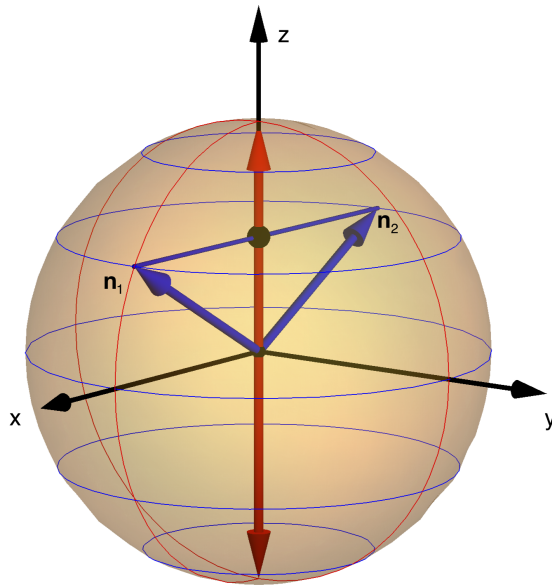


Figure 8.3.: The Bloch sphere. The same mixed density matrix (here with $p_z = \frac{1}{2}$) is represented in terms of two orthogonal pure states (along z), or two non-orthogonal pure states.

Take two arbitrary opposite points on that circle (associated to ϕ_1 and $\phi_2 = \phi_1 + \pi$), call the \mathbf{n}_1 and \mathbf{n}_2 , and observe that:

$$\hat{\rho} = \lambda_1 \hat{\rho}_{\mathbf{z}} + \lambda_2 \hat{\rho}_{-\mathbf{z}} = \frac{1}{2} \hat{\rho}_{\mathbf{n}_1} + \frac{1}{2} \hat{\rho}_{\mathbf{n}_2} . \quad (8.68)$$

Fig. 8.3 illustrates the simple geometry behind such a construction. Obviously, the non-orthogonality of the two states involved follows from the fact that, in general, $\langle +, \mathbf{n}_1 | +, \mathbf{n}_2 \rangle > 0$.

Exercise 8.4. Calculate $\langle +, \mathbf{n}_1 | +, \mathbf{n}_2 \rangle$ for two arbitrary spin (pure) states on the Bloch sphere.

In full generality, if \mathbf{p} is inside the unit ball, then you can take any point \mathbf{n}_1 on the sphere, define the chord from \mathbf{n}_1 that contains \mathbf{p} and ends at a second point \mathbf{n}_2 on the sphere, and express \mathbf{p} as a convex combination of \mathbf{n}_1 and \mathbf{n}_2 :

$$\mathbf{p} = \lambda \mathbf{n}_1 + (1 - \lambda) \mathbf{n}_2 .$$

As you see, there is two-parameter family of chords of this type. For all these choices, as you readily verify

$$\hat{\rho}_{\mathbf{p}} = \lambda \hat{\rho}_{\mathbf{n}_1} + (1 - \lambda) \hat{\rho}_{\mathbf{n}_2} ,$$

realises the promised convex combination of (non-orthogonal) pure states.

9. Open Quantum Systems and Quantum Maps

We recall that a density matrix is a *positive Hermitian operator* — which we denote by $\hat{\rho} \geq 0$ — with a unit trace: $\text{Tr } \hat{\rho} = 1$. Recall also that one of the ways in which density matrices emerge is by “tracing out an environment”. In this respect, the role of the system-environment interaction, which we have so far not discussed, is crucial. Indeed, if the total Hamiltonian is split as

$$\hat{H}^{\text{tot}}(t) = \hat{H}^{\text{S}}(t) + \hat{H}^{\text{B}} + g\hat{H}^{\text{SB}}, \quad (9.1)$$

where $\hat{H}^{\text{S}}(t)$ is the system Hamiltonian, \hat{H}^{B} is the bath (or environment) ¹ Hamiltonian and \hat{H}^{SB} describes the interaction between the two, with an overall coupling constant g , then starting from a pure separable state

$$|\Psi^{\text{SB}}(0)\rangle = |\psi^{\text{S}}(0)\rangle \otimes |\phi^{\text{B}}(0)\rangle \quad (9.2)$$

the evolution leads to a state

$$|\Psi^{\text{SB}}(t)\rangle = \hat{U}_{\text{tot}}(t)|\Psi^{\text{SB}}(0)\rangle \quad (9.3)$$

which is generally *not separable*, except for the trivial case $g = 0$. ²

9.1. Kraus representation of the dynamics

In general, for $g > 0$, by tracing out the environment using an arbitrary orthonormal basis $\{|\phi_k^{\text{B}}\rangle\}$, we get a density matrix for the system:

$$\begin{aligned} \hat{\rho}_{\text{S}}(t) &= \text{Tr}_{\text{B}} |\Psi^{\text{SB}}(t)\rangle\langle\Psi^{\text{SB}}(t)| \\ &= \text{Tr}_{\text{B}} \left(\hat{U}_{\text{tot}}(t)|\psi^{\text{S}}(0)\rangle\langle\psi^{\text{S}}(0)| \otimes |\phi^{\text{B}}(0)\rangle\langle\phi^{\text{B}}(0)| \hat{U}_{\text{tot}}^\dagger(t) \right) \\ &= \sum_k \langle\phi_k^{\text{B}}|\hat{U}_{\text{tot}}(t)|\phi^{\text{B}}(0)\rangle |\psi^{\text{S}}(0)\rangle\langle\psi^{\text{S}}(0)| \langle\phi^{\text{B}}(0)|\hat{U}_{\text{tot}}^\dagger(t)|\phi_k^{\text{B}}\rangle \\ &= \sum_k \hat{K}_k(t) |\psi^{\text{S}}(0)\rangle\langle\psi^{\text{S}}(0)| \hat{K}_k^\dagger(t) \\ &= \sum_k \hat{K}_k(t) \hat{\rho}_{\text{S}}(0) \hat{K}_k^\dagger(t) \end{aligned} \quad (9.4)$$

where we have defined the Kraus operators acting on the system Hilbert space \mathcal{H}_{S} but labelled by the quantum number k of the environment state $|\phi_k^{\text{B}}\rangle$

$$\hat{K}_k(t) \stackrel{\text{def}}{=} \langle\phi_k^{\text{B}}|\hat{U}_{\text{tot}}(t)|\phi^{\text{B}}(0)\rangle \implies \hat{K}_k^\dagger(t) = \langle\phi^{\text{B}}(0)|\hat{U}_{\text{tot}}^\dagger(t)|\phi_k^{\text{B}}\rangle. \quad (9.5)$$

¹A bath (or reservoir) is an environment with very many (in principle, infinite) degrees of freedom. We assume in our derivations that the number of degrees of freedom is *finite*, but otherwise unspecified: nothing forbids us from applying our derivations to “bath” which are made by a single spin-1/2, in which case a possible alternative name is *ancilla*.

²Clearly, for $g = 0$ we have:

$$|\Psi^{\text{SB}}(t)\rangle = \left(\hat{U}_{\text{S}}(t)|\psi^{\text{S}}(0)\rangle \right) \otimes \left(\hat{U}_{\text{B}}(t)|\phi^{\text{B}}(0)\rangle \right),$$

hence the state remains separable.

Observe that the same algebra holds for a more general separable state in which the initial system state $\hat{\rho}_s(0)$ is *mixed*. For a finite-dimensional bath system, there are in principle $D_K \leq d_B = \dim(\mathcal{H}_B)$ Kraus operators. Notice that *in general* the Kraus operators are neither unitary nor Hermitian (hence they are not projector operators).

A simple to prove³ but important property of Kraus operators is that:

①

Completeness.

$$\sum_k \hat{K}_k^\dagger \hat{K}_k = \mathbb{1}_S . \quad (9.7)$$

Such a property is instrumental in showing that indeed density matrices evolve into density matrices by application of the Kraus operators.

The conservation of the unit trace is trivial:

$$\mathrm{Tr}_S \hat{\rho}_s(t) = \sum_k \mathrm{Tr}_S \left(\hat{K}_k(t) \hat{\rho}_s(0) \hat{K}_k^\dagger(t) \right) = \sum_k \mathrm{Tr}_S \left(\hat{K}_k^\dagger(t) \hat{K}_k(t) \hat{\rho}_s(0) \right) = \mathrm{Tr}_S \hat{\rho}_s(0) . \quad (9.8)$$

The conservation of the Hermitian nature is equally trivial. To prove positivity, we have to show that $\forall |\chi^S\rangle$ we have $\langle \chi^S | \hat{\rho}_s(t) | \chi^S \rangle \geq 0$. Let us carry out the proof for the case in which $\hat{\rho}_s(0) = |\psi^S(0)\rangle\langle\psi^S(0)|$, the general result following by linearity. We have:

$$\langle \chi^S | \hat{\rho}_s(t) | \chi^S \rangle = \sum_k \langle \chi^S | \hat{K}_k(t) | \psi^S(0) \rangle \langle \psi^S(0) | \hat{K}_k^\dagger(t) | \chi^S \rangle = \sum_k \left| \langle \chi^S | \hat{K}_k(t) | \psi^S(0) \rangle \right|^2 \geq 0 . \quad (9.9)$$

We said that *in general* the Kraus operators are not unitary. But there is a trivial case in which indeed unitarity is obtained. This is the case $g = 0$, in which case:

$$\hat{\rho}_s(t) = \hat{U}_s(t) \hat{\rho}_s(0) \hat{U}_s^\dagger(t) , \quad (9.10)$$

and, as we already know, purity is preserved.

Let us now show that *purity is not preserved* by the general Kraus representation, hence there is no single effective $\hat{U}_s(t)$ which can represent the evolution in a unitary fashion. To show this, we calculate $\mathrm{Tr}_S \hat{\rho}_s^2(t)$, which is an indicator of purity, in the sense that a mixed state is signalled by $\mathrm{Tr}_S \hat{\rho}_s^2(t) < 1$. Again, we assume that the initial state is pure, $\hat{\rho}_s(0) = |\psi^S(0)\rangle\langle\psi^S(0)|$. We have:

$$\begin{aligned} \mathrm{Tr}_S \hat{\rho}_s^2(t) &= \sum_{k,k'} \mathrm{Tr}_S \left(\hat{K}_k(t) \hat{\rho}_s(0) \hat{K}_k^\dagger(t) \hat{K}_{k'}(t) \hat{\rho}_s(0) \hat{K}_{k'}^\dagger(t) \right) \\ &= \sum_{k,k'} \left| \langle \psi^S(0) | \hat{K}_{k'}^\dagger(t) \hat{K}_k(t) | \psi^S(0) \rangle \right|^2 . \end{aligned} \quad (9.11)$$

Let us now consider each term appearing in the sum. To shorten our notation, we define $|v_k\rangle = \hat{K}_k | \psi^S(0) \rangle$, omitting also the time label in the Kraus operator. Then we have, by the Cauchy-Schwartz inequality:

$$\begin{aligned} \left| \langle \psi^S(0) | \hat{K}_{k'}^\dagger \hat{K}_k | \psi^S(0) \rangle \right|^2 &= |\langle v_{k'} | v_k \rangle|^2 \\ &\leq \langle v_{k'} | v_{k'} \rangle \langle v_k | v_k \rangle \\ &\leq \langle \psi^S(0) | \hat{K}_{k'}^\dagger \hat{K}_{k'} | \psi^S(0) \rangle \langle \psi^S(0) | \hat{K}_k^\dagger \hat{K}_k | \psi^S(0) \rangle \end{aligned} \quad (9.12)$$

³Simply observe that:

$$\begin{aligned} \sum_k \hat{K}_k^\dagger \hat{K}_k &= \sum_k \langle \phi^B(0) | \hat{U}_{\mathrm{tot}}^\dagger(t) | \phi_k^B \rangle \langle \phi_k^B | \hat{U}_{\mathrm{tot}}(t) | \phi^B(0) \rangle \\ &= \langle \phi^B(0) | \hat{U}_{\mathrm{tot}}^\dagger(t) \hat{U}_{\mathrm{tot}}(t) | \phi^B(0) \rangle = \langle \phi^B(0) | \mathbb{1}_{\mathrm{tot}} | \phi^B(0) \rangle = \mathbb{1}_S \end{aligned} \quad (9.6)$$

where the equality is realized only if $|v_k\rangle$ and $|v_{k'}\rangle$ are *parallel*. You realise immediately that for a generic $|\psi^S(0)\rangle_S$ this is impossible, unless there is a *single Kraus operator*, in which case $|v_k\rangle \equiv |v_{k'}\rangle$. Hence we conclude that:

$$\begin{aligned} \text{Tr}_S \hat{\rho}_S^2(t) &= \sum_{k,k'} \left| \langle \psi^S(0) | \hat{K}_k^\dagger \hat{K}_{k'} | \psi^S(0) \rangle \right|^2 \\ &\leq \left(\sum_k \langle \psi^S(0) | \hat{K}_k^\dagger \hat{K}_k | \psi^S(0) \rangle \right)^2 = 1, \end{aligned} \quad (9.13)$$

where the equality — hence purity preservation — is realized only if there is a single Kraus operator, i.e.,

$$\hat{\rho}_S(t) = \hat{K}(t) \hat{\rho}_S(0) \hat{K}^\dagger(t), \quad (9.14)$$

in which case we must have

$$\hat{K}^\dagger \hat{K} = \mathbb{1}_S, \quad (9.15)$$

hence \hat{K} is *unitary*, at least for a finite-dimensional \mathcal{H}_S .

i

The Kraus map. Summarising, any unitary dynamics in a suitably larger Hilbert space induces, when regarded within the system Hilbert space, a linear transformation between density matrices of the form:

$$\hat{\rho}_S(0) \longrightarrow \hat{\rho}_S(t) = \sum_k \hat{K}_k(t) \hat{\rho}_S(0) \hat{K}_k^\dagger(t) \quad \text{with} \quad \sum_k \hat{K}_k^\dagger \hat{K}_k = \mathbb{1}_S. \quad (9.16)$$

This is known as a *Kraus quantum map*. We will discuss more about this in Sec. 9.4. Recalling that this comes from choosing an arbitrary basis in the “bath” Hilbert space, you might suspect that there is a large arbitrariness involved in the process, as we will discuss in Secs. 9.3 and 9.2.3.

Example: the cNOT gate. Consider a case in which system and environment are *both* a single Qbit. We denote the corresponding Hilbert spaces by indicating the computational basis $\{|0\rangle, |1\rangle\}$, and a general state as $|\psi\rangle = z_0|0\rangle + z_1|1\rangle$. The cNOT (or control-NOT) — the control being the system Qbit —, is defined by the following unitary:

$$\begin{cases} \hat{U}_{\text{cNOT}} |0^S\rangle \otimes |\phi^B\rangle = |0^S\rangle \otimes |\phi^B\rangle \\ \hat{U}_{\text{cNOT}} |1^S\rangle \otimes |\phi^B\rangle = |1^S\rangle \otimes (\hat{\sigma}^x |\phi^B\rangle) \end{cases} \quad (9.17)$$

By linearity, this fully defines \hat{U}_{cNOT} in the product Hilbert space of the two Qbits. In the standard computational basis of the tensor product:

$$\{|0^S\rangle \otimes |0^B\rangle = |00\rangle, |0^S\rangle \otimes |1^B\rangle = |01\rangle, |1^S\rangle \otimes |0^B\rangle = |10\rangle, |1^S\rangle \otimes |1^B\rangle = |11\rangle\},$$

the 4×4 matrix representing the cNOT is:

$$\hat{U}_{\text{cNOT}} \rightarrow \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right). \quad (9.18)$$

Consider now \hat{U}_{cNOT} applied to a separable state made of a general linear combination of the system Qbit times the $|0\rangle_B$ state:

$$|\Psi(t = \tau)\rangle = \hat{U}_{\text{cNOT}}(z_0|0^S\rangle + z_1|1^S\rangle) \otimes |0^B\rangle = z_0|00\rangle + z_1|11\rangle, \quad (9.19)$$

where τ is, supposedly, the time over which the gate has acted. Notice how the interaction has created an *entangled state* starting from a product state. As a consequence, the system density matrix obtained by partial trace is:

$$\hat{\rho}_s(\tau) = \text{Tr}_B |\Psi(\tau)\rangle\langle\Psi(\tau)| = |z_0|^2|0^s\rangle\langle 0^s| + |z_1|^2|1^s\rangle\langle 1^s|. \quad (9.20)$$

Hence, the populations are preserved, but *coherence is lost*: the final state of the system Qbit is *mixed*. Incidentally, this is the same density matrix you would obtain as a “post-measurement” ensemble upon measuring the system Qbit in the computational basis, i.e., along $\hat{\sigma}^z$.

Recalling that the cNOT gate can be written as

$$\hat{U}_{\text{cNOT}} = \frac{1}{2}(1 + \hat{\sigma}_1^z) + \frac{1}{2}(1 - \hat{\sigma}_1^z) \hat{\sigma}_2^x, \quad (9.21)$$

with the NOT gate $\hat{\sigma}^x$ on the second Qbit acting only when the first Qbit (the control Qbit) is $|\downarrow\rangle = |1\rangle$, the two Kraus operators associated to the standard computational basis are in this case *projectors*:

$$\hat{K}_0 = \langle 0^B | \hat{U}_{\text{cNOT}} | 0^B \rangle = \frac{1}{2}(1 + \hat{\sigma}_1^z) = \hat{\Pi}_0^s$$

and

$$\hat{K}_1 = \langle 1^B | \hat{U}_{\text{cNOT}} | 0^B \rangle = \frac{1}{2}(1 - \hat{\sigma}_1^z) = \hat{\Pi}_1^s.$$

The two Kraus operators lead to the two collapsed states:

$$|\psi_0^s\rangle = \frac{\hat{K}_0|\psi^s\rangle}{\|\hat{K}_0|\psi^s\rangle\|} = |0^s\rangle \quad \text{and} \quad |\psi_1^s\rangle = \frac{\hat{K}_1|\psi^s\rangle}{\|\hat{K}_1|\psi^s\rangle\|} = |1^s\rangle.$$

As mentioned, the system density matrix in Eq. (9.20) coincides with the post-measurement density matrix you would have written down by doing a von Neumann measurement of the system Qbit in the computational basis. Indeed, the probability of measuring 0 and 1 are:

$$P_0 = \langle \psi^s | \hat{\Pi}_0^s | \psi^s \rangle = |z_0|^2 \quad \text{and} \quad P_1 = \langle \psi^s | \hat{\Pi}_1^s | \psi^s \rangle = |z_1|^2,$$

and the post-measurement mixed state is:

$$\hat{\rho}_{p-m}^s = |z_0|^2|0^s\rangle\langle 0^s| + |z_1|^2|1^s\rangle\langle 1^s|. \quad (9.22)$$

9.2. Quantum measurements and POVM

Quantum measurements provide clear illustrations for Kraus maps, adding considerable physical understanding of the subject. We will start from the standard von Neumann paradigm of projective measurement, which generalises the previous CNOT example. We then move to a more general setting, that of *generalised measurements*, which provides an even more illuminating discussion about Kraus maps.

9.2.1. von Neumann projective measurements

Recall that in a von Neumann projective measurement, you consider a state $|\psi^s\rangle$, with the associated initial pure-state density matrix $\hat{\rho}_{\text{in}}^s = |\psi^s\rangle\langle\psi^s|$. You assume to have a large ensemble of identical pure states $|\psi^s\rangle$, over which you measure a system observable A , obtaining the eigenvalues a of the associated Hermitian operator \hat{A} with probability

$$P_a = \text{Prob}(a|\psi^s) = \langle \psi^s | \hat{\Pi}_a | \psi^s \rangle = \|\hat{\Pi}_a|\psi^s\rangle\|^2.$$

After measuring the eigenvalue a , the state collapses to

$$|\psi^S\rangle \xrightarrow{\text{measure } a} |\psi_a^S\rangle = \frac{\hat{\Pi}_a |\psi^S\rangle}{\|\hat{\Pi}_a |\psi^S\rangle\|}.$$

The ensemble of states *after the measurement* — assuming we do not make any filtering selection of the states according to the outcome of the measurement — is given by $E = \{P_a, |\psi_a^S\rangle\}$, and is represented by post-measurement density matrix

$$\hat{\rho}_{p-m}^S = \sum_a P_a |\psi_a^S\rangle \langle \psi_a^S| = \sum_a P_a \frac{\hat{\Pi}_a |\psi^S\rangle \langle \psi^S| \hat{\Pi}_a}{\|\hat{\Pi}_a |\psi^S\rangle\|^2} = \sum_a \hat{\Pi}_a |\psi^S\rangle \langle \psi^S| \hat{\Pi}_a = \sum_a \hat{\Pi}_a \hat{\rho}_{in}^S \hat{\Pi}_a. \quad (9.23)$$

Question: What if the initial state is mixed?

What happens if the initial state is not pure, but an ensemble preparation $E = \{p_k, |\psi_k^S\rangle\}$. It turns out that the answer for the post-measurement state is still correct:

$$\hat{\rho}_{p-m}^S = \sum_a \hat{\Pi}_a \hat{\rho}_{in}^S \hat{\Pi}_a, \quad (9.24)$$

but the various collapsed states are now *no longer pure states*, as you will learn by doing the following exercise.

Exercise 9.1. Take an initial mixed state

$$\hat{\rho}_{in}^S = \sum_{k=1}^{N_E} p_k |\psi_k^S\rangle \langle \psi_k^S|.$$

Call $P_{a|k}$ the probability of measuring the eigenvalue a on the state $|\psi_k^S\rangle$, and $|\psi_{a|k}^S\rangle$ the collapsed pure state after measuring a .

1) Show that the measurement of a is associated to a collapsed mixed state

$$\hat{\rho}_a^S = \sum_{k=1}^{N_E} p_k |\psi_{a|k}^S\rangle \langle \psi_{a|k}^S|$$

2) The probability of measuring a over the whole ensemble is

$$P_a = \sum_{k=1}^{N_E} P_{a|k} p_k.$$

Express P_a in terms of $\hat{\rho}_a^S$ and of the projector $\hat{\Pi}_a$.

3) Show that the post-measurement density matrix can be written as:

$$\hat{\rho}_{p-m}^S = \sum_a P_a \hat{\rho}_a^S = \sum_a \hat{\Pi}_a \hat{\rho}_{in}^S \hat{\Pi}_a.$$

i

The projective measurement as a Kraus quantum map. The process of a projective von Neumann measurement contained in Eq. (9.24) can therefore be regarded as a Kraus quantum map where the Kraus operators are simply the (Hermitian) projectors $\hat{\Pi}_a = \hat{\Pi}_a^\dagger$ associated to the eigenvalues of the operator we are measuring. The Kraus completeness in Eq. (9.7) follows from the completeness property of projectors associated to Hermitian operators.

9.2.2. Generalised quantum measurements

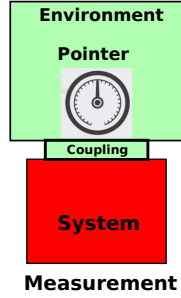


Figure 9.1: Illustration of a quantum measurement: a pointer (measurement apparatus, possibly microscopic) interacts with the system, while many other degrees of freedom (the environmental bath) are present. The measurement consists of a projective von Neumann measurement of some pointer operator \hat{B} .

Let us now imagine that the bath includes a measurement *pointer*, see Fig. 9.1, with Hilbert space \mathcal{H}_P plus possibly a large environment \mathcal{H}_E :

$$\mathcal{H}_B = \mathcal{H}_P \otimes \mathcal{H}_E \quad (9.25)$$

Assume that the system+bath state is initially the usual product state $|\psi^S\rangle \otimes |\phi^B\rangle$, with $|\phi^B\rangle \in \mathcal{H}_B$.

i

Info: So far, we chose an arbitrary basis of \mathcal{H}_B to do our algebra. Now we think differently, and we imagine measuring some Hermitian operator \hat{B} for the *pointer* (*not* the system). The operator \hat{B} has eigenvalues b and an associated orthonormal basis $\{|\phi_{b,q}^B\rangle\}$, where q denotes the possible degeneracy of the eigenvalue b .^a To have a concrete example in mind, you could think that the “pointer” is a single spin, and \hat{B} is a spin operator along a certain direction \mathbf{n} . As a second example, you could think of the “pointer” as the electromagnetic field, with its photon modes. More about this in later discussions.

^aEspecially in the presence of a large bath, the fact that an eigenvalue of the pointer operator \hat{B} is degenerate is almost unavoidable.

Let us now return to a general setting of arbitrary system and pointer/environment. Let the entangling interaction \hat{U}_{tot} act on the state, leading to⁴

$$|\Psi^{SB}\rangle = \hat{U}_{\text{tot}} \left(|\psi^S\rangle \otimes |\phi^B\rangle \right) = \sum_{b,q} \left(\hat{K}_{b,q} |\psi^S\rangle \right) \otimes |\phi_{b,q}^B\rangle, \quad (9.26)$$

where — recall the original derivation of the Kraus operators in Eqs. (9.4,9.5) — we have that:

$$\hat{K}_{b,q} = \langle \phi_{b,q}^B | \hat{U}_{\text{tot}} | \phi^B \rangle, \quad (9.27)$$

which are in general *not Hermitian*, but, in order to preserve the norm, see Eq. (9.7), do satisfy:

$$\sum_{b,q} \hat{K}_{b,q}^\dagger \hat{K}_{b,q} = \mathbb{1}_S. \quad (9.28)$$

i

Pre-measurement. This part of the process in which the interaction, through a non-separable \hat{U}_{tot} creates *entanglement* between the system and the pointer+environment is often called the *pre-measurement*. Notice that, in general, the pre-measurement changes both the state of the system as well as that of the environment.

⁴Simply use the total identity to write:

$$|\Psi^{SB}\rangle = \sum_{b,q} \left(|\phi_{b,q}^B\rangle \langle \phi_{b,q}^B| \right) \hat{U}_{\text{tot}} \left(|\psi^S\rangle \otimes |\phi^B\rangle \right) = \sum_{b,q} \left(\hat{K}_{b,q} |\psi^S\rangle \right) \otimes |\phi_{b,q}^B\rangle.$$

Suppose that the pointer interacts in some way with a *further apparatus* — collectively included in the large bath — that performs a projective von Neumann measurement of the pointer operator \hat{B} , *reading* the eigenvalue b of such observable, and collapsing the environment/pointer state on the corresponding eigenvector basis $|\phi_{b,q}^B\rangle$ of \hat{B} , with associated orthogonal projectors $\hat{\Pi}_b^B = \sum_q |\phi_{b,q}^B\rangle\langle\phi_{b,q}^B|$, hence $\mathbb{1}_S \otimes \hat{\Pi}_b^B$ for the combined system.

i

The pointer read-out (measurement). The probability of obtaining the outcome “ b ” in an ensemble of measurements of this type is evidently: ^a

$$P_b = \text{Prob}(b|\Psi^{SB}) = \langle\Psi^{SB}|\mathbb{1}_S \otimes \hat{\Pi}_b^B|\Psi^{SB}\rangle = \langle\psi^S|\sum_q \hat{K}_{b,q}^\dagger \hat{K}_{b,q}|\psi^S\rangle, \quad (9.29)$$

which is correctly normalised:

$$\sum_b \text{Prob}(b|\Psi^{SB}) = \langle\psi^S|\sum_{b,q} \hat{K}_{b,q}^\dagger \hat{K}_{b,q}|\psi^S\rangle = 1. \quad (9.30)$$

^aIndeed:

$$\begin{aligned} P_b &= \langle\Psi^{SB}|\mathbb{1}_S \otimes \hat{\Pi}_b^B|\Psi^{SB}\rangle = \langle\psi^S|\otimes\langle\phi^B|\hat{U}_{\text{tot}}^\dagger(\mathbb{1}_S \otimes \hat{\Pi}_b^B)\hat{U}_{\text{tot}}|\phi^B\rangle\otimes|\psi^S\rangle \\ &= \sum_q \langle\psi^S|\otimes\langle\phi^B|\hat{U}_{\text{tot}}^\dagger(\mathbb{1}_S \otimes |\phi_{b,q}^B\rangle\langle\phi_{b,q}^B|)\hat{U}_{\text{tot}}|\phi^B\rangle\otimes|\psi^S\rangle = \langle\psi^S|\sum_q \hat{K}_{b,q}^\dagger \hat{K}_{b,q}|\psi^S\rangle. \end{aligned}$$

As you see, the probability of measuring the eigenvalue b involves an operator

$$\hat{E}_b = \sum_q \hat{K}_{b,q}^\dagger \hat{K}_{b,q}. \quad (9.31)$$

Such an operator is Hermitian and *positive*, as we will later show explicitly. Moreover, completeness implies that:

$$\sum_b \hat{E}_b = \sum_{b,q} \hat{K}_{b,q}^\dagger \hat{K}_{b,q} = \mathbb{1}_S. \quad (9.32)$$

The probability can be expressed in terms of \hat{E}_b as

$$P_b = \text{Prob}(b|\hat{\rho}_{\text{in}}^S) = \text{Tr}_S(\hat{E}_b \hat{\rho}_{\text{in}}^S), \quad (9.33)$$

which, by linearity, holds also for an initial mixed state $\hat{\rho}_{\text{in}}^S$.

We now want to discuss how the state collapses after measuring b . Here, the result is slightly surprising.

◆

The collapse of a pure state does not lead to a pure state. In the standard von Neumann framework, a pure state $|\psi^S\rangle$ would collapse into a pure state. This would suggest here that the collapse of $|\psi^S\rangle$, upon measuring b for the pointer leads to a pure state

$$|\psi^S\rangle \xrightarrow{\text{measure } b} |\psi_b^S\rangle = \frac{\sum_q \hat{K}_{b,q}|\psi^S\rangle}{\|\sum_q \hat{K}_{b,q}|\psi^S\rangle\|}. \quad (9.34)$$

This is however **incorrect**, in the presence of a degeneracy of the eigenvalue b .

Indeed, the probability I would calculate from the square modulus of the projection amplitude of such a hypothetical pure state gives:

$$\|\sum_q \hat{K}_{b,q}|\psi^S\rangle\|^2 = \sum_{q,q'} \langle\psi^S|\hat{K}_{b,q'}^\dagger \hat{K}_{b,q}|\psi^S\rangle \neq P_b = \langle\psi^S|\sum_q \hat{K}_{b,q}^\dagger \hat{K}_{b,q}|\psi^S\rangle.$$

The reason for such a fact should be traced in the entangled nature of the state in Eq. (9.26), which, in the presence of degeneracy, does not lead to a well-defined collapsed pure state for the system.

The correct expression for the collapsed state is obtained in terms of density matrices as follows:

$$|\psi^S\rangle\langle\psi^S| \xrightarrow{\text{measure } b} \hat{\rho}_b^S \equiv \frac{1}{P_b} \sum_q \hat{K}_{b,q} |\psi^S\rangle\langle\psi^S| \hat{K}_{b,q}^\dagger, \quad (9.35)$$

where you observe that the right-hand side is *not a pure state*, in general.

❶

The correct density matrix after the collapse. By linearity, the previous expression generalises to a mixed initial state. Upon measuring the eigenvalue b of the pointer the state of the system collapses to:

$$\hat{\rho}_{\text{in}}^S \xrightarrow{\text{measure } b} \hat{\rho}_b^S \equiv \frac{1}{P_b} \sum_q \hat{K}_{b,q} \hat{\rho}_{\text{in}}^S \hat{K}_{b,q}^\dagger. \quad (9.36)$$

At this point, it is crucial to decide if you *keep track* of the outcome “ b ” of the measurement, or if you completely *disregard* it. If you keep track of the outcome, you are doing some sort of filtering of states, and you have to deal with $\hat{\rho}_b^S$. If you disregard b , you essentially combine the various $\hat{\rho}_b^S$, with their probability P_b , into a post-measurement ensemble, leading to a final density matrix $\hat{\rho}_{\text{p-m}}^S$.

❶

The post-measurement state. The post-measurement density matrix obtained by disregarding the outcome b of the measurement is given by a Kraus quantum map:

$$\hat{\rho}_{\text{in}}^S \xrightarrow{\text{disregard outcome}} \hat{\rho}_{\text{p-m}}^S = \sum_b P_b \hat{\rho}_b^S = \sum_{b,q} \hat{K}_{b,q} \hat{\rho}_{\text{in}}^S \hat{K}_{b,q}^\dagger. \quad (9.37)$$

Observe that the post-measurement state differs from the initial state because of the “pre-measurement” entangling interaction.

9.2.3. Ambiguity in the preparation of a post-measurement state

A projective measurement on the *pointer* leads to a Kraus map with Kraus operators that are generally *different* from system projectors. Before we adventure into the general discussion of how *different measurements* on the pointer — hence different preparation procedures — lead to the *same* post-measurement density matrix for the system, it is useful to revisit the cNOT gate example, by doing the following:

Exercise 9.2 (The cNOT gate revisited). Consider a Qbit (the system) interacting with a pointer (=environment) made of a single Qbit. The entangling interaction is the cNOT gate we have seen before, with the “control” bit being the system. If the system initial state is $|\psi^S\rangle = z_0|0\rangle_S + z_1|1\rangle_S$, and the initial state of the pointer is $|\phi^B\rangle = |0\rangle_B$, then the entangled state after the cNOT is:

$$|\Psi^{\text{SB}}\rangle = \hat{U}_{\text{cNOT}} |\psi^S\rangle \otimes |\phi^B\rangle = \hat{U}_{\text{cNOT}} \left(z_0|0\rangle_S + z_1|1\rangle_S \right) \otimes |0\rangle_B = z_0|0\rangle_S \otimes |0\rangle_B + z_1|1\rangle_S \otimes |1\rangle_B.$$

Imagine measuring the pointer spin $\hat{\sigma}^z$.

- 1) Calculate the probabilities P_0 and P_1 of obtaining $|0\rangle_B$ and $|1\rangle_B$.
- 2) Write the resulting collapsed system states in terms of a set of Kraus operators.
- 3) Express the final (i.e., after measurement) system density matrix $\hat{\rho}_{\text{p-m}}^S$.

Think now of measuring the pointer spin $\hat{\sigma}^x$.

- 4) Calculate the probabilities Q_+ and Q_- of obtaining $|+, \mathbf{x}\rangle_B$ and $|-, \mathbf{x}\rangle_B$.
- 5) Write the resulting collapsed system states in terms of a set of Kraus operators. Are the two states orthogonal? If not, when they would be orthogonal?
- 6) Express the final (i.e., after measurement) system density matrix $\hat{\rho}_{p-m}^S$. Is it different from that obtained at point 3) above?

We now discuss this fact in a more general setting. For simplicity, we consider the non-degenerate case, in which the extra quantum number q is not present. Then the collapse of a pure state $|\psi^S\rangle$, upon measuring a pointer operator \hat{B} with outcome one of its eigenvalues b , prepares a pure state:

$$|\psi^S\rangle \xrightarrow{\text{measure } b} |\psi_b^S\rangle = \frac{\hat{K}_b |\psi^S\rangle}{\|\hat{K}_b |\psi^S\rangle\|} \quad \text{where} \quad \hat{K}_b = \langle \phi_b^B | \hat{U}_{\text{tot}} | \phi^B \rangle. \quad (9.38)$$

Upon repeating the measurements, we have prepared an “ensemble” :

$$\left\{ P_b, |\psi_b^S\rangle \right\} \quad \text{with} \quad |\psi_b^S\rangle = \frac{\hat{K}_b |\psi^S\rangle}{\|\hat{K}_b |\psi^S\rangle\|} \quad \text{and} \quad P_b = \langle \psi^S | \hat{K}_b^\dagger \hat{K}_b | \psi^S \rangle \quad (9.39)$$

representing the system post-measurement density matrix:

$$\hat{\rho}_{p-m}^{S,B} = \sum_b P_b |\psi_b^S\rangle \langle \psi_b^S| = \sum_b \hat{K}_b |\psi^S\rangle \langle \psi^S| \hat{K}_b^\dagger. \quad (9.40)$$

Interestingly, system states associated to different eigenvalues b and b' are *non orthogonal*, i.e.,

$$\langle \psi_{b'}^S | \psi_b^S \rangle \neq \delta_{b,b'}.$$

One can show that the conditional probability of measuring “ b' ” *immediately after* having measured b is now:

$$\text{Prob}(b'|b \wedge \Psi^{SB}) = \frac{\|\hat{K}_{b'} \hat{K}_b |\psi\rangle_s\|^2}{\|\hat{K}_b |\psi\rangle_s\|^2}, \quad (9.41)$$

hence the two measurements agree, ending with $\text{Prob}(b'|b \wedge \Psi^{SB}) = \delta_{b',b}$, *only if* $\hat{K}_{b'} \hat{K}_b = \delta_{b',b} \hat{K}_b$, which means that the Kraus operators are orthogonal projectors.

Imagine that we decide to measure a different observable \hat{C} , associated to a *different* orthonormal basis $\{|\phi_c^B\rangle\}$ of the bath Hilbert space \mathcal{H}_B . In the concrete example of the pointer/environment being a single spin, the operator \hat{C} might be the spin of the pointer in a different direction \mathbf{n}' . The collapse of a pure state $|\psi^S\rangle$, upon measuring c , prepares a pure state:

$$|\psi^S\rangle \xrightarrow{\text{measure } c} |\chi_c^S\rangle = \frac{\hat{M}_c |\psi^S\rangle}{\|\hat{M}_c |\psi^S\rangle\|} \quad \text{where} \quad \hat{M}_c = \langle \phi_c^B | \hat{U}_{\text{tot}} | \phi^B \rangle, \quad (9.42)$$

with a probability

$$Q_c = \text{Prob}(c|\Psi^{SB}) = \langle \Psi^{SB} | \mathbb{1}_S \otimes \hat{\Pi}_c^B | \Psi^{SB} \rangle = \langle \psi^S | \hat{M}_c^\dagger \hat{M}_c | \psi^S \rangle. \quad (9.43)$$

Such a measurement produces a *different preparation ensemble*

$$\left\{ Q_c, |\chi_c^S\rangle \right\} \quad \text{with} \quad |\chi_c^S\rangle = \frac{\hat{M}_c |\psi^S\rangle}{\|\hat{M}_c |\psi^S\rangle\|}, \quad (9.44)$$

for the system post-measurement state,

$$\hat{\rho}_{p-m}^{S,C} = \sum_c Q_c |\chi_c^S\rangle \langle \chi_c^S| = \hat{M}_c |\psi^S\rangle \langle \psi^S| \hat{M}_c^\dagger. \quad (9.45)$$

1 **Different preparations, same state.** The remarkable fact is that the two post-measurement states are indeed *identical*:

$$\hat{\rho}_{p-m}^{S,B} = \hat{\rho}_{p-m}^{S,C} = \hat{\rho}_{p-m}^S . \quad (9.46)$$

To convince yourself that this is indeed the case, ⁵ consider that the two different orthonormal basis sets $\{|\phi_b^B\rangle\}$ and $\{|\phi_c^C\rangle\}$ for \hat{B} and \hat{C} , respectively, must be related by a unitary transformation \hat{U}_B :

$$|\phi_c^C\rangle = \sum_b (\hat{U}_B)_{cb} |\phi_b^B\rangle . \quad (9.47)$$

Then, you conclude that:

$$\hat{M}_c = \langle \phi_c^B | \hat{U}_{\text{tot}} | \phi^B \rangle = \sum_b (\hat{U}_B)_{cb}^* \langle \phi_b^B | \hat{U}_{\text{tot}} | \phi^B \rangle = \sum_b (\hat{U}_B)_{cb}^* \hat{K}_b ,$$

and, as a consequence: ⁶

$$\hat{\rho}_{p-m}^{S,C} = \sum_c \sum_{b,b'} (\hat{U}_B)_{cb}^* (\hat{U}_B)_{cb'} \hat{K}_b |\psi^S\rangle \langle \psi^S| \hat{K}_{b'}^\dagger = \sum_b \hat{K}_b |\psi^S\rangle \langle \psi^S| \hat{K}_b^\dagger \equiv \hat{\rho}_{p-m}^{S,B} . \quad (9.48)$$

9.2.4. The von Neumann protocol

The von Neumann postulate — concerning projective measurements — should not be confused with the von Neumann *protocol*, which is an interesting example of a generalised measurement, which partially clarifies the mechanism behind the collapse of the state. The protocol is best illustrated with the example of a Stern-Gerlach apparatus “measuring” the electronic spin of atoms, like Ag in the original experiment, passing through the apparatus. ⁷ See Fig. 8.1.

An interesting aspect of the story is that the natural role of a pointer variable is here played by the *center-of-mass* wave-function of the very same atom whose spin we are trying to measure: the center-of-mass of the atom is indeed subject to a different force due to the magnetic field gradient, depending on the *spin state* of the atom. With a slightly simplified notation, let me denote by $\psi(\mathbf{x})$ the orbital part of the atomic wave-function, \mathbf{x} being the center-of-mass coordinate, and by $|\psi^S\rangle$ the *spin state* of the atoms, which we assume to be $S = 1/2$. Let the initial state entering the SG apparatus be $|\Psi_{\text{in}}\rangle = \psi_0(\mathbf{x}) \otimes |\psi^S\rangle$. ⁸ The atom then passes through the two specially designed magnets, where a magnetic field gradient — assumed along the axis of the apparatus, which we denote by \mathbf{z} — provokes a *force* along the \mathbf{z} -direction on the center-of-mass:

$$F_z(\mathbf{x}) = -\mu_B \frac{\partial B_z}{\partial z}(\mathbf{x}) \otimes \hat{\sigma}^z , \quad (9.49)$$

formally derived from a Zeemann Hamiltonian term $\mu_B B_z(\mathbf{x}) \otimes \hat{\sigma}^z$, where $\mu_B = e\hbar/2mc$ is the Bohr magneton.

⁵You should observe the complete similarity of this discussion with that given a while ago when discussing the ambiguity of the Kraus map and of the purification of a state.

⁶Use that:

$$\sum_c (\hat{U}_B)_{cb}^* (\hat{U}_B)_{cb'} = \sum_c (\hat{U}_B^\dagger)_{bc} (\hat{U}_B)_{cb'} = (\hat{U}_B^\dagger \hat{U}_B)_{b,b'} = \delta_{b,b'} .$$

⁷A similar illustration would be given by a beam of linearly polarised photons passing through a thick calcite crystal.

⁸Assume that the spin and the orbital part of the electronic wave-function are not entangled: essentially, neglect any spin-orbit effects. We further assume that the spin-state has been prepared to be a *pure state* $|\psi^S\rangle$.

i

The force operator. Notice that, strictly speaking, this is a *force operator*, acting on the combined system — center-of-mass (in principle the gradient of the field depends on its position \mathbf{x}) and spin — and entangling these two parts of the state. Nevertheless, thinking classically — the atom is, after all, quite heavy and classical mechanics applies as a good approximation — we can go on describing the “dynamics of the center-of-mass” in a classical framework.

The complicated quantum dynamics of the center-of-mass – with a time-dependent interaction, because the system feels the effect of the magnetic field gradient only when inside the poles of the magnet, while essentially free motion follows outside the magnet — boils down to a unitary *pre-measurement* operator of the form:

$$\hat{U}_{\text{tot}} = \exp\left(-\frac{i}{\hbar}a\hat{P}_z \otimes \hat{\sigma}^z\right) \quad (9.50)$$

where a is the deflection along the z -direction of the center of mass, clearly dependent on the total “pre-measurement” time t , and \hat{P}_z the z -component of the center-of-mass momentum operator. Notice that a unitary operator of this type implies that the initial state is transformed as:

$$|\Psi_{\text{in}}\rangle = \psi_0(\mathbf{x}) \otimes (z_+|\uparrow\rangle + z_-|\downarrow\rangle) \xrightarrow{\hat{U}_{\text{tot}}} |\Psi_{\text{f}}\rangle = z_+\psi_+(\mathbf{x}) \otimes |\uparrow\rangle + z_-\psi_-(\mathbf{x}) \otimes |\downarrow\rangle \quad (9.51)$$

where $\psi_{\pm}(\mathbf{x}) = \psi_0(\mathbf{x} - \mathbf{d} \mp a\hat{\mathbf{z}})$, and \mathbf{d} denotes the position of the detector where the atom would have ended in the absence of the magnet. Also observe that, by doing so, I have hidden a piece of “free-evolution-operator”, which I have not included in \hat{U}_{tot} , and neglected, as well, any spreading of the wave-function ψ_0 .

Technically, such *pre-measurement evolution* has led to a *pure state* $|\Psi_{\text{f}}\rangle$ which shows, however, *entanglement* between the spin, which I was willing to measure, and the “position of the atom center-of-mass”. I have deliberately neglected any possible source of decoherence that might have occurred while the atom goes through the magnet and arrives at the detector. But now, some form of further interaction with the detector must occur, so that I can “read out” the result of the measurement: is the atom deviated upwards, or downwards?

In absence of such a “read-out” interaction, I might still dream of “joining together” the two entangled-superposed components of the total state $|\Psi_{\text{f}}\rangle$ without observing the system — remember, no form of *which-way detection* must occur — and getting again a final un-entangled pure state: a kind of “*undoing*” of the pre-measurement unitary evolution, which is, in principle at least, invertible.

But, if a form of “which-way detection” occurs ⁹ the phase coherence of the two pieces of the state $|\Psi_{\text{f}}\rangle$ is irretrievably lost. Incidentally, this is a “macroscopic measurement” — a long magnet which can provide enough deflection to the atoms that I can unambiguously say if the atom has deviated upwards or downwards: The atom is revealed in *either one of the two paths*, never in a superposition. ¹⁰ The amount of information lost in the state can be gauged by ignoring the center-of-mass degree of freedom and calculating the partial-trace of the pre-measured state:

$$\hat{\rho}^{\text{S}} = \text{Tr}_{\text{orb}}(|\Psi_{\text{f}}\rangle\langle\Psi_{\text{f}}|) = \begin{pmatrix} |z_+|^2 & \langle\psi_-|\psi_+\rangle z_+ z_-^* \\ \langle\psi_+|\psi_-\rangle z_+^* z_- & |z_-|^2 \end{pmatrix}. \quad (9.52)$$

⁹I need not necessarily “demolish” the state by having the atoms hitting a detector screen: the detection might involve some more clever apparatus which can perform a “which-way” detection.

¹⁰This is at variance with having a microscopic pointer, like a single spin-1/2, where you could think of measuring the pointer spin also along some other direction, for instance, $\hat{\mathbf{x}}$.

1 **The role of the overlap.** Quite interestingly, the more the measurement is “macroscopic” and the two components are spatially separated, the smaller the overlap $\langle \psi_- | \psi_+ \rangle$, implying, in essence, a mixed final state where the coherence is nearly lost when the overlap is exponentially small.

9.2.5. POVM and summary of quantum measurement

Given the Kraus operators \widehat{K}_k , we can form positive Hermitian operators $\widehat{E}_k = \widehat{K}_k^\dagger \widehat{K}_k$ which have the properties:

Hermiticity) $\widehat{E}_k = \widehat{K}_k^\dagger \widehat{K}_k$ are Hermitian.

Positivity) $\widehat{E}_k = \widehat{K}_k^\dagger \widehat{K}_k$ are positive (really, non-negative), since $\forall |\psi\rangle$:

$$\langle \psi | \widehat{E}_k | \psi \rangle = \|\widehat{K}_k |\psi\rangle\|^2 \geq 0. \quad (9.53)$$

Completeness) They realise a resolution of the identity:

$$\sum_k \widehat{E}_k = \sum_k \widehat{K}_k^\dagger \widehat{K}_k = \mathbb{1}_S. \quad (9.54)$$

A system of operators with these properties is known as *positive operator-values measure*, or POVM.



Warning: In general, the probability of measurements are given in terms of the $\{\widehat{E}_k\}$, while the “state after the measurement” is known only if the $\{\widehat{K}_k\}$ are given. If I give you a set of Kraus operators $\{\widehat{K}_k\}$, you can easily construct a POVM set $\{\widehat{E}_k\}$. The opposite process involves in general a polar decomposition. If you write $\widehat{K}_k = \widehat{U}_k (\widehat{E}_k)^{\frac{1}{2}}$, where \widehat{U}_k is an *arbitrary* unitary, you have $\widehat{E}_k = \widehat{K}_k^\dagger \widehat{K}_k$. See Preskill’s [lecture notes](#) 3.1.2. If you are not interested in the “state after the measurement” (for instance, because the measurement involves demolition of the system, like when you do a photon detection), then the POVM set $\{\widehat{E}_k\}$ is enough.

In the pointer measurement framework discussed previously, we would write, given the set of Kraus operators $\widehat{K}_{b,q}$:

$$\widehat{E}_b = \sum_q \widehat{K}_{b,q}^\dagger \widehat{K}_{b,q}, \quad (9.55)$$

which is a positive Hermitian, and satisfies the completeness relation:

$$\sum_b \widehat{E}_b = \sum_{b,q} \widehat{K}_{b,q}^\dagger \widehat{K}_{b,q} = \mathbb{1}_S. \quad (9.56)$$

The probability of measuring b is given by

$$P_b = \text{Prob}(b | \hat{\rho}_{\text{in}}^S) = \text{Tr}_S(\widehat{E}_b \hat{\rho}_{\text{in}}^S). \quad (9.57)$$

The collapsed state is given by:

$$\hat{\rho}_{\text{in}}^S \xrightarrow{\text{measure } b} \hat{\rho}_b^S \equiv \frac{1}{P_b} \sum_q \widehat{K}_{b,q} \hat{\rho}_{\text{in}}^S \widehat{K}_{b,q}^\dagger. \quad (9.58)$$

The post-measurement density matrix is:

$$\hat{\rho}_{p-m}^S = \mathcal{E}(\hat{\rho}_{\text{in}}^S) = \sum_b P_b \hat{\rho}_b^S = \sum_{b,q} \widehat{K}_{b,q} \hat{\rho}_{\text{in}}^S \widehat{K}_{b,q}^\dagger. \quad (9.59)$$

i

The projective measurement case. The von Neumann case is obtained when $\widehat{K}_{b,q} \rightarrow \widehat{\Pi}_b$, so that $\widehat{E}_b = \widehat{\Pi}_b$ as well.

9.3. Inverting Kraus: how to “invent” unitaries

So far, we have shown that

$$\begin{aligned}\hat{\rho}_s(t) &= \text{Tr}_B \left(\widehat{U}_{\text{tot}}(t) |\psi^S(0)\rangle \langle \psi^S(0)| \otimes |\phi^B(0)\rangle \langle \phi^B(0)| \widehat{U}_{\text{tot}}^\dagger(t) \right) \\ &= \sum_k \widehat{K}_k(t) \hat{\rho}_s(0) \widehat{K}_k^\dagger(t).\end{aligned}\quad (9.60)$$

The first representation of the dynamics is the standard *unitary representation*: it is constructed out of $|\psi^S(0)\rangle$, $|\phi^B(0)\rangle$ and an explicit unitary evolution operator $\widehat{U}_{\text{tot}}(t)$. Out of that, by choosing a basis for the bath we can always deduce, in principle, the corresponding *Kraus representation*, given in the second line, as we have already shown. Now we consider the reverse problem:

Question:

If you are given a Kraus map representation for the system density matrix dynamics:

$$\hat{\rho}_s(t) = \sum_k^{\text{DK}} \widehat{K}_k(t) \hat{\rho}_s(0) \widehat{K}_k^\dagger(t) \quad (9.61)$$

with the Kraus operators satisfying the completeness in Eq. (9.7), can you reconstruct back a unitary representation of some sort?

It is perhaps not surprising that the answer is yes. Indeed, a moment of reflection will convince you that upon having the system density matrix you have lost an immense amount of information on the environment, hence reconstructing back a unitary representation involves a very large arbitrariness. So, the proper answer is: yes and in an **infinite number of ways!**

To show this, we consider as usual the case of $\hat{\rho}_s(0) = |\psi^S\rangle \langle \psi^S|$: by linearity, you can extend it to an arbitrary mixed state for $\hat{\rho}_s(0)$. The first observation is that the operators \widehat{K}_k (we omit the fixed time t from now on) *know nothing about the environment or bath*. We can *invent* an arbitrary \mathcal{H}_B with a basis $\{|\phi_k^B\rangle\}$ — with the proviso that $\dim(\mathcal{H}_B) \geq D_K$, where D_K is the number of terms of the Kraus representation —, and define an *isometry* $\widehat{V} : \mathcal{H}_S \rightarrow \mathcal{H}_S \otimes \mathcal{H}_B$ — by definition, a linear map that conserves the scalar product — in the following way:

$$\widehat{V} |\psi^S\rangle \stackrel{\text{def}}{=} \sum_k \left(\widehat{K}_k |\psi^S\rangle \right) \otimes |\phi_k^B\rangle. \quad (9.62)$$

The first thing that we need to check is that \widehat{V} is indeed an isometry. We take two input states $|\psi_1^S\rangle$ and $|\psi_2^S\rangle$, and calculate:

$$\begin{aligned}\langle \psi_2^S | \widehat{V}^\dagger \widehat{V} | \psi_1^S \rangle &= \sum_{k',k} \langle \phi_{k'}^B | \otimes \langle \psi_2^S | \widehat{K}_{k'}^\dagger \widehat{K}_k | \psi_1^S \rangle \otimes |\phi_k^B\rangle \\ &= \sum_k \langle \psi_2^S | \widehat{K}_k^\dagger \widehat{K}_k | \psi_1^S \rangle = \langle \psi_2^S | \psi_1^S \rangle,\end{aligned}\quad (9.63)$$

where we used $\langle \phi_{k'}^{\text{B}} | \phi_k^{\text{B}} \rangle = \delta_{k',k}$. So, \hat{V} is an isometry. The second thing to check is that tracing over \mathcal{H}_{B} we get the correct density matrix:

$$\text{Tr}_{\text{B}} \left(\hat{V} |\psi^{\text{S}}\rangle \langle \psi^{\text{S}}| \hat{V}^\dagger \right) = \sum_k \hat{\text{K}}_k |\psi^{\text{S}}\rangle \langle \psi^{\text{S}}| \hat{\text{K}}_k^\dagger = \hat{\rho}_{\text{S}}. \quad (9.64)$$

This follows easily from the construction of \hat{V} .

The final step is that an isometry can be *extended into a unitary* \hat{U}_{tot} over the product space $\mathcal{H}_{\text{S}} \otimes \mathcal{H}_{\text{B}}$. And indeed this can be done in an infinite number of ways!

❶

Unitary extension of isometries. Let $\hat{V} : \mathcal{W} \rightarrow \mathcal{H}$ be an isometry defined on an m -dimensional subspace $\mathcal{W} \subset \mathcal{H}$ of an n -dimensional Hilbert space \mathcal{H} . Then you can extend it in many ways as a unitary $\hat{U} : \mathcal{H} \rightarrow \mathcal{H}$ such that $\hat{U}\psi = \hat{V}\psi$ if $\psi \in \mathcal{W}$. Here is the (simple) idea of the proof. Let $\{w_1, \dots, w_m\}$ be an orthonormal basis of \mathcal{W} . You can extend it in infinitely many ways to a full orthonormal basis for \mathcal{H} : $\{w_1, \dots, w_m, w_{m+1}, \dots, w_n\}$. Since \hat{V} is an isometry, the m vectors $\psi_j = \hat{V}w_j$ for $j = 1 \dots m$ are orthonormal in the image subspace $\text{Range}(\hat{V}) \subset \mathcal{H}$. Extend such a basis (in an arbitrary way) to an orthonormal basis of \mathcal{H} : $\{\psi_1, \dots, \psi_m, \psi_{m+1}, \dots, \psi_n\}$. Now define \hat{U} as follows:

$$\hat{U} w_j = \psi_j \quad \text{for } j = 1 \dots n. \quad (9.65)$$

Clearly \hat{U} is unitary because it maps an orthonormal basis of \mathcal{H} into an orthonormal basis of \mathcal{H} . For $j = 1 \dots m$ we have $\hat{U}w_j = \psi_j = \hat{V}w_j$, hence \hat{U} coincides with \hat{V} on \mathcal{W} .

❷

Stinespring dilation. The process we have just outlined is related to what is known as *Stinespring dilation*. Incidentally, we might have defined the Stinespring isometry as $\hat{V} : \mathcal{H}_{\text{S}} \otimes \mathcal{H}_{\text{B}} \rightarrow \mathcal{H}_{\text{S}} \otimes \mathcal{H}_{\text{B}}$ with:

$$\hat{V} |\psi^{\text{S}}\rangle \otimes |\phi_1^{\text{B}}\rangle \stackrel{\text{def}}{=} \sum_k \left(\hat{\text{K}}_k |\psi^{\text{S}}\rangle \right) \otimes |\phi_k^{\text{B}}\rangle, \quad (9.66)$$

where $|\phi_1^{\text{B}}\rangle$ is an arbitrary element of the basis of \mathcal{H}_{S} . As you notice, \hat{V} is still defined on a subspace $\mathcal{W} \subset \mathcal{H}_{\text{S}} \otimes \mathcal{H}_{\text{B}}$, and hence can/need to be extended in many ways to a full unitary on $\mathcal{H}_{\text{S}} \otimes \mathcal{H}_{\text{B}}$.

⚠

Warning: The large arbitrariness implied by the previous construction implies in turn that most of the constructed unitary representations have no physical meaning! We will see this fact later on when discussing some useful quantum maps on a single Qbit.

9.4. Quantum maps

We have seen that the evolution operator for density matrices can be written as:

$$\mathcal{E}_{(t,0)}(\hat{\rho}_{\text{S}}(0)) = \hat{\rho}_{\text{S}}(t) = \sum_k^{\text{D}_{\text{K}}} \hat{\text{K}}_k(t) \hat{\rho}_{\text{S}}(0) \hat{\text{K}}_k^\dagger(t) \quad \text{with} \quad \sum_k^{\text{D}_{\text{K}}} \hat{\text{K}}_k^\dagger \hat{\text{K}}_k = \mathbb{1}_{\text{S}}. \quad (9.67)$$

This is called *Kraus quantum map*, or *quantum channel*. It is a particular *super-operator*, in the sense that maps an operator on the Hilbert space of the system into another operator.

i

Quantum map. In principle, we can define a *quantum map* in a more general setting, as any super-operator acting on system operators \widehat{O} , i.e., $\widehat{O} \rightarrow \mathcal{E}(\widehat{O})$ which verifies the following conditions:

1: Linear) $\mathcal{E}(\widehat{O}_1 + \widehat{O}_2) = \mathcal{E}(\widehat{O}_1) + \mathcal{E}(\widehat{O}_2)$.

2: Trace Preserving) $\text{Tr}_S(\mathcal{E}(\widehat{O})) = \text{Tr}_S(\widehat{O})$.

3: Positive) If $\widehat{O} \geq 0$, then $\mathcal{E}(\widehat{O}) \geq 0$.

3*: C-Positive) If an arbitrary ancillary systems \mathcal{H}_A is interacting with the system \mathcal{H}_S and we consider a positive operator $\widehat{O}_{AS} \geq 0$ acting on $\mathcal{H}_A \otimes \mathcal{H}_S$, then the trivially extended quantum map $\mathbb{1}_A \otimes \mathcal{E}$ is still positive:

$$\widehat{O}_{AS} \geq 0 \quad \Longrightarrow \quad (\mathbb{1}_A \otimes \mathcal{E})(\widehat{O}_{AS}) \geq 0. \quad (9.68)$$

i

Complete positivity. Property 3*) is known as *complete positivity* or CP for short. It obviously implies the positivity in 3) — simply apply it to any factorised positive operator $\widehat{O}_A \otimes \widehat{O}_S$, and you deduce 3) from 3*) —, but is indeed a *stronger* requirement, as shown by the example of partial transpose below. The reason why we insist on a map being CP is the following. Imagine that the system (and only the system) interacts with an environment \mathcal{H}_B , and that the quantum map \mathcal{E} emerges from this interaction. Then, an ancillary system \mathcal{H}_A is made to interact with the system, and *only the system*. The global quantum map that accounts for the effect of the environment must now be $\mathbb{1}_A \otimes \mathcal{E}$, and this must still transform positive operators of $\mathcal{H}_A \otimes \mathcal{H}_S$ into positive operators, as the CP condition requires.

Transpose map. An interesting example of a map which is clearly positive but, as shown in the example below, *not completely positive*, is the transpose map. Take an arbitrary density matrix, written in an arbitrary basis, $\{|\phi_j\rangle\}$, on which it is associated to a matrix $[\hat{\rho}]_{j',j} = \langle \phi_{j'} | \hat{\rho} | \phi_j \rangle$, and consider the following quantum map:

$$\hat{\rho} = \sum_{j,j'} [\hat{\rho}]_{j',j} |\phi_{j'}\rangle \langle \phi_j| \longrightarrow \mathcal{E}_T(\hat{\rho}) = \sum_{j,j'} [\hat{\rho}]_{j,j'}^T |\phi_{j'}\rangle \langle \phi_j|, \quad (9.69)$$

which maps the matrix $[\hat{\rho}]$ associated to $\hat{\rho}$ into its transpose $[\hat{\rho}]^T$, hence into a legitimate density matrix: Hermitian, positive, trace-1. But it does not define a completely positive quantum map, as the following 2-Qbit example shows.

Example: Partial transpose for 2-Qbits. Consider the transpose map \mathcal{E}_T for a system made of a single Qbit. It is simple to verify that, when acting on a general pure state $|\psi^S\rangle = z_0|0\rangle + z_1|1\rangle$, this map acts as:

$$\mathcal{E}_T(|\psi^S\rangle \langle \psi^S|) = |\bar{\psi}^S\rangle \langle \bar{\psi}^S| \quad \text{with} \quad |\bar{\psi}^S\rangle = z_0^*|0\rangle + z_1^*|1\rangle.$$

Take now an ancillary Qbit and consider the entangled (Bell) state $|\psi^{AS}\rangle = \frac{1}{\sqrt{2}}(|00\rangle_{AS} + |11\rangle_{AS})$ which you can view as a legitimate positive (entangled) state:

$$\widehat{O}_{AS} = |\psi^{AS}\rangle \langle \psi^{AS}| = \frac{1}{2} (|00\rangle \langle 00| + |00\rangle \langle 11| + |11\rangle \langle 00| + |11\rangle \langle 11|).$$

Upon applying the partial transpose $\mathbb{1}_A \otimes \mathcal{E}_T$ we get:

$$(\mathbb{1}_A \otimes \mathcal{E}_T)(\widehat{O}_{AS}) = \frac{1}{2} (|00\rangle \langle 00| + |10\rangle \langle 01| + |01\rangle \langle 10| + |11\rangle \langle 11|).$$

When represented in the standard product basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ the matrix associated to such an operator is:

$$\left(\mathbb{1}_A \otimes \mathcal{E}_T\right)(\widehat{O}_{AS}) \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (9.70)$$

and you recognize that it possesses an eigenvalue -1 originated by the $\hat{\sigma}^x$ block in the center. Hence \mathcal{E}_T is positive but not CP: it is therefore not a valid (physical) quantum map.

Evidently, a Kraus map

$$\hat{\rho}_S \rightarrow \mathcal{E}(\hat{\rho}_S) = \sum_k^{D_K} \widehat{K}_k \hat{\rho}_S \widehat{K}_k^\dagger \quad \text{with} \quad \sum_k^{D_K} \widehat{K}_k^\dagger \widehat{K}_k = \mathbb{1}_S$$

verifies all 4 conditions of a quantum map. We have already shown that a Kraus map (which is linear) is trace-preserving and positive. To prove that it is CP, simply observe that $\mathcal{H}_A \otimes \mathcal{H}_S$ can be seen as an enlarged system to which the original Kraus operator can be extended, by acting as an identity on the ancillary space. Hence, the proof of positivity we have made automatically implies CP.

Interestingly, one can also prove (but we will not do that) the reverse result:

Theorem 9.1. If a quantum map \mathcal{E} is linear, trace-preserving and CP — i.e., it satisfies properties 1), 2) and 3*) — then there exists a Kraus representation with $D_K = d_S^2$.

9.5. Ambiguity of the Kraus representation and purification

Consider a set of D_K Kraus operators \widehat{K}_k , and a second set of D_M Kraus operators \widehat{M}_m . If, for instance, $D_M > D_K$ you can extend the \widehat{K}_k to a larger set by adding zeros (seen as trivial operators) to the set: $\{\widehat{K}_1, \dots, \widehat{K}_{D_K}, 0, \dots, 0\}$.

It is simple to verify that if these two sets are unitarily related, in the sense that a $D_M \times D_M$ unitary matrix \mathcal{U}_{mk} exists such that:

$$\widehat{M}_m = \sum_k \mathcal{U}_{mk} \widehat{K}_k, \quad (9.71)$$

then they generate the same Kraus map.

Proof. It is easy to show that ¹¹

$$\begin{cases} \sum_{m=1}^{D_M} \widehat{M}_m^\dagger \widehat{M}_m = \sum_{k=1}^{D_K} \widehat{K}_k^\dagger \widehat{K}_k = \mathbb{1}_S \\ \sum_{m=1}^{D_M} \widehat{M}_m \hat{\rho}_S(0) \widehat{M}_m^\dagger = \sum_{k=1}^{D_K} \widehat{K}_k \hat{\rho}_S(0) \widehat{K}_k^\dagger = \hat{\rho}_S(t) \end{cases}. \quad (9.73)$$

■

In the following, we will prove the reverse of this statement, known as **Hughston-Jozsa-Wootters (HJW) theorem**, which essentially states that if two different sets of Kraus operators provide the

¹¹To prove the second, for instance:

$$\sum_m \widehat{M}_m \hat{\rho}_S(0) \widehat{M}_m^\dagger = \sum_{k,k'} \underbrace{\left(\sum_m \mathcal{U}_{mk'}^* \mathcal{U}_{mk} \right)}_{(\mathcal{U}^\dagger \mathcal{U})_{k'k} = \delta_{k',k}} \widehat{K}_k \hat{\rho}_S(0) \widehat{K}_{k'}^\dagger = \sum_k \widehat{K}_k \hat{\rho}_S(0) \widehat{K}_k^\dagger. \quad (9.72)$$

same Kraus map, then they must be unitarily related. In the process of proving such a theorem, we will make a digression on the issue of purification, which shows a nice application of the Schmidt decomposition.

We rewrite the Kraus map

$$\hat{\rho}_S = \sum_{k=1}^{D_K} \hat{K}_k \hat{\rho}_S(0) \hat{K}_k^\dagger,$$

for the case of a pure initial state $\hat{\rho}_S(0) = |\psi^S\rangle\langle\psi^S|$, by defining

$$|\psi_k^S\rangle = \frac{\hat{K}_k |\psi^S\rangle}{\|\hat{K}_k |\psi^S\rangle\|} \quad \Longrightarrow \quad \hat{\rho}_S = \sum_{k=1}^{D_K} P_k |\psi_k^S\rangle\langle\psi_k^S|, \quad (9.74)$$

with $P_k = \|\hat{K}_k |\psi^S\rangle\|^2 = \langle\psi^S|\hat{K}_k^\dagger\hat{K}_k|\psi^S\rangle$. This shows that we have expressed $\hat{\rho}_S$ as an ensemble preparation $E_K = \{P_k, |\psi_k^S\rangle\}$ of *generally non-orthogonal* states. But it is very simple to verify that $\hat{\rho}_S$ can be purified as follows:

$$\hat{\rho}_S = \text{Tr}_B |\Psi_1^{SB}\rangle\langle\Psi_1^{SB}| \quad \text{with} \quad |\Psi_1^{SB}\rangle = \sum_{k=1}^{D_K} \sqrt{P_k} |\psi_k^S\rangle \otimes |\phi_k^{(1,B)}\rangle \quad (9.75)$$

where $\{|\phi_k^{(1,B)}\rangle\}$ is an orthonormal basis for \mathcal{H}_B .

We now repeat the same procedure with the second, equivalent, form of the Kraus map

$$\hat{\rho}_S = \sum_{m=1}^{D_M} \hat{M}_m \hat{\rho}_S(0) \hat{M}_m^\dagger$$

obtaining:

$$|\chi_m^S\rangle = \frac{\hat{M}_m |\psi^S\rangle}{\|\hat{M}_m |\psi^S\rangle\|} \quad \Longrightarrow \quad \hat{\rho}_S = \sum_m Q_m |\chi_m^S\rangle\langle\chi_m^S|, \quad (9.76)$$

with $Q_m = \|\hat{M}_m |\psi^S\rangle\|^2 = \langle\psi^S|\hat{M}_m^\dagger\hat{M}_m|\psi^S\rangle$. This is, evidently, a second equivalent ensemble preparation $E_M = \{Q_m, |\chi_m^S\rangle\}$ with *different* non-orthogonal states of the *same* density matrix $\hat{\rho}_S$, which can, again, be purified as:

$$\hat{\rho}_S = \text{Tr}_B |\Psi_2^{SB}\rangle\langle\Psi_2^{SB}| \quad \text{with} \quad |\Psi_2^{SB}\rangle = \sum_m \sqrt{Q_m} |\chi_m^S\rangle \otimes |\phi_m^{(2,B)}\rangle \quad (9.77)$$

where $\{|\phi_m^{(2,B)}\rangle\}$ is another orthogonal basis for \mathcal{H}_B .

Question:

How are the two different purifications $|\Psi_2^{SB}\rangle$ and $|\Psi_1^{SB}\rangle$ related?

The answer requires the following lemma:

Theorem 9.2. If $|\Psi_1^{SB}\rangle$ and $|\Psi_2^{SB}\rangle$ are two different purifications of the same state $\hat{\rho}_S$, then a unitary \hat{U}_B exists such that

$$|\Psi_1^{SB}\rangle = (\mathbb{1}_S \otimes \hat{U}_B) |\Psi_2^{SB}\rangle \quad (9.78)$$

Proof. We write the Schmidt decomposition of $|\Psi_1^{\text{SB}}\rangle$ and $|\Psi_2^{\text{SB}}\rangle$ as:

$$\begin{aligned} |\Psi_1^{\text{SB}}\rangle &= \sum_{k=1}^{N_S} \sqrt{\lambda_k} |\Lambda_k^{\text{S}}\rangle \otimes |\widehat{\Lambda}_k^{(1,\text{B})}\rangle \\ |\Psi_2^{\text{SB}}\rangle &= \sum_{k=1}^{N_S} \sqrt{\lambda_k} |\Lambda_k^{\text{S}}\rangle \otimes |\widehat{\Lambda}_k^{(2,\text{B})}\rangle \end{aligned} \quad (9.79)$$

where $|\widehat{\Lambda}_k^{(1,\text{B})}\rangle$ and $|\widehat{\Lambda}_k^{(2,\text{B})}\rangle$ are different orthonormal basis of \mathcal{H}_B , and λ_k are the eigenvalues of $\hat{\rho}_\text{S}$ with orthonormal eigenvectors $|\Lambda_k^{\text{S}}\rangle$, i.e., $\hat{\rho}_\text{S} = \sum_{k=1}^{N_S} \lambda_k |\Lambda_k^{\text{S}}\rangle \langle \Lambda_k^{\text{S}}|$. The two different orthonormal basis of \mathcal{H}_B must be related by a unitary matrix \widehat{U}_B such that:

$$|\widehat{\Lambda}_k^{(1,\text{B})}\rangle = \widehat{U}_\text{B} |\widehat{\Lambda}_k^{(2,\text{B})}\rangle. \quad (9.80)$$

With the same matrix we evidently have $|\Psi_1^{\text{SB}}\rangle = (\mathbb{1}_\text{S} \otimes \widehat{U}_\text{B}) |\Psi_2^{\text{SB}}\rangle$. \blacksquare

We are now ready to prove the following **Hughston-Jozsa-Wootters (HJW) Theorem**:

Theorem 9.3. Given two sets of Kraus operators that realise the same quantum map:

$$\hat{\rho}_\text{S} = \sum_{k=1}^{D_\text{K}} \widehat{K}_k \hat{\rho}_\text{S}(0) \widehat{K}_k^\dagger = \sum_{m=1}^{D_\text{M}} \widehat{M}_m \hat{\rho}_\text{S}(0) \widehat{M}_m^\dagger \quad (9.81)$$

there exist a unitary transformation \mathbb{U} between the two sets of Kraus operators:

$$\widehat{K}_k = \sum_m \mathbb{U}_{km} \widehat{M}_m. \quad (9.82)$$

Proof. Consider $\hat{\rho}_\text{S} = \sum_k \widehat{K}_k \hat{\rho}_\text{S}(0) \widehat{K}_k^\dagger$, and assume an initial pure state $\hat{\rho}_\text{S}(0) = |\psi^{\text{S}}\rangle \langle \psi^{\text{S}}|$. We *purify* the evolved state as before, but we now write it:

$$\hat{\rho}_\text{S} = \text{Tr}_\text{B} \left(|\Psi_1^{\text{SB}}\rangle \langle \Psi_1^{\text{SB}}| \right) \quad \text{with} \quad |\Psi_1^{\text{SB}}\rangle = \sum_k \widehat{K}_k |\psi^{\text{S}}\rangle \otimes |\phi_k^{(1,\text{B})}\rangle,$$

The second Kraus representation allows us to construct the second *different* purification:

$$\hat{\rho}_\text{S} = \text{Tr}_\text{B} \left(|\Psi_2^{\text{SB}}\rangle \langle \Psi_2^{\text{SB}}| \right) \quad \text{with} \quad |\Psi_2^{\text{SB}}\rangle = \sum_m \widehat{M}_m |\psi^{\text{S}}\rangle \otimes |\phi_m^{(2,\text{B})}\rangle.$$

As proved before via the Schmidt decomposition, a \widehat{U}_B exists such that $|\Psi_1^{\text{SB}}\rangle = (\mathbb{1}_\text{S} \otimes \widehat{U}_\text{B}) |\Psi_2^{\text{SB}}\rangle$. Hence:

$$\begin{aligned} |\Psi_1^{\text{SB}}\rangle &= \sum_k \widehat{K}_k |\psi^{\text{S}}\rangle \otimes |\phi_k^{(1,\text{B})}\rangle = \sum_m \widehat{M}_m |\psi^{\text{S}}\rangle \otimes \left(\widehat{U}_\text{B} |\phi_m^{(2,\text{B})}\rangle \right) \\ &= \sum_m \sum_k \widehat{M}_m |\psi^{\text{S}}\rangle \otimes \left(|\phi_k^{(1,\text{B})}\rangle \langle \phi_k^{(1,\text{B})}| \widehat{U}_\text{B} |\phi_m^{(2,\text{B})}\rangle \right) \\ &= \sum_k \left(\sum_m \mathbb{U}_{km} \widehat{M}_m \right) |\psi^{\text{S}}\rangle \otimes |\phi_k^{(1,\text{B})}\rangle, \end{aligned} \quad (9.83)$$

where we defined $\mathbb{U}_{km} = \langle \phi_k^{(1,\text{B})} | \widehat{U}_\text{B} | \phi_m^{(2,\text{B})} \rangle$. Since this equality is true for any state $|\psi^{\text{S}}\rangle$, we deduce that $\widehat{K}_k = \sum_m \mathbb{U}_{km} \widehat{M}_m$. \blacksquare

9.6. Composition laws of Quantum Maps

From now on, by definition, a physically allowed quantum map will be a Trace-preserving (T) Completely-Positive (CP) map, or TPCP map for shortness. ¹² There are several games that we can play with such TPCP maps. We will omit indicating the time t in all our equations since it is assumed to be fixed.

¹²Some use the acronym CPT, which might create a bit of confusion with fundamental particle symmetries.

Maps of two independent systems) If two independent systems \mathcal{H}_1 and \mathcal{H}_2 evolve with the TPCP maps \mathcal{E}_1 and \mathcal{E}_2 , respectively, then the combined system described by $\mathcal{H}_1 \otimes \mathcal{H}_2$ will evolve with a product map $\mathcal{E}_1 \otimes \mathcal{E}_2$ which is constructed from the Kraus form of the two maps, and can be shown to be TPCP, because we can write it in Kraus form as well.

Composition of two maps for the same system) Let now \mathcal{E}_1 and \mathcal{E}_2 be two TPCP maps for the same system \mathcal{H}_s . One can obviously define the composition (or product) of the two as:

$$\left(\mathcal{E}_2 \circ \mathcal{E}_1\right)(\hat{\rho}_s) = \mathcal{E}_2(\mathcal{E}_1(\hat{\rho}_s)) . \quad (9.84)$$

Obviously, the order is important in general: $\mathcal{E}_2 \circ \mathcal{E}_1 \neq \mathcal{E}_1 \circ \mathcal{E}_2$.

Irreversibility) One might wonder if a map \mathcal{E}^{-1} exists such that:

$$\mathcal{E}^{-1} \circ \mathcal{E} = \mathbb{1}_s \quad ?$$

Unfortunately, such a map that does a sort of *undo* of the generally dissipative evolution induced by \mathcal{E} does not exist, at least *in general*. In some sense, this is a manifestation of *irreversibility*. Obviously, the inverse exists for a coherent unitary evolution. Other cases in which an inverse exists are discussed in Ref. [44]. The ability to construct an inverse would of course be crucial for *quantum error correction*. One can show that this can be partly achieved in particular subspaces.

Convex combination) Given two TPCP maps \mathcal{E}_0 and \mathcal{E}_1 one can consider the convex interpolation:

$$\mathcal{E}_q = q\mathcal{E}_1 + (1 - q)\mathcal{E}_0 \quad \text{with } q \in [0, 1] , \quad (9.85)$$

which can be shown to be TPCP as well. The generalisation to a convex linear combination of $n > 2$ maps, with coefficients $q_k \geq 0$ such that $\sum_{k=1, n} q_k = 1$, is quite obvious.

Unitary map) A unitary map is the exceptional (and easily invertible) case describing the evolution under a unitary evolution operator \hat{U} :

$$\mathcal{E}(\hat{\rho}_s) = \hat{U} \hat{\rho}_s \hat{U}^\dagger .$$

Convex combination of unitary maps) Consider now n unitary operators $\hat{U}_{k=1 \dots n}$ and the corresponding convex combination of unitary maps:

$$\mathcal{E}(\hat{\rho}_s) = \sum_{k=1}^n q_k \hat{U}_k \hat{\rho}_s \hat{U}_k^\dagger , \quad (9.86)$$

which can be shown to be TPCP. You can view it as a special Kraus map with $\hat{K}_k = \sqrt{q_k} \hat{U}_k$. Notice that such a map is *not unitary*, in general, except for $n = 1$.

Unital map) A TPCP map is called *unital* if it leaves the identity invariant:

$$\mathcal{E}(\mathbb{1}_s) = \mathbb{1}_s . \quad (9.87)$$

Since the identity often emerges as an *infinite temperature* density matrix, we understand the role of unital maps in connection to the fact that the infinite temperature thermal state would be conserved by such a map. Notice that the map in Eq. (9.86) — a convex combination of unitaries — is automatically *unital*, as you can easily check. ¹³ But the inverse is not true: there are unital maps which are not convex combinations of unitaries.

13

$$\mathcal{E}(\mathbb{1}_s) = \sum_{k=1}^n q_k \hat{U}_k \mathbb{1}_s \hat{U}_k^\dagger = \left(\sum_{k=1}^n q_k \right) \mathbb{1}_s = \mathbb{1}_s .$$

Heisenberg representation of a map) In the ordinary coherent evolution, you can shift the time dependence from the state (density matrix) to the operators. In our open system dynamics, the expectation value of a system operator \widehat{O} at time t would be given by:

$$\langle \widehat{O} \rangle_t = \text{Tr}_S(\widehat{\rho}_S(t) \widehat{O}) = \text{Tr}_S(\mathcal{E}_{(t,0)}(\widehat{\rho}_S(0)) \widehat{O}) \stackrel{?}{=} \text{Tr}_S(\widehat{\rho}_S(0) \mathcal{E}_{H,t}(\widehat{O})), \quad (9.88)$$

where the question mark highlights the fact that this would be our requirement for the Heisenberg map \mathcal{E}_H transforming now the operator \widehat{O} , and the equality should hold for all possible $\widehat{\rho}_S(0)$ and \widehat{O} . If you recall the Kraus representation in Eq. (9.60)

$$\widehat{\rho}_S(t) = \mathcal{E}_{(t,0)}(\widehat{\rho}_S(0)) = \sum_k \widehat{K}_k(t) \widehat{\rho}_S(0) \widehat{K}_k^\dagger(t), \quad (9.89)$$

we can easily get, using the cyclic property of the trace, that:

$$\mathcal{E}_{H,t}(\widehat{O}) \stackrel{\text{def}}{=} \sum_k \widehat{K}_k^\dagger(t) \widehat{O} \widehat{K}_k(t) \quad (9.90)$$

This is called *dual map*. Notice that such a map is *unital* since, as you recall (see Eq. (9.7)):

$$\mathcal{E}_{H,t}(\mathbb{1}_S) = \sum_k \widehat{K}_k^\dagger(t) \widehat{K}_k(t) = \mathbb{1}_S.$$

Nevertheless, the dual map is in general *not trace-preserving*, as it is not of the Kraus form, since:

$$\sum_k \widehat{K}_k(t) \widehat{K}_k^\dagger(t) \neq \mathbb{1}_S \quad \text{in general.}$$

9.7. Useful examples of single-Qbit maps

We consider here very useful maps for a system made of a single Qbit. ¹⁴ The general input state of these maps will be a mixed state of the form:

$$\widehat{\rho}_S = \frac{\mathbb{1} + \mathbf{p} \cdot \widehat{\boldsymbol{\sigma}}}{2} = \begin{pmatrix} p_0 & \gamma^* \\ \gamma & p_1 \end{pmatrix}, \quad (9.91)$$

where $\mathbf{p} = (p_x, p_y, p_z)$ with $|\mathbf{p}| \leq 1$ is the polarisation vector in the Bloch sphere, $p_0 = (1 + p_z)/2$, and $p_1 = (1 - p_z)/2$ are the *populations* of the two states

$$|0\rangle = |\uparrow\rangle \quad \text{and} \quad |1\rangle = |\downarrow\rangle.$$

Finally, $\gamma = p_x + ip_y$ is the so-called *coherence*, with $|\gamma|^2 = p_x^2 + p_y^2 \leq 1 - p_z^2 = 4p_0p_1$.

9.7.1. Phase damping (or dephasing)

Consider a system Qbit $\{|0^S\rangle, |1^S\rangle\}$ in interaction with a “pointer” system. We would like to devise an “interaction” term in such a way that the pointer might effectively “measure” (with some small probability) the state in which the Qbit is, *without changing* the state of the Qbit itself. This is a kind of “*which way*” measurement. To have a picture in mind, think that you have a spin that with probability $(1 - q)$ passes undisturbed the apparatus, while with probability q it enters a Stern-Gerlach device that can effectively “measure” the $\widehat{\sigma}^z$ operator. Our “pointer” Hilbert space \mathcal{H}_B will therefore have *three* states $\{|0^B\rangle, | + 1^B\rangle, | - 1^B\rangle\}$: a state $|0^B\rangle$ — a kind of fiducial or idle state — if the spin passes undisturbed, a state $| + 1^B\rangle$ to which the pointer switches if the spin is in $|0^S\rangle$, and a state

¹⁴The present section is heavily based on John Preskill’s [lecture notes](#).

$| - 1^B \rangle$ to which the pointer moves if the spin is in $| 1^S \rangle$. The “*Stinespring representation*”¹⁵ of such a system-pointer interaction is:

$$\begin{cases} \widehat{U}_{\text{deph}} | 0^S \rangle \otimes | 0^B \rangle = | 0^S \rangle \otimes \left(\sqrt{1-q} | 0^B \rangle + \sqrt{q} | + 1^B \rangle \right) \\ \widehat{U}_{\text{deph}} | 1^S \rangle \otimes | 0^B \rangle = | 1^S \rangle \otimes \left(\sqrt{1-q} | 0^B \rangle + \sqrt{q} | - 1^B \rangle \right) \end{cases} . \quad (9.92)$$

To get a Kraus representation,¹⁶ we trace over the pointer, obtaining the following three Hermitian Kraus operators:

$$\widehat{K}_0 = \sqrt{1-q} \mathbb{1}_S \quad \widehat{K}_+ = \sqrt{q} \frac{\mathbb{1} + \hat{\sigma}^z}{2} \quad \widehat{K}_- = \sqrt{q} \frac{\mathbb{1} - \hat{\sigma}^z}{2} , \quad (9.94)$$

where you recognise the projector on $| 0^S \rangle$ in \widehat{K}_+ and the projector on $| 1^S \rangle$ in \widehat{K}_- . These three Kraus operators are clearly *redundant*, as the basic ingredients are $\mathbb{1}$ and $\hat{\sigma}^z$. Substituting in the Kraus map we get:

$$\mathcal{E}_q^{\text{deph}}(\hat{\rho}_S) = (1-q)\hat{\rho}_S + q \frac{\mathbb{1} + \hat{\sigma}^z}{2} \hat{\rho}_S \frac{\mathbb{1} + \hat{\sigma}^z}{2} + q \frac{\mathbb{1} - \hat{\sigma}^z}{2} \hat{\rho}_S \frac{\mathbb{1} - \hat{\sigma}^z}{2} = \left(1 - \frac{q}{2} \right) \hat{\rho}_S + \frac{q}{2} \hat{\sigma}^z \hat{\rho}_S \hat{\sigma}^z . \quad (9.95)$$

i

Info: We recognise here a convex combination of two unitary maps: the identity and the map $\hat{\rho}_S \rightarrow \hat{\sigma}^z \hat{\rho}_S \hat{\sigma}^z$. Recall that $\hat{\sigma}^z \hat{\sigma}^z \hat{\sigma}^z = \hat{\sigma}^z$, while $\hat{\sigma}^z \hat{\sigma}^{x,y} \hat{\sigma}^z = -\hat{\sigma}^{x,y}$. Hence $\hat{\rho}_S \rightarrow \hat{\sigma}^z \hat{\rho}_S \hat{\sigma}^z$ *changes sign to the off-diagonal elements*:

$$\hat{\rho}_S = \begin{pmatrix} p_0 & \gamma^* \\ \gamma & p_1 \end{pmatrix} \longrightarrow \hat{\sigma}^z \hat{\rho}_S \hat{\sigma}^z = \begin{pmatrix} p_0 & -\gamma^* \\ -\gamma & p_1 \end{pmatrix} . \quad (9.96)$$

Putting the two terms together, we realise that the diagonal elements (the “populations”) are untouched, while the off-diagonal elements (the “coherences”) are multiplied by a factor $(1-q)$:

$$\hat{\rho}_S = \begin{pmatrix} p_0 & \gamma^* \\ \gamma & p_1 \end{pmatrix} \longrightarrow \mathcal{E}_q^{\text{deph}}(\hat{\rho}_S) = \begin{pmatrix} p_0 & (1-q)\gamma^* \\ (1-q)\gamma & p_1 \end{pmatrix} . \quad (9.97)$$

Bloch-sphere representation. The dephasing map leaves untouched the p_z component of the Bloch polarization vector \mathbf{p} , while it shrinks $p_{x,y} \rightarrow (1-q)p_{x,y}$. The Bloch sphere is shrunk into a “cigar-like” ellipsoid aligned along the z axis.¹⁷

Continuous dephasing. Consider a pure dephasing map occurring continuously in time. To get to the time-continuum limit, we imagine that $q = \Gamma_\varphi \Delta t$, where Γ_φ is the probability-rate (or probability per-unit-time) of dephasing, so that $q \ll 1$ when $\Delta t \ll 1$. In a time $t = n\Delta t$ we can think of applying the map n times, each time multiplying the off-diagonal elements by $(1-q) = (1 - \Gamma_\varphi t/n)$, hence obtaining in the off-diagonal elements a factor:

$$(1-q)^n = \left(1 - \frac{\Gamma_\varphi t}{n} \right)^n \xrightarrow{n \rightarrow \infty} e^{-\Gamma_\varphi t} . \quad (9.98)$$

¹⁵Observe that we define it only on a subspace in which the initial pointer state is $| 0^B \rangle$.

¹⁶One can verify that these other choices are also possible Kraus representations:

$$\widehat{K}_1 = \begin{pmatrix} 1 & 0 \\ 0 & (1-q) \end{pmatrix} = \widehat{K}_1^\dagger \quad \text{and} \quad \widehat{K}_2 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{q(2-q)} \end{pmatrix} = \widehat{K}_2^\dagger \quad (9.93)$$

¹⁷Interestingly, there is no TPCP map that can shrink *one component* of the Bloch polarisation vector only, mapping the Bloch sphere into a “pancake” close to the equator plane. This is called the *no-pancake theorem*.

So, the off-diagonal elements decrease in time with a rate Γ_φ , and eventually the density matrix becomes purely diagonal in the “preferred basis” $|0^S\rangle$ and $|1^S\rangle$:

$$\hat{\rho}_s = \begin{pmatrix} p_0 & \gamma^* \\ \gamma & p_1 \end{pmatrix} \longrightarrow \begin{pmatrix} p_0 & e^{-\Gamma_\varphi t} \gamma^* \\ e^{-\Gamma_\varphi t} \gamma & p_1 \end{pmatrix} \xrightarrow{n \rightarrow \infty} \begin{pmatrix} p_0 & 0 \\ 0 & p_1 \end{pmatrix}. \quad (9.99)$$

Visibility. The previous discussion did not take into account the intrinsic system dynamics. Suppose that we have a system initially in the state $|\psi^S(0)\rangle = \frac{1}{\sqrt{2}}(|0^S\rangle + |1^S\rangle)$, an equal superposition of the two basis states obtained by the Hadamard, $\mathbf{H}|0^S\rangle$, and that the system Hamiltonian is $\hat{H}_S = \hbar\omega(\mathbb{1} - \hat{\sigma}^z)/2$, so that the unperturbed evolution would be

$$|\psi^S(t)\rangle = \frac{1}{\sqrt{2}}(|0^S\rangle_S + e^{-i\omega t}|1^S\rangle) \implies \hat{\rho}_s(t) = \frac{1}{2} \begin{pmatrix} 1 & e^{-i\omega t} \\ e^{i\omega t} & 1 \end{pmatrix}. \quad (9.100)$$

Now, let us account for a *continuous dephasing* on top of the free evolution:

$$\hat{\rho}_s(t) = \frac{1}{2} \begin{pmatrix} 1 & e^{-i\omega t} \\ e^{i\omega t} & 1 \end{pmatrix} \xrightarrow{\text{dephasing}} \hat{\rho}_s(t) = \frac{1}{2} \begin{pmatrix} 1 & e^{-i\omega t - \Gamma_\varphi t} \\ e^{i\omega t - \Gamma_\varphi t} & 1 \end{pmatrix}. \quad (9.101)$$

Now imagine we measure repeatedly, for different realisations of the dynamics up to time t , the two-level system in the basis of $\hat{\sigma}^x$, and calculate

$$\text{Prob}(+, \mathbf{x} | \hat{\rho}_s(t)) = \langle +, \mathbf{x} | \hat{\rho}_s(t) | +, \mathbf{x} \rangle = \frac{1}{2} \left(1 + e^{-\Gamma_\varphi t} \cos \omega t \right). \quad (9.102)$$

We see that the dephasing rate can be measured by fitting the exponential decay of the *visibility* of the measured coherent oscillations versus t . See 3.4.2 in Preskill’s [lecture notes](#) for more comments on this issue.

A different look at phase-damping. Imagine we have an apparatus (like a Stern-Gerlach) that produces a certain Qbit state of the form:

$$|\psi_\phi\rangle = z_0|0\rangle + z_1 e^{i\phi}|1\rangle,$$

where z_0 and z_1 are *fixed* (for instance, real), but the phase ϕ is not entirely determined: it can fluctuate from a given preparation of the state to the next, with a probability density $p(\phi) = p(-\phi)$. The state produced by a large ensemble of similar preparations is therefore:

$$\hat{\rho}_s = \int_{-\pi}^{\pi} d\phi p(\phi) |\psi_\phi\rangle \langle \psi_\phi| = \begin{pmatrix} |z_0|^2 & (1-q)z_0z_1^* \\ (1-q)z_0^*z_1 & |z_1|^2 \end{pmatrix} \quad (9.103)$$

where $(1-q) = \int_{-\pi}^{\pi} p(\phi) \cos \phi$. This is, in essence, another physical realisation of a phase-damping map, but this time the mechanism behind the map is not induced by the usual framework of system-plus-environment interaction, but rather by an imperfect preparation.

9.7.2. Amplitude damping (or relaxation)

This is a model for a map describing the spontaneous emission of a photon from an excited two-level atom, in the state $|1^S\rangle$, decaying with probability q to its ground state $|0^S\rangle$. We call $|0^B\rangle$ the state of the “environment” without the photon, and $|1^B\rangle$ the state where one photon has been emitted: evidently, you could think of the “environment”, here, as the quantum electromagnetic field surrounding the atom, whose degrees of freedom have been quite drastically truncated. A “Stinespring” isometry representing such an interaction might be written as:

$$\begin{cases} \hat{U}_{\text{relax}}|0^S\rangle \otimes |0^B\rangle = |0^S\rangle \otimes |0^B\rangle \\ \hat{U}_{\text{relax}}|1^S\rangle \otimes |0^B\rangle = \sqrt{1-q}|1^S\rangle \otimes |0^B\rangle + \sqrt{q}|0^S\rangle \otimes |1^B\rangle \end{cases}, \quad (9.104)$$

where, notice, we have restricted the environment's initial state to $|0^B\rangle$, the state without photons. To get a Kraus representation, we trace over the bath, obtaining the following two Kraus operators:

$$\widehat{K}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-q} \end{pmatrix} \quad \widehat{K}_1 = \begin{pmatrix} 0 & \sqrt{q} \\ 0 & 0 \end{pmatrix} = \sqrt{q} \hat{\sigma}^+ , \quad (9.105)$$

where you should recall that $|1^S\rangle = |\downarrow\rangle$, hence $\hat{\sigma}^+$ describes the decay to the ground state $|0^S\rangle = |\uparrow\rangle$. The operator \widehat{K}_1 describes the so-called “*quantum jump*”, while \widehat{K}_0 describes the change of the state when there is no jump.

Disregarding the state of the environment, we get the following Kraus map:

$$\mathcal{E}_q^{\text{relax}}(\hat{\rho}_S) = \widehat{K}_0 \hat{\rho}_S \widehat{K}_0^\dagger + \widehat{K}_1 \hat{\rho}_S \widehat{K}_1^\dagger = \begin{pmatrix} p_0 + q p_1 & \sqrt{1-q} \gamma^* \\ \sqrt{1-q} \gamma & (1-q) p_1 \end{pmatrix} . \quad (9.106)$$

In the extreme limit $q = 1$, we have:

$$\mathcal{E}_{q=1}^{\text{relax}}(\hat{\rho}_S) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = |0^S\rangle\langle 0^S| , \quad (9.107)$$

hence the system is mapped into the ground state $|0^S\rangle\langle 0^S|$. Such a map describes *relaxation towards the ground state*, and hence *dissipation*.¹⁸

The arbitrariness of the reconstruction. Recall that we defined $\widehat{U}_{\text{relax}}$ only on the subspace of input states in which the photon is absent: $|0^B\rangle$. One might extend it on the full 4×4 Hilbert space at hand, but the extension has no physical meaning. Here is one possibility. Order the states of the basis as

$$\{|0^S\rangle \otimes |0^B\rangle, |1^S\rangle \otimes |0^B\rangle, |0^S\rangle \otimes |1^B\rangle, |1^S\rangle \otimes |1^B\rangle\} ,$$

and write the matrix for the extended $\widehat{U}_{\text{relax}}$ as:

$$\widehat{U}_{\text{relax}} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-q} & -\frac{\sqrt{q}}{2} & \frac{\sqrt{q}}{2} \\ 0 & \sqrt{q} & \frac{\sqrt{1-q}}{2} & -\frac{\sqrt{1-q}}{2} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{array} \right) . \quad (9.109)$$

Notice the elements of the right 2 columns, which have been “invented” to make the matrix a 4×4 unitary. They possess no physical meaning! Here is a second possible extension, less meaningless:

$$\widehat{U}_{\text{relax}} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & \sqrt{1-q} & -\sqrt{q} & 0 \\ 0 & \sqrt{q} & \sqrt{1-q} & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) , \quad (9.110)$$

predicting a re-excitation by photon absorption:

$$\widehat{U}_{\text{relax}} |0^S\rangle \otimes |1^B\rangle = \sqrt{1-q} |0^S\rangle \otimes |1^B\rangle - \sqrt{q} |1^S\rangle \otimes |0^B\rangle . \quad (9.111)$$

Needless to say, the full physics comes out properly only from an appropriate treatment of the QED interaction Hamiltonian [45].

¹⁸Notice that for this relaxation map the composition is closed, multiplicative and abelian:

$$\mathcal{E}_{q_2} \circ \mathcal{E}_{q_1} = \mathcal{E}_{q_2 q_1} . \quad (9.108)$$

Continuous relaxation. As before, consider a relaxation map occurring continuously in time. To get to the time-continuum limit, we take $q = \Gamma_R \Delta t$, where Γ_R is the probability-rate (or probability per-unit-time) of the relaxation process, so that $q \ll 1$ when $\Delta t \ll 1$. At time $t = n\Delta t$, i.e. after n applications of the map, the excited state population becomes

$$(1 - q)^n p_1 = \left(1 - \frac{\Gamma_R t}{n}\right)^n p_1 \xrightarrow{n \rightarrow \infty} e^{-\Gamma_R t} p_1. \quad (9.112)$$

The off-diagonal elements after n applications of the map become:

$$\left(\sqrt{1 - q}\right)^n \gamma = \left(1 - \frac{\Gamma_R t}{n}\right)^{\frac{n}{2}} \gamma \xrightarrow{n \rightarrow \infty} e^{-\Gamma_R t/2} \gamma. \quad (9.113)$$

The whole continuous-time map then reads:

$$\hat{\rho}_s = \begin{pmatrix} p_0 & \gamma^* \\ \gamma & p_1 \end{pmatrix} \longrightarrow \begin{pmatrix} p_0 + (1 - e^{-\Gamma_R t}) p_1 & e^{-\Gamma_R t/2} \gamma^* \\ e^{-\Gamma_R t/2} \gamma & e^{-\Gamma_R t} p_1 \end{pmatrix} \xrightarrow{t \rightarrow \infty} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \quad (9.114)$$

❶

Relaxation, decoherence, dephasing: T_1 and T_2 . There are standard names — originating from the NMR literature — for the inverse rates appearing in the previous expression: one calls T_1 the *relaxation time*, i.e., the exponential decay time of the excited population, and T_2 the *decoherence time*, i.e., the exponential decay time of the off-diagonal terms in the density matrix, the so-called *coherences*. For the present “pure-relaxation” map we have:

$$T_2 = \frac{2}{\Gamma_R} = 2T_1. \quad (9.115)$$

If we consider that a dephasing mechanism accompanies the relaxation process, the off-diagonal element will decay as

$$\gamma \rightarrow e^{-\left(\frac{\Gamma_R}{2} + \Gamma_\varphi\right)t} \gamma \stackrel{\text{def}}{=} e^{-\Gamma_D t} \gamma$$

hence the decoherence time would be: ^a

$$\frac{1}{T_2} = \Gamma_D = \frac{\Gamma_R}{2} + \Gamma_\varphi = \frac{1}{2T_1} + \Gamma_\varphi. \quad (9.116)$$

^aNotice that T_2 can be arbitrarily small in the presence of a large dephasing, and approaches at most $2T_1$ in the absence of dephasing.

Watching the environment. So far we have considered a relaxation map, even occurring repeatedly n times, *disregarding* the environment: this is what quantum maps are supposed to do. Now we take a different perspective and imagine *keeping track of the measurements performed on the environment* (i.e., the photon detection). Let us look again at the unitary representation of the relaxation map in Eq. (9.104), but this time we write it for an arbitrary initial state $|\psi^S\rangle = z_0|0^S\rangle + z_1|1^S\rangle$:

$$\hat{U}_{\text{relax}} \left(z_0|0^S\rangle + z_1|1^S\rangle \right) \otimes |0^B\rangle = \left(z_0|0^S\rangle + z_1\sqrt{1 - q}|1^S\rangle \right) \otimes |0^B\rangle + z_1\sqrt{q}|0^S\rangle \otimes |1^B\rangle. \quad (9.117)$$

Imagine applying the evolution for a second time:

$$\begin{aligned} \hat{U}_{\text{relax}}^2 \left(z_0|0^S\rangle + z_1|1^S\rangle \right) \otimes |0^B\rangle &= \left(z_0|0^S\rangle + z_1\sqrt{(1 - q)^2}|1^S\rangle \right) \otimes |0^B\rangle \\ &+ z_1 \left(\sqrt{q} + \sqrt{1 - q}\sqrt{q} \right) |0^S\rangle \otimes |1^B\rangle \\ &+ z_1\sqrt{q} \hat{U}_{\text{relax}} |0^S\rangle \otimes |1^B\rangle, \end{aligned} \quad (9.118)$$

where the last term describes the action of \hat{U}_{relax} on an input state with a photon, hence that part of the Hilbert space where the extension of \hat{U}_{relax} has a large arbitrariness.

Let us *assume* that the photon, once emitted, is gone forever, and it will not be able to induce back a transition described by \hat{U}_{relax} . So, we drop this term, and do the same for the next steps. We proceed iterating n times. If we project on the state with no photons we get:

$$\langle 0^{\text{B}} | \hat{U}_{\text{relax}}^n (z_0 |0^{\text{S}}\rangle + z_1 |1^{\text{S}}\rangle) \otimes |0^{\text{B}}\rangle = z_0 |0^{\text{S}}\rangle + z_1 \sqrt{(1-q)^n} |1^{\text{S}}\rangle. \quad (9.119)$$

So, if we detect no photon, then we have automatically projected on a state that, up to normalization, can be written as:

$$(\hat{K}_0)^n |\psi^{\text{S}}\rangle = z_0 |0^{\text{S}}\rangle + z_1 \sqrt{(1-q)^n} |1^{\text{S}}\rangle \xrightarrow{n \rightarrow \infty} z_0 |0^{\text{S}}\rangle + z_1 e^{-\Gamma_{\text{R}} t/2} |1^{\text{S}}\rangle. \quad (9.120)$$

The *a posteriori* quantum state of the system, given that no photon was detected, approaches the ground state, as $t \rightarrow \infty$. Strange but true: we have projected on $|0^{\text{S}}\rangle$ by *not having detected a photon*.

Exercise 9.3. Assume that the full 4×4 unitary \hat{U}_{relax} is given by Eq. (9.110). Calculate the effect of applying n times the unitary, writing down the system state when no photon is detected, $\langle 0^{\text{B}} | \hat{U}_{\text{relax}}^n (z_0 |0^{\text{S}}\rangle + z_1 |1^{\text{S}}\rangle) \otimes |0^{\text{B}}\rangle$, and when a photon is detected, $\langle 1^{\text{B}} | \hat{U}_{\text{relax}}^n (z_0 |0^{\text{S}}\rangle + z_1 |1^{\text{S}}\rangle) \otimes |0^{\text{B}}\rangle$.

9.7.3. Depolarising channel

We have already seen that the dephasing map can be regarded as a convex combination of the unit map with $\hat{\rho}_{\text{S}} \rightarrow \hat{\sigma}^z \hat{\rho}_{\text{S}} \hat{\sigma}^z$, describing the effect of a phase flip error. To be more precise, the effect of the three Pauli matrices on a state is:

$$\begin{aligned} \text{Phase flip error)} \quad & |\psi\rangle \rightarrow \hat{\sigma}^z |\psi\rangle & \text{or:} \quad & |0\rangle \rightarrow |0\rangle, \quad |1\rangle \rightarrow -|1\rangle \\ \text{Bit flip error)} \quad & |\psi\rangle \rightarrow \hat{\sigma}^x |\psi\rangle & \text{or:} \quad & |0\rangle \rightarrow |1\rangle, \quad |1\rangle \rightarrow |0\rangle \\ \text{Phase and Bit flip error)} \quad & |\psi\rangle \rightarrow \hat{\sigma}^y |\psi\rangle & \text{or:} \quad & |0\rangle \rightarrow i|1\rangle, \quad |1\rangle \rightarrow -i|0\rangle \end{aligned} \quad (9.121)$$

Now we define an environment \mathcal{H}_{B} with 4 states: $\{|0^{\text{B}}\rangle, |1^{\text{B}}\rangle, |2^{\text{B}}\rangle, |3^{\text{B}}\rangle\}$, and a Stinespring isometry of the form:

$$\hat{V} |\psi^{\text{S}}\rangle = \sqrt{q_0} |\psi^{\text{S}}\rangle \otimes |0^{\text{B}}\rangle + \sum_{k=1}^3 \sqrt{q_k} \hat{\sigma}^{(k)} |\psi^{\text{S}}\rangle \otimes |k^{\text{B}}\rangle, \quad (9.122)$$

where $q_0 + q_1 + q_2 + q_3 = 1$, and $\hat{\sigma}^{(1,2,3)}$ is an alternative notation for $\hat{\sigma}^{x,y,z}$. We trace over the environment, $\text{Tr}_{\text{B}}(\hat{V} |\psi^{\text{S}}\rangle \langle \psi^{\text{S}}| \hat{V}^\dagger)$, obtaining the quantum map:

$$\mathcal{E}(\hat{\rho}_{\text{S}}) = q_0 \hat{\rho}_{\text{S}} + \sum_{k=1}^3 q_k \hat{\sigma}^{(k)} \hat{\rho}_{\text{S}} \hat{\sigma}^{(k)} \quad \text{with} \quad \sum_{k=0}^3 q_k = 1. \quad (9.123)$$

This is a convex combination of unitary maps obtained with the Pauli matrices. It is called the *depolarising map* or channel. It is a *unital* map, i.e., it keeps the identity unchanged. If we express the final density matrix in the usual Bloch polarisation vector representation:

$$\mathcal{E}(\hat{\rho}_{\text{S}}) = \frac{\mathbb{1}_{\text{S}} + \mathbf{p}' \cdot \hat{\boldsymbol{\sigma}}}{2} \quad (9.124)$$

we have a map that transforms $\mathbf{p} \rightarrow \mathbf{p}'$.

Exercise 9.4. Show that for the depolarising map one has:

$$p'_x = p_x (1 - 2(q_2 + q_3)) \quad p'_y = p_y (1 - 2(q_1 + q_3)) \quad p'_z = p_z (1 - 2(q_1 + q_2)). \quad (9.125)$$

Notice that for $q_0 = q_1 = q_2 = q_3 = \frac{1}{4}$ we get $\mathbf{p}' = 0$ no matter what the initial polarisation \mathbf{p} is: the final density matrix is the “infinite temperature” state $\frac{1}{2} \mathbb{1}_{\text{S}}$.

Exercise 9.5. Show that, as a consequence of the previous exercise:

$$\frac{1}{2}\mathbb{1}_s = \frac{1}{4}\left(\hat{\rho}_s + \sum_{k=1}^3 \hat{\sigma}^{(k)} \hat{\rho}_s \hat{\sigma}^{(k)}\right). \quad (9.126)$$

Using this result, show that

$$\mathcal{E}(\hat{\rho}_s) = \frac{p}{2}\mathbb{1} + (1-p)\hat{\rho}_s, \quad (9.127)$$

is a particular depolarising map. Identify the corresponding values of q_k , for $k = 0, \dots, 3$.

10. Open Quantum Systems and Lindblad Quantum Master Equation

So far we were concerned with quantum maps where the evolution time t was considered *fixed*, and might be eliminated from most of the notation. Now we want to consider a continuous time evolution, and we want to see under what conditions we can write a differential equation for the system density matrix, of the form:

$$\frac{d}{dt}\hat{\rho}_s(t) = \mathcal{L}(\hat{\rho}_s(t)) , \quad (10.1)$$

where $\mathcal{L}(\cdot)$ is a suitable super-operator. The crucial requirement is that the r.h.s depends only on $\hat{\rho}_s(t)$.

In this chapter we will discuss the Markovian conditions which we need to write a time-evolution differential equation for the state in this form, known as *quantum master equation* (QME). Later in the Chapter, we will see the Lindblad construction [46] for such a QME. Appendix D contains a perturbative derivation of various forms of QME valid in the limit in which the interaction between the system and the environment is weak.

10.1. The Markovian condition

Obviously, for an isolated system the evolution is described by the von Neumann equation, hence

$$\mathcal{L}(\hat{\rho}_s(t)) = \frac{1}{i\hbar}[\hat{H}_s, \hat{\rho}_s(t)] . \quad (10.2)$$

For a system interacting with a bath (or environment) we know that we can write a TPCP (Kraus) map:

$$\hat{\rho}_s(0) \rightarrow \hat{\rho}_s(t) = \mathcal{E}_{(t,0)}(\hat{\rho}_s(0)) = \sum_k \hat{K}_k(t)\hat{\rho}_s(0)\hat{K}_k^\dagger(t) , \quad (10.3)$$

but taking a time derivative of such an expression leads us nowhere. It turns out that an important requirement for the quantum map \mathcal{E} describing the dissipative evolution is a *Markovian condition* [44, 47]. More precisely, we need a one-parameter map of the form

$$\hat{\rho}_s(0) \xrightarrow{\mathcal{E}_t} \hat{\rho}_s(t) = \mathcal{E}_t(\hat{\rho}_s(0)) , \quad (10.4)$$

where \mathcal{E}_t — notice the absence of the usual label $(t, 0)$ keeping track of the initial time — satisfies the following three conditions:

- 1) $\mathcal{E}_0 = \mathbb{1}_s$
- 2) \mathcal{E}_t is continuous in t , which means that:

$$\langle \hat{O} \rangle_t = \text{Tr}_s(\hat{O}\mathcal{E}_t(\hat{\rho}_s)) \quad \text{is continuous in } t \quad \forall \hat{\rho}_s, \quad \forall \hat{O} \text{ observable} . \quad (10.5)$$

- 3) \mathcal{E}_t is a semi-group, which means that:

$$\mathcal{E}_{t+s} = \mathcal{E}_t \circ \mathcal{E}_s = \mathcal{E}_s \circ \mathcal{E}_t . \quad (10.6)$$



Divisibility vs one-parameter semigroup. Observe how condition 3) is different from the composition law:

$$\mathcal{E}_{t+s,0} = \mathcal{E}_{t+s,s} \circ \mathcal{E}_{s,0} = \mathcal{E}_{t+s,t} \circ \mathcal{E}_{t,0} ,$$

which is often referred to as *divisibility* condition. For more discussions about quantum Markovianity, see Refs. [44,47]. ^a

^aIndeed, $\mathcal{E}_{t+s,s}$ has nothing to do with $\mathcal{E}_{t,0}$: the initial time is generally important, because the whole history of the evolution leaves in principle important effects on the combined state of the system (recall that we assume that system and environment are in a product state at time $t = 0$). To better understand this point, consider the Stinespring form of a *time-independent* \hat{H}_S :

$$\begin{aligned} \hat{\rho}_S(t+s) &= \text{Tr}_B \left(\hat{U}_{\text{tot}}(t+s) \hat{\rho}_S(0) \otimes \hat{\rho}_B(0) \hat{U}_{\text{tot}}^\dagger(t+s) \right) \\ &= \text{Tr}_B \left(\hat{U}_{\text{tot}}(t) \left(\hat{U}_{\text{tot}}(s) \hat{\rho}_S(0) \otimes \hat{\rho}_B(0) \hat{U}_{\text{tot}}^\dagger(s) \right) \hat{U}_{\text{tot}}^\dagger(t) \right) \\ &\neq \text{Tr}_B \left(\hat{U}_{\text{tot}}(t) \left(\mathcal{E}_s(\hat{\rho}_S(0)) \otimes \hat{\rho}_B(0) \right) \hat{U}_{\text{tot}}^\dagger(t) \right) , \end{aligned} \quad (10.7)$$

where our candidate \mathcal{E}_s would be given by:

$$\mathcal{E}_s(\hat{\rho}_S(0)) = \text{Tr}_B \left(\hat{U}_{\text{tot}}(s) \hat{\rho}_S(0) \otimes \hat{\rho}_B(0) \hat{U}_{\text{tot}}^\dagger(s) \right) . \quad (10.8)$$

Observe that, in the correct expression, the initial bath state $\hat{\rho}(0)$ is evolved for a time s and entangled with the system, while in the candidate expression, the initial state $\hat{\rho}_B(0)$ of the bath appears. We understand the Markovian nature as requiring, in some sense, that the bath' state is not affected by the evolution.



Info: In some sense, the semi-group property is reminiscent of the ordinary exponential, or of the Schrödinger evolution operator of a free system:

$$e^{-i(t+s)\hat{H}_S/\hbar} = e^{-it\hat{H}_S/\hbar} e^{-is\hat{H}_S/\hbar} = e^{-is\hat{H}_S/\hbar} e^{-it\hat{H}_S/\hbar} .$$

Let us see the crucial role played by condition 3) in obtaining a Quantum Master Equation (QME). Start from $\hat{\rho}_S(t) = \mathcal{E}_t(\hat{\rho}_S(0))$ and take a time derivative with respect to t :

$$\hat{\rho}_S(t) = \mathcal{E}_t(\hat{\rho}_S(0)) \quad \implies \quad \frac{d}{dt} \hat{\rho}_S(t) = \left(\frac{d}{dt} \mathcal{E}_t \right) (\hat{\rho}_S(0)) . \quad (10.9)$$

Now calculate the derivative of \mathcal{E}_t , which we will show to exist thanks to the semi-group property:

$$\begin{aligned} \frac{d}{dt} \mathcal{E}_t &= \lim_{s \rightarrow 0} \frac{\mathcal{E}_{t+s} - \mathcal{E}_t}{s} = \lim_{s \rightarrow 0} \frac{(\mathcal{E}_s \circ \mathcal{E}_t - \mathcal{E}_t)}{s} = \lim_{s \rightarrow 0} \frac{(\mathcal{E}_s - \mathbb{1}_s) \circ \mathcal{E}_t}{s} \\ &= \mathcal{L} \circ \mathcal{E}_t , \end{aligned} \quad (10.10)$$

where we have defined:

$$\mathcal{L} \stackrel{\text{def}}{=} \lim_{s \rightarrow 0} \frac{(\mathcal{E}_s - \mathbb{1}_s)}{s} , \quad (10.11)$$

which can be shown to be well defined thanks to continuity *and* the semi-group property. ¹



The Quantum Master Equation. Hence we finally deduce that:

$$\frac{d}{dt} \hat{\rho}_S(t) = \left(\frac{d}{dt} \mathcal{E}_t \right) (\hat{\rho}_S(0)) = \left(\mathcal{L} \circ \mathcal{E}_t \right) (\hat{\rho}_S(0)) = \mathcal{L}(\mathcal{E}_t(\hat{\rho}_S(0))) = \mathcal{L}(\hat{\rho}_S(t)) , \quad (10.12)$$

which is the desired QME.

¹Recall that in ordinary analysis one can show that a continuous function with the property $f(t+s) = f(t)f(s)$ and $f(0) = 1$ is indeed infinitely differentiable: it uniquely defines the exponential function!

Notice that sometimes, in analogy with the Schrödinger evolution operator, this differential equation is formally integrated as:

$$\hat{\rho}_s(t) = e^{t\mathcal{L}} \hat{\rho}_s(0), \quad (10.13)$$

which is however only useful for formal purposes, as the exponential of the Liouvillian super-operator is a formidable object.

i

Lindblad form of the QME. We will now show that the most general Quantum Master Equation (QME) compatible with the three requirements given above — in particular with the (Markovian) semi-group property of the evolution map \mathcal{E}_t — has the Lindblad form:

$$\frac{d}{dt} \hat{\rho}_s = \mathcal{L}(\hat{\rho}_s) = \frac{1}{i\hbar} [\tilde{H}_s, \hat{\rho}_s] + \sum_{\mu=1}^{d_s^2-1} \gamma_{\mu} \left(\hat{L}_{\mu} \hat{\rho}_s \hat{L}_{\mu}^{\dagger} - \frac{1}{2} \{ \hat{L}_{\mu}^{\dagger} \hat{L}_{\mu}, \hat{\rho}_s \} \right). \quad (10.14)$$

10.2. The Lindblad construction

Consider the Kraus representation of \mathcal{E}_t :

$$\mathcal{E}_t(\hat{\rho}_s) = \sum_{k=1}^{D_K} \hat{K}_k(t) \hat{\rho}_s \hat{K}_k^{\dagger}(t), \quad (10.15)$$

where $D_K = d_s^2$. We fix a basis for the d_s^2 operators in \mathcal{H}_s in this way: $\hat{F}_0 = \mathbb{1}_s$, while \hat{F}_j for $j = 1 \cdots D_K - 1$ are traceless matrices which generalise the Pauli matrix set $\{\hat{\sigma}^+, \hat{\sigma}^-, \hat{\sigma}^z\}$ for $d_s > 2$. For instance, denoting by $\{|a\rangle$ with $a = 1 \cdots d_s\}$ a system orthonormal basis, we take:

$$\hat{F}_0 = \mathbb{1}_s \quad \hat{F}_{j=1 \cdots d_s-1} = \sum_{a=1}^{d_s} e^{i2\pi a j / d_s} |a\rangle \langle a| \quad \hat{F}_{j \geq d_s} = |a\rangle \langle a'|, \quad (10.16)$$

where the last $d_s(d_s - 1)$ terms have $a' \neq a$, and are represented by matrices having a single 1 in any of the off-diagonal elements.² In terms of this fixed basis we write:

$$\hat{K}_k(t) = \sum_{j=0}^{D_K-1} C_{kj}(t) \hat{F}_j, \quad (10.17)$$

with appropriate complex coefficients $C_{kj}(t)$. From this, by substituting into the Kraus representation, we deduce:

$$\mathcal{E}_t(\hat{\rho}_s) = \sum_{j=0}^{D_K-1} \sum_{j'=0}^{D_K-1} \Phi_{jj'}(t) \hat{F}_j \hat{\rho}_s \hat{F}_{j'}^{\dagger} \quad (10.18)$$

where

$$\Phi_{jj'}(t) = \sum_{k=1}^{D_K} C_{kj}(t) C_{kj'}^*(t) = \Phi_{j'j}^*(t). \quad (10.19)$$

Notice the Hermitean nature of the matrix of coefficients $\Phi_{jj'}(t)$. In a short while, we will also show that such a matrix is *positive definite*.

²The traceless nature of \hat{F}_j for $j = 1 \cdots d_s - 1$ is guaranteed by the fact that we are using the roots-of-unity in the diagonal.

Now we calculate the Liouvillian operator, by taking the time-derivative of the map at $t = 0$:

$$\begin{aligned}\mathcal{L}(\hat{\rho}_s) &= \left. \frac{d}{dt} \mathcal{E}_t(\hat{\rho}_s) \right|_{t=0} = \sum_{j=0}^{D_K-1} \sum_{j'=0}^{D_K-1} \dot{\Phi}_{jj'}(0) \hat{F}_j \hat{\rho}_s \hat{F}_{j'}^\dagger \\ &= \hat{A} \hat{\rho}_s + \hat{\rho}_s \hat{A}^\dagger + \sum_{j=1}^{D_K-1} \sum_{j'=1}^{D_K-1} \Gamma_{jj'} \hat{F}_j \hat{\rho}_s \hat{F}_{j'}^\dagger,\end{aligned}\quad (10.20)$$

where, using $\hat{F}_0 = \mathbb{1}_s$, we singled out the $j = 0$ and $j' = 0$ terms by defining:

$$\hat{A} \stackrel{\text{def}}{=} \frac{1}{2} \dot{\Phi}_{00}(0) \mathbb{1}_s + \sum_{j=1}^{D_K-1} \dot{\Phi}_{j0}(0) \hat{F}_j \quad \text{and} \quad \Gamma_{jj'} \stackrel{\text{def}}{=} \dot{\Phi}_{jj'}(0). \quad (10.21)$$

Observe that the trace conservation implies that $\forall \hat{\rho}_s$:

$$0 = \text{Tr}_s \left(\mathcal{L}(\hat{\rho}_s) \right) = \text{Tr}_s \left(\left(\hat{A} + \hat{A}^\dagger + \sum_{j=1}^{D_K-1} \sum_{j'=1}^{D_K-1} \Gamma_{jj'} \hat{F}_j \hat{F}_{j'}^\dagger \right) \hat{\rho}_s \right), \quad (10.22)$$

which in turns implies that the quantity multiplying $\hat{\rho}_s$ inside the trace must vanish identically:

$$\hat{A} + \hat{A}^\dagger = - \sum_{j=1}^{D_K-1} \sum_{j'=1}^{D_K-1} \Gamma_{jj'} \hat{F}_j \hat{F}_{j'}^\dagger. \quad (10.23)$$

To proceed, two properties are now useful:

$$\begin{cases} \mathcal{E}_0 = \mathbb{1}_s \implies \Phi_{jj'}(0) = \delta_{j,0} \delta_{j',0} \\ \sum_{j=1}^{D_K-1} \sum_{j'=1}^{D_K-1} v_j^* \Phi_{jj'}(t) v_{j'} = \sum_{k=1}^{D_K} \left| \sum_{j=1}^{d_S^2-1} C_{kj}(t) v_j^* \right|^2 \geq 0 \end{cases} \quad (10.24)$$

The second property tells us that the restriction of $\Phi_{jj'}(t)$ to non-zero indices is not only Hermitean, but also non-negative definite. It follows therefore that the rate matrix $\Gamma_{jj'}$ (defined only for $j, j' \geq 1$) is also Hermitean and non-negative:

$$\Gamma_{jj'} = \lim_{s \rightarrow 0} \frac{\Phi_{jj'}(s) - \Phi_{jj'}(0)}{s} = \lim_{s \rightarrow 0} \frac{\Phi_{jj'}(s)}{s}. \quad (10.25)$$

Finally, the operator \hat{A} is in general not Hermitean, but can always be decomposed as:

$$\hat{A} = \frac{1}{2}(\hat{A} + \hat{A}^\dagger) + \frac{1}{2}(\hat{A} - \hat{A}^\dagger) = \frac{1}{2}(\hat{A} + \hat{A}^\dagger) + \frac{1}{i\hbar} \tilde{H}_s, \quad (10.26)$$

where the Hermitean part of \hat{A} is fixed by Eq. (10.23), while the anti-Hermitean part has been identified with a suitable ‘‘Hamiltonian’’ of the system, \tilde{H}_s , generating the von Neumann coherent evolution.



Lamb shift terms. Notice that in general \tilde{H}_s differs from the free Hamiltonian \hat{H}_s of the system by the presence of *bath-induced extra terms*, usually called ‘‘Lamb shift terms’’, by analogy with the very small QED corrections to the hydrogen spectrum.

Collecting all pieces we have:

$$\mathcal{L}(\hat{\rho}_s) = \frac{1}{i\hbar} [\tilde{H}_s, \hat{\rho}_s] + \sum_{j=1}^{D_K-1} \sum_{j'=1}^{D_K-1} \left(\Gamma_{jj'} \hat{F}_j \hat{\rho}_s \hat{F}_{j'}^\dagger - \frac{1}{2} \{ \Gamma_{jj'} \hat{F}_j \hat{F}_{j'}^\dagger, \hat{\rho}_s \} \right). \quad (10.27)$$

To get the final Lindblad diagonal form, we use the fact that rate matrix $\Gamma_{jj'}$ is Hermitean and non-negative. Hence we can find a unitary matrix \mathbb{U} such that

$$\mathbb{U}^\dagger \mathbf{\Gamma} \mathbb{U} = \text{diag}(\gamma_\mu) \quad \text{with} \quad \gamma_{\mu=1 \dots D_K-1} \geq 0. \quad (10.28)$$

Define now the new $D_K - 1$ Lindblad operators:

$$\hat{L}_\mu = \sum_{j=1}^{D_K-1} \hat{F}_j(\mathbb{U})_{j\mu} \quad \Longrightarrow \quad \hat{F}_j = \sum_{\mu=1}^{D_K-1} \hat{L}_\mu(\mathbb{U}^\dagger)_{\mu j}. \quad (10.29)$$

We finally arrive at the final QME in Lindblad form anticipated in Eq. (10.14):

$$\frac{d}{dt} \hat{\rho}_s = \mathcal{L}(\hat{\rho}_s) = \frac{1}{i\hbar} [\tilde{H}_s, \hat{\rho}_s] + \sum_{\mu=1}^{d_s^2-1} \gamma_\mu \left(\hat{L}_\mu \hat{\rho}_s \hat{L}_\mu^\dagger - \frac{1}{2} \{ \hat{L}_\mu^\dagger \hat{L}_\mu, \hat{\rho}_s \} \right). \quad (10.30)$$

For the specific example of a two-level system, $d_s = 2$, one can show, see App. D, that the 4 Lindblad operators, with the associated rate constants, are:

$$\left\{ \begin{array}{ll} \gamma_1 = \gamma_{g \leftarrow e} & \rightarrow \quad \hat{L}_1 = \hat{\sigma}^+ \\ \gamma_2 = \gamma_{e \leftarrow g} & \rightarrow \quad \hat{L}_2 = \hat{\sigma}^- \\ \gamma_3 = \gamma_\varphi & \rightarrow \quad \hat{L}_3 = \frac{1}{\sqrt{2}} \hat{\sigma}^z \\ \gamma_4 = 0 & \rightarrow \quad \hat{L}_4 = \frac{1}{\sqrt{2}} \mathbb{1} \end{array} \right. \quad (10.31)$$

You recognise the *relaxation* map term involving $\hat{\sigma}^+$, and the *dephasing* map term involving $\hat{\sigma}^z$. On top of those terms, we have, in general, an excitation term involving $\hat{\sigma}^-$. For a two-level system coupled to a large thermal bath, the two rates $\gamma_{e \leftarrow g}$ and $\gamma_{g \leftarrow e}$ will eventually lead to a final steady state in which the population ratio of the two levels — ground and excited state — obeys what is predicted by the detailed balance requirement.

11. Introduction to quantum error correction

... omni autem cui multum datum est multum quaeretur ab eo et cui commendaverunt multum plus petent ab eo.^a

Luca 12:48.

^aFrom everyone who has been given much, much will be demanded; and from the one who has been entrusted with much, much more will be asked.

Let me start with a few useful references for the subject quantum error correction which we briefly discuss here. First of all, as usual, a good elementary introduction is given by Chap. 5 of Mermin's book [1][Chap.5]. A companion reading is also given by a small review [48], by David Bacon's chapter [49], and by Chap. 10 of Nielsen [3]. Moving to more advanced treatments, I would suggest Preskill's lecture notes, and Chapter 15 of Kitaev's book [19]. A comprehensive reference for classical coding and error correction is the book by MacWilliams and Sloane [50]. For quantum error correction, consider also studying [51].

Correcting errors might sound like a dreary practical problem, of little aesthetic or conceptual interest. But aside from being of crucial importance for the feasibility of quantum computation, it is also one of the most beautiful and surprising parts of the subject. The surprise is that error correction is possible at all, since the only way to detect errors is to make measurements, but measurement gates disruptively alter the states of the measured Qbits, apparently making things even worse. "Quantum error correction" would seem to be an oxymoron. The beauty lies in the ingenious ways that people have found to get around this apparently insuperable obstacle.

N. David Mermin, *Quantum Computer Science*, Chapter 5

We will see how one can circumvent the difficulty associated to quantum measurement and collapse. Moreover, we will appreciate how the great power of the quantum computation scheme brings about a much larger possibility of errors, without a classical counterpart. For Qbits, you not only have the possibility of bit flips (operated by $\hat{\sigma}^x = \mathbf{X}$): there are, for instance, phase errors, like in $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. More generally, the continuous nature of the Qbit is in principle much more fragile than a digital Cbit, which can only flip.

11.1. Classical error correction and Shannon's theorem

Classical bits in our current digital computers are encoded by the miniaturised transistors packed in silicon chips, each transistor involving a very large number of microscopic degrees of freedom. As such, the probability that a Cbit is flipped due to thermal fluctuations or electromagnetic interactions

with the surrounding world is extremely small: classical error correction is not an important issue for classical computation in our digital computers.

In the classical world, classical error correction is important for *classical communication* of Cbits over noisy communication channels.

❶

The binary symmetric channel. The simplest example of a noisy classical communication model is a *binary symmetric channel*: each single transmitted Cbit, independently of any other Cbits, can suffer a flip with a (small) probability p . In formulas, if $P(b_r|b_t)$ is the conditional probability that we receive the bit $b_r = 0, 1$ conditional on the fact that the transmitted bit was $b_t = 0, 1$, then

$$P(1|0) = P(0|1) = p \quad \text{and} \quad P(0|0) = P(1|1) = 1 - p. \quad (11.1)$$

If $p = 0.1$, and we aim at transmitting a black-and-white picture of $100 \times 100 = 10^4$ bits (pixels), on average 10^3 pixels will be received wrong: a quick look at Fig. 11.1 (under the column “r”) gives a feeling for that error.

To improve on this, a simple strategy is given by *repetition codes*. The simplest such repetition code R_3 consists in adding redundancy in the coding of bits — each bit is repeated three times, mapping $0 \rightarrow 000$ and $1 \rightarrow 111$. Suppose that the source message is the 8-bit string $\underline{s} = 00101100$, then the transmitted message is the 24-bit string $\underline{t} = 000\ 000\ 111\ 000 \dots$ (space added for clarity). A noisy binary symmetric channel can be thought as a random binary string \underline{e} where the probability of a 1 is p . The “noise” or “error” string \underline{e} is summed modulo-2 to the transmitted string \underline{t} , giving a received string $\underline{r} = \underline{t} \oplus \underline{e}$. To *decode* the received string you apply the *majority vote*¹: you split the received message in chunks of 3 bits and read out

$$000 \rightarrow 0, \quad 100 \rightarrow 0, \quad 010 \rightarrow 0, \quad 001 \rightarrow 0, \quad 110 \rightarrow 1, \quad 101 \rightarrow 1, \quad 011 \rightarrow 1, \quad 111 \rightarrow 1.$$

Here is the effect of encoding-transmission-decoding of a string:

\underline{s}	0	0	1	0	1	1	0	0
\underline{t}	0 0 0	0 0 0	1 1 1	0 0 0	1 1 1	1 1 1	0 0 0	0 0 0
\underline{e}	0 0 0	0 0 1	0 0 0	0 0 0	1 0 1	0 0 0	0 0 0	0 0 0
\underline{r}	0 0 0	0 0 1	1 1 1	0 0 0	0 1 0	1 1 1	0 0 0	0 0 0
$\underline{\hat{s}}$	0	0	1	0	0	1	0	0

Here $\underline{\hat{s}}$ is the decoded string obtained by applying the majority vote to \underline{r} . Notice also that while an error was successfully corrected (in the 2nd bit), the occasional presence of *two* flips in the same repeated block of bits leads to uncorrected decoding for the 5th bit.

Figure 11.1 shows the effect of repetition code R_3 on a simple binary picture of 10^4 pixels transmitted over a channel with an error probability $p = 0.1$. Notice how the decoded picture is much better than each of the three received copies. All the blocks in which *at most one* flip error occurs (a single 1 in the corresponding block of \underline{n}) are decoded correctly. Incorrect decoding occurs in the less likely event that *two or more flip errors* occur within the same block. You can calculate the probability of error in decoding a block of 3 bits as:

$$p_b = 3p^2(1 - p) + p^3 \approx 3p^2, \quad (11.2)$$

where the approximation applies for p sufficiently small. Hence, the probability of error in decoding is $p_b \approx 0.03$ for $p = 0.1$: we have considerably reduced the error probability from the bare value 0.1,

¹See MacKay [52][Sec.1.2] for a proof, based on Bayes’ theorem, that the majority vote is the optimal decoding algorithm, in the sense of being the that with the smallest probability of being wrong.

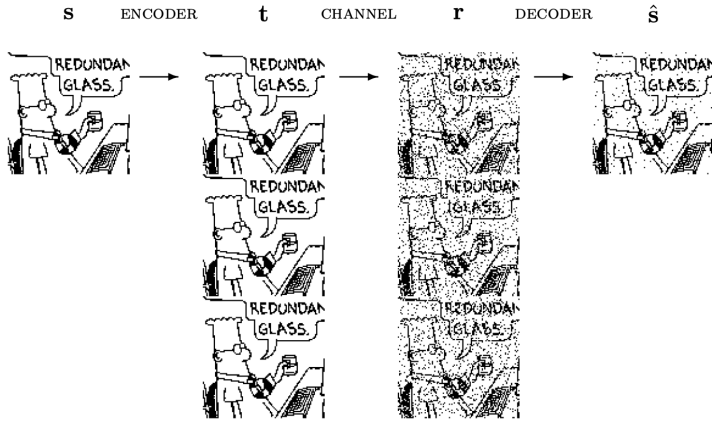


Figure 11.1: Transmission of a figure in a noisy classical channel with $p = 0.1$ with a repetition code R_3 . The probability of a mistake in the decoding is $p_b \approx 3p^2 = 0.03$. But the rate of transmission is reduced to $R = 1/3$. Figure adapted from MacKay (Fig.1.11).

at the price of having to transmit 3 copies of the same bit, hence reducing the rate of transmission to $R = 1/3$.

Repetition codes of longer length, for instance, R_5 , might be adopted to further reduce the decoding error: one can show that R_{61} would reduce p_b to below 10^{-15} ², but the rate would be reduced to $R = 1/61$. For a very nice introduction to this topic, I urge you to read Chapter 1 of MacKay [52].

Figure 11.2 shows the phase diagram of the probability of decoding error p_b versus the transmission rate for repetition codes, and other alternative codes, like the Hamming (7,4) block code, based on adding 3 parity check bits every 4 transmitted bits.³ This figure shows the great achievement of Shannon in his famous *noisy-channel coding theorem*: Shannon proved that you can in principle reach *arbitrarily small decoding errors* p_b at a rate R which is *finite*, depending on the so-called *channel capacity* C , which is related to the *Shannon entropy* of the noisy channel

$$C(p) = 1 - H(p) = 1 - \left(p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p} \right). \tag{11.3}$$

For a noise level of $p = 0.1$, one calculates $C \approx 0.53$, showing the remarkable result that one can *in principle* achieve coding algorithms with arbitrarily small error and rates as large as $R \sim 0.53$, much larger than the value $R = 1/3$ for the R_3 code!

We will not discuss more the theory of classical error correction, which, as remarked, is central in classical *communication* and *information theory*. More information on that in Appendix E, which gives a short overview of classical linear codes, following Ref. [50]. For the remainder of our present discussion, confined to error correction, it is enough to have in mind the majority vote idea behind the repetition scheme R_3 , which will inspire some of the considerations which follow.

²A 100-Terabyte hard-disk, with about 8×10^{14} bits, would be guaranteed to be essentially free from errors. But the rate reduction in repetition codes is terrible: we would need 61 such hard-disks to perform repetition majority voting with that low level of error.

³Hamming codes $H(n, n - r)$ are linear codes defined by r modulo-2 addition constraints. You consider elements of $\{0, 1\}^n$, i.e., bit strings $\underline{x} = (x_{n-1}, \dots, x_1, x_0)$. Then, you take $n = 2^r - 1$ so that you can assign to each index $j = 0, \dots, n$ an r -bit binary string, $j = (j_{r-1}, \dots, j_0)$, with $j_l = 0, 1$, and $l = 0, \dots, r - 1$. Finally, you define the r check-sum constraints:

$$\mu_l(\underline{x}) = \sum_{j|j_l=1}^{n-1} x_j \pmod{2}.$$

The set of binary strings satisfying all the constraints defines the "codewords". For instance, for $H(7, 4)$, where $r = 3$ and $n = 2^3 - 1 = 7$ you require the codewords to be binary strings satisfying:

$$\begin{array}{cccccc} x_{111}+ & & x_{101}+ & & x_{011}+ & & x_{001} & = \mu_0(\underline{x}) = 0 \pmod{2} \\ x_{111}+ & x_{110}+ & & & x_{011}+ & x_{010} & & = \mu_1(\underline{x}) = 0 \pmod{2} \\ x_{111}+ & x_{110}+ & x_{101}+ & x_{100}+ & & & & = \mu_2(\underline{x}) = 0 \pmod{2} \end{array} .$$

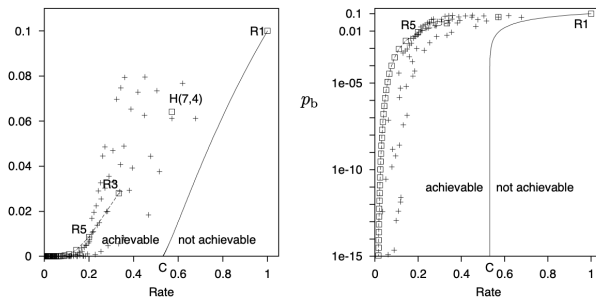


Figure 11.2: The phase diagram of the probability of decoding error p_b versus the transmission rate R . R_3 denotes the repetition code R_3 we have discussed. $H(7,4)$ denotes a different code, the (7,4) Hamming code, the simplest example of a block code using *parity checks* bits: 3 parity check bits every 4 source bits. C denotes the channel capacity, discovered by Shannon. Figure taken from MacKay (Fig.1.19).

11.2. Quantum error correction: the simple case of bit flips

As Mermin’s quote suggests, it is far from clear, at the outset — and, indeed, it was unclear until Peter Shor came out with a brilliant idea in 1996 — that quantum error correction is possible at all. Various difficulties come to mind:

- 1) The *no-cloning theorem* forbids from making “copies” of arbitrary single-Qbit quantum states $|\psi\rangle$, to mimick the classical repetition-code idea.
- 2) While classically only bit-flip errors have to be considered, quantum mechanically you have to consider phase errors, for instance, $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, or, more generally, small unitary errors like

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \rightarrow \cos \frac{\theta+\delta\theta}{2} |0\rangle + e^{i(\phi+\delta\phi)} \sin \frac{\theta+\delta\theta}{2} |1\rangle .$$

- 3) Every time you attempt to “measure” if an error has occurred, you should be aware of the collapse of the wavefunction, which might ruin any quantum superposition.

We start with the simplest example of quantum error correction, which shows some of the remarkable facts behind this topic. The example discussed is a toy model since it shows how to correct only a very simple type of error: single-bit flips. It does so by exploiting, essentially, the R_3 classical repetition code.

Encoding. So, let us start encoding, like in the classical case, a bit into three identical bits. However, the non-cloning theorem forbids us from constructing an encoding machine that does $|\psi\rangle \rightarrow |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$. The idea is, on the contrary, to encode $|0\rangle \rightarrow |0_L\rangle = |000\rangle$, $|1\rangle \rightarrow |1_L\rangle = |111\rangle$, so that a general Qbit is encoded, by the superposition principle, as:

$$|\psi\rangle = z_0|0\rangle + z_1|1\rangle \rightarrow |\psi_L\rangle = z_0|000\rangle + z_1|111\rangle . \tag{11.4}$$

We define the **code Hilbert space** \mathcal{H}_C as:

$$|\psi_L\rangle \in \mathcal{H}_C = \text{span}\{|000\rangle, |111\rangle\} . \tag{11.5}$$

Physical vs Logical bits. The subscript L here reminds us that a *logical* bit (or Qbit) is different from a *physical* bit: in the present case, 3 physical Qbits are used to encode a single logical Qbit.

One can easily construct a small quantum circuit which does precisely this encoding, using two CNOT gates, see Fig. 11.3.

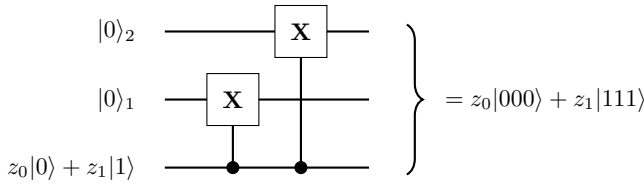


Figure 11.3: The 3-Qbit encoding. Two CNOT are used to transform a product state $|0\rangle_2 \otimes |0\rangle_1 \otimes (z_0|0\rangle_0 + z_1|1\rangle_0)$ into a 3-Qbit entangled state $z_0|000\rangle + z_1|111\rangle$.

A simple model of noise: rare bit flips.

Let us now assume our extremely simplified model of error occurring on the encoded logical Qbit: *at most one* of the three bits can suffer from the presence of an unwanted bit flip operator \mathbf{X} . If $|\psi_L\rangle$ denotes the encoded Qbit, there could be three corruptions occurring:

$$\begin{aligned} |\psi^{(0)}\rangle &= \mathbf{X}_0|\psi_L\rangle = z_0|001\rangle + z_1|110\rangle \in \text{span}\{|001\rangle, |110\rangle\} = \mathcal{H}_0 \\ |\psi^{(1)}\rangle &= \mathbf{X}_1|\psi_L\rangle = z_0|010\rangle + z_1|101\rangle \in \text{span}\{|010\rangle, |101\rangle\} = \mathcal{H}_1 \\ |\psi^{(2)}\rangle &= \mathbf{X}_2|\psi_L\rangle = z_0|100\rangle + z_1|011\rangle \in \text{span}\{|100\rangle, |011\rangle\} = \mathcal{H}_2 \end{aligned} \tag{11.6}$$

Notice that the full Hilbert space \mathcal{H} for $n = 3$ spins has dimension $2^3 = 8$, and is given by the direct sum of the various

$$\mathcal{H} = \mathcal{H}_C \oplus \mathcal{H}_0 \oplus \mathcal{H}_1 \oplus \mathcal{H}_2 . \tag{11.7}$$

We take therefore the state after encoding and corruption to be given by:

$$|\psi_L^{\text{err}}\rangle = \underbrace{\left(\sqrt{1-3p}\mathbf{1} + i\sqrt{p}(\mathbf{X}_0 + \mathbf{X}_1 + \mathbf{X}_2) \right)}_{\hat{E}} |\psi_L\rangle = \hat{E}|\psi_L\rangle , \tag{11.8}$$

where the i might be safely omitted, as it is largely irrelevant here. You might wonder about normalisation of the state: the state is normalized because

$$\langle \psi_L | \mathbf{X}_j | \psi_L \rangle = 0 \quad \text{and} \quad \langle \psi_L | \mathbf{X}_j \mathbf{X}_{j'} | \psi_L \rangle = 0 \quad \text{for} \quad j' \neq j .$$

However, strictly speaking the operator \hat{E} applied to the state $|\psi_L\rangle$ is *not* a unitary 3-Qbit operator: you only have that $\langle \psi_L | \hat{E}^\dagger \hat{E} | \psi_L \rangle = 1$, which is enough to guarantee normalization. One might wonder how to write a “small” unitary operator acting on the three-Qbit state. There are many possibilities, and a quite natural one is to assume that each Qbit is acted independently by a small unitary $\mathbf{U}_j = \sqrt{1-p}\mathbf{1}_j + i\sqrt{p}\mathbf{X}_j$, so that we have

$$\begin{aligned} \mathbf{U} = \mathbf{U}_0 \mathbf{U}_1 \mathbf{U}_2 &= \sqrt{(1-p)^3}\mathbf{1} + i\sqrt{p}(1-p)(\mathbf{X}_0 + \mathbf{X}_1 + \mathbf{X}_2) \\ &\quad - p\sqrt{1-p}(\mathbf{X}_0\mathbf{X}_1 + \mathbf{X}_0\mathbf{X}_2 + \mathbf{X}_1\mathbf{X}_2) - i\sqrt{p^3}\mathbf{X}_0\mathbf{X}_1\mathbf{X}_2 . \end{aligned}$$

Detecting (rare) bit flips without losing coherence.

The issue is now how the get information on the corrupted state without a direct measurement of the logical Qbits, which would destroy the superposition. The idea is to measure information on *correlations* among the Qbits, in particular about the presence of pairs of *parallel spins*. We can do that with CNOT gates, by entangling the Qbits with an ancillary Qbit, as shown in Fig. 11.4.

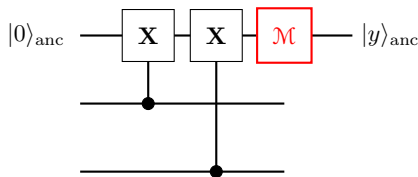


Figure 11.4: Detecting information on two Qbits with an ancillary Qbit. If $y = 0$ upon measuring the ancillary Qbit, then the two Qbits are guaranteed to be in $|00\rangle$, $|11\rangle$ or *any linear combination* of such two states. Incidentally, notice that this is precisely the bitwise-modulo-2 sum used in the quantum adder.

Quite clearly, if we measure $y = 1$ in the ancillary Qbit, then the two Qbits must have components of the form $|01\rangle$ or $|10\rangle$. If we perform a similar measurement on both Qbits 0-1 (through the first

ancilla $|y\rangle_1$) and Qbits 1-2 (through a second ancilla $|y\rangle_2$) we can easily deduce, from these two measurements, *where* the bit flip occurred, and correct it. Figure 11.5 illustrates the whole process, including the encoding, the possible error, the error detection — known as *syndrome* — and the final correction, through an appropriate **X**.

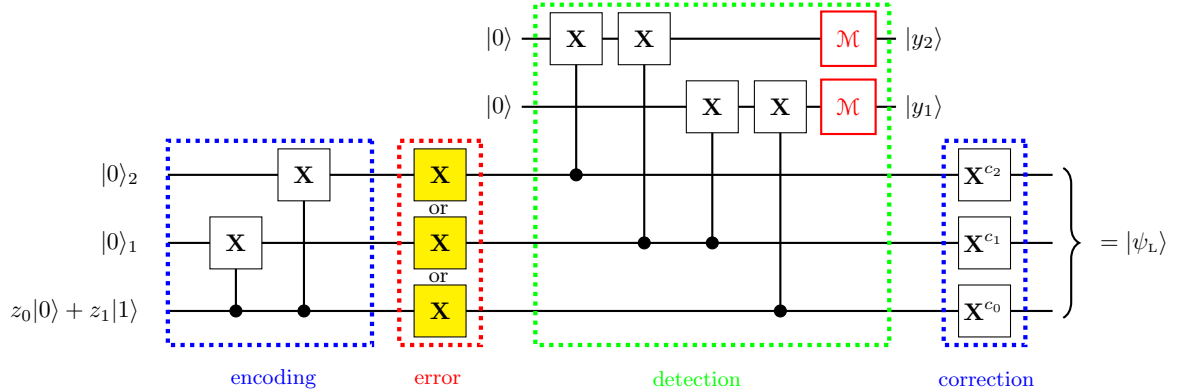


Figure 11.5.: The 3-Qbit encoding (left), followed by error (at most one bit flip), followed by a circuit that detects the presence of error by making measurements on two supplementary ancillary Qbits, with results y_0 and y_1 . The final stage is the coherence-preserving correction of the detected error by appropriate controlled Qbit flips. Here the control bits are — as you can easily verify — to be set to $c_0 = (1 - y_2)y_1$, $c_1 = y_2y_1$, $c_2 = y_2(1 - y_1)$, depending on the results y_1 and y_2 of the measurements on the two ancillary Qbits.

11.3. Measuring error syndromes: general idea

There is an interesting reformulation of the syndrome detection which will be useful later on. It starts from the circuit equivalence shown in Fig. 3.7, which we report here again. Using this, we can

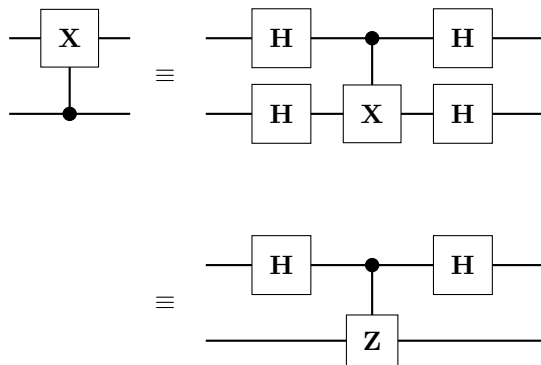


Figure 11.6: The identity in Eq. (3.37), illustrating how to exchange control- and target-Qbit by a sandwich with **H** on both lines. The second form (below) comes from observing that $\mathbf{HXH} = \mathbf{Z}$.

equivalently rewrite the circuit in Fig. 11.4 to measure the correlations between two spins as shown in Fig. 11.7.

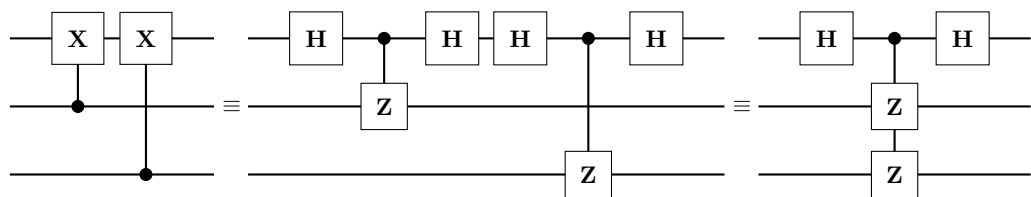


Figure 11.7.: Transforming the circuit to detect information on two Qbits with an ancillary Qbit.

This reformulation deserves a few general remarks, which will prove useful later on, when discussing more general “syndrome detections”. Consider a multi-Qbit controlled unitary operator, where the control Qbit is a single ancilla. Let $\mathbf{C}_{n,a}^{\mathbf{U}}$ denote this controlled-unitary, hence, by definition:

$$\mathbf{C}_{n,a}^{\mathbf{U}} |0\rangle_a \otimes |\psi\rangle_n = |0\rangle_a \otimes |\psi\rangle_n \quad \mathbf{C}_{n,a}^{\mathbf{U}} |1\rangle_a \otimes |\psi\rangle_n = |1\rangle_a \otimes \mathbf{U}|\psi\rangle_n. \quad (11.9)$$

Hence, by sandwiching the ancilla between two Hadamards, we find:

$$\begin{aligned} (\mathbf{H} \times \mathbf{1}) \mathbf{C}_{n,a}^{\mathbf{U}} (\mathbf{H} \times \mathbf{1}) |0\rangle_a \otimes |\psi\rangle_n &= (\mathbf{H} \times \mathbf{1}) \mathbf{C}_{n,a}^{\mathbf{U}} \frac{1}{\sqrt{2}} (|0\rangle_a + |1\rangle_a) \otimes |\psi\rangle_n \\ &= (\mathbf{H} \times \mathbf{1}) \frac{1}{\sqrt{2}} (|0\rangle_a \otimes |\psi\rangle_n + |1\rangle_a \otimes \mathbf{U}|\psi\rangle_n) \\ &= \frac{1}{2} (|0\rangle_a + |1\rangle_a) \otimes |\psi\rangle_n + \frac{1}{2} (|0\rangle_a - |1\rangle_a) \otimes \mathbf{U}|\psi\rangle_n \\ &= |0\rangle_a \otimes \frac{1}{2} (\mathbf{1} + \mathbf{U})|\psi\rangle_n + |1\rangle_a \otimes \frac{1}{2} (\mathbf{1} - \mathbf{U})|\psi\rangle_n. \end{aligned} \quad (11.10)$$

Particularly relevant, for our discussion, is the case in which the unitary \mathbf{U} squares to the identity,

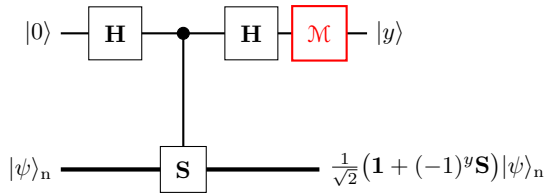


Figure 11.8: Measuring an Hermitean operator \mathbf{S} such that $\mathbf{S}^2 = \mathbf{1}$. The thick lower line denotes an n -Qbit state $|\psi\rangle_n$. If $y = 0$ is measured for the ancilla, the state collapses into an eigenstate with $\lambda = +1$ for \mathbf{S} , and vice-versa for $y = 1$, the state collapses into an eigenstate with $\lambda = -1$. The two measurement outcomes have equal probability $\frac{1}{2}$.

$\mathbf{U}^2 = \mathbf{1}$, which immediately implies that it must be also *Hermitean*. Hence, taking $\mathbf{U} = \mathbf{S}$, with $\mathbf{S}^\dagger = \mathbf{S}$ and $\mathbf{S}^2 = \mathbf{1}$, we know that \mathbf{S} can have only eigenvalues $+1$ or -1 , and

$$\frac{1}{2} (\mathbf{1} \pm \mathbf{S}) = \widehat{\Pi}_\pm^{\mathbf{S}} \quad (11.11)$$

are projector operators on the subspace with eigenvalue ± 1 .

i

Hermitean \mathbf{S} which squares to the identity. If $\mathbf{S} = \mathbf{S}^\dagger$ and $\mathbf{S}^2 = \mathbf{1}$, then:

$$(\mathbf{H} \times \mathbf{1}) \mathbf{C}_{n,a}^{\mathbf{S}} (\mathbf{H} \times \mathbf{1}) |0\rangle_a \otimes |\psi\rangle_n = |0\rangle_a \otimes \widehat{\Pi}_+^{\mathbf{S}} |\psi\rangle_n + |1\rangle_a \otimes \widehat{\Pi}_-^{\mathbf{S}} |\psi\rangle_n. \quad (11.12)$$

Hence, upon measuring the ancilla, see Fig. 11.8, obtaining 0, one is guaranteed that the system is in an eigenstate of \mathbf{S} with eigenvalue $+1$, while if 1 is obtained, the system is in an eigenstate of \mathbf{S} with eigenvalue -1 . This idea can be **generalised** by considering a set of **commuting operators** $\mathbf{S}_1, \mathbf{S}_2, \dots$ which square to $\mathbf{1}$. Since they commute, they have common eigenstates, which can be classified by measuring an appropriate ancilla for each of the \mathbf{S}_j . The eigenvalues of these commuting operators are called **syndromes**, because an eigenvalue -1 can diagnose the presence of an error.

To conclude, we rewrite here the syndrome detection part of the 3-Qbit encoding, as shown in Fig. 11.9. Observe that the detection part effectively involve measuring the two commuting “syndrome detectors” $\mathbf{S}_1 = \mathbf{Z}_0 \mathbf{Z}_1$ and $\mathbf{S}_2 = \mathbf{Z}_1 \mathbf{Z}_2$. The results of the measurements of the two ancillas associated to them unambiguously tell us which bit-flip operators we need to apply to correct the state. More about this device of measuring commuting Hermitean operators which “square to $\mathbf{1}$ ” in the following.

Automating the error correction. It is worth remarking that the measurement part of the circuit might be implemented as a unitary “automatic correction”, by using (multi)-controlled cNOTs and avoiding measurements [1]. This is shown in Fig. 11.10.

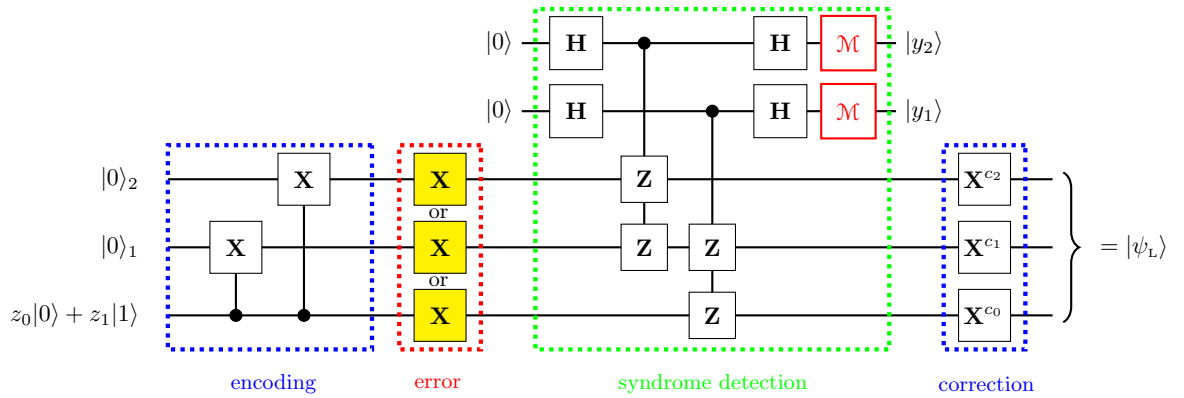


Figure 11.9.: Alternative syndrome formulation. The 3-Qbit encoding (left), followed by error (at most one bit flip), followed by a circuit that detects the presence of error by making measurements on two supplementary ancillary Qbits, with results y_0 and y_1 . The final stage is the coherence-preserving correction of the detected error by appropriate controlled Qbit flips. Here the control bits are — as you can easily verify — to be set to $c_0 = (1 - y_2)y_1$, $c_1 = y_2y_1$, $c_2 = y_2(1 - y_1)$, depending on the results y_1 and y_2 of the measurements on the two ancillary Qbits.

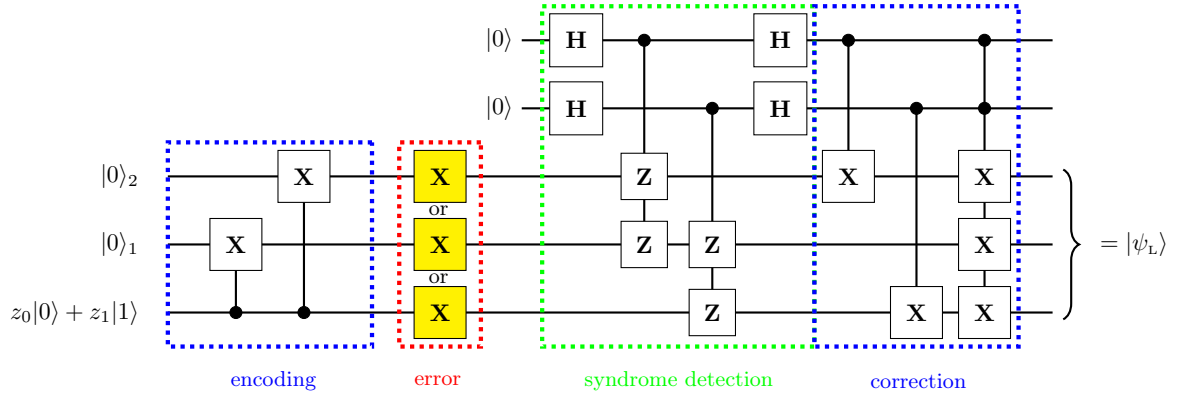


Figure 11.10.: Error correction without measurements. The 3-Qbit encoding (left), followed by error (at most one bit flip), followed by a circuit that detects the presence of error with two supplementary ancillary Qbits. The final stage is the coherence-preserving error correction by appropriate (multi)-controlled cNOTs.

11.4. More general errors: error digitisation

This section is mostly based on Ref. [1], but a good reading is also Ref. [3][Sec. 10.3.2]. Let me start with the simple case of small unitary (coherent) errors for a single Qbit. You can think of having a 2×2 unitary matrix \mathbf{u} , close to the identity,

$$\mathbf{u} = \mathbf{1} + O(\epsilon) = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix},$$

such that the unitary error acts as $|\psi\rangle \rightarrow \mathbf{u}|\psi\rangle$. On the computational basis we then have:

$$\begin{aligned} |0\rangle &\xrightarrow{\mathbf{u}} u_{00}|0\rangle + u_{01}|1\rangle \\ |1\rangle &\xrightarrow{\mathbf{u}} u_{10}|0\rangle + u_{11}|1\rangle \end{aligned} \tag{11.13}$$

It is easy to verify that if $|x\rangle$ denotes an element of the Qbit computational basis (with $x = 0, 1$) you can combine the two previous equations as:

$$|x\rangle \xrightarrow{\mathbf{u}} \left(\frac{u_{00} + u_{11}}{2} \mathbf{1} + \frac{u_{00} - u_{11}}{2} \mathbf{Z} + \frac{u_{01} + u_{10}}{2} \mathbf{X} + \frac{u_{01} - u_{10}}{2i} \mathbf{Y} \right) |x\rangle. \tag{11.14}$$

Being linear, this implies that small unitary errors on any single Qbit state can always be “expanded” into the effect of the identity (the largest part) plus small contributions from the other three Pauli matrices:

$$|\psi\rangle \rightarrow \mathbf{u}|\psi\rangle = (\alpha_0 \mathbf{1} + \alpha_1 \mathbf{X} + \alpha_2 \mathbf{Y} + \alpha_3 \mathbf{Z})|\psi\rangle, \quad (11.15)$$

with appropriate complex coefficients α_j , to guarantee normalisation.

i

Digitisation. Remarkably, the action of any unitary coherent error (a continuum of them) on a single Qbit can be expanded in terms of a *discrete* subset: bit flip errors (\mathbf{X}), phase-errors (\mathbf{Z}) and a combination of bit-and-phase error (\mathbf{Y}). This is, after all, no surprise, since any 2×2 unitary matrix can be expanded in terms of the identity and the three Pauli matrices — all of them being *unitary* as well as Hermitian — by using appropriate *complex* coefficients.

What about errors that involve the entangling effect with other degrees of freedom, an “environment”, for the Qbit under consideration? Suppose that Qbit and environment are originally decoupled, and that a small entangling interaction — officially unwanted, and represented by some unitary evolution \hat{U}^{tot} — acts. The computational basis states are changed as follows:

$$\begin{aligned} |0\rangle \otimes |\phi^{\text{B}}\rangle &\xrightarrow{\hat{U}^{\text{tot}}} |0\rangle \otimes |\chi_{00}^{\text{B}}\rangle + |1\rangle \otimes |\chi_{01}^{\text{B}}\rangle \\ |1\rangle \otimes |\phi^{\text{B}}\rangle &\xrightarrow{\hat{U}^{\text{tot}}} |0\rangle \otimes |\chi_{10}^{\text{B}}\rangle + |1\rangle \otimes |\chi_{11}^{\text{B}}\rangle \end{aligned}, \quad (11.16)$$

where the environment states $|\chi_{jj'}^{\text{B}}\rangle$ are in general neither normalized, nor orthogonal.⁴ Once again, for a general computational basis state $|x\rangle$ (with $x = 0, 1$) we can rearrange these two expressions as:

$$\left\{ \begin{array}{l} |x\rangle \otimes |\phi^{\text{B}}\rangle \xrightarrow{\hat{U}^{\text{tot}}} \mathbf{1}|x\rangle \otimes |\chi_0^{\text{B}}\rangle + \mathbf{X}|x\rangle \otimes |\chi_1^{\text{B}}\rangle + \mathbf{Y}|x\rangle \otimes |\chi_2^{\text{B}}\rangle + \mathbf{Z}|x\rangle \otimes |\chi_3^{\text{B}}\rangle \\ \text{with} \\ |\chi_0^{\text{B}}\rangle = \frac{1}{2}(|\chi_{00}^{\text{B}}\rangle + |\chi_{11}^{\text{B}}\rangle), \quad |\chi_3^{\text{B}}\rangle = \frac{1}{2}(|\chi_{00}^{\text{B}}\rangle - |\chi_{11}^{\text{B}}\rangle) \\ |\chi_1^{\text{B}}\rangle = \frac{1}{2}(|\chi_{01}^{\text{B}}\rangle + |\chi_{10}^{\text{B}}\rangle), \quad |\chi_2^{\text{B}}\rangle = \frac{1}{2i}(|\chi_{01}^{\text{B}}\rangle - |\chi_{10}^{\text{B}}\rangle) \end{array} \right. . \quad (11.17)$$

By linearity, for a general state $|\psi\rangle = z_0|0\rangle + z_1|1\rangle$ we get:

$$|\psi\rangle \otimes |\phi^{\text{B}}\rangle \rightarrow \hat{U}^{\text{tot}}|\psi\rangle \otimes |\phi^{\text{B}}\rangle = \left(\mathbf{1}|\psi\rangle \otimes |\chi_0^{\text{B}}\rangle + \mathbf{X}|\psi\rangle \otimes |\chi_1^{\text{B}}\rangle + \mathbf{Y}|\psi\rangle \otimes |\chi_2^{\text{B}}\rangle + \mathbf{Z}|\psi\rangle \otimes |\chi_3^{\text{B}}\rangle \right). \quad (11.18)$$

Notice that the algebra to get to this form is identical to unitary coherent case, which is recovered by taking $|\chi_\mu^{\text{B}}\rangle = \alpha_\mu |\phi^{\text{B}}\rangle$ for $\mu = 0, \dots, 3$, so that no entanglement results. Just be aware of the fact that the environment states are *neither normalised nor orthogonal*, as opposed to the basis we used in writing the Kraus representation, although in general they can be assumed to be *linearly independent*. You can always insist on a Kraus form, by rewriting the bath states in terms of a proper orthonormal set $\{|\phi_k^{\text{B}}\rangle\}$ with an invertible matrix $\mathbb{A}_{k,\mu}$:

$$|\chi_\mu^{\text{B}}\rangle = \sum_{k=1}^4 \mathbb{A}_{k,\mu} |\phi_k^{\text{B}}\rangle.$$

⁴Unitarity requires:

$$\langle \chi_{00}^{\text{B}} | \chi_{00}^{\text{B}} \rangle + \langle \chi_{01}^{\text{B}} | \chi_{01}^{\text{B}} \rangle = 1, \quad \langle \chi_{10}^{\text{B}} | \chi_{10}^{\text{B}} \rangle + \langle \chi_{11}^{\text{B}} | \chi_{11}^{\text{B}} \rangle = 1, \quad \langle \chi_{10}^{\text{B}} | \chi_{00}^{\text{B}} \rangle + \langle \chi_{11}^{\text{B}} | \chi_{01}^{\text{B}} \rangle = 0.$$

1 An environment with 4 states is enough. The orthonormal basis of the bath is so chosen that the first 4 elements are obtained by a Gram-Schmidt orthogonalization of the 4 states $|\chi_\mu^B\rangle$, while the remaining states are arbitrary. This shows that a bath with only 4 states is enough for the purpose of writing the dynamics.

Hence, you can rewrite:

$$\begin{aligned} |\psi\rangle \otimes |\phi^B\rangle \rightarrow \widehat{U}^{\text{tot}} |\psi\rangle \otimes |\phi^B\rangle &= \sum_{\mu=0}^3 \hat{\sigma}^{(\mu)} |\psi\rangle \otimes |\chi_\mu^B\rangle = \sum_{k=1}^4 \overbrace{\left(\sum_{\mu=0}^3 \mathbb{A}_{k,\mu} \hat{\sigma}^{(\mu)} \right)}^{\widehat{E}_k} |\psi\rangle \otimes |\phi_k^B\rangle \\ &= \sum_{k=1}^4 \widehat{E}_k |\psi\rangle \otimes |\phi_k^B\rangle, \end{aligned} \quad (11.19)$$

where you realise that at most 4 independent Kraus operators \widehat{E}_k need to be invoked and, to simplify our notation, we redefined the identity and Pauli operators by using an index:

$$\hat{\sigma}^{(\mu)} = \begin{cases} \mathbf{1} & \text{for } \mu = 0 \\ \mathbf{X} & \text{for } \mu = 1 \\ \mathbf{Y} & \text{for } \mu = 2 \\ \mathbf{Z} & \text{for } \mu = 3 \end{cases}. \quad (11.20)$$

In Kraus form, therefore, the most general error that a single Qbit can suffer can always be written as:

$$\hat{\rho}_C \xrightarrow{\widehat{U}^{\text{tot}}} \sum_{k=1}^4 \widehat{E}_k \hat{\rho}_C \widehat{E}_k^\dagger, \quad (11.21)$$

where $\hat{\rho}_C$ is a general state in the code Hilbert space \mathcal{H}_C .

Let us now consider the case of n Qbits. The entangling interaction starting from the initial state $|\psi\rangle \otimes |\phi^B\rangle$ brings to a generally corrupted and entangled superposition with 4^n terms:

$$|\psi\rangle \otimes |\phi^B\rangle \xrightarrow{\widehat{U}^{\text{tot}}} \widehat{U}^{\text{tot}} |\psi\rangle \otimes |\phi^B\rangle = \sum_{\mu_1=0}^3 \dots \sum_{\mu_n=0}^3 \hat{\sigma}_1^{(\mu_1)} \hat{\sigma}_2^{(\mu_2)} \dots \hat{\sigma}_n^{(\mu_n)} |\psi\rangle \otimes |\chi_{\mu_1 \dots \mu_n}^B\rangle. \quad (11.22)$$

The general Kraus form would be, once again:

$$\hat{\rho}_C \xrightarrow{\widehat{U}^{\text{tot}}} \mathcal{E}(\hat{\rho}_C) = \sum_{k=1}^{4^n} \widehat{E}_k \hat{\rho}_C \widehat{E}_k^\dagger, \quad (11.23)$$

where the Kraus operators are linear combinations of Pauli string operators:

$$\widehat{E}_k = \sum_{\mu_1=0}^3 \dots \sum_{\mu_n=0}^3 \mathbb{A}_{k,\mu_1 \dots \mu_n} \hat{\sigma}_1^{(\mu_1)} \hat{\sigma}_2^{(\mu_2)} \dots \hat{\sigma}_n^{(\mu_n)}. \quad (11.24)$$

Suppose that n is the number of physical Qbits encoding each logical Qbit. And suppose that your hardware is good enough that the probability of corruption of the codewords is small, hence the terms differing from the uncorrupted codeword $|\psi\rangle$ — that involving only identities, hence all $\mu_j = 0$, in the previous equation — are dominated by the errors involving a *single physical Qbit*. Disregarding all terms affecting more than one Qbit, we would therefore write an expression involving $3n + 1$ terms:

$$|\psi\rangle \otimes |\phi^B\rangle \xrightarrow{\text{unitary}} \widehat{U}^{\text{tot}} |\psi\rangle \otimes |\phi^B\rangle \approx \left(|\psi\rangle \otimes |\chi_{00 \dots 0}^B\rangle + \sum_{j=0}^{n-1} \sum_{\mu_j=1}^3 \hat{\sigma}_j^{(\mu_j)} |\psi\rangle \otimes |\chi_{0 \dots 0 \mu_j 0 \dots 0}^B\rangle \right), \quad (11.25)$$

corresponding to the uncorrupted codewords, plus a possible corruption due to action of a single Pauli operator — either \mathbf{X} , \mathbf{Y} or \mathbf{Z} — in each of the n physical Qbits. In Kraus form, we now have:

$$\hat{\rho}_C \xrightarrow[\approx]{\text{single Qbit errors}} \mathcal{E}_t(\hat{\rho}_C) = \sum_{k=1}^{3n+1} \hat{E}_k \hat{\rho}_C \hat{E}_k^\dagger, \quad (11.26)$$

with the only caveat that \mathcal{E}_t is a truncated map, hence **not trace preserving**, in general.

As a particular example of a noise process describing independent errors occurring on the various Qbits, we can consider the case where each Qbit is affected by a depolarising map, see Sec. 9.7.3 and Eq. 9.123, for instance:

$$\mathcal{E}^{\text{dep}}(\hat{\rho}_{1\text{-Qbit}}) = (1-p)\hat{\rho}_{1\text{-Qbit}} + \frac{p}{3} \sum_{\mu=1}^3 \hat{\sigma}^{(\mu)} \hat{\rho}_{1\text{-Qbit}} \hat{\sigma}^{(\mu)}. \quad (11.27)$$

Evidently the combined map where the depolarising map acts on each Qbit independently can be written as:

$$\mathcal{E}(\hat{\rho}_C) = \underbrace{(1-p)^n \hat{\rho}_C + (1-p)^{n-1} \frac{p}{3} \sum_{j=0}^{n-1} \sum_{\mu_j=1}^3 \hat{\sigma}_j^{(\mu_j)} \hat{\rho}_C \hat{\sigma}_j^{(\mu_j)} + \dots}_{\mathcal{E}_t(\hat{\rho}_C)}, \quad (11.28)$$

where the dots represent terms with two or more Pauli operators acting independently on various Qbits. Evidently, we can take:

$$\hat{E}_1 = \sqrt{(p-1)^n} \mathbf{1} \quad \hat{E}_{k>1} = \sqrt{(1-p)^{n-1} p/3} \hat{\sigma}_j^{(\mu_j)} \quad \text{with} \quad \mu_j = 1, 2, 3.$$

Hamming bound for single logical-Qbit encoding. This form is suggestive. Suppose that we use n physical Qbits to code a single logical Qbit, with the uncorrupted state $|\psi_L\rangle$ being a combination of the two logical computational states $\{|0_L\rangle, |1_L\rangle\}$ (more details later on how these are expressed).⁵ Out of the 2^n possible states of the n physical Qbits, the uncorrupted codewords form a 2-dimensional subspace \mathcal{H}_C , with elements $z_0|0_L\rangle + z_1|1_L\rangle$. Next, we need to clearly identify *orthogonal* 2-dimensional subspaces corresponding to each of the possible corruptions $\hat{\sigma}_j^{(\mu_j)}|\psi_L\rangle$, to be able to *diagnose the error* with some appropriate ancilla measurement, and then correct it. The total n Qbit physical space must therefore be large enough to allow for $3n+1$ two-dimensional orthogonal subspaces, hence:

$$2^n \geq 2(3n+1) \quad \implies \quad 2^{n-1} \geq (3n+1). \quad (11.29)$$

The minimal n that satisfies this equation is $n=5$, which is therefore the minimal number of physical Qbits that allows us to detect (and correct) single-Qbit errors in either of the three Pauli directions. Incidentally, a similar reasoning applied to the case in which *only bit flip errors* are possible, suggests that we would have to satisfy $2^n \geq 2(n+1)$, hence $2^{n-1} \geq (n+1)$: the minimal n to do that is $n=3$. This was indeed our initial toy exercise with bit flip errors only.

11.5. The five-Qbit encoding

Let us now consider in more detail an $n=5$ Qbit encoding, which is able to detect and correct errors in either of the three Pauli directions for all the physical Qbits. We need to distinguish $1+3 \times 5 = 16$ two-dimensional subspaces, corresponding to the uncorrupted and to any of the possible corruptions, in the $2^5 = 32$ dimensional total Hilbert space. The idea is to introduce – building them in terms of Pauli operators for the physical Qbits — 4 mutually commuting Hermitean operators which *square to the identity*, hence have eigenvalues ± 1 and can be measured simultaneously, to discriminate the different $2^4 = 16$ subspaces and realise if an error occurred or not.

⁵In general, one might code k logical Qbits into $n > k$ physical Qbits.

1 Stabilizers. A set of n_s independent ^a operators $\langle \mathbf{S}_1, \dots, \mathbf{S}_{n_s} \rangle$ build out of the physical Qbits Pauli matrices, Hermitean and squaring to the identity, which are mutually *commuting* is known as the *stabilizers* of the (error correction) code.

^aTechnically, see later, the *generators* of the stabilizer group.

For $n = 5$ we need $n_s = 4$ stabilizers, having $2^4 = 16$ different eigenvalues, all of the form ± 1 . Consider the following:

$$\begin{cases} \mathbf{S}_1 = \mathbf{Z}_1 \mathbf{X}_2 \mathbf{X}_3 \mathbf{Z}_4 \\ \mathbf{S}_2 = \mathbf{Z}_2 \mathbf{X}_3 \mathbf{X}_4 \mathbf{Z}_0 \\ \mathbf{S}_3 = \mathbf{Z}_3 \mathbf{X}_4 \mathbf{X}_0 \mathbf{Z}_1 \\ \mathbf{S}_4 = \mathbf{Z}_4 \mathbf{X}_0 \mathbf{X}_1 \mathbf{Z}_2 \end{cases} \quad (11.30)$$

It is clear that \mathbf{S}_j are Hermitean and $\mathbf{S}_j^2 = \mathbf{1}$. It is also easy to realise that they are *mutually commuting*. For instance, consider $\mathbf{S}_1 = \mathbf{Z}_1 \mathbf{X}_2 \mathbf{X}_3 \mathbf{Z}_4$ and $\mathbf{S}_2 = \mathbf{Z}_2 \mathbf{X}_3 \mathbf{X}_4 \mathbf{Z}_0$, where we have highlighted with colors Pauli operators that do not commute: indeed, recall that $\mathbf{ZX} = -\mathbf{XZ}$. But, by construction, these anticommutation minus signs always *appear in pairs*. Hence $\mathbf{S}_1 \mathbf{S}_2 = \mathbf{S}_2 \mathbf{S}_1$. Similar reasoning apply to all other pairs you might consider.

1 Is that all? The four stabilizers have a clear cyclic pattern \mathbf{ZXXZ} , with a fifth possibility which we omitted:

$$\mathbf{S}_5 = \mathbf{Z}_0 \mathbf{X}_1 \mathbf{X}_2 \mathbf{Z}_3 .$$

Why? It is clear that if you multiply all of them you get the identity, because each Pauli matrix appears squared, hence the fifth “stabilizer” is not really independent:

$$\mathbf{S}_1 \mathbf{S}_2 \mathbf{S}_3 \mathbf{S}_4 \mathbf{S}_5 = \mathbf{1} \quad \implies \quad \mathbf{S}_5 = \mathbf{S}_1 \mathbf{S}_2 \mathbf{S}_3 \mathbf{S}_4 .$$

Here are the codewords for $|0_L\rangle$ and $|1_L\rangle$, directly written in terms of the stabilizers:

1 Coding with 5 Qbits.

$$\begin{aligned} |0_L\rangle &= \frac{1}{4} (\mathbf{1} + \mathbf{S}_1) (\mathbf{1} + \mathbf{S}_2) (\mathbf{1} + \mathbf{S}_3) (\mathbf{1} + \mathbf{S}_4) |00000\rangle \\ |1_L\rangle &= \frac{1}{4} (\mathbf{1} + \mathbf{S}_1) (\mathbf{1} + \mathbf{S}_2) (\mathbf{1} + \mathbf{S}_3) (\mathbf{1} + \mathbf{S}_4) |11111\rangle \end{aligned} \quad (11.31)$$

Since each \mathbf{S}_j flips two spins, $|0_L\rangle$ is a superposition of computational basis states with an *even* number of 1, and $|1_L\rangle$ a superposition with an even number of 0. Consequently, the two codewords are orthogonal $\langle 1_L | 0_L \rangle = 0$. For the actual circuit by which you can encode the logical 5-Qbit states, see Mermin [1][Sec. 5.9].

Verifying that these states are normalised is very simple. ⁶ Moreover, since:

$$\mathbf{S}_j (\mathbf{1} + \mathbf{S}_j) = \mathbf{1} + \mathbf{S}_j ,$$

it is clear that $|0_L\rangle, |1_L\rangle$ as well as any state in the two-dimensional uncorrupted logical subspace

$$|\psi_L\rangle = z_0 |0_L\rangle + z_1 |1_L\rangle ,$$

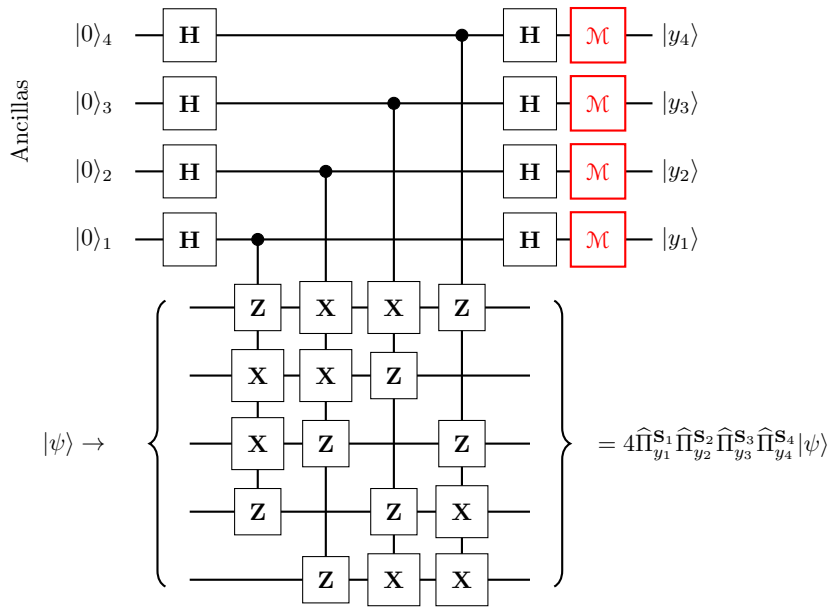


Figure 11.11.: The syndrome measurement for the 5-Qbits code. Here $|\psi\rangle = \hat{\sigma}_j^{\mu_j} |\psi_L\rangle$ is one of the possible corrupted states.

are eigenstates of all the stabilizers with eigenvalue +1. Here is a table of all possible eigenvalues of the stabilizers (16 possibilities) with a corresponding signature for the eigenvalues of the corrupted state $\hat{\sigma}_j^{\mu_j} |\psi_L\rangle$. The idea behind is very simple: some stabilizer will *anticommute* with the single Pauli matrix for the corrupted state, hence you *detect* the error by simply measuring together all stabilizers and looking at the table below.

	1	X ₀	Y ₀	Z ₀	X ₁	Y ₁	Z ₁	X ₂	Y ₂	Z ₂	X ₃	Y ₃	Z ₃	X ₄	Y ₄	Z ₄
S ₁ = Z ₁ X ₂ X ₃ Z ₄	+	+	+	+	-	-	+	+	-	-	+	-	-	-	-	+
S ₂ = Z ₂ X ₃ X ₄ Z ₀	+	-	-	+	+	+	+	-	-	+	+	-	-	+	-	-
S ₃ = Z ₃ X ₄ X ₀ Z ₁	+	+	-	-	+	-	+	+	+	+	-	-	+	+	-	-
S ₄ = Z ₄ X ₀ X ₁ Z ₂	+	+	-	-	+	-	-	-	-	+	+	+	+	-	-	+

Figure 11.11 shows a circuit that reveals the error, the so called *syndrome*, by measuring all 4 stabilizers with associated ancilla Qbits.

There is still one non-trivial aspect of the story to discuss: how the standard single-Qbit and two-Qbit gates are translated into the appropriate multi-Qbit operators for the logical Qbits.

Exercise 11.1. Show that $Z_L = Z_0Z_1Z_2Z_3Z_4$ is such that

$$Z_L|0_L\rangle = |0_L\rangle \quad \text{and} \quad Z_L|1_L\rangle = -|1_L\rangle .$$

Similarly, show that $X_L = X_0X_1X_2X_3X_4$ is such that

$$X_L|0_L\rangle = |1_L\rangle \quad \text{and} \quad X_L|1_L\rangle = |0_L\rangle .$$

This shows that these two multi-Qbit operators play the role of the **Z** and **X** for a single-Qbit.

The problem is that other standard gates that you would need to perform Quantum Computation are difficult to write for this 5-Qbit encoding. Mermin [1][Sec. 5.6] discusses in some detail a 7-Qbit encoding, the Steane code, where this task is easier. See also App. E.2.1.

⁶Using $(1 + S_j)^2 = 2(1 + S_j)$.

11.6. General criteria for quantum error correction

Let $\hat{\rho}_C$ be a state with support in the code Hilbert space \mathcal{H}_C . Suppose that we want to correct an approximate truncated quantum map affecting the state of the system $\hat{\rho}_C$:

$$\hat{\rho}_C \xrightarrow[\approx]{\text{truncated errors}} \mathcal{E}_t(\hat{\rho}_C) = \sum_{e=1}^{d_{\text{err}}} \hat{E}_e \hat{\rho}_C \hat{E}_e^\dagger, \quad (11.32)$$

where, from now on, we will use the index e for the Kraus error operators. As discussed, the map is in general **not trace preserving**. To get a feeling for this idea of a truncated, hence approximate, Kraus map, consider the following exercise.

Exercise 11.2. Consider the 3-Qbit bit-flip code discussed previously, with $\mathcal{H}_C = \{|000\rangle, |111\rangle\}$. Consider now the noise process with 4 Kraus operators:

$$\hat{E}_1 = \sqrt{(1-p)^3} \mathbf{1}, \quad \hat{E}_2 = \sqrt{p(1-p)^2} \mathbf{X}_0, \quad \hat{E}_3 = \sqrt{p(1-p)^2} \mathbf{X}_1, \quad \hat{E}_4 = \sqrt{p(1-p)^2} \mathbf{X}_2.$$

where p is the probability of a bit flip. Verify that \mathcal{E}_t is **not trace-preserving**, since we have disregarded operation elements corresponding to bit flips on two and three qubits.

We now pretend that, while a general quantum map \mathcal{E} is **not invertible**, we are still able to **undo** the effects of the (truncated) error set on the state represented by \mathcal{E}_t , by an appropriate **recovery quantum map** \mathcal{R} such that:

$$\mathcal{R} \circ \mathcal{E}_t(\hat{\rho}_C) \propto \hat{\rho}_C, \quad (11.33)$$

where the proportionality sign \propto is, once again, due to the truncation performed in considering \mathcal{E}_t . Let $\{|\phi_j^C\rangle\}$ be an orthonormal basis for the code Hilbert space \mathcal{H}_C , with associated projector

$$\hat{\Pi}_C = \sum_j |\phi_j^C\rangle\langle\phi_j^C|. \quad (11.34)$$

While \mathcal{E}_t brings the state “out of \mathcal{H}_C ”, we would like to construct \mathcal{R} such that it brings the state back in the code Hilbert space \mathcal{H}_C .

The Knill-Laflamme quantum error correction code criterion. A necessary and sufficient condition for a recovery map \mathcal{R} to exist is that:

$$\langle\phi_{j'}^C|\hat{E}_{e'}^\dagger\hat{E}_e|\phi_j^C\rangle = C_{e',e}\delta_{j',j}, \quad (11.35)$$

where C is a $d_{\text{err}} \times d_{\text{err}}$ Hermitean matrix, sometimes called **the code matrix**. Alternatively, and equivalently, in terms of the projector $\hat{\Pi}_C$ we can write the criterion as:

$$\hat{\Pi}_C\hat{E}_{e'}^\dagger\hat{E}_e\hat{\Pi}_C = C_{e',e}\hat{\Pi}_C. \quad (11.36)$$

It follows that C is positive-definite,⁷ hence with non-negative eigenvalues λ_m , so that:

$$\text{Tr}(C) = \sum_m \lambda_m > 0.$$

⁷Indeed:

$$\sum_{e,e'} v_{e'}^* C_{e',e} v_e = \left\| \sum_e v_e \hat{E}_e |\phi_j^C\rangle \right\|^2 \geq 0.$$

On the other hand:

$$\mathrm{Tr}(C) = \langle \phi_j^C | \sum_{e=1}^{d_{\mathrm{err}}} \widehat{E}_e^\dagger \widehat{E}_e | \phi_j^C \rangle \leq 1 ,$$

and the deviation from 1 tells about how much the map fails to be trace-preserving.

Before proving the quantum error correction criterion, it is useful to discuss two general and useful facts.

1. Unitary freedom. There is a unitary-mixing freedom in the choice of the Kraus error operators, similarly to what discussed in Sec. 9.5, even if the map is truncated and not trace preserving. To see this, consider redefining our error Kraus operators \widehat{E}_e by mixing them with an arbitrary unitary matrix \mathbb{U} :

$$\widehat{M}_m = \sum_{e=1}^{d_{\mathrm{err}}} \widehat{E}_e \mathbb{U}_{e,m} . \quad (11.37)$$

The fact that the quantum map does not change is easy to verify:

$$\sum_m \widehat{M}_m \hat{\rho}_C \widehat{M}_m^\dagger = \sum_{e,e'} \underbrace{\left(\sum_m \mathbb{U}_{e,m} \mathbb{U}_{e',m}^* \right)}_{(\mathbb{U}\mathbb{U}^\dagger)_{e,e'} = \delta_{e',e}} \widehat{E}_e \hat{\rho}_C \widehat{E}_{e'}^\dagger = \sum_e \widehat{E}_e \hat{\rho}_C \widehat{E}_e^\dagger . \quad (11.38)$$

Hence, the error map \mathcal{E}_t is identical.

2. Diagonal errors. Next, we show that we can construct, by using the unitary freedom, a new set of Kraus error operators for which the code matrix is diagonal. Consider the criterion in Eq. (11.36), written for the new operators:

$$\begin{aligned} \widehat{\Pi}_C \widehat{M}_{m'}^\dagger \widehat{M}_m \widehat{\Pi}_C &= \sum_{e,e'} \mathbb{U}_{e',m'}^* \widehat{\Pi}_C \widehat{E}_{e'}^\dagger \widehat{E}_e \widehat{\Pi}_C \mathbb{U}_{e,m} \\ &= \sum_{e,e'} \mathbb{U}_{e',m'}^* C_{e'e} \mathbb{U}_{e,m} \widehat{\Pi}_C \\ &= (\mathbb{U}^\dagger C \mathbb{U})_{m',m} \widehat{\Pi}_C = \lambda_m \delta_{m',m} \widehat{\Pi}_C , \end{aligned} \quad (11.39)$$

provided the matrix \mathbb{U} is so chosen as to diagonalise C , i.e., $\mathbb{U}^\dagger C \mathbb{U} = \mathrm{diag}(\lambda_m)$. The (non-negative) eigenvalues λ_m of C are such that $\sum_m \lambda_m = \mathrm{Tr}(C) < 1$ if \mathcal{E}_t is not trace-preserving.

Proof of sufficiency. We first show that if Eq. (11.36) is satisfied, then we can construct a recovery map \mathcal{R} . Observe that if $\hat{\rho}_C$ is a state with support in \mathcal{H}_C , we can always write:

$$\hat{\rho}_C = \widehat{\Pi}_C \hat{\rho}_C \widehat{\Pi}_C . \quad (11.40)$$

We proceed by working with these **diagonal error operators** \widehat{M}_m , and construct the recovery map \mathcal{R} with Kraus operators defined as follows:

$$\widehat{R}_m = \frac{1}{\sqrt{\lambda_m}} \widehat{\Pi}_C \widehat{M}_m^\dagger \quad \text{for} \quad \lambda_m > 0 . \quad (11.41)$$

The fact that this is a valid system of Kraus operators should be verified, by checking $\sum_m \widehat{R}_m^\dagger \widehat{R}_m$. We postpone this check to the end, and proceed with the proof. Let us check the action of $\mathcal{R} \circ \mathcal{E}_t(\hat{\rho}_C)$,

by working with the Kraus operators:

$$\begin{aligned}
\sum_{m|\lambda_m>0} \hat{R}_m \left(\sum_{m'} \hat{M}_{m'} \hat{\rho}_C \hat{M}_{m'}^\dagger \right) \hat{R}_m^\dagger &\stackrel{\text{Eq. (11.41)}}{=} \sum_{m|\lambda_m>0} \frac{1}{\lambda_m} \hat{\Pi}_C \hat{M}_m^\dagger \left(\sum_{m'} \hat{M}_{m'} \hat{\rho}_C \hat{M}_{m'}^\dagger \right) \hat{M}_m \hat{\Pi}_C \\
&\stackrel{\text{Eq. (11.40)}}{=} \sum_{m|\lambda_m>0} \frac{1}{\lambda_m} \sum_{m'} \hat{\Pi}_C \hat{M}_m^\dagger \hat{M}_{m'} \hat{\Pi}_C \hat{\rho}_C \hat{\Pi}_C \hat{M}_{m'}^\dagger \hat{M}_m \hat{\Pi}_C \\
&\stackrel{\text{Eq. (11.39)}}{=} \sum_{m|\lambda_m>0} \frac{1}{\lambda_m} \sum_{m'} \lambda_m^2 \delta_{m',m} \hat{\Pi}_C \hat{\rho}_C \hat{\Pi}_C \\
&\stackrel{\text{Eq. (11.40)}}{=} \left(\sum_m \lambda_m \right) \hat{\rho}_C . \tag{11.42}
\end{aligned}$$

Hence, in summary, $\mathcal{R} \circ \mathcal{E}_t$ keeps the state in the code Hilbert space \mathcal{H}_C :

$$\mathcal{R} \circ \mathcal{E}_t(\hat{\rho}_C) = \left(\sum_m \lambda_m \right) \hat{\rho}_C . \tag{11.43}$$

It remains to be checked that $\{\hat{R}_m\}$ is a valid system of Kraus operators to define a quantum map. For that purpose, consider $\hat{\Pi}_R = \sum_m \hat{R}_m^\dagger \hat{R}_m$.

Exercise 11.3. Show that:

$$\hat{\Pi}_R = \sum_{m|\lambda_m>0} \frac{1}{\lambda_m} \hat{M}_m \hat{\Pi}_C \hat{M}_m^\dagger ,$$

and that $\hat{\Pi}_R$ is a projector, i.e., $\hat{\Pi}_R^2 = \hat{\Pi}_R$.

Hence, by possibly adding a further Kraus operator with support in $\mathbf{1} - \hat{\Pi}_R$, the system of Kraus operators $\{\hat{R}_m\}$ can be made to satisfy the completeness relationship:

$$\sum_m \hat{R}_m^\dagger \hat{R}_m = \mathbf{1} . \tag{11.44}$$

Proof of necessity. By hypothesis, a trace-preserving recovery quantum map \mathcal{R} exists such that:

$$\mathcal{R} \circ \mathcal{E}_t(\hat{\rho}_C) = c \hat{\rho}_C , \tag{11.45}$$

with $c \in (0, 1]$. Let us now write this condition in terms of Kraus operators, using also that $\hat{\rho}_C = \hat{\Pi}_C \hat{\rho}_C \hat{\Pi}_C$. We have:

$$\sum_m \hat{R}_m \left(\sum_e \hat{E}_e \hat{\Pi}_C \hat{\rho}_C \hat{\Pi}_C \hat{E}_e^\dagger \right) \hat{R}_m^\dagger = c \hat{\Pi}_C \hat{\rho}_C \hat{\Pi}_C .$$

Let us rewrite the last equation in the following equivalent but suggestive form:

$$\sum_{m,e} \left(\hat{R}_m \hat{E}_e \hat{\Pi}_C \right) \hat{\rho}_C \left(\hat{R}_m \hat{E}_e \hat{\Pi}_C \right)^\dagger = \left(\sqrt{c} \hat{\Pi}_C \right) \hat{\rho}_C \left(\sqrt{c} \hat{\Pi}_C \right) .$$

So, the two Kraus maps, the LHS-one with $(d_{\text{err}})^2$ Kraus operators $\{\hat{R}_m \hat{E}_e \hat{\Pi}_C\}$, and the RHS-one with a *single* Kraus operator $\{\sqrt{c} \hat{\Pi}_C\}$ are equivalent maps. Hence, by the HJW theorem 9.3⁸ there exists a unitary matrix, of which only the first column vector u_{me} is relevant, such that:

$$\hat{R}_m \hat{E}_e \hat{\Pi}_C = u_{me} \sqrt{c} \hat{\Pi}_C \quad \implies \quad \hat{\Pi}_C \hat{E}_e^\dagger \hat{R}_m^\dagger = u_{me}^* \sqrt{c} \hat{\Pi}_C .$$

Hence, by multiplying the two expressions we obtain:

$$\hat{\Pi}_C \hat{E}_e^\dagger \hat{R}_m^\dagger \hat{R}_m \hat{E}_e \hat{\Pi}_C = c u_{me}^* u_{me} \hat{\Pi}_C .$$

⁸Check that in the proof of that theorem we never actually used a completeness requirement, so that the quantum map need not be necessarily trace preserving.

Upon summing over m , recalling that $\sum_m \hat{R}_m^\dagger \hat{R}_m = \mathbf{1}$, we therefore get:

$$\sum_m \hat{\Pi}_C \hat{E}_{e'}^\dagger \hat{R}_m^\dagger \hat{R}_m \hat{E}_e \hat{\Pi}_C = \hat{\Pi}_C \hat{E}_{e'}^\dagger \hat{E}_e \hat{\Pi}_C = \underbrace{\left(c \sum_m u_{me'}^* u_{me} \right)}_{C_{e',e}} \hat{\Pi}_C. \quad (11.46)$$

From this we conclude that the quantum error correction criterion in Eq. (11.36) is satisfied, with the code matrix

$$C_{e',e} = c \sum_m u_{me'}^* u_{me}, \quad (11.47)$$

which is manifestly Hermitian: $C_{e',e} = C_{e,e'}^*$. Incidentally, we get $\text{Tr} C = c$, since $\sum_m |u_{me}|^2 = 1$.

11.6.1. Content of the QEC criterion and the quantum Hamming bound

Let us discuss more what the quantum error correction criterion tells us about errors affecting codewords, following Ref. [49][Sec. 2.5]. First of all, you can consider the basis states $|\phi_j^C\rangle$ to be the (orthogonal) codewords of the code. Consider now two *different* codewords, $|\phi_j^C\rangle$ and $|\phi_{j'}^C\rangle$ with $j' \neq j$. The criterion tells us that $\hat{E}_e |\phi_j^C\rangle$ is **orthogonal** to $\hat{E}_{e'} |\phi_{j'}^C\rangle$, for all possible errors $\hat{E}_{e'}$, including $e' = e$. Hence, orthogonality is maintained. Next, consider a single codeword $|\phi_j^C\rangle$, and two different errors \hat{E}_e and $\hat{E}_{e'}$. The criterion tells us that:

$$\langle \phi_j^C | \hat{E}_{e'}^\dagger \hat{E}_e | \phi_j^C \rangle = C_{e'e}. \quad (11.48)$$

Hence, if C is **non-diagonal**, two different errors might lead to two different states $\hat{E}_e |\phi_j^C\rangle$ and $\hat{E}_{e'} |\phi_j^C\rangle$ which are **not orthogonal**, but the scalar product is totally *independent* of the codeword $|\phi_j^C\rangle$. Such codes exist, and are called **degenerate codes**. The situation we encountered so far was that different errors lead to *mutually orthogonal* subspaces, and correspond to the so-called **non-degenerate codes**, where the code matrix C is **diagonal**.

Exercise 11.4. Consider again Exercise 11.2. Verify that the quantum error-correction condition is satisfied by the code with the given noise process, verifying that the code is **non-degenerate**.

For **non-degenerate** codes there is a simple bound on the dimensionality of the encoding n we need. Indeed, let us be more general here. Assume that we want to encode k Qbits into a large space of dimensionality 2^n . So, the code subspace \mathcal{H}_C is now 2^k dimensional. Suppose that we have a code that can correct up to $t \geq 1$ errors, generalizing our so-far restricted examples where $t = 1$ (single-Qbit errors). A code of this type is conventionally denoted as $[[n, k, 2t + 1]]$, where $d = 2t$ or $d = 2t + 1$ is the so-called *distance* between codewords, whose concept is simple to explain in classical linear codes, as explained in Appendix E, but will be briefly mentioned at the end of Sec. 11.9.

The errors can affect j Qbits, with $j = 0, 1, \dots, t$ ($j = 0$ is actually the code space, without errors). So, there are $\binom{n}{j}$ way of choosing the Qbits where the errors occur, and for each of these j Qbits, there are 3 possible errors — **X**, **Y**, and **Z** —, hence a total of $\binom{n}{j} 3^j$ errors, for each of which we need to distinguish 2^k -dimensional orthogonal subspace (recall that the code is non-degenerate, hence errors lead to orthogonal states). And all these subspaces have to fit within the 2^n -dimensional encoding space. Hence we find the general **quantum Hamming bound**:

$$2^n \geq 2^k \sum_{j=0}^t \binom{n}{j} 3^j \quad \implies \quad 2^{(n-k)} \geq \sum_{j=0}^t \binom{n}{j} 3^j. \quad (11.49)$$

For $k = 1$ and $t = 1$ (single-Qbit errors for a single logical Qbit), this bound reduces to what we derived at the end of Sec. 11.4:

$$2^{n-1} \geq (3n + 1).$$

11.6.2. Digitization of quantum noise: again

Suppose that we have a truncated map \mathcal{E}_t , defined by a set of Kraus errors $\{\widehat{E}_e\}$ — or equivalently, by the corresponding diagonal errors $\{\widehat{M}_m\}$ — which are *correctable*. For instance, you might have checked that when $\{\widehat{E}_e\}$ are a particular restricted set of Pauli string operators, the quantum error correction criterion is verified.

You now consider a *different* quantum error map $\widetilde{\mathcal{E}}_t$ with Kraus operators which are made by *arbitrary linear superpositions* of the correctable errors. Without loss of generality, we can assume that the new Kraus error operators are expressed in terms of the correctable diagonal errors as follows:

$$\widehat{K}_e = \sum_m \widehat{M}_m \mathbb{A}_{m,e}, \quad (11.50)$$

and that the new quantum error map is given by:

$$\hat{\rho}_C \xrightarrow[\approx]{\text{truncated errors}} \widetilde{\mathcal{E}}_t(\hat{\rho}_C) = \sum_{e=1}^{\text{derr}} \widehat{K}_e \hat{\rho}_C \widehat{K}_e^\dagger. \quad (11.51)$$

Question: Arbitrary linear combinations are correctable?

Q1: Is $\widetilde{\mathcal{E}}_t$ correctable? **Q2:** If so, what is the recovery map?

Q1. Write down the quantum error correction criterion for the new error operators:

$$\begin{aligned} \widehat{\Pi}_C \widehat{K}_{e'}^\dagger \widehat{K}_e \widehat{\Pi}_C &= \sum_{m,m'} \mathbb{A}_{m',e'}^* \widehat{\Pi}_C \widehat{M}_{m'}^\dagger \widehat{M}_m \widehat{\Pi}_C \mathbb{A}_{m,e} \stackrel{\text{Eq. (11.39)}}{=} \sum_m (\mathbb{A}^\dagger)_{e',m} \lambda_m \mathbb{A}_{m,e} \widehat{\Pi}_C \\ &= (\mathbb{A}^\dagger \text{diag}(\lambda_m) \mathbb{A})_{e',e} \widehat{\Pi}_C. \end{aligned} \quad (11.52)$$

Hence, the new code matrix $C^K = \mathbb{A}^\dagger \text{diag}(\lambda_m) \mathbb{A}$ is Hermitean, and the quantum error correction criterion in Eq. (11.36) is satisfied. Hence, the new map $\widetilde{\mathcal{E}}_t$ is correctable as well.

Q2. To answer the second part of the question, you need to show that the same recovery map correcting \mathcal{E}_t works fine here, by doing the following exercise. See Ref. [49][Sec. 2.6] for more discussions.

Exercise 11.5. Show that if \widehat{R}_m are the recovery operators constructed in terms of the diagonal errors \widehat{M}_m as in Eq. (11.41), then we have:

$$\sum_{m|\lambda_m>0} \widehat{R}_m \left(\sum_e \widehat{K}_e \hat{\rho}_C \widehat{K}_e^\dagger \right) \widehat{R}_m^\dagger = \left(\sum_m \sum_e \lambda_m |\mathbb{A}_{m,e}|^2 \right) \hat{\rho}_C. \quad (11.53)$$

Hence \mathcal{R} can correct the new linear superposition of errors as well.

11.7. The stabilizers and the Pauli group

i **Stabilizers.** A set of stabilizers $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots\}$ is:

- 1) a set of **mutually commuting Hermitean operators** which square to $\mathbf{1}$.
- 2) the identity $\mathbf{1} \in \mathcal{S}$, while $-\mathbf{1} \notin \mathcal{S}$.

\mathcal{S} is an **Abelian group**.^a All the stabilizers in \mathcal{S} , since they square to $\mathbf{1}$, have eigenvalues ± 1 .

^aIndeed: $\mathbf{1} \in \mathcal{S}$; $\mathbf{S}_s \mathbf{S}_{s'} \in \mathcal{S}$, since $(\mathbf{S}_s \mathbf{S}_{s'})^2 = \mathbf{1}$; $\mathbf{S}_s^{-1} = \mathbf{S}_s$, since $\mathbf{S}_s^2 = \mathbf{1}$.

i **The stabilized space.** The **stabilized** space $\mathcal{H}_{\mathcal{S}}$ is the subspace of the Hilbert space \mathcal{H} made up by all states $|\psi\rangle$ which are eigenstates with eigenvalue $+1$ of all stabilizers:

$$|\psi\rangle \in \mathcal{H}_{\mathcal{S}} \quad \iff \quad \mathbf{S}_s |\psi\rangle = |\psi\rangle \quad \forall \mathbf{S}_s \in \mathcal{S}. \quad (11.54)$$

The generators. There exist a minimal set of n_s *independent*⁹ stabilizers which *generate* the group \mathcal{S} by taking products of the generators. We will denote the generators as follows:

$$\mathcal{S} = \langle \mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_{n_s} \rangle. \quad (11.55)$$

Any element of \mathcal{S} can be written as:

$$\mathbf{S}_1^{a_1} \mathbf{S}_2^{a_2} \dots \mathbf{S}_{n_s}^{a_{n_s}} \quad \text{with} \quad a_j = 0, 1.$$

Encoding with stabilizers. You can encode logical Qbits into a 2^n -dimensional encoding space \mathcal{H} , with the set of vectors $|\psi\rangle$ which have eigenvalues $+1$ on all stabilizers generators, which form a linear subspace $\mathcal{H}_{\mathcal{S}} \equiv \mathcal{H}_{\mathcal{C}}$:

$$\mathbf{S}_s |\psi\rangle = |\psi\rangle \quad \text{for} \quad s = 1, \dots, n_s. \quad (11.56)$$

Simple counting — for each of the “independent constraints” of having eigenvalue $+1$ on the generators, you lower the space dimensionality by a factor 2; for a more formal proof see App. E.3 — tells us that the dimensionality of such a subspace is 2^{n-n_s} , hence we have $k = n - n_s$ independent codewords in $\mathcal{H}_{\mathcal{C}}$. As usual, we consider a basis $\{|\phi_j^{\mathcal{C}}\rangle\}$, with $j = 1, \dots, 2^k$, of orthogonal states (codewords) for $\mathcal{H}_{\mathcal{C}}$.

Question: How to construct stabilizers?

One might wonder how to construct stabilizers in practice. We now see that they can be written in terms of Pauli matrices.

The Pauli group. Suppose that we have a single Qbit, and we would be asked to write a multiplicative *group* containing the Pauli matrices. A naive idea is to think that this is made by $\{\mathbf{1}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$, but this is wrong, because this set is not a group. Indeed, for instance, $\mathbf{XZ} = -i\mathbf{Y}$ and $\mathbf{ZX} = +i\mathbf{Y}$, hence also $\pm i\mathbf{Y}$ must be included in the set. By similar reasoning $\pm i\mathbf{X}$, and $\pm i\mathbf{Z}$ must be included. But also $(-i\mathbf{X})\mathbf{Z} = -\mathbf{Y}$, hence $\pm\mathbf{Y}$ must be included, and similarly $\pm\mathbf{X}$, and $\pm\mathbf{Z}$ must be included. Finally

⁹Independent here means, effectively, that by eliminating any of the generators you obtain a smaller group. Proving independence can be done, in a rather simple way, by relying on a check-matrix and on the familiar concept of linear independence of linear algebra. See App. E for details.

$(i\mathbf{X})^2 = -\mathbf{1}$, and $(\pm i\mathbf{X})\mathbf{X} = \pm i\mathbf{1}$. So, we must admit an overall root-of-unity factor $w_m = e^{im\pi/2}$ with $m = 0, 1, 2, 3$. Summarizing:

$$\mathcal{P}_1 = \{w_m \mathbf{1}, w_m \mathbf{X}, w_m \mathbf{Y}, w_m \mathbf{Z}\} = \{w_m \hat{\sigma}^{(\mu)}\} \quad \text{with} \quad \mu = 0, 1, 2, 3. \quad (11.57)$$

For n Qbits — for simplicity, we number them from 1 to n rather than from 0 to $n - 1$, as usual — we have the Pauli group

$$\mathcal{P}_n = \{w_m \hat{\sigma}_1^{(\mu_1)} \hat{\sigma}_2^{(\mu_2)} \dots \hat{\sigma}_n^{(\mu_n)}\} \quad \text{with} \quad \mu_j = 0, 1, 2, 3, \quad (11.58)$$

made up by all possible **Pauli strings** with an overall factor w_m , hence with 4^{n+1} elements, half of them, 2^{2n+1} are Hermitean and squaring to $\mathbf{1}$, the remaining half are anti-Hermitean and squaring to $-\mathbf{1}$. Stabilizers groups \mathcal{S} can be constructed as appropriate **subgroups** of the Pauli group \mathcal{P}_n : $\mathcal{S} \subset \mathcal{P}_n$.

Examples. The 3-Qbit encoding which protect against single Qbit-flip errors is evidently described by the stabilizer group:

$$\mathcal{S} = \langle \mathbf{S}_1 = \mathbf{Z}_0 \mathbf{Z}_1, \mathbf{S}_2 = \mathbf{Z}_1 \mathbf{Z}_2 \rangle. \quad (11.59)$$

The 5-Qbit encoding which protect against single Qbit error of any type (\mathbf{X} , \mathbf{Y} , or \mathbf{Z}) is described by the stabilizer group:

$$\mathcal{S} = \langle \mathbf{S}_1 = \mathbf{Z}_1 \mathbf{X}_2 \mathbf{X}_3 \mathbf{Z}_4, \mathbf{S}_2 = \mathbf{Z}_2 \mathbf{X}_3 \mathbf{X}_4 \mathbf{Z}_0, \mathbf{S}_3 = \mathbf{Z}_3 \mathbf{X}_4 \mathbf{X}_0 \mathbf{Z}_1, \mathbf{S}_4 = \mathbf{Z}_4 \mathbf{X}_0 \mathbf{X}_1 \mathbf{Z}_2 \rangle. \quad (11.60)$$

11.8. Unitary transformations and the Clifford group

Consider a unitary transformation \mathbf{U} in the Hilbert space and a state in a stabilized subspace $|\psi\rangle \in \mathcal{H}_{\mathcal{S}}$. Then for any stabilizer \mathbf{S}_s :

$$\mathbf{U}|\psi\rangle = \mathbf{U}\mathbf{S}_s|\psi\rangle = (\mathbf{U}\mathbf{S}_s\mathbf{U}^\dagger)\mathbf{U}|\psi\rangle. \quad (11.61)$$

Hence $\mathbf{U}\mathbf{S}_s\mathbf{U}^\dagger$ stabilizes the vector space $\mathbf{U}\mathcal{H}_{\mathcal{S}}$. Similarly to the more familiar Heisenberg representation of operators, you can write the transformed stabilizer group as

$$\mathbf{U}\mathcal{S}\mathbf{U}^\dagger = \langle \mathbf{U}\mathbf{S}_1\mathbf{U}^\dagger, \dots, \mathbf{U}\mathbf{S}_{n_s}\mathbf{U}^\dagger \rangle. \quad (11.62)$$

The operation $g \rightarrow \mathbf{U}g\mathbf{U}^\dagger$ is known as **conjugation**. Let us now restrict our attention to elements of the Pauli group, $g \in \mathcal{P}_n$. We want to study the set of unitaries \mathbf{U} that preserve the Pauli group by conjugation, i.e.,

$$\{\text{unitary } \mathbf{U} \text{ such that: } g \in \mathcal{P}_n \implies \mathbf{U}g\mathbf{U}^\dagger \in \mathcal{P}_n\}. \quad (11.63)$$

Let us start from a single Qbit, $n = 1$. If $\mathbf{U} \in \mathcal{P}_1$, then obviously $\mathbf{U}g\mathbf{U}^\dagger$ for all $g \in \mathcal{P}_1$, because of the group property. But there are unitaries that *are not* in the Pauli group, which still transform Pauli group elements into Pauli group elements. Let us see some examples.

The Hadamard gate Consider the Hadamard gate $\mathbf{U} = \mathbf{H} = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z})$. Notice that $\mathbf{H} = \mathbf{H}^\dagger$ and $\mathbf{H}^2 = \mathbf{1}$, but $\mathbf{H} \notin \mathcal{P}_1$. Still:

$$\mathbf{H}\mathbf{X}\mathbf{H}^\dagger = \mathbf{Z}, \quad \mathbf{H}\mathbf{Z}\mathbf{H}^\dagger = \mathbf{X}, \quad \mathbf{H}\mathbf{Y}\mathbf{H}^\dagger = -\mathbf{Y}.$$

Similarly, for any other element g of the Pauli group, you can show that $\mathbf{H}g\mathbf{H}^\dagger \in \mathcal{P}_1$. ¹⁰

¹⁰Incidentally, $|0\rangle$ is a state stabilized by \mathbf{Z} , since $\mathbf{Z}|0\rangle = |0\rangle$. And $\mathbf{H}|0\rangle = |+\rangle$ is a state stabilized by $\mathbf{H}\mathbf{Z}\mathbf{H}^\dagger = \mathbf{X}$, since $\mathbf{X}|+\rangle = |+\rangle$. There are interesting consequences of this simple idea, for which we refer the reader to Ref. [3][Sec. 10.5.2].

The S-gate. Consider the S-gate, $\mathbf{S} = \text{diag}(1, i)$, which is unitary, *not Hermitean* and such that $\mathbf{S}^2 = \mathbf{Z}$. It is easy to show that:

$$\mathbf{SXS}^\dagger = \mathbf{Y}, \quad \mathbf{SZS}^\dagger = \mathbf{Z}, \quad \mathbf{SYS}^\dagger = -\mathbf{X}.$$

Similarly, for any other element g of the Pauli group, you can show that $\mathbf{SgS}^\dagger \in \mathcal{P}_1$.



The T-gate. Notice, however, that the T-gate, where $\mathbf{T} = \text{diag}(1, e^{i\pi/4})$, which is unitary, *not Hermitean* and such that $\mathbf{T}^4 = \mathbf{Z}$, behaves very differently, as it **does not preserve the Pauli group by conjugation**:

$$\mathbf{TXT}^\dagger = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Y}), \quad \mathbf{TZT}^\dagger = \mathbf{Z}, \quad \mathbf{TYT}^\dagger = \frac{1}{\sqrt{2}}(\mathbf{Y} - \mathbf{X}).$$

The CNOT-gate. Let us now move the $n = 2$ Qbits, and consider the cNOT-gate. Recall that \mathbf{C}_{10} — the cNOT with 1 as a control-bit, and 0 as a target-bit — is given by:

$$\mathbf{C}_{10} = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_1 + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_1 \mathbf{X}_0. \quad (11.64)$$

We know that $\mathbf{C}_{10} = \mathbf{C}_{10}^\dagger$, $\mathbf{C}_{10}^2 = \mathbf{1}$, but evidently $\mathbf{C}_{10} \notin \mathcal{P}_2$. Still, you can easily show that the Pauli group is preserved by conjugation.

Exercise 11.6. Show that:

$$\mathbf{C}_{10} \begin{pmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{Z}_0 \\ \mathbf{Z}_1 \end{pmatrix} \mathbf{C}_{10}^\dagger = \begin{pmatrix} \mathbf{X}_0 \\ \mathbf{X}_0 \mathbf{X}_1 \\ \mathbf{Z}_0 \mathbf{Z}_1 \\ \mathbf{Z}_1 \end{pmatrix}. \quad (11.65)$$



The Clifford group. The set of all n -bit unitaries \mathbf{U} which preserve the Pauli group by conjugation is known as *Clifford group* \mathcal{C}_n , or also as the **normalizer** of the Pauli group, $\mathcal{N}(\mathcal{P}_n)$:

$$\mathcal{C}_n \equiv \mathcal{N}(\mathcal{P}_n) \stackrel{\text{def}}{=} \{ \mathbf{U} \mid g \in \mathcal{P}_n \implies \mathbf{U}g\mathbf{U}^\dagger \in \mathcal{P}_n \}. \quad (11.66)$$

The Clifford group is a *proper* subgroup of the group of all possible n -Qbit unitaries, while, clearly $\mathcal{P}_n \subset \mathcal{C}_n$. Interestingly, the Clifford group is generated by a restricted set of gates, known as **Clifford gates**, given by:

$$\text{Clifford gates} = \{ \mathbf{H}, \mathbf{S}, \mathbf{C}_{ij} \}, \quad (11.67)$$

where “generated” means that any $\mathbf{U} \in \mathcal{C}_n$ can be decomposed into $O(n^2)$ Clifford gates, see Ref. [3][Theorem 10.6].



Clifford gates are not universal. The Clifford gates are **not a universal set** of gates. To make them universal, you would need to add, for instance, the T-gate:

$$\text{A universal set of gates} = \{ \mathbf{H}, \mathbf{S}, \mathbf{T}, \mathbf{C}_{ij} \}. \quad (11.68)$$

Notice that Clifford gates do **create entanglement**, by using **H** and cNOT. Still, the **Gottesmann-Knill theorem**, see App. E.4, demonstrates that quantum circuits composed only with Clifford gates

can be **efficiently simulated with classical computers**. This would open the discussion to an important recent topic in the quantum computing literature, the so-called **Magic**.

Fortunately, there are many interesting things that can be done invoking only Clifford gates and using the stabilizer's formalism, including:

- Create quantum codes, the so-called **stabilizers codes**. See Sec. 11.9 and App. E.2.
- Encode (prepare) and decode (if necessary) logical states. See Ref. [3][Sec. 10.5.8].
- Construct the logical gates. See App. E.3.2 and Ref. [3][Sec. 10.5.7].
- Do measurements. See App. E.3.1 and Ref. [3][Sec. 10.5.3].
- Do quantum error correction. See Sec. 11.9 and Ref. [3][Sec. 10.5.8].

11.9. Stabilizer codes

As mentioned, encoding of logical Qbits into a large space of n physical Qbits can be done by using an appropriate stabilizer group \mathcal{S} , a subgroup of \mathcal{P}_n . There is a standard form in which you can put the stabilizer group, see App. E.3.2 and Ref. [3][Sec. 10.5.7], which I will not discuss here, which makes relatively simple to construct logical \mathbf{Z}_L and \mathbf{X}_L gates. Assuming that, I now illustrate a bit more in dept the stabilizer code construction, concentrating in particular on the ability to perform error correction.

Consider a stabilizer group with n_s independent generators, for which we know that the stabilized space \mathcal{H}_S has dimensionality 2^k , with $k = n - n_s$, hence:

$$\mathcal{S} = \langle \mathbf{S}_1, \dots, \mathbf{S}_{n-k} \rangle. \quad (11.69)$$

We denote the code associated to such a \mathcal{S} as $C(\mathcal{S})$: it is a $[n, k]$ code, where n is the number of physical Qbits, and k the number of logical Qbits.

As mentioned, there is a **standard form** for writing the stabilizer generators, see App. E.3.2, that makes it not difficult to construct k Pauli operators $\mathbf{Z}_{j,L} \in \mathcal{P}_n$, with $j = 1, \dots, k$ which play the role of the \mathbf{Z} logical gates. These operators obey $\mathbf{Z}_{j,L}^2 = \mathbf{1}$, and, moreover, there is a state in \mathcal{H}_S , which we denote as $|\underline{0}_L\rangle \in \mathcal{H}_S$ — where $\underline{0} = (0, \dots, 0)$ denotes a k -bit string of 0s —, such that:

$$\mathbf{Z}_{j,L}|\underline{0}_L\rangle = |\underline{0}_L\rangle.$$

You realise that these k logical \mathbf{Z} operators must all commute with the stabilizers. Consider then the enlarged stabilizer group

$$\mathcal{S}_{\underline{0}} = \langle \mathbf{S}_1, \dots, \mathbf{S}_{n-k}, \mathbf{Z}_{1,L}, \dots, \mathbf{Z}_{k,L} \rangle,$$

with n elements. Its associated stabilized space is evidently just $|\underline{0}_L\rangle$. Alternatively, you say that $|\underline{0}_L\rangle$ is the **stabilized state** associated to $\mathcal{S}_{\underline{0}}$. Similarly, for any binary string $\underline{x} = (x_k, \dots, x_1)$ with $x_j = 0, 1$, you can define an enlarged stabilizer group with n elements

$$\mathcal{S}_{\underline{x}} = \langle \mathbf{S}_1, \dots, \mathbf{S}_{n-k}, (-1)^{x_1} \mathbf{Z}_{1,L}, \dots, (-1)^{x_k} \mathbf{Z}_{k,L} \rangle,$$

which stabilizes the logical computational basis state $|\underline{x}_L\rangle$. The k logical \mathbf{X} operators are defined as $\mathbf{X}_{j,L} \in \mathcal{P}_n$ such that $\mathbf{X}_{j,L} \mathbf{S}_s \mathbf{X}_{j,L}^\dagger = \mathbf{S}_s$ and:

$$\mathbf{X}_{j,L} \mathbf{Z}_{j',L} \mathbf{X}_{j,L}^\dagger = \begin{cases} \mathbf{Z}_{j',L} & \text{if } j' \neq j \\ -\mathbf{Z}_{j,L} & \text{if } j' = j \end{cases}.$$

Although all this sounds mysterious at this stage, because we did not enter into the details of *how* these logical operators can be constructed, let us assume that this can be done. We now turn our attention to errors.

11.9.1. Error correction for stabilizer codes

Let us examine possible **Pauli errors** $\hat{E} \in \mathcal{P}_n$.

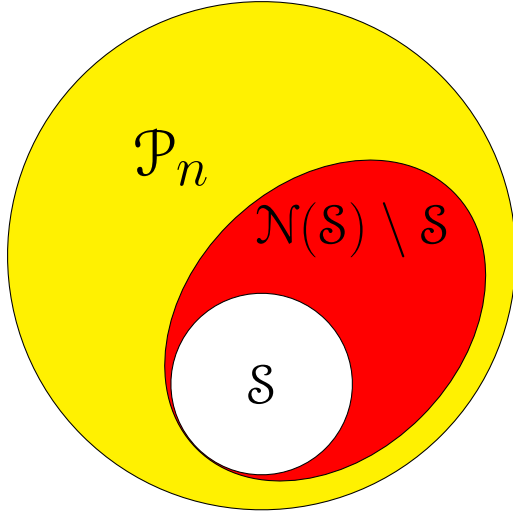


Figure 11.12: The stabilizer \mathcal{S} , a subgroup of the normalizer $\mathcal{N}(\mathcal{S})$, coincident with the centralizer $\mathcal{C}(\mathcal{S})$, formed by all Pauli strings that *commute* with the stabilizers. Both are subgroups of the Pauli group \mathcal{P}_n . In red, the difference set $\mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$, i.e., all the Pauli string operators that commute with \mathcal{S} but *do not belong* to \mathcal{S} : any set of errors such that some $\hat{E}_e^\dagger \hat{E}_e \in \mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$ is **not correctable**.

i

The centralizer and normalizer of \mathcal{S} . The **centralizer** of the stabilizer group \mathcal{S} , denoted as $\mathcal{C}(\mathcal{S})$, is the subgroup of the Pauli group \mathcal{P}_n composed by operators that **commute** with the stabilizers:

$$\mathcal{C}(\mathcal{S}) \stackrel{\text{def}}{=} \{g \in \mathcal{P}_n \mid g \mathbf{S}_s g^\dagger = \mathbf{S}_s \quad \forall \mathbf{S}_s \in \mathcal{S}\}. \quad (11.70)$$

This should be contrasted with the definition of the normalizer $\mathcal{N}(\mathcal{S})$:

$$\mathcal{N}(\mathcal{S}) \stackrel{\text{def}}{=} \{g \in \mathcal{P}_n \mid g \mathbf{S}_s g^\dagger \in \mathcal{S} \quad \forall \mathbf{S}_s \in \mathcal{S}\}, \quad (11.71)$$

where we *do not require* the conjugation $\mathbf{S}_s \rightarrow g \mathbf{S}_s g^\dagger$ to lead to the *same* stabilizer element (totally equivalent to a commutation requirement). Evidently, in general, $\mathcal{S} \subset \mathcal{C}(\mathcal{S}) \subset \mathcal{N}(\mathcal{S})$. However, it is easy to realize ^a that for the specific case of stabilizers in the Pauli group, centralizers and normalizers do coincide: $\mathcal{S} \subset \mathcal{C}(\mathcal{S}) \equiv \mathcal{N}(\mathcal{S})$.

^aThe reason is that two Pauli strings either commute or anti-commute, hence conjugation (i.e., commutation) *must* bring to the same element of the stabilizer. This is at variance with the case of $\mathcal{N}(\mathcal{P}_n)$ where the elements that realize the conjugation are taken from all the unitary operators.

Since two Pauli string operators either **commute or anti-commute**, there are only three possibilities, illustrated in Fig. 11.12.

- 1) There is a stabilizer $\mathbf{S}_s \in \mathcal{S}$ such that $\hat{E} \mathbf{S}_s = -\mathbf{S}_s \hat{E}$, i.e., \hat{E} anti-commutes with some \mathbf{S}_s : $\{\hat{E}, \mathbf{S}_s\} = 0$. This means that $\hat{E} \in \mathcal{P}_n \setminus \mathcal{N}(\mathcal{S})$, the yellow part in Fig. 11.12. Then, for any $|\psi\rangle \in \mathcal{H}_{\mathcal{S}}$, we have:

$$\hat{E}|\psi\rangle = \hat{E} \mathbf{S}_s |\psi\rangle = -\mathbf{S}_s \hat{E} |\psi\rangle,$$

which means that $\widehat{E}|\psi\rangle$ has eigenvalue $\lambda_s = -1$ for \mathbf{S}_s , hence belongs to a subspace orthogonal to \mathcal{H}_S . $\lambda_s = -1$ is known as the **syndrome**, because it allows to spot the occurrence of the error.

- 2) \widehat{E} commutes with all the stabilizers, $[\widehat{E}, \mathbf{S}_s] = 0$, and $\widehat{E} \in \mathcal{S}$. Then for any $|\psi\rangle \in \mathcal{H}_S$ we have that $\widehat{E}|\psi\rangle \in \mathcal{H}_S$, hence no corruption at all.
- 3) \widehat{E} commutes with all the stabilizers, $[\widehat{E}, \mathbf{S}_s] = 0$, but $\widehat{E} \notin \mathcal{S}$. This implies that \widehat{E} belongs to the difference set $\mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$, the red part in Fig. 11.12. This is a very bad case, because $\widehat{E}|\psi\rangle \notin \mathcal{H}_S$, and yet it is impossible to measure any stabilizer \mathbf{S}_s with an eigenvalue $\lambda_s = -1$: **no syndrome** for the error is available!

With these preliminary considerations, it is very easy to prove the following theorem.

❶

Error-correction conditions for stabilizer codes. Let \mathcal{S} be the stabilizer of some code $C(\mathcal{S})$, and $\{\widehat{E}_e\}$ a set of errors in \mathcal{P}_n such that

$$\widehat{E}_{e'}^\dagger \widehat{E}_e \notin \mathcal{N}(\mathcal{S}) \setminus \mathcal{S} \quad \forall (e', e).$$

Then $\{\widehat{E}_e\}$ is a **correctable set** of errors for the code $C(\mathcal{S})$.

Proof. The proof of this theorem is a simple application of the quantum error correction criterion discussed previously, which we here repeat for convenience:

$$\widehat{\Pi}_C \widehat{E}_{e'}^\dagger \widehat{E}_e \widehat{\Pi}_C = C_{e',e} \widehat{\Pi}_C. \quad (11.72)$$

The projector $\widehat{\Pi}_C$ on the code/stabilized subspace $\mathcal{H}_C \equiv \mathcal{H}_S$ can be written in terms of projectors on the eigenvalues $+1$ for all the stabilizer generators:

$$\widehat{\Pi}_C = \frac{1}{2^{n-k}} \prod_{s=1}^{n-k} (\mathbf{1} + \mathbf{S}_s). \quad (11.73)$$

By hypothesis $\widehat{E}_{e'}^\dagger \widehat{E}_e \notin \mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$, hence, as illustrated in Fig. 11.12, two possibilities are left:

$$(e', e) \longrightarrow \begin{cases} \widehat{E}_{e'}^\dagger \widehat{E}_e \in \mathcal{S} & \text{Case 1} \\ \widehat{E}_{e'}^\dagger \widehat{E}_e \in \mathcal{P}_n \setminus \mathcal{N}(\mathcal{S}) & \text{Case 2} \end{cases}$$

In **Case 1**), since $\widehat{\Pi}_C$ is invariant under multiplication by an element of the stabilizer group, we have evidently:

$$\widehat{\Pi}_C \widehat{E}_{e'}^\dagger \widehat{E}_e \widehat{\Pi}_C = \widehat{\Pi}_C \quad \implies \quad C_{e',e} = 1.$$

In **Case 2**), there exist a stabilizer \mathbf{S}_s , which without loss of generality we can call \mathbf{S}_1 , which anti-commutes with $\widehat{E}_{e'}^\dagger \widehat{E}_e$:

$$\widehat{E}_{e'}^\dagger \widehat{E}_e \mathbf{S}_1 = -\mathbf{S}_1 \widehat{E}_{e'}^\dagger \widehat{E}_e.$$

This implies that

$$\widehat{E}_{e'}^\dagger \widehat{E}_e \widehat{\Pi}_C = (\mathbf{1} - \mathbf{S}_1) \widehat{E}_{e'}^\dagger \widehat{E}_e \frac{1}{2^{n-k}} \prod_{s=2}^{n-k} (\mathbf{1} + \mathbf{S}_s).$$

But since $(\mathbf{1} - \mathbf{S}_1)(\mathbf{1} + \mathbf{S}_1) = 0$ by orthogonality of the two subspaces, then you conclude that:

$$\widehat{\Pi}_C \widehat{E}_{e'}^\dagger \widehat{E}_e \widehat{\Pi}_C = 0 \quad \implies \quad C_{e',e} = 0.$$

This concludes our proof, since we have shown that a simple code matrix $C_{e',e}$ exists, hence the quantum error correction criterion is satisfied. ■

11.9.2. Syndrome detection for stabilizer codes

It remain to discuss how, given a set $\{\widehat{E}_e\}$ of correctable errors with $\widehat{E}_e \in \mathcal{P}_n$ for the stabilizer code $C(\mathcal{S})$, to clearly identify the error, and eventually apply the recovery operation. This is done through a **syndrome measurement**. In more detail, given the set of correctable Pauli string errors $\{\widehat{E}_e\}$, consider the conjugation of all the stabilizer generators:

$$\widehat{E}_e \mathbf{S}_s \widehat{E}_e^\dagger = \lambda_{se} \mathbf{S}_s \iff \widehat{E}_e \mathbf{S}_s = \lambda_{se} \mathbf{S}_s \widehat{E}_e. \quad (11.74)$$

Evidently we must have $\lambda_{se} = \pm 1$: we have $\lambda_{se} = +1$ if \widehat{E}_e commutes with \mathbf{S}_s (and is in the stabilizer, since errors are correctable), while $\lambda_{se} = -1$ if \widehat{E}_e anti-commutes with \mathbf{S}_s . Since for any $|\psi\rangle \in \mathcal{H}_S$ we have:

$$\widehat{E}_e |\psi\rangle = \widehat{E}_e \mathbf{S}_s |\psi\rangle = \lambda_{se} \mathbf{S}_s \widehat{E}_e |\psi\rangle,$$

you realize that λ_{se} is indeed the eigenvalue of \mathbf{S}_s on the state $\widehat{E}_e |\psi\rangle$ — recall the table we constructed for the 5-Qbit encoding in Sec. 11.5. The stabilizer generators can be *measured* by the usual ancilla-trick, as explained in Sec. 11.5 and illustrated in Fig. 11.11, providing a column vector of syndromes $\boldsymbol{\lambda}^{\text{meas}} = (\lambda_1^{\text{meas}}, \dots, \lambda_{n_s}^{\text{meas}})^T$.

Now, given the $n_s \times d_{\text{err}}$ syndrome table λ_{se} two possibilities are given:

- 1) \widehat{E}_e is the unique error which realizes the given measured syndrome $\boldsymbol{\lambda}^{\text{meas}}$. Spot the eigenvalue -1 in the syndrome column vector, and apply the appropriate recovery.
- 2) The measured syndrome $\boldsymbol{\lambda}^{\text{meas}}$ is realized by two (or more) different errors, \widehat{E}_e and $\widehat{E}_{e'}$, say, having the same syndrome:

$$\lambda_{se} = \lambda_{se'} \quad \forall s = 1 \dots n_s.$$

This implies, show it, that:

$$\widehat{E}_e \widehat{\Pi}_C \widehat{E}_e^\dagger = \widehat{E}_{e'} \widehat{\Pi}_C \widehat{E}_{e'}^\dagger.$$

Hence, you deduce that

$$\widehat{\Pi}_C = \widehat{E}_e^\dagger \widehat{E}_{e'} \widehat{\Pi}_C \widehat{E}_{e'}^\dagger \widehat{E}_e.$$

From this, it follows that $\widehat{E}_e^\dagger \widehat{E}_{e'} \in \mathcal{S}$. Hence you need to apply \widehat{E}_e^\dagger after $\widehat{E}_{e'}$ to perform the error recovery.

The distance d of a stabilizer code. Let me conclude with a useful concept: the distance of a code. First we need to discuss the concept of **weight** of a Pauli string operator $g \in \mathcal{P}_n$.

i **The weight of a Pauli string.** The weight $\text{wt}(g)$ of an operator $g \in \mathcal{P}_n$ is the number on non-identity elements (i.e., \mathbf{X} , \mathbf{Y} and \mathbf{Z}) in the string. So, for instance, $g = i\mathbf{Y}_0\mathbf{Z}_3\mathbf{X}_4$ has weight $\text{wt}(g) = 3$.

Now consider all the Pauli string errors $\widehat{E} \in \mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$, the dangerous errors that cannot be detected because they are not stabilizers, but they commute with \mathcal{S} , see red part of Fig. 11.12.

i **The distance of a stabilizer code.** The distance d of a stabilizer code $C(\mathcal{S})$ (recall that it is an $[n, k]$ code with $n_s = n - k$) is given by:

$$d = \{\min(\text{wt}(\widehat{E})) \mid \widehat{E} \in \mathcal{N}(\mathcal{S}) \setminus \mathcal{S}\}. \quad (11.75)$$

In words, the distance d is the minimum weight of non-correctable Pauli string errors. A code of distance d is often denoted as $[n, k, d]$.

Exercise 11.7. Show that the 3-Qbit code that corrects single-Qbit bit flip errors had distance $d = 2$, hence it is a $[3, 1, 2]$ code. Show that the 5-Qbit code that corrects single-Qbit errors of any type has distance $d = 3$, hence it is a $[5, 1, 3]$ code.

11.10. The Toric code

An interesting route is to devise stabilizer codes that have intrinsic *topological* properties, like the toric code introduced by Kitaev [53].

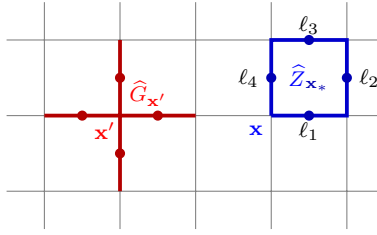


Figure 11.13.: Representation of a star operator $\hat{G}_{\mathbf{x}'}$ (in red) and a plaquette operator $\hat{Z}_{\mathbf{x}^*}$ (in blue), with the corresponding Qbits on all the links (solid circles, only partially shown).

For that purpose consider a square lattice with quantum spins on the *links*, as pictorially indicated in Fig. 11.13. We denote the links by the shorthand $\ell = \mathbf{x}, \nu \longleftrightarrow (\mathbf{x}, \mathbf{x} + \mathbf{e}_\nu)$ where \mathbf{e}_ν are the nearest-neighbor vectors on the lattice ($\nu = \pm 1, \pm 2$, with $\mathbf{e}_{-1} = -\mathbf{e}_1$ and $\mathbf{e}_{-2} = -\mathbf{e}_2$) and \mathbf{x} are the sites of the lattice. We then define a *plaquette operator* made-up by four Pauli-Z matrices on the links of each plaquette:

$$\hat{Z}_{\mathbf{x}^*} = \hat{\sigma}_{\ell_1}^z \hat{\sigma}_{\ell_2}^z \hat{\sigma}_{\ell_3}^z \hat{\sigma}_{\ell_4}^z = \prod_{\ell \in \text{plaq}(\mathbf{x}^*)} \hat{\sigma}_\ell^z. \quad (11.76)$$

Also, we define the “star” at a lattice site \mathbf{x} to be set of all 4 links touching \mathbf{x} :

$$\text{star}(\mathbf{x}) = \{ \ell = \mathbf{x}, \nu \text{ with } \nu = \pm 1, \pm 2 \}, \quad (11.77)$$

and the associated “flip” operator ¹¹ at each vertex \mathbf{x} , defined as:

$$\hat{G}_{\mathbf{x}} = \prod_{\nu=1}^2 \left(\hat{\sigma}_{\mathbf{x}, \nu}^x \hat{\sigma}_{\mathbf{x}, -\nu}^x \right) = \prod_{\ell \in \text{star}(\mathbf{x})} \hat{\sigma}_\ell^x. \quad (11.78)$$

i

Toric code model. On a $L \times L$ square lattice with PBC in both directions, Kitaev’s toric code model reads: ^a

$$\hat{H}_{\text{K}} = -h_{\text{K}} \sum_{\mathbf{x}} \hat{G}_{\mathbf{x}} - J_{\text{K}} \sum_{\mathbf{x}^*} \hat{Z}_{\mathbf{x}^*} \quad (11.79)$$

Since, as argued below, all the individual terms appearing in this Hamiltonian *commute* — this is a so-called *frustration free Hamiltonian* — hence, for $h_{\text{K}}, J_{\text{K}} > 0$, the ground states of the model are *simultaneous eigenvectors* of all these (stabilizer) operators with eigenvalue +1:

$$\hat{G}_{\mathbf{x}} |\psi_{\text{gs}}\rangle = |\psi_{\text{gs}}\rangle \quad \forall \mathbf{x} \quad \text{and} \quad \hat{Z}_{\mathbf{x}^*} |\psi_{\text{gs}}\rangle = |\psi_{\text{gs}}\rangle \quad \forall \mathbf{x}^*. \quad (11.80)$$

^aKitaev calls A_s our $\hat{G}_{\mathbf{x}}$, s being a “star” associated to a site (or vertex) of the lattice, and B_p our $\hat{Z}_{\mathbf{x}^*}$, p being a “plaquette” associated to the center of the lattice squares. Since the two terms commute a common choice of parameters is to set $h_{\text{K}} = J_{\text{K}} = 1$, the energy unit.

¹¹In the language of lattice gauge theory, this would be the local gauge symmetry operator [54].

Plaquette and star operators as stabilizers. Clearly $\widehat{Z}_{\mathbf{x}_*}^2 = \mathbf{1}$ and $\widehat{G}_{\mathbf{x}}^2 = \mathbf{1}$, hence they have eigenvalues ± 1 . Plaquette operators and star operators commute among themselves, because they are made all of \mathbf{Z} or \mathbf{X} operators, respectively. Moreover, since $\widehat{G}_{\mathbf{x}}$ flips *pairs of spin* on the star, it is simple to show that $\widehat{G}_{\mathbf{x}}$ commutes with any plaquette operator, $[\widehat{G}_{\mathbf{x}'}, \widehat{Z}_{\mathbf{x}_*}] = 0$.

We will now describe the ground states of Kitaev’s toric code model in more detail.

11.10.1. The toric code ground states

Constraints and counting of ground states. As mentioned above, since $\widehat{G}_{\mathbf{x}}^2 = \mathbf{1}$ and $\widehat{Z}_{\mathbf{x}_*}^2 = \mathbf{1}$, both operators can have only eigenvalues ± 1 . The total number of states in the Hilbert space is $2^{2L^2} = 4^{N_V}$, where $N_V = L^2$ is the number of vertices, the factor 2 coming from the two links at each vertex. Imposing each of the constraints, for instance $\widehat{G}_{\mathbf{x}}|\psi\rangle = |\psi\rangle$, reduces by a factor 2 the number of states. One might think that there are $2N_V$ constraints implied by Eq. (11.80), one for each vertex \mathbf{x} and one for each plaquette \mathbf{x}_* (the number of plaquettes is equal to N_V in two dimensions). But in reality, when the system has PBC, you can show that

$$\prod_{\mathbf{x}} \widehat{G}_{\mathbf{x}} = 1 \quad \text{and} \quad \prod_{\mathbf{x}_*} \widehat{Z}_{\mathbf{x}_*} = 1$$

simply because each \mathbf{X} and \mathbf{Z} appear *twice*. Hence, we have only $(N_V - 1)$ constraints for the vertices and $(N_V - 1)$ constraints for the plaquettes. Therefore, we expect

$$\frac{4^{N_V}}{2^{N_V-1} 2^{N_V-1}} = 4 \quad \text{ground states .}$$

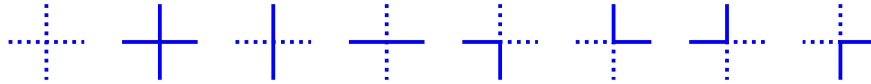


Figure 11.14.: The 8 vertices for which the constraint $\widehat{G}_{\mathbf{x}} = 1$ is satisfied. Dashed lines are links with a spin in state $|+, x\rangle$, while solid lines denote links with a spin in state $|-, x\rangle$.

To write the ground states explicitly, we start from the reference state $|+, x\rangle = \prod_{\ell} |+, x\rangle_{\ell}$ with all links on the spin state $|+, x\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle)$. Since this is an eigenstate of \mathbf{X} with eigenvalue $+1$, it satisfies $\widehat{G}_{\mathbf{x}}|+, x\rangle = |+, x\rangle$, but there are many more configurations that satisfy such constraints everywhere. By analysing all possible vertex configurations, $2^4 = 16$ in total, one quickly discovers that 8 of them indeed satisfy $\widehat{G}_{\mathbf{x}} = 1$ because an *even number* of lines in states $|-, x\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle)$ enter the vertex. The 8 vertices satisfying $\widehat{G}_{\mathbf{x}} = 1$ are shown in Fig. 11.14.

To construct the ground states, you can “glue together” the 8 types of vertices in Fig. 11.14 in such a way that *no vertex of the wrong type*, with an odd number of solid lines (spin in states $|-, x\rangle$), is present. A moment’s reflection should convince you that we need to impose that full lines always close into *loops* \mathcal{C} . Since a solid-line link is created by the application of $\hat{\sigma}_{\ell}^z$ on the state $|+, x\rangle_{\ell} - \hat{\sigma}_{\ell}^z |+, x\rangle_{\ell} = |-, x\rangle_{\ell}$ — you immediately realise that such a closed-loop of solid lines is precisely associated with the Wilson loop ¹² operator $\widehat{Z}_{\mathcal{C}}$:

$$\widehat{Z}_{\mathcal{C}} = \prod_{\ell \in \mathcal{C}} \hat{\sigma}_{\ell}^z, \tag{11.81}$$

which can be easily shown to be *gauge invariant*, which here simply means that they commute with any $\widehat{G}_{\mathbf{x}}$, again for the very simple reason that $\widehat{G}_{\mathbf{x}}$ flips pairs of spins on the loop.

¹²The smallest such contour is a plaquette $\text{plaq}(\mathbf{x}_*)$, and for these, you recognise the plaquette operator:

$$\widehat{Z}_{\mathcal{C}=\text{plaq}(\mathbf{x}_*)} \equiv \widehat{Z}_{\mathbf{x}_*} .$$

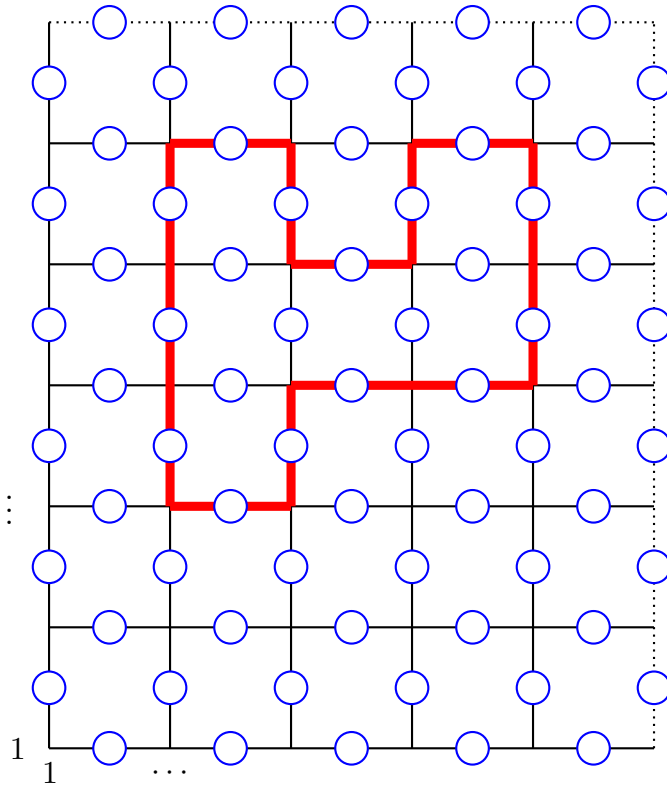


Figure 11.15: The Wilson loop. The closed contour \mathcal{C} (red thick line) contains links ℓ and the Wilson loop operator is defined by $\hat{Z}_{\mathcal{C}} = \prod_{\ell \in \mathcal{C}} \hat{\sigma}_{\ell}^z$. Equivalently, $\hat{Z}_{\mathcal{C}}$ can be seen as the product of all $\hat{Z}_{\mathbf{x}_*}$ plaquette operators where \mathbf{x}_* is *inside* the contour \mathcal{C} .

1

States satisfying $\hat{G}_{\mathbf{x}} = 1$ everywhere. You immediately conclude that any state

$$|\mathcal{C}\rangle = \hat{Z}_{\mathcal{C}}|+, x\rangle \quad \implies \quad \hat{G}_{\mathbf{x}}|\mathcal{C}\rangle = |\mathcal{C}\rangle \quad (11.82)$$

where \mathcal{C} is a closed loop of solid lines on the lattice, possibly made of several disconnected sub-loops. The simple reason for this result is that, as previously remarked, the Wilson loop operators are gauge invariant, $[\hat{G}_{\mathbf{x}}, \hat{Z}_{\mathcal{C}}] = 0$, and, trivially, $\hat{G}_{\mathbf{x}}|+, x\rangle = |+, x\rangle$. All these states $|\mathcal{C}\rangle$ are completely degenerate, for $J_K = 0$, and have energy $-N_V h_K$.

Let us now consider the effect of the other term, $-J_K \hat{Z}_{\mathbf{x}_*}$, in the toric code Hamiltonian. As discussed, $\hat{Z}_{\mathbf{x}_*} \equiv \hat{Z}_{\mathcal{C}=\text{plaq}(\mathbf{x}_*)}$, the smallest closed loop \mathcal{C} in the lattice: the plaquette. Hence, by applying $\hat{Z}_{\mathbf{x}_*}$ to a loop configuration $|\mathcal{C}\rangle$ you obtain *another loop configuration* $|\mathcal{C}'\rangle$ differing from $|\mathcal{C}\rangle$ by a single plaquette loop. All in all, you realise that the best thing you can do to gain energy — by the **Perron-Frobenius theorem** — is to create a *superposition of all loop configurations* with a uniform positive coefficient

$$|\Phi_{00}\rangle = \mathcal{N} \sum_{\mathcal{C}} \hat{Z}_{\mathcal{C}}|+, x\rangle = \frac{1}{\sqrt{2^{N_V-1}}} \sum_{\mathcal{C}} \hat{Z}_{\mathcal{C}}|+, x\rangle, \quad (11.83)$$

with a suitable normalisation coefficient, which one can show to be $\mathcal{N} = \frac{1}{\sqrt{2^{N_V-1}}}$. We can get this result also by an explicit route. Indeed, consider the alternative writing

$$|\Phi_{00}\rangle = \prod_{\mathbf{x}_*} \left(\frac{1 + \hat{Z}_{\mathbf{x}_*}}{\sqrt{2}} \right) |+, x\rangle = \frac{1}{\sqrt{2^{N_V}}} \sum_{\{n_{\mathbf{x}_*}\}} \prod_{\mathbf{x}_*} \hat{Z}_{\mathbf{x}_*}^{n_{\mathbf{x}_*}} |+, x\rangle \quad (11.84)$$

with $n_{\mathbf{x}_*} = 0, 1$ on each plaquette. Using the fact that $\prod_{\mathbf{x}_*} \hat{Z}_{\mathbf{x}_*} = 1$, you can show that all configurations are indeed included *twice*. The fact that, given a bipartition $A \cup \bar{A}$ of all plaquettes, we have

$$\prod_{\mathbf{x}_* \in A} \hat{Z}_{\mathbf{x}_*} = \hat{Z}_{\mathcal{C}}$$

completes the proof that Eqs. (11.83)-(11.84) are equivalent.

The other three ground states of the model on a 2-torus are obtained by applying non-contractible Wilson loop operators to $|\Phi_{00}\rangle$.

i

Non-contractible Wilson loop operators. We define γ_2 to be a closed straight contour of links that goes through the boundary of the 2-torus in the direction \mathbf{e}_2 , and define the associated Wilson loop operator

$$\widehat{Z}_2 \equiv \widehat{Z}_{\gamma_2} = \prod_{\ell \in \gamma_2} \hat{\sigma}_\ell^z. \quad (11.85)$$

By switching $\mathbf{e}_2 \leftrightarrow \mathbf{e}_1$, a similar definition can be given for γ_1 and associated operator $\widehat{Z}_1 \equiv \widehat{Z}_{\gamma_1}$.

i

Non-contractible 't Hooft loop operators On the dual lattice, we define γ_2^* to be a non-contractible loop that *cuts a line of parallel links* along the direction \mathbf{e}_2 — hence, running on the dual lattice defined by the plaquette centers — and the associated 't Hooft loop operator

$$\widehat{X}_2 \equiv \widehat{X}_{\gamma_2^*} = \prod_{\ell \perp \gamma_2^*} \hat{\sigma}_\ell^x. \quad (11.86)$$

In words, see Fig. 11.16: $\widehat{X}_{\gamma_2^*}$ creates a line of spin-flips on all parallel links in the direction \mathbf{e}_2 . By switching $\mathbf{e}_2 \leftrightarrow \mathbf{e}_1$, similar definitions can be given for γ_1^* , and associated operator $\widehat{X}_1 = \widehat{X}_{\gamma_1^*}$.

Since $\widehat{G}_\mathbf{x}$ always *flips spin in pairs* along any direction, while obviously commutes with $\hat{\sigma}^x$, it is easy to show that:

$$\widehat{G}_\mathbf{x}^\dagger \widehat{Z}_\nu \widehat{G}_\mathbf{x} = \widehat{Z}_\nu \quad \text{and} \quad \widehat{G}_\mathbf{x}^\dagger \widehat{X}_\nu \widehat{G}_\mathbf{x} = \widehat{X}_\nu \quad \forall \mathbf{x}. \quad (11.87)$$

However, since γ_2 and γ_2^* necessarily share a *single link* ℓ where $\hat{\sigma}_\ell^x \hat{\sigma}_\ell^z \hat{\sigma}_\ell^x = -\hat{\sigma}_\ell^z$ (and similarly for γ_1 and γ_1^*) you must have that:

$$\widehat{Z}_2 \widehat{X}_2 = -\widehat{X}_2 \widehat{Z}_2 \quad \text{and} \quad \widehat{Z}_1 \widehat{X}_1 = -\widehat{X}_1 \widehat{Z}_1. \quad (11.88)$$

On the contrary, γ_1 and γ_2^* share no link (and similarly γ_2 and γ_1^*) hence:

$$\widehat{Z}_1 \widehat{X}_2 = \widehat{X}_2 \widehat{Z}_1 \quad \text{and} \quad \widehat{Z}_2 \widehat{X}_1 = \widehat{X}_1 \widehat{Z}_2. \quad (11.89)$$

i

The two non-local Qbit operators. Summarizing, the four operators $(\widehat{Z}_1, \widehat{X}_1)$ and $(\widehat{Z}_2, \widehat{X}_2)$ both realise the Pauli algebra:

$$\{\widehat{Z}_1, \widehat{X}_1\} = 0 \quad \{\widehat{Z}_2, \widehat{X}_2\} = 0, \quad (11.90)$$

while they mutually commute:

$$[\widehat{Z}_1, \widehat{X}_2] = 0 \quad [\widehat{Z}_2, \widehat{X}_1] = 0. \quad (11.91)$$

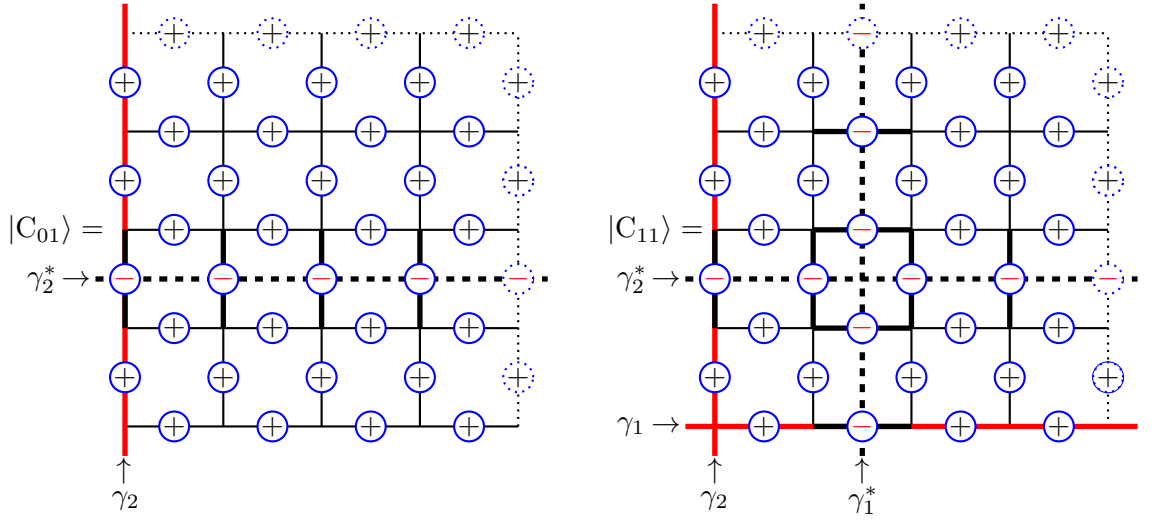


Figure 11.16.: Starting from a reference configuration $|C_{00}\rangle$ with all spins in state \uparrow , denoted by a $+$ inside a circle, the configurations $|C_{01}\rangle$ (left) and $|C_{11}\rangle$ (right) in a PBC system, where a $-$ inside a circle indicates a spin in state \downarrow . On the left, γ_2 (red solid line) is the non-contractible Wilson loop in the direction of \mathbf{e}_2 , associated to \widehat{Z}_{γ_2} , and γ_2^* (black dashed line) is the non-contractible 't Hooft loop cutting parallel links in the direction of \mathbf{e}_2 , associated to $\widehat{X}_{\gamma_2^*}$. On the right, the corresponding γ_1 and γ_1^* in the other direction are shown. Here \pm denote spin components along the z -axis, i.e., in the standard configuration basis. The dotted circles denote spins on links on the boundary.

1

The four ground states on the 2-torus. Define, for $v_1, v_2 = 0, 1$, the four (degenerate) ground states

$$|\Phi_{v_1, v_2}\rangle = \widehat{Z}_2^{v_2} \widehat{Z}_1^{v_1} |\Phi_{0,0}\rangle. \quad (11.92)$$

The non-contractible 't Hooft operators act now as non-local order parameters:

$$\widehat{X}_\nu |\Phi_{v_1, v_2}\rangle = (-1)^{v_\nu} |\Phi_{v_1, v_2}\rangle \quad \implies \quad \langle \Phi_{v_1, v_2} | \widehat{X}_\nu | \Phi_{v_1, v_2} \rangle = (-1)^{v_\nu}. \quad (11.93)$$

The four ground states of the Kitaev toric code model on the 2-torus effectively realize an encoding of *two Qbits*, robust against local perturbations, which would immediately generate frustrated plaquettes, $\widehat{Z}_{\mathbf{x}_*} = -1$, or stars where $\widehat{G}_{\mathbf{x}} = -1$.

Obviously, on such states, the expectation value of any Wilson loop operator is exactly 1:

$$\langle \Phi_{v_1, v_2} | \widehat{Z}_{\mathcal{C}} | \Phi_{v_1, v_2} \rangle = 1, \quad (11.94)$$

for the very simple reason that any $\widehat{Z}_{\mathcal{C}}$ can be written as a product of $\widehat{Z}_{\mathbf{x}_*}$, and all of these have eigenvalues $+1$ on the ground states.

1

The Toric code as a topological stabilizer code. Summarizing, on an $L \times L$ lattice with PBC, with $n = 2L^2$ physical spins, we have been able to encode $k = 2$ logical Qbits. Moreover, the distance d of the Toric code, by definition the minimum weight of a Pauli string that commutes with all stabilizers but does not belong to the stabilizer group, is $d = L$, the dangerous Pauli strings being precisely the non-contractible Wilson loop operators.

Part I.
Appendices

A. Simple tools from arithmetics

A.1. The Euclid algorithm for the greatest common divisor

The problem is to find the greatest common divisor of two integers a and b , denoted by $\gcd(a, b)$. The algorithm is based on the division, with an integer remainder, which we learn in primary school. Assume that $a > b$. Then divide a by b , finding a quotient q_0 and an integer remainder $r_0 < b$:

$$a = b q_0 + r_0 \quad \text{with} \quad q_0 = \left\lfloor \frac{a}{b} \right\rfloor \quad \text{and} \quad r_0 = a - q_0 b ,$$

where $\lfloor x \rfloor$ denotes the integer part of x . Do the same for the division of b by the remainder found, r_0 :

$$b = r_0 q_1 + r_1 \quad \text{with} \quad q_1 = \left\lfloor \frac{b}{r_0} \right\rfloor \quad \text{and} \quad r_1 = b - q_1 r_0 ,$$

where now $r_1 < r_0$. Continue in this way with integer divisions of the (decreasing) remainders we find, until we get a remainder that vanishes. More precisely, at step k of the algorithm we write:

$$\text{Step } k : \quad r_{k-2} = r_{k-1} q_k + r_k \quad \text{with} \quad \begin{cases} q_k = \left\lfloor \frac{r_{k-2}}{r_{k-1}} \right\rfloor \\ r_k = r_{k-2} - r_{k-1} q_k \end{cases} , \quad (\text{A.1})$$

where r_{k-1} and r_{k-2} are inputs from previous steps, while the new quotient q_k and remainder r_k are calculated. With these recursive equations, you see that you need to pose $r_{-2} = a$ and $r_{-1} = b$ to recover step 0. Since $0 \leq r_k < r_{k-1}$ the algorithm necessarily stops at some step $k = K$ where you find a vanishing remainder, $r_K = 0$.

i

Euclid's algorithm for $\gcd(a, b)$. The *last remainder before stopping* is $\gcd(a, b)$:

$$r_{K-1} = \gcd(a, b) . \quad (\text{A.2})$$

The crucial point in the proof is that r_{K-1} is a divisor of r_{K-2} , since $r_{K-2} = r_{K-1} q_K$ without remainder. r_{K-1} also divides its next predecessor r_{K-3} because:

$$r_{K-3} = r_{K-2} q_{K-1} + r_{K-1} .$$

Proceeding backwards, you realise that r_{K-1} divides both a and b , hence $r_{K-1} \leq \gcd(a, b)$. By a simple argument, you can show that any integer c that divides both a and b , must divide the initial remainder r_0 and all the subsequent ones, including r_{K-1} . Hence $r_{K-1} = \gcd(a, b)$.

One can estimate the speed with which the iterations get to the final result: quite fast. With some improvements Gabriel Lamè showed in 1844 that $K \leq 5N_{\text{digits}}$ where $N_{\text{digits}} = n_{\text{bits}}/\log 10$ is the number of decimal digits of $\min(a, b)$. This finding marks the beginning of the studies in **computational complexity theory**.

Example.

Take $a = 1071$ and $b = 462$. Here is a table with the various steps:

Step k		
0	$1071 = 462 q_0 + r_0$	$q_0 = 2, r_0 = 147$
1	$462 = 147 q_1 + r_1$	$q_1 = 3, r_1 = 21$
K \rightarrow 2	$147 = 21 q_2 + r_2$	$q_2 = 7, r_2 = 0$

Hence:

$$r_1 = 21 = \gcd(1071, 462).$$



Euclid vs factorisation. In primary school they teach us to calculate the $\gcd(a, b)$ by a prime factorisation of a and b , taking then the “*common prime factors, with smallest exponent*”. But prime factorisation is a *difficult* problem, while Euclid’s algorithm is *easy*.



Co-prime numbers. A useful definition. You say that a and b are *co-prime* if $\gcd(a, b) = 1$.

A.2. Finding the multiplicative inverse in modular arithmetics



The multiplicative inverse in modular arithmetics. We show that the multiplicative inverse in modulo- a arithmetics of a number b *co-prime* with a , i.e., the integer c such that:

$$cb \equiv 1 \pmod{a} \quad \text{with} \quad \gcd(a, b) = 1, \quad (\text{A.3})$$

can be easily found by applying Euclid’s algorithm.

Indeed, let us apply Euclid’s algorithm. Since $\gcd(a, b) = 1$, at step $K - 1$ of the algorithm we get $r_{K-1} = 1$, before getting $r_K = 0$ at the final step K . We know all the r_k and q_k from the algorithm. Then we write the following chain of equalities:

$$\begin{aligned} 1 = r_{K-1} &= r_{K-3} - r_{K-2} q_{K-1} \\ r_{K-2} &= r_{K-4} - r_{K-3} q_{K-2} \\ r_{K-3} &= r_{K-5} - r_{K-4} q_{K-3} \\ &\vdots \\ r_1 &= b - r_0 q_1 \\ r_0 &= a - b q_0 \end{aligned}$$

This means that, by iterating the equalities in the chain, I can find two integers j_a and j_b such that:

$$1 = j_a a + j_b b \quad \text{with} \quad j_a, j_b \in \mathbb{Z}. \quad (\text{A.4})$$

Observe that j_b *cannot be a multiple* of a . Hence a non-vanishing remainder $c < a$ can be found by dividing j_b by a :

$$j_b = la + c \quad \text{with} \quad l \in \mathbb{Z}, \quad 1 \leq c < a.$$

Substituting, we get:

$$1 = j_a a + (la + c)b = \underbrace{(j_a + lb)a}_{\equiv 0 \pmod{a}} + cb \equiv cb \pmod{a}, \quad (\text{A.5})$$

hence c is the desired multiplicative inverse \pmod{a} of b .

A.3. The probability of two random integers being co-prime

i

Euler Basel's problem. Here is one of the many brilliant findings by Euler. The probability that two random integers a and b are *co-prime*, hence that $\gcd(a, b) = 1$, is:

$$\text{Prob}_{\gcd(a,b)=1} = \frac{6}{\pi^2} \sim 0.6079. \quad (\text{A.6})$$

We test for the probability of all prime factors one after the other. The probability of a number being multiple of 2 is $\frac{1}{2}$, the probability that *both* are multiple of 2 is $\frac{1}{4}$, hence:

$$\text{Prob}_{2 \text{ not common factor}} = 1 - \frac{1}{4} = 1 - \frac{1}{2^2}.$$

Now we test by 3. The probability of a number being multiple of 3 is $\frac{1}{3}$, the probability that *both* are multiple of 3 is $\frac{1}{3^2}$, hence:

$$\text{Prob}_{3 \text{ not common factor}} = 1 - \frac{1}{3^2}.$$

By the same argument you discover that:

$$\text{Prob}_{5 \text{ not common factor}} = 1 - \frac{1}{5^2},$$

and so forth. So:

$$\text{Prob}_{\gcd(a,b)=1} = \prod_p^{\text{all primes}} \left(1 - \frac{1}{p^2}\right). \quad (\text{A.7})$$

Now, in the typical Euler's approach, you expand the inverse:

$$\frac{1}{1 - \frac{1}{p^2}} = 1 + \frac{1}{p^2} + \frac{1}{p^4} + \dots.$$

Hence:

$$\frac{1}{\text{Prob}_{\gcd(a,b)=1}} = \left(1 + \frac{1}{2^2} + \frac{1}{2^4} + \dots\right) \left(1 + \frac{1}{3^2} + \frac{1}{3^4} + \dots\right) \left(1 + \frac{1}{5^2} + \frac{1}{5^4} + \dots\right) \dots,$$

where the product should include all primes. Euler realised that, multiplying term by term, you get in fact a much simpler object:

$$\frac{1}{\text{Prob}_{\gcd(a,b)=1}} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6},$$

where perhaps you recognise the Riemann $\zeta(2)$. Taking the inverse, we are done:

$$\text{Prob}_{\gcd(a,b)=1} = \frac{6}{\pi^2}. \quad (\text{A.8})$$

B. Uniaxial birefringence

The following Appendix is intended to provide you a small guide into the world of optics phenomena that have to do with the polarisation of photons. It is also a useful companion guide in case you plan to read the wonderful book on Quantum Mechanics written by Asher Peres [29], whose first chapter deals with what we are going to illustrate.

Let me start by recalling a few basic facts concerning classical electromagnetic waves in a medium.

i Maxwell's equations in a medium.

$$\left\{ \begin{array}{l} \nabla \cdot \mathbf{D} = \rho_f \\ \nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t} \end{array} \right. \quad \left\{ \begin{array}{l} \nabla \cdot \mathbf{B} = 0 \\ \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \end{array} \right. . \quad (\text{B.1})$$

i Constitutive relations between fields.

The fields are related as:

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P} = \bar{\varepsilon} \cdot \mathbf{E} \quad \mathbf{B} = \mu_0 \mathbf{H} + \mathbf{M} = \bar{\mu} \cdot \mathbf{H} , \quad (\text{B.2})$$

where ε_0 and μ_0 are the so-called vacuum permittivity and permeability, with $\varepsilon_0 \mu_0 = 1/c^2$, and the second equality applies in the *linear* regime. These relationships are better interpreted in Fourier space, where the matrices $\bar{\varepsilon}$ and $\bar{\mu}$ are generally (\mathbf{k}, ω) dependent: the usual assumption is that in the optical range the microscopic crystalline structure is not important, since the wave-length of light is much larger than the interatomic distances, and one recovers space translational symmetry. In a non-magnetic material, one often assumes $\bar{\mu} = \mu_0 \mathbf{1}$. Further, for dielectric materials one usually takes $\rho_f = 0$, $\mathbf{J}_f = 0$: free charges and currents are absent. In a lossless dielectric material one assumes $\bar{\varepsilon}(\omega)$ to be real in the (optical) frequency range of interest, although general causality requirements forbid a dielectric function from being real for all frequencies.

In crystals which are *optically anisotropic* $\bar{\varepsilon}$ is a matrix. A simple but very important case is the of *uniaxial crystals* where $\bar{\varepsilon}$ has a special axis — the *optic axis* — which is different from the other two. Assuming that we work in a definite (optical) range of \mathbf{k} and ω , where the dielectric properties can be taken to be constant, we parameterise the eigenvalues of $\bar{\varepsilon}$ as $\varepsilon_0 n_e^2$, along the optic (extraordinary) axis, and $\varepsilon_0 n_o^2$ along the other two (ordinary) axes. The two values n_e and n_o refer to the corresponding *index of refraction*, as we will see. Taking $\hat{\mathbf{y}}$ to be the optic axis, we parameterise:

$$\bar{\varepsilon} = \varepsilon_0 \begin{pmatrix} n_o^2 & 0 & 0 \\ 0 & n_e^2 & 0 \\ 0 & 0 & n_o^2 \end{pmatrix} . \quad (\text{B.3})$$

The most noteworthy uniaxial crystals are summarised here: ¹

	n_o	n_e	Δn
Calcite (CaCO_3)	1.658	1.486	-0.172
Barium borate (BaB_2O_4)	1.6776	1.5534	-0.1242
Quartz (SiO_2)	1.544	1.553	+0.009

(B.4)

❶

The electric field in an optically anisotropic crystal.

The Ampère-Maxwell equation, written in terms of \mathbf{B} and \mathbf{D} reads:

$$\nabla \times \mathbf{B} = \mu_0 \frac{\partial \mathbf{D}}{\partial t} .$$

Taking the curl of the Faraday's equation we get:

$$\nabla \times (\nabla \times \mathbf{E}) = -\frac{\partial}{\partial t} (\nabla \times \mathbf{B}) = -\mu_0 \frac{\partial^2 \mathbf{D}}{\partial t^2} .$$

In Fourier transform, $\nabla \rightarrow i\mathbf{k}$ and $\partial_t \rightarrow -i\omega$, therefore:

$$\mathbf{k}(\mathbf{k} \cdot \mathbf{E}) - k^2 \mathbf{E} = -\mu_0 \omega^2 \bar{\epsilon} \cdot \mathbf{E} , \quad (\text{B.5})$$

where we used that $\mathbf{D} = \bar{\epsilon} \cdot \mathbf{E}$.

We now consider specifically the uniaxial case, and set our axes such that \mathbf{k} lays in the $y-z$ plane ($k_x = 0$), at an angle θ with the z axis. Figure B.1 shows our setting. The explicit form of Eq. (B.5)

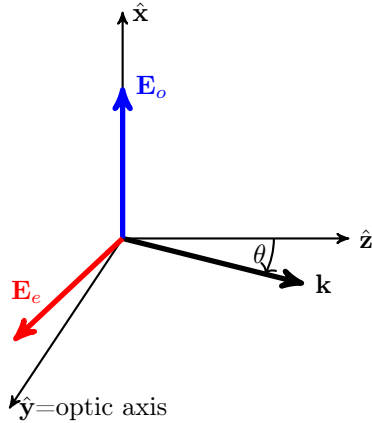


Figure B.1: An electromagnetic plane-wave travelling along \mathbf{k} inside a uniaxial crystals. The ordinary ray, with polarisation \mathbf{E}_o , and the extra-ordinary ray, with polarisation \mathbf{E}_e , are shown, both orthogonal to the propagation direction \mathbf{k} .

reads, with our choices:

$$\begin{pmatrix} -k^2 + \frac{\omega^2}{c^2} n_o^2 & 0 & 0 \\ 0 & -k^2 + k_y^2 + \frac{\omega^2}{c^2} n_e^2 & k_y k_z \\ 0 & k_y k_z & -k^2 + k_z^2 + \frac{\omega^2}{c^2} n_e^2 \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} = 0 . \quad (\text{B.6})$$

The equation for E_x is decoupled, and shows that a wave polarised along $\hat{\mathbf{x}}$, with $\epsilon_{\mathbf{k},o} = (1, 0, 0)^T$ would satisfy the equation $\omega^2 = \frac{k^2 c^2}{n_o^2}$. So, for given ω , the wave-vector k is larger by a factor n_o than the wave-vector in vacuum: $k_0 = \frac{\omega}{c}$. For electric field in the $y-z$ plane, we see that there is

¹ β -BaB₂O₄, often abbreviated as BBO, is crucial in most of the papers dealing with generation of polarisation entangled photon pairs. What one exploits in that context is the crucial fact that the indices of refractions are indeed *frequency dependent* over the whole optical range: they decrease for increasing wave-length. This is used in many non-linear phenomena of quantum optics, most notably in parametric down-conversion.

an *unphysical longitudinal* solution with $\mathbf{E} \propto \mathbf{k}$, for $\omega = 0$, which we discard, and a second physical *transverse* solution with $\boldsymbol{\epsilon}_{\mathbf{k},e} = \frac{\mathbf{k}}{k} \times \boldsymbol{\epsilon}_{\mathbf{k},o}$ with a k vs ω that depends on the angle of propagation θ . To summarise:

i **Ordinary and extra-ordinary rays.**

Expressing our results in terms of the wave-vector in vacuum k_0 , i.e., setting $\omega = k_0 c$, we write the two physical transverse solutions for the electric field propagating inside the uniaxial crystal, with our choice of axes, as:

$$\begin{cases} \text{Ordinary ray: } k = k_0 n_o & \boldsymbol{\epsilon}_{\mathbf{k},o} = (1, 0, 0)^T \\ \text{Extra-ordinary ray: } k = k_0 n_\theta & \boldsymbol{\epsilon}_{\mathbf{k},e} = (0, -\cos \theta, \sin \theta)^T \end{cases} \quad (\text{B.7})$$

where the effective angle-dependent index of refraction n_θ is given by the ellipse equation illustrated in Fig. B.2:

$$\frac{1}{n_\theta^2} = \frac{\cos^2 \theta}{n_e^2} + \frac{\sin^2 \theta}{n_o^2}. \quad (\text{B.8})$$

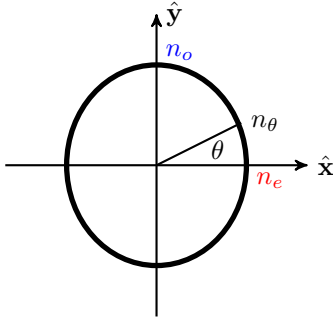


Figure B.2: The ellipse of the index of refraction. For $\theta = 0$ we have $n_{\theta=0} = n_e$. As θ increases the index of refraction goes towards the ordinary value n_o , obtained for $\theta = \frac{\pi}{2}$, corresponding to a wave propagating along the optic axis $\hat{\mathbf{y}}$, and therefore polarised along the ordinary axis $\hat{\mathbf{z}}$.

The previous considerations are easy to generalise to an arbitrary direction of \mathbf{k} . Indeed, taking $\mathbf{k} = k(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, one verifies that the polarisation vectors of the ordinary and extra-ordinary rays are:

$$\begin{cases} \boldsymbol{\epsilon}_{\mathbf{k},o} = \frac{1}{\sqrt{\cos^2 \theta + \sin^2 \theta \cos^2 \phi}} (\hat{\mathbf{x}} \cos \theta - \hat{\mathbf{z}} \sin \theta \cos \phi) \\ \boldsymbol{\epsilon}_{\mathbf{k},e} = \frac{1}{\sqrt{\cos^2 \theta + \sin^2 \theta \cos^2 \phi}} (-\hat{\mathbf{x}} \sin^2 \theta \sin \phi \cos \phi + \hat{\mathbf{y}} (\cos^2 \theta + \sin^2 \theta \cos^2 \phi) - \hat{\mathbf{z}} \sin \theta \cos \theta \sin \phi) = \frac{\mathbf{k}}{k} \times \boldsymbol{\epsilon}_{\mathbf{k},o} \end{cases} \quad (\text{B.9})$$

The Maxwell's equation Eq. (B.5) implies, for a transverse wave, that the polarisation vectors should satisfy:

$$k^2 \boldsymbol{\epsilon}_{\mathbf{k}} = k_0^2 \frac{1}{\epsilon_0} \bar{\boldsymbol{\epsilon}} \cdot \boldsymbol{\epsilon}_{\mathbf{k}}.$$

$\boldsymbol{\epsilon}_{\mathbf{k},o}$ is an eigenvector of $\bar{\boldsymbol{\epsilon}}$ with eigenvalue $\epsilon_0 n_o^2$, hence $k = k_0 n_o$ follows immediately for the ordinary ray. For the extra-ordinary ray, we first take the inverse of the previous equation, and then multiply both terms by $\boldsymbol{\epsilon}_{\mathbf{k}}$, obtaining:

$$k^2 \boldsymbol{\epsilon}_{\mathbf{k}} \cdot \left(\frac{1}{\epsilon_0} \bar{\boldsymbol{\epsilon}} \right)^{-1} \cdot \boldsymbol{\epsilon}_{\mathbf{k}} = k_0^2 \boldsymbol{\epsilon}_{\mathbf{k}} \cdot \boldsymbol{\epsilon}_{\mathbf{k}} = k_0^2.$$

The LHS defines precisely the ellipsoid relationship for the effective index of refraction $n_{\mathbf{k}}$ of the extra-ordinary ray:

$$\boldsymbol{\epsilon}_{\mathbf{k}} \cdot \left(\frac{1}{\epsilon_0} \bar{\boldsymbol{\epsilon}} \right)^{-1} \cdot \boldsymbol{\epsilon}_{\mathbf{k}} \stackrel{\text{def}}{=} \frac{1}{n_{\mathbf{k}}^2} \implies k = k_0 n_{\mathbf{k}}.$$

Explicitly, we have:

$$\frac{1}{n_{\mathbf{k}}^2} = \frac{\cos^2 \theta + \sin^2 \theta \cos^2 \phi}{n_e^2} + \frac{\sin^2 \theta \sin^2 \phi}{n_o^2}, \quad (\text{B.10})$$

which reduces to Eq. (B.8) when \mathbf{k} is in the y - z plane ($\phi = \frac{\pi}{2}$).

B.1. The wave-plate geometry

One of the most interesting applications of uniaxial crystals is in a planar geometry where the wave enters the crystal orthogonally to the surface, and the optic axis is along the surface. This means that the wave suffers *no refraction*, and propagates along the \hat{z} axis, with $\theta = 0$. Fig. B.3 illustrates the wave-plate geometry, with L the thickness of the crystal along the propagation direction.

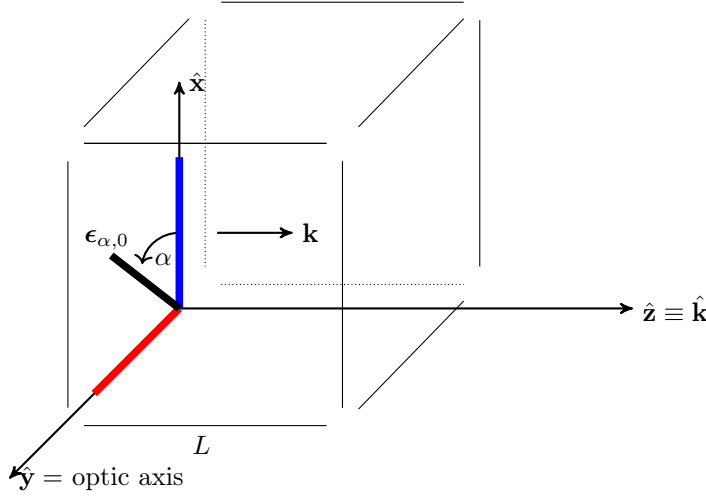


Figure B.3: An electromagnetic plane-wave travelling along \mathbf{z} , with an incoming electric field $\mathbf{E} = \text{Re} \epsilon_{\alpha,0} e^{i(k_0 z - \omega t)}$ linearly polarised along $\epsilon_{\alpha,0} = \hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha$, entering a uniaxial crystal of thickness L orthogonally to the surface where the optic axis lays. When exiting the polarisation is generally complex: $\epsilon_{\alpha,\delta} = \hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} e^{i\delta} \sin \alpha$, with $\delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0}$.

For $z < 0$ the plane-wave electromagnetic field is given by $\mathbf{E} = \text{Re} \epsilon_{\alpha} e^{i(k_0 z - \omega t)}$, where the transverse polarisation ϵ_{α} forms an angle α with respect to the $\hat{\mathbf{x}}$ axis in the xy -plane containing the optic axis: $\epsilon_{\alpha} = \hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha$. Inside the crystal, the electric field component along the ordinary direction $\hat{\mathbf{x}}$ travels with a wave-vector $k_0 n_o$, while the component along the optic axis $\hat{\mathbf{y}}$ travels with wave-vector $k_0 n_e$. This implies that

$$\left\{ \begin{array}{ll} \text{For } z < 0 : & \mathbf{E} = \text{Re} \left((\hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha) e^{ik_0(z-ct)} \right) \\ \text{For } 0 < z < L : & \mathbf{E} = \text{Re} \left(\hat{\mathbf{x}} \cos \alpha e^{ik_0 n_o(z - \frac{c}{n_o} t)} + \hat{\mathbf{y}} \sin \alpha e^{ik_0 n_e(z - \frac{c}{n_e} t)} \right) \\ \text{For } z \geq L : & \mathbf{E} = \text{Re} \left((\hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha e^{ik_0 L(n_e - n_o)}) e^{ik_0(z-L-ct) + ik_0 L n_o} \right) \end{array} \right. \quad . \quad (\text{B.11})$$

The second expression shows very clearly that the two components travel inside the crystal with different velocities and have different wavelengths. The expression, for $z \geq L$, back in vacuum, shows that the polarisation vector is in general *no longer real*, with an extra phase which will make the field to “spiral” as it travels further.

❶

Elliptical polarisation.

An elliptical polarisation is obtained on exit from the crystal:

$$\epsilon_{\alpha,\delta} = \hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} e^{i\delta} \sin \alpha \quad \text{with} \quad \delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0}, \quad (\text{B.12})$$

where $\lambda_0 = \frac{2\pi}{k_0}$ is the wave-length in vacuum.

As you see, the final polarisation depends on the difference between the two refractive indices, and on the ratio between L , the thickness of the crystal, and λ_0 , the wave-length of the radiation in vacuum. Two cases are particular noteworthy. The first is known as *quarter-wave-plate*.

i

Quarter-wave plates (QWP).

In a quarter-wave-plate L is such that:

$$\delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0} = \pm \frac{\pi}{2}. \quad (\text{B.13})$$

The \hat{x} and \hat{y} components of the field now advance out-of-phase by $\frac{\pi}{2}$. In the particularly important case in which the original polarisation was perfectly *diagonal*, $\alpha = \frac{\pi}{4}$, the exit polarisation is circular:

$$\epsilon_{\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{y}) \quad \rightarrow \quad \epsilon_{\frac{\pi}{4},\pm\frac{\pi}{2}} = \frac{1}{\sqrt{2}}(\hat{x} \pm i\hat{y}). \quad (\text{B.14})$$

The second quite important case is that of a *half-wave-plate*.

i

Half-wave plates (HWP).

In a half-wave-plate L is twice as much as in the corresponding QWP:

$$\delta = 2\pi(n_e - n_o) \frac{L}{\lambda_0} = \pi. \quad (\text{B.15})$$

The \hat{y} component is precisely reversed, hence $\alpha \rightarrow -\alpha$. In the particularly important case in which the original polarisation was perfectly *diagonal*, $\alpha = \frac{\pi}{4}$, the exit polarisation is anti-diagonal:

$$\epsilon_{\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{y}) \quad \rightarrow \quad \epsilon_{-\frac{\pi}{4},0} = \frac{1}{\sqrt{2}}(\hat{x} - \hat{y}). \quad (\text{B.16})$$

Practical remarks

- 1) The so-called *zero-order* plates have L such that $\delta = 2\pi(\Delta n) \frac{L}{\lambda_0}$ with $|\delta| \leq 2\pi$, where $\Delta n = n_e - n_o$. For $|\Delta n| = 0.172$, as for calcite, this requires extremely thin plates, hard to fabricate:

$$L \leq \frac{\lambda_0}{|\Delta n|} \sim 10\lambda_0 \quad \text{for Calcite.}$$

The case of Quartz, where $\Delta n = 0.009$, is much easier to deal with:

$$L \leq \frac{\lambda_0}{|\Delta n|} \sim 100\lambda_0 \quad \text{for Quartz.}$$

- 2) Even full-wave-plates find their usefulness, for instance in minarology, where a plate with $\delta = 2\pi$ for $\lambda_0 = 540$ nm (green light) is such that only such green light remains linearly polarised, all other wave-lengths becoming elliptically polarised. With a polaroid filter, one could therefore eliminate the $\lambda = \lambda_0$ component.
- 3) Multiple order wave-plates are thicker and therefore simpler to build. With two wave-plates of, for instance, $37\frac{\pi}{2}$ and $36\frac{\pi}{2} = 18\pi$, glued in *opposite direction*, one can create a QWP.

B.2. The double-refraction geometry

Depending on the relative orientation of beam axis, and its polarisation, with respect to the optic axis of the birefringent crystal, a classical beam is seen to be “split into two beams” of different polarisations, due to the different Snell’s-law-induced refraction.

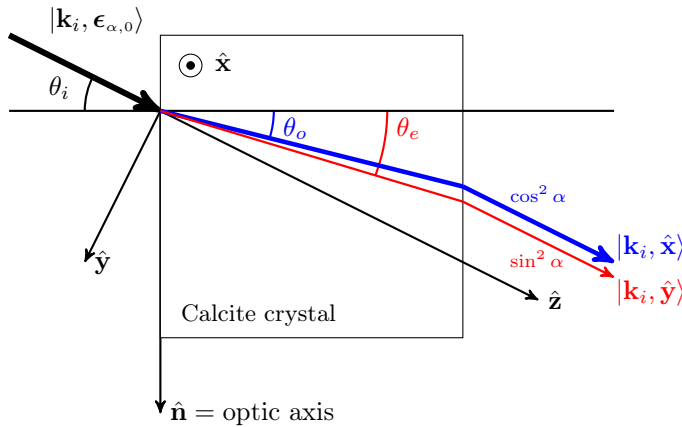


Figure B.4: Polarisation-dependent refraction of a photon passing through a calcite crystal. We are assuming that the optic axis of the crystal is parallel to the crystal surface, as you would have for a wave-plate. The incoming momentum \mathbf{k}_i is now tilted at an angle θ_i with respect to the surface normal, provoking refraction of the incoming wave. Notice the orientation of the axes: the optic axis is now denoted by $\hat{\mathbf{n}}$. The convention is identical to that of Ref. [29][Fig. 1.2].

The geometry is now slightly more complex. Figure B.4 illustrates the phenomenon of double-refraction from a top-viewpoint. An ordinary ray, polarised along $\hat{\mathbf{x}}$ — out of the page in Fig. B.4 — would have a normal refraction. According to Snell’s law, its angle of refraction θ_o would be:

$$\sin \theta_i = n_o \sin \theta_o, \quad (\text{B.17})$$

and it would come out still linearly polarised along $\hat{\mathbf{x}}$. Imagine now an incoming wave with linear polarisation $\epsilon_{\alpha,0} = \hat{\mathbf{x}} \cos \alpha + \hat{\mathbf{y}} \sin \alpha$. Let us consider first the case $\alpha = \frac{\pi}{2}$. The wave has a component along the optic axis $\hat{\mathbf{n}}$, but no component along $\hat{\mathbf{x}}$. It suffers a refraction with an effective index of refraction n_θ . Snell’s law would require:

$$\sin \theta_i = n_{\theta_e} \sin \theta_e. \quad (\text{B.18})$$

Notice that the refracting index is itself a function of the angle θ_e , which is therefore slightly more complex to calculate. In particular, n_{θ_e} should not be confused with n_e : recall the ellipse of the index of refraction in Fig. B.2. The outgoing beam would still be linearly polarised as the incoming beam.

The interesting situation is that of an incoming beam with a linear polarisation having a generic α . The two components, that along $\hat{\mathbf{x}}$, of amplitude $\cos \alpha$, and the other along $\hat{\mathbf{y}}$, of amplitude $\sin \alpha$, now suffer two different refractions with angles θ_o and θ_e , and come out, for a *sufficiently thick* crystal, as two separate beams, each of them linearly polarised as in input, with classical intensities partitioned as $\cos^2 \alpha$ and $\sin^2 \alpha$.

The first interesting question is what happens at the *quantum level*, for such a thick crystal. A photon of energy $\hbar\omega$ cannot be split in two: this is a linear optics setup. What happens, quantum mechanically, is that a photon *either goes* in the beam polarised as $\hat{\mathbf{x}}$, with probability $\mathbb{P}_\alpha = \cos^2 \alpha$, *or* in the beam polarised as $\hat{\mathbf{y}}$, with probability $1 - \mathbb{P}_\alpha = \sin^2 \alpha$. If you put two separate detectors you can count and measure the individual photons arriving. If you include a coincidence measurement electronics, you would see that no coincidence events are measured, if single photon states are sent into the “measuring device”.²

i

Thick uniaxial crystals as linear polarisation analysers.

It is clear that a thick uniaxial crystal acts as a *Stern-Gerlach* device for the polarisation. Contrary to the case of a polaroid filter, here no photon is absorbed. The outgoing photon, however, is “measured” into the two different linear polarisations associated to the crystal: $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, the latter having a component along the optic axis $\hat{\mathbf{n}}$.

²For a discussion regarding the difference between a *single photon state* and a very weak laser beam, see A. Aspect’s public lecture upon receiving the N. Bohr Gold Medal 2013, available on [YouTube](#).

The second interesting part of the story is what happens when the crystal is *not thick enough*, so that the two beams, classically, have an *overlap region*. Classically, the electromagnetic field has an elliptical polarisation in the overlap region, see [29][Fig. 1.3], as known, long before Maxwell, from studies by Arago and Fresnel on interference of polarised light.³ Quantum mechanically, the photon is never a *mixture* of linearly polarised photons: it would be in a pure state of elliptical polarisation. The apparatus, however, does not act as a “measuring device” in the ordinary sense, because it is not able to clearly discriminate the different “beams”.

B.3. Quantum optics single-Qbit gates with photon polarisation

The two polarisation states of a photon provide a good way to encode a Qbit in Quantum Optics implementation. With our Qbit computational states we would write:

$$\mathbf{Z} - \text{states: } \begin{cases} |0\rangle = |\uparrow\rangle \mapsto |\downarrow\rangle \\ |1\rangle = |\downarrow\rangle \mapsto |\uparrow\rangle \end{cases} \quad \mathbf{X} - \text{states: } \begin{cases} \mathbf{H}|0\rangle = |+, \mathbf{x}\rangle \mapsto |\nearrow\rangle \\ \mathbf{H}|1\rangle = |-, \mathbf{x}\rangle \mapsto |\searrow\rangle \end{cases} . \quad (\text{B.19})$$

Clearly, there is an intrinsic arbitrariness in what you call $|\downarrow\rangle$ and $|\leftrightarrow\rangle$, which for us would be $|\downarrow\rangle = |\epsilon = \hat{\mathbf{x}}\rangle$ and $|\leftrightarrow\rangle = |\epsilon = \hat{\mathbf{y}}\rangle$, with the previous convention: people sometimes call them $|H\rangle$ (horizontal) and $|V\rangle$ (vertical), respectively. The arbitrariness is removed once you start using birefringent crystals for measurements (in a double-refraction geometry), or for manipulating the polarisation state (in a wave-plate geometry).

i General elliptical polarisation.

A photon Qbit in a state of general elliptical polarisation

$$|\epsilon_{\alpha,\delta}\rangle = \cos \alpha |\downarrow\rangle + e^{i\delta} \sin \alpha |\leftrightarrow\rangle \quad (\text{B.20})$$

represents a computational Qbit in the usual Bloch sphere of spin-1/2 states.

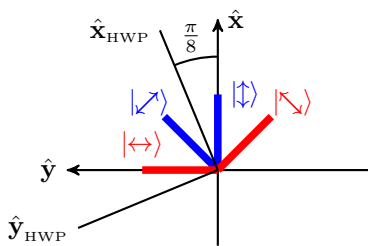


Figure B.5: A half-wave-plate (HWP) used as a Hadamard gate. Recall that a HWP transforms $\alpha \rightarrow -\alpha$, when a polarisation is expressed in terms of the HWP axes: $\epsilon = \hat{\mathbf{x}}_{\text{HWP}} \cos \alpha + \hat{\mathbf{y}}_{\text{HWP}} \sin \alpha$. Here $\hat{\mathbf{y}}_{\text{HWP}}$ denotes the *optic axis* of the HWP.

We already saw that a diagonally polarised photon, $|\nearrow\rangle$, is turned anti-diagonal, $|\searrow\rangle$, by a half-wave plate (HWP). This assumed that the optic axis of the HWP is along $\hat{\mathbf{y}}$, as in all our figures so far. But suppose that a photon linearly polarised as $|\downarrow\rangle$ goes through a HWP which has its “ $\hat{\mathbf{x}}$ -axis” tilted at an angle $\alpha = \pi/8$ with respect to the photon polarisation: that would invert a component of the photon polarisation, transforming $|\downarrow\rangle$ into a $|\nearrow\rangle$ -photon. Viceversa, the $|\leftrightarrow\rangle$ -photon would be transformed into $|\searrow\rangle$. See Fig. B.5. Hence:

i The half-wave plate acting like a Hadamard. An appropriately oriented HWP, tilted by $\frac{\pi}{8}$ with respect to the standard $|\downarrow\rangle$, acts as Hadamard gate for polarisation encoded optical Qbits.

³F. Arago and A. Fresnel, *Ann. de Chimie et Physique* **10**, 288 (1819).

Arbitrary single-Qbit rotations in the “Bloch sphere” can be implemented by a sequence of QWP-HWP-QWP.

B.4. Hands-on: Peres’ problems with calcite crystals

I propose here three exercises taken from Chapter 1 of Peres’ book, Ref. [29]. Start with Exercise 1.3.

Exercise B.1. Design an optical system which converts photons of a given linear polarisation into photons of given elliptical polarisation. [**Hint:** A polariser at angle α would select the linear polarisation. Use afterwards a wave-plate with the appropriate δ .]

Next, I propose you Exercise 1.4.

Exercise B.2. Show that a device consisting of a QWP, followed by a thick calcite crystal with its optic axis at 45° to that of the QWP, followed in turn by a second QWP orthogonal to the first one, is a selector of circular polarisations: Circularly polarised incident photons emerge from it with their original circular polarisation, but in two separate beams, depending on their helicity. What happens if the optic axes of the QWP are parallel, rather than orthogonal? [**Hint:** The middle calcite crystal should be with its normal at an angle θ_i from the beam: any θ_i would do. It is crucial, however, that its optic axis is at $\hat{\mathbf{n}} = \frac{1}{\sqrt{2}}(\hat{\mathbf{x}} + \hat{\mathbf{y}})$ with respect to the standard axis of the first QWP. The second QWP has to have $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ exchanged. Verify that if the two QWP are identical, the final circular polarisation is switched.]

Finally, try Exercise 1.1.

Exercise B.3. Consider a beam of photons having a wave vector \mathbf{k} along the $\hat{\mathbf{z}}$ -axis, and linear polarisation initially along the $\hat{\mathbf{x}}$ -axis. These photons pass through N consecutive identical calcite crystals, with gradually increasing tilts: the direction O of the optic axis of the m -th crystal ($m = 1, \dots, N$) is given, with respect to the fixed coordinate system defined above, by $O_x = \sin(\pi m/2N)$ and $O_y = \cos(\pi m/2N)$. Show that there are 2^N outgoing beams. What are their polarisations? What are their intensities (neglecting absorption)? Show that, as $N \rightarrow \infty$, nearly all the outgoing light is found in one of the beams, which is polarised in the $\hat{\mathbf{y}}$ -direction. [**Hint:** Put all thick calcite crystals with their surface normal tilted by θ_i . The crystals have optic axis $\hat{\mathbf{n}}_m = (\sin \alpha_m, \cos \alpha_m, 0)$ with $\alpha_m = m\pi/(2N)$. The first crystal will separate the beam into two components of intensities $\cos^2 \alpha_1$ and $\sin^2 \alpha_1$, where $\alpha_1 = \pi/(2N)$. Then these two are further separated into two, and so on. Half of the beams are polarised along $\hat{\mathbf{x}}$, half along $\hat{\mathbf{y}}$. The most intense beam will have intensity $\cos^{2N}(\alpha_1)$. Taking the limit $N \rightarrow \infty$ is simple.]

C. Superconductivity

Here is a small introduction to a few standard concepts from the theory of superconductivity, mostly based on the books by de Gennes [37] and Tinkham [38]. The goal is to provide the basic ingredients to understand Josephson junctions, including its flux-tunable version, the dc-SQUID.

C.1. The BCS problem

Consider a system of interacting electrons:

$$\hat{H} = \sum_{\mathbf{k}, \sigma} \xi_{\mathbf{k}} \hat{c}_{\mathbf{k}, \sigma}^\dagger \hat{c}_{\mathbf{k}, \sigma} + \frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2, \mathbf{q}} \sum_{\sigma_1, \sigma_2} V(\mathbf{k}_1 + \mathbf{q}, \mathbf{k}_2 - \mathbf{q} | \mathbf{k}_1, \mathbf{k}_2) \hat{c}_{\mathbf{k}_1 + \mathbf{q}, \sigma_1}^\dagger \hat{c}_{\mathbf{k}_2 - \mathbf{q}, \sigma_2}^\dagger \hat{c}_{\mathbf{k}_2, \sigma_2} \hat{c}_{\mathbf{k}_1, \sigma_1}, \quad (\text{C.1})$$

where $\xi_{\mathbf{k}} = \epsilon_{\mathbf{k}} - \mu$ is the single-particle energy dispersion minus the chemical potential μ , as appropriate for a grand-canonical description, and V is the interaction potential:

$$V(\mathbf{k}_1 + \mathbf{q}, \mathbf{k}_2 - \mathbf{q} | \mathbf{k}_1, \mathbf{k}_2) = \int d\mathbf{x}_1 d\mathbf{x}_2 \phi_{\mathbf{k}_1 + \mathbf{q}}^*(\mathbf{x}_1) \phi_{\mathbf{k}_2 - \mathbf{q}}^*(\mathbf{x}_2) V_{\text{int}}(|\mathbf{x}_1 - \mathbf{x}_2|) \phi_{\mathbf{k}_2}(\mathbf{x}_2) \phi_{\mathbf{k}_1}(\mathbf{x}_1),$$

with $\phi_{\mathbf{k}}(\mathbf{x})$ single-particle orbitals labelled by a wave-vector \mathbf{k} . Let us neglect most of the interaction terms, retaining only those where $\mathbf{k}_2 = -\mathbf{k}_1$, and $\sigma_2 = -\sigma_1$. We arrive at the following reduced Hamiltonian:

$$\hat{H}_{\text{red}} = \sum_{\mathbf{k}, \sigma} \xi_{\mathbf{k}} \hat{c}_{\mathbf{k}, \sigma}^\dagger \hat{c}_{\mathbf{k}, \sigma} + \sum_{\mathbf{k}, \mathbf{k}'} V_{\mathbf{k}, \mathbf{k}'} \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger \hat{c}_{-\mathbf{k}'\downarrow} \hat{c}_{\mathbf{k}'\uparrow}, \quad (\text{C.2})$$

where $V_{\mathbf{k}, \mathbf{k}'} = V(\mathbf{k}, -\mathbf{k} | \mathbf{k}', -\mathbf{k}')$ and the factor $\frac{1}{2}$ is cancelled by the presence of two spin contributions. The final step is a mean-field approximation, where we introduce a parameter to be determined self-consistently:¹

$$\Delta_{\mathbf{k}} = - \sum_{\mathbf{k}'} V_{\mathbf{k}, \mathbf{k}'} \langle \hat{c}_{-\mathbf{k}'\downarrow} \hat{c}_{\mathbf{k}'\uparrow} \rangle, \quad (\text{C.3})$$

and rewrite a quadratic Hamiltonian in the form:

$$\hat{H}_{\text{BCS}} = \sum_{\mathbf{k}, \sigma} \xi_{\mathbf{k}} \hat{c}_{\mathbf{k}, \sigma}^\dagger \hat{c}_{\mathbf{k}, \sigma} - \sum_{\mathbf{k}} \left(\Delta_{\mathbf{k}} \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger + \Delta_{\mathbf{k}}^* \hat{c}_{-\mathbf{k}\downarrow} \hat{c}_{\mathbf{k}\uparrow} \right). \quad (\text{C.4})$$

This quadratic fermionic problem can be solved by a Bogoljubov transformation, as follows.

We first notice that $\hat{c}_{\mathbf{k}\downarrow}^\dagger \hat{c}_{\mathbf{k}\downarrow} = 1 - \hat{c}_{\mathbf{k}\downarrow} \hat{c}_{\mathbf{k}\downarrow}^\dagger$, so that, dropping a constant term and using the $\xi_{\mathbf{k}} = \xi_{-\mathbf{k}}$, we can rewrite the kinetic term as $\sum_{\mathbf{k}} \xi_{\mathbf{k}} (\hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{\mathbf{k}\uparrow} - \hat{c}_{-\mathbf{k}\downarrow} \hat{c}_{-\mathbf{k}\downarrow}^\dagger)$. The Hamiltonian can then be rewritten in a convenient matrix (Nambu) form:

$$\hat{H}_{\text{BCS}} = \sum_{\mathbf{k}} (\hat{c}_{\mathbf{k}\uparrow}^\dagger, \hat{c}_{-\mathbf{k}\downarrow}) \begin{pmatrix} \xi_{\mathbf{k}} & -\Delta_{\mathbf{k}} \\ -\Delta_{\mathbf{k}}^* & -\xi_{\mathbf{k}} \end{pmatrix} \begin{pmatrix} \hat{c}_{\mathbf{k}\uparrow} \\ \hat{c}_{-\mathbf{k}\downarrow}^\dagger \end{pmatrix}. \quad (\text{C.5})$$

The goal is to find new combinations of the operators that diagonalize this 2×2 problem. With a bit of imagination, you realize that we are dealing, effectively, with a Pauli matrix problem. Setting

¹The minus sign is useful because superconductivity occurs for attractive interactions.

$\Delta_{\mathbf{k}} = |\Delta_{\mathbf{k}}|e^{i\varphi_{\mathbf{k}}}$ we have:

$$\mathbf{H}_{\mathbf{k}} = \begin{pmatrix} \xi_{\mathbf{k}} & -|\Delta_{\mathbf{k}}|e^{i\varphi_{\mathbf{k}}} \\ -|\Delta_{\mathbf{k}}|e^{-i\varphi_{\mathbf{k}}} & -\xi_{\mathbf{k}} \end{pmatrix} = \xi_{\mathbf{k}}\hat{\sigma}^z - |\Delta_{\mathbf{k}}|(\cos\varphi_{\mathbf{k}}\hat{\sigma}^x - \sin\varphi_{\mathbf{k}}\hat{\sigma}^y) = E_{\mathbf{k}}\mathbf{n}_{\mathbf{k}} \cdot \hat{\boldsymbol{\sigma}}, \quad (\text{C.6})$$

where the effective magnetic field $E_{\mathbf{k}}$ and the unit vector $\mathbf{n}_{\mathbf{k}}$ along which the spin points are:

$$E_{\mathbf{k}} = \sqrt{\xi_{\mathbf{k}}^2 + |\Delta_{\mathbf{k}}|^2} \quad \text{and} \quad \mathbf{n}_{\mathbf{k}} = \left(-\frac{|\Delta_{\mathbf{k}}|}{E_{\mathbf{k}}} \cos\varphi_{\mathbf{k}}, \frac{|\Delta_{\mathbf{k}}|}{E_{\mathbf{k}}} \sin\varphi_{\mathbf{k}}, \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \right). \quad (\text{C.7})$$

Let us define the angle $\theta_{\mathbf{k}} \in [0, \pi]$ in the usual spherical coordinate convention:

$$\cos\theta_{\mathbf{k}} = \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \quad \text{and} \quad \sin\theta_{\mathbf{k}} = \frac{|\Delta_{\mathbf{k}}|}{E_{\mathbf{k}}}. \quad (\text{C.8})$$

The standard azimuthal angle $\phi_{\mathbf{k}}$ would then be $\phi_{\mathbf{k}} = \pi - \varphi_{\mathbf{k}}$, so that $\cos\phi_{\mathbf{k}} = -\cos\varphi_{\mathbf{k}}$ and $\sin\phi_{\mathbf{k}} = \sin\varphi_{\mathbf{k}}$. As known from your lectures on the spinor eigenstates $|\pm, \mathbf{n}\rangle$ in an arbitrary direction \mathbf{n} , the positive and negative spin states would be

$$|+, \mathbf{n}_{\mathbf{k}}\rangle = \begin{pmatrix} \cos\frac{\theta_{\mathbf{k}}}{2} \\ -e^{-i\varphi_{\mathbf{k}}}\sin\frac{\theta_{\mathbf{k}}}{2} \end{pmatrix} \quad \text{and} \quad |-, \mathbf{n}_{\mathbf{k}}\rangle = \begin{pmatrix} e^{i\varphi_{\mathbf{k}}}\sin\frac{\theta_{\mathbf{k}}}{2} \\ \cos\frac{\theta_{\mathbf{k}}}{2} \end{pmatrix}. \quad (\text{C.9})$$

Let us now define the two real and positive quantities

$$u_{\mathbf{k}} = \cos\frac{\theta_{\mathbf{k}}}{2} \quad \text{and} \quad v_{\mathbf{k}} = \sin\frac{\theta_{\mathbf{k}}}{2} \quad \implies \quad u_{\mathbf{k}}^2 + v_{\mathbf{k}}^2 = 1, \quad (\text{C.10})$$

and organize the two spinor eigenvectors as columns of a 2×2 unitary matrix

$$\mathbf{U}_{\mathbf{k}} = \begin{pmatrix} u_{\mathbf{k}} & v_{\mathbf{k}}e^{i\varphi_{\mathbf{k}}} \\ -v_{\mathbf{k}}e^{-i\varphi_{\mathbf{k}}} & u_{\mathbf{k}} \end{pmatrix} \quad \implies \quad \mathbf{U}_{\mathbf{k}}^\dagger = \begin{pmatrix} u_{\mathbf{k}} & -v_{\mathbf{k}}e^{i\varphi_{\mathbf{k}}} \\ v_{\mathbf{k}}e^{-i\varphi_{\mathbf{k}}} & u_{\mathbf{k}} \end{pmatrix}. \quad (\text{C.11})$$

The explicit expression for $u_{\mathbf{k}}$ and $v_{\mathbf{k}}$ follow from simple trigonometry. In particular:

$$\begin{cases} 2u_{\mathbf{k}}v_{\mathbf{k}} &= 2\cos\frac{\theta_{\mathbf{k}}}{2}\sin\frac{\theta_{\mathbf{k}}}{2} = \sin\theta_{\mathbf{k}} = \frac{|\Delta_{\mathbf{k}}|}{E_{\mathbf{k}}} \\ u_{\mathbf{k}}^2 - v_{\mathbf{k}}^2 &= \cos^2\frac{\theta_{\mathbf{k}}}{2} - \sin^2\frac{\theta_{\mathbf{k}}}{2} = \cos\theta_{\mathbf{k}} = \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \end{cases} \quad (\text{C.12})$$

This, together with $u_{\mathbf{k}}^2 + v_{\mathbf{k}}^2 = 1$, immediately leads to:

$$\begin{cases} v_{\mathbf{k}}^2 &= \frac{1}{2}\left(1 - \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}}\right) \\ u_{\mathbf{k}}^2 &= \frac{1}{2}\left(1 + \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}}\right) \end{cases}. \quad (\text{C.13})$$

We can now introduce the two new (Bogoljubov) fermion operators as follows:

$$\begin{pmatrix} \hat{c}_{\mathbf{k}\uparrow} \\ \hat{c}_{-\mathbf{k}\downarrow}^\dagger \end{pmatrix} = \mathbf{U}_{\mathbf{k}} \begin{pmatrix} \hat{\gamma}_{\mathbf{k}\uparrow} \\ \hat{\gamma}_{-\mathbf{k}\downarrow}^\dagger \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \hat{\gamma}_{\mathbf{k}\uparrow} \\ \hat{\gamma}_{-\mathbf{k}\downarrow}^\dagger \end{pmatrix} = \mathbf{U}_{\mathbf{k}}^\dagger \begin{pmatrix} \hat{c}_{\mathbf{k}\uparrow} \\ \hat{c}_{-\mathbf{k}\downarrow}^\dagger \end{pmatrix}, \quad (\text{C.14})$$

such that the Hamiltonian is diagonalized:

$$\begin{aligned} \hat{H}_{\text{BCS}} &= \sum_{\mathbf{k}} (\hat{\gamma}_{\mathbf{k}\uparrow}^\dagger, \hat{\gamma}_{-\mathbf{k}\downarrow}) \mathbf{U}_{\mathbf{k}}^\dagger \mathbf{H}_{\mathbf{k}} \mathbf{U}_{\mathbf{k}} \begin{pmatrix} \hat{\gamma}_{\mathbf{k}\uparrow} \\ \hat{\gamma}_{-\mathbf{k}\downarrow}^\dagger \end{pmatrix} = \sum_{\mathbf{k}} E_{\mathbf{k}} (\hat{\gamma}_{\mathbf{k}\uparrow}^\dagger \hat{\gamma}_{\mathbf{k}\uparrow} - \hat{\gamma}_{-\mathbf{k}\downarrow} \hat{\gamma}_{-\mathbf{k}\downarrow}^\dagger) \\ &= \sum_{\mathbf{k}} E_{\mathbf{k}} (\hat{\gamma}_{\mathbf{k}\uparrow}^\dagger \hat{\gamma}_{\mathbf{k}\uparrow} + \hat{\gamma}_{-\mathbf{k}\downarrow}^\dagger \hat{\gamma}_{-\mathbf{k}\downarrow}) + E_{\text{gs}}, \end{aligned} \quad (\text{C.15})$$

where we used the canonical anticommutation for the new fermions, defining also the ground state energy:

$$E_{\text{gs}} = - \sum_{\mathbf{k}} E_{\mathbf{k}}. \quad (\text{C.16})$$

The ground state is also known as *Bogoljubov vacuum* because it is annihilated by all $\hat{\gamma}_{\mathbf{k}\sigma}$. To get it, consider the following (unnormalized) state obtained by applying all the operators:

$$\begin{aligned} |\Psi\rangle &= \prod_{\mathbf{k}} \hat{\gamma}_{-\mathbf{k}\downarrow} \hat{\gamma}_{\mathbf{k}\uparrow} |0\rangle = \prod_{\mathbf{k}} \left(v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} \hat{c}_{\mathbf{k}\uparrow}^{\dagger} + u_{\mathbf{k}} c_{-\mathbf{k}\downarrow} \right) \left(u_{\mathbf{k}} c_{\mathbf{k}\uparrow} - v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger} \right) |0\rangle \\ &= \prod_{\mathbf{k}} \left(-v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} \right) \left(u_{\mathbf{k}} + v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} \hat{c}_{\mathbf{k}\uparrow}^{\dagger} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger} \right) |0\rangle. \end{aligned} \quad (\text{C.17})$$

This state is evidently annihilated by any $\hat{\gamma}_{\mathbf{k}\uparrow}$ and $\hat{\gamma}_{-\mathbf{k}\downarrow}$ for all \mathbf{k} , hence it is the ground state of the BCS Hamiltonian. By normalizing it, we finally arrive at the standard form of the BCS ground state. Summarizing, we have:

❶

The BCS problem. The BCS Hamiltonian

$$\begin{aligned} \hat{H}_{\text{BCS}} &= \sum_{\mathbf{k}} \xi_{\mathbf{k}} (\hat{c}_{\mathbf{k}\uparrow}^{\dagger} \hat{c}_{\mathbf{k}\uparrow} - \hat{c}_{-\mathbf{k}\downarrow} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger}) - \sum_{\mathbf{k}} (\Delta_{\mathbf{k}} \hat{c}_{\mathbf{k}\uparrow}^{\dagger} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger} + \Delta_{\mathbf{k}}^* \hat{c}_{-\mathbf{k}\downarrow} \hat{c}_{\mathbf{k}\uparrow}) \\ &= E_{\mathbf{k}} \sum_{\mathbf{k}} \left(\hat{\gamma}_{\mathbf{k}\uparrow}^{\dagger} \hat{\gamma}_{\mathbf{k}\uparrow} + \hat{\gamma}_{-\mathbf{k}\downarrow}^{\dagger} \hat{\gamma}_{-\mathbf{k}\downarrow} - 1 \right). \end{aligned} \quad (\text{C.18})$$

is diagonalized by the Bogoljubov fermionic operators:

$$\begin{cases} \hat{\gamma}_{\mathbf{k}\uparrow} &= u_{\mathbf{k}} c_{\mathbf{k}\uparrow} - v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger} \\ \hat{\gamma}_{-\mathbf{k}\downarrow}^{\dagger} &= v_{\mathbf{k}} e^{-i\varphi_{\mathbf{k}}} c_{\mathbf{k}\uparrow} + u_{\mathbf{k}} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger} \end{cases}, \quad (\text{C.19})$$

where $\Delta_{\mathbf{k}} = |\Delta_{\mathbf{k}}| e^{i\varphi_{\mathbf{k}}}$, $E_{\mathbf{k}} = \sqrt{\xi_{\mathbf{k}}^2 + |\Delta_{\mathbf{k}}|^2}$ and:

$$v_{\mathbf{k}}^2 = \frac{1}{2} \left(1 - \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \right) \quad u_{\mathbf{k}}^2 = \frac{1}{2} \left(1 + \frac{\xi_{\mathbf{k}}}{E_{\mathbf{k}}} \right). \quad (\text{C.20})$$

Its normalized ground state is given by:

$$|\Psi_{\text{BCS}}\rangle = \prod_{\mathbf{k}} \left(u_{\mathbf{k}} + v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} \hat{c}_{\mathbf{k}\uparrow}^{\dagger} \hat{c}_{-\mathbf{k}\downarrow}^{\dagger} \right) |0\rangle. \quad (\text{C.21})$$

The gap equation. It is now simple to verify that:

$$\langle \Psi_{\text{BCS}} | \hat{c}_{-\mathbf{k}\downarrow} \hat{c}_{\mathbf{k}\uparrow} | \Psi_{\text{BCS}} \rangle = u_{\mathbf{k}} v_{\mathbf{k}} e^{i\varphi_{\mathbf{k}}} = \frac{|\Delta_{\mathbf{k}}|}{2E_{\mathbf{k}}} e^{i\varphi_{\mathbf{k}}}. \quad (\text{C.22})$$

Hence, the self-consistency equation for $\Delta_{\mathbf{k}}$ reads:

$$\Delta_{\mathbf{k}} = |\Delta_{\mathbf{k}}| e^{i\varphi_{\mathbf{k}}} = - \sum_{\mathbf{k}'} V_{\mathbf{k},\mathbf{k}'} \langle \hat{c}_{-\mathbf{k}'\downarrow} \hat{c}_{\mathbf{k}'\uparrow} \rangle = - \sum_{\mathbf{k}'} V_{\mathbf{k},\mathbf{k}'} \frac{|\Delta_{\mathbf{k}'}|}{2\sqrt{\xi_{\mathbf{k}'}^2 + |\Delta_{\mathbf{k}'}|^2}} e^{i\varphi_{\mathbf{k}'}}. \quad (\text{C.23})$$

Let us now assume that $V_{\mathbf{k},\mathbf{k}'}$ is *negative* (attractive) and only dependent on the energy $\xi_{\mathbf{k}}$ and $\xi_{\mathbf{k}'}$ of initial and final states, hence, in particular, rotationally invariant. Even more drastically, we assume it to be a constant within an energy shell $\pm \hbar\omega_D$ of the Fermi energy:

$$V_{\mathbf{k},\mathbf{k}'} = -\frac{V}{\text{Vol}} \Theta(\hbar\omega_D - |\xi_{\mathbf{k}}|) \Theta(\hbar\omega_D - |\xi_{\mathbf{k}'}|) = \begin{cases} -\frac{V}{\text{Vol}} & \text{if } |\xi_{\mathbf{k}}|, |\xi_{\mathbf{k}'}| < \hbar\omega_D \\ 0 & \text{otherwise} \end{cases}. \quad (\text{C.24})$$

This leads to the following **s-wave Ansatz** for $\Delta_{\mathbf{k}}$:

$$\Delta_{\mathbf{k}} = \Theta(\hbar\omega_D - |\xi_{\mathbf{k}}|) \Delta e^{i\varphi} \quad \text{with} \quad \Delta > 0. \quad (\text{C.25})$$

By introducing the single-particle density-of-states ² $\rho(\xi) = (1/\text{Vol}) \sum_{\mathbf{k}} \delta(\xi - \xi_{\mathbf{k}})$ we can write the self-consistent equation for Δ as follows:

$$\Delta = V \int_{-\hbar\omega_D}^{\hbar\omega_D} d\xi \rho(\xi) \frac{\Delta}{2\sqrt{\xi^2 + \Delta^2}} \approx V\rho(0) \int_0^{\hbar\omega_D} d\xi \frac{\Delta}{\sqrt{\xi^2 + \Delta^2}}, \quad (\text{C.26})$$

where the second expression follows by approximating $\rho(\xi) \approx \rho(0)$ close to the Fermi energy. This finally leads to an analytic expression for the gap:

$$\frac{1}{\rho(0)V} = \int_0^{\hbar\omega_D} d\xi \frac{\Delta}{\sqrt{\xi^2 + \Delta^2}} = \text{arcsinh}\left(\frac{\hbar\omega_D}{\Delta}\right) \implies \Delta = \frac{\hbar\omega_D}{\sinh\left(\frac{1}{\rho(0)V}\right)} \approx \hbar\omega_D e^{-\frac{1}{\rho(0)V}}, \quad (\text{C.27})$$

where the final approximation applies in the weak-coupling limit $\rho(0)V \ll 1$.

The role of the phase. For an s-wave superconductor, where $\varphi_{\mathbf{k}} = \varphi$, we write the BCS ground state as:

$$|\Psi_{\text{BCS}}(\varphi)\rangle = \prod_{\mathbf{k}} \left(u_{\mathbf{k}} + e^{i\varphi} v_{\mathbf{k}} \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger \right) |0\rangle. \quad (\text{C.28})$$

We recall that $u_{\mathbf{k}}$ and $v_{\mathbf{k}}$ real and positive. The relative phase $e^{i\varphi}$ plays a relatively minor role if you deal with a single superconductor: it will play a very important role in describing the Josephson tunnelling between *two superconductors* separated by a thin insulating (oxide) layer.

$|\Psi_{\text{BCS}}(\varphi)\rangle$ describes a superposition of states with all possible (even) fermion numbers. To simplify our writing, let us denote by $\hat{b}_{\mathbf{k}}^\dagger = \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{-\mathbf{k}\downarrow}^\dagger$ the operator that creates a pair of fermions in the Cooper-pair state ($\mathbf{k}\uparrow, -\mathbf{k}\downarrow$). By expanding the factor $\prod_{\mathbf{k}} (u_{\mathbf{k}} + e^{i\varphi} v_{\mathbf{k}} \hat{b}_{\mathbf{k}}^\dagger)$ you can write:

$$\begin{aligned} |\Psi_{\text{BCS}}(\varphi)\rangle &= \left(\prod_{\mathbf{k}} u_{\mathbf{k}} \right) \left(|0\rangle + e^{i\varphi} \sum_{\mathbf{k}_1} \frac{v_{\mathbf{k}_1}}{u_{\mathbf{k}_1}} \hat{b}_{\mathbf{k}_1}^\dagger |0\rangle + e^{2i\varphi} \sum_{(\mathbf{k}_1, \mathbf{k}_2)} \frac{v_{\mathbf{k}_1}}{u_{\mathbf{k}_1}} \frac{v_{\mathbf{k}_2}}{u_{\mathbf{k}_2}} \hat{b}_{\mathbf{k}_1}^\dagger \hat{b}_{\mathbf{k}_2}^\dagger |0\rangle + \dots \right. \\ &\quad \left. + e^{2ni\varphi} \sum_{(\mathbf{k}_1, \dots, \mathbf{k}_n)} \frac{v_{\mathbf{k}_1}}{u_{\mathbf{k}_1}} \dots \frac{v_{\mathbf{k}_n}}{u_{\mathbf{k}_n}} \hat{b}_{\mathbf{k}_1}^\dagger \dots \hat{b}_{\mathbf{k}_n}^\dagger |0\rangle + \dots \right) \\ &= \sum_{n=0}^{\infty} e^{in\varphi} A_n |\Psi_n\rangle \end{aligned} \quad (\text{C.29})$$

where the notation $(\mathbf{k}_1, \dots, \mathbf{k}_n)$ means that the n -uple of wave-vectors should be included only once, and $|\Psi_n\rangle$ denotes a *normalised* state with exactly n Cooper pairs (hence $N = 2n$ fermions), appearing with (real) amplitude A_n but with an overall phase $e^{in\varphi}$. Normalisation of all states implies that the coefficients A_n^2 can be thought as a probability distribution of the various n in the BCS state:

$$\langle \Psi_{\text{BCS}} | \Psi_{\text{BCS}} \rangle = 1 \implies \sum_{n=0}^{\infty} A_n^2 = 1. \quad (\text{C.30})$$

At this stage, φ could be used as a technical tool to single-out the various fixed particle number states. Indeed, by integrating over φ we get:

$$A_n |\Psi_n\rangle = \int_0^{2\pi} \frac{d\varphi}{2\pi} e^{-in\varphi} |\Psi_{\text{BCS}}(\varphi)\rangle. \quad (\text{C.31})$$

The coefficients A_n^2 could also be explicitly calculated by an integral over φ , but this will not be relevant to our discussion. ³ What is relevant, is that in a macroscopic superconductor, A_n^2 is

²Recall that this implies that for large quantization volumes:

$$\frac{1}{\text{Vol}} \sum_{\mathbf{k}} F(\xi_{\mathbf{k}}) = \int d\xi \rho(\xi) F(\xi).$$

³One can verify that:

$$A_n^2 = \int_0^{2\pi} \frac{d\varphi}{2\pi} e^{-in\varphi} \prod_{\mathbf{k}} (u_{\mathbf{k}}^2 + e^{i\varphi} v_{\mathbf{k}}^2). \quad (\text{C.32})$$

peaked around a mean value of the number of Cooper pairs, call it n_0 , which is *extensive*. Indeed, if $\hat{N} = \sum_{\mathbf{k},\sigma} \hat{c}_{\mathbf{k}\sigma}^\dagger \hat{c}_{\mathbf{k}\sigma}$ denotes the total number of fermions operator, with average $N_0 = \langle \Psi_{\text{BCS}} | \hat{N} | \Psi_{\text{BCS}} \rangle$, you can write:

$$n_0 = \frac{1}{2} N_0 = \frac{1}{2} \langle \Psi_{\text{BCS}} | \hat{N} | \Psi_{\text{BCS}} \rangle = \sum_{\mathbf{k}} v_{\mathbf{k}}^2 = \text{Vol} \int \frac{d\mathbf{k}}{(2\pi)^3} v_{\mathbf{k}}^2. \quad (\text{C.33})$$

Interestingly, the *width* of the distribution A_n^2 scales with $\sqrt{\text{Vol}}$, as you can show that:

$$\begin{aligned} (\Delta n)^2 &= \frac{1}{4} (\Delta N)^2 = \frac{1}{4} (\langle \Psi_{\text{BCS}} | \hat{N}^2 | \Psi_{\text{BCS}} \rangle - \langle \Psi_{\text{BCS}} | \hat{N} | \Psi_{\text{BCS}} \rangle^2) \\ &= \sum_{\mathbf{k}} u_{\mathbf{k}}^2 v_{\mathbf{k}}^2 = \text{Vol} \int \frac{d\mathbf{k}}{(2\pi)^3} u_{\mathbf{k}}^2 v_{\mathbf{k}}^2. \end{aligned} \quad (\text{C.34})$$

Hence, for a macroscopic superconductor, it makes no difference if you calculate physical properties by using $|\Psi_{\text{BCS}}\rangle$ — a state that is simple to work with — or rather by using the much more complicated state with fixed number of Cooper pairs $|\Psi_n\rangle$ with $n \sim n_0$. The rationale behind is very similar to the grand-canonical description, as opposed to a canonical one, which become equivalent in the thermodynamic limit.

C.2. The Josephson effect

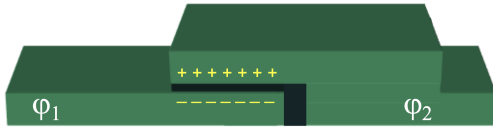


Figure C.1: Sketch of a Josephson junction, with the two superconductors characterized by a phase $\varphi_{1,2}$, separated by a thin insulating layer. The charges highlight that a small enough junction should show Coulomb effects due to a finite capacitance.

Let us consider now two macroscopic superconductors separated by a thin insulating layer (of order of 30 Angstroms), as sketched in Fig. C.1. Neglecting charging effects, for the time being, we take:

$$\hat{H} = \hat{H}_1^{\text{BCS}} + \hat{H}_2^{\text{BCS}} + \hat{H}_{\text{tunn}}, \quad (\text{C.35})$$

where the tunnelling term is:

$$\hat{H}_{\text{tunn}} = \sum_{\sigma} \sum_{\mathbf{k}_1, \mathbf{k}_2} \left(T_{\mathbf{k}_1, \mathbf{k}_2} \hat{c}_{\mathbf{k}_1, \sigma, 1}^\dagger \hat{c}_{\mathbf{k}_2, \sigma, 2} + \text{H.c.} \right). \quad (\text{C.36})$$

Suppose we have a state with a total number of Cooper pairs $n_t = n_1 + n_2$ written as $|\Psi_{n_1}\rangle \otimes |\Psi_{n_2}\rangle$, denoted as $|n_1; n_2\rangle$ for shortness. All possible sharing of n_t in the two superconductors are perfectly degenerate, with energy $2E_{\text{gs}}$, as you can transfer a Cooper pair from one superconductor to the other without affecting the energy: recall that indeed a BCS state is a superposition of states with different number of Cooper pairs. The action of $T_{\mathbf{k}_1, \mathbf{k}_2} \hat{c}_{\mathbf{k}_1, \sigma, 1}^\dagger \hat{c}_{\mathbf{k}_2, \sigma, 2}$ on $|n_1; n_2\rangle$ is to create a single unpaired hole in 2, and a single unpaired electron in 1. This intermediate state

$$\mathcal{N}_{\mathbf{k}_1, \mathbf{k}_2} |I; e\mathbf{k}_1\sigma; h\mathbf{k}_2\sigma\rangle = \hat{c}_{\mathbf{k}_1, \sigma, 1}^\dagger \hat{c}_{\mathbf{k}_2, \sigma, 2} |n_1; n_2\rangle,$$

has a higher energy $2E_{\text{gs}} + E_{\mathbf{k}_1} + E_{\mathbf{k}_2}$. Here $\mathcal{N}_{\mathbf{k}_1, \mathbf{k}_2} = u_{\mathbf{k}_1} v_{\mathbf{k}_2}$ is a normalization constant.⁴ To “undo” that, you can use the same tunnelling term, but now with opposite momenta and spin, so has

⁴This gives, correctly, $A_0^2 = \prod_{\mathbf{k}} u_{\mathbf{k}}^2$, $A_1^2 = \sum_{\mathbf{k}_1} (\prod_{\mathbf{k} \neq \mathbf{k}_1} u_{\mathbf{k}}^2) v_{\mathbf{k}_1}^2$, etc.

⁴To calculate it, use the fact that, for instance:

$$\langle \Psi_{n_1} | \hat{c}_{\mathbf{k}_1, \sigma} \hat{c}_{\mathbf{k}_1, \sigma}^\dagger | \Psi_{n_1} \rangle = \langle \Psi_{\text{BCS}} | \hat{c}_{\mathbf{k}_1, \sigma} \hat{c}_{\mathbf{k}_1, \sigma}^\dagger | \Psi_{\text{BCS}} \rangle = u_{\mathbf{k}_1}^2.$$

to return to a fully paired situation, but with a Cooper pair effectively donated from 2 to 1. More precisely, you can reach the same intermediate state also as follows:

$$\tilde{N}_{\mathbf{k}_1, \mathbf{k}_2} |I; e\mathbf{k}_1\sigma; h\mathbf{k}_2\sigma\rangle = \hat{c}_{-\mathbf{k}_2-\sigma, 2}^\dagger \hat{c}_{-\mathbf{k}_1-\sigma, 1} |n_1 + 1; n_2 - 1\rangle ,$$

with a normalization constant $\tilde{N}_{\mathbf{k}_1, \mathbf{k}_2} = v_{\mathbf{k}_1} u_{\mathbf{k}_2}$. Therefore, to second order in perturbation theory in the tunnelling, you predict a coupling between $|n_1, n_2\rangle$ and $|n_1 + 1, n_2 - 1\rangle$ with a matrix element:

$$\begin{aligned} & - \sum_{\mathbf{k}_1, \mathbf{k}_2} |T_{\mathbf{k}_1, \mathbf{k}_2}|^2 \sum_{\sigma} \langle n_1 + 1, n_2 - 1 | \hat{c}_{-\mathbf{k}_1, -\sigma, 1}^\dagger \hat{c}_{-\mathbf{k}_2, -\sigma, 2} | I \rangle \frac{1}{E_{\mathbf{k}_1} + E_{\mathbf{k}_2}} \langle I | \hat{c}_{\mathbf{k}_1, \sigma, 1}^\dagger \hat{c}_{\mathbf{k}_2, \sigma, 2} | n_1, n_2 \rangle = \\ & = -2 \sum_{\mathbf{k}_1, \mathbf{k}_2} |T_{\mathbf{k}_1, \mathbf{k}_2}|^2 \frac{u_{\mathbf{k}_1} v_{\mathbf{k}_1} u_{\mathbf{k}_2} v_{\mathbf{k}_2}}{E_{\mathbf{k}_1} + E_{\mathbf{k}_2}} \equiv -\frac{E_J}{2} , \end{aligned} \quad (\text{C.37})$$

where we used that $T_{-\mathbf{k}_1, -\mathbf{k}_2} = T_{\mathbf{k}_1, \mathbf{k}_2}^*$, and the factor 2 is due to the spin.⁵ A similar calculation predicts that an identical coupling exists, due to the Hermitean conjugate term, with the state $|n_1 - 1; n_2 + 1\rangle$. Summarizing, to second-order in \hat{H}_{tunn} you predict that the Schrödinger equation reads:

$$\hat{H}|n_1; n_2\rangle = 2E_{\text{gs}}|n_1; n_2\rangle - \frac{E_J}{2} (|n_1 + 1; n_2 - 1\rangle + |n_1 - 1; n_2 + 1\rangle) . \quad (\text{C.38})$$

The Josephson energy constant E_J is a macroscopic parameter characteristic of the junction. One can show that, when dealing with two BCS superconductors:

$$E_J = \frac{1}{8} \frac{\hbar}{e^2} G_{\text{tunn}} \Delta , \quad (\text{C.39})$$

where G_{tunn} is the normal state tunnelling conductance of the barrier — proportional to the transparency of the barrier and to the surface of the junction — and Δ the superconducting gap.

Notice that $n_t = n_1 + n_2$ is conserved by the tunnelling. The only relevant variable is the difference:

$$n = \frac{n_1 - n_2}{2} \quad \implies \quad n_1 = \frac{n_t}{2} + n \quad \text{and} \quad n_2 = \frac{n_t}{2} - n , \quad (\text{C.40})$$

where we assume that n_t is even. If we define the state $|n\rangle$ as follows:

$$|n\rangle = |\Psi_{\frac{n_t}{2} + n}\rangle \otimes |\Psi_{\frac{n_t}{2} - n}\rangle , \quad (\text{C.41})$$

then Eq. (C.38) reads:

$$\hat{H}|n\rangle = 2E_{\text{gs}}|n\rangle - \frac{E_J}{2} (|n + 1\rangle + |n - 1\rangle) . \quad (\text{C.42})$$

This looks formally very similar to a tight-binding problem, but there are differences, since n is bounded to stay between $-n_t/2$ and $+n_t/2$, and there are no periodic boundary conditions which we can impose. To simplify our discussion, it helps considering the fact that n_t is **macroscopically large**, hence we can effectively consider that n runs over *all* integers, from $-\infty$ to $+\infty$. With this approximation, we proceed by constructing a “Bloch combination” of wave-vector ϕ :

$$|\Psi(\phi)\rangle = \frac{1}{\sqrt{2\pi}} \sum_{n=-\infty}^{+\infty} e^{in\phi} |n\rangle . \quad (\text{C.43})$$

It is simple to verify that, as in the familiar tight-binding problem on a line:

$$\hat{H}|\Psi(\phi)\rangle = (2E_{\text{gs}} - E_J \cos \phi) |\Psi(\phi)\rangle . \quad (\text{C.44})$$

The physical meaning of the Bloch “wavevector” ϕ is just the difference between the phases of the two-superconductors:⁶

$$\phi = \varphi_1 - \varphi_2 . \quad (\text{C.46})$$

⁵de Gennes gives a result larger by a factor of 2.

⁶You can make sense of this statement by considering that the product of two BCS states would give a wavefunction,

Charging effects. So far, we did not consider the fact that, when a Cooper pair moves from one superconductor to the other, there is a charging energy associated. This effect is important for small junctions, which have an intrinsically small capacitance C . This implies that a term in the Hamiltonian of the junction should be associated to this charging effect:

$$\hat{H}_{\text{charge}} = \frac{e^2}{2C} \hat{Q}^2 = \frac{4e^2}{2C} \hat{n}^2 = 4E_C \hat{n}^2, \quad (\text{C.47})$$

where $E_C = e^2/(2C)$ and we assumed that $n = 0$ describes a charge-neutral junction. The factor 4 comes from $\hat{Q} = 2e\hat{n}$, the charge associated to a Cooper pair being $2e$. The operator \hat{n} counts the number of Cooper pairs transferred. It is a variable conjugate to ϕ , i.e.,

$$[\hat{\phi}, \hat{n}] = i, \quad (\text{C.48})$$

a relationship which is actually a bit disrespectful of the periodic nature of the “coordinate” ϕ , but can be made fully periodic-compliant by writing:

$$[\hat{n}, e^{\pm i\hat{\phi}}] = \pm e^{\pm i\hat{\phi}}. \quad (\text{C.49})$$

Equivalently, we can think of \hat{n} as the momentum conjugate to $\hat{\phi}$, and represent $\hat{n} = -i\frac{\partial}{\partial\phi}$.

The full Hamiltonian for the Josephson junction, including the charging term but neglecting the constant $2E_{\text{gs}}$, therefore reads:

$$\hat{H}_{\text{JJ}} = 4E_C \hat{n}^2 - E_J \cos \hat{\phi}. \quad (\text{C.50})$$

The current. The current operator \hat{I} is extracted from the time derivative of \hat{n} — more precisely, its Heisenberg representation —, hence we write:

$$\hat{I} = -2e \frac{d\hat{n}}{dt}. \quad (\text{C.51})$$

The time-derivative of \hat{n} is obtained from the Heisenberg equation:

$$i\hbar \frac{d\hat{n}}{dt} = [\hat{n}, \hat{H}_{\text{JJ}}] = -E_J [\hat{n}, \cos \hat{\phi}] = -iE_J \sin \hat{\phi} \quad (\text{C.52})$$

where we used Eq. (C.49) in the last step. Hence:

$$\hat{I} = \frac{2e}{\hbar} E_J \sin \hat{\phi} = I_J \sin \hat{\phi}, \quad (\text{C.53})$$

where $I_J = \frac{2e}{\hbar} E_J$ is the so-called **critical current of the junction**, the maximum current that can flow through the junction, in absence of any voltage applied, due simply to a non-vanishing phase difference ϕ .

The voltage. So far, we assumed that no voltage is applied to the junction. In presence of a voltage V , there is a difference in the chemical potential of the two superconductors which we can associate to an extra term:

$$\hat{H}_{\text{Voltage}} = 2eV \hat{n}. \quad (\text{C.54})$$

which, upon projecting on a total number of Cooper pairs equal to n_t , would read:

$$\begin{aligned} \hat{\Pi}_{n_t} (|\Psi_{\text{BCS}}(\varphi_1)\rangle \otimes |\Psi_{\text{BCS}}(\varphi_2)\rangle) &= \sum_{n=-n_t/2}^{n_t/2} A_{n_t/2+n} e^{i(n_t/2+n)\varphi_1} A_{n_t/2-n} e^{i(n_t/2-n)\varphi_2} |\Psi_{\frac{n_t}{2}+n}\rangle \otimes |\Psi_{\frac{n_t}{2}-n}\rangle \\ &= e^{in_t(\varphi_1+\varphi_2)/2} \sum_{n=-n_t/2}^{n_t/2} A_{n_t/2+n} A_{n_t/2-n} e^{in(\varphi_1-\varphi_2)} |n\rangle. \end{aligned} \quad (\text{C.45})$$

One can calculate the Heisenberg equation of motion for $\hat{\phi}$, obtaining:

$$i\hbar \frac{d\hat{\phi}}{dt} = [\hat{\phi}, \hat{H}] = [\hat{\phi}, 4E_C \hat{n}^2 + 2eV \hat{n}] = i(8E_C \hat{n} + 2eV). \quad (\text{C.55})$$

Remarkably, neglecting charging effects, for a large junction, this would predict that the average phase difference ϕ between the two superconductors would linearly increase in time:

$$\dot{\phi} = \frac{2e}{\hbar} V, \quad (\text{C.56})$$

a relationship that is known as **ac Josephson effect**, as it would lead to a sinusoidal supercurrent in presence of a constant voltage:

$$I(t) = I_J \sin\left(\frac{2e}{\hbar} Vt\right). \quad (\text{C.57})$$

C.3. The Ginsburg-Landau description

The BCS approach is unable to treat space inhomogeneous superconductors in the presence of external magnetic fields. In principle, a more sophisticated mean-field description can be developed, leading to the Bogoljubov-de Gennes mean-field equations, see Ref. [37][Chap.5]. An alternative phenomenological approach, proposed by Ginsburg and Landau in 1951, well before the microscopic BCS theory, and later confirmed, in 1959, by a microscopic derivation using Green's functions, due to Gorkov, is quite useful in many cases. I will give here the essential ingredients of this theory, following Refs. [37,38].

The superconducting order parameter $\psi(\mathbf{x})$ is complex, and proportional to the local pair-potential $\Delta(\mathbf{x}) = -V\langle\hat{\Psi}_\downarrow(\mathbf{x})\hat{\Psi}_\uparrow(\mathbf{x})\rangle$. The total free-energy, in gaussian units, reads:

$$F_s = \int_{\text{Vol}} d\mathbf{x} \left(f_n + a|\psi|^2 + \frac{b}{2}|\psi(\mathbf{x})|^4 + \frac{1}{2m_*} \left| \left(-i\hbar\nabla + \frac{2e}{c}\mathbf{A} \right) \psi \right|^2 + \frac{1}{8\pi} \mathbf{h}^2(\mathbf{x}) \right), \quad (\text{C.58})$$

where $\mathbf{h}(\mathbf{x}) = \nabla \times \mathbf{A}$ is the (microscopic) magnetic field. The transition is dictated by a change of sign of the quadratic term, $a(T) = a'(T - T_c)$, with $a' > 0$, and the theory should be appropriate to describe the physics near T_c , where $|\psi|$ is small. The presence of the vector potential \mathbf{A} in the minimal-coupling gradient terms, with a charge $2e$, is dictated by gauge invariance. ⁷

The equilibrium GL equations are obtained by the standard calculus of variations approach: modify $\psi(\mathbf{x}) \rightarrow \psi(\mathbf{x}) + \delta\psi(\mathbf{x})$ and $\mathbf{A}(\mathbf{x}) \rightarrow \mathbf{A}(\mathbf{x}) + \delta\mathbf{A}(\mathbf{x})$, and impose that the first order variation vanishes. By calculating the variation we get:

$$\begin{aligned} \delta F_s &= \int_{\text{Vol}} d\mathbf{x} \left\{ \delta\psi^* \left[a\psi + b|\psi|^2\psi + \frac{1}{2m_*} \left(-i\hbar\nabla + \frac{2e}{c}\mathbf{A} \right)^2 \psi \right] + \text{cc.} \right\} \\ &+ \int_{\text{Vol}} d\mathbf{x} \delta\mathbf{A} \cdot \left\{ \frac{1}{4\pi} \nabla \times \mathbf{h} + \frac{e}{m_*c} \left[\psi^* \left(-i\hbar\nabla + \frac{2e}{c}\mathbf{A} \right) \psi + \text{cc.} \right] \right\}. \end{aligned} \quad (\text{C.61})$$

⁷The microscopic form of the kinetic energy in presence of \mathbf{A} is given by:

$$\hat{H}_{\text{kin}} = \frac{1}{2m} \sum_{\sigma} \int d\mathbf{x} \left(\left(-i\hbar\nabla + \frac{e}{c}\mathbf{A} \right) \hat{\Psi}_{\sigma}(\mathbf{x}) \right)^{\dagger} \cdot \left(\left(-i\hbar\nabla + \frac{e}{c}\mathbf{A} \right) \hat{\Psi}_{\sigma}(\mathbf{x}) \right). \quad (\text{C.59})$$

A gauge transformation $\mathbf{A} \rightarrow \mathbf{A} + \nabla\Lambda$ leaves the kinetic energy invariant provided we also transform the field operator as

$$\hat{\Psi}_{\sigma}(\mathbf{x}) \rightarrow \hat{\Psi}_{\sigma}(\mathbf{x}) e^{-i\frac{e}{\hbar c}\Lambda(\mathbf{x})}.$$

This implies that the local pair potential $\Delta(\mathbf{x})$, and therefore the local order parameter $\psi(\mathbf{x})$, containing *two* fermionic annihilation operators, should transform as:

$$\psi(\mathbf{x}) \rightarrow \psi(\mathbf{x}) e^{-i\frac{2e}{\hbar c}\Lambda(\mathbf{x})} = \psi(\mathbf{x}) e^{-i\frac{2\pi}{\Phi_0}\Lambda(\mathbf{x})}, \quad (\text{C.60})$$

where $\Phi_0 = hc/(2e)$ is the flux quantum.

By setting δF_s for arbitrary $\delta\psi$ and $\delta\mathbf{A}$ we get the following two GL equations:

$$\text{GL equations:} \quad \begin{cases} a\psi + b|\psi|^2\psi + \frac{1}{2m_*} \left(-i\hbar\nabla + \frac{2e}{c}\mathbf{A} \right)^2 \psi = 0 \\ \mathbf{j}_s = \frac{ie\hbar}{m_*} \left(\psi^* \nabla \psi - \psi \nabla \psi^* \right) - \frac{4e^2}{m_*c} |\psi|^2 \mathbf{A} \end{cases}, \quad (\text{C.62})$$

where we used that $\mathbf{j}_s = \frac{c}{4\pi} \nabla \times \mathbf{h}$ is the (superconducting) current density, as dictated by Maxwell's equations. Notice the similarity between the expression of \mathbf{j}_s and the quantum mechanical current for a particle of charge $2e$, and mass $m = m_*/2$, with wave-function $\psi(\mathbf{x})$.

Let us briefly recall some simple consequences of these equations, i.e., the existence of two lengths that naturally emerge: the coherence length $\xi(T)$ and the penetration length $\lambda(T)$, both diverging as $|T_c - T|^{-1/2}$.

The coherence length. Consider a one-dimensional superconductor, in absence of magnetic field, for $T < T_c$, where $a = -|a|$. The order parameter, which we can take to be real, satisfies:

$$-\frac{\hbar^2}{2m_*} \frac{d^2\psi}{dx^2} - |a|\psi + b\psi^3 = 0.$$

The equilibrium constant value is $\psi_0^2 = \frac{|a|}{b}$. By posinf $\psi = \psi_0 f$, you can rewrite this equation as:

$$|a|\psi_0 \left(-\xi^2(T) \frac{d^2\psi}{dx^2} - f + f^3 \right) = 0,$$

where

$$\xi^2(T) = \frac{\hbar^2}{2m_*|a(T)|}. \quad (\text{C.63})$$

This implies that spatial variations of f — hence of ψ — occur on a lengthscale $\xi(T)$ — the coherence length — which diverges for $T \rightarrow T_c$ because $|a| \propto |T - T_c|$.

The penetration length. Consider now how the magnetic field and current behave, when the order parameter is approximately constant: $\psi \sim \psi_0$. The superconducting current is therefore:

$$\mathbf{j}_s = -\frac{4e^2}{m_*c} \psi_0^2 \mathbf{A} \quad \implies \quad \nabla \times \mathbf{j}_s = -\frac{4e^2}{m_*c} \psi_0^2 \mathbf{h}.$$

Together with Maxwell's equation $\nabla \times \mathbf{h} = \frac{4\pi}{c} \mathbf{j}_s$, this implies that:

$$\lambda^2(T) \nabla \times (\nabla \times \mathbf{h}) + \mathbf{h} = 0 \quad \text{with} \quad \lambda^2 = \frac{m_*c^2}{16\pi e^2 \psi_0^2}. \quad (\text{C.64})$$

Consider what happens in a simple planar geometry, where the superconductor occupies the space with $z > 0$, while the region $z < 0$ is empty. It is simple to show [37][pag.180] that the only possibility is that the magnetic field is parallel to the xy plane, for instance along x , and that $h_x(z)$ decays exponentially inside the superconducting region as:

$$h_x(z) = h_x(0) e^{-z/\lambda(T)}.$$

$\lambda(T)$ is therefore the penetration length for the field: the bulk superconductor cannot have any magnetic field (Meissner effect).

The complex nature of ψ . Let us examine the role that the complex nature of $\psi(\mathbf{x})$ plays. Let us define

$$\psi(\mathbf{x}) = |\psi(\mathbf{x})| e^{i\varphi(\mathbf{x})} .$$

Consider first the superconducting current \mathbf{j}_s . After simple algebra, we can write:

$$\mathbf{j}_s = -\frac{2e}{m_*} |\psi|^2 \underbrace{\left(\hbar \nabla \varphi + \frac{2e}{c} \mathbf{A} \right)}_{\stackrel{\text{def}}{=} m_* \mathbf{v}_s} = -2e |\psi|^2 \mathbf{v}_s , \quad (\text{C.65})$$

where we defined the superfluid velocity

$$m_* \mathbf{v}_s \stackrel{\text{def}}{=} \hbar \nabla \varphi + \frac{2e}{c} \mathbf{A} . \quad (\text{C.66})$$

This suggests a natural interpretation of $|\psi|^2 = n_s$ as the superfluid (Cooper pair) *density*, so that $\mathbf{j}_s = -2en_s \mathbf{v}_s$.

Equally inspiring is to consider the free-energy density contribution. With similar algebra, you obtain:

$$\begin{aligned} \frac{1}{2m_*} \left| \left(-i\hbar \nabla + \frac{2e}{c} \mathbf{A} \right) \psi \right|^2 &= \frac{\hbar^2}{2m_*} (\nabla |\psi|)^2 + \frac{1}{2m_*} |\psi|^2 \left(\hbar \nabla \varphi + \frac{2e}{c} \mathbf{A} \right)^2 \\ &= \frac{\hbar^2}{2m_*} (\nabla |\psi|)^2 + \frac{1}{2} n_s m_* \mathbf{v}_s^2 . \end{aligned} \quad (\text{C.67})$$

The first term is the kinetic cost for changing the modulus of the order parameter, while the second piece is associated to the superfluid kinetic energy. This suggests that, inside a bulk superconductor, it is **energetically favourable** to have $\mathbf{v}_s \equiv 0$, hence:

$$\nabla \varphi = -\frac{2e}{\hbar c} \mathbf{A} = -\frac{2\pi}{\Phi_0} \mathbf{A} \quad (\mathbf{v}_s \equiv 0) . \quad (\text{C.68})$$

The gauge-invariant phase difference. Gauge invariance dictates that if $\mathbf{A} \rightarrow \mathbf{A} + \nabla \Lambda$, then you should change

$$\varphi(\mathbf{x}) \rightarrow \varphi(\mathbf{x}) - \frac{2\pi}{\Phi_0} \Lambda(\mathbf{x}) . \quad (\text{C.69})$$

Notice that the expression for \mathbf{v}_s is explicitly gauge-invariant. These considerations suggest that the phase difference appearing in the JJ energy and current should be modified and made **gauge invariant** as follows:

$$\phi = \varphi_1 - \varphi_2 + \frac{2\pi}{\Phi_0} \int_{\text{link } 2 \rightarrow 1} \mathbf{A} \cdot d\mathbf{l} . \quad (\text{C.70})$$

Flux quantization inside a superconducting ring. As a first (simple) consequence of the previous relation, consider a metallic ring with a magnetic field piercing the ring, see Fig. C.2(a). As the metal becomes superconducting, see Fig. C.2(b), the magnetic flux trapped inside the ring is modified by superconducting screening currents, and can be shown to be quantized. Even more surprising, the quantized trapped flux survives even when the external field is turned off, see Fig. C.2(c). To deduce that, consider a closed contour C well inside the bulk of the ring, which we suppose to be sufficiently thick so that $\mathbf{v}_s = 0$ on the contour. Now integrate both sides of Eq. (C.68) along C . Since the phase $\varphi(\mathbf{x})$ has to return to the same value, modulo $2\pi n$ with $n \in \mathbb{Z}$, we deduce that:

$$2\pi n = \oint_C \nabla \varphi \cdot d\mathbf{l} = -\frac{2e}{\hbar c} \oint_C \mathbf{A} \cdot d\mathbf{l} = -2\pi \frac{\Phi}{\Phi_0} . \quad (\text{C.71})$$

So, the superconducting currents flowing along the surface of the ring partially screen the magnetic field trapped inside, in such a way that the trapped flux Φ is an integer multiple of the flux quantum $\Phi_0 = hc/(2e)$.

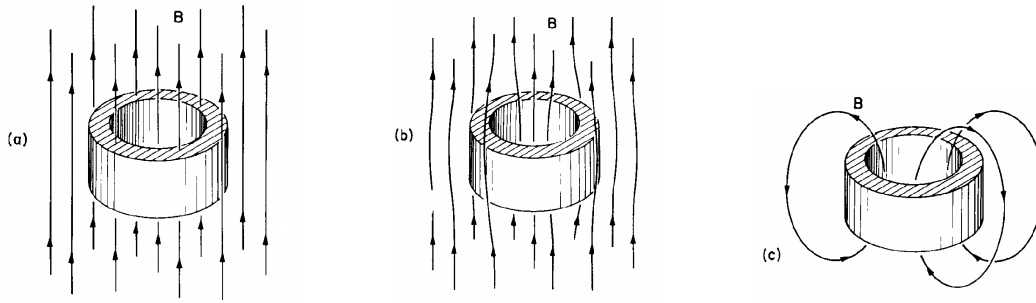


Figure C.2.: This is Fig. 21-4 in Feynman’s book. A metallic ring (a) is immersed in a magnetic field. As the ring becomes superconducting (b) screening currents modify the trapped flux in such a way as to quantize it (see text). The trapped flux survives (c) even when the external field is turned off.

C.4. Quantum interference of two JJ: The dc-SQUID

Let me now come to the application that justifies our excursion into the GL theory and the discussion of gauge invariance. Consider a ring geometry with **two Josephson junctions**, A and B as sketched in Fig. C.3, symmetrically placed and separating a first arm of the ring, with superconductor “1”, from the second arm, with superconductor “2”. The two superconductors are connected to leads and a current I is driven through the circuit. In the center of the ring, there is an external magnetic field H , which can be changed. We want to show that the current flowing $I(H)$ is periodically modulated by the magnetic field H , in a way that closely resemble the interference effects in a double slit, or in the two arms of a Mach-Zehnder interferometer.

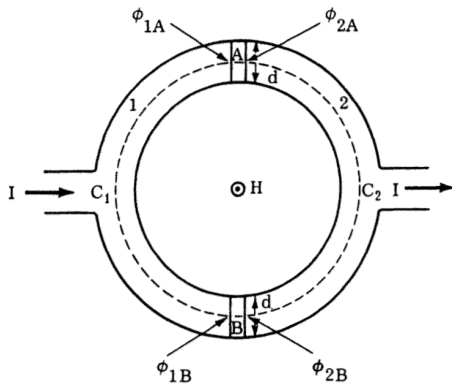


Figure C.3: This is Fig. 7-7 in de Gennes’ book [37]. In the text, we indicate ϕ_{1A} as φ_{1A} , and so on.

The superconducting current flowing through the two JJ in parallel is simply the sum of the two currents along the branches:

$$I = I_J^A \sin \phi_A + I_J^B \sin \phi_B, \quad (\text{C.72})$$

where ϕ_A and ϕ_B are the gauge-invariant phase differences at the two junctions.

As discussed in the main text, see Sec. 7.2.1, there is a definite relationship between the two gauge-invariant phase difference at the two JJ, ϕ_A and ϕ_B , and the magnetic flux trough the ring Φ .

i

The relationship between the phases at the two junctions. Therefore, we conclude that:

$$\phi_A - \phi_B = 2\pi n + 2\pi \frac{\Phi}{\Phi_0}. \quad (\text{C.73})$$

By parameterising ϕ_A and ϕ_B as follows:

$$\phi_A = \phi_+ + \pi \left(n + \frac{\Phi}{\Phi_0} \right) \quad \text{and} \quad \phi_B = \phi_+ - \left(n + \frac{\Phi}{\Phi_0} \right), \quad (\text{C.74})$$

with $\phi_+ = (\phi_A + \phi_B)/2$, the average phase difference, the total current is predicted to be:

$$I = I_J^A \sin \left(\phi_+ + \pi \left(n + \frac{\Phi}{\Phi_0} \right) \right) + I_J^B \sin \left(\phi_+ - \pi \left(n + \frac{\Phi}{\Phi_0} \right) \right). \quad (\text{C.75})$$

For two **identical** junctions, $I_J^A = I_J^B = I_J$, simple trigonometry allows us to conclude that:

$$I = 2I_J \cos \left(\pi \left(n + \frac{\Phi}{\Phi_0} \right) \right) \sin \phi_+. \quad (\text{C.76})$$

Hence, effectively, the two junctions in parallel act as a **single junction** with average phase difference ϕ_+ , and a critical current — hence Josephson energy constant — that can be **tuned by the magnetic flux**:

$$I_J^{\text{eff}}(\Phi) = 2I_J \left| \cos \left(\pi \frac{\Phi}{\Phi_0} \right) \right| \quad \implies \quad E_J^{\text{eff}}(\Phi) = \frac{\hbar}{2e} I_J^{\text{eff}}(\Phi) = 2E_J \left| \cos \left(\pi \frac{\Phi}{\Phi_0} \right) \right|. \quad (\text{C.77})$$

Notice the constructive interference for all fluxes that are multiples of Φ_0 , while the interference is **destructive** for $\Phi = \Phi_0/2$.

This calculation can be generalised to the case of two different JJs, with slightly more involved trigonometry [41][Sec.IIE]. You define the total Josephson energy, and the asymmetry parameter:

$$E_J^+ = E_J^A + E_J^B \quad d = \frac{E_J^B - E_J^A}{E_J^A + E_J^B}. \quad (\text{C.78})$$

The total potential energy of the two JJ can then be written as:

$$\begin{aligned} \hat{H}_J &= -E_J^A \cos \phi_A - E_J^B \cos \phi_B \\ &= -E_J^+ \cos \left(\pi \frac{\Phi}{\Phi_0} \right) \left(\cos \phi_+ + d \tan \left(\pi \frac{\Phi}{\Phi_0} \right) \sin \phi_+ \right) \\ &= -E_J^+ \cos \left(\pi \frac{\Phi}{\Phi_0} \right) \sqrt{1 + d^2 \tan^2 \left(\pi \frac{\Phi}{\Phi_0} \right)} \cos(\phi_+ - \phi_{\text{offset}}), \end{aligned} \quad (\text{C.79})$$

where the offset phase is defined by:

$$\tan \phi_{\text{offset}} = d \tan \left(\pi \frac{\Phi}{\Phi_0} \right). \quad (\text{C.80})$$

D. Quantum master equations for dissipative systems

We give here a perturbative derivation of the quantum Master equation (QME), closely following the treatment of Gaspard and Nagaoka [55], except for a generalization to the time-dependent case. Similar derivations are given in many books, for instance in [45].

D.1. A general framework: system plus environment

Suppose that we have a system in interaction with its environment and we want to write the total Hamiltonian that describes both. A general expression for the Hamiltonian of system-plus-environment can be [56, 57]

$$\hat{H}^{\text{tot}}(t) = \hat{H}^{\text{S}}(t) + \hat{H}^{\text{B}} + \hat{H}^{\text{SB}}, \quad (\text{D.1})$$

where $\hat{H}^{\text{S}}(t)$ is the system Hamiltonian (which can be time-dependent), \hat{H}^{B} is the environment Hamiltonian ¹ and \hat{H}^{SB} describes the interaction between the two. The interaction between system and environment can be conveniently modelled as ²

$$\hat{H}^{\text{SB}} = \sum_{\nu} \hat{A}_{\nu} \otimes \hat{B}_{\nu}, \quad (\text{D.3})$$

\hat{A}_{ν} and \hat{B}_{ν} acting respectively on the system and on the bath Hilbert space. We take these operators to be Hermitean, without loss of generality. ³ Sometimes, however, one considers operators that are explicitly *not* Hermitean, like $\hat{\sigma}^+ \hat{b} + \hat{\sigma}^- \hat{b}^{\dagger}$, in which case we might write:

$$\hat{H}^{\text{SB}} = \sum_{\nu} \left(\hat{A}_{\nu} \otimes \hat{B}_{\nu} + \hat{A}_{\nu}^{\dagger} \otimes \hat{B}_{\nu}^{\dagger} \right).$$

¹When the interaction is weak, a good description for the environment Hamiltonian is to consider one or more sets of harmonic oscillators with different frequencies [58, 59],

$$\hat{H}^{\text{B}} = \sum_{\nu} \sum_l \hbar \omega_{l\nu} \hat{b}_{l\nu}^{\dagger} \hat{b}_{l\nu} \quad (\text{D.2})$$

where ν identifies the set of harmonic oscillators and l indicates their modes. The $\hat{b}_{l\nu}^{\dagger}$ operator creates an excitation of energy $\hbar \omega_{l\nu}$.

²In principle, we can allow for a possible time-dependence of \hat{A} , which we omit from our notation. This happens, for instance, whenever the problem is described in an appropriate “rotating basis”, as in the dissipative Landau-Zener case, or when doing NMR in the rotating frame representation.

³If they are not, simply define the four Hermitean combinations

$$\begin{aligned} \hat{A}_{\nu,1} &= \frac{1}{\sqrt{2}} (\hat{A}_{\nu} + \hat{A}_{\nu}^{\dagger}) & \text{and} & & \hat{A}_{\nu,2} &= +\frac{i}{\sqrt{2}} (\hat{A}_{\nu} - \hat{A}_{\nu}^{\dagger}) \\ \hat{B}_{\nu,1} &= \frac{1}{\sqrt{2}} (\hat{B}_{\nu} + \hat{B}_{\nu}^{\dagger}) & \text{and} & & \hat{B}_{\nu,2} &= -\frac{i}{\sqrt{2}} (\hat{B}_{\nu} - \hat{B}_{\nu}^{\dagger}) \end{aligned}$$

and the interaction term will simply read:

$$\hat{H}^{\text{SB}} = \sum_{\nu} \left(\hat{A}_{\nu} \otimes \hat{B}_{\nu} + \hat{A}_{\nu}^{\dagger} \otimes \hat{B}_{\nu}^{\dagger} \right) = \sum_{\nu} \left(\hat{A}_{\nu,1} \otimes \hat{B}_{\nu,1} + \hat{A}_{\nu,2} \otimes \hat{B}_{\nu,2} \right).$$

The last form, however, can always be recast, in the end, in the compact form (D.3). Even more directly, indeed, with a suitable extension of the labels ν over which we sum: for every ν , there is a suitable $\bar{\nu}$ such that $\hat{A}_{\bar{\nu}} = \hat{A}_{\nu}^{\dagger}$ and $\hat{B}_{\bar{\nu}} = \hat{B}_{\nu}^{\dagger}$.

We will assume that the initial state of the system at time $t = 0$ is *factorised*:

$$\hat{\rho}_{\text{tot}}(0) = \hat{\rho}_{\text{S}}(0) \otimes \hat{\rho}_{\text{B}}(0) . \quad (\text{D.4})$$

and that $\hat{\rho}_{\text{B}}$ is an equilibrium state for the bath in absence of \hat{H}^{SB} , which implies that:

$$e^{-i\hat{H}^{\text{B}}t/\hbar} \hat{\rho}_{\text{B}}(0) e^{i\hat{H}^{\text{B}}t/\hbar} = \hat{\rho}_{\text{B}}(0) .$$

Technically, this means that $\hat{\rho}_{\text{B}}(0)$ can be represented in the basis of the bath Hamiltonian eigenstates $|\Phi_n\rangle$, such that $\hat{H}^{\text{B}}|\Phi_n\rangle = E_n|\Phi_n\rangle$, as:

$$\hat{\rho}_{\text{B}}(0) = \sum_n p_n |\Phi_n\rangle \langle \Phi_n| , \quad (\text{D.5})$$

with $\sum_n p_n = 1$. In particular, the bath state might be thermal in which case $p_n = \frac{e^{-\beta E_n}}{Z_{\text{B}}}$, but this is a priori not required.

We also assume that the bath operator \hat{B}_{ν} have vanishing averages over the density matrix $\hat{\rho}_{\text{B}}$:

$$\langle \hat{B}_{\nu} \rangle \stackrel{\text{def}}{=} \text{Tr}_{\text{B}}(\hat{B}_{\nu} \hat{\rho}_{\text{B}}) = 0 . \quad (\text{D.6})$$

This is certainly appropriate when the \hat{B}_{ν} operators are ‘‘position operators’’ of a bath of ‘‘harmonic oscillators’’, but might otherwise seem a loss of generality. In the end, it is not really so,⁴ but these terms certainly would give rise to simple *shifts* of the system energy levels.⁵

We will encounter later the crucial ingredient of the bath entering in our story: the free bath correlation function:

$$C_{\nu\nu'}(t-t') = \text{Tr}_{\text{B}} \left(\hat{B}_{\nu}(t) \hat{B}_{\nu'}(t') \hat{\rho}_{\text{B}} \right) , \quad (\text{D.7})$$

where $\hat{B}_{\nu}(t) = e^{i\hat{H}^{\text{B}}t/\hbar} \hat{B}_{\nu} e^{-i\hat{H}^{\text{B}}t/\hbar}$ is the interaction representation bath operator.

There are some general expressions that directly follow from the definition of $C_{\nu\nu'}(t)$ and the Hermitean nature of \hat{B}_{ν} , for instance that

$$C_{\nu\nu'}^*(t) = C_{\nu'\nu}(-t) . \quad (\text{D.8})$$

In the course of our QME calculation, we will encounter spectral densities in the form of one-sided Fourier transforms of $C_{\nu\nu'}(t)$:⁶

$$\Gamma_{\nu\nu'}(\omega^+) = \int_0^{\infty} dt e^{i(\omega+i\epsilon)t} C_{\nu\nu'}(t) , \quad (\text{D.9})$$

⁴You can in principle *redefine* the Hamiltonian by adding and subtracting the unwanted average as follows:

$$\begin{aligned} \hat{H}^{\text{S}}(t) &\rightarrow \hat{H}^{\text{S}}(t) + \sum_{\nu} \langle \hat{B}_{\nu} \rangle \hat{A}_{\nu} , \\ \hat{H}^{\text{SB}} &\rightarrow \sum_{\nu} \hat{A}_{\nu} \otimes \left(\hat{B}_{\nu} - \langle \hat{B}_{\nu} \rangle \right) \end{aligned}$$

and the trick is done. The price paid is that the system Hamiltonian now knows something about the state of the bath.

⁵These shifts are *not related* to the *Lamb shift* of atomic physics, for instance between $2P_{1/2}$ and $2S_{1/2}$ hydrogen levels, which comes from second-order effects.

⁶An equivalent way would be to introduce the Laplace transform

$$\hat{\Gamma}_{\nu\nu'}(z) = \int_0^{\infty} dt e^{-zt} C_{\nu\nu'}(t) .$$

with the one-sided Fourier transform being given by

$$\Gamma_{\nu\nu'}(\omega^+) = \hat{\Gamma}_{\nu\nu'}(z = -i\omega + 0^+) ,$$

where 0^+ denotes a small (infinitesimal) real part.

where $\omega^+ = \omega + i\epsilon$, with the infinitesimal imaginary part added to guarantee convergence of the integral. This expression should be contrasted with the ordinary Fourier transform which satisfies the usual relationships:

$$\begin{aligned}\gamma_{\nu\nu'}(\omega) &= \int_{-\infty}^{+\infty} dt e^{i\omega t} C_{\nu\nu'}(t) \\ C_{\nu\nu'}(t) &= \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} e^{-i\omega t} \gamma_{\nu\nu'}(\omega).\end{aligned}\quad (\text{D.10})$$

Notice that the relationship $C_{\nu\nu'}^*(t) = C_{\nu'\nu}(-t)$ immediately implies that $\gamma_{\nu\nu'}(\omega) = \gamma_{\nu'\nu}^*(\omega)$, i.e., the ordinary Fourier transform is a *Hermitian matrix*: hence the diagonal terms $\gamma_{\nu\nu}(\omega) \in \mathbb{R}$. On the contrary, we will now show that the one-sided Fourier transform has both a real and an *imaginary part*. To see this, you start by inserting the expression for $C_{\nu\nu'}(t)$ in terms of its Fourier transform in the expression for $\Gamma_{\nu\nu'}(\omega^+)$, obtaining: ⁷

$$\begin{aligned}\Gamma_{\nu\nu'}(\omega^+) &= \int_0^\infty dt e^{i(\omega+i\epsilon)t} \int_{-\infty}^{+\infty} \frac{d\omega'}{2\pi} e^{-i\omega't} \gamma_{\nu\nu'}(\omega') \\ &= i \int_{-\infty}^{+\infty} \frac{d\omega'}{2\pi} \frac{\gamma_{\nu\nu'}(\omega')}{\omega - \omega' + i\epsilon} \\ &= i \mathcal{f} \int_{-\infty}^{+\infty} \frac{d\omega'}{2\pi} \frac{\gamma_{\nu\nu'}(\omega')}{\omega - \omega'} + i(-i)\pi \int_{-\infty}^{+\infty} \frac{d\omega'}{2\pi} \gamma_{\nu\nu'}(\omega') \delta(\omega - \omega') \\ &= \frac{1}{2} \gamma_{\nu\nu'}(\omega) + i\sigma_{\nu\nu'}(\omega),\end{aligned}\quad (\text{D.11})$$

where we have introduced the Hilbert transform of γ

$$\sigma_{\nu\nu'}(\omega) = \mathcal{f} \int_{-\infty}^{+\infty} \frac{d\omega'}{2\pi} \frac{\gamma_{\nu\nu'}(\omega')}{\omega - \omega'} = \sigma_{\nu'\nu}^*(\omega), \quad (\text{D.12})$$

(here \mathcal{f} denotes the Cauchy principal value prescription) which is also an Hermitian matrix, hence $\sigma_{\nu\nu}(\omega) \in \mathbb{R}$ as well.

We observe one last important property of the bath: the matrix $\gamma_{\nu\nu'}$ is not only Hermitian, but also *positive*, which means that:

$$\sum_{\nu, \nu'} x_\nu^* \gamma_{\nu\nu'}(\omega) x_{\nu'} \geq 0 \quad \forall \omega \quad \forall x_\nu. \quad (\text{D.13})$$

⁷In the first step we use that:

$$\int_0^\infty dt e^{i(\omega^+ - \omega')t} = \frac{i}{\omega - \omega' + i\epsilon}.$$

In the second step, we use that standard relationship

$$\frac{1}{\omega - \omega' + i\epsilon} = P \frac{1}{\omega - \omega'} - i\pi\delta(\omega - \omega'),$$

where P denotes the Cauchy principal value prescription.

To prove the latter result, define $\hat{B} = \sum_{\nu} x_{\nu} \hat{B}_{\nu}$ and observe that:

$$\begin{aligned}
\sum_{\nu, \nu'} x_{\nu}^* \gamma_{\nu \nu'}(\omega) x_{\nu'} &= \int_{-\infty}^{+\infty} dt e^{i\omega t} \text{Tr}_{\text{B}} \left(e^{\frac{i}{\hbar} \hat{H}^{\text{B}} t} \left(\sum_{\nu} x_{\nu}^* \hat{B}_{\nu} \right) e^{-\frac{i}{\hbar} \hat{H}^{\text{B}} t} \left(\sum_{\nu'} x_{\nu'} \hat{B}_{\nu'} \right) \hat{\rho}_{\text{B}} \right) \\
&= \int_{-\infty}^{+\infty} dt e^{i\omega t} \sum_n p_n \langle \Phi_n | e^{\frac{i}{\hbar} \hat{H}^{\text{B}} t} \hat{B}^{\dagger} e^{-\frac{i}{\hbar} \hat{H}^{\text{B}} t} \hat{B} | \Phi_n \rangle \\
&= \int_{-\infty}^{+\infty} dt e^{i\omega t} \sum_{n,m} p_n e^{\frac{i}{\hbar} (E_n - E_m) t} \langle \Phi_n | \hat{B}^{\dagger} | \Phi_m \rangle \langle \Phi_m | \hat{B} | \Phi_n \rangle \\
&= \sum_{n,m} p_n \int_{-\infty}^{+\infty} dt e^{i(\omega + \frac{E_n - E_m}{\hbar}) t} \left| \langle \Phi_m | \hat{B} | \Phi_n \rangle \right|^2 \\
&= 2\pi\hbar \sum_{n,m} p_n \delta(\hbar\omega + (E_n - E_m)) \left| \langle \Phi_m | \hat{B} | \Phi_n \rangle \right|^2 \geq 0. \tag{D.14}
\end{aligned}$$

D.2. The Bloch-Redfield quantum master equation

We derive here a useful tool to compute the dissipative dynamics of quantum systems: the Bloch-Redfield Quantum Master Equation (QME) [45, 56, 57].⁸

As a first step, we need to move to the interaction picture, to focus just on the evolution induced by the interaction between system and environment. Given the “non-interacting” Hamiltonian $\hat{H}_0(t) = \hat{H}^{\text{S}}(t) + \hat{H}^{\text{B}}$, the corresponding free evolution operator is

$$\hat{U}_0(t, 0) = \text{Texp} \left(-\frac{i}{\hbar} \int_0^t dt' \hat{H}_0(t') \right) = \hat{U}_{0\text{S}}(t, 0) \otimes \hat{U}_{0\text{B}}(t, 0), \tag{D.15}$$

where Texp stands for the time-ordered exponential and $\hat{U}_{0\text{S}}(t, 0)$ and $\hat{U}_{0\text{B}}(t, 0)$ are the non-interacting propagators for the system and the bath respectively. The second equality holds simply because the system and bath Hamiltonians belong to different Hilbert spaces and therefore commute. The density matrix in the interaction representation,⁹

$$\hat{\rho}_{\text{tot},\text{I}}(t) = \hat{U}_0^{\dagger}(t, 0) \hat{\rho}_{\text{tot}}(t) \hat{U}_0(t, 0), \tag{D.16}$$

obeys a Liouville-von Neumann equation,

$$\frac{d}{dt} \hat{\rho}_{\text{tot},\text{I}}(t) = \frac{1}{i\hbar} \left[\hat{H}_{\text{SB},\text{I}}(t), \hat{\rho}_{\text{tot},\text{I}}(t) \right], \tag{D.17}$$

where $\hat{H}_{\text{SB},\text{I}}(t) = \hat{U}_0^{\dagger}(t, 0) \hat{H}^{\text{SB}}(t) \hat{U}_0(t, 0)$ is the system-bath Hamiltonian in interaction representation. Integrating Eq. (D.17) in the interval $(0, t)$ we have

$$\hat{\rho}_{\text{tot},\text{I}}(t) = \hat{\rho}_{\text{tot},\text{I}}(0) + \frac{i}{\hbar} \int_0^t dt_1 \left[\hat{H}_{\text{SB},\text{I}}(t_1), \hat{\rho}_{\text{tot},\text{I}}(t_1) \right]. \tag{D.18}$$

We can then iterate Eq. (D.18) to express $\hat{\rho}_{\text{tot},\text{I}}(t_1)$ on the r.h.s., to get

$$\begin{aligned}
\hat{\rho}_{\text{tot},\text{I}}(t) &= \hat{\rho}_{\text{tot},\text{I}}(0) + \frac{1}{i\hbar} \int_0^t dt_1 \left[\hat{H}_{\text{SB},\text{I}}(t_1), \hat{\rho}_{\text{tot},\text{I}}(0) \right] \\
&\quad - \frac{1}{\hbar^2} \int_0^t dt_1 \int_0^{t_1} dt_2 \left[\hat{H}_{\text{SB},\text{I}}(t_1), \left[\hat{H}_{\text{SB},\text{I}}(t_2), \hat{\rho}_{\text{tot},\text{I}}(t_2) \right] \right]. \tag{D.19}
\end{aligned}$$

⁸A totally equivalent derivation, making use of projector techniques, leads to the so-called *Nakajima-Zwanzig equation*, a non-Markovian QME which reduces, in the Markovian limit, to the same result we will derive below. This derivation using projector techniques is similar in spirit to a derivation one could give of the stochastic Schrödinger equation.

⁹Observe that the density matrix has to do with how the *states* evolve. In absence of interaction \hat{H}^{SB} , $\hat{\rho}_{\text{tot}}(t)$ would evolve as $\hat{\rho}_{\text{tot}}(t) = \hat{U}_0(t, 0) \hat{\rho}_{\text{tot}}(0) \hat{U}_0^{\dagger}(t, 0)$, hence $\hat{\rho}_{\text{tot},\text{I}}(t) = \hat{\rho}_{\text{tot},\text{I}}(0)$. So, the interaction representation “discounts” the state from the evolution occurring in absence of \hat{H}^{SB} .

At this point, we make the crucial assumption of **weak coupling**. We redefine $\widehat{H}_{\text{SB},\text{I}} \rightarrow g\widehat{H}_{\text{SB},\text{I}}$, with $g \ll 1$ which quantifies the coupling strength. Then, each occurrence of $\widehat{H}_{\text{SB},\text{I}}$ in Eq. (D.19) would yield a factor g in front. Moreover, the system's state is perturbatively expanded in g , so that $\hat{\rho}_{\text{tot},\text{I}}(t_2) = \hat{\rho}_{\text{tot},\text{I}}(0) + O(g)$ for $t_2 \in [0, t]$. We can thus write Eq. (D.19) up to second order in g as

$$\begin{aligned} \hat{\rho}_{\text{tot},\text{I}}(t) = & \hat{\rho}_{\text{tot},\text{I}}(0) + \frac{g}{i\hbar} \int_0^t dt_1 \left[\widehat{H}_{\text{SB},\text{I}}(t_1), \hat{\rho}_{\text{tot},\text{I}}(0) \right] \\ & - \frac{g^2}{\hbar^2} \int_0^t dt_1 \int_0^{t_1} dt_2 \left[\widehat{H}_{\text{SB},\text{I}}(t_1), \left[\widehat{H}_{\text{SB},\text{I}}(t_2), \hat{\rho}_{\text{tot},\text{I}}(0) \right] \right] + O(g^3). \end{aligned} \quad (\text{D.20})$$

To obtain a master equation in differential form, we take a time derivative and trace out the bath degrees of freedom, getting an evolution equation for the system alone, $\hat{\rho}_{\text{S}}(t) = \text{Tr}_{\text{B}}(\hat{\rho}_{\text{tot}})$. After this, we obtain

$$\frac{d}{dt} \hat{\rho}_{\text{S},\text{I}}(t) = -\frac{g^2}{\hbar^2} \int_0^t dt_2 \text{Tr}_{\text{B}} \left[\widehat{H}_{\text{SB},\text{I}}(t), \left[\widehat{H}_{\text{SB},\text{I}}(t_2), \hat{\rho}_{\text{tot},\text{I}}(0) \right] \right] + O(g^3). \quad (\text{D.21})$$

where the first correction in g , after tracing out the bath, is null due to the assumption that $\text{Tr}_{\text{B}}(\hat{\rho}_{\text{B}} \widehat{B}_{\nu\text{I}}(t)) = 0$. We can calculate the trace by using Eq. (D.3) and assuming that the system and the bath start in a separable state, see Eq. (D.4). The crucial quantity emerging from such a calculation is the free bath correlation function in Eq. (D.7):

$$C_{\nu\nu'}(t-t') = \text{Tr}_{\text{B}} \left(\widehat{B}_{\nu\text{I}}(t) \widehat{B}_{\nu'\text{I}}(t') \hat{\rho}_{\text{B}} \right), \quad (\text{D.22})$$

where we have implicitly assumed that $\hat{\rho}_{\text{B}}(0)$ is an equilibrium state of the bath. We find:

$$\frac{d}{dt} \hat{\rho}_{\text{S},\text{I}}(t) = -\frac{g^2}{\hbar^2} \sum_{\nu} \left(\left[\widehat{A}_{\nu\text{I}}(t), \widehat{S}_{\nu,\text{I}}(t) \hat{\rho}_{\text{S},\text{I}}(0) \right] + \text{H.c.} \right) + O(g^3), \quad (\text{D.23})$$

where we defined the convoluted and integrated system operators

$$\widehat{S}_{\nu,\text{I}}(t) \equiv \sum_{\nu'} \int_0^t dt' C_{\nu\nu'}(t-t') \widehat{A}_{\nu'\text{I}}(t') = \sum_{\nu'} \int_0^t d\tau C_{\nu\nu'}(\tau) \widehat{A}_{\nu'\text{I}}(t-\tau). \quad (\text{D.24})$$

In the second equality we simply made the change of variable $t-t' = \tau$, to get an expression that will be useful later on. Since, up to zero order in g , $\hat{\rho}_{\text{S},\text{I}}(t) = \hat{\rho}_{\text{S},\text{I}}(0) + O(g)$, Eq. (D.23) can be equivalently rewritten, in "closed" differential form, as:

i

Bloch-Redfield QME.

$$\frac{d}{dt} \hat{\rho}_{\text{S},\text{I}}(t) = -\frac{g^2}{\hbar^2} \sum_{\nu} \left(\left[\widehat{A}_{\nu\text{I}}(t), \widehat{S}_{\nu,\text{I}}(t) \hat{\rho}_{\text{S},\text{I}}(t) \right] + \text{H.c.} \right) + O(g^3), \quad (\text{D.25})$$

valid up to second order in g . This is the so-called **Bloch-Redfield quantum master equation** in interaction representation. Going back to the Schrödinger picture, Eq. (D.25) becomes

$$\frac{d}{dt} \hat{\rho}_{\text{S}}(t) = \frac{1}{i\hbar} \left[\widehat{H}^{\text{S}}(t), \hat{\rho}_{\text{S}}(t) \right] - \frac{g^2}{\hbar^2} \sum_{\nu} \left(\left[\widehat{A}_{\nu}, \widehat{S}_{\nu}(t) \hat{\rho}_{\text{S}}(t) \right] + \text{H.c.} \right) + O(g^3), \quad (\text{D.26})$$

where the convoluted operator in the Schrödinger picture now reads

$$\begin{aligned} \widehat{S}_{\nu}(t) \equiv \widehat{U}_0(t,0) \widehat{S}_{\nu,\text{I}}(t) \widehat{U}_0^\dagger(t,0) &= \sum_{\nu'} \int_0^t dt' C_{\nu\nu'}(t-t') \widehat{U}_0(t,t') \widehat{A}_{\nu'} \widehat{U}_0^\dagger(t,t') \\ &= \sum_{\nu'} \int_0^t d\tau C_{\nu\nu'}(\tau) \widehat{U}_0(t,t-\tau) \widehat{A}_{\nu'} \widehat{U}_0^\dagger(t,t-\tau), \end{aligned} \quad (\text{D.27})$$

and the second expression again is obtained by changing variables $t-t' = \tau$.

Notice that, as a result of the approximations done on $\hat{\rho}_{\text{tot},\text{I}}(t)$ and $\hat{\rho}_{\text{s},\text{I}}(t)$ to lowest order in g , we now have an equation that considers the evolution of the system disregarding completely the evolution of the bath, which is kept unchanged in time. Therefore, this approach is consistent with the application of the so-called *Born approximation* [56], *i.e.* neglecting the build-up of correlations — in essence, the system and the bath get *entangled* — between system and bath in time:

$$\hat{\rho}_{\text{tot},\text{I}}(t) \simeq \hat{\rho}_{\text{s},\text{I}}(t) \otimes \hat{\rho}_{\text{B}}. \quad (\text{D.28})$$

Moreover, notice that the QME only depends on the system's state at time t and not on previous times. This lack of memory is usually called *first Markov approximation*. But this does not mean that Eqs. (D.25) and (D.26) describe a truly Markovian interaction. Indeed, the non-Markovian nature of such equations is hidden in the fact that the operator $\hat{S}_\nu(t)$ appearing in Eq. (D.27) *depends on the past* through the integral over t' . However, in many physical situations, it is possible to perform a further simplifying assumption, called the *second Markov approximation* [56, 57]. Suppose one can define a characteristic time-scale of the bath τ_B , after which the bath correlation functions go to zero, $C_{\nu\nu'}(\tau > \tau_B) \simeq 0$. Then, one often assumes that the system's dynamics is much slower than the bath one, so that $t \gg \tau_B$ in Eq. (D.27). This means that the system's dynamics is insensible to the short memory of the bath, leading to an effective Markovian system's dynamics. In this setting, for all $t \gg \tau_B$, Eqs. (D.24) and (D.27) can be approximated with

$$\hat{S}_\nu(t) \longrightarrow \hat{S}_\nu^\infty(t) = \sum_{\nu'} \int_0^\infty d\tau C_{\nu\nu'}(\tau) \hat{U}_0(t, t-\tau) \hat{A}_{\nu'} \hat{U}_0^\dagger(t, t-\tau) \quad (\text{D.29})$$

$$\hat{S}_{\nu,\text{I}}(t) \longrightarrow \hat{S}_{\nu,\text{I}}^\infty(t) = \sum_{\nu'} \int_0^\infty d\tau C_{\nu\nu'}(\tau) \hat{A}_{\nu',\text{I}}(t-\tau), \quad (\text{D.30})$$

where we sent the upper limit of the integral in τ to infinity. Within this approximation, we can now regard Eqs. (D.25) and (D.26) as describing a Markovian dynamics.¹⁰

A very important property of Eq. (D.26) is that *it preserves the trace of the density matrix*. Indeed,

$$\frac{d}{dt} \text{Tr}_{\text{s}}(\hat{\rho}_{\text{s}}(t)) = \text{Tr}_{\text{s}}\left(\frac{d}{dt} \hat{\rho}_{\text{s}}(t)\right) = 0, \quad (\text{D.31})$$

the last equality coming from the fact that in Eq. (D.26) only commutators appear and their trace must be zero because of the cyclic property of the trace. Moreover, observe that the right-hand side of Eq. (D.26) is manifestly Hermitean, which implies that *the Hermitean nature of $\hat{\rho}_{\text{s}}(t)$ is evidently preserved* during the evolution. Unfortunately, the *positivity* of the system's density matrix *is not a priori preserved* by the Bloch-Redfield quantum master equation. However, there are some special cases in which one can write the Bloch-Redfield QME in Lindblad form, thus guaranteeing positivity preservation. This topic is discussed further below.

D.3. The secular approximation and the Lindblad form

Under further specific approximations, it is possible to cast the Bloch-Redfield QME Eq. (D.26) in Lindblad form, thus ensuring positivity preservation. For example, a bath with no memory at all, *i.e.* $\tau_B \rightarrow 0$, leads almost straightforwardly to a Lindblad QME. However, this is an extreme limit and we will not use it. On the other hand, one can also recover the Lindblad form by applying the so-called **Rotating-Wave Approximation** (RWA) [56, 57]. Despite some similarities, there may be different ways of performing the RWA, which lead to different equations. For systems with sufficiently slow drivings (or simply with static Hamiltonians), one may employ the RWA by looking at the system energy levels [60], as detailed in Sec D.3.1. For periodic drivings, one can exploit the Floquet representation of states and perform the RWA according to the system's quasi-energies [61, 62].

¹⁰Notice that in general t does not disappear from $\hat{S}_\nu^\infty(t)$, because the system Hamiltonian might depend on time. If there is no time dependence of $\hat{H}^{\text{s}}(t)$, then \hat{S}_ν^∞ is time-independent.

D.3.1. Rotating-wave (or secular) approximation

We will consider here a *time-independent* system and show how we can get a Lindblad QME with the RWA [56, 57].

We start from the QME in interaction representation, Eq. (D.25). We work in the basis of system eigenstates $\{|a\rangle\}$, where $\hat{H}^S |a\rangle = E_a |a\rangle$, inserting identities $\mathbb{1} = \sum_a |a\rangle\langle a|$ to get

$$\begin{aligned} \hat{A}_{\nu 1}(t) &= e^{\frac{i}{\hbar}\hat{H}^S t} \hat{A}_\nu e^{-\frac{i}{\hbar}\hat{H}^S t} = \sum_{ab} e^{i(E_a - E_b)t/\hbar} |a\rangle\langle a| \hat{A}_\nu |b\rangle\langle b| \\ &= \sum_{ab} e^{-i\omega_{ba}t} A_{\nu,ba}^* \hat{L}_{ba}^\dagger, \end{aligned} \quad (\text{D.32})$$

where $\hbar\omega_{ba} = E_b - E_a$ and:

$$A_{\nu,ab} = \langle a|\hat{A}_\nu|b\rangle = A_{\nu,ba}^* \quad \text{and} \quad \hat{L}_{ab} = |a\rangle\langle b| = \hat{L}_{ba}^\dagger. \quad (\text{D.33})$$

Similarly, we write:

$$\hat{A}_{\nu' 1}(t - \tau) = \sum_{a'b'} e^{i(E_{a'} - E_{b'})(t - \tau)/\hbar} A_{\nu',a'b'} \hat{L}_{a'b'}, \quad (\text{D.34})$$

hence:

$$\begin{aligned} \hat{S}_{\nu,1}^\infty(t) &= \sum_{\nu'} \int_0^\infty d\tau C_{\nu\nu'}(\tau) \hat{A}_{\nu' 1}(t - \tau) \\ &= \sum_{\nu'} \sum_{a'b'} \left(\int_0^\infty d\tau C_{\nu\nu'}(\tau) e^{i(E_{b'} - E_{a'})\tau/\hbar} \right) e^{i(E_{a'} - E_{b'})t/\hbar} A_{\nu',a'b'} \hat{L}_{a'b'} \\ &= \sum_{\nu'} \sum_{a'b'} \Gamma_{\nu\nu'}(\omega_{b'a'}) e^{-i\omega_{b'a'}t} A_{\nu',a'b'} \hat{L}_{a'b'}. \end{aligned} \quad (\text{D.35})$$

where we have defined the one-sided Fourier transform:¹¹

$$\Gamma_{\nu\nu'}(\omega) = \int_0^\infty d\tau C_{\nu\nu'}(\tau) e^{i\omega\tau}, \quad (\text{D.36})$$

and posed $\hbar\omega_{b'a'} = E_{b'} - E_{a'}$. We can insert these expressions in the Bloch-Redfield QME in interaction picture, obtaining

$$\begin{aligned} \frac{d}{dt} \hat{\rho}_{S,1}(t) &= -\frac{g^2}{\hbar^2} \sum_{\nu\nu'} \sum_{aba'b'} \left(e^{-i(\omega_{ba} + \omega_{b'a'})t} A_{\nu,ba}^* A_{\nu',a'b'} \Gamma_{\nu\nu'}(\omega_{b'a'}) \times \right. \\ &\quad \left. \times \left[\hat{L}_{ba}^\dagger, \hat{L}_{a'b'} \hat{\rho}_{S,1}(t) \right] + \text{H.c.} \right). \end{aligned} \quad (\text{D.37})$$

Let us take a look at the exponential $e^{-i(\omega_{ba} + \omega_{b'a'})t} = e^{i(E_a - E_b + E_{a'} - E_{b'})t/\hbar}$. If t is large enough the corresponding factor in the master equation would oscillate fast, and average out during the evolution, unless there is a precise matching of the energies. Therefore, it would be legitimate to neglect these terms, performing the so-called *rotating-wave approximation* (RWA), sometimes also referred to as *secular approximation* [56, 57]:

$$e^{i(E_a - E_b + E_{a'} - E_{b'})t/\hbar} \xrightarrow{\text{RWA}} \delta_{\omega_{ba} + \omega_{b'a'}, 0}. \quad (\text{D.38})$$

We also recall that $\Gamma_{\nu\nu'}$ can be expressed in terms of the ordinary Fourier transform $\gamma_{\nu\nu'}$ and its Hilbert transform $\sigma_{\nu\nu'}$ (both Hermitean matrices):

$$\Gamma_{\nu\nu'}(\omega) = \frac{1}{2} \gamma_{\nu\nu'}(\omega) + i\sigma_{\nu\nu'}(\omega). \quad (\text{D.39})$$

¹¹Notice that the second Markov approximation consisted in having the upper limit in the integral to ∞ . In the original derivation, the upper limit would be t .

This suggests defining:

$$\begin{aligned}\gamma_{ba,a'b'} &= \frac{g^2}{\hbar^2} \sum_{\nu\nu'} \delta_{\omega_{b'a'}+\omega_{ba},0} A_{\nu,ba}^* A_{\nu',a'b'} \gamma_{\nu\nu'}(\omega_{b'a'}) \\ &= \frac{g^2}{\hbar^2} \sum_{\nu\nu'} \sum_{\omega} \delta_{\omega_{b'a'},\omega} \delta_{\omega_{ab},\omega} A_{\nu,ba}^* A_{\nu',a'b'} \gamma_{\nu\nu'}(\omega),\end{aligned}\quad (\text{D.40})$$

where the second form is useful to show that this matrix is Hermitean and *positive*.¹²

The terms involving $\sigma_{\nu\nu'}$ partly cancel upon taking the complex conjugate. After exchanging some dummy indices and using the Hermitean nature of all the quantities involved we arrived at a final expression of the form:

$$\frac{d}{dt} \hat{\rho}_{s,I}(t) = \frac{1}{i\hbar} \left[\hat{H}_{\text{LS}}, \hat{\rho}_{s,I}(t) \right] + \sum_{ab,a'b'} \gamma_{ba,a'b'} \left(\hat{L}_{a'b'} \hat{\rho}_{s,I}(t) \hat{L}_{ba}^\dagger - \frac{1}{2} \left\{ \hat{L}_{ba}^\dagger \hat{L}_{a'b'}, \hat{\rho}_{s,I}(t) \right\} \right), \quad (\text{D.42})$$

where we observe that there is a *Lamb-shift* term originating from the $\sigma_{\nu\nu'}$ terms (taking due notice to the fact that $\hat{L}_{ba}^\dagger \hat{L}_{a'b'} = \delta_{b,a'} \hat{L}_{ab'}$):

$$\hat{H}_{\text{LS}} = \frac{g^2}{\hbar} \sum_{ab} \delta_{\omega_{ba},0} \left(\sum_{\nu\nu'} \sum_{b'} A_{\nu,b'a}^* A_{\nu',b'b} \sigma_{\nu\nu'}(\omega_{bb'}) \right) \hat{L}_{ab}. \quad (\text{D.43})$$

Notice that the Lamb-shift term couples levels which are precisely degenerate in energy. Hence, if you switch representation, from interaction to Schrödinger, the phase factors cancel. Moreover, you immediately conclude that it commutes with the system Hamiltonian \hat{H}^s and simply splits the possible degeneracy of the unperturbed levels.

We can now rewrite Eq. (D.42) in the Schrödinger representation by using

$$\frac{d}{dt} \hat{\rho}_s(t) = \frac{1}{i\hbar} \left[\hat{H}^s, \hat{\rho}_s(t) \right] + \hat{U}_{0s}(t,0) \frac{d}{dt} \hat{\rho}_{s,I}(t) \hat{U}_{0s}^\dagger(t,0),$$

and the free time evolution operator $\hat{U}_{0s}(t,0) = e^{-\frac{i}{\hbar} \hat{H}^s t}$, which however does not bring any extra phase-factors, due to the energy conservation constraints intrinsic in the RWA.

1

Bloch-Redfield RWA-QME. The final form of the Bloch-Redfield **RWA-QME** in Schrödinger representation is therefore:

$$\frac{d}{dt} \hat{\rho}_s(t) = \frac{1}{i\hbar} \left[\hat{H}^s + \hat{H}_{\text{LS}}, \hat{\rho}_s(t) \right] + \sum_{ab,a'b'} \gamma_{ba,a'b'} \left(\hat{L}_{a'b'} \hat{\rho}_s(t) \hat{L}_{ba}^\dagger - \frac{1}{2} \left\{ \hat{L}_{ba}^\dagger \hat{L}_{a'b'}, \hat{\rho}_s(t) \right\} \right). \quad (\text{D.44})$$

Such QME can be brought to a standard form — the so-called *Lindblad* form — by diagonalizing the positive Hermitean matrix of $\gamma_{ba,a'b'}$.

¹²Indeed:

$$\begin{aligned}\sum_{ba,a'b'} x_{ba}^* \gamma_{ba,a'b'} x_{a'b'} &= \frac{g^2}{\hbar^2} \sum_{\nu\nu'} \sum_{\omega} \overbrace{\left(\sum_{ba} \delta_{\omega_{ab},\omega} x_{ba}^* A_{\nu,ba}^* \right)}^{x_{\nu'}^*(\omega)} \gamma_{\nu\nu'}(\omega) \overbrace{\left(\sum_{a'b'} \delta_{\omega_{b'a'},\omega} x_{a'b'} A_{\nu',a'b'} \right)}^{x_{\nu'}(\omega)} \\ &= \frac{g^2}{\hbar^2} \sum_{\omega} \sum_{\nu\nu'} x_{\nu'}^*(\omega) \gamma_{\nu\nu'}(\omega) x_{\nu'}(\omega) \geq 0,\end{aligned}\quad (\text{D.41})$$

where we have used the fact that $\gamma_{\nu\nu'}$ is positive for any ω .

D.3.2. The Lindblad form

The matrix $\gamma_{ba,a'b'}$ is a positive Hermitean $d_S^2 \times d_S^2$ matrix, where $d_S = \dim(\mathcal{H}_S)$. We can therefore *diagonalize* it with a unitary transformation \mathbb{U} :

$$\mathbb{U}^\dagger \gamma \mathbb{U} = \text{diag}(\gamma_\mu),$$

where $\gamma_\mu > 0$ are the eigenvalues. One can show that:

$$\sum_{ab,a'b'} \hat{L}_{ba}^\dagger \gamma_{ba,a'b'} \hat{L}_{a'b'} = \sum_\mu \hat{L}_\mu^\dagger \hat{L}_\mu, \quad (\text{D.45})$$

where

$$\hat{L}_{ab} = \sum_\mu U_{ab,\mu} \hat{L}_\mu \quad (\text{D.46})$$

and the inverse is simple.

i **Lindblad form of the Bloch-Redfield RWA-QME.** One can show that the full QME can be put in the form:

$$\frac{d}{dt} \hat{\rho}_S(t) = \frac{1}{i\hbar} \left[\hat{H}^S + \hat{H}_{\text{LS}}, \hat{\rho}_S(t) \right] + \sum_\mu \gamma_\mu \left(\hat{L}_\mu \hat{\rho}_S(t) \hat{L}_\mu^\dagger - \frac{1}{2} \left\{ \hat{L}_\mu^\dagger \hat{L}_\mu, \hat{\rho}_S(t) \right\} \right). \quad (\text{D.47})$$

which is the *Lindblad form* of the Bloch-Redfield QME obtained after the RWA is performed.

D.3.3. Non-degenerate spectrum and population dynamics

The final possible twist of the story is when the system \hat{H}^S has a *non-degenerate spectrum*, which implies:

$$\delta_{\omega_{b'a'}, \omega_{ab}} \implies \delta_{a,b} \delta_{a',b'} + \delta_{a,b'} \delta_{a',b} (1 - \delta_{a,b}), \quad (\text{D.48})$$

where the second term with $(1 - \delta_{a,b})$ prevents the double counting of the case where $a = b = a' = b'$. As a consequence, the RWA-QME for non-degenerate spectrum in Schrödinger representation is:

$$\begin{aligned} \frac{d}{dt} \hat{\rho}_S(t) = & \frac{1}{i\hbar} \left[\hat{H}^S + \hat{H}_{\text{LS}}, \hat{\rho}_S(t) \right] + \sum_{ab}^{\substack{a \neq b \\ ab}} \gamma_{ba,ba} \left(\hat{L}_{ba} \hat{\rho}_S(t) \hat{L}_{ba}^\dagger - \frac{1}{2} \left\{ \hat{L}_{ba}^\dagger \hat{L}_{ba}, \hat{\rho}_S(t) \right\} \right) + \\ & + \sum_{ab} \gamma_{aa,bb} \left(\hat{L}_{bb} \hat{\rho}_S(t) \hat{L}_{aa}^\dagger - \frac{1}{2} \left\{ \hat{L}_{aa}^\dagger \hat{L}_{bb}, \hat{\rho}_S(t) \right\} \right). \end{aligned} \quad (\text{D.49})$$

and the Lamb-shift term reads:

$$\hat{H}_{\text{LS}} = \frac{g^2}{\hbar} \sum_a \left(\sum_{\nu\nu'} \sum_{b'} A_{\nu,b'a}^* A_{\nu',b'a} \sigma_{\nu\nu'}(\omega_{ab'}) \right) \hat{L}_{aa}. \quad (\text{D.50})$$

Notice that the first term, involving $\gamma_{ba,ba}$, is responsible for transitions between energy levels, since we have $a \neq b$. The second term, involving $\gamma_{aa,bb}$, is a sum of terms proportional to $\gamma_{\nu\nu'}(\omega = 0)$: it does not involve transitions (indeed $\hat{L}_{aa}^\dagger = \hat{L}_{aa}$) and is only responsible for the so-called *pure dephasing*.

Furthermore, it can be shown that Eq. (D.49) brings to a decoupling of the dynamics of populations and coherences of the density matrix [57], in the form of classical rate equations for the populations.

Indeed, one can show that the equation for the diagonal elements $(\hat{\rho}_s)_{aa}(t) = P_a(t)$ — known as *populations* — has the following form:

$$\begin{aligned} \frac{d}{dt}P_a(t) &= \sum_{a' \neq a} \gamma_{a \leftarrow a'} P_{a'} - \left(\sum_{a' \neq a} \gamma_{a' \leftarrow a} \right) P_a \\ &= \sum_{a'} \gamma_{a \leftarrow a'} P_{a'} - \left(\sum_{a'} \gamma_{a' \leftarrow a} \right) P_a \end{aligned} \quad (\text{D.51})$$

where

$$\gamma_{a \leftarrow a'} \stackrel{\text{def}}{=} \gamma_{aa',aa'} \quad (\text{D.52})$$

and the second expression uses the fact that the diagonal $a' = a$ term cancels from the two contributions. Notice that this equation looks precisely as a *classical master equation* would look like, i.e., the diagonal matrix elements of the density matrix are totally independent of the off-diagonal elements $(\hat{\rho}_s)_{ab}(t)$, the so-called *coherences*, and obey a classical master equation, with rates $\gamma_{a \leftarrow a'}$ calculated quantum-mechanically.

D.4. Application to a two-level system

Suppose we have a generic two-level system. Its Hamiltonian, neglecting a constant energy shift proportional to the identity, can always be written in terms of Pauli matrices as

$$\hat{H}^s = \mathbf{h} \cdot \hat{\boldsymbol{\sigma}}, \quad (\text{D.53})$$

where \mathbf{h} is a real three-component vector \mathbf{h} and $\hat{\boldsymbol{\sigma}}$ is the vector of Pauli matrices. The two energies are $\pm E$ with $E = |\mathbf{h}|$, hence we can define the energy splitting $\hbar\omega_0 = \Delta E = 2|\mathbf{h}|$.

We consider a coupling to an environment, with interaction $\hat{H}^{sB} = \hat{A} \otimes \hat{B}$ and the operator acting on the system Hilbert space is in a generic direction in spin space

$$\hat{A} = \boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}, \quad (\text{D.54})$$

with $|\boldsymbol{\lambda}| = 1$.

For this system, we are now going to write the Bloch-Redfield RWA-QME. We will proceed with two techniques. The first is computationally convenient, and amounts to writing the QME in a Bloch vector representation. The second is physically more transparent, and identifies the relevant Lindblad operators appearing in the problem.

We start from Eq. (D.49) which we can recast in the equivalent form:

$$\begin{aligned} \frac{d}{dt}\hat{\rho}_s &= \frac{1}{i\hbar} \left[\hat{H}^s, \hat{\rho}_s \right] + \sum_{ab} \gamma_{ba,ba} \langle a | \hat{\rho}_s | a \rangle |b\rangle\langle b| + \sum_{a,b}^{b \neq a} \gamma_{aa,bb} |b\rangle\langle b| \hat{\rho}_s |a\rangle\langle a| \\ &\quad - \frac{1}{2} \sum_a \left(\sum_b \gamma_{ba,ba} \right) \left(|a\rangle\langle a| \hat{\rho}_s + \text{H.c.} \right), \end{aligned} \quad (\text{D.55})$$

where we recall that $|a\rangle$ is an eigenstate of \hat{H}^s and the second (off-diagonal) term, for a two-level system, restricts $b = \bar{a}$, where $|\bar{a}\rangle$ is the opposite eigenstate.

In order to go on with the calculation, it is very convenient to write all the operators in Bloch

notation.¹³ We will write the time-evolved system state as

$$\hat{\rho}_s(t) = \frac{1}{2} (\mathbb{1} + \mathbf{p}(t) \cdot \hat{\boldsymbol{\sigma}}) \quad (\text{D.56})$$

and the system's eigenstate projectors as

$$|a\rangle\langle a| = \frac{1}{2} (\mathbb{1} + \mathbf{p}_a \cdot \hat{\boldsymbol{\sigma}}) . \quad (\text{D.57})$$

This will be particularly convenient because the ground and excited states of the two-level system in Eq. (D.53) are simply represented by

$$\begin{aligned} \mathbf{p}_g &= -\frac{\mathbf{h}}{|\mathbf{h}|} \\ \mathbf{p}_e &= +\frac{\mathbf{h}}{|\mathbf{h}|} . \end{aligned} \quad (\text{D.58})$$

There are some quantities we need to compute. Let us start with

$$\begin{aligned} |a\rangle\langle a| \hat{\rho}_s &= \frac{1}{4} (\mathbb{1} + \mathbf{p}_a \cdot \hat{\boldsymbol{\sigma}}) (\mathbb{1} + \mathbf{p} \cdot \hat{\boldsymbol{\sigma}}) = \\ &= \frac{1}{4} \left[(1 + \mathbf{p}_a \cdot \mathbf{p}) \mathbb{1} + (\mathbf{p} + \mathbf{p}_a + i(\mathbf{p}_a \times \mathbf{p})) \cdot \hat{\boldsymbol{\sigma}} \right] \end{aligned}$$

and, consequently,

$$\langle a | \hat{\rho}_s | a \rangle = \text{Tr} (|a\rangle\langle a| \hat{\rho}_s) = \frac{1}{2} (1 + \mathbf{p} \cdot \mathbf{p}_a) .$$

The last term in Eq. (D.55) shows factors like

$$\begin{aligned} |b\rangle\langle b| \hat{\rho}_s |a\rangle\langle a| &= \frac{1}{8} \left[\left(1 + (\mathbf{p}_b + \mathbf{p}_a) \cdot \mathbf{p} + \mathbf{p}_b \cdot \mathbf{p}_a + i(\mathbf{p}_b \times \mathbf{p}) \cdot \mathbf{p}_a \right) \mathbb{1} + \right. \\ &\quad \left. + \left((1 + \mathbf{p}_a \cdot \mathbf{p}) \mathbf{p}_b + (1 + \mathbf{p}_b \cdot \mathbf{p}) \mathbf{p}_a + (1 - \mathbf{p}_b \cdot \mathbf{p}_a) \mathbf{p} + \right. \right. \\ &\quad \left. \left. + i(\mathbf{p}_b \times \mathbf{p} + \mathbf{p} \times \mathbf{p}_a + \mathbf{p}_b \times \mathbf{p}_a) \right) \cdot \hat{\boldsymbol{\sigma}} \right] , \end{aligned}$$

which simplify considerably if we take $b = \bar{a}$, since this implies $\mathbf{p}_{\bar{a}} = -\mathbf{p}_a$, leading to

$$|\bar{a}\rangle\langle \bar{a}| \hat{\rho}_s |a\rangle\langle a| = \frac{1}{4} \left(\mathbf{p} - (\mathbf{p}_a \cdot \mathbf{p}) \mathbf{p}_a - i \mathbf{p}_a \times \mathbf{p} \right) \cdot \hat{\boldsymbol{\sigma}} .$$

Finally, the rate terms to be computed are

$$\begin{aligned} \gamma_{ba,ba} &= \frac{g^2}{\hbar^2} \gamma(\omega_{ab}) \langle a | (\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | b \rangle \langle b | (\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | a \rangle = \\ &= \frac{g^2}{\hbar^2} \gamma(\omega_{ab}) \text{Tr} \left(|a\rangle\langle a| (\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | b \rangle \langle b | (\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) \right) = \\ &= \frac{g^2}{\hbar^2} \gamma(\omega_{ab}) \frac{1}{2} \left(1 - \mathbf{p}_a \cdot \mathbf{p}_b + 2(\boldsymbol{\lambda} \cdot \mathbf{p}_a)(\boldsymbol{\lambda} \cdot \mathbf{p}_b) \right) , \\ \gamma_{aa,bb} &= \frac{g^2}{\hbar^2} \gamma(0) \langle a | (\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | a \rangle \langle b | (\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | b \rangle = \\ &= \frac{g^2}{\hbar^2} \gamma(0) \text{Tr} \left((\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | a \rangle \langle a | \right) \text{Tr} \left((\boldsymbol{\lambda} \cdot \hat{\boldsymbol{\sigma}}) | b \rangle \langle b | \right) = \\ &= \frac{g^2}{\hbar^2} \gamma(0) (\boldsymbol{\lambda} \cdot \mathbf{p}_a)(\boldsymbol{\lambda} \cdot \mathbf{p}_b) . \end{aligned}$$

¹³The following identity will be extremely useful to carry out the calculations:

$$(\mathbf{a} \cdot \hat{\boldsymbol{\sigma}}) (\mathbf{b} \cdot \hat{\boldsymbol{\sigma}}) = (\mathbf{a} \cdot \mathbf{b}) \mathbb{1} + i(\mathbf{a} \times \mathbf{b}) \cdot \hat{\boldsymbol{\sigma}}$$

With all these ingredients, one can now compute all the terms appearing in Eq. (D.55). Putting all the pieces together, we get

$$\dot{\mathbf{p}} = \frac{2}{\hbar} \mathbf{h} \times \mathbf{p} - \gamma_{\text{R}} \frac{\mathbf{h}(\mathbf{h} \cdot \mathbf{p})}{|\mathbf{h}|^2} - \gamma_{\text{D}} \left(\mathbf{p} - \frac{\mathbf{h}(\mathbf{h} \cdot \mathbf{p})}{|\mathbf{h}|^2} \right) - \gamma_{\text{R}} \frac{\mathbf{h}}{|\mathbf{h}|} \tanh(\beta|\mathbf{h}|). \quad (\text{D.59})$$

Here, the first term describes the coherent evolution free precession of \mathbf{p} around \mathbf{h} , the second the relaxation of its *longitudinal* component $\mathbf{p}_{\parallel} = \mathbf{h}(\mathbf{h} \cdot \mathbf{p})/|\mathbf{h}|^2$, the third the decoherence of the *transverse* component $\mathbf{p}_{\perp} = \mathbf{p} - \mathbf{p}_{\parallel}$, and last its steady-state thermal equilibrium value.

The *relaxation* and *decoherence* rates γ_{R} and γ_{D} , calculated within the usual weak-coupling assumption [45, 60, 63], are given by:

$$\gamma_{\text{R}} = \left(1 - \frac{(\boldsymbol{\lambda} \cdot \mathbf{h})^2}{|\mathbf{h}|^2} \right) \frac{\gamma(\omega_0) + \gamma(-\omega_0)}{\hbar^2} \quad (\text{D.60a})$$

$$\gamma_{\varphi} = \frac{(\boldsymbol{\lambda} \cdot \mathbf{h})^2}{|\mathbf{h}|^2} \frac{2\gamma(0)}{\hbar^2} \quad (\text{D.60b})$$

$$\gamma_{\text{D}} = \left(\frac{\gamma_{\text{R}}}{2} + \gamma_{\varphi} \right) \quad (\text{D.60c})$$

where $\omega_0 = 2|\mathbf{h}|/\hbar$ is the free precession frequency and $\gamma(\omega)$ is the Fourier transform of the free thermal bath correlation function¹⁴

$$\gamma(\omega) = \int_{-\infty}^{+\infty} dt e^{i\omega t} \langle \hat{B}(t) \hat{B}(0) \rangle_{\text{B}}, \quad (\text{D.61})$$

for which one can show [60] that:

$$\gamma(0) = \frac{2\pi}{\hbar} k_{\text{B}} T \lim_{\omega \rightarrow 0} \frac{J(\omega)}{\omega} \quad (\text{D.62})$$

and

$$\gamma(\omega_0) + \gamma(-\omega_0) = 2\pi J(\omega_0) \coth(\beta|\mathbf{h}|). \quad (\text{D.63})$$

Notice that the direction $\boldsymbol{\lambda}$ of the coupling in spin space influences γ_{R} and γ_{φ} . In particular, for $\boldsymbol{\lambda}$ in the direction of \mathbf{h} , relaxation would be absent. Notice also that the so-called pure-dephasing γ_{φ} constant depends crucially on the choice of the spectral function $J(\omega)$ of the bath: it is not well-defined (formally divergent) for a sub-Ohmic bath $J(\omega) \propto \omega^{\nu}$ with $\nu \in [0, 1)$, it is finite and non-zero for the Ohmic case $\nu = 1$, while it vanishes in the super-Ohmic case $\nu > 1$.

D.4.1. Lindblad form for the two-level system

It is instructive to redo the same calculation by looking for the explicitly diagonal Lindblad form. We observe that the rate constants are (recall that $\omega_0 = \omega_{eg} = -\omega_{ge}$):

$$\left\{ \begin{array}{l} \gamma_{ee,ee} = \frac{\gamma(0)}{\hbar^2} (\boldsymbol{\lambda} \cdot \mathbf{p}_e)^2 = \frac{\gamma(0)}{\hbar^2} (\boldsymbol{\lambda} \cdot \mathbf{p}_g)^2 = \gamma_{gg,gg} \equiv \frac{\gamma_{\varphi}}{2} \\ \gamma_{ee,gg} = \frac{\gamma(0)}{\hbar^2} (\boldsymbol{\lambda} \cdot \mathbf{p}_e)(\boldsymbol{\lambda} \cdot \mathbf{p}_g) = \gamma_{gg,ee} = -\frac{\gamma_{\varphi}}{2} \\ \gamma_{eg,eg} = \frac{\gamma(-\omega_0)}{\hbar^2} (1 - (\boldsymbol{\lambda} \cdot \mathbf{p}_g)^2) \equiv \gamma_{e \leftarrow g} \\ \gamma_{ge,ge} = \frac{\gamma(+\omega_0)}{\hbar^2} (1 - (\boldsymbol{\lambda} \cdot \mathbf{p}_g)^2) \equiv \gamma_{g \leftarrow e} \end{array} \right. \quad (\text{D.64})$$

¹⁴Observe that in both terms we have the Fourier transform of the symmetrised correlation function, $S_B(\omega) = \gamma(\omega) + \gamma(-\omega)$. γ_{φ} is known as *pure-dephasing* rate.

If we assume the ordering $1 = eg, 2 = ge, 3 = ee, 4 = gg$ we can write the 4×4 matrix γ as:

$$\gamma = \begin{pmatrix} \gamma_{e \leftarrow g} & 0 & 0 & 0 \\ 0 & \gamma_{g \leftarrow e} & 0 & 0 \\ 0 & 0 & +\frac{\gamma_\varphi}{2} & -\frac{\gamma_\varphi}{2} \\ 0 & 0 & -\frac{\gamma_\varphi}{2} & +\frac{\gamma_\varphi}{2} \end{pmatrix}. \quad (\text{D.65})$$

Evidently, the only 2×2 block that needs to be diagonalised is that on the bottom block, which is $\frac{\gamma_\varphi}{2}(\mathbb{1} - \hat{\sigma}^x)$, with eigenvalues γ_φ and 0. Evidently the full 4×4 matrix \mathbb{U} that diagonalises the problem is:

$$\mathbb{U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & +\frac{1}{\sqrt{2}} & +\frac{1}{\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & +\frac{1}{\sqrt{2}} \end{pmatrix} \implies \mathbb{U}^\dagger \gamma \mathbb{U} = \begin{pmatrix} \gamma_{e \leftarrow g} & 0 & 0 & 0 \\ 0 & \gamma_{g \leftarrow e} & 0 & 0 \\ 0 & 0 & \gamma_\varphi & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (\text{D.66})$$

The 4 associated Lindblad operators are:

$$\begin{cases} \gamma_1 = \gamma_{e \leftarrow g} & \rightarrow & \hat{L}_1 = \hat{L}_{eg} = \hat{\sigma}^+ \\ \gamma_2 = \gamma_{g \leftarrow e} & \rightarrow & \hat{L}_2 = \hat{L}_{ge} = \hat{\sigma}^- \\ \gamma_3 = \gamma_\varphi & \rightarrow & \hat{L}_3 = \frac{1}{\sqrt{2}}(\hat{L}_{ee} - \hat{L}_{gg}) = \frac{1}{\sqrt{2}}\hat{\sigma}^z \\ \gamma_4 = 0 & \rightarrow & \hat{L}_4 = \frac{1}{\sqrt{2}}(\hat{L}_{ee} + \hat{L}_{gg}) = \frac{1}{\sqrt{2}}\mathbb{1} \end{cases} \quad (\text{D.67})$$

D.4.2. Decoherence and relaxation towards equilibrium

Let us visualise the numerical results obtained for convenience with a very specific choice of Hamiltonian and coupling:

$$\hat{H}^{\text{tot}} = \frac{\hbar\omega_0}{2}\hat{\sigma}^z + (\sin\theta\hat{\sigma}^x + \cos\theta\hat{\sigma}^z) \otimes \hat{B} + \hat{H}^{\text{B}}, \quad (\text{D.68})$$

where $\hbar\omega_0$ is the gap between the two system's eigenstates, while θ tunes the coupling direction with respect to the system energy basis.

We are interested in the dynamics of the system starting from a generic initial state $\hat{\rho}_s(0)$. We will now see the role of the two different time-scales which govern the system's time-evolution [63]. On the one hand, we have the so-called *dephasing* or *decoherence*, *i.e.* the off-diagonal elements of the density matrix in the system energy eigenbasis tend to zero after a characteristic time-scale τ_{decoh} . This means that the system tends to become a mixed state, losing the quantum superposition between the energy eigenstates. On the other hand, the populations of the density matrix, again in the system energy eigenbasis, tend to acquire a Boltzmann distribution depending on the bath temperature. This phenomenon is called *relaxation* and it also takes place within a proper relaxation time-scale τ_{R} . The interplay of relaxation and dephasing leads the system to reach a thermal steady state after a transient,

$$\hat{\rho}_s(t \rightarrow \infty) = \frac{e^{-\beta\hat{H}^{\text{S}}}}{\text{Tr}(e^{-\beta\hat{H}^{\text{S}}})}, \quad (\text{D.69})$$

where β is the bath inverse temperature. Notice that $\hat{\rho}_s(t \rightarrow \infty)$, written in the system energy eigenbasis, is diagonal.

It is instructive to visualize the processes of dephasing and relaxation on the Bloch sphere. We express the system density matrix in the energy eigenbasis as

$$\hat{\rho}_s^{\text{eig}}(t) = \frac{1}{2}(\mathbb{1} + \mathbf{p}(t) \cdot \hat{\boldsymbol{\sigma}}), \quad (\text{D.70})$$

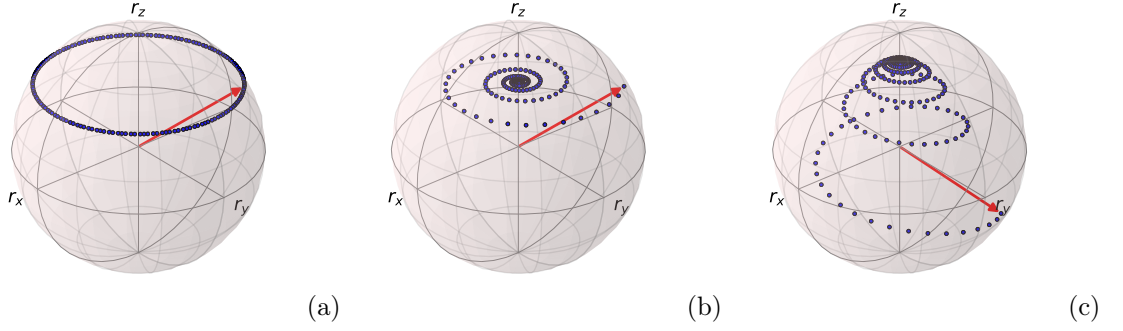


Figure D.1.: Dynamics of a spin-1/2 under the spin-boson Hamiltonian for (a) no dissipation, (b) pure-dephasing and (c) relaxation plus decoherence. In (a), the spin oscillates coherently. In (b), the coherences of the density matrix go to zero, but the populations do not change. Plots realized using the Qutip package [64,65].

where $\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t))$ and $\hat{\sigma} = (\hat{\sigma}^x, \hat{\sigma}^y, \hat{\sigma}^z)$. By using $\boldsymbol{\lambda} = (\sin \theta, 0, \cos \theta)$ and $\mathbf{h} = \frac{\hbar\omega_0}{2}\hat{\mathbf{z}}$, we find that:

$$\dot{\mathbf{p}} = \omega_0 \hat{\mathbf{z}} \times \mathbf{p} - \begin{pmatrix} \frac{\gamma_R}{2} + \gamma_\varphi & 0 & 0 \\ 0 & \frac{\gamma_R}{2} + \gamma_\varphi & 0 \\ 0 & 0 & \gamma_R \end{pmatrix} \cdot \mathbf{p} - \gamma_R p_z^{\text{eq}} \hat{\mathbf{z}}. \quad (\text{D.71})$$

The equation for $p_z(t)$ is evidently decoupled and has solution

$$p_z(t) = p_z(0) e^{-t/\tau_R} + p_z^{\text{eq}}, \quad (\text{D.72})$$

where $p_z^{\text{eq}} = \tanh(\beta\hbar\omega_0/2)$ characterises the thermal equilibrium for the populations. The other two components, p_x and p_y , are coupled by the coherent dynamics and have solution:

$$p_\pm(t) = p_x(t) \pm ip_y(t) = p_\pm(0) e^{\pm i\omega_0 t - t/\tau_{\text{decoh}}}. \quad (\text{D.73})$$

Here τ_{decoh} and τ_R are respectively the decoherence and relaxation time-scales, associated to the following corresponding rates [63]

$$\gamma_R = \frac{1}{\tau_R} = \frac{\gamma(\omega_0) + \gamma(-\omega_0)}{\hbar^2} \sin^2 \theta, \quad (\text{D.74})$$

$$\gamma_{\text{decoh}} = \frac{1}{\tau_{\text{decoh}}} = \frac{1}{2}\gamma_R + \gamma_\varphi = \frac{1}{2}\gamma_R + \frac{2\gamma(0)}{\hbar^2} \cos^2 \theta. \quad (\text{D.75})$$

These results are illustrated in Fig. D.1, where we show the dissipative dynamics of the Bloch vector $\mathbf{p}(t)$, starting from a generic initial pure state, indicated by the red arrows. Recall that pure states correspond to vectors that point at the surface of the Bloch sphere while, if the state becomes mixed, the associated vector points inside the sphere. Panel (a) shows non-dissipative dynamics: the state is always pure and thus the Bloch vector always points at the surface of the sphere, precessing at fixed frequency $\omega_0 = \Delta E/\hbar$. Panel (b) illustrates the peculiar pure-dephasing case, obtained for $\theta = 0$: the coherent oscillations of the previous case are damped in time, as the Bloch vector goes deeper and deeper inside the sphere. However, the value of $p_z(0)$ never changes, because populations cannot be modified by the interaction with the bath. Panel (c) eventually displays a generic relaxation plus decoherence process ($\theta > 0$), where now $p_z(t)$ relaxes to the equilibrium value p_z^{eq} .

Whenever the system Hamiltonian is time-dependent, the phenomena of dephasing and relaxation still occur, but in general it is more difficult to disentangle them. On the one hand, their time-scales are no more constant and the dynamics is more complicated than a pure exponential; on the other hand, the energy eigenbasis also depends on time.

E. Classical and Quantum Error Correction

Here I present some more advanced material on classical and quantum error correction. The main references are Ref. [50] for classical error correction, and Nielsen-Chuang [3] and Preskill's lecture notes, for quantum error correction.

E.1. Linear codes in classical error correction

Here I follow, with minor modifications, a standard reference for classical coding and error correction: the book by MacWilliams and Sloane [50].

Example 1: A parity check code. I start considering the encoding of $k = 3$ bits $\mathbf{u} = (u_1, u_2, u_3)^T \in \mathbb{B}^3$, by adding redundant information, into $n = 6$ bits (codewords) $\mathbf{x} = (x_1, \dots, x_6)^T \in \mathbb{B}^6$ provided we enforce $6 - 3 = 3$ independent constraints, here in the form of parity checks. More precisely, we take the first 3 bits of the codeword \mathbf{x} to coincide with the 3 message bits: $x_j = u_j$ for $j = 1, 2, 3$. The remaining 3 bits of \mathbf{x} (i.e., x_4, x_5, x_6) perform/enforce the following parity checks (in integer arithmetic modulo-2) for the message bits:

$$\begin{cases} x_2 + x_3 + x_4 = 0 \\ x_1 + x_3 + x_5 = 0 \pmod{2} \\ x_1 + x_2 + x_6 = 0 \end{cases} \quad (\text{E.1})$$

These 3 equations can be written more compactly in terms of a 3×6 matrix \mathbb{H} as follows:

$$\mathbb{H}\mathbf{x} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right] \mathbf{x} = \left[\mathbb{A} | \mathbb{0}_3 \right] \mathbf{x} = \mathbf{0} \pmod{2}, \quad (\text{E.2})$$

where we have highlighted a (here 3×3) matrix \mathbb{A} which will play a role in the following. It takes little algebra to show that the $2^3 = 8$ possible messages \mathbf{u} generates the corresponding 8 codewords ¹

\mathbf{u}	\mathbf{x}
000	000 000
001	001 110
010	010 101
011	011 011
100	100 011
101	101 101
110	110 110
111	111 000

(E.3)

Clearly, transmitting a codeword of length $n = 6$ to convey the information of a $k = 3$ bit message, means that the rate of transfer of information is $R = k/n = 0.5$ here.

¹For a more compact notation, rather than writing, say, $\mathbf{x} = (0, 0, 1, 1, 1, 0)^T$, I will denote such an \mathbf{x} as (001 110). The same condensed notation is applied to the message vectors \mathbf{u} . I also add a small space, for visibility purposes, to separate the message bits from the parity check bits in the codewords \mathbf{x} .

1

The linear code $[n, k]$. Let $\mathbf{u} \in \mathbb{B}^k$ be k -bit messages. For $n \geq k$, we consider codewords $\mathbf{x} \in \mathbb{B}^n$ such that $x_j = u_j$ for $j = 1 \cdots k$, while the remaining $n - k$ bits $x_{k+1} \cdots x_n$ enforce parity checks through the following linear $n - k$ linear equations

$$\mathbb{H} \mathbf{x} = [\mathbb{A} | \mathbb{I}_{n-k}] \mathbf{x} = \mathbf{0} \pmod{2}, \tag{E.4}$$

where $\mathbb{H} = [\mathbb{A} | \mathbb{I}_{n-k}]$ is an $(n - k) \times n$ check matrix, with $n - k$ linearly independent rows, \mathbb{A} is an $(n - k) \times k$ fixed matrix and \mathbb{I}_{n-k} the identity $(n - k)$ -dimensional matrix. We define the code \mathcal{C} to be the set of \mathbf{x} satisfying these linear equations:

$$\mathcal{C} = \{ \mathbf{x} \in \mathbb{B}^n \mid \mathbb{H} \mathbf{x} = \mathbf{0} \pmod{2} \}. \tag{E.5}$$

Equivalently, the code is defined to be the **kernel** of \mathbb{H} , $\mathcal{C} = \ker(\mathbb{H})$.^a n is the *length* of the code, k its *dimension* (there are 2^k codewords), and $R = k/n$ is the *transmission rate* or *efficiency*. The check matrix \mathbb{H} in the form given above is known as the **standard form**: we will see that there are equivalent ways of expressing the parity check equations.

^aA code encoding k bits has 2^k possible codewords: hence the kernel of \mathbb{H} must be k -dimensional. By the rank-nullity theorem of linear algebra:

$$\text{rank}(\mathbb{H}) + \dim(\text{Ker}(\mathbb{H})) = n,$$

and therefore we require that $r = \text{rank}(\mathbb{H})$ is $r = n - k$. Hence \mathbb{H} must have $r = n - k$ independent rows.

Example 2: Repetition code with $k = 1$ and $n = 3$. Consider the following 2×3 check matrix \mathbb{H} with associated linear equations:

$$\mathbb{H} \mathbf{x} = \left[\begin{array}{c|cc} 1 & 1 & 0 \\ 1 & 0 & 1 \end{array} \right] \mathbf{x} = [\mathbb{A} | \mathbb{I}_2] \mathbf{x} = \mathbf{0} \pmod{2}. \tag{E.6}$$

Here the message has a single bit $u_1 = 0, 1$. As usual $x_1 = u_1$ and bits x_2, x_3 require

$$x_1 + x_2 = 0 \quad \& \quad x_1 + x_3 = 0 \pmod{2},$$

i.e., $u_1 = x_1 = x_2 = x_3$.² Hence the two codewords are $u_1 = 0 \rightarrow \mathbf{x}_1 = (000)$ and $u_1 = 1 \rightarrow \mathbf{x}_2 = (111)$. This is nothing but a repetition code with $n = 3$, with a transmission rate $R = 1/3$.

Example 3: Repetition code with $k = 1$ and $n = 5$. Consider the following $(n - k, n) = 4 \times 5$ check matrix \mathbb{H} with associated linear equations:

$$\mathbb{H} \mathbf{x} = \left[\begin{array}{c|cccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{array} \right] \mathbf{x} = [\mathbb{A} | \mathbb{I}_4] \mathbf{x} = \mathbf{0} \pmod{2}. \tag{E.7}$$

Here the message has a single bit $u_1 = 0, 1$. As usual $x_1 = u_1$ and bits x_2, x_3, x_4, x_5 require

$$x_1 + x_2 = 0 \quad x_1 + x_3 = 0 \quad x_1 + x_4 = 0 \quad x_1 + x_5 = 0 \pmod{2},$$

i.e., $u_1 = x_1 = x_2 = x_3 = x_4 = x_5$. Hence the two codewords are $u_1 = 0 \rightarrow \mathbf{x}_1 = (00000)$ and $u_1 = 1 \rightarrow \mathbf{x}_2 = (11111)$. This is nothing but a repetition code with $n = 5$, with a transmission rate $R = 1/5$.

²Recall that in arithmetic modulo 2, $x_j = -x_j$.

Example 4: A $[4, 3]$ linear code with a single check. Consider the check matrix with $k = 3$, $n = 4$ given by

$$\mathbb{H} = [1 \ 1 \ 1 \mid 1]$$

The check equations imply that $x_4 = x_1 + x_2 + x_3 \pmod{2}$. The transmission rate is $R = 3/4$. The $2^3 = 8$ codewords are all strings with an even number of 1s: 0000, 1001, 0101, 1100, 0011, 1010, 1100, 1111.

General procedure to generate the codewords. We now show that there is a straightforward procedure to generate the 2^k codewords of an $[n, k]$ code. Let us assume that the check matrix is in the standard form $\mathbb{H} = [\mathbb{A} \mid \mathbb{I}_{n-k}]$. We know that the first k components of \mathbf{x} are simply the messages \mathbf{u} . The $(n - k)$ check equations read therefore:

$$\mathbf{0} = [\mathbb{A} \mid \mathbb{I}_{n-k}] \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = \mathbb{A} \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} x_{k+1} \\ \vdots \\ x_n \end{pmatrix} \implies \begin{pmatrix} x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = -\mathbb{A} \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix}. \quad (\text{E.8})$$

This shows that we can generate the codewords as follows.

i

Generator matrix \mathbb{G} . The codewords \mathbf{x} can be generated from the messages \mathbf{u} via the matrix equation:

$$\mathbf{x} = \mathbb{G} \mathbf{u} = \begin{bmatrix} \mathbb{I}_k \\ -\mathbb{A} \end{bmatrix} \mathbf{u} \quad (\text{E.9})$$

The matrix \mathbb{G} is an $(n \times k)$ matrix constructed from \mathbb{H} — more properly, from \mathbb{A} — and is known as the *generator matrix* of the code. The equation $\mathbf{x} = \mathbb{G} \mathbf{u}$ immediately implies that the codewords \mathbf{x} are all possible linear combinations with the coefficients given by $\mathbf{u} = (u_1, \dots, u_k)^T$ of the (linearly independent) k columns of \mathbb{G} . In particular, the k columns of \mathbb{G} are codewords. Notice that for binary codes $-\mathbb{A} = \mathbb{A}$. Since $\mathbb{H}\mathbf{x} = \mathbf{0}$, it is simple to show that:

$$\mathbb{H}\mathbb{G} = [\mathbf{0}]_{(n-k) \times k}. \quad (\text{E.10})$$

The generator matrix of the $[6, 3]$ code in our Example 1 is given by:

$$\mathbb{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (\text{E.11})$$

Notice that the 3 columns of \mathbb{G} are the codewords associated to, respectively, $\mathbf{u}_1 = (1, 0, 0)^T$, $\mathbf{u}_2 = (0, 1, 0)^T$ and $\mathbf{u}_3 = (0, 0, 1)^T$, see Eq. (E.3).

Exercise E.1. Show that by taking all possible messages \mathbf{u} , you obtain from the previous \mathbb{G} all codewords in Eq. (E.3).

Exercise E.2. Construct the generator matrix \mathbb{G} for the repetition code with $k = 1$ and $n = 3$, and $k = 1$ and $n = 5$ (Examples 2 and 3, above).

Equivalent, non-standard forms of \mathbb{H} and \mathbb{G} . Interestingly, any maximal set of k linearly independent codewords can be used as columns of the generator matrix, although this in general does not lead to the standard form of \mathbb{G} described above. Symmetrically, if \mathbf{h} denotes a parity check vector such that $\mathbf{h} \cdot \mathbf{x} = 0$ (i.e., one of the rows of the matrix \mathbb{H}), then you can argue that any maximal set of $n - k$ parity check vectors can be used to write the rows of a totally equivalent matrix \mathbb{H} . See Sec. E.1.3 for an illustration of this for the case of the Hamming code.

Linearity. If \mathbf{x}_1 and \mathbf{x}_2 denote codewords, then $\mathbf{x}_1 + \mathbf{x}_2$ is a codeword, since:

$$\mathbb{H}(\mathbf{x}_1 + \mathbf{x}_2) = \mathbb{H}\mathbf{x}_1 + \mathbb{H}\mathbf{x}_2 = \mathbf{0} .$$

The construction we gave for binary codes, actually holds also for codes defined over other finite fields, for instance $F = \{0, 1, 2\}$, which gives the ternary code. In that case, if c is an element of the field, $c\mathbf{x}$ is also a codeword. For the ternary codes, for instance, $2\mathbf{x} = -\mathbf{x}$ is also a codeword.

The codeword space \mathcal{C} . The space of codewords \mathbf{x} , or simply the *code*, is therefore an additive group, and a vector space over the field, which we will denote by \mathcal{C} .

E.1.1. Errors induced by the communication channel.

Suppose that, because of channel noise, the transmitted codeword \mathbf{x} is received as $\mathbf{y} = (y_1, \dots, y_n)^T$, with $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{e} = (e_1, \dots, e_n)^T$ is the **error vector**. The receiver must decide, based on \mathbf{y} , which codeword \mathbf{x} was actually transmitted, i.e., which error \mathbf{e} was introduced by the channel. The strategy is to choose the **most likely** error vector \mathbf{e} . To describe that, we need two important definitions.

Hamming distance and weight. The (Hamming) **distance** $\text{dist}(\mathbf{x}, \mathbf{y})$ between two vectors \mathbf{x} and \mathbf{y} , is the number of places where they differ. For instance, $\text{dist}(10111, 00101) = 2$. This definition also works also for non-binary vectors, e.g., $\text{dist}(0122, 1220) = 3$. ^a The (Hamming) **weight** $\text{wt}(\mathbf{x})$ of a vector \mathbf{x} is the number of its non-zero components x_j . For instance, $\text{wt}(10111) = 4$ and $\text{wt}(0122102) = 5$. Obviously, distance and weight are related:

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \text{wt}(\mathbf{y} - \mathbf{x}) , \tag{E.12}$$

because both express the number of places where \mathbf{x} and \mathbf{y} differ.

^aNotice, however, that for a field with elements $\{0, 1, 2, 3, 4\}$, $\text{dist}(0144, 0142) = 1$.

The decoding problem. Let us consider again the binary case. Assume that errors can occur independently on the different bit, and that $p < \frac{1}{2}$ is the probability that an error occurs. If we have $n = 5$, for instance, $\text{Prob}(\mathbf{e} = 00000) = (1 - p)^5$, $\text{Prob}(\mathbf{e} = 01000) = p(1 - p)^4$, $\text{Prob}(\mathbf{e} = 10100) = p^2(1 - p)^3$, etc. In general, if \mathbf{v} is a vector of weight m , then $\text{Prob}(\mathbf{e} = \mathbf{v}) = p^m(1 - p)^{n-m}$. Obviously, the probabilities of errors are ordered as:

$$(1 - p)^n > p(1 - p)^{n-1} > p^2(1 - p)^{n-2} > \dots ,$$

which implies that errors with least weight have larger probability. The obvious strategy for decoding, known as **nearest-neighbor decoding**, is then to pick up \mathbf{e} which has *least weight*:

$$\text{Given } \mathbf{y} \implies \text{Find } \mathbf{x} \text{ such that: } \min_{\mathbf{x}}(\text{dist}(\mathbf{x}, \mathbf{y})) = \min_{\mathbf{x}}(\text{wt}(\mathbf{y} - \mathbf{x})) = \min_{\mathbf{x}}(\text{wt}(\mathbf{e})). \quad (\text{E.13})$$

A brute-force approach, however, is impossible for large k . Given a received \mathbf{y} , I should calculate the possible 2^k error vectors \mathbf{e} by calculating $\mathbf{e} = \mathbf{y} - \mathbf{x}$ with respect to all possible codewords \mathbf{x} , to pick up the error with least weight.



Fast decoding is needed. One of the goals of coding theory is to find decoding methods which are faster than the brute-force approach of checking which of the 2^k possible codewords \mathbf{x} is the nearest-neighbor of the received \mathbf{y} .

Minimum distance of a code. There is a third important parameter of a code, beyond its length n and its dimension k : the minimum distance d between its codewords.



Distance d of a code \mathcal{C} . The distance d of a code is the minimum distance between its codewords:

$$d = \min(\text{dist}(\mathbf{x}_1, \mathbf{x}_2)) \quad \text{with } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}, \mathbf{x}_1 \neq \mathbf{x}_2. \quad (\text{E.14})$$

To find the minimum distance, it is not necessary to compare all pairs of different codewords. Indeed, since $\mathbf{x}_1 - \mathbf{x}_2 = \mathbf{w}$ is also a codeword, we can calculate d as the minimum weight of any non-zero codeword:

$$d = \min_{\mathbf{w} \in \mathcal{C}}(\text{wt}(\mathbf{w})) \quad \text{with } \mathbf{w} \neq \mathbf{0}. \quad (\text{E.15})$$

From now on, we denote by $[n, k, d]$ a code of length n , dimension k , distance d .

The $[6, 3]$ code in our Example 1 has $d = 3$. It is a $[6, 3, 3]$ linear code.

How many errors can be corrected? Here comes an important result.



Errors that can be corrected/detected. A code with minimum distance d can correct $\lfloor \frac{d-1}{2} \rfloor$ errors. If d is even, it can correct $\frac{d}{2} - 1$ errors, but it can detect $\frac{d}{2}$ errors.

The proof of this result is not difficult. Suppose first that d is odd. Let $r = (d-1)/2$ be the radius of a sphere around each codeword. ^a Spheres around different codewords **do not overlap**, because the minimum distance between codewords is $d = 2r + 1$. Then, if \mathbf{x} is transmitted and $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is received, with $\text{wt}(\mathbf{e}) \leq r$, then \mathbf{y} belongs to the sphere of \mathbf{x} and the nearest-neighbor decoding will correct $\mathbf{y} \rightarrow \mathbf{x}$. Now suppose that d is even, so that $\lfloor \frac{d-1}{2} \rfloor = \frac{d-2}{2} = \frac{d}{2} - 1$. Take $r = \frac{d}{2} - 1$ and draw non-overlapping (since $d = 2r + 2$) spheres around each codeword: once again, up to $r = \frac{d}{2} - 1$ errors are corrected by the nearest-neighbor decoding. What happens if d is even and there are $\frac{d}{2}$ error is peculiar: the received message \mathbf{y} is precisely **in between two neighboring spheres**, and you will be unable to say which \mathbf{x} I should associate to \mathbf{y} . Nevertheless, we are sure that there was an error: hence, we say we can **detect** if $\frac{d}{2}$ errors occur. If more than $\frac{d}{2}$ errors occur, the received vector \mathbf{y} might be closer to some other codeword than the correct \mathbf{x} , and the decoder might be fooled.

^aA sphere of radius r around \mathbf{x} consists of all $\mathbf{w} \in \mathcal{C}$ such that $\text{dist}(\mathbf{w}, \mathbf{x}) \leq r$.

◆ **Error detection vs error correction.** Sometimes, we might adopt the extreme approach of simply **detecting if an error occurred**, which can be done with simple techniques — based on the concept of *syndrome*, which we are going to explain —, and, if so, ask for re-transmission of the message, rather than embarking in a more costly error-correction procedure.

The $[6, 3]$ code in our Example 1 has $d = 3$. It is a $[6, 3, 3]$ linear code, which can correct 1 error.

E.1.2. More about decoding: cosets and syndromes.

❶ **The coset.** Let \mathcal{C} be a linear code $[n, k]$. For any vector $\mathbf{a} \in \mathbb{B}^n$, the set

$$\mathbf{a} + \mathcal{C} = \{\mathbf{a} + \mathbf{x} : \mathbf{x} \in \mathcal{C}\} \tag{E.16}$$

is the *coset* (or translate) of \mathcal{C} . It contains 2^k elements. Every vector $\mathbf{a} \in \mathbb{B}^n$ is in some coset, for instance $\mathbf{a} + \mathcal{C}$.^a \mathbf{a} and \mathbf{b} are in the same coset if $\mathbf{a} - \mathbf{b} \in \mathcal{C}$. As for groups, it is simple to prove that **cosets are disjoint**, and all together — there are $n_c = 2^{(n-k)}$ different cosets — exhaust all possible vectors in \mathbb{B}^n :

$$\mathbb{B}^n = \mathcal{C} \cup (\mathbf{a}_1 + \mathcal{C}) \cup \dots \cup (\mathbf{a}_{n_c-1} + \mathcal{C}) . \tag{E.17}$$

Everything we said applies also to finite fields $F = \{1, 2, \dots, q\}$ with simple modifications. Each coset has q^k elements and there are $n_c = q^{n-k}$ different cosets.

^aIf $\mathbf{a} \in \mathcal{C}$, the coset coincides with \mathcal{C} .

The standard array. Let us consider, for simplicity, binary codes. Let $c = 2^k$ be the number of codewords. We can in principle lists all the codewords and cosets as follows:

$\mathbf{x}^{(1)} = \mathbf{0}$	$\mathbf{x}^{(2)}$	\dots	$\mathbf{x}^{(c)} \leftarrow \mathcal{C}$	(E.18)
$\mathbf{a}_1 + \mathbf{x}^{(1)}$	$\mathbf{a}_1 + \mathbf{x}^{(2)}$	\dots	$\mathbf{a}_1 + \mathbf{x}^{(c)} \leftarrow \mathbf{a}_1 + \mathcal{C}$	
\vdots	\vdots	\dots	\vdots	
$\mathbf{a}_{n_c-1} + \mathbf{x}^{(1)}$	$\mathbf{a}_{n_c-1} + \mathbf{x}^{(2)}$	\dots	$\mathbf{a}_{n_c-1} + \mathbf{x}^{(c)} \leftarrow \mathbf{a}_{n_c-1} + \mathcal{C}$	

❶ **The coset leaders \mathbf{a}_j .** The elements $\mathbf{a}_1 \dots \mathbf{a}_{n_c-1}$ appearing in the standard array shown above can be chosen in an arbitrary manner among the 2^k elements of the coset. Nevertheless, it is convenient to eliminate this arbitrariness, by choosing \mathbf{a}_j as the vector with **minimum weight** in the coset $\mathbf{a}_j + \mathcal{C}$. This choice defines \mathbf{a}_j as the *coset leader* of $\mathbf{a}_j + \mathcal{C}$.

Suppose you now receive \mathbf{y} while \mathbf{x} was transmitted. \mathbf{y} must belong to \mathcal{C} or to some coset in the standard array, say $\mathbf{y} = \mathbf{a}_j + \mathbf{w}$ with $\mathbf{w} \in \mathcal{C}$, and $\mathbf{a}_0 = \mathbf{0}$ in case $\mathbf{y} \in \mathcal{C}$. The error is $\mathbf{e} = \mathbf{y} - \mathbf{x} = \mathbf{a}_j + \mathbf{w} - \mathbf{x} \in (\mathbf{a}_j + \mathcal{C})$, i.e., the possible error vectors are exactly the vectors in the coset $(\mathbf{a}_j + \mathcal{C})$ containing \mathbf{y} . Then the decoder strategy is to take $\hat{\mathbf{e}}$ as the minimum weight vector in the coset $(\mathbf{a}_j + \mathcal{C})$ containing \mathbf{y} , which is, by definition, the coset leader \mathbf{a}_j , and decode $\mathbf{y} \rightarrow \hat{\mathbf{x}} = \mathbf{y} - \hat{\mathbf{e}}$. How do we do that in practice?



The syndrome. Suppose we transmit \mathbf{x} and $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is received. Then by checking

$$\mathbf{S} = \mathbb{H}\mathbf{y} = \mathbb{H}(\mathbf{x} + \mathbf{e}) = \mathbb{H}\mathbf{e} , \tag{E.19}$$

we calculate the **syndrome** vector \mathbf{S} (of dimension $n - k$). If $\mathbf{S} \neq \mathbf{0}$ then there was an error: ^a this already gives us an effective way of doing **error detection**. For binary codes, if \mathbf{e} has a 1 at position, say, j_1, j_2, \dots , then the relation $\mathbf{S} = \mathbb{H}\mathbf{e}$ tells us that

$$\mathbf{S} = \mathbf{H}_{j_1} + \mathbf{H}_{j_2} + \dots \tag{E.20}$$

where \mathbf{H}_j denotes the j -th column vector in \mathbb{H} . To determine the error \mathbf{e} given the syndrome \mathbf{S} , we should be able to “invert” the relationship $\mathbf{S} = \mathbb{H}\mathbf{e}$, finding the coset leader \mathbf{e} of the coset to which \mathbf{y} belongs. A brute force approach to determining \mathbf{e} is, once again, possible only for small n .

^aThe converse is not guaranteed.

Let us illustrate this procedure with a simple example. Consider the $[6, 3, 3]$ code with parity check matrix (in standard form):

$$\mathbb{H} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right] = \left[\mathbb{A} | \mathbb{I}_3 \right] . \tag{E.21}$$

The associated generator matrix is:

$$\mathbb{G} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \right] . \tag{E.22}$$

Here is the standard array you would set up, ³ with the syndrome vectors shown to the right:

$\mathbf{u} \rightarrow$	000	001	010	011	100	101	110	111	\mathbf{S}
$\mathcal{C} \rightarrow$	000 000	001 110	010 101	011 011	100 011	101 101	110 110	111 000	$(0, 0, 0)^T$
$\mathbf{a}_1 + \mathcal{C} \rightarrow$	000 001	001 111	010 100	011 010	100 010	101 100	110 111	111 001	$(0, 0, 1)^T$
$\mathbf{a}_2 + \mathcal{C} \rightarrow$	000 010	001 100	010 111	011 001	100 001	101 111	110 100	111 010	$(0, 1, 0)^T$
$\mathbf{a}_3 + \mathcal{C} \rightarrow$	000 100	001 010	010 001	011 111	100 111	101 001	110 010	111 100	$(1, 0, 0)^T$
$\mathbf{a}_4 + \mathcal{C} \rightarrow$	001 000	000 110	011 101	010 011	101 011	100 101	111 110	110 000	$(1, 1, 0)^T$
$\mathbf{a}_5 + \mathcal{C} \rightarrow$	010 000	011 110	000 101	001 011	110 011	111 101	100 110	101 000	$(1, 0, 1)^T$
$\mathbf{a}_6 + \mathcal{C} \rightarrow$	100 000	101 110	110 101	111 011	000 011	001 101	010 110	011 000	$(0, 1, 1)^T$
$\mathbf{a}_7 + \mathcal{C} \rightarrow$	100 100	101 010	110 001	111 111	000 111	001 001	010 010	011 100	$(1, 1, 1)^T$

All the $2^6 = 64$ vectors of B^6 are listed, with $2^4 = 8$ elements in each of the 8 cosets: the code, a trivial coset with $\mathbf{a}_0 = (0, 0, 0, 0, 0, 0)^T$, and 7 non-trivial cosets, 6 with weight 1, $\mathbf{a}_1, \dots, \mathbf{a}_6$, and 1, \mathbf{a}_7 , with weight 2. In red I show a possible received message $\mathbf{y} = (0, 1, 1, 0, 0, 1)^T$, for which you immediately calculate a non-zero syndrome $\mathbf{S} = \mathbb{H}\mathbf{y} = (0, 1, 0)^T$, which coincides with the 5th column of \mathbb{H} . Now, nothing forbids in principle that \mathbf{y} is a corruption, for instance, of $\mathbf{x}^{(3)} = (0, 1, 0, 1, 0, 1)^T$, with $\mathbf{e} = (0, 0, 1, 1, 0, 0)^T$, i.e., a flip of bits 3 and 4. But the *least* number of bit flips that might have occurred — hence the most likely situation when the bit flip error p is small — it that the error is $\mathbf{e} = \mathbf{a}_2 = (0, 0, 0, 0, 1, 0)^T$, which is indeed a coset leader of the coset $\mathbf{a}_2 + \mathcal{C}$ to which \mathbf{y} belongs, and therefore that the original codeword is $\mathbf{x}^{(4)} = (0, 1, 1, 0, 1, 1)^T = \mathbf{y} - \mathbf{a}_2$.

³For simplicity of notation, we write, say 001110 instead of $(0, 0, 1, 1, 1, 0)^T$.

The last row of the standard array, $\mathbf{a}_7 + \mathcal{C}$, deserves a comment. Here the coset leader has weight $w = \text{wt}(\mathbf{a}_7) = 2$, but you immediately notice that there are two other possible coset leaders I might have used, both in blue and with weight 2. Now suppose you receive $\mathbf{y} = (1, 1, 1, 1, 1, 1)^T$: you check the syndrome, and you calculate $\mathbf{S} = \mathbb{H}\mathbf{y} = (1, 1, 1)^T$, hence you would associate with the error with coset leader $\mathbf{a}_7 = (1, 0, 0, 1, 0, 0)^T$, and predict that \mathbf{y} is a corruption of $\mathbf{x}^{(4)} = (0, 1, 1, 0, 1, 1)^T$. But in principle \mathbf{y} might be a corruption, still with two bit flips, of $\mathbf{x}^{(7)} = (1, 1, 0, 1, 1, 0)^T$ with error $\mathbf{e} = (0, 0, 1, 0, 0, 1)^T$, one of the two other possible coset leaders. Hence, you detect that there was an error, but even assuming the maximum likelihood of only two bit flip errors, you would not be able to uniquely reconstruct the original codeword. Needless to say, even less likely, \mathbf{y} might be a corruption of $\mathbf{x}^{(8)} = (1, 1, 1, 0, 0, 0)^T$ with error $\mathbf{e} = (0, 0, 0, 1, 1, 1)^T$, which is *not a coset leader*.

Exercise E.3. Construct yourself from scratch a standard array for the $[6, 3, 3]$ code of our Example 1, without looking at the result given above. Use it to decode vectors $\mathbf{y} = (1, 1, 0, 1, 1, 1)^T$ and $\mathbf{y} = (0, 0, 0, 1, 0, 1,)^T$.



Cosets and syndromes. There is a one-to-one correspondence between cosets and syndromes. Indeed, two vectors \mathbf{y}_1 and \mathbf{y}_2 are in the same coset of \mathcal{C} if and only if they have the same syndrome. To see this, notice that two vectors in the same coset are such that $\mathbf{y}_1 - \mathbf{y}_2 \in \mathcal{C}$, hence $\mathbb{H}(\mathbf{y}_1 - \mathbf{y}_2) = \mathbf{0}$, which is equivalent to saying that $\mathbf{S}_1 = \mathbb{H}\mathbf{y}_1 = \mathbb{H}\mathbf{y}_2 = \mathbf{S}_2$.

E.1.3. The binary Hamming code

The binary Hamming code is an important family of single-error-correction codes which are very **easy to decode**. They were introduced by Richard W. Hamming in 1950 for punched card readers.

Suppose we want to construct a code that corrects single errors, i.e., $t = 1$. Let us denote by \mathbf{H}_j the j -th column of \mathbb{H} . The columns of \mathbb{H} should all be different from $\mathbf{0}$, otherwise an error occurring at position j , where $\mathbf{H}_j = \mathbf{0}$ would never contribute to the syndrome, and would go undetected. Also, no two columns should coincide, because if $\mathbf{H}_{j_1} = \mathbf{H}_{j_2}$ with $j_1 \neq j_2$, then errors in positions j_1 and j_2 would be indistinguishable. Suppose that \mathbb{H} has $r = n - k \geq 2$ rows, then all possible $2^r - 1$ distinct and non-vanishing binary strings vectors of r bits are candidate column vectors for \mathbb{H} . If we use them all, we have a check matrix which is $r \times (2^r - 1)$, i.e., $n = 2^r - 1$ is the length of the code, and $k = n - r = 2^r - 1 - r$ its dimension.

As an example, suppose we take $r = 3$, hence $n = 2^3 - 1 = 7$ and $k = n - r = 4$, and we write \mathbb{H} as the 3×7 matrix with all 3-bit non-zero binary strings as columns:

$$\mathbb{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \tag{E.23}$$

where you should observe that we have written the columns as the binary string in increasing order, with \mathbf{H}_j being the binary string for integer $j = 1, \dots, 7$ (least significant bits at the bottom).



Standard form. Observe that the check matrix so written is not in standard form, but it is easy to permute the columns — amounting to permuting the labels of the bits of the codewords \mathbf{x} — to put it in standard form:

$$\mathbb{H}^{\text{standard}} = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right] = [\mathbb{A}|\mathbb{I}_3]. \quad (\text{E.24})$$

The permutation, as already stressed, does not change the code, but simply the labelling of the variables.

The generator \mathbb{G} is easy to write from the standard form of Eq. (E.9):

$$\mathbb{G}^{\text{standard}} = \left[\begin{array}{c} \mathbb{I}_4 \\ -\mathbb{A} \end{array} \right] = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array} \right]. \quad (\text{E.25})$$

As the columns of \mathbb{G} are codewords (as well as their linear combinations) you immediately see that there are codewords with weight 3, hence $d = 3$. Alternatively, from the original form, the codeword $\mathbf{x} = (1, 1, 1, 0, 0, 0, 0)^T$ satisfies $\mathbb{H}\mathbf{x} = 0$.

This is not the only meaningful permutation of labels we can think of. With a further permutation of columns $(1234567) \rightarrow (4523167)$, hence of labels of the codeword bits, we could put the matrix in a cyclic form, where the 3 rows are obtained one from the other with a **cyclic end-around shift**:

$$\mathbb{H}^{\text{cyclic}} = \left[\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array} \right], \quad (\text{E.26})$$

which leads to the fact that a cyclic shift of any codeword is again a codeword. Clearly, the corresponding generator matrix $\mathbb{G}^{\text{cyclic}}$ is obtained from $\mathbb{G}^{\text{standard}}$ in Eq. (E.25) by applying the same permutation to the rows.



Binary Hamming code with general r . For general r , we can think, similarly, of a Hamming code with $n = 2^r - 1$, $k = n - r$, i.e., a code $[n = 2^r - 1, k = 2^r - 1 - r, d = 3]$. The next case has $r = 4$, corresponding to a $[15, 11, 3]$ code.

Why is decoding simple for a Hamming code. A general binary Hamming code has a (non-standard) matrix \mathbb{H} whose j -th column is the r -bit binary representation of the number $j = 1, \dots, n = 2^r - 1$. If a *single* error occurs in bit j , $\mathbf{e}_j = (00 \dots 010 \dots 0)$ (with a single 1 in position j), then the syndrome measured will be:

$$\mathbf{S} = \mathbb{H}\mathbf{y} = \mathbb{H}\mathbf{e}_j = \mathbf{H}_j. \quad (\text{E.27})$$



Decoding from the syndrome. By transforming the binary string syndrome vector \mathbf{S} into an integer, I would immediately read off the index j where the single error was made.

Exercise E.4. Construct a standard array for the $[7, 4, 3]$ Hamming code, calculating all cosets and associated syndromes.

E.1.4. The probability of error

If the decoding is done using the standard array — which, recall, is the most likely possibility — the probability of error is simply given by:

$$P_{err} = \text{Prob}(\mathbf{e} \neq \text{coset leader}) . \quad (\text{E.28})$$

Let us examine more closely the coset leader, which, recall, are chosen as minimum weight representatives of the coset. There is a trivial coset leader $\mathbf{a}_0 = \mathbf{0}$, the coset leader of \mathcal{C} , whose weight is $\text{wt}(\mathbf{a}_0) = 0$. The remaining $n_c - 1 = 2^{n-k} - 1$ coset leaders (we are considering here binary codes for simplicity) have weights ≥ 1 and, at most, n . Let us call α_w the number of coset leaders with weight equal to w , where $w = 1 \cdots n$. Since the probability of a coset leader with weight w is obviously $\text{Prob}(\mathbf{a}_j | \text{wt}(\mathbf{a}_j) = w) = p^w (1-p)^{n-w}$, then, you predict that:

$$P_{err} = \text{Prob}(\mathbf{e} \neq \text{coset leader}) = 1 - \sum_{w=0}^n \alpha_w p^w (1-p)^{n-w} . \quad (\text{E.29})$$

For instance, for the $[6, 4, 3]$ parity check code analysed previously, we have a coset \mathcal{C} (the code) with weight 0, 6 cosets with weight 1, $\mathbf{a}_1 + \mathcal{C} \cdots \mathbf{a}_6 + \mathcal{C}$, and a coset with weight 2, $\mathbf{a}_7 + \mathcal{C}$. Hence $\alpha_0 = 1$, $\alpha_1 = 6$, $\alpha_2 = 1$. Hence, the error probability would be:

$$P_{err} = 1 - \sum_{w=0}^n \alpha_w p^w (1-p)^{n-w} = 1 - (1-p)^6 - 6p(1-p)^5 - p^2(1-p)^4 . \quad (\text{E.30})$$

Numerically, if $p = 0.01$, then $P_{err} \approx 0.00136$.

Suppose that a code can correct t errors, i.e., the distance between the codewords is $d = 2t + 1$ or (if even) $d = 2t + 2$. Then every codeword with a weight $w \leq t$ is a coset leader. Their number is easy to get using simple combinatorics:

$$\alpha_w = \binom{n}{w} \quad w = 0, \dots, t . \quad (\text{E.31})$$

Unfortunately, the α_w for other coset leader with weights $w > t$ are not simple to calculate. With this, we could write an upper bound for the error probability as:

$$P_{err} = 1 - \sum_{w=0}^n \alpha_w p^w (1-p)^{n-w} \leq 1 - \sum_{w=0}^t \binom{n}{w} p^w (1-p)^{n-w} . \quad (\text{E.32})$$

Exercise E.5. Evaluate numerically the probability of error P_{err} for the $[6, 4, 3]$ code previously discussed, for $p = 0.01$.

Perfect codes. If $\alpha_w = 0$ for all $w > t$, then the previous bound is exact, and the code is called *exact*. This means that the code can correct all errors of weight $w \leq t$, and no errors with weight $w > t$. Equivalently, the spheres of radius t around each codeword are *disjoint* and together contain all vectors in B^n .

Exercise E.6. Given the standard array for the $[7, 4, 3]$ Hamming code, find out all the α_w for the code. Is it a perfect code? Calculate the probability of error P_{err} . Evaluate it numerically for $p = 0.01$.

The Hamming codes are perfect codes. There is a much simpler way to prove that Hamming codes are perfect codes without writing explicitly the standard array, a very boring exercise in general. Let us argue as follows. Since the Hamming codes can correct single errors, $t = 1$, sphere of radius 1 from each codeword must be disjoint. There are 2^k spheres, and $n + 1 = 2^r$ vectors in each sphere, hence a total of $2^{k+r} = 2^{2^r-1} = 2^n$ vectors, i.e., the disjoint spheres exhaust all vectors in B^n , hence the code is perfect.

i

Sphere packing or Hamming bound. For an $[n, k, d]$ binary code, with $d = 2t + 1$ or $2t + 2$, the following inequality holds:

$$2^k \left(1 + \binom{n}{1} + \dots + \binom{n}{t} \right) \leq 2^n . \quad (\text{E.33})$$

The proof is quite simple. Since errors up to weight t can be corrected, the sphere of radius $w = 0 \dots t$ around each codewords are all disjoint. The number of elements in a sphere of radius w around a codeword is evidently $\binom{n}{w}$. The inequality then follows.

The symbol probability error. So far we were concerned with the probability of error of the *entire codeword*. You might be willing to estimate what is the (average) probability of error of each single bit in a codeword. Let E_j be the event where the j -th bit of the codeword is wrong, and E the event where the codeword is wrong. Then, evidently:

$$E = E_1 \cup E_2 \cup \dots \cup E_k .$$

The events E_j are neither independent nor disjoint nor equiprobable. Nevertheless:

$$P_{err} = \text{Prob}(E) \leq \sum_{j=1}^k \text{Prob}(E_j) .$$

The symbol probability error is defined as the *arithmetic mean* of the various $\text{Prob}(E_j)$.

i

The symbol probability error.

$$P_{symp} \stackrel{\text{def}}{=} \frac{1}{k} \sum_{j=1}^k \text{Prob}(E_j) \geq \frac{1}{k} P_{err} . \quad (\text{E.34})$$

Evidently, there is also an inequality $P_{symp} \leq P_{err}$, because the probability that the codeword is wrong is certainly larger than the probability that any single bit is wrong. All in all, we can write:

$$\frac{1}{k} P_{err} \leq P_{symp} \leq P_{err} . \quad (\text{E.35})$$

It is generally difficult to calculate P_{symp} for a code, see Ref. [50][Chap. 1.5] for details, but these inequalities help estimating it.

E.1.5. Shannon’s theorem: the existence of good codes

We saw examples of codes that reduce the error probability from the bare channel value p . For instance, for the $[6, 4, 3]$ parity check code one can show that $P_{err} = 0.00136$ and $P_{symb} = 0.00072$ when $p = 0.01$. The code transmits $n = 6$ bits instead of $k = 4$, hence it has a transmission rate of $R = k/n = 2/3 = 0.666$. As a second example, the $[7, 4, 3]$ Hamming code can be shown to have $P_{err} = 0.002$ and $P_{symb} = 0.001$ when $p = 0.01$, with a transmission rate of $R = k/n = 4/7 = 0.571$.

In general, by increasing n , for given k , one can decrease the error probability further, at the cost, however, of decreasing also the transmission rate $R = k/n$. We would like to know how small we can make P_{err} — and P_{symb} — for a given rate R with an $[n, k, d]$ code. A remarkable theorem, due to Shannon, provides the answer. But first we need to define the *capacity* of the binary channel.

1 **The capacity of a binary channel.** If p is the probability that a single transmitted symbol is erroneously flipped, you define the capacity $C(p)$ of the channel in terms of the Shannon entropy $H(p)$ as:

$$C(p) = 1 - H(p) = 1 - p \log_2 \frac{1}{p} - (1 - p) \log_2 \frac{1}{1 - p} . \tag{E.36}$$

The capacity tends to 1 for $p \rightarrow 0^+$ (and for $p \rightarrow 1^-$, as there is a symmetry $p \leftrightarrow (1 - p)$), while $C(p = \frac{1}{2}) = 0$.

1 **Shannon’s coding theorem.** For any $\epsilon > 0$, and for any transmission rate $R < C(p)$, if n is sufficiently large, then an $[n, k]$ binary code with error probability $P_{err} < \epsilon$ exists.

Unfortunately, the proof of this theorem (which we omit) is *probabilistic*, and **does not tell how to construct such good codes**.

In practice, it is rather difficult to rely on P_{err} , and even more so on P_{symb} , but the minimum distance d of the code can be used to gauge how good the code is, recalling that $d = 2t + 1$ or $d = 2t + 2$, where t is the number of errors that can be corrected.

So, the goal in classical error correction can be reformulated as follows: **find codes with as large as possible $R < C(p)$ (for an efficient transmission rate), and with large d (to correct many errors)**. Of course, these are conflicting goals. The sphere-packing bound already tells us something about the size n of the code, since:

$$2^{n-k} \geq \left(1 + \binom{n}{1} + \dots + \binom{n}{t} \right) . \tag{E.37}$$

In general, we know that good linear codes exist, but, at present, we do not know how to find such codes. Moreover, recall that another important requisite for a good code is that **decoding is easy**, without having to resort to a time-consuming standard array.

E.1.6. Dual codes

Given two binary vectors $\mathbf{v}, \mathbf{w} \in \mathbb{B}^n$, you can define a scalar product between them by a bitwise modulo-two sum as follows:

$$\mathbf{v} \cdot \mathbf{w} \stackrel{\text{def}}{=} v_1 w_1 \oplus v_2 w_2 \oplus \dots \oplus v_n w_n . \tag{E.38}$$

For instance: $\mathbf{v} = 1101$ and $\mathbf{w} = 1110$, then $\mathbf{v} \cdot \mathbf{w} = 1 + 1 + 0 + 0 \pmod{2} = 0$. By definition, two vectors are **orthogonal** if $\mathbf{v} \cdot \mathbf{w} = 0$. You might also define the bitwise product of two vectors as:

$$\mathbf{v} * \mathbf{w} \stackrel{\text{def}}{=} (v_1 w_1, v_2 w_2, \dots, v_n w_n). \quad (\text{E.39})$$

Then evidently, $\mathbf{v} \cdot \mathbf{w} = 0$ if and only if $\text{wt}(\mathbf{v} * \mathbf{w})$ is *even*. Also notice that $\mathbf{v} \cdot \mathbf{v} = 0$ if and only if $\text{wt}(\mathbf{v})$ is even. This is quite different, therefore, from ordinary scalar product with vectors.

i

The dual code. Now consider a linear code \mathcal{C} , and all the codewords $\mathbf{x} \in \mathcal{C}$. Next consider the linear subspace of \mathbb{B}^n made by all vectors which are orthogonal to a codeword in \mathcal{C} :

$$\mathcal{C}^\perp = \{\mathbf{w} \mid \mathbf{w} \cdot \mathbf{x} = 0 \text{ for all } \mathbf{x} \in \mathcal{C}\}. \quad (\text{E.40})$$

\mathcal{C}^\perp is known as the **dual code** of \mathcal{C} .

Evidently, the elements $\mathbf{w} \in \mathcal{C}^\perp$ are nothing but the **parity checks** on \mathcal{C} . But recall that if \mathbb{H} is the parity check matrix of \mathcal{C} , the *rows* of \mathbb{H} are the parity check vectors. Hence the generator matrix of \mathcal{C}^\perp is precisely given by the *transpose* of \mathbb{H} , so that the vectors are given as columns, i.e., an $n \times (n - k)$ matrix of the form:

$$[\mathbb{G}^\perp]_{n \times (n-k)} = \mathbb{H}^\text{T} \quad (\text{E.41})$$

Moreover, recall that for the code \mathcal{C} we have, see Eq. (E.10):

$$\mathbb{H}\mathbb{G} = [0]_{(n-k) \times k} \implies \mathbb{G}^\text{T}\mathbb{H}^\text{T} = [0]_{k \times (n-k)}. \quad (\text{E.42})$$

i

Check and generator matrices of the dual code. Hence you conclude that the check and generator matrix of the dual code are simply given by:

$$\mathbb{H}^\perp = \mathbb{G}^\text{T} \quad \mathbb{G}^\perp = \mathbb{H}^\text{T}. \quad (\text{E.43})$$

The dual code \mathcal{C}^\perp is an $[n, n - k]$ code, if \mathcal{C} is an $[n, k]$ code, i.e., $k^\perp = n - k$. It is very easy to show that $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

Self-dual codes. A code \mathcal{C} is called **weakly self-dual** if $\mathcal{C} \subseteq \mathcal{C}^\perp$. A necessary condition for this to happen is that $k^\perp = n - k \geq k$, hence $2k \leq n$. The code is called **strictly self-dual** if $\mathcal{C} = \mathcal{C}^\perp$. A necessary condition for this to happen is that n is even and $k = n/2 = k^\perp$.

i

Info: We will see that the dual construction arises very naturally in the context of quantum error correction: it is at the heart of the construction of an important class of QEC codes known as **Calderbank–Shor–Steane** (CSS) codes.

Exercise E.7. Recall that \mathbb{G} is an $n \times k$ matrix. Show that a code with generator matrix \mathbb{G} is weakly self-dual if and only if $\mathbb{G}^\text{T}\mathbb{G} = [0]_{(k \times k)}$.

Exercise E.8. Let \mathcal{C} be a linear code. Show that:

$$\left\{ \begin{array}{l} \mathbf{w} \in \mathcal{C}^\perp \implies \sum_{\mathbf{x} \in \mathcal{C}} (-1)^{\mathbf{w} \cdot \mathbf{x}} = |\mathcal{C}| \\ \mathbf{w} \notin \mathcal{C}^\perp \implies \sum_{\mathbf{x} \in \mathcal{C}} (-1)^{\mathbf{w} \cdot \mathbf{x}} = 0 \end{array} \right. .$$

Exercise E.8 will be very important later, as we will use in the CSS construction. I invite you to do it.

Hint at the solution of Exercise E.8 The first part is trivial. Suppose $\mathbf{w} \notin \mathcal{C}^\perp$. Then at least an $\mathbf{x}'_1 \in \mathcal{C}$ exists for which $\mathbf{w} \cdot \mathbf{x}'_1 = 1$. But, for sure, for $\mathbf{x}_1 = \mathbf{0}$, we have $\mathbf{w} \cdot \mathbf{x}_1 = 0$. Now argue as follows. If $|\mathcal{C}| > 2$, at least another element of \mathcal{C} must exist. Two cases are possible: 1) $\exists \mathbf{x}_2 \neq \mathbf{x}_1$ such that $\mathbf{w} \cdot \mathbf{x}_2 = 0$, but then $\mathbf{x}'_2 = \mathbf{x}'_1 + \mathbf{x}_2 \neq \mathbf{x}'_1$ is such that $\mathbf{w} \cdot \mathbf{x}'_2 = 1$; 2) viceversa if $\exists \mathbf{x}'_2 \neq \mathbf{x}'_1$ such that $\mathbf{w} \cdot \mathbf{x}'_2 = 1$, then $\mathbf{x}_2 = \mathbf{x}'_1 - \mathbf{x}'_2 \neq \mathbf{x}_1$ is such that $\mathbf{w} \cdot \mathbf{x}_2 = 0$. Proceeding in this way you can show that \mathcal{C} is split into equal parts: 2^{k-1} vectors \mathbf{x}'_j such that $\mathbf{w} \cdot \mathbf{x}'_j = 1$, and 2^{k-1} vectors \mathbf{x}_j such that $\mathbf{w} \cdot \mathbf{x}_j = 0$.

E.1.7. Construction of new codes from old ones

There are several techniques to obtain new codes starting from old ones. I will not enter into details here. If interested, read Ref. [50][Chap. 1.9] for details. I simply list the various possibilities, being very sketchy.

Extending a code by adding an overall parity check. Suppose your $[n, k, d]$ code \mathcal{C} has codewords of even and odd weights. Consider a new code $\hat{\mathcal{C}}$ living in \mathbb{B}^{n+1} with an extra coordinate x_{n+1} , hence $\hat{\mathbf{x}} = (\mathbf{x}, x_{n+1})$ are the new codewords, where $x_{n+1} = 1$ if the weight of \mathbf{x} is odd, and 0 viceversa. Then:

$$x_1 + x_2 + \dots + x_n + x_{n+1} = 0 \pmod{2} .$$

Therefore, the new check matrix is of dimension $(n + 1 - k) \times (n + 1)$:

$$\hat{\mathbb{H}} = \left(\begin{array}{ccc|c} 1 & 1 & \dots & 1 \\ \hline & \mathbb{H} & & 0 \\ & & & 0 \\ & & & 0 \end{array} \right)$$

The new code is an $[n + 1, k]$ code, but, interestingly, if d was *odd* for \mathcal{C} , then $\hat{\mathcal{C}}$ has distance $d + 1$.

Puncturing a code by deleting coordinates. This is a bit the opposite of the process of extending the code. Here you simply eliminate some coordinates, reducing n and (usually) d , but keeping k the same.

Expurgating by throwing away codewords. For instance, if \mathcal{C} contains codewords of even and odd weight, you can throw away all codewords of odd weight, which you can show are exactly half of the total. The new code is $[n, k - 1, d']$, and often $d' > d$ (for instance, if d is odd).

Augmenting a code by adding new codewords. See Ref. [50].

Lengthening. See Ref. [50].

Shortening by taking cross sections. For instance, consider only the codewords that begin with $x_1 = 0$, and then delete the first coordinate, obtaining a code of length $n - 1$.

E.1.8. General properties of linear codes

I mention here some general properties of linear codes, based on linear algebra theorems.

i **Theorem on the dimension k .** If \mathbb{H} is the parity check matrix of a code of length n , then the code has dimension $k = n - r$, if and only if some r **columns** are linearly independent, while no $r + 1$ columns are.

Proof. Recall indeed that the check matrix \mathbb{H} has $r = n - k$ linearly independent rows, hence r is the *rank* of \mathbb{H} , which applies guarantees the linear independence of r columns as well. ■

i **Theorem on the distance d .** If \mathbb{H} is the parity check matrix of a code of length n , then the code has distance d if and only if every $d - 1$ **columns** of \mathbb{H} are linearly independent, and some d columns are linearly dependent. ^a

^aNotice that this simply says that $r = \text{rank}(\mathbb{H}) \geq d - 1$.

Proof. Recall that $d = \min_{\mathbf{x} \neq \mathbf{0}}(\text{wt}(\mathbf{x}))$. For such codewords we must have:

$$\mathbb{H} \mathbf{x} = \mathbf{0} ,$$

which implies that d columns of \mathbb{H} sum to $\mathbf{0}$, hence they are not linearly independent. ■

i **The Singleton bound.** If \mathcal{C} is an $[n, k, d]$ code, then:

$$n - k \geq d - 1 .$$

Proof. Recall that $r = n - k$ is the rank of \mathbb{H} , and that $r \geq d - 1$, because every set of $d - 1$ columns is for sure linearly independent, by the theorem on the distance d . ■

Other more interesting bounds can be derived from these theorems, but I refer you to Ref. [50] for details.

E.2. Quantum codes

E.2.1. Calderbank–Shor–Steane (CSS) quantum codes

Consider two classical codes, $\mathcal{C}_1 = [n, k_1]$ and $\mathcal{C} = [n, k_2]$ such that $\mathcal{C}_2 \subset \mathcal{C}_1$, which implies that $k_2 < k_1$. We can easily define a coset structure $\mathcal{C}_1/\mathcal{C}_2$. Define the cosets as follows:

$$\mathbf{x} + \mathcal{C}_2 = \{\mathbf{x} + \mathbf{y} \mid \mathbf{y} \in \mathcal{C}_2\} \quad \forall \mathbf{x} \in \mathcal{C}_1 . \quad (\text{E.44})$$

Evidently if \mathbf{x}' is such that $\mathbf{x} - \mathbf{x}' \in \mathcal{C}_2$, then $\mathbf{x} + \mathcal{C}_2 = \mathbf{x}' + \mathcal{C}_2$, which means that you can trade \mathbf{x} for any other representative in the coset. As usual, cosets are disjoint. Indeed, given two cosets $\mathbf{x} + \mathcal{C}_2$ and $\mathbf{x}' + \mathcal{C}_2$, if an element in common exists, then you can find $\mathbf{y}, \mathbf{y}' \in \mathcal{C}_2$ such that $\mathbf{x} + \mathbf{y} = \mathbf{x}' + \mathbf{y}'$, which implies that $\mathbf{x} - \mathbf{x}' = \mathbf{y}' - \mathbf{y} \in \mathcal{C}_2$, which in turn implies, see above, that $\mathbf{x} + \mathcal{C}_2 = \mathbf{x}' + \mathcal{C}_2$. Since the number of elements of \mathcal{C}_1 and \mathcal{C}_2 is $|\mathcal{C}_1| = 2^{k_1}$ and $|\mathcal{C}_2| = 2^{k_2}$, the number of cosets is evidently:

$$\text{Number of cosets} = \frac{|\mathcal{C}_1|}{|\mathcal{C}_2|} = 2^{k_1 - k_2} . \quad (\text{E.45})$$

If $|\mathbf{x}\rangle$ denotes as usual computational states in the n -Qbit Hilbert space, we can consider the following coset-superposition:

$$|\psi_{\mathbf{x} + \mathcal{C}_2}\rangle = |\mathbf{x} + \mathcal{C}_2\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} |\mathbf{x} + \mathbf{y}\rangle . \quad (\text{E.46})$$

It is easy to prove that these $2^{k_1-k_2}$ states — as many as the cosets — are normalized: $\langle \mathbf{x} + \mathcal{C}_2 | \mathbf{x} + \mathcal{C}_2 \rangle = 1$. Also, since the cosets have disjoint vectors, you can easily show that

$$\langle \mathbf{x}' + \mathcal{C}_2 | \mathbf{x} + \mathcal{C}_2 \rangle = 0 \quad \text{if} \quad \mathbf{x} + \mathcal{C}_2 \neq \mathbf{x}' + \mathcal{C}_2 .$$

Hence, this is an orthonormal basis of $2^{k_1-k_2}$ elements, which spans a subspace of the full n -Qbit Hilbert space. We denote it temporarily as:

$$\mathcal{H}_{\mathcal{C}_1/\mathcal{C}_2} = \text{span}(|\mathbf{x} + \mathcal{C}_2\rangle) .$$

In order to become a useful code, we need some more hypothesis.

1 **The CSS code of \mathcal{C}_1 over \mathcal{C}_2 .** Consider $\mathcal{C}_1 = [n, k_1]$ and $\mathcal{C}_2 = [n, k_2]$ such that $\mathcal{C}_2 \subset \mathcal{C}_1$, as before. Assume further that:

$$\mathcal{C}_1 \text{ and } \mathcal{C}_2^\perp \text{ are classical codes that can both correct up to } t \text{ errors .}$$

Then $\text{CSS}(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{H}_{\mathcal{C}_1/\mathcal{C}_2}$ is a quantum code $[n, k_1 - k_2]$ that can correct up to t bit-flip (\mathbf{X}) and phase-flip (\mathbf{Z}) errors.

Suppose an error \mathbf{e}_1 with $\text{wt}(\mathbf{e}_1) \leq t$ of the bit-flip type occurs, affecting all computational states in the same way. Then the corrupted state would be:

$$|\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{bit-flips}}\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} |\mathbf{x} + \mathbf{y} + \mathbf{e}_1\rangle .$$

If \mathbf{e}_2 , still with $\text{wt}(\mathbf{e}_2) \leq t$, denotes an error of the phase-flip type, then you would end up with a corrupted state of the form:

$$|\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{err}}\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{e}_2} |\mathbf{x} + \mathbf{y} + \mathbf{e}_1\rangle . \tag{E.47}$$

Let us see how to correct both types of errors. We start with the bit-flip errors.

Correcting bit-flip errors. We use the check-matrix of \mathcal{C}_1 to get the syndrome. More precisely, we add $n - k_1$ ancillary Qbits, all initially in $|0\rangle_{n-k_1}$ and consider applying the (function) \mathbb{H}_1 with the usual reversible computation trick. For every computational state we would have:

$$|\mathbf{x} + \mathbf{y} + \mathbf{e}_1\rangle_n \otimes |0\rangle_{n-k_1} \xrightarrow{\text{apply } \mathbb{H}_1} |\mathbf{x} + \mathbf{y} + \mathbf{e}_1\rangle_n \otimes |\mathbb{H}_1(\mathbf{x} + \mathbf{y} + \mathbf{e}_1)\rangle_{n-k_1} = |\mathbf{x} + \mathbf{y} + \mathbf{e}_1\rangle_n \otimes |\mathbb{H}_1 \mathbf{e}_1\rangle_{n-k_1} ,$$

where the last step follows because $\mathbf{x} + \mathbf{y} \in \mathcal{C}_1$, hence $\mathbb{H}_1(\mathbf{x} + \mathbf{y}) = 0$.

1 **CNOT are enough.** Quite interestingly, the transformation $|\mathbf{x}\rangle \otimes |0\rangle \rightarrow |\mathbf{x}\rangle |\mathbb{H}\mathbf{x}\rangle$ can be realized with a circuit composed only of CNOTs. Try to prove this.

After this application, the state is transformed into:

$$\frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{e}_2} |\mathbf{x} + \mathbf{y} + \mathbf{e}_1\rangle \otimes |\mathbb{H}_1 \mathbf{e}_1\rangle_{n-k_1}$$

Hence, the ancillary bits contain the error syndrome $\mathbf{S} = \mathbb{H}\mathbf{e}_1$. If \mathcal{C}_1 is a good easy-to-decode classical code — for instance a Hamming code —, you can uniquely reconstruct the error that occurs with highest probability, very likely \mathbf{e}_1 . To correct the state, you have to apply \mathbf{X} operators on all the bits where \mathbf{e}_1 has a bit 1 (at most t of them), so that, disregarding the ancillas, you revert to a state containing now only phase-flip errors:

$$|\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{phase-flips}}\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{e}_2} |\mathbf{x} + \mathbf{y}\rangle .$$

Correcting phase-flip errors.

i

The crucial idea. The key to the story is that by applying Hadamards $\mathbf{H}^{\otimes n}$ to the state, a phase-flip error is transformed into a bit-flip error, which is then corrected as explained above. The only thing to do is to reapply a final set of Hadamards $\mathbf{H}^{\otimes n}$, to get the final corrected state.

To correct phase-flip errors, we apply Hadamards to all n Qbits:

$$\mathbf{H}^{\otimes n} |\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{phase-flips}}\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{e}_2} \frac{1}{\sqrt{2^n}} \sum_{\mathbf{z}} (-1)^{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{z}} |\mathbf{z}\rangle,$$

where \mathbf{z} runs over all possible n -Qbit computational states. By setting $\mathbf{z}' = \mathbf{z} + \mathbf{e}_2$ (which still runs over all computational states, since \mathbf{e}_2 is fixed), and re-labelling $\mathbf{z}' \rightarrow \mathbf{z}$ we can finally write:

$$\mathbf{H}^{\otimes n} |\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{phase-flips}}\rangle = \frac{1}{\sqrt{2^n |\mathcal{C}_2|}} \sum_{\mathbf{z}} \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{z}} |\mathbf{z} + \mathbf{e}_2\rangle.$$

Now we recall Exercise E.8, which I report here with a slightly adapted notation:

$$\begin{cases} \mathbf{z} \in \mathcal{C}_2^\perp & \implies \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{\mathbf{z} \cdot \mathbf{y}} = |\mathcal{C}_2| \\ \mathbf{z} \notin \mathcal{C}_2^\perp & \implies \sum_{\mathbf{y} \in \mathcal{C}_2} (-1)^{\mathbf{z} \cdot \mathbf{y}} = 0 \end{cases}.$$

By using this we arrive at:

$$\mathbf{H}^{\otimes n} |\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{phase-flips}}\rangle = \frac{|\mathcal{C}_2|}{\sqrt{2^n |\mathcal{C}_2|}} \sum_{\mathbf{z} \in \mathcal{C}_2^\perp} (-1)^{\mathbf{x} \cdot \mathbf{z}} |\mathbf{z} + \mathbf{e}_2\rangle.$$

Notice how the error \mathbf{e}_2 now appears as a **bit-flip error**. Hence, we now use the same strategy applied before for the bit-flip errors, using this time the parity check matrix $\mathbb{H}_2^\perp = \mathbb{C}_2^T$. For each component of the state we use k_2 ancillas:

$$|\mathbf{z} + \mathbf{e}_2\rangle_n \otimes |0\rangle_{k_2} \xrightarrow{\text{apply } \mathbb{H}_2^\perp} |\mathbf{z} + \mathbf{e}_2\rangle_n \otimes |\mathbb{H}_2^\perp(\mathbf{z} + \mathbf{e}_2)\rangle_{k_2} = |\mathbf{z} + \mathbf{e}_2\rangle_n \otimes |\mathbb{H}_2^\perp \mathbf{e}_2\rangle_{k_2},$$

where the last step follows because $\mathbf{z} \in \mathcal{C}_2^\perp$, hence $\mathbb{H}_2^\perp \mathbf{z} = 0$. The total state then becomes:

$$\mathbf{H}^{\otimes n} |\psi_{\mathbf{x}+\mathcal{C}_2}^{\text{phase-flips}}\rangle \otimes |0\rangle_{k_2} \xrightarrow{\text{apply } \mathbb{H}_2^\perp} \frac{|\mathcal{C}_2|}{\sqrt{2^n |\mathcal{C}_2|}} \sum_{\mathbf{z} \in \mathcal{C}_2^\perp} (-1)^{\mathbf{x} \cdot \mathbf{z}} |\mathbf{z} + \mathbf{e}_2\rangle \otimes |\mathbb{H}_2^\perp \mathbf{e}_2\rangle_{k_2}.$$

Once again, we can measure a syndrome $\mathbf{S} = \mathbb{H}_2^\perp \mathbf{e}_2$, which, by maximum likelihood decoding would give us the error \mathbf{e}_2 . We apply the appropriate bit-flips (\mathbf{X}) to the state, and, disregarding the ancillas, finally arrive at the state:

$$\xrightarrow{\text{correct } \mathbf{e}_2} \frac{|\mathcal{C}_2|}{\sqrt{2^n |\mathcal{C}_2|}} \sum_{\mathbf{z} \in \mathcal{C}_2^\perp} (-1)^{\mathbf{x} \cdot \mathbf{z}} |\mathbf{z}\rangle.$$

Now we apply again $\mathbf{H}^{\otimes n}$ to the state:

$$\xrightarrow{\text{apply } \mathbf{H}^{\otimes n}} |\psi^{\text{corr}}\rangle = \frac{1}{\sqrt{2^n}} \frac{|\mathcal{C}_2|}{\sqrt{2^n |\mathcal{C}_2|}} \sum_{\mathbf{w}} \sum_{\mathbf{z} \in \mathcal{C}_2^\perp} (-1)^{(\mathbf{x}+\mathbf{w}) \cdot \mathbf{z}} |\mathbf{w}\rangle.$$

By repeating the same steps we did before, changing the variable over which we sum to $\mathbf{w}' = \mathbf{x} + \mathbf{w}$ including a further application of Exercise E.8, we finally realize that the state obtained is precisely the correct one:

$$|\psi^{\text{corr}}\rangle = \frac{1}{\sqrt{2^n}} \frac{|\mathcal{C}_2| |\mathcal{C}_2^\perp|}{\sqrt{2^n |\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} |\mathbf{x} + \mathbf{y}\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\mathbf{y} \in \mathcal{C}_2} |\mathbf{x} + \mathbf{y}\rangle \equiv |\mathbf{x} + \mathcal{C}_2\rangle,$$

where we used that $k_2^\perp = n - k_2$, hence $|\mathcal{C}_2| |\mathcal{C}_2^\perp| = 2^n$, cancelling the factor 2^n in the denominator.



Warning: You might be puzzled that we proved error correction only for \mathbf{X} and \mathbf{Z} errors acting on pure states, certainly not the most general errors. However, once we will show that the **CSS codes are stabilizer codes**, then the full theory of error correction can be unleashed.

The Steane [7, 1, 3] code

As an application of the CSS construction, let us consider \mathcal{C}_1 to be the Hamming [7, 4, 3] code. For \mathcal{C}_2 we take \mathcal{C}_1^\perp , for which $k_2 = n - k_1 = 3$, hence a [7, 3] code, for which we should first check that $\mathcal{C}_2 = \mathcal{C}_1^\perp \subset \mathcal{C}_1$. To do that, we write \mathbb{H}_1 for the Hamming code in its standard form, see Eq. (E.24), which we report here:

$$\mathbb{H}_1 = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right] = [\mathbb{A}|\mathbb{I}_3], \quad (\text{E.48})$$

with a corresponding generator matrix:

$$\mathbb{G}_1 = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array} \right].$$

Recall that $\mathcal{C}_1 = \text{Ker}(\mathbb{H}_1)$. Now consider $\mathcal{C}_2 = \mathcal{C}_1^\perp$ whose check matrix is:

$$\mathbb{H}_2 = \mathbb{G}_1^\top = \left[\begin{array}{cccccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right] \rightarrow \left[\begin{array}{ccc|cccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right],$$

where the second, standard, form is obtained by permutation of the columns. Again, $\mathcal{C}_2 = \text{Ker}(\mathbb{H}_2)$. To check that $\mathcal{C}_1^\perp \subset \mathcal{C}_1$, you need to apply the criterion of Exercise E.7, i.e., check that $(\mathbb{G}_1^\perp)^\top \mathbb{G}_1^\perp = [0]$. But recall that $\mathbb{G}_1^\perp = \mathbb{H}_1^\top$. Hence you need to check that:

$$(\mathbb{H}_1^\top)^\top \mathbb{H}_1^\top = \mathbb{H}_1 \mathbb{H}_1^\top = [0].$$

This is easily checked by using the \mathbb{H}_1 of the Hamming code. Next, we need to check that \mathcal{C}_2^\perp also corrects $t = 1$ errors, as \mathcal{C}_1 does. But for any code $(\mathcal{C}^\perp)^\perp = \mathcal{C}$, hence

$$\mathcal{C}_2^\perp = (\mathcal{C}_1^\perp)^\perp = \mathcal{C}_1,$$

which certainly corrects $t = 1$ errors, because the Hamming code \mathcal{C}_1 does so.

The requirements of the CSS construction are all satisfied. Hence $\text{CSS}(\mathcal{C}_1, \mathcal{C}_1^\perp)$ is a $[7, (k_1 - k_2) = 1, 3]$ quantum code, since $k_1 = 4$ and $k_2 = 3$. This code can correct all $t = 1$ bit-flip and phase-flip errors. It is known as the [7, 1, 3] **Steane code**.

E.2.2. The CSS codes seen as stabilizers codes

In the context of error correction for classical codes, we introduced *syndromes*, for instance $\mathbf{S} = \mathbb{H}_1 \mathbf{e}_1$, to check for bit flip errors. In the context of **stabilizers codes** we also had syndromes: the

collection of the ± 1 eigenvalues of the generators of the stabilizer group. Can we find stabilizers generators for the CSS code such that these two concepts actually coincide? Yes, indeed.

To begin, we will show that bit-flip errors associated to check matrix syndromes are simply related to stabilizers that are product of \mathbf{Z} operators in precisely the same position suggested by the check matrix \mathbb{H} .

The simplest illustration is the $[3, 1]$ repetition code which is capable of correcting only 1 bit-flip error. The check matrix is:

$$\mathbb{H} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \xrightarrow{(123) \rightarrow (321)} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \quad (\text{E.49})$$

where the second form is obtained by a permutation of the columns, which we do here simply because we want to adhere to the bit-ordering we have used in the rest of the course.⁴ The two codewords are $\mathcal{C} = \{(000), (111)\}$. If $\mathbf{e} = \mathbf{y} - \mathbf{x}$ is an error vector, you reveal it by using the syndrome vector:

$$\mathbf{S} = \mathbb{H}\mathbf{e} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{pmatrix} e_2 \\ e_1 \\ e_0 \end{pmatrix} = \begin{pmatrix} e_0 + e_1 \\ e_0 + e_2 \end{pmatrix}.$$

In the stabilizer's formalism, the two logical states are $|0_L\rangle = |000\rangle$ and $|1_L\rangle = |111\rangle$. Errors are revealed by the stabilizers:⁵

$$\mathcal{S} = \langle \hat{S}_1 = \mathbf{Z}_0\mathbf{Z}_1, \hat{S}_2 = \mathbf{Z}_0\mathbf{Z}_2 \rangle.$$

It is immediate to verify that the correspondence is:

$$\text{Eigenvalues of } \frac{1 - \hat{S}_{1,2}}{2} \xrightarrow{\text{correspond to}} (\mathbf{S})_{1,2},$$

where $(\mathbf{S})_j$ denotes the $j = 1, 2$ component of the syndrome vector \mathbf{S} . The correspondence for the states is quite obvious: for instance, the corrupted state $|100\rangle$, having stabilizer eigenvalues $(+1, -1)$, corresponds to the error $\mathbf{e} = 100$, with syndrome $(0, 1)$.

Let us consider now how to construct stabilizers for the simplest example of CSS code, the Steane $[7, 1, 3]$ code. We need $6 = n - k$ stabilizers, since $n = 7$ and $k = 1$. From now on, we will be more relaxed with our **bit ordering**, and assume the one suggested by the linear-algebra way of writing row-vectors — with bits 0 to $n - 1$ from **left to right** —, so that it is simpler to read out the stabilizers. The bit-flip errors are signaled by the Hamming $[7, 4, 3]$ code check matrix. In the cyclic form we have:

$$\mathbb{H}_1^{\text{cyclic}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \xrightarrow{\text{stabilizers}} \begin{matrix} \hat{S}_4 = \mathbf{Z}_0\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_4 \\ \hat{S}_5 = \mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_5 \\ \hat{S}_6 = \mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{Z}_6 \end{matrix}, \quad (\text{E.50})$$

where we convene that \mathbf{Z} -based stabilizers come after the \mathbf{X} -based stabilizers $\hat{S}_1, \hat{S}_2, \hat{S}_3$, which we now construct. The phase-flip errors are precisely converted into bit flip errors, with the code $\mathcal{C}_2^\perp = \mathcal{C}_1$, the very same Hamming code, upon sandwiching the state with Hadamards. But $\mathbf{HZH} = \mathbf{X}$, hence we can immediately write the full list of stabilizers.

⁴In linear algebra, you would write a binary (row) vector of 5 bits, for instance, as $(x_1, x_2, x_3, x_4, x_5) = (1, 1, 0, 1, 0)$.

If you pretend that the bits are ordered in the standard binary way, starting from the least significant (bit 0) to the right, and proceeding towards the most significant (bit 4) to the left, then you would write the same vector reversed, as $(0, 1, 0, 1, 1) = 01011 \rightarrow$ binary expression for integer 11.

⁵Notice that we are here using *hats*, \hat{S} , to indicate a stabilizer, which is an operator, to avoid confusion with the syndrome vector \mathbf{S} . Notice also that in the main text we used $\mathbf{Z}_1\mathbf{Z}_2 = \hat{S}_1\hat{S}_2$ as a second generator of the stabilizer group.

1 The stabilizers of the Steane code.

$$\mathcal{S} = \langle \mathbf{X}_0\mathbf{X}_1\mathbf{X}_2\mathbf{X}_4, \quad \mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_5, \quad \mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_6, \\ \mathbf{Z}_0\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_4, \quad \mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_5, \quad \mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_5\mathbf{Z}_6 \rangle. \tag{E.51}$$

With a similar technique, by employing the appropriate \mathbb{H}_1 and \mathbb{H}_2^\perp , you can write the $n_s = n - (k_1 - k_2)$ stabilizer generators for the general CSS($\mathcal{C}_1, \mathcal{C}_2$).

E.3. Pauli group and stabilizers reloaded

The Pauli group. Recall that the Pauli group of n -Qbits is given by

$$\mathcal{P}_n = \{w_m \hat{\sigma}_1^{(\mu_1)} \hat{\sigma}_2^{(\mu_2)} \dots \hat{\sigma}_n^{(\mu_n)}\} \quad \text{with} \quad \mu_j = 0, 1, 2, 3, \tag{E.52}$$

i.e., is made up by all possible **Pauli strings** with an overall factor $w_m = e^{im\pi/2}$ with $m = 0, 1, 2, 3$, hence with 4^{n+1} elements. Elements $g \in \mathcal{P}_n$ of the Pauli group have the following properties:

- 1) Each $g \in \mathcal{P}_n$ is **unitary**, $g^{-1} = g^\dagger$.
- 2) Each $g \in \mathcal{P}_n$ is such that $g^2 = \pm \mathbf{1}$. More precisely, $g^2 = \mathbf{1}$ if $w_m = \pm 1$, while $g^2 = -\mathbf{1}$ if $w_m = \pm i$.
Indeed, recall that Pauli matrices on different sites commute, and that $\mathbf{X}^2 = \mathbf{Y}^2 = \mathbf{Z}^2 = \mathbf{1}$.
- 3) If $g^2 = \mathbf{1}$, then g is **unitary and Hermitean**, $g^{-1} = g^\dagger = g$, while if $g^2 = -\mathbf{1}$, then g is **unitary and anti-Hermitean**, $g^{-1} = g^\dagger = -g$.
- 4) Two different elements of the Pauli group g and g' , either **commute or anti-commute**, $gg' = \pm g'g$. Observe that the overall factor w_m in front of each of them is totally irrelevant. Commutation or anti-commutation depends on the **parity of the number of exchanges** of anti-commuting Pauli matrices on each site, since $\mathbf{X}_j\mathbf{Z}_j = -\mathbf{Z}_j\mathbf{X}_j$, and so for any other pair of different Pauli matrices on the same site.

A useful representation. Recall that $\mathbf{Y}_j = i\mathbf{X}_j\mathbf{Z}_j$, so that any element of the Pauli group, within an overall factor w_m , and with the **convention** that all the \mathbf{Z} operators stay **to the right** of all the \mathbf{X} , can be uniquely represented in the following way:

$$g = w \mathbf{X}_1^{x_1} \dots \mathbf{X}_n^{x_n} \mathbf{Z}_1^{z_1} \dots \mathbf{Z}_n^{z_n} \stackrel{\text{def}}{=} w g_{(\mathbf{x}, \mathbf{z})}, \tag{E.53}$$

where $w = \pm 1, \pm i$, while $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{z} = (z_1, \dots, z_n)$ are two **Boolean row vectors** specifying if \mathbf{X}_j is present (for $x_j = 1$ and $z_j = 0$), or \mathbf{Y}_j is present (for $x_j = 1$ and $z_j = 1$), or \mathbf{Z}_j is present (for $x_j = 0$ and $z_j = 1$). Take now a second element $g' = w'g_{(\mathbf{x}', \mathbf{z}')}$. It is very simple to realize that

$$gg' = ww'g_{(\mathbf{x}, \mathbf{z})}g_{(\mathbf{x}', \mathbf{z}')} = ww'(-1)^{\mathbf{x}' \cdot \mathbf{z}}g_{(\mathbf{x} \oplus \mathbf{x}', \mathbf{z} \oplus \mathbf{z}')}, \tag{E.54}$$

where $\mathbf{x}' \cdot \mathbf{z}$ is the scalar product (mod 2) which counts how many \mathbf{X} in g' at positions specified by \mathbf{x}' have to be brought to the left of the \mathbf{Z} at positions \mathbf{z} in the expression for g , so that you can collect all the \mathbf{X} , at positions $\mathbf{x} \oplus \mathbf{x}'$, to the left of all the \mathbf{Z} , at positions $\mathbf{z} \oplus \mathbf{z}'$. Try with an example, and it will be clear. This already implies the following interesting conclusion:

$$g_{(\mathbf{x}, \mathbf{z})}^2 = (-1)^{\mathbf{x} \cdot \mathbf{z}}g_{2\mathbf{x}, 2\mathbf{z}} = (-1)^{\mathbf{x} \cdot \mathbf{z}}\mathbf{1}, \tag{E.55}$$

where we used that $g_{2\mathbf{x}, 2\mathbf{z}} = \mathbf{1}$. So, a pure (i.e., without prefactor w) $\mathbf{X} - \mathbf{Z}$ Pauli string $g_{(\mathbf{x}, \mathbf{z})}$ squares to $\mathbf{1}$ if and only if $\mathbf{x} \cdot \mathbf{z} = 0$; otherwise, it squares to $-\mathbf{1}$. Notice that $\mathbf{x} \cdot \mathbf{z}$ counts the parity of the

number of $\mathbf{XZ} = -i\mathbf{Y}$ operators in the Pauli string. Now consider $g'g$. With a similar approach, you find:

$$g'g = ww'g_{(\mathbf{x}',\mathbf{z}')}g_{(\mathbf{x},\mathbf{z})} = ww'(-1)^{\mathbf{x}\cdot\mathbf{z}'}g_{(\mathbf{x}\oplus\mathbf{x}',\mathbf{z}\oplus\mathbf{z}')} . \quad (\text{E.56})$$

i

Commutation properties of Pauli string operators. From Eqs. (E.54)-(E.56) you deduce that:

$$g_{(\mathbf{x}',\mathbf{z}')}g_{(\mathbf{x},\mathbf{z})} = (-1)^{\mathbf{x}\cdot\mathbf{z}' - \mathbf{x}'\cdot\mathbf{z}}g_{(\mathbf{x},\mathbf{z})}g_{(\mathbf{x}',\mathbf{z}')} . \quad (\text{E.57})$$

Hence $gg' = g'g$, i.e., the two operators **commute**, if and only if $\mathbf{x}\cdot\mathbf{z}' - \mathbf{x}'\cdot\mathbf{z} = 0$.

A symplectic scalar product. The expression $\mathbf{x}\cdot\mathbf{z}' - \mathbf{x}'\cdot\mathbf{z} = 0$ can be put in a very appealing matrix form, related to a symplectic product. Define the $2n \times 2n$ matrix ⁶ \mathbf{J} and, given $g = wg_{\mathbf{x},\mathbf{z}}$, the $2n$ row-vector $\mathbf{r}_g = (\mathbf{x}, \mathbf{z})$:

$$\mathbf{J} \stackrel{\text{def}}{=} \left[\begin{array}{c|c} \mathbf{0}_n & \mathbf{1}_n \\ \hline -\mathbf{1}_n & \mathbf{0}_n \end{array} \right] \quad \text{and} \quad \mathbf{r}_g \stackrel{\text{def}}{=} (\mathbf{x}, \mathbf{z}) . \quad (\text{E.58})$$

Then, if $g = wg_{\mathbf{x},\mathbf{z}}$ and $g' = w'g_{\mathbf{x}',\mathbf{z}'}$, we have that:

$$gg' = g'g \iff \mathbf{r}_g \mathbf{J} \mathbf{r}_{g'}^T = \mathbf{x}\cdot\mathbf{z}' - \mathbf{x}'\cdot\mathbf{z} = 0 . \quad (\text{E.59})$$

Stabilizers. Recall that stabilizer groups \mathcal{S} can be constructed as appropriate subgroups of the Pauli group \mathcal{P}_n : $\mathcal{S} \subset \mathcal{P}_n$. The idea is that a group of stabilizers \mathcal{S} must define a **stabilized subsector** $\mathcal{H}_{\mathcal{S}}$ of the Hilbert space \mathcal{H}_n made up by all states $|\psi\rangle \in \mathcal{H}_n$ which are **common eigenstates** with eigenvalue $+1$ of all the stabilizers:

$$|\psi\rangle \in \mathcal{H}_{\mathcal{S}} \iff \hat{S}|\psi\rangle = |\psi\rangle \quad \forall \hat{S} \in \mathcal{S} . \quad (\text{E.60})$$

A few observations:

1) Every stabilizer must **square to 1**, $\hat{S}^2 = \mathbf{1}$, otherwise it would **not** have an eigenvalue $+1$: indeed, if $\hat{S}^2 = -\mathbf{1}$, then \hat{S} would be **anti-Hermitian**. So, stabilizers must be Hermitian elements of \mathcal{P}_n that square to $\mathbf{1}$. All the stabilizers $\hat{S} \in \mathcal{S}$ have eigenvalues ± 1 .

2) Different stabilizers **must commute**. Indeed if two stabilizers \hat{S} and \hat{S}' were to *anti-commute*, then:

$$|\psi\rangle = \hat{S}\hat{S}'|\psi\rangle = -\hat{S}'\hat{S}|\psi\rangle = -|\psi\rangle \implies |\psi\rangle = 0 ,$$

which means that the stabilized space $\mathcal{H}_{\mathcal{S}}$ would be the **trivial subspace** with only the zero vector.

3) No stabilizer can be equal to $-\mathbf{1}$ (although $-\mathbf{1}$ is Hermitian and squares to $\mathbf{1}$), again for a very simple reason. If $\hat{S} = -\mathbf{1}$, then

$$\hat{S}|\psi\rangle = -\mathbf{1}|\psi\rangle = |\psi\rangle \implies |\psi\rangle = 0 ,$$

so again a trivial subspace with only the zero vector.

4) No stabilizer can be equal to $\pm i\mathbf{1}$, because it would not be Hermitian (it would square to $-\mathbf{1}$).

5) \mathcal{S} is an **Abelian subgroup** of \mathcal{P}_n . Indeed: $\mathbf{1} \in \mathcal{S}$; If $\hat{S}, \hat{S}' \in \mathcal{S}$ then $\hat{S}\hat{S}' \in \mathcal{S}$ (simple to show); Since $\hat{S}^2 = \mathbf{1}$ then $\hat{S}^{-1} = \hat{S} \in \mathcal{S}$.

⁶As usual, in the Boolean case it does not matter if you use $-\mathbf{1}_n$ or rather $\mathbf{1}_n$ in the bottom left block of \mathbf{J} .

The generators of the stabilizer group. There exist a minimal set of n_s **independent** stabilizers (more about this below) which **generate** the group \mathcal{S} by taking products of the generators. We will denote the generators as follows:

$$\mathcal{S} = \langle \hat{S}_1, \dots, \hat{S}_{s-1}, \hat{S}_s, \hat{S}_{s+1}, \dots, \hat{S}_{n_s} \rangle, \tag{E.61}$$

\mathcal{S} has 2^{n_s} elements, and any element $\hat{S} \in \mathcal{S}$ can be written as:

$$\hat{S} = \hat{S}_1^{a_1} \dots \hat{S}_s^{a_s} \dots \hat{S}_{n_s}^{a_{n_s}} \quad \text{with} \quad a_s = 0, 1. \tag{E.62}$$

The fact that stabilizers commute and square to $\mathbf{1}$ means that the order of the generators in this expression is irrelevant. Obviously, you take non-trivial generators, such that $\hat{S}_s \neq \mathbf{1}$.

Independence of the generators. The fact that generators are independent means that by eliminating one one of them, say \hat{S}_s , you generate a proper (smaller) subgroup of \mathcal{S} :

$$\langle \hat{S}_1, \dots, \hat{S}_{s-1}, \hat{S}_{s+1}, \dots, \hat{S}_{n_s} \rangle \subset \mathcal{S}.$$

Question: How to check independence?

Given a set of **putative generators** of \mathcal{S} , how can we actually **prove that they are independent**? We use the powerful tools of linear algebra with the canonical representation $g = wg_{\mathbf{x}, \mathbf{z}}$.

The check matrix of the generators. Let $\langle \hat{S}_1, \dots, \hat{S}_s, \dots, \hat{S}_{n_s} \rangle$ be the set of putative generators. To each generator $\hat{S}_s = g_s = wg_{\mathbf{x}_s, \mathbf{z}_s}$ we associate the $2n$ row-vector $\mathbf{r}_{g_s} = (\mathbf{x}_s, \mathbf{z}_s)$ (disregarding multiplicative factors) and we collect all these row-vectors into an $n_s \times (2n)$ dimensional stabilizer generators check matrix

$$\mathbb{H}^{\text{gen}} = \begin{bmatrix} \mathbf{r}_{g_1} \\ \mathbf{r}_{g_2} \\ \dots \\ \mathbf{r}_{g_{n_s}} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{x}_1 & \mathbf{z}_1 \\ \mathbf{x}_2 & \mathbf{z}_2 \\ \dots & \dots \\ \mathbf{x}_{n_s} & \mathbf{z}_{n_s} \end{array} \right]. \tag{E.63}$$



Warning: Observe that $\mathbf{r}_g = (\mathbf{x}, \mathbf{z})$ does not keep track of possible multiplicative factors in the stabilizer.

Nevertheless, the fact that

$$g_{(\mathbf{x}, \mathbf{z})} g_{(\mathbf{x}', \mathbf{z}')} = (-1)^{\mathbf{x}' \cdot \mathbf{z}} g_{(\mathbf{x} \oplus \mathbf{x}', \mathbf{z} \oplus \mathbf{z}')},$$

implies that, given two generator row-vectors, say \mathbf{r}_{g_1} and \mathbf{r}_{g_2} , then:

$$\mathbf{r}_{g_1 g_2} = \mathbf{r}_{g_1} \oplus \mathbf{r}_{g_2}. \tag{E.64}$$

This is a very important relationship for the row-vector representatives of Pauli string operators, which we will use later on.

i **Rank of check matrix and independence of generators.** We will now show that if $\text{rank}(\mathbb{H}^{\text{gen}}) < n_s$, then the generators are not independent.

Proof. Suppose $\text{rank}(\mathbb{H}^{\text{gen}}) < n_s$. Then the n_s rows of \mathbb{H}^{gen} are linearly dependent, which means that

$$\sum_{s=1}^{n_s} a_s \mathbf{r}_{g_s} = 0 \pmod{2} \quad \text{with two or more of the } a_s = 1 .$$

Applying Eq. (E.64) repeatedly you deduce that:

$$g = \prod_{s=1}^{n_s} g_s^{a_s} \quad \Longrightarrow \quad \mathbf{r}_g = 0 .$$

But, if g has a row-vector representative which is 0, it means that g must be a multiple of the identity. Since $g \in \mathcal{S}$, then $g = \mathbf{1}$. Suppose now, without loss of generality, that $a_1 = 1$ in the previous expression, hence:

$$g = \mathbf{1} = g_1 \prod_{s=2}^{n_s} g_s^{a_s} \quad \Longrightarrow \quad g_1 = \prod_{s=2}^{n_s} g_s^{a_s} ,$$

where we used that $g_s^{-1} = g_s$. Hence the generator g_1 is expressed as a product of the other generators: g_1 is not independent. You can actually show that the rank is maximum **if and only if** the generators are independent. ■

Examples. To illustrate the generator check matrix, consider the $[7,1,3]$ Steane code, one of the simplest CSS quantum codes. It is immediate to write down the check matrix of its stabilizers. In block form, it is given by the following 6×14 matrix:

$$\mathbb{H}_{\text{Steane}}^{\text{gen}} = \left[\begin{array}{c|c} \mathbb{H}_1 & 0 \\ \hline 0 & \mathbb{H}_1 \end{array} \right] , \quad (\text{E.65})$$

where $\mathbb{H}_1 = \mathbb{H}_1^{\text{cyclic}} = \mathbb{H}_2^\perp$ is the 3×7 parity check matrix of the $[7, 4, 3]$ Hamming code, see Eq. (E.50), or any other column-permuted variant of it, associated to a different bit-ordering. For a general CSS quantum code, you would have an $n_s \times (2n)$, with $n_s = n - (k_1 - k_2)$, matrix:

$$\mathbb{H}_{\text{CSS}}^{\text{gen}} = \left[\begin{array}{c|c} \mathbb{H}_2^\perp & 0 \\ \hline 0 & \mathbb{H}_1 \end{array} \right] , \quad (\text{E.66})$$

where \mathbb{H}_1 and \mathbb{H}_2^\perp are the parity check matrices of \mathcal{C}_1 and \mathcal{C}_2^\perp . The fact that the classical parity check matrices have maximum rank guarantees that the generator check matrix has also **maximum rank**, hence the generators are independent.

Exercise E.9. Write down $\mathbb{H}_{[5,1,3]}^{\text{gen}}$ — with a bit ordering of your choice — for the five-Qbit code $[5, 1, 3]$ examined in the main text, see Sec. 11.5, with stabilizers given in Eq. (11.30). Observe that, since this is not a CSS quantum code, the generator check matrix is not in block form.

Conjugation with Pauli group operators. An important result which is rather easy to prove with the generator check matrix is the following. Let $\mathcal{S} = \langle \hat{S}_1, \dots, \hat{S}_s, \dots, \hat{S}_{n_s} \rangle$ be a set of independent generators of \mathcal{S} . Denote, as before, $\hat{S}_s = g_s = wg_{(\mathbf{x}_s, \mathbf{z}_s)}$ the generators with their row-vector representative $\mathbf{r}_{g_s} = (\mathbf{x}_s, \mathbf{z}_s)$. Fix a given $s \in \{1, \dots, n_s\}$ in the list of the generator indices. Then, we can show that:

$$\exists g \in \mathcal{P}_n \quad \text{such that} \quad gg_s = -g_s g \quad \text{while} \quad gg_{s'} = g_{s'} g \quad \forall s' \neq s . \quad (\text{E.67})$$

Proof. The independence of the generators guarantees that the $\text{rank}(\mathbb{H}^{\text{gen}}) = n_s$. This implies that \mathbb{H}^{gen} also possesses n_s linearly independent **columns**, hence you can represent *any* vector in \mathbb{B}^{n_s} as $\mathbb{H}^{\text{gen}}\mathbf{w}$, with an appropriate $2n$ -dimensional column vector \mathbf{w} . Consider therefore the \mathbf{w} such that:

$$\left(0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0 \right)^T = \mathbb{H}^{\text{gen}}\mathbf{w},$$

where a single 1 is at position s in the left n_s -dimensional column vector. By considering a row vector \mathbf{r}_g such that $\mathbf{w} = \mathbf{J}\mathbf{r}_g^T$ (i.e., $\mathbf{r}_g^T = -\mathbf{J}\mathbf{w} = \mathbf{J}\mathbf{w}$, since $\mathbf{J}^2 = -\mathbf{1}$ and we are considering Boolean variables) you deduce that:

$$\left(0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0 \right)^T = \mathbb{H}^{\text{gen}}\mathbf{J}\mathbf{r}_g^T \iff \begin{cases} \mathbf{r}_{g_s}\mathbf{J}\mathbf{r}_g^T = 1 \\ \mathbf{r}_{g_{s'}}\mathbf{J}\mathbf{r}_g^T = 0 \quad \forall s' \neq s \end{cases},$$

which implies the thesis, due to Eq. (E.59). ■

Exercise E.10. Using the same ideas of the previous proof, show that if $\mathbf{a} = (a_1, \dots, a_s, \dots, a_{n_s})^T \in \mathbb{B}^{n_s}$ is a Boolean vector denoting the positions (where $a_s = 1$) of the generators you want to anti-commute with a $g \in \mathcal{P}_n$, with all the other generators (where $a_s = 0$) commuting, then such a $g_{\mathbf{a}} \in \mathcal{P}_n$ can always be found:

$$\exists g_{\mathbf{a}} \in \mathcal{P}_n \quad \text{such that} \quad g_{\mathbf{a}}\hat{S}_s = -\hat{S}_s g_{\mathbf{a}} \quad (\text{if } a_s = 1) \quad \text{and} \quad g_{\mathbf{a}}\hat{S}_s = \hat{S}_s g_{\mathbf{a}} \quad (\text{if } a_s = 0). \quad (\text{E.68})$$

The dimension of the stabilized space. Given a set of independent generators $\mathcal{S} = \langle \hat{S}_1, \dots, \hat{S}_{n_s} \rangle$ acting in the full Hilbert space \mathcal{H}_n , we argued several times that the dimension of the stabilized subspace $\mathcal{H}_{\mathcal{S}}$ is given by $k = n - n_s$, essentially because every independent constraint $\hat{S}_s|\psi\rangle = |\psi\rangle$ imposed on the states reduces the dimension of the Hilbert space by a factor 2. This is very reasonable. We will now prove it more formally by using the techniques we have developed.

Indeed, consider an $\mathbf{a} = (a_1, \dots, a_s, \dots, a_{n_s})^T \in \mathbb{B}^{n_s}$. Define now the following projectors:

$$\hat{\Pi}_S^{(\mathbf{a})} = \frac{1}{2^{n_s}} \prod_{s=1}^{n_s} (\mathbf{1} + (-1)^{a_s} \hat{S}_s). \quad (\text{E.69})$$

It is easy to see that these are **orthogonal projectors**, with $\hat{\Pi}_S^{(\mathbf{a})}$ projecting onto the subsector with an eigenvalues syndrome $\boldsymbol{\lambda} = (-1)^{\mathbf{a}}$ for the generators. In particular $\hat{\Pi}_S^{(\mathbf{0})}$ is the projector on the stabilized subsector $\mathcal{H}_{\mathcal{S}}$. Moreover, they give us a resolution of the identity:

$$\mathbf{1} = \sum_{\mathbf{a} \in \mathbb{B}^{n_s}} \hat{\Pi}_S^{(\mathbf{a})}. \quad (\text{E.70})$$

Now, by applying the result of Exercise E.10, you see that for any \mathbf{a} , a Pauli operator $g_{\mathbf{a}} \in \mathcal{P}_n$ exists such that $g_{\mathbf{a}}$ **anti-commutes** with all the generators in which $a_s = 1$, and **commutes** with the others in which $a_s = 0$. With this trick you can effectively cancel the factor $(-1)^{a_s}$ in the projectors, arriving at:

$$g_{\mathbf{a}}\hat{\Pi}_S^{(\mathbf{0})} = \hat{\Pi}_S^{(\mathbf{a})}g_{\mathbf{a}} \implies g_{\mathbf{a}}\hat{\Pi}_S^{(\mathbf{0})}g_{\mathbf{a}}^\dagger = \hat{\Pi}_S^{(\mathbf{a})}. \quad (\text{E.71})$$

Since $g_{\mathbf{a}}$ is **unitary**, this shows that all the projectors have the same dimensionality 2^k of the space they project on. Hence, using Eq. (E.70):

$$2^n = 2^{n_s} 2^k \implies k = n - n_s. \quad (\text{E.72})$$

E.3.1. Measurements in the Stabilizer formalism

We now show how measurements of Pauli string operators can be described in the Stabilizer formalism. Suppose you have a Pauli operator $g \in \mathcal{P}_n$ which you want to measure on a state $|\psi\rangle \in \mathcal{H}_S$ belonging to the stabilized subsector. Clearly, g must be *Hermitean*, since we pretend to be an observable. Then $g = g^\dagger$ and $g^2 = \mathbf{1}$, so that g has eigenvalues ± 1 . Let $\mathcal{S} = \langle \hat{S}_1, \hat{S}_2, \dots, \hat{S}_{n_s} \rangle$ be the stabilizers of \mathcal{H}_S . Two situations are possible:

Case 1) g commutes with all the generators: $g\hat{S}_s = \hat{S}_s g$, for $s = 1, \dots, n_s$.

Case 2) g anti-commutes with one of the generators, say \hat{S}_1 , $g\hat{S}_1 = -\hat{S}_1 g$, and commutes with all the others. If you suspect that this is not general enough, argue as follows. If also $g\hat{S}_2 = -\hat{S}_2 g$, then g commutes with $\hat{S}_1\hat{S}_2$, and you can always substitute $\hat{S}_2 \rightarrow \hat{S}_1\hat{S}_2$, returning to the stated case. In the same way you can proceed if g anti-commutes with more than two generators.

Case 1): Since for all generators $\hat{S}_s g|\psi\rangle = g\hat{S}_s|\psi\rangle = g|\psi\rangle$ we deduce that $g|\psi\rangle = \lambda|\psi\rangle$. But $g^2 = \mathbf{1}$, hence $\lambda = \pm 1$. But $g|\psi\rangle = \pm|\psi\rangle$ means that either g or $-g$ is in \mathcal{S} . If $g \in \mathcal{S}$, then $g|\psi\rangle = |\psi\rangle$ and a measurement of g on $|\psi\rangle$ gives the result $+1$ with probability 1, $\text{Prob}_g(+1|\psi) = 1$, and the **state is not modified by the measurement**. If $-g \in \mathcal{S}$, then $g|\psi\rangle = -|\psi\rangle$ and a measurement of g on $|\psi\rangle$ gives the result -1 with probability 1, $\text{Prob}_g(-1|\psi) = 1$: again the state is not modified by the measurement.

Case 2): The probability of measuring each of the eigenvalues, ± 1 , is expressed in terms of projectors as follows:

$$\text{Prob}_g(\pm 1|\psi) = \text{Tr}(\hat{\Pi}_\pm^g |\psi\rangle\langle\psi|) \quad \text{with} \quad \hat{\Pi}_\pm^g = \frac{1}{2}(\mathbf{1} \pm g). \quad (\text{E.73})$$

Now, by exploiting the anti-commutation with \hat{S}_1 and the fact that $\hat{S}_1|\psi\rangle = |\psi\rangle$, we deduce that:

$$\begin{aligned} \text{Prob}_g(+1|\psi) &= \text{Tr}\left(\frac{1}{2}(\mathbf{1} + g)\hat{S}_1|\psi\rangle\langle\psi|\right) = \text{Tr}\left(\hat{S}_1\frac{1}{2}(\mathbf{1} - g)|\psi\rangle\langle\psi|\right) \\ &= \text{Tr}\left(\frac{1}{2}(\mathbf{1} - g)|\psi\rangle\langle\psi|\right) = \text{Prob}_g(-1|\psi), \end{aligned} \quad (\text{E.74})$$

where we also used the cyclic property of the trace to reabsorb \hat{S}_1 in the bra $\langle\psi|$. Since the two probabilities are identical, we conclude that:

$$\text{Prob}_g(+1|\psi) = \text{Prob}_g(-1|\psi) = \frac{1}{2}. \quad (\text{E.75})$$

Concerning the collapse of the state after the measurement, in each of the two possible outcomes, the post-measurement state is:

$$|\psi^\pm\rangle = \frac{1}{\sqrt{2}}(\mathbf{1} \pm g)|\psi\rangle \quad (\text{E.76})$$

with the two states connected by the former stabilizer \hat{S}_1 :

$$\hat{S}_1|\psi^-\rangle = \hat{S}_1\frac{1}{\sqrt{2}}(\mathbf{1} - g)|\psi\rangle = \frac{1}{\sqrt{2}}(\mathbf{1} + g)\hat{S}_1|\psi\rangle = |\psi^+\rangle. \quad (\text{E.77})$$

i

The post-measurement stabilized subspace. A moment reflection shows that the post-measurement state $|\psi^\pm\rangle$ is stabilized by $\mathcal{S}^\pm = \langle \pm g, \hat{S}_2, \dots, \hat{S}_{n_s} \rangle$.

E.3.2. The construction of logical \mathbf{X} and \mathbf{Z} for stabilizer codes

Generally speaking, the generator check matrix is an $n_s \times 2n$ matrix of the form:

$$\mathbb{H}^{\text{gen}} = [\mathbb{H}_x \parallel \mathbb{H}_z] , \quad (\text{E.78})$$

where \mathbb{H}^x and \mathbb{H}^z are $n_s \times n$ and take care of the \mathbf{X} and \mathbf{Z} positions, respectively. While \mathbb{H}^{gen} has maximum rank n_s , in general $r = \text{rank}(\mathbb{H}_x) \leq n_s$. There are a number of transformations that you can do to your bit-ordering and generators that modify the form of the matrix:

Swap rows) This corresponds to relabelling the generators.

Swap columns) If you swap columns in the **same** way in both \mathbb{H}^x and \mathbb{H}^z , this corresponds to a different bit-ordering.

Sum rows) According to Eq. (E.64), we have that $\mathbf{r}_{g_1 g_2} = \mathbf{r}_{g_1} \oplus \mathbf{r}_{g_2}$, which means that you can replace the pair $\hat{S}_1, \hat{S}_2 \rightarrow \hat{S}_1, \hat{S}_1 \hat{S}_2$ by simply substituting $\mathbf{r}_{g_2} \rightarrow \mathbf{r}_{g_1} \oplus \mathbf{r}_{g_2}$ in the check matrix. And this can be done for any pairs of rows.

The standard form of the generator check matrix. It is a rather boring but not difficult exercise of Gaussian elimination with Boolean variables, see [3][Sec. 10.5.7] for details, to show that you can always transform the generator check matrix to the following canonical form:

$$\mathbb{H}^{\text{gen}} \rightarrow \mathbb{H}_{\text{canonical}}^{\text{gen}} = \begin{array}{c} r \\ \left[\begin{array}{c|c|c||c|c|c} \mathbf{1} & \mathbb{B}_x & \mathbb{A}_x & \mathbb{B}_{1z} & \mathbf{0} & \mathbb{A}_{1z} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{B}_{2z} & \mathbf{1} & \mathbb{A}_{2z} \end{array} \right] , \quad (\text{E.79}) \\ n_s - r \\ \underbrace{\hspace{1.5cm}}_r \quad \underbrace{\hspace{1.5cm}}_{n_s - r} \quad \underbrace{\hspace{1.5cm}}_k \quad \underbrace{\hspace{1.5cm}}_r \quad \underbrace{\hspace{1.5cm}}_{n_s - r} \quad \underbrace{\hspace{1.5cm}}_k \end{array}$$

with appropriate block matrices. Particularly relevant to the following discussion are the blocks \mathbb{A}_{1z} and \mathbb{A}_{2z} , entering in the expression for the logical \mathbf{X} operator, and the block \mathbb{A}_x entering in the logical \mathbf{Z} operator.

The goal is to define logical \mathbf{Z}_{jL} and \mathbf{X}_{jL} operators (with $j = 1, \dots, k$) — picking them from the Pauli group \mathcal{P}_n and representing them with the usual $2n$ -dimensional row-vectors — in such a way that:

1) \mathbf{Z}_{jL} commute with each other and with all the generators \hat{S}_s , and are such that

$$\langle \hat{S}_1, \dots, \hat{S}_{n_s}, \mathbf{Z}_{1L}, \dots, \mathbf{Z}_{kL} \rangle \quad (\text{E.80})$$

is a subgroup generated by $n_s + s = n$ commuting **independent** operators. This guarantees that they uniquely identify a common eigenstate $|0_L\rangle \in \mathcal{H}_S$ with eigenvalue $+1$ for all of them.

2) \mathbf{X}_{jL} commute with each other and with all the generators \hat{S}_s , and are such that

$$\langle \hat{S}_1, \dots, \hat{S}_{n_s}, \mathbf{X}_{1L}, \dots, \mathbf{X}_{kL} \rangle \quad (\text{E.81})$$

is a subgroup generated by $n_s + s = n$ commuting **independent** operators.

3) \mathbf{Z} and \mathbf{X} **anti-commute** for the same index, $\mathbf{X}_{jL} \mathbf{Z}_{jL} = -\mathbf{Z}_{jL} \mathbf{X}_{jL}$, while $\mathbf{X}_{jL} \mathbf{Z}_{j'L} = \mathbf{Z}_{j'L} \mathbf{X}_{jL}$ if $j' \neq j$.

To check commutation with the generators, we recall the symplectic product trick in Eq. (E.59): we should check that $\mathbf{r}_{g_s} \mathbf{J} \mathbf{r}_g^T = 0$, where \mathbf{r}_{g_s} is the row-vector representative of \hat{S}_s , and \mathbf{r}_g that of $g = \mathbf{Z}_{j_L}$ or \mathbf{X}_{j_L} . Hence, we prepare the expression for $\mathbb{H}_{\text{canonical}}^{\text{gen}} \mathbf{J}$, which is given by:

$$\mathbb{H}_{\text{canonical}}^{\text{gen}} \mathbf{J} = \begin{array}{c} r \\ \left\{ \begin{array}{c|c|c||c|c|c} \mathbb{B}_{1z} & 0 & \mathbb{A}_{1z} & \mathbf{1} & \mathbb{B}_x & \mathbb{A}_x \\ \hline \mathbb{B}_{2z} & \mathbf{1} & \mathbb{A}_{2z} & 0 & 0 & 0 \end{array} \right. \\ n_s - r \end{array}, \quad (\text{E.82})$$

$\underbrace{\hspace{2cm}}_r \quad \underbrace{\hspace{2cm}}_{n_s-r} \quad \underbrace{\hspace{2cm}}_k \quad \underbrace{\hspace{2cm}}_r \quad \underbrace{\hspace{2cm}}_{n_s-r} \quad \underbrace{\hspace{2cm}}_k$

It takes little work to show that, if we construct all the \mathbf{Z}_{j_L} only out of physical \mathbf{Z} , without \mathbf{X} , a representation of the k logical \mathbf{Z}_{j_L} given by the following matrix of rank k would work:

$$\mathbb{H}^{\mathbf{Z}} = k \left\{ \left[\begin{array}{c|c|c|c} 0 & \mathbb{A}_x^T & 0 & \mathbf{1} \\ \hline \hline \hline \hline \end{array} \right] \right\}. \quad (\text{E.83})$$

$\underbrace{\hspace{4cm}}_n \quad \underbrace{\hspace{2cm}}_r \quad \underbrace{\hspace{2cm}}_{n_s-r} \quad \underbrace{\hspace{2cm}}_k$

The commutation with the generators follows from: ⁷

$$\mathbb{H}_{\text{canonical}}^{\text{gen}} \mathbf{J} (\mathbb{H}^{\mathbf{Z}})^T = 0. \quad (\text{E.84})$$

Moreover, the independence of the operators in Eq. (E.80) is checked by verifying that the rank of the representative matrix — whose first n_s rows are given by $\mathbb{H}_{\text{canonical}}^{\text{gen}}$ (which has maximum rank n_s), and the last k rows by $\mathbb{H}^{\mathbf{Z}}$ —, is also maximum and equal to $n_s + k = n$.

Next we move to the \mathbf{X}_{j_L} . You can verify that the following choice for the (rank k) representative matrix works:

$$\mathbb{H}^{\mathbf{X}} = k \left\{ \left[\begin{array}{c|c|c||c|c|c} 0 & \mathbb{A}_{2z}^T & \mathbf{1} & \mathbb{A}_{1z}^T & 0 & 0 \\ \hline \hline \hline \hline \hline \end{array} \right] \right\}. \quad (\text{E.85})$$

$\underbrace{\hspace{2cm}}_r \quad \underbrace{\hspace{2cm}}_{n_s-r} \quad \underbrace{\hspace{2cm}}_k \quad \underbrace{\hspace{2cm}}_r \quad \underbrace{\hspace{2cm}}_{n_s-r} \quad \underbrace{\hspace{2cm}}_k$

The independence of the system in Eq. (E.81) is guaranteed by the usual maximum rank argument. The commutator with the stabilizers is easy to check: ⁸

$$\mathbb{H}_{\text{canonical}}^{\text{gen}} \mathbf{J} (\mathbb{H}^{\mathbf{X}})^T = 0. \quad (\text{E.86})$$

It remains to verify that \mathbf{X}_{j_L} and \mathbf{Z}_{j_L} anti-commute for the same j and commute otherwise. We use the symplectic product trick, by calculating:

$$\mathbb{H}^{\mathbf{Z}} \mathbf{J} (\mathbb{H}^{\mathbf{X}})^T = \left[\mathbb{A}_z^T \mid 0 \mid \mathbf{1} \mid 0 \mid 0 \mid 0 \right] \begin{array}{c} \frac{0}{\mathbb{A}_{2z}} \\ \frac{\mathbf{1}}{\mathbb{A}_{1z}} \\ 0 \\ 0 \end{array} = [\mathbf{1}]_{k \times k}. \quad (\text{E.87})$$

⁷You observe that the commutation with the last $n_s - r$ generators is trivial, while that with the first r generators follows from $\mathbf{1}(\mathbb{A}_x^T)^T + \mathbb{A}_x \mathbf{1} = 0$.

⁸For the first r generators, you end up with $\mathbb{A}_{1z} \mathbf{1} + \mathbf{1}(\mathbb{A}_{1z}^T)^T = 0$. For the remaining $n_s - r$ generators, you write $\mathbf{1}(\mathbb{A}_{2z}^T)^T + \mathbb{A}_{2z} \mathbf{1} = 0$.

Indeed, recall that $gg' = -g'g$ if and only if $\mathbf{r}_g \mathbf{J} \mathbf{r}_{g'}^T = 1$, and the diagonal elements equal to 1 in the RHS of Eq. (E.87) guarantee that $\mathbf{X}_{j_L} \mathbf{Z}_{j_L} = -\mathbf{Z}_{j_L} \mathbf{X}_{j_L}$.

E.4. The Gottesman-Knill theorem

The techniques employed so far allow to establish a rather remarkable theorem concerning the fact that quantum circuits composed only by Clifford gates — recall that they are composed exclusively by the Hadamard \mathbf{H}_j , the phase-gate \mathbf{S}_j , and the cNOT $\mathbf{C}_{jj'}$ — are classically simple to simulate, although the relevant states are arbitrarily entangled.

Recall that, for a general a unitary transformation \mathbf{U} in the Hilbert space and a state $|\psi\rangle \in \mathcal{H}_n$ which is eigenstate with eigenvalue $+1$ of a given stabilizer \hat{S} , an Hermitean Pauli string such that $\hat{S}^2 = \mathbf{1}$:

$$\hat{S}|\psi\rangle = |\psi\rangle \implies \mathbf{U}|\psi\rangle = \mathbf{U}\hat{S}|\psi\rangle = (\mathbf{U}\hat{S}\mathbf{U}^\dagger)\mathbf{U}|\psi\rangle. \quad (\text{E.88})$$

Hence $\mathbf{U}\hat{S}\mathbf{U}^\dagger$ has eigenvalue $+1$ on $\mathbf{U}|\psi\rangle$. The transformation $\hat{S} \rightarrow \mathbf{U}\hat{S}\mathbf{U}^\dagger$ is known as **conjugation** by \mathbf{U} .

Stabilized states. There are states in the Hilbert space \mathcal{H}_n that are **uniquely** associated (within an overall phase) to a stabilizer group $\mathcal{S} \subset \mathcal{P}_n$ which has $n_s = n$ independent generators, hence $k = n - n_s = 0$.⁹ For instance $|0\rangle^{\otimes n}$ is stabilized by¹⁰

$$\text{Stab}(|0\rangle^{\otimes n}) = \mathcal{S} = \langle \mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n \rangle.$$

More generally, if $|\psi\rangle$ is stabilized by $\mathcal{S} = \langle \hat{S}_1, \dots, \hat{S}_n \rangle$, then $\mathbf{U}|\psi\rangle$ is stabilized by:

$$\mathbf{U}\mathcal{S}\mathbf{U}^\dagger = \langle \mathbf{U}\hat{S}_1\mathbf{U}^\dagger, \dots, \mathbf{U}\hat{S}_n\mathbf{U}^\dagger \rangle, \quad (\text{E.89})$$

in a way that should reminds you of the Heisenberg representation of operators in QM. Moreover, knowing as the generators transform under conjugation is enough to deduce how a general product of generators transforms, since:

$$\hat{S}_1 \hat{S}_2 \rightarrow \mathbf{U}\hat{S}_1\hat{S}_2\mathbf{U}^\dagger = (\mathbf{U}\hat{S}_1\mathbf{U}^\dagger)(\mathbf{U}\hat{S}_2\mathbf{U}^\dagger). \quad (\text{E.90})$$

Unfortunately, for a general unitary \mathbf{U} the transformation $\hat{S} \rightarrow \mathbf{U}\hat{S}\mathbf{U}^\dagger$ might bring out of the Pauli group. The particular unitaries \mathbf{U} composed only by Clifford gates are such that the conjugation of a Pauli string is still a Pauli string.

The Clifford gates. Indeed, recall that, for the Hadamard $\mathbf{H} = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z})$, which is unitary, Hermitean, and $\mathbf{H}^2 = \mathbf{1}$ (but $\mathbf{H} \notin \mathcal{P}_1$):

$$\mathbf{H}_j \begin{pmatrix} \mathbf{X}_j \\ \mathbf{Y}_j \\ \mathbf{Z}_j \end{pmatrix} \mathbf{H}_j^\dagger = \begin{pmatrix} \mathbf{Z}_j \\ -\mathbf{Y}_j \\ \mathbf{X}_j \end{pmatrix} \quad \text{while} \quad \mathbf{H}_j \begin{pmatrix} \mathbf{X}_{j'} \\ \mathbf{Y}_{j'} \\ \mathbf{Z}_{j'} \end{pmatrix} \mathbf{H}_j^\dagger = \begin{pmatrix} \mathbf{X}_{j'} \\ \mathbf{Y}_{j'} \\ \mathbf{Z}_{j'} \end{pmatrix} \quad (j' \neq j). \quad (\text{E.91})$$

For the \mathbf{S} -gate, $\mathbf{S} = \text{diag}(1, i)$, which is unitary, *non-Hermitean* and such that $\mathbf{S}^2 = \mathbf{Z}$ (but $\mathbf{S} \notin \mathcal{P}_1$):

$$\mathbf{S}_j \begin{pmatrix} \mathbf{X}_j \\ \mathbf{Y}_j \\ \mathbf{Z}_j \end{pmatrix} \mathbf{S}_j^\dagger = \begin{pmatrix} \mathbf{Y}_j \\ -\mathbf{X}_j \\ \mathbf{Z}_j \end{pmatrix} \quad \text{while} \quad \mathbf{S}_j \begin{pmatrix} \mathbf{X}_{j'} \\ \mathbf{Y}_{j'} \\ \mathbf{Z}_{j'} \end{pmatrix} \mathbf{S}_j^\dagger = \begin{pmatrix} \mathbf{X}_{j'} \\ \mathbf{Y}_{j'} \\ \mathbf{Z}_{j'} \end{pmatrix} \quad (j' \neq j). \quad (\text{E.92})$$

⁹See Theorem 1 in Ref. [7].

¹⁰In this section we will number the bits from 1 to n for a simpler notation. Also, the ordering is assumed to be that of linear algebra, with bits ordered from left to right in writing row-vectors.

Finally, for the cNOT-gate $\mathbf{C}_{jj'}$ (with bit j as control and j' as target), which is unitary, Hermitean, and $\mathbf{C}_{jj'}^2 = \mathbf{1}$ (but $\mathbf{C}_{jj'} \notin \mathcal{P}_2$):¹¹

$$\mathbf{C}_{jj'} \begin{pmatrix} \mathbf{X}_j \\ \mathbf{Y}_j \\ \mathbf{Z}_j \end{pmatrix} \mathbf{C}_{jj'}^\dagger = \begin{pmatrix} \mathbf{X}_j \mathbf{X}_{j'} \\ \mathbf{Y}_j \mathbf{X}_{j'} \\ \mathbf{Z}_j \end{pmatrix} \quad \text{and} \quad \mathbf{C}_{jj'} \begin{pmatrix} \mathbf{X}_{j'} \\ \mathbf{Y}_{j'} \\ \mathbf{Z}_{j'} \end{pmatrix} \mathbf{C}_{jj'}^\dagger = \begin{pmatrix} \mathbf{X}_{j'} \\ \mathbf{Z}_j \mathbf{Y}_{j'} \\ \mathbf{Z}_j \mathbf{Z}_{j'} \end{pmatrix}. \quad (\text{E.93})$$

In all the previous expressions, the transformation for the \mathbf{Y} derives from $\mathbf{Y} = i\mathbf{XZ}$, and from the fact that $\mathbf{UYU}^\dagger = i(\mathbf{UXU}^\dagger)(\mathbf{UZU}^\dagger)$.

The Heisenberg representation: follow the operators. Suppose, to illustrate the ideas, that we have $n = 2$ Qbits, and we consider the Clifford group unitary transformation¹²

$$\mathbf{U} = \mathbf{C}_{12} \mathbf{H}_2 \mathbf{C}_{12} \mathbf{S}_2 \mathbf{H}_1. \quad (\text{E.94})$$

Suppose we start from the state $|\psi\rangle = |00\rangle$, which is stabilized by:

$$\text{Stab}(|00\rangle) = \mathcal{S} = \langle \mathbf{Z}_1, \mathbf{Z}_2 \rangle.$$

Since some of the gates generate \mathbf{X} and \mathbf{Y} , starting from \mathbf{Z} , it is mandatory that we study how $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{Z}_1, \mathbf{Z}_2\}$ transform upon conjugation by each gate. From that, you would know how to transform \mathbf{Y} , and indeed any Pauli string in \mathcal{P}_2 .

Exercise E.11. Show that for $\mathbf{U} = \mathbf{C}_{12} \mathbf{H}_2 \mathbf{C}_{12} \mathbf{S}_2 \mathbf{H}_1$:

$$\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \rightarrow \mathbf{U} \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \mathbf{U}^\dagger = \begin{pmatrix} \mathbf{Z}_1 \\ -\mathbf{Y}_2 \\ -\mathbf{Y}_1 \mathbf{Y}_2 \\ \mathbf{Z}_1 \mathbf{X}_2 \end{pmatrix}. \quad (\text{E.95})$$

Having followed the operators in their Heisenberg transformation through the gates, we now know that:

$$\mathbf{U} \text{Stab}(|00\rangle) \mathbf{U}^\dagger = \langle -\mathbf{Y}_1 \mathbf{Y}_2, \mathbf{Z}_1 \mathbf{X}_2 \rangle.$$

The state $\mathbf{U}|00\rangle$ stabilized by these two transformed operators (conjugate to \mathbf{Z}_1 and \mathbf{Z}_2) can be calculated (or verified) to be:

$$\mathbf{U}|00\rangle = \frac{1}{2} \left(|0\rangle_1 \otimes (|0\rangle_2 + |1\rangle_2) - |1\rangle_1 \otimes (|0\rangle_2 - |1\rangle_2) \right).$$

Similarly, since $|\psi\rangle = |01\rangle = \mathbf{X}_2|00\rangle$, then

$$\mathbf{U}|01\rangle = \mathbf{U} \mathbf{X}_2 \mathbf{U}^\dagger \mathbf{U}|00\rangle = (-\mathbf{Y}_2) \mathbf{U}|00\rangle = \frac{i}{2} \left(-|0\rangle_1 \otimes (|0\rangle_2 - |1\rangle_2) + |1\rangle_1 \otimes (|0\rangle_2 + |1\rangle_2) \right).$$

In a similar way we can calculate $\mathbf{U}|10\rangle$ and $\mathbf{U}|11\rangle$, giving full access to the effect of \mathbf{U} on the computational basis.



States vs operators. It should be clear how much more cumbersome is working with states, with respect to working with a set of rather simple rules to transform the operators.

¹¹Recall that $\mathbf{H}_j \mathbf{H}_{j'} \mathbf{C}_{jj'} \mathbf{H}_j \mathbf{H}_{j'} = \mathbf{C}_{j'j}$, and:

$$\mathbf{C}_{jj'} = \frac{1}{2}(\mathbf{1} + \mathbf{Z})_j + \frac{1}{2}(\mathbf{1} - \mathbf{Z})_j \mathbf{X}_{j'}.$$

¹²This is example 3 from Gottesman paper arXiv:9807006.

Entanglement and Clifford gates. The previous example already shows that entanglement is created by applying Clifford gates. As a further illustration of the stabilizer formalism, consider the Bell state construction for $n = 2$ -Qbits. Recall that:

$$\mathbf{C}_{12}\mathbf{H}_1|0\rangle_1 \otimes |0\rangle_2 = \frac{1}{2}(|0\rangle_1 \otimes |0\rangle_2 + |1\rangle_1 \otimes |1\rangle_2).$$

Exercise E.12. Show that for $\mathbf{U} = \mathbf{C}_{12}\mathbf{H}_1$:

$$\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \rightarrow \mathbf{U} \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \mathbf{U}^\dagger = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_1\mathbf{X}_2 \\ \mathbf{Z}_1\mathbf{Z}_2 \end{pmatrix}. \quad (\text{E.96})$$

The initial state of the computation $|\psi\rangle = |00\rangle$ is stabilized by $\mathcal{S} = \langle \mathbf{Z}_1, \mathbf{Z}_2 \rangle$. By following the stabilizer generators through the Heisenberg transformation, we discover that:

$$\mathbf{U} \langle \mathbf{Z}_1, \mathbf{Z}_2 \rangle \mathbf{U}^\dagger = \langle \mathbf{X}_1\mathbf{X}_2, \mathbf{Z}_1\mathbf{Z}_2 \rangle.$$

The final Bell (maximally entangled) state is the stabilized state of $\langle \mathbf{X}_1\mathbf{X}_2, \mathbf{Z}_1\mathbf{Z}_2 \rangle$, after the application of the two Clifford gates.

A general approach. In order to perform the Heisenberg transformations more efficiently — after all, the rules are very simple, see Eqs. (E.91)-(E.93) —, we get equipped with a tool very similar to that used for generator matrices of stabilizer groups, except that now we want a notation that keeps track of the phase factor in front of a Pauli string. Once again, we use the convention that:

$$\begin{cases} \pm \mathbf{X}_j \leftrightarrow (0 \cdots 0 \ 1_j \ 0 \cdots 0 \parallel 0 \cdots 0 \ 0_j \ 0 \cdots 0 \mid \pm 1) \\ \pm \mathbf{Z}_j \leftrightarrow (0 \cdots 0 \ 0_j \ 0 \cdots 0 \parallel 0 \cdots 0 \ 1_j \ 0 \cdots 0 \mid \pm 1) \\ \pm \mathbf{Y}_j \leftrightarrow (0 \cdots 0 \ 1_j \ 0 \cdots 0 \parallel 0 \cdots 0 \ 1_j \ 0 \cdots 0 \mid \pm 1) \end{cases}. \quad (\text{E.97})$$

Notice that we do not trade here \mathbf{Y} for $i\mathbf{XZ}$, disregarding the i .

1 Row-vector associated to an Hermitean Pauli string. To represent an Hermitean Pauli string that squares to $\mathbf{1}$, including the overall \pm sign, we need $2n + 1$ bits. For instance, for $n = 4$ Qbits:

$$-\mathbf{Y}_1\mathbf{X}_3\mathbf{Z}_4 \leftrightarrow (1 \ 0 \ 1 \ 0 \parallel 1 \ 0 \ 0 \ 1 \mid -1). \quad (\text{E.98})$$

It is obvious that all you need is to follow how the operators $\{\mathbf{X}_1, \dots, \mathbf{X}_n, \mathbf{Z}_1, \dots, \mathbf{Z}_n\}$ “evolve”, as the various gates are applied. The relevant information is therefore contained in a *table* which is $2n \times (2n + 1)$, which is updated with simple rules each time a Clifford gate is applied.

To illustrate the idea, consider just the first transformation $\mathbf{H}_1\mathbf{S}_2$ in the \mathbf{U} considered in Eq. (E.94). Then:

$$\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \leftrightarrow \begin{pmatrix} 1 & 0 & \parallel & 0 & 0 & \parallel & +1 \\ 0 & 1 & \parallel & 0 & 0 & \parallel & +1 \\ 0 & 0 & \parallel & 1 & 0 & \parallel & +1 \\ 0 & 0 & \parallel & 0 & 1 & \parallel & +1 \end{pmatrix} \xrightarrow{\mathbf{H}_1\mathbf{S}_2} \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Y}_2 \\ \mathbf{X}_1 \\ \mathbf{Z}_2 \end{pmatrix} \leftrightarrow \begin{pmatrix} 0 & 0 & \parallel & 1 & 0 & \parallel & +1 \\ 0 & 1 & \parallel & 0 & 1 & \parallel & +1 \\ 1 & 0 & \parallel & 0 & 0 & \parallel & +1 \\ 0 & 0 & \parallel & 0 & 1 & \parallel & +1 \end{pmatrix}.$$

And we might proceed in a similar way, by applying then \mathbf{C}_{12} and so forth, obtaining in the end:

$$\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \leftrightarrow \left(\begin{array}{cc|cc|c} 1 & 0 & 0 & 0 & +1 \\ 0 & 1 & 0 & 0 & +1 \\ \hline 0 & 0 & 1 & 0 & +1 \\ 0 & 0 & 0 & 1 & +1 \end{array} \right) \xrightarrow{\mathbf{U}} \begin{pmatrix} \mathbf{Z}_1 \\ -\mathbf{Y}_2 \\ -\mathbf{Y}_1\mathbf{Y}_2 \\ \mathbf{Z}_1\mathbf{X}_2 \end{pmatrix} \leftrightarrow \left(\begin{array}{cc|cc|c} 0 & 0 & 1 & 0 & +1 \\ 0 & 1 & 0 & 1 & -1 \\ \hline 1 & 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & 0 & +1 \end{array} \right).$$

Measurements As an example of measurement, let us assume that we have a state $|\psi_{\text{in}}\rangle = |\psi\rangle_1 \otimes |0\rangle_2$, with $|\psi\rangle_1 = z_0|0\rangle_1 + z_1|1\rangle_1$: indeed a $k = 1$ stabilized subsector \mathcal{H}_S , whose stabilizer is $\mathcal{S} = \langle \mathbf{Z}_2 \rangle$. Let us now apply $\mathbf{U} = \mathbf{C}_{12}$ to $|\psi_{\text{in}}\rangle$:

$$\mathbf{U}|\psi_{\text{in}}\rangle = \mathbf{C}_{12}|\psi_{\text{in}}\rangle = |\psi_{\text{fin}}\rangle = z_0|0\rangle_1 \otimes |0\rangle_2 + z_1|1\rangle_1 \otimes |1\rangle_2,$$

which is stabilized by $\mathbf{USU}^\dagger = \langle \mathbf{Z}_1\mathbf{Z}_2 \rangle$, as seen from Eq. (E.93). Now suppose we measure $g = \mathbf{Y}_2$ on this state. \mathbf{Y}_2 **anti-commutes** with the Heisenberg transformed stabilizer $\hat{\mathbf{S}}_1 = \mathbf{Z}_1\mathbf{Z}_2$, hence we are in Case 2) of the general discussion of Sec. E.3.1. If you revisit that discussion, you realize that the outcome is random, with $\text{Prob}_g(+1) = \text{Prob}_g(-1) = \frac{1}{2}$. If the outcome is $+1$, then the state collapses to

$$|\psi^+\rangle = \frac{1}{\sqrt{2}}(\mathbf{1} + \mathbf{Y}_2)|\psi_{\text{fin}}\rangle = (z_0|0\rangle_1 - iz_1|1\rangle_1) \otimes \frac{1}{\sqrt{2}}(|0\rangle_2 + i|1\rangle_2),$$

which is stabilized by $\langle \mathbf{Y}_2 \rangle$. If the outcome is -1 , then the state collapses to

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}(\mathbf{1} - \mathbf{Y}_2)|\psi_{\text{fin}}\rangle = (z_0|0\rangle_1 + iz_1|1\rangle_1) \otimes \frac{1}{\sqrt{2}}(|0\rangle_2 - i|1\rangle_2),$$

which is stabilized by $\langle -\mathbf{Y}_2 \rangle$. The two states are connected by the old stabilizer:

$$\mathbf{Z}_1\mathbf{Z}_2|\psi^-\rangle = |\psi^+\rangle.$$

This can be rephrased as follows: Although the act of measuring involves a collapse and a projector, hence not an operator in the Pauli group, still, **conditioned** on the result of the measurement of \mathbf{Y}_2 , if we get $+1$, we do nothing and obtain $|\psi^+\rangle$, if we get -1 , we apply the old stabilizer $\mathbf{Z}_1\mathbf{Z}_2$ and we obtain again $|\psi^+\rangle$.

Setting up a table for the computation. Let us see how we can actually compute with a classical algorithm the Heisenberg transformations induced by a given circuit. To simplify our treatment, let us assume that we start from the standard computational state $|\psi_{\text{in}}\rangle = |0\rangle^{\otimes n}$. This is a stabilized state with stabilizer $\text{Stab}(|0\rangle^{\otimes n}) = \langle \mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n \rangle$. In order to be able to follow the evolution of any Pauli string operator, we need also to invoke the $\mathbf{X}_1, \dots, \mathbf{X}_n$, which are not stabilizers, but together with the stabilizer they generate any Hermitean Pauli string. The initial value of our table is therefore:

$$\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_n \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_n \end{pmatrix} \leftrightarrow \left(\begin{array}{cccc|cccc|c} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & +1 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & +1 \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & +1 \\ \hline 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & +1 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & +1 \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & +1 \end{array} \right). \quad (\text{E.99})$$

As the computation proceeds, by applying the different Clifford gates in \mathbf{U} , the operators are transformed into more complex Pauli strings, represented by a matrix of the form:

$$\begin{pmatrix} \mathbf{UX}_1\mathbf{U}^\dagger \\ \mathbf{UX}_2\mathbf{U}^\dagger \\ \vdots \\ \mathbf{UX}_n\mathbf{U}^\dagger \\ \hline \mathbf{UZ}_1\mathbf{U}^\dagger \\ \mathbf{UZ}_2\mathbf{U}^\dagger \\ \vdots \\ \mathbf{UZ}_n\mathbf{U}^\dagger \end{pmatrix} \leftrightarrow \begin{pmatrix} \mathbb{X}_{11}^D & \mathbb{Z}_{12}^D & \cdots & \mathbb{X}_{1n}^D & \parallel & \mathbb{Z}_{11}^D & \mathbb{Z}_{12}^D & \cdots & \mathbb{Z}_{1n}^D & \mid & \mathbf{P}_1^D \\ \mathbb{X}_{21}^D & \mathbb{X}_{22}^D & \cdots & \mathbb{X}_{2n}^D & \parallel & \mathbb{Z}_{21}^D & \mathbb{Z}_{22}^D & \cdots & \mathbb{Z}_{2n}^D & \mid & \mathbf{P}_2^D \\ \vdots & \vdots & \ddots & \vdots & \parallel & \vdots & \vdots & \ddots & \vdots & \mid & \vdots \\ \mathbb{X}_{n1}^D & \mathbb{X}_{n2}^D & \cdots & \mathbb{X}_{nn}^D & \parallel & \mathbb{Z}_{n1}^D & \mathbb{Z}_{n2}^D & \cdots & \mathbb{Z}_{nn}^D & \mid & \mathbf{P}_n^D \\ \hline \mathbb{X}_{11}^S & \mathbb{Z}_{12}^S & \cdots & \mathbb{X}_{1n}^S & \parallel & \mathbb{Z}_{11}^S & \mathbb{Z}_{12}^S & \cdots & \mathbb{Z}_{1n}^S & \mid & \mathbf{P}_1^S \\ \mathbb{X}_{21}^S & \mathbb{X}_{22}^S & \cdots & \mathbb{X}_{2n}^S & \parallel & \mathbb{Z}_{21}^S & \mathbb{Z}_{22}^S & \cdots & \mathbb{Z}_{2n}^S & \mid & \mathbf{P}_2^S \\ \vdots & \vdots & \ddots & \vdots & \parallel & \vdots & \vdots & \ddots & \vdots & \mid & \vdots \\ \mathbb{X}_{n1}^S & \mathbb{X}_{n2}^S & \cdots & \mathbb{X}_{nn}^S & \parallel & \mathbb{Z}_{n1}^S & \mathbb{Z}_{n2}^S & \cdots & \mathbb{Z}_{nn}^S & \mid & \mathbf{P}_n^S \end{pmatrix} = \left(\begin{array}{c|c|c} \mathbb{X}^D & \mathbb{Z}^D & \mathbf{P}^D \\ \mathbb{X}^S & \mathbb{Z}^S & \mathbf{P}^S \end{array} \right). \quad (\text{E.100})$$

The state being given by $|\psi\rangle = \mathbf{U}|\psi_{\text{in}}\rangle$, you recognize that the lower part of the matrix has to do with the **stabilizer** of $|\psi\rangle$:

$$\text{Stab}(|\psi\rangle) = \langle \mathbf{UZ}_1\mathbf{U}^\dagger, \mathbf{UZ}_2\mathbf{U}^\dagger, \dots, \mathbf{UZ}_n\mathbf{U}^\dagger \rangle$$

associated to the binary matrices \mathbb{X}^S , \mathbb{Z}^S , and the binary vector \mathbf{P}^S encoding the ± 1 phase factor. ¹³ The upper part of the matrix is related to what might be called the **destabilizers**, i.e., Pauli strings that help in generating the Pauli group, with a characteristic **anti-commutation** of $\mathbf{UX}_j\mathbf{U}^\dagger$ with the corresponding stabilizer $\mathbf{UZ}_j\mathbf{U}^\dagger$. The corresponding matrices and phase factors are given by \mathbb{X}^D , \mathbb{Z}^D , and \mathbf{P}^D . As you apply any Clifford gate, the table

$$\mathbb{H}^{\text{tab}} = \left(\begin{array}{c|c|c} \mathbb{X}^D & \mathbb{Z}^D & \mathbf{P}^D \\ \mathbb{X}^S & \mathbb{Z}^S & \mathbf{P}^S \end{array} \right), \quad (\text{E.101})$$

should be updated following Eqs. (E.91)-(E.93), and you can easily verify that:

Action of \mathbf{H}_j) For all $i = 1, \dots, n$ you set: ¹⁴

$$\begin{cases} \mathbb{X}_{ij}^{D/S} \longleftrightarrow \mathbb{Z}_{ij}^{D/S} \\ \mathbf{P}_i^{D/S} \longleftarrow \mathbf{P}_i^{D/S} \oplus \mathbb{X}_{ij}^{D/S} \mathbb{Z}_{ij}^{D/S} \end{cases}. \quad (\text{E.102})$$

Action of \mathbf{S}_j) For all $i = 1, \dots, n$ you set:

$$\begin{cases} \mathbb{Z}_{ij}^{D/S} \longleftarrow \mathbb{Z}_{ij}^{D/S} \oplus \mathbb{X}_{ij}^{D/S} \\ \mathbf{P}_i^{D/S} \longleftarrow \mathbf{P}_i^{D/S} \oplus \mathbb{X}_{ij}^{D/S} \mathbb{Z}_{ij}^{D/S} \end{cases}. \quad (\text{E.103})$$

Action of $\mathbf{C}_{jj'}$) For all $i = 1, \dots, n$ you set:

$$\begin{cases} \mathbb{X}_{ij'}^{D/S} \longleftarrow \mathbb{X}_{ij'}^{D/S} \oplus \mathbb{X}_{ij}^{D/S} \\ \mathbb{Z}_{ij}^{D/S} \longleftarrow \mathbb{Z}_{ij}^{D/S} \oplus \mathbb{Z}_{ij'}^{D/S} \\ \mathbf{P}_i^{D/S} \longleftarrow \mathbf{P}_i^{D/S} \oplus \mathbb{X}_{ij}^{D/S} \mathbb{Z}_{ij'}^{D/S} (\mathbb{X}_{ij'}^{D/S} \oplus \mathbb{Z}_{ij}^{D/S} \oplus 1) \end{cases}. \quad (\text{E.104})$$

Exercise E.13. Verify that these rules in Eqs. (E.102)-(E.104) encode precisely the content of Eqs. (E.91)-(E.93), with the table convention on how the various Pauli operators are represented.

¹³It might be convenient to switch from the binary sign ± 1 , to the more standard bit encoding of sign $\mathbf{P} \rightarrow (1 - \mathbf{P})/2$.

¹⁴Notice the phase flip associated to \mathbf{Y} .

i **Clifford gate rules for updating the table.** It is clear that the rules behind the action of the Clifford gates on Pauli strings are very simple, see Eqs. (E.102)-(E.104), and they can be implemented in a classical digital algorithm which works on the $(2n) \times (2n + 1)$ representative table \mathbb{H}^{tab} . The algorithm scales **polynomially** with the number of Qbits n and with the number of elementary Clifford gates composing the unitary \mathbf{U} .

How to deal with measurements using the computational table. It remains to see what happens as we measure some Pauli operators — for simplicity, think of measuring in the computational basis, i.e., operators $g \in \mathcal{P}_n$ which are composed only of \mathbf{Z}_j terms, or, even simpler, by a single term, $g = \mathbf{Z}_j$. The general theory was already presented in Sec. (E.3.1), but we now want to understand how these rules are encoded in \mathbb{H}^{tab} . We have to understand if:

Case 1) $g = \mathbf{Z}_j$ commutes with the stabilizer group, in which case the outcome is determinate, either $\lambda = +1$ or $\lambda = -1$ is obtained with probability 1, and the state is unchanged.

Case 2) $g = \mathbf{Z}_j$ anti-commutes with one (or more) generator of the stabilizer group, call it $\hat{S}_{i_1} = \mathbf{U}\mathbf{Z}_{i_1}\mathbf{U}^\dagger$, in which case the outcome is random, $\lambda = \pm 1$ with probability 1/2, the state is collapsed

$$|\psi\rangle \rightarrow |\psi^\pm\rangle = \frac{1}{\sqrt{2}}(\mathbf{1} \pm g)|\psi\rangle,$$

and the operator $\pm g$ takes the place of the anti-commuting Pauli string in the new stabilizer group. Even simpler, you can always reduce yourself to the case where $+g$ is installed in the stabilizer, by observing that:

$$\hat{S}_{i_1}|\psi^-\rangle = \hat{S}_{i_1}\frac{1}{\sqrt{2}}(\mathbf{1} - g)|\psi\rangle = \frac{1}{\sqrt{2}}(\mathbf{1} + g)|\psi\rangle = |\psi^+\rangle.$$

Question: How do we understand from \mathbb{H}^{tab} if we are in Case 1) or 2)?

The answer is very simple. Just check if \mathbf{X}_j or \mathbf{Y}_j is present in one of the stabilizer generators, i.e., check the value of $\mathbb{X}_{i_j}^S$ for all $i = 1 \dots n$. If $\mathbb{X}_{i_j}^S = 0$ for all $i = 1 \dots n$, then \mathbf{Z}_j commutes with the stabilizer, and we are in **Case 1)**. If an $i_1 \in \{1, \dots, n\}$ exist (take the smallest) such that $\mathbb{X}_{i_1 j}^S = 1$, then \mathbf{Z}_j anti-commutes with the corresponding stabilizer generator, and we are in **Case 2)**.

What to do in Case 2) Let i_1 be the smallest index where $\mathbb{X}_{i_1 j}^S = 1$, corresponding to the stabilizer generator $\hat{S}_{i_1} = \mathbf{U}\mathbf{Z}_{i_1}\mathbf{U}^\dagger$ which have a \mathbf{X}_j or a \mathbf{Y}_j in their Pauli string. Then:

1. Multiply by \hat{S}_{i_1} all the Pauli strings — except that of \hat{S}_{i_1} — where either $\mathbb{X}_{i_j}^S = 1$ or $\mathbb{X}_{i_j}^D = 1$, i.e., anti-commuting with $g = \mathbf{Z}_j$. For the anti-commuting stabilizers, this amounts to setting $\hat{S}_i \rightarrow \hat{S}_{i_1}\hat{S}_i$, so that they now properly commute with g . The opposite effect is obtained for the destabilizers $g_k = \mathbf{U}\mathbf{X}_{i_k}\mathbf{U}^\dagger$ such that $\mathbb{X}_{i_k j}^D = 1$: setting $g_k \rightarrow \hat{S}_{i_1}g_k$ effective makes all these operators *anti-commuting* with g , but commuting among themselves. This multiplication of operators is practically obtained by **summing representative rows** in \mathbb{H}^{tab} , with a little extra care needed in determining the corresponding parity bits \mathbb{P}_i^S , for which I refer the reader to Ref. [7].
2. Move \hat{S}_{i_1} to the destabilizers, by setting for all $j = 1, \dots, n$:

$$\mathbb{X}_{i_1, j}^D \leftarrow \mathbb{X}_{i_1, j}^S, \quad \mathbb{Z}_{i_1, j}^D \leftarrow \mathbb{Z}_{i_1, j}^S, \quad \mathbb{P}_{i_1}^D \leftarrow \mathbb{P}_{i_1}^S.$$

3. Put $\mathbf{Z}_{\underline{j}}$ in the list of stabilizer generators in place of \hat{S}_{i_1} , by setting for all j :

$$\mathbb{X}_{i_1,j}^S = 0, \quad \mathbb{Z}_{i_1,j}^S = 0, \quad \text{except for} \quad \mathbb{Z}_{i_1,j}^S = 1.$$

The value of $P_{i_1}^S$ is set to 0 or 1 with probability 1/2: it is the eigenvalue outcome of the random measurement.

What to do in Case 1) The fact that $g = \mathbf{Z}_{\underline{j}}$ commutes with all the stabilizers $\mathbf{U}\mathbf{Z}_i\mathbf{U}^\dagger$ implies that one must have, for an appropriate binary string $\mathbf{a} = (a_1, \dots, a_n)$:

$$\mathbf{Z}_{\underline{j}} = \lambda \prod_{i=1}^n (\mathbf{U}\mathbf{Z}_i\mathbf{U}^\dagger)^{a_i}, \tag{E.105}$$

where $\lambda = \pm 1$ is the determinate outcome of the measurement. The row representative of the LHS is simply $(0 \cdots 0 | 0 \cdots 0 | 0)$. The RHS operator (without overall) has a row-representative (which should match with the LHS):

$$\sum_{i=1}^n a_i (\mathbb{X}_{i_1}^S \cdots \mathbb{X}_{i_n}^S | \mathbb{Z}_{i_1}^S \cdots \mathbb{Z}_{i_n}^S | P_i^S).$$

Hence, evidently:

$$\left\{ \begin{array}{l} \sum_{i=1}^n a_i P_i^S = 0 \quad \implies \quad \lambda = +1 \\ \sum_{i=1}^n a_i P_i^S = 1 \quad \implies \quad \lambda = -1 \end{array} \right. . \tag{E.106}$$

All you need to do, to determine if $\lambda = +1$ or -1 , is to find out the binary string \mathbf{a} , by solving the linear equations

$$(0 \cdots 0 | 0 \cdots 0) = \sum_{i=1}^n a_i (\mathbb{X}_{i_1}^S \cdots \mathbb{X}_{i_n}^S | \mathbb{Z}_{i_1}^S \cdots \mathbb{Z}_{i_n}^S).$$

In conclusion, we have given the bits and pieces of the mechanism behind the Gottesman-Knill theorem.

1 **Gottesman-Knill theorem.** Any quantum computer performing only: 1) Clifford group gates (\mathbf{H}_j , \mathbf{S}_j , and $\mathbf{C}_{jj'}$), 2) measurements of Pauli string operators, and 3) Clifford group operations conditioned on classical bits, which may be the results of earlier measurements, can be perfectly simulated on a classical computer with **polynomial efforts** in the number of Qbits n and of applied gates.

This implies that quantum computation is possibly more powerful than classical computation only when it used gates outside of the Clifford group, for instance, $\mathbf{T} = \text{diag}(1, e^{i\pi/4})$ -gates.

Bibliography

- [1] N David Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007.
- [2] Giuliano Benenti, Giulio Casati, Davide Rossini, and Giuliano Strini. *Principles of Quantum Computation and Information: A Comprehensive Textbook*. World Scientific, 2018.
- [3] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [4] Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell’s Inequalities. *Phys. Rev. Lett.*, 49:91–94, Jul 1982.
- [5] Gregor Weihs, Thomas Jennewein, Christoph Simon, Harald Weinfurter, and Anton Zeilinger. Violation of Bell’s Inequality under Strict Einstein Locality Conditions. *Phys. Rev. Lett.*, 81:5039–5043, Dec 1998.
- [6] Daniel Gottesman. The heisenberg representation of quantum computers. 1998.
- [7] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, Nov 2004.
- [8] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, third edition, 2006.
- [9] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, 1997.
- [10] Hannes Bernien, Sylvain Schwartz, Alexander Keesling, Harry Levine, Ahmed Omran, Hannes Pichler, Soonwon Choi, Alexander S. Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Probing many-body dynamics on a 51-atom quantum simulator. *Nature*, 551(7682):579–584, 2017.
- [11] Sepehr Ebadi, Tout T. Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, Soonwon Choi, Subir Sachdev, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595(7866):227–232, 2021.
- [12] D. Bluvstein, A. Omran, H. Levine, A. Keesling, G. Semeghini, S. Ebadi, T. T. Wang, A. A. Michailidis, N. Maskara, W. W. Ho, S. Choi, M. Serbyn, M. Greiner, V. Vuletić, and M. D. Lukin. Controlling quantum many-body dynamics in driven rydberg atom arrays. *Science*, 371(6536):1355–1359, 2021.
- [13] Pascal Scholl, Michael Schuler, Hannah J. Williams, Alexander A. Eberharter, Daniel Barredo, Kai-Niklas Schymik, Vincent Lienhard, Louis-Paul Henry, Thomas C. Lang, Thierry Lahaye, Andreas M. Läuchli, and Antoine Browaeys. Quantum simulation of 2d antiferromagnets with hundreds of rydberg atoms. *Nature*, 595(7866):233–238, 2021.

- [14] Vincent Jacques, E Wu, Frédéric Grosshans, Franccois Treussart, Philippe Grangier, Alain Aspect, and Jean-Franccois Roch. Experimental realization of wheeler’s delayed-choice gedanken experiment. *Science*, 315(5814):966–968, 2007.
- [15] Yoon-Ho Kim, Rong Yu, Sergei P. Kulik, Yanhua Shih, and Marlan O. Scully. Delayed “choice” quantum eraser. *Phys. Rev. Lett.*, 84:1–5, Jan 2000.
- [16] Tabish Qureshi. Demystifying the delayed-choice quantum eraser. *European Journal of Physics*, 41(5):055403, aug 2020.
- [17] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer’s guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019.
- [18] Marc Mézard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.
- [19] M. N. Vyalyi A. Yu. Kitaev, A. H. Shen. *Classical and quantum computation*. Graduate studies in mathematics 47. American Mathematical Society, 2002.
- [20] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997.
- [21] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv e-prints*, page arXiv:1411.4028, 2014.
- [22] Glen Bigan Mbeng, Rosario Fazio, and Giuseppe E Santoro. Quantum annealing: a journey through digitalization, control, and hybrid quantum variational schemes. *arXiv preprint arXiv:1906.08948*, 2019.
- [23] Zhang Jiang, Eleanor G. Rieffel, and Zhihui Wang. Near-optimal quantum circuit for grover’s unstructured search using a transverse field. *Phys. Rev. A*, 95:062317, Jun 2017.
- [24] Matteo M. Wauters, Glen B. Mbeng, and Giuseppe E. Santoro. Polynomial scaling of the quantum approximate optimization algorithm for ground-state preparation of the fully connected p -spin ferromagnet in a transverse field. *Phys. Rev. A*, 102:062404, Dec 2020.
- [25] Robert B. Griffiths and Chi-Sheng Niu. Semiclassical fourier transform for quantum computation. *Phys. Rev. Lett.*, 76:3228–3231, Apr 1996.
- [26] César Miquel, Juan Pablo Paz, and Roberto Perazzo. Factoring in a dissipative quantum computer. *Phys. Rev. A*, 54:2605–2613, Oct 1996.
- [27] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998.
- [28] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
- [29] Asher Peres. *Quantum theory: concepts and methods*, volume 57. Springer Science & Business Media, 2006.
- [30] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, 2014. Theoretical Aspects of Quantum Cryptography — celebrating 30 years of BB84.
- [31] David Elkouss, Jesús Martínez-Mateo, and Vicente Martin. Information reconciliation for QKD. *Quantum Inf. Comput.*, 11(3&4):226–238, 2011.

-
- [32] Artur K. Ekert. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.
- [33] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.
- [34] Alexandre Blais, Arne L. Grimsmo, S. M. Girvin, and Andreas Wallraff. Circuit quantum electrodynamics. *Rev. Mod. Phys.*, 93:025005, May 2021.
- [35] Uri Vool and Michel Devoret. Introduction to quantum electromagnetic circuits. *International Journal of Circuit Theory and Applications*, 45(7):897–934, 2017.
- [36] B.D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251–253, 1962.
- [37] P. G. de Gennes. *Superconductivity of Metals and Alloys*. Benjamin, New York, 1966.
- [38] Michael Tinkham. *Introduction to superconductivity*. Courier Corporation, 2004.
- [39] V Bouchiat, D Vion, P Joyez, D Esteve, and M H Devoret. Quantum coherence with a single cooper pair. *Physica Scripta*, 1998(T76):165, jan 1998.
- [40] Y. Nakamura, Yu. A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Nature*, 398(6730):786–788, 1999.
- [41] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Phys. Rev. A*, 76:042319, Oct 2007.
- [42] Fei Yan, Philip Krantz, Youngkyu Sung, Morten Kjaergaard, Daniel L. Campbell, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. Tunable coupling scheme for implementing high-fidelity two-qubit gates. *Phys. Rev. Appl.*, 10:054062, Nov 2018.
- [43] Yu Chen, C. Neill, P. Roushan, N. Leung, M. Fang, R. Barends, J. Kelly, B. Campbell, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, A. Megrant, J. Y. Mutus, P. J. J. O'Malley, C. M. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, Michael R. Geller, A. N. Cleland, and John M. Martinis. Qubit architecture with high coherence and fast tunable coupling. *Phys. Rev. Lett.*, 113:220502, Nov 2014.
- [44] Heinz-Peter Breuer, Elsi-Mari Laine, Jyrki Piilo, and Bassano Vacchini. Colloquium: Non-markovian dynamics in open quantum systems. *Rev. Mod. Phys.*, 88:021002, Apr 2016.
- [45] C. Cohen-Tannoudji, J. Dupont-Roc, and G. Grynberg. *Atom-Photon Interactions: Basic Processes and Applications*. John Wiley & Sons, 1992.
- [46] G. Lindblad. On the generators of quantum dynamical semigroups. *Commun. Math. Phys.*, 48:119, 1976.
- [47] Li, C.-F., Guo, G.-C., and Piilo, J. Non-markovian quantum dynamics: What does it mean? *EPL*, 127(5):50001, 2019.
- [48] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, 2019.
- [49] Dave Bacon. Introduction to quantum error correction. In Daniel A Lidar and Todd A Brun, editors, *Quantum error correction*, chapter 2, pages 46–76. Cambridge University Press, Cambridge, 2013.

- [50] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier Science, 1978.
- [51] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996.
- [52] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [53] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.
- [54] John B. Kogut. An introduction to lattice gauge theory and spin systems. *Rev. Mod. Phys.*, 51:659–713, Oct 1979.
- [55] P. Gaspard and M. Nagaoka. Slippage of initial conditions for the Redfield master equation. *J. Chem. Phys.*, 111:5668, 1999.
- [56] Heinz-Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2002.
- [57] Gernot Schaller. *Open Quantum Systems Far from Equilibrium*, volume 881. 11 2013.
- [58] A.O. Caldeira and A.J. Leggett. Path integral approach to quantum brownian motion. *Physica A: Statistical Mechanics and its Applications*, 121(3):587 – 616, 1983.
- [59] A.O Caldeira and A.J Leggett. Quantum tunnelling in a dissipative system. *Annals of Physics*, 149(2):374 – 456, 1983.
- [60] T. Albash, S. Boixo, D. A. Lidar, and P. Zanardi. Quantum adiabatic Markovian master equations. *New J. Phys.*, 14:123016, 2012.
- [61] M. Grifoni and P. Hänggi. Driven quantum tunneling. *Physics Reports*, 304:229–354, 1998.
- [62] Angelo Russomanno, Stefano Pugnetti, Valentina Brosco, and Rosario Fazio. Floquet theory of cooper pair pumping. *Phys. Rev. B*, 83:214508, Jun 2011.
- [63] Alexander Shnirman, Yuriy Makhlin, and Gerd Schön. Noise and decoherence in quantum two-level systems. *Phys. Scr.*, T102:147–154, 2002.
- [64] J.R. Johansson, P.D. Nation, and Franco Nori. QuTiP: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, aug 2012.
- [65] J.R. Johansson, P.D. Nation, and Franco Nori. QuTiP 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240, apr 2013.