

# QUANTUM COMPUTATION

Summer Semester 2022

Falk Eilenberger, Institute of Applied Physics, Friedrich Schiller University, Jena

Fabian Steinlechner, Fraunhofer-Institute for Applied Optics and Precision Engineering IOF, Jena

<b>1</b>	<b>Algorithms and Complexity .....</b>	<b>3</b>
1.1	Turing Machines and Universal Algorithmic Devices .....	4
1.2	Computational Complexity and Scaling Behavior .....	7
1.3	The Strong Church-Turing Hypothesis and Path to Quantum Computers .....	10
1.4	Definition of a Quantum Computer .....	12
<b>2</b>	<b>Fundamentals of Quantum Physics.....</b>	<b>12</b>
2.1	A Somewhat Physical Introduction to Quantum Physics .....	12
2.2	The Postulates of Quantum Theory .....	13
2.3	Matrix representations .....	23
2.4	Mixed States and the density matrix .....	25
<b>3</b>	<b>From Single Qubits to Circuits.....</b>	<b>28</b>
3.1	The Qubit .....	28
3.2	The Bloch Sphere .....	30
3.3	Single Qubit Gates, Rotations, Universality.....	31
3.4	Observables and the Pauli-Matrices .....	34
3.5	Mixed Single-Qubit States.....	35
3.6	The Circuit Representation .....	37
<b>4</b>	<b>Multiple Qubits, Entanglement, and Universality .....</b>	<b>39</b>
4.1	Two-Qubit States and Entanglement.....	40
4.2	Controlled Operations on a single Qubit .....	45
4.3	Classic Computation on a Quantum Computer.....	51
4.4	Generic Operations and Universality .....	54
<b>5</b>	<b>Quantum Algorithms .....</b>	<b>58</b>
5.1	Josza-Deutsch's Algorithm: a Case of Useless but Powerful.....	58
5.2	Quantum Fourier Transformation: Divide et Conquera.....	63
5.3	Quantum Phase Estimation: Eigenvalue where Art Thou?.....	67
5.4	Shor's Algorithm: The Internet will Hate You .....	72
5.5	Grover's Algorithm: Whacking the Oracle.....	80
<b>6</b>	<b>Quantum Galore .....</b>	<b>86</b>
<b>7</b>	<b>Alternative Computational Models.....</b>	<b>88</b>
7.1	Measurement-based Quantum Computing .....	88
7.2	One-Way Quantum Computing .....	92
<b>A 1</b>	<b>The No-Cloning Theorem.....</b>	<b>97</b>

# 1 Algorithms and Complexity

This part of the lecture is not yet concerned with quantum physics or quantum computation. Instead, it shall serve as an introduction to some of the minimum parts information science, which we shall require in order to help understand, why and in what areas quantum computers are actually useful.

These aspects shall also serve as a reminder that there is an intrinsic connection of the physical world and the computer world, which is all too easily forgotten with contemporary computer systems. After all, computer and thus algorithms must use physical effects to work their magic and as such they must represent information in physical entities. So let's get started.

At the centre of information science is the concept of an algorithm. Consider an algorithm to be the equivalent of a cooking recipe:

*Definition 1: An algorithm contains specific set of procedures to carry out with a set of specific resources to solve a specific problem / to COMPUTE the solution to a specific problem.*

In the case of a cooking recipe this would be to turn shoppable ingredients into a tasty meal or to impress your guests. Ideally both. Algorithms are ubiquitous in our civilization (with the aforementioned cooking being a – we believe – very down to earth example) and it is no wonder that there is a fair amount of them floating around in mathematics; some of which have been invented by the old greeks or even further back. Probably the first one you get to learn in school is the addition of two large numbers, which you learn to break down into digit-wise addition of number smaller than ten. Just in case you like to be reminded of the good of time in elementary school, when life was simple, as long as you could avoid the ubiquitous school yard bully, I have sketched the algorithm carried out for you. The result is, of course, 4242, because....can there be any other meaningful result?

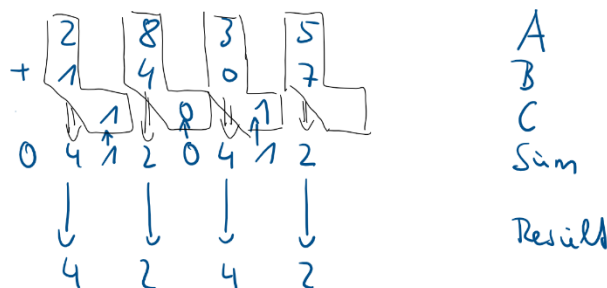


Figure 1: Addition of two large numbers. This is probably the first mathematical algorithm you learn in school, that deserves a closer look at.

It is no wonder then, that algorithms in by itself became the subject of scientific research, which lead to a series of discoveries, which date back to the invention of the first computers. The reason is quite straightforward: with the invention of electronic computers the computation capabilities of mankind skyrocketed (and it still is), which immediately led to a very simple question: given a computation machine of sufficient size and speed, can we compute everything? More specifically:

1. Is there a (class of) computational machine, which can run any kind of conceivable algorithm?
2. And if so, can we do so in any kind of limited time, or is this hopeless to begin with?

Spoiler alert: the answer to the first question is a resounding: yes, whereas the answer to the second question is a mildly disappointing: very frequently no.

## 1.1 Turing Machines and Universal Algorithmic Devices

Let's at first turn our attention to the first question. It certainly is helpful to formalize the description of an algorithm to, to help give the kind of clarity, which we need to analyse its features. Let's go back to the summation example of above and try to formalize the produce of the addition of large numbers into a kind of pseudo-code:

### Algorithm to calculate A+B

1. Write numbers A and B beneath each other onto a sheet of paper, with the digits aligned (e.g. the  $10^0$  under each other and the  $10^1$  under each other and so forth)
2. Investigate the least significant digits and introduce an auxiliary variable C, which is set to zero.
3. Add the two digits under investigation from numbers A and B and the auxiliary variable C.
4. Write down the least significant digit of the sum under the digit under investigation.
5. If the sum was ten or larger set the auxiliary variable C to value 1, else to zero.
6. Move to the next most significant digit and repeat, starting from point 3.
7. If you have run out of digits, check your auxiliary. If it is zero, then terminate. If it is one, then write a 1 in front of the result and terminate.

This is already quite formal but we aim to take this one step further and design a hypothetical machine out of this pseudo-code, because this lends itself much better for analysis than the pseud-code representation, which is still too close to natural language, to fit into a convenient mathematical apparatus. The most well-known and well-investigated of this type of machines is the so-called Turing Machine.

*Definition 2: A Turing-Machine is a set of four elements, namely:*

- (1) A finite state control  $Q = \{q_1, \dots, q_S\}$ , defining all possible states  $q_s$  of the TM and a current state  $q \in Q$ .  
*The set of states  $Q$  has a minimum number of two members, namely  $q_{s_0}$ , the starting state and  $q_h$ , the halting state. The machine is initially in  $q_{s_0}$ . If it reaches state  $q_h$  is has finished its calculation.*
- (2) A semi-infinite tape  $S = \{s_1, \dots, s_p\}$ , which consists of a numbered sequence of elements  $s_p$  called tape squares, where the individual elements belong to a set of symbols of an alphabet  $s_p \in \Gamma$ .  
*The alphabet  $\Gamma$  usually has a minimum number of four symbols, namely 0,1,b, start. The symbol start is reserved to indicate the beginning of the tape, b is for blank elements of the tape.*
- (3) A read/write head, which is pointing at a specific position  $p'$  of the tape. The write head can be used to read the symbol  $s_{p'}$  off the tape and to overwrite its content with any one of the symbols in  $\Gamma$ .
- (4) A program table, consisting of a sequence of program lines of the form  $\langle q, s, q', s', m \rangle_l$ , where  $l$  is the number of the program line, with  $l \in \{1, \dots, L\}$ . Here  $q, q' \in Q$ ,  $s, s' \in \Gamma$ , and  $m \in \mathbb{Z}$ . The lines are unique in the sense that there exists no more than one line for each combination of  $\langle q, s, \dots \rangle$

All notes subject to change, no guarantee to correctness, corrections welcome.

The TM is initialized to be in the starting state  $q_{s_0}$  and at tape position  $p = 0$ . At every iteration the state  $q$  and tape value  $s$  is checked. If there exists no element within the program table with this specific combination  $\langle q, s, \dots \rangle$  then the TM is set onto state  $q_h$  and the program is finished. If there exists a line then the state is change according to  $q \rightarrow q'$ , the value of the tape is changed according to  $s \rightarrow s'$  and the read/write head position  $p$  is changes according to  $p \rightarrow p + m$ . The process is repeated until the machine is halted.

Of course, this is very abstract so let's try and turn our addition algorithm into a TM. For the sake of simplicity, we shall, however, change the notation of number into binary and we shall fix the number of digits for both  $A$  and  $B$  to be eight. We shall also adopt only eight bits for the result and thus we will in reality calculate  $C = (A + B) \bmod 256$ . Thus we have:

- For the state control we have  $Q = \{q_{s_0}, q_h, R_{A0}, R_{A1}, R_{B0}, R_{B1}, R_{B2}, W_0, W_1, W_2, W_3\}$
- For the alphabet of the tape  $\Gamma = \{0, 1, b, start\}$
- The tape is initialized as follows:  $\langle s, A_0, \dots, A_7, b, B_0, \dots, B_7, b, \dots \rangle$ , where  $A_i$  and  $B_i$  are the binary digits of  $A$  and  $B$  in big endian notation, respectively.
- The program table is as follows:

$q$	$s$	$q'$	$s'$	$m$
$q_{s_0}$	start	$R_{A0}$	start	+1
$R_{A0}$	0	$R_{B0}$	0	+9
$R_{A0}$	1	$R_{B1}$	1	+9
$R_{A0}$	$b$	$q_h$	$b$	0
$R_{A1}$	0	$R_{B1}$	0	+9
$R_{A1}$	1	$R_{B2}$	1	+9
$R_{A1}$	$b$	$q_h$	$b$	0
$R_{B0}$	0	$R_{W0}$	0	+9
$R_{B0}$	1	$R_{W1}$	1	+9
$R_{B1}$	0	$R_{W1}$	0	+9
$R_{B1}$	1	$R_{W2}$	1	+9
$R_{B2}$	0	$R_{W2}$	0	+9
$R_{B2}$	1	$R_{W3}$	1	+9
$R_{W0}$	$b$	$R_{A0}$	0	-17
$R_{W1}$	$b$	$R_{A0}$	1	-17
$R_{W2}$	$b$	$R_{A1}$	0	-17
$R_{W3}$	$b$	$R_{A1}$	1	-17

A typical layout of the band after the machine has run may look like this:

start	0	1	0	1	0	1	0	1	<b>b</b>	1	1	1	1	0	0	0	0	<b>b</b>	1	0	0	1	1	1	0	1	<b>b</b>
-------	---	---	---	---	---	---	---	---	----------	---	---	---	---	---	---	---	---	----------	---	---	---	---	---	---	---	---	----------

Please feel free to do the back conversion into decimal numbers yourself or believe me, that the algorithm has just calculated  $170 + 15 = 185$  for you. Also note that we have just marked the blanks  $b$  in boldface to make the result a bit easier to read. The first two part of the band still contain the numbers  $A$  and  $B$  in their initial form and the result is in the third block of the band.

A few things to note here are :

- we are using the state of the TM  $q$  to store intermediate results; this is quite cumbersome but does the job
- an alternative approach is to use the tape itself to store intermediate results
- the TM notation is quite cumbersome and it was never intended as a programming language but as a tool to ponder on algorithms and programming schemes

All notes subject to change, no guarantee to correctness, corrections welcome.



and any modern computer that we operate nowadays is an almost (!) perfect example for such a machine. Therefore, using a computer, we can, in principle, calculate the result to any type of computation problems, whose solution can be attained algorithmically.

“In principle” here is, however, an important caveat. Real computers have finite memory and, of course, there is a finite amount of time that we have to calculate any solution to a problem, or else we may end up like the investors of the great thinking machine in the “Hitchhiker’s Guide to the Galaxy”, long extinct, before the program has finished.

This leads back to the second question from above, which we shall discuss in detail in the following section. To do so, we shall, however, clarify this question by breaking it up into three subquestions:

- 2a. Can we solve any type of algorithmic problem in finite time?
- 2b. Can we solve any type of algorithmic problems in efficiently? E.g. can we solve any algorithmic problem in such a way that the resources (time, memory) do not grow faster than polynomial for any given increase in the problems parametric complexity?
- 2c. Are TMs an efficient model for computation? I.e. can TMs simulate any conceivable algorithmic machine efficiently? I.e. are TMs the per-se most efficient type of computational machine?

The answer to question (2a) is a resounding NO. This was already found in the early days of information science, when it became clear that there are classes problems that cannot be solved in finite time by a TM or any other computation device. The first and most prominent example is Hilberts “Entscheidungsproblem”. We have added to the number of such problems in the meantime. Thus, we know that there are problems which are intrinsically hard to solve in and by themselves. About everything we know surprisingly little and what we know is surprisingly circumstantial and unsystematic.

## 1.2 Computational Complexity and Scaling Behavior

Question 2b is probably the one with the most unsatisfactory answer: we just don’t know. But before we go into any level of detail here, let’s just reconsider, how such a question may be answered at all, e.g. what does an efficient solution actually mean? Suppose for any given class of problem there is a order-parameter  $N$ , which describes the size of the problem. For the summation algorithm from above this could, e.g. be the size of the numbers to be added. The concept of computation complexity then describes the asymptotic scaling behaviour of the computational resources (time, memory space, energy) to be required for a solution, as the order parameter scales to large  $N \rightarrow \infty$ .

For the summation algorithm from above it should be clear that we have the scaling behaviour  $\mathcal{O}(\log(N))$ ; e.g. if the numbers to be added grow by a factor of 10 we just have to carry out one more addition step. This seems like a pretty efficient algorithm; particularly if you compare it with the more straightforward approach of addition by counting (e.g. adding with your fingers). This approach would scale according to  $\mathcal{O}(N)$  and therefore much less efficiently.  $\mathcal{O}(N)$  being worse than  $\mathcal{O}(\log(N))$  is of course only strictly true for large numbers and this might be the reason that first graders, who only operate on fairly small numbers, might be tempted to hone their skills in the addition by counting algorithm, instead of learning digit-wise addition (you see, I have a small kid in school).

The digit-wise addition with its  $\mathcal{O}(\log(N))$  scaling then the most efficient algorithm there is? Is not turns out, probably yes. But already for the textbook-style multiplication of integers the questions becomes much more complicated. Be  $n = \log(N)$  the approximate number of digits of the two numbers to be multiplied. Then the textbook multiplication scales according to  $\mathcal{O}(n^2)$ . From a deeper analysis of the problem, however, we know from first principles (top-down) that the problem in and by itself

must be solvable in  $\mathcal{O}(n \log n)$  steps. Indeed in 2019 such an algorithm was demonstrated by Harvey and Hoeven, but its validity proof rely on at least one unproven (but likely) conjecture, so....make of this what you like. At any rate for the purpose of this course we shall assume the bottom-up (algorithmic) scaling for integer multiplication has met with the top-down scaling, which is a nice result.

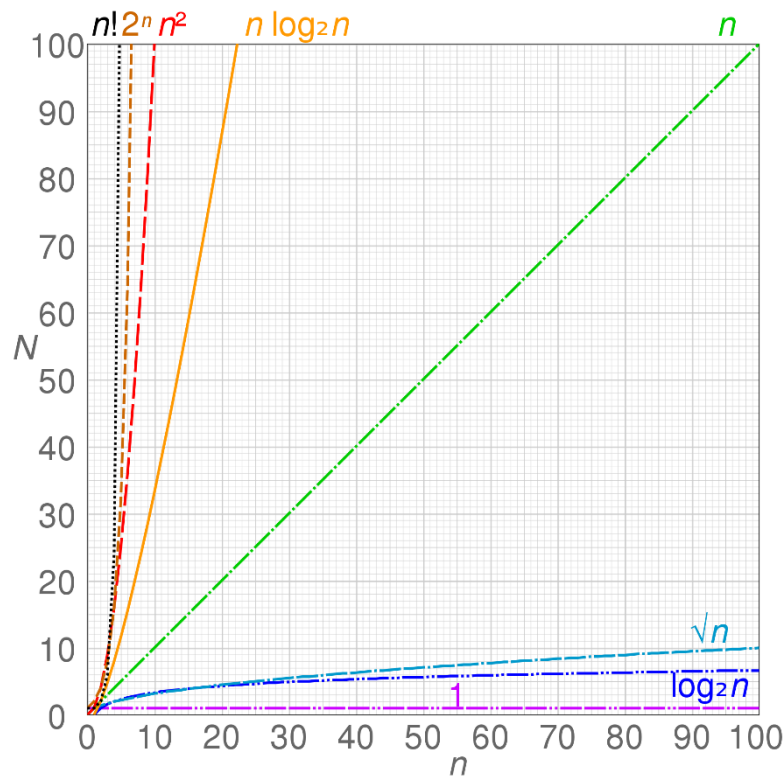


Figure 3: Illustration of the scaling of different types of  $\mathcal{O}$ . Everything above the green line is pretty bad. For everything above the brown line, even Moore's law is no consolation. Stolen from the QISKIT book.

Integer multiplication is a seemingly simple problem and the discussion already points at a fundamental problem. For any given problem we do probably have a set of algorithms with a specific scaling (bottom-up). However, the cases, where we know from first principles (top down), what the best possible scaling is, are rare. So, the question for many computation problems remains: is there a much better algorithm out there? This question is aggravated by the fact that we in fact know many algorithms with appallingly bad scaling, such as  $\mathcal{O}(\exp n)$  or  $\mathcal{O}(n!) = \mathcal{O}(n^n)$ . Many of such problems are related to field of information science of high impact, such as graph problems (frequently encountered in database and optimization problems) or the simulation of many particle systems in quantum physics.

Of course, the search for a best possible scaling for computational problems is a big thing, because it promises algorithmic speedup beyond the power of the scaling of hardware. Therefore, scientists have not been idle, and they have come up with a zoo of interesting results in this direction. We shall first discuss some results from the top-down perspective and then switch to the bottom-up perspective in the next section.

The most successful approach in top-down analysis of problems is the grouping of problems into complexity classes. A problem  $P'$  is said to be in the same complexity class as another problem  $P$ , if  $P'$  can be reduced onto the problem  $P$  with no more than polynomial complexity. Complexity classes according to this definition are rather large things and a lot of conjectures about their mutual relations are known, which are usually formulated in the concept of mathematical languages. We are not going into details, but we will just discuss the two three relevant classes here:



- **P**: Is the class of problems, which can be solved deterministically in polynomial time.
- **NP**: is the class of problems, for which solutions can be verified deterministically in polynomial time but there is not necessarily a possibility to find solutions in polynomial time.
- **NP-hard**: is a subclass of problems in **NP**, onto which all **NP**-problems can be reduced

As a remark: please ignore the word “deterministically” here. We shall get back to it at the beginning of the next chapter.

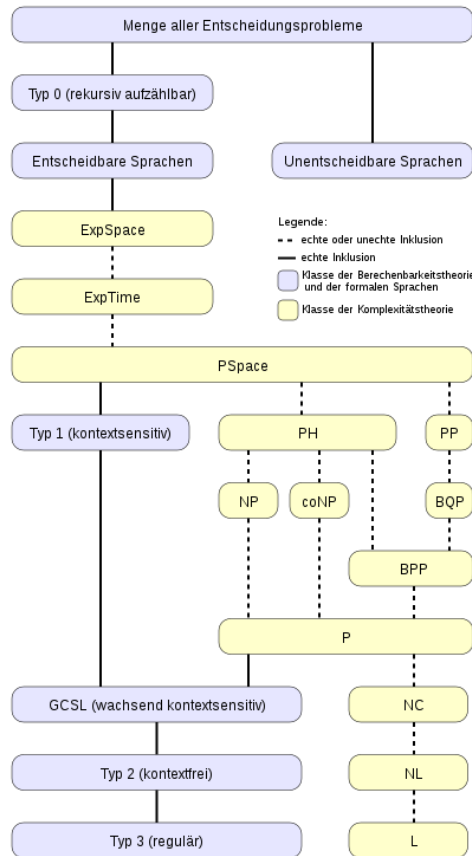


Figure 4: A few complexity classes and what Wikipedia knows about their mutual relations.

Of course, it is trivial to see:  $P \subseteq NP$ . The question however remains is  $P = NP$  or  $P \subset NP$ ? This question is usually approached by introducing a sub-classification into the **NP** class, namely the **NP-hard**, class. **NP-hard** problems is a set of problems onto which every **NP** problem can be reduced. The most famous of which is the so-called “boolean satisfiability problem”. The  $P = NP$  question can then be reduced to the following two problems: (bottom-up) Can we find any single **NP-hard** problem, which is solvable in polynomial time? If so, then  $P = NP$ . (top-down) if we can, however, show that such a such an algorithm cannot exist, then  $P \subset NP$ . The latter problem can be considered as the holy grail of information science and there is – to this date – no solution. There is – however – also no bottom-up solution, e.g. no algorithm, which can be run on a TM-complete computer and which can solve **NP-hard** problems. After 80 years, or so, of computer science with Turing-complete system since may serve as a strong hint, that either of the following explanations is true:

- Explanation 1: there is no such solution and indeed  $P \subset NP$ , i.e. there are problems, which will forever remain hard to solve but easy to verify.
- Explanation 2: the TM-model although universal may not be universally efficient.

We have now set the stage for the quantum computer.

## 1.3 The Strong Church-Turing Hypothesis and Path to Quantum Computers

There are two reasons to suspect that Explanation 2, might be worthwhile to investigate. The first reason is related to the word “deterministically”, which we had asked you to ignore in the last chapter. In fact, people have discovered quite early, that many  $NP$ -problems can be solved efficiently on a probabilistic TM (i.e. a TM with an added random number generator), if we allow for a margin of error in our solution (e.g. we may get a wrong solution with an arbitrary probability  $\varepsilon \ll 1$ ). One example is the travelling salesman-problem: here a deterministic solution has a scaling of  $\mathcal{O}(n!)$  but we can (with certainty) get to within a factor of 1.5 to the best solution within  $\mathcal{O}(\text{poly}(n))$  using e.g. the so-called algorithm of Christofides.

In other words: we know that probabilistic Turing Machines are much more efficient at problem solving than ordinary TMs and thus people have come up with the Strong Church Turing Thesis as a consequence:

*Theorem 2 (Strong Church-Turing-Thesis): Any model of computation can be simulated on a probabilistic Turing Machine with at most polynomial increase (i.e. efficiently) in the number of elementary operation required.*

And that's it. At least from the point of view of the first half of the 20<sup>th</sup> century. Because, what else would you add to a Turing Machine? What else is there to add? Of course, this cannot be true, because otherwise we would not make such a hype of Quantum Computer you would not be reading this script, right?

The first serious cracks in the strong CTT are typically attributed to our most favourite Richard Feynman and a few of his lectures in and around 1982. There he elaborated on the notion, that:

*Can physics be simulated by a universal computer? [...] the physical world is quantum mechanical, and therefore the proper problem is the simulation of quantum physics [...] the full description of quantum mechanics for a large system with  $R$  particles [...] has too many variables, it cannot be simulated with a normal computer with a number of elements proportional to  $R$  [...] but it can be simulated with ] quantum computer elements. [...] Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? [...] If you take the computer to be the classical kind I've described so far [...] the answer is certainly, No!*

*Richard Feynman (1980)*

What is he actually referring to? As it turns out many-particle quantum systems are intrinsically hard to simulate, because each particle (e.g. electron, proton, etc...) lives in its “own” version of (three-dimensional) space; all of which interact. If you have  $R$  particles and discretise space into  $n$  points it turns out that each simulation step will require at least  $\mathcal{O}(n^R)$  data points. Thus quantum many-particle systems are incredibly hard to handle in a classical computer.

Although Feynman is certainly very famous, his ideas (worries?) had been independently formulated by a few others before:

*Perhaps [...] we need a mathematical theory of quantum automata. [...] the quantum state space has far greater capacity than the classical one: for a classical system with  $N$  states, its quantum version allowing superposition accommodates  $c^N$  states. When we join two classical systems, their number of states  $N_1$  and  $N_2$  are multiplied, and in the*

*quantum case we get the exponential growth  $c^{N_1 N_2}$ . [...] These crude estimates show that the quantum behavior of the system might be much more complex than its classical simulation.*

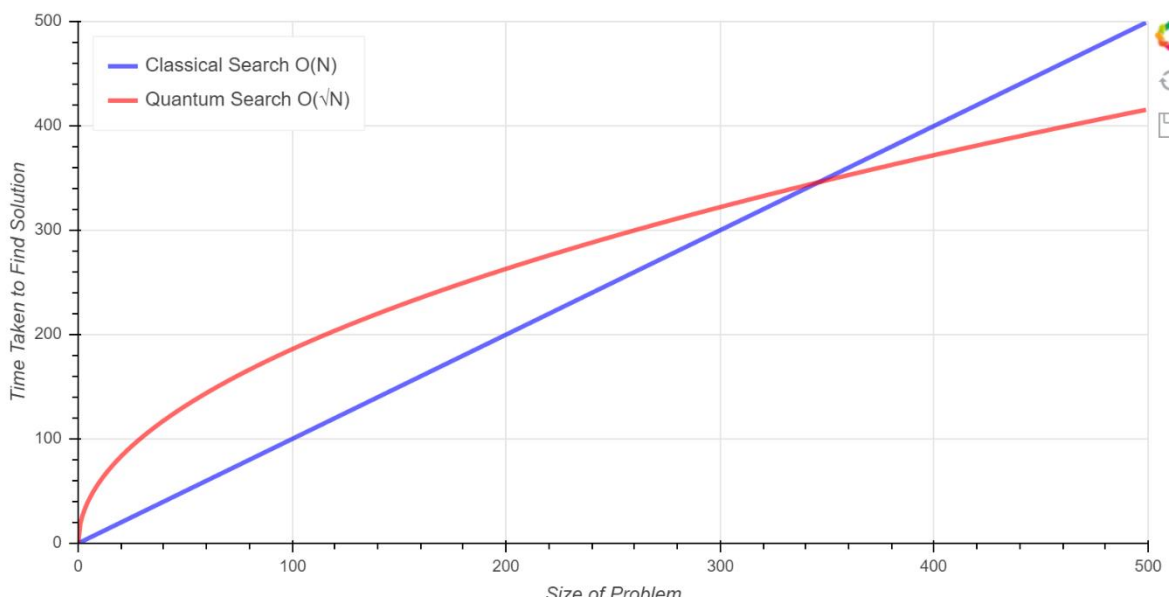
*Yu Manin (1980)*

and even earlier:

*The quantum-mechanical computation of one molecule of methane requires  $10^{42}$  grid points. Assuming that at each point we have to perform only 10 elementary operations, [...] we would still have to use all the energy produced on Earth during the last century [for its simulation].*

*R. P. Poplavskii (1975)*

So, what do we make of this? We could certainly just give up and say: quantum systems will forever remain unsimulatable but we could also choose a more pragmatic approach. We could, as we say in German “Den Bock zum Gärtner machen”. What I am trying to say is: if nature has bequeathed us with a class of physical systems, whose behaviour we understand but which we cannot predict in detail, because the systems complexity is intractably large, then, why should we not attempt to use these systems to make predictions for us? Why should we not try to build a computer, which operates on a quantum many particle system, to solve quantum mechanical problems, which we cannot solve on classical computers? If that is possible, what other algorithmic problems can such a computer solve efficiently, that we cannot solve on a Turing machine efficiently? Are there any such problems at all? And how can we find them?



*Figure 5: A case for Quantum Computers. Even a modest improvement from  $\mathcal{O}(n)$  to  $\mathcal{O}(\sqrt{n})$ , as experienced for quantum search algorithms will outperform a classical computer if the search space is sufficiently big, even if the quantum computer is much, muuuuuch slower. Stolen from the QISKIT book.*

We shall of course see in the following chapter, that all of these questions can be answered positively with some level of confidence. This also means, that the Strong CTT is definitely wrong. Moreover, it points to the fact that Theses such as the Strong CTT should not be written down in ignorance of the limitations of the physical systems, on which our models of computation are based. Or in other words: if any new physical theory, more fundamental than Quantum Physics is discovered, go looking for the

underlying problems, which are hard to simulate and see if you can make a new and powerful class of computers from it.

## 1.4 Definition of a Quantum Computer

Since every computational system is ultimately described by quantum physics, we also need to define, what we mean, when we say a “quantum computer”. What separates a quantum computer from a classical computer, is defined at the operational resource level:

*Definition 3: A quantum computer is a computational device, which uses quantum information (frequently but not necessarily in the form of a set of Qubits) to perform algorithmic tasks, using quantum processes which are not accessible to classical systems.*

While state-of-the-art classical computers may well leverage quantum technology at the level of hardware (lasers, semiconductor technology, photodetection), they do not take advantage of quantum principles at the level of information or processing itself. Not yet.

Now that we have convinced you that quantum computers are a hot topic, because they operate on new physical principles, let's delve into quantum physics and see what these principles are.

## 2 Fundamentals of Quantum Physics

Before we can understand quantum computers, we must first understand (some basics) of quantum physics. What is *quantum* physics? To put it simply: quantum physics is a theoretical framework, which describes the behaviour of everything in the world, except for gravity. More specifically, quantum theory provides a set of tools for calculating **probabilities for outcomes of measurements**<sup>1</sup> applied to a certain state of the quantum system to be measured. A measurement corresponds to anything we may observe in a laboratory using a suitable measurement apparatus. Mathematically such an apparatus is represented by a so-called observable. The toolset for its description comes in the form of postulates, which are discussed below.

### 2.1 A Somewhat Physical Introduction to Quantum Physics

This definition is as broad as it is useless, so for the sake of simplicity, we discuss some of the key ingredients. A central role in quantum physics is described by the notion of modes. If you are coming from a classical field theory (e.g. electrodynamics), these modes carry over to the quantum world without any change in the way they are calculated (e.g. there is either an eigenmode equation or a Hamiltonian from which they are derived). The difference is that the modes do not have a scalar excitation strength (e.g. modal amplitude), instead they are populated by a series of discrete states, starting from the vacuum  $|\text{vac}\rangle$ . These states are what is typically considered a quantum; they get their specific name from the type of field they describe, usually ending with an “-on”, such as photon for the electric field (also: electron, proton, phonon, etc...).

Mathematically the population of a mode  $j$  with quanta is done by the repeated operation of a creation operator  $\hat{a}_j^\dagger$  on the vacuum. For example,  $\hat{a}_j^\dagger |\text{vac}\rangle$  is a field which has one and only one quantum in the  $j$  mode. Here  $j$  is a quantum number uniquely denoting a specific mode, e.g. a  $\mathbf{k}$ -vector and a polarization such as  $H$  or  $V$ . The depopulation is similarly done by the annihilation operator  $\hat{a}_j$ , which is the Hermitian conjugate of the creation operator. Both operators are related to a complex superposition of the canonical fields and canonical momenta (e.g. a complex superposition of the electric and

---

<sup>1</sup> And nothing more. If you find that non-satisfactory, then deal with it. We shall later see that this is not a problem of the theory but the very essence of nature itself, as can be tested in e.g. a Bell-Test.

magnetic fields or the superposition of wave packet location and momentum). Depending on the kind of field, which is described these operators have different commutators. For boson fields we have:

$$[\hat{a}_i, \hat{a}_j] = [\hat{a}_i^\dagger, \hat{a}_j^\dagger] = 0, [\hat{a}_i, \hat{a}_j^\dagger] = \hat{a}_i \hat{a}_j^\dagger - \hat{a}_j^\dagger \hat{a}_i = \delta_{ij} \quad (1)$$

Whereas for fermion fields we have:

$$\{\hat{a}_i, \hat{a}_j\} = \{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0, \{\hat{a}_i, \hat{a}_j^\dagger\} = \hat{a}_i \hat{a}_j^\dagger + \hat{a}_j^\dagger \hat{a}_i = \delta_{ij} \quad (2)$$

For both types of fields we can construct a modal number operator

$$\hat{N}_j = \hat{a}_j^\dagger \hat{a}_j \quad (3)$$

which corresponds to an actual observable and which tells us exactly, how many quanta there are in a specific mode. We shall later see, what this actually means and how such a measurement can be constructed. The difference in the commutation relations for the two observables has the consequence that for bosonic fields any mode can have excitations with any positive integer number of bosons in them (e.g. the creation operators define an infinite ladder), whereas a fermionic mode can only have zero or one fermions in them. Keep in mind this actually means: it can have superposition of such quantum number states.

Each boson or fermion number state behaves very much like a classical mode, in the sense that its excitation is now described by a complex number, which is likewise called “amplitude”. The difference in classical field theories and quantum theory thus boils down to the fact that each field mode now consists of a series of quantum modes, which can each be excited by a complex numbered amplitude and superpositions thereof.

Any quantum system is in a superposition of these fundamental modal number states.

## 2.2 The Postulates of Quantum Theory

We shall now turn to the postulates of Quantum Physics, which describe how quanta evolve and how they are related to measurements. Why postulates? Well, it turns out that the rules of quantum physics cannot be derived from a more underlying theory (this may change, if, one day, quantum gravitation is developed). These rules have been derived from the results of many experiments and as such are laws of nature. In other words: the rules have been written down in a way as to be the simplest set of rules, which describe experiments. If this seems a little unsatisfactory to you, the opposite is true. It turns out they can be and have been used to describe gazillions of experimental observations with mind-numbing precision.

We will assume some level of familiarity with linear algebra and probability theory extensively throughout. The reader is encouraged to consult the standard quantum theory textbooks for a review if deemed necessary.

### 2.2.1 Quantum States and Superposition

*Postulate 1: Associated to any isolated physical system is a complex vector space  $\mathcal{H}$  with inner product  $\langle \phi | \psi \rangle = \langle \psi | \phi \rangle^* \in \mathbb{C}$  (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector  $|\psi\rangle$ , which is a unit vector in the system's state space, e.g.  $\langle \psi | \psi \rangle = |\psi|^2 = 1$ .*

The state vector  $|\psi\rangle$  represents a state of *complete knowledge* about the preparation of the physical system, i.e., everything that we need to know, and everything that is principle knowable. Implicit in

the structure of the linear vector space structure is the following statement: If  $|\psi_1\rangle$  and  $|\psi_2\rangle$  are possible quantum states, then so is any superposition state:

$$|\psi\rangle = \alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle \quad (4)$$

with complex amplitudes  $\alpha_1$  and  $\alpha_2$ . While this may look trivial, it is arguably among the most profound concept in quantum theory: the superposition principle is not only the culprit responsible for much quantum *weirdness*, such as the Heisenberg uncertainty principle, it is also the key feature in many quantum-enhancements such as exponential speedups in computing and secure communication.

Experimentally accessible quantities, such as expectation values and probabilities are described by numbers and not the state vectors themselves. Or to put it more bluntly: you cannot measure the state  $|\psi\rangle$  by any conceivable means (no matter how much money or brains you throw at the problem). To arrive at these, we need a mapping from vectors to numbers, i.e. an inner product. Denoting the dual vector to  $|\psi\rangle$  by the Dirac "bra":

$$\langle\psi| = |\psi\rangle^\dagger = \alpha_1^*\langle\psi_1| + \alpha_2^*\langle\psi_2| \quad (5)$$

The inner product can be written conveniently as a „bra-ket“:

$$\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^* \quad (6)$$

In particular, the norm of any vector is a real number  $\langle\psi|\psi\rangle > 0$  that of a permissible state is 1.

Just exactly what the state space of a quantum system is, is subject to quantum physics and must be treated in underlying theories. The awesome power of quantum physics is related to the fact that there is a huge set of physical systems, which behave this way, irrespective of their physical origin.

### Qubits

The simplest Hilbert-Space is a two-dimensional one. From the laws of linear Algebra we know, that within such a Hilbert space we may chose an orthonormal basis set, which we shall simply denote as

$$|0\rangle, |1\rangle \quad (7)$$

Thus, any state within this 2d Hilbert-Space can be written as a superposition according to:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (8)$$

With  $|\alpha|^2 + |\beta|^2 = 1$ . Such a system is called a Quantum Bit or in Short a QuBit. The notion comes from the idea, that a QuBit, just like an ordinary bit, can take the form of two-well defined states  $|0\rangle$  and  $|1\rangle$  but, as opposed to an ordinary QuBit, it can also take any superposition of such as state.

*Definition 4: A system, which can be described by a two-dimensional Hilbert-space is called QuBit. Any possible state within that system is a valid state of the QuBit. Physical implementations for QuBits are manyfold but details are irrelevant for the concept.*

### 2.2.2 Evolution

*Postulate 2: The evolution of a closed quantum system is described by a unitary transformation  $\hat{U}$ . That is, the state  $|\psi\rangle$  of the system at time  $t_1$  is related to the state  $|\psi'\rangle$  of the system at time  $t_2$  by a unitary operator  $\hat{U}$  (e.g.  $\hat{U}^\dagger = \hat{U}^{-1}$ ) which depends only on the times  $t_1$  and  $t_2$ , such that*

$$|\psi'\rangle = \hat{U}|\psi\rangle \quad (9)$$

While this has the status of a postulate, we may also discuss, why this is a plausible postulate. Unitary operators do not change the norm of a vector upon which it is applied, which can easily be seen by calculating the norm of  $\hat{U}|\psi\rangle$ :

$$\langle\psi'|\psi'\rangle = \langle\psi|\hat{U}^\dagger\hat{U}|\psi\rangle = \langle\psi|\psi\rangle \quad (10)$$

Just like the quantum states, we can't make any statement here, as to the specifics of the unitary evolution operator, which define the evolution of a real-world system. This is again subject to quantum physics and depends on the system in question. For many systems, however, there are external fields, which can be used to imprint evolution operators with specific properties. Such external fields may be laser beams, RF-pulses, Lorenz-forces, or anything else. These operators can often be thought of as acting in a time-discrete manner, i.e. they are active until a certain evolution of the state is achieved and then they are switched off. These operators will play a crucial role in quantum computers and there take the notion of a quantum gate, e.g. a discrete step in a computation algorithm that is used to manipulate the state of a quantum system.

A special role, however, is played by the free evolution operator, which, of course, acts in a time continuous manner:

*Postulate 2': The time evolution of the state of a closed quantum system is described by the Schrödinger equation:*

$$i\hbar \frac{d|\psi\rangle}{dt} = \hat{H} |\psi\rangle \quad (11)$$

*In this equation,  $\hbar$  is a physical constant known as Planck's constant whose value must be experimentally determined. The exact value is not important to us. In practice, it is common to absorb the factor  $\hbar$  into  $\hat{H}$ , effectively setting  $\hbar = 1$ .  $\hat{H}$  is a fixed Hermitian operator known as the Hamiltonian of the closed system.*

The details of the Hamiltonian are again subject to quantum physics and – depending on the system in question – the finding of a Hamiltonian and the study of its effects on the free evolution of some systems are long-standing and ongoing research topics. However, with  $\hat{H}$  being Hermitian, we know that we can decompose it into a set of eigenfunction-eigenvalue pairs

$$\hat{H} = \sum_j E_j |E_j\rangle\langle E_j|. \quad (12)$$

With  $|E_j\rangle$  being the systems energy eigenstates and  $E_j$  its energy. The state with the lowest  $E_j$  is called the system's ground state and plays an important role in many physical systems.

If the Hamiltonian acts over a certain time span, then the evolution of the quantum state, e.g. the solution to the Schrödinger Equation will be:

$$|\psi'\rangle = \hat{U}|\psi\rangle = \exp\left\{-\frac{i\hat{H}(t_2 - t_1)}{\hbar}\right\} |\psi\rangle. \quad (13)$$

You have to keep in mind that the exponential function is an operator-exponential, which in the most cases has to be treated in the infinite sum representation. Note that the relation between the Hermitian  $\hat{H}$  and the evolution operator  $\hat{U}$  is generic and you can use the relation to convert any Hermitian operator into a unitary operator (and vice versa). This Hermitian for any specific Unitary operator is then frequently called the gate's generator and can sometimes be very helpful to gain insight into the physical system.



### 2.2.3 Observables

In accordance with our every-day lab experience, we can think of the measurement of a quantity  $A$  (called observable) as numbers on a read-out device. Thus, we should require measurement outcomes to be real-valued numbers  $a_i \in \mathbb{R}$  (and not, e.g., complex numbers). Moreover, we note that any known measurement apparatus gives a specific result, which is necessarily inconsistent with the notion of superposition. After a measurement, the system's observable is of course known and therefore, irrespective of the prior state, the system after the measurement must be in a subspace of the Hilbert space, that belongs to those quantum states, which would lead to the specific measurement results. This also ensures that two repeated measurements of the same quantity give consistent results. Moreover, a measurement should not change the nature of the system, e.g., it must not force the system out of its Hilbert-space. Taking all this into account we arrive at the next postulate:

*Postulate 3 (Born's Rule): An observable/measurable  $A$  is physical quantity which is described using a Hermitian operator  $\hat{A}$ . It can be decomposed into a series of eigenvalue-projector-pairs  $\hat{A} = \sum_i a_i \hat{P}_i$ , where  $a_i$  are the possible measurement results for the specific eigenstates of the observable and  $\hat{P}_i$  are the projectors onto the subspace of the Hilbert-Space, which belong to a measurement value  $a_i$ . The measurement process is probabilistic process, which is conducted according to the following rules:*

1. The measurement will yield result  $a_i$  with a probability

$$p(A = a_i) = p_i = \langle \psi | \hat{P}_i | \psi \rangle \quad (14)$$

2. Given that the result  $a_i$  occurred, the wavefunction of the system collapsed onto the subspace consistent with that result, i.e.  $|\psi\rangle$  is replaced by:

$$|\psi\rangle \rightarrow \frac{\hat{P}_i |\psi\rangle}{\sqrt{p_i}} \quad (15)$$

The replacement is totally random, a-priori unpredictable, instantaneous and leaves no trace of the original system.

#### Projection operators

If the measurement operator is composed of entirely non-degenerate eigenvalues then the projectors are all one-dimensional projectors onto an orthonormal basis set, e.g.  $\hat{P}_i = |i\rangle\langle i|$ . If this is not the case then an arbitrary orthogonal basis can be constructed with each projector subspace and the projector operators may be written according to:  $\hat{P}_i = \sum_{k=1}^D |i_k\rangle\langle i_k|$ , where  $D$  is the number of dimensions of the degenerate subspace.

All projection operators, irrespective, if they are single-dimensional or multi-dimensional projectors fulfil the following relations:

$$\begin{aligned} \hat{P}_i^2 &= \hat{P}_i \\ \hat{P}_i \hat{P}_j &= \delta_{ij} \hat{P}_i \end{aligned} \quad (16)$$

We can think of a projection operator as an elementary observable that essentially "asks" the quantum system the question: "are you in my subspace or not"? The operators' eigenvalues (1 and 0) can be interpreted as the response (yes=1/no=0) to such a query:

$$\hat{P}_i |j\rangle = \delta_{ij} |j\rangle \quad (17)$$



After application of the “are you in my subspace or not” operator the systems state is either exclusively in the operators subspace (if the answer was “yes”) or completely out of the subspace (if the answer was “no”).

For the case of non-degenerate projectors, we may use this knowledge to phrase Born’s rule slightly differently. The probability  $p_i$  of measuring a particular value  $a_i$  when we perform a projective measurement on a state prepared in a state  $|\psi\rangle$  is the expected value of the corresponding projection operator or in other words its overlap with the projection subspace:

$$p(A = a_i) = p_i = \langle \psi | \hat{P}_i | \psi \rangle = \langle \psi | i \rangle \langle i | \psi \rangle = |\langle \psi | i \rangle|^2 \quad (18)$$

Whenever a measurement is made our knowledge about the state of the system also changes according to the outcome of the measurement. From the numerous potential outcomes, only one occurs in the measurement. Correspondingly the normalized post-measurement state becomes:

$$|\psi\rangle \rightarrow \frac{\hat{P}_i |\psi\rangle}{\sqrt{p_i}} = \frac{|i\rangle \langle i | \psi \rangle}{\sqrt{p_i}} = |i\rangle \quad (19)$$

The mere process of measurement will thus project the quantum state  $|\psi\rangle$  onto the eigenstate of the observable  $|i\rangle$ , which corresponds to the measurement result  $a_i$ .

### Expected Values and Variance of Measurables

If you have multiple, identical quantum systems at hand, you may attempt to repeat the measurement and construct statistics from them. The two most important statistical properties of a measurement are its expectation value  $E(\hat{A}) = \langle \hat{A} \rangle$  and its standard deviation  $\Delta(\hat{A})$ . They are calculated according to:

$$\begin{aligned} E(\hat{A})_{\psi} &= \langle \hat{A} \rangle_{\psi} = \sum_i p_i a_i = \sum_i a_i \langle \psi | \hat{P}_i | \psi \rangle = \langle \psi | \sum_i a_i \hat{P}_i | \psi \rangle \\ &= \langle \psi | \hat{A} | \psi \rangle \\ (\Delta(\hat{A})_{\psi})^2 &= \left\langle (\hat{A} - \langle \hat{A} \rangle_{\psi})^2 \right\rangle_{\psi} = \langle \hat{A}^2 \rangle_{\psi} - 2\langle \hat{A} \rangle_{\psi}^2 + \langle \hat{A} \rangle_{\psi}^2 \\ &= \langle \hat{A}^2 \rangle_{\psi} - \langle \hat{A} \rangle_{\psi}^2 \end{aligned} \quad (20)$$

### Complementarity of Observables

The collapse of a wavefunction leaves quite a bit of room for interpretation and discussion. One of the most immediate consequences is, the outcome of two different measurements  $\hat{A}, \hat{B}$  may depend on their respective ordering. This is clear because  $\hat{B}$ , the second measurement, may be sensitive to the part of the wavefunction that gets collapsed by the application of  $\hat{A}$ . This is, however, not necessarily the case because it may also be, that  $\hat{B}$ , still gives meaningful results, if it operates only within the degenerate subspaces that  $\hat{A}$  projects onto. Everything in between is possible, as well.

If the two measurables depend on each other, they are called complementary; if they don’t depend on each other they are called compatible/commuting. In quantum formalism, the complementarity of observables is measured by the respective difference of their reverse-ordered application of the wavefunction, e.g.  $\hat{A}$  and  $\hat{B}$  are compatible/commuting, if

$$\hat{A}\hat{B}|\psi\rangle = \hat{B}\hat{A}|\psi\rangle \quad (21)$$

Since this relation must hold, irrespective of the selected wavefunction  $|\psi\rangle$ , we can write down a simple metric for the complementarity of the two operators in terms of the commutation relation:

All notes subject to change, no guarantee to correctness, corrections welcome.

$$[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A} \quad (22)$$

For any pair of non-commuting observables  $[\hat{A}, \hat{B}] \neq 0$ , we can define an uncertainty relation for the expectation values of measurements:

$$\Delta(\hat{A}) \cdot \Delta(\hat{B}) \geq \frac{1}{2} | \langle [\hat{A}, \hat{B}] \rangle | \quad (23)$$

This means that any consecutive measurement of the quantities  $A$  and  $B$  on an ensemble of identical wavefunctions will lead to a product uncertainty of the kind given above. In simple words, this means: you can get good statistics on  $A$  and the expense of a high level of noise on  $B$  or vice versa.

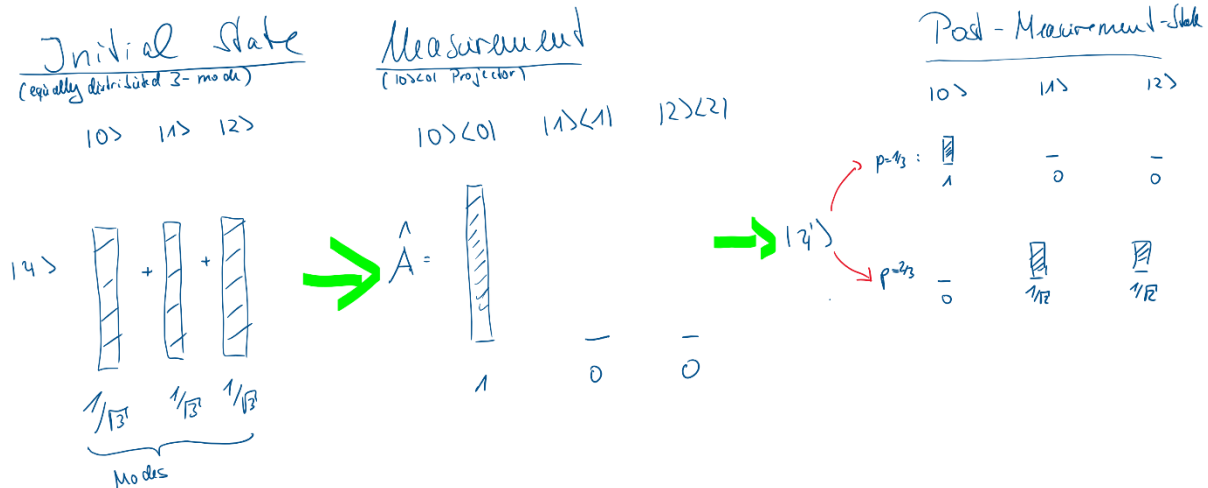


Figure 6: The measurement of an observable and the effect it has on a quantum state  $|\psi\rangle$ , defined as a superposition of three modes.

The relation however also has a meaning on the level of an identical wavefunction. It means that a precise measurement of quantity  $A$  will collapse the wavefunction onto a state, where the quantity  $B$  is particularly ill-defined. An example: if you propagate light through a pinhole with diameter  $d$ , you have knowledge on the location of any photon in the plane of the pinhole with precision  $\Delta x = d$ . This comes at the expense of having very little knowledge of the light direction of propagation after the pinhole. The uncertainty of its  $\vec{k}$ -vector is at least (in case the pinhole is illuminated by a plane wave)  $\Delta k \geq \frac{1}{2} d^{-1}$ . The product of the two uncertainties is a constant.

### 2.2.4 Composite Quantum Systems

Up until now we have only been concerned with individual quantum systems (whatever that may be; there is a more in-depth discussion of the physical background on how to count quantum systems, based on the state space of an electric field, below), now when shall discuss quantum systems composed of multiple subsystems.

*Postulate 4: The state space  $H$  of a composite physical system composed of subsystems numbered 1 through  $n$  is the tensor product  $H = H_1 \otimes H_2 \otimes \dots \otimes H_n$  of the state spaces of the component physical systems. Moreover, if system number  $i$  is prepared in the state  $|\psi_i\rangle$ , then the joint state of the total system is  $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle$ .*

You can accept this as a postulate but there is, of course, some physical reason, as to why this postulate is plausible. Assume that you have a physical system  $A$  in state  $|A\rangle$  and another physical system  $B$  in state  $|B\rangle$ . Of course there must be a way to describe the composite system  $AB$ , because it is still the

subject to the quantum nature of the world. This composite system must again be describable by a state vector (because of postulate 1) and we may call the vector  $|A\rangle|B\rangle$ . This is true for an possible state  $|A\rangle$  and  $|B\rangle$ .

Because each of the states must be describable as some kind of superposition of basis states, e.g.  $|A\rangle = \sum_i a_i |a_i\rangle$ ,  $|B\rangle = \sum_j b_j |b_j\rangle$  and  $|A\rangle|B\rangle = \sum_k ab_k |ab_k\rangle$ , we quickly come to the conclusion that the tensor product is a plausible choice to describe the state  $|A\rangle|B\rangle = \sum_{i,j} a_i b_j |a_i\rangle|b_j\rangle$ .

There are two profound consequences of this postulate, which give a hint of the complexity of quantum physical system.

### Exponential Scaling of Measurables

First, we see that each individual quantum system lives in its own individual state space. If you have, for example two free electrons, each of the systems state space is  $\mathbb{R}^3$ , because each of the electrons is free to move around in 3D-space. The state space of the composite system is, however,  $\mathbb{R}^3 \otimes \mathbb{R}^3 = \mathbb{R}^6$ . To describe two electrons, you require a six-dimensional space!

As we certainly live in a many-particle world, why do we perceive it as three-dimensional? It turns out that many body interaction (particularly those with thermal baths) tend to destroy (dephase) information from the higher dimensions, and you end up with systems that behaves very much like you would have  $n$  particles that all share the same 3D-space. In fact, lots of our difficulties in quantum systems arise, when the interaction within many-body quantum systems is much stronger than that with a thermal bath, e.g. in molecule and atom physics. This is also the very resource we want to harness with Quantum Computers.

Secondly, and this is really just a quantification of the first argument, we see that composite quantum systems tends to explode their degrees of freedom (e.g. the number of possible commuting observables) in an exponential manner.

Assume that system  $i$  has an  $n_i$ -dimensional Hilbert-Space  $H_i = \mathbb{C}^{n_i}$ . Then we know that there exists a set of basis vectors  $|j_i\rangle$  with  $j = 1 \dots n_i$  within each of that Hilbert spaces. To each basis there exists a projection operator  $\hat{P}_j^i = |j_i\rangle\langle j_i|$ , that commutes with each other, e.g.  $[\hat{P}_{j_1}^i, \hat{P}_{j_2}^i] = 0$ , which is easy to show. As we can construct any other measurable of that subsystem from superpositions of these projectors, there are no more commuting observables for that system. In other words: the consecutive application of the projection observables will give us as much info on the system as we may ever hope to extract. The consecutive application of  $\hat{P}_j^i$ , will give a series of results with  $n - 1$  zeros and a single 1, e.g.  $\{0,0, \dots, 1, \dots, 0\}$ , if we denote the position of the one-result with number  $N_i$ , it is clear that  $N_i \in \{1, \dots, n_i\}$ , e.g. the number of possible different results is for any measurement in subspace  $i$  is therefore  $n_i$ .

Within the Hilbert space  $H_{i_1} \otimes H_{i_2}$  the projection operators of different subspaces commute with each other, too, e.g.:  $[\hat{P}_{j_1}^{i_1}, \hat{P}_{j_2}^{i_2}] = 0$ . This is obvious because a measurement on subsystem A must not, by definition, affect the independent system B. Moreover, the measurement-collapse of the wavefunction does only affect the subspace within which the projector is active, because in the context of the joint Hilbert-Space  $H = H_1 \otimes H_2 \otimes \dots \otimes H_n$  the projection operator  $\hat{P}_j^i$ , really has the form  $\mathbb{I} \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I} \otimes \hat{P}_j^i \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I}$ . This means that measurements in the individual subspaces are independent of each other, because the subsystems are independent.

Therefore, from a composite quantum system with  $n$  subsystems we may extract  $N = \prod_{i=1}^n N_i$  different measurement results (composed of the  $n$  different measurements with  $N_i$  different possible results). If we have  $n$  identical systems, we can get

$$\text{Joint Number of Measurables} = (\text{Individual Number of Measurables})^{\text{Number of Particles}} \quad (24)$$

different possible measurement results. Increasing the number of particles is therefore an immensely more powerful tool in increasing the numbers of degree of freedom of the quantum system as compared to increasing the number of individual degrees of freedom.

A good example here is optics. We can easily distinguish  $10^6$  degrees of freedom of a single photon by mapping it to a single-photon sensitive camera (sCMOS, EMCCD), measuring, which pixel clicks. This seems like a lot. On the other hand, if you have a composite state of 30 photons, each of which is measured with only a simple left/right-detector, then you already have  $2^{30} \approx 10^9$  degrees of freedom, which is of course much more. The difficulty is, however, in creating such a photon state.

### Composition versus Modes

We like to put in a word of caution here. If you, like myself, have background in photonics, you may be confused as to how the discussion in this chapter goes together with the discussion of chapter 2.1. There we have focussed on the notion of modes, which play two distinct roles here:

- Every quantum system is defined on modes of the underlying field; you can think of the modes as the natural basis choice for the basis vectors of a system  $|j_i\rangle$ . For example, a photon may be defined to “live” in the superposition of horizontal and vertical polarization or as a superposition of three different waveguide modes. The number of modes, which are permissible for superposition therefore also defines the dimensionality of the  $H_i$  and thus of the different number of measurement results  $N_i$ . For the first example this would be  $N_i = 2$  and  $N_i = 3$  for the second. Thus, modes play the role of spanning the vector space for individual quantum particles.
- In chapter 2.1, we had tried to convince you, that in quantum field theories every mode is excited by a succession of photons, which can be thought of a modes in their own right. The application of the creation operator on the quantum vacuum  $\hat{a}_j^\dagger |\text{vac}\rangle$  creates a photon, by populating the first number state mode of the spatial mode  $j$ . Number state modes thus play the role of creating individual photons. A repeated application of the creation operator will create a composite system of multiple modes.

Mathematically the two types of modes are, however, not different at all. So why do they seem to play such a different role, as is visible in the equation from above? Why do the number of modes, that we excite on the one hand appear in the basis of the formula for the degrees of freedom and the other one in the exponent?

We would like to give two explanations here, one more mathematically inspired, whereas the other one is more physically inspired. They both come down to the very same thing; the question of possible correlation measurements and the way that projection operators act on the quantum fields.

From a mathematical point of view the difference lies entirely in the structure of the projection operators. A projection operator  $\hat{P}_j^i = \mathbb{I} \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I} \otimes \hat{P}_j^i \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I}$  collapses the subspace only for the degrees of freedom of the  $i$ -th particle and leaves the degrees of freedom, which belong to the other operators entirely untouched. The structure is, of course a matter of definition (or of the specific experiment) and is entirely negotiable. Superposition modes and composite particles are thus – to a certain extend – in quantum physics negotiable concepts and the currency is the type of measurement,

All notes subject to change, no guarantee to correctness, corrections welcome.

which is applied (as you will see in what follows, you are however not entire free in your choice here: many thinkable measurements are pointless because they ALWAYS will return the same result). This is a profound statement and again highlights, that in Quantum Physics the observer and his observables are an intrinsic and irreducible part of any experiment.

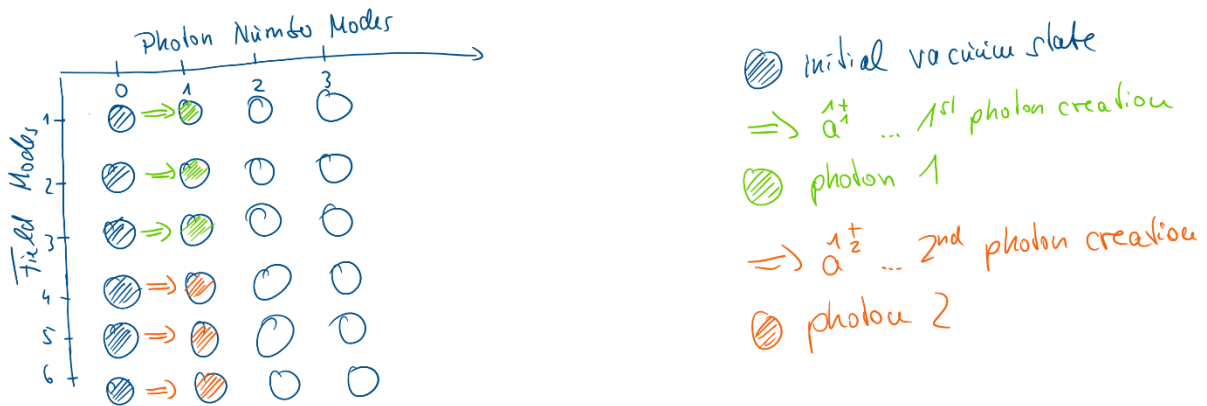


Figure 7: A composite quantum systems composed of two photons, created as superpositions of two distinguishable sets of three modes, each.

We may also answer the question from a more physical point of view. Suppose we have a system of two photons, which each occupy three different, and thus independent, modes. These could be, for example, photons which travel in a two-different three mode-waveguides or two electrons, which populate two copies of a three-level system. We can create the state by applying the creation operator for the two particles onto the vacuum. For the sake of simplicity, we shall excite the particles, such that they have equal amplitude in each of their respective modes:

$$|\psi\rangle = \hat{a}_1^{1\dagger} \hat{a}_2^{2\dagger} |\text{vac}\rangle = \frac{1}{3} (\hat{a}_1^{1\dagger} + \hat{a}_2^{1\dagger} + \hat{a}_3^{1\dagger}) (\hat{a}_1^{2\dagger} + \hat{a}_2^{2\dagger} + \hat{a}_3^{2\dagger}) |\text{vac}\rangle \quad (25)$$

This creates the state:

$$|\psi\rangle = \frac{1}{3} (|1_1\rangle + |2_1\rangle + |3_1\rangle) (|1_2\rangle + |2_2\rangle + |3_2\rangle) \quad (26)$$

Without loss of generality, we will first measure, if the first particle is in state  $|1_1\rangle$ , by applying the projector  $\hat{P}_1^1 = |1_1\rangle\langle 1_1| \otimes \mathbb{I}_2$ . Let's assume we find that the observable comes out with result 1 (which happens with probability  $1/3$ ), the resulting state of the system should then be:

$$|\psi\rangle = \frac{1}{\sqrt{3}} |1_1\rangle (|1_2\rangle + |2_2\rangle + |3_2\rangle) \quad (27)$$

This has indeed destroyed all left-over info on the first particle but perfectly retained the information carried by the second particles, just as expected. But how is that done on a level of the fields themselves? To better understand this operation, we need to look at the structure of the measurement operator. Keep in mind that:

$$|1_1\rangle = \hat{a}_1^{1\dagger} |\text{vac}_1\rangle \rightarrow \hat{P}_1^1 = |1_1\rangle\langle 1_1| \otimes \mathbb{I}_2 = (\hat{a}_1^{1\dagger} |\text{vac}_1\rangle\langle \text{vac}_1| \hat{a}_1^1) \otimes \mathbb{I}_2 \quad (28)$$

Where  $|\text{vac}_1\rangle$  denotes the vacuum state for the first class of modes only,  $|\text{vac}_2\rangle$  is the same for the second mode and  $|\text{vac}\rangle = |\text{vac}_1\rangle|\text{vac}_2\rangle$ . Note that  $(\hat{a}_1^{1\dagger} |\text{vac}_1\rangle\langle \text{vac}_1| \hat{a}_1^1) \otimes \mathbb{I}_2$  can be thought of as a selective photon counting operator, which only counts, if the first mode is in a one photon state.

The measurement (with the result 1) is carried out by the application of  $\hat{P}_1^1$  onto  $|\psi\rangle$ :

$$\begin{aligned}
 \hat{P}_1^1 |\psi\rangle &= (\hat{a}_1^{1\dagger} |\text{vac}_1\rangle \langle \text{vac}_1| \hat{a}_1^1) \otimes \mathbb{I}_2 (\hat{a}_1^{1\dagger} + \hat{a}_2^{1\dagger} + \hat{a}_3^{1\dagger}) (\hat{a}_1^{2\dagger} + \hat{a}_2^{2\dagger} + \hat{a}_3^{2\dagger}) |\text{vac}_1\rangle |\text{vac}_2\rangle \\
 &= \hat{a}_1^{1\dagger} |\text{vac}_1\rangle \langle \text{vac}_1| \hat{a}_1^1 (\hat{a}_1^{1\dagger} + \hat{a}_2^{1\dagger} + \hat{a}_3^{1\dagger}) |\text{vac}_1\rangle \otimes (\hat{a}_1^{2\dagger} + \hat{a}_2^{2\dagger} + \hat{a}_3^{2\dagger}) |\text{vac}_2\rangle \\
 &= (\hat{a}_1^{1\dagger} |\text{vac}_1\rangle \langle \text{vac}_1| (\hat{a}_1^{1\dagger} + \hat{a}_2^{1\dagger} + \hat{a}_3^{1\dagger}) \hat{a}_1^1 |\text{vac}_1\rangle + \hat{a}_1^{1\dagger} |\text{vac}_1\rangle \langle \text{vac}_1| \mathbb{I} |\text{vac}_1\rangle) \\
 &\quad \otimes (\hat{a}_1^{2\dagger} + \hat{a}_2^{2\dagger} + \hat{a}_3^{2\dagger}) |\text{vac}_2\rangle \\
 &= \hat{a}_1^{1\dagger} |\text{vac}_1\rangle \otimes (\hat{a}_1^{2\dagger} + \hat{a}_2^{2\dagger} + \hat{a}_3^{2\dagger}) |\text{vac}_2\rangle \\
 &= \hat{a}_1^{1\dagger} (\hat{a}_1^{2\dagger} + \hat{a}_2^{2\dagger} + \hat{a}_3^{2\dagger}) |\text{vac}\rangle \\
 &= |1_1\rangle (|1_2\rangle + |2_2\rangle + |3_2\rangle)
 \end{aligned} \tag{29}$$

We have gotten from the first line to the second by sorting all the terms according to the photon they operate at. The next step is to note that  $\hat{a}_1^1$  commutes with all creation operators, except with  $\hat{a}_1^{1\dagger}$ , here we have  $[\hat{a}_1^1, \hat{a}_1^{1\dagger}] = 1$ . The first term from the third line is dropped out, because  $\hat{a}_1^1 |\text{vac}\rangle = 0$ . Which leads, together with  $\langle \text{vac}_1 | \mathbb{I} | \text{vac}_1 \rangle = 1$  to the fourth line. The fourth and fifth line are then trivial.

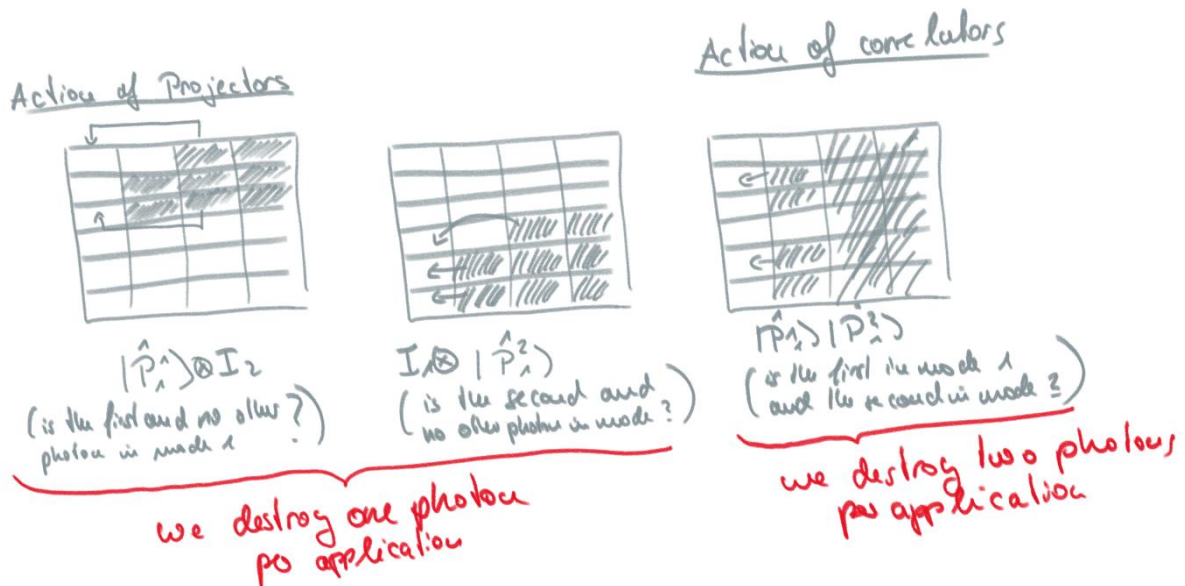


Figure 8: The action of different types of observables (Single Particle Projectors and Two Particle Correlators) on the above introduced system.

Thus, the projection operation really does the following: it destroys all photons which are in superposition (e.g. which have been created at the same times as) the target mode and recreated the photon in the target mode exclusively. It does literally nothing to the product modes, e.g. the photon, which has been created in a second step. We could now take this state and apply any of the three  $\hat{P}_{j_2}^2$  projectors to measure a correlation: e.g.  $\hat{P}_{j_1}^1 \hat{P}_{j_2}^2$  measures if the first photon in mode  $j_1$  is correlated (is simultaneously measured) with the second photon in mode  $j_2$ . Results which belong to different photons can correlate, whereas results which belong to the same photon cannot correlate.

This concept of correlation-based observables brings the two ideas together: the mathematical structure of the projection (measurement) operators defines the kind of correlations that we measure. All possible outcomes and correlation measurements span the (composite) systems Hilbert space. However, we can't just arbitrarily define correlation operators and then go about measure them, because

All notes subject to change, no guarantee to correctness, corrections welcome.



we must make sure that the system is in a quantum state as to even have a chance of getting a result from that correlation measurement. This is done by constructing an appropriate quantum state, e.g. by creating photons in distinguishable modes.

In our example: the photon state is constructed in such a way that either of the nine correlation measurements could  $\hat{P}_{j_1}^1 \hat{P}_{j_2}^2$  return true, however,  $\hat{P}_{j_1}^1 \hat{P}_{j_2}^1$  with  $j_1 \neq j_2$  will never give a "yes" answer (photon 1 is never measured in two different modes at the same time), because that part of the Hilbert-Space was never populated by the way our photons have been created.

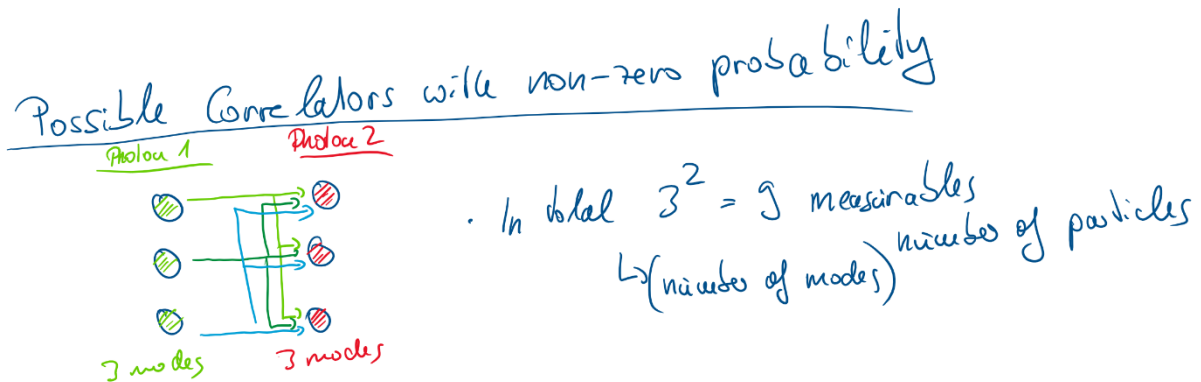


Figure 9: The number of distinguishable measurement, which can be made on the above introduced system, counted as two-mode correlators. The scaling is according to "number of modes per particle" to the power of the number of particles.

In that sense the creation of a series of photons boils down to the preparation of your quantum system in such a way as to predefine the set of possible (and impossible) outcomes of modal correlation measurements with photon counting detectors. From an observable point of view, a photon is thus nothing more, than a measurement (with non-zero information content) waiting to happen.

## 2.3 Matrix representations

If the so-far pursued bra-ket notation is a bit abstract for your taste, rest assured, all what we have really done is matrix operations. And if the observables are discrete the matrices in question are even finite-dimensional! In this chapter we shall see how this works.

Using any set of orthonormal eigenvectors  $|n\rangle$  we can write any state vector in terms of the orthonormal eigenvector basis, i.e.:

$$\begin{aligned} |\psi\rangle &= \sum \alpha_n |n\rangle \\ \langle\psi| &= \sum \alpha_n^* \langle n| \end{aligned} \quad (30)$$

where  $\alpha_n$  are complex coefficients. If we group the ket coefficients  $\alpha_n$  into a column vector

$$|\psi\rangle \rightarrow \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \vdots \end{pmatrix} \quad (31)$$

and bra vectors into row vectors

$$\langle\psi| \rightarrow (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \dots)^* \quad (32)$$

We can express the action of any operator  $\hat{O}$  on a state vector as a simple matrix multiplication:

All notes subject to change, no guarantee to correctness, corrections welcome.

$$\hat{O}|\psi\rangle \rightarrow O_{ij}\alpha_j \quad (33)$$

with a matrix with elements  $O_{ij} = \langle i|\hat{O}|j\rangle$ .

$$\hat{O} \rightarrow \begin{bmatrix} o_{11} & \dots & o_{1j} & \dots & \vdots \\ o_{21} & \dots & o_{2j} & \dots & \vdots \\ o_{31} & \dots & o_{3j} & \dots & \vdots \\ \vdots & \dots & \dots & \ddots & \vdots \\ \vdots & \dots & \dots & \dots & \ddots \end{bmatrix} \quad (34)$$

The matrix elements of a Hermitian operator  $\hat{A}$  are then given by transposition and complex conjugation  $O'_{ij}=O_{ji}^*$ . In the eigenvector basis of the observable  $\hat{A}$ , the matrix representation  $A_{ij}$  is diagonal matrix:

$$\hat{A} \rightarrow \begin{bmatrix} a_1 & 0 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 & 0 \\ 0 & 0 & a_3 & 0 & 0 \\ 0 & 0 & 0 & a_4 & 0 \\ 0 & 0 & 0 & 0 & \ddots \end{bmatrix} \quad (35)$$

which is called the spectral decomposition of the observable.

Note that the same is true for unitary evolution (gate) operators, with the difference that the diagonal elements here are not real numbers  $a_i \in \mathbb{R}$  but complex number of unit length  $\exp(i\phi_i)$  with  $\phi_i \in \mathbb{R}$ .

In the following we will mostly consider cases in which possible measurement outcomes are discrete and finite  $\{a_1, a_2, \dots, a_n\}$ , i.e. we will mostly deal with vectors of dimensionality  $N$  and matrices of dimensionality of  $N \times N$ .

Tensor products can also be expressed in terms of their matrix representations. Suppose that we want to express  $\hat{A} \otimes \hat{B}$  then we can write according to:

$$\hat{A} \otimes \hat{B} \rightarrow \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & & \\ \vdots & & \ddots & \\ a_{n1}B & & & a_{nn}B \end{bmatrix} \quad (36)$$

Let's express the above-discussed  $\hat{P}_1^1 = |1_1\rangle\langle 1_1| \otimes \mathbb{I}_2$  operator:

$$\hat{P}_1^1 = |1_1\rangle\langle 1_1| \otimes \mathbb{I}_2 \rightarrow \begin{bmatrix} \mathbb{I}_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (37)$$

Keep in mind this is a  $9 \times 9$  matrix! A joint projection operator  $\hat{P}_{j_1}^1 \hat{P}_{j_2}^2$  (e.g. a correlation operator) therefore has the form:

$$\hat{P}_{j_1}^1 \hat{P}_{j_2}^2 = \begin{pmatrix} 0 & & \dots & \\ & 1 & & \\ \vdots & & 0 & \\ & & & \ddots & \\ & & & \dots & 0 \end{pmatrix} \quad (38)$$

Which as you can see is the "real" type of projection operator for a joint system, as you can see. We can carry out the same type of exercise for a state vector:



$$|\psi\rangle = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ \vdots \\ a_1 b_n \\ a_2 b_1 \\ \vdots \\ a_n b_n \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Where the latter example is the state of the example above:  $|\psi\rangle = \frac{1}{3}(|1_1\rangle + |2_1\rangle + |3_1\rangle)(|1_2\rangle + |2_2\rangle + |3_2\rangle)$ .

## 2.4 Mixed States and the density matrix

So far, we have looked into the state of a particular quantum system per-se. In reality, however, we will typically make experiments on a series of more-or-less identical copies of a system, for example to generate some kind of statistical data. In practice it may well be that any quantum system is in fact far from reproducible and will generate a different quantum state for each repetition. In a summary, we will get an ensemble of quantum states, with some degree of statistical distribution between the different pure quantum states.

In practice, many things can contribute to such effects: emitters may have multiple decay channels, dipole-vectors jitter in their orientation, various processes may lead to inhomogeneous broadening of spectroscopic lines, your helpful co-worker may occasionally change the temperature of some nonlinear crystal, just because he can. And he will. Your hands may shake slightly upon adjustment of some setup, due to a lack of Thorlabs sending lab snacks or the coffee machine being broken down. May that never happen to you. But it will.

Such statistical ensembles of quantum states may be described with the help of the density operator

$$\hat{\rho} = \sum_i p_i \hat{\rho}_i = \sum_i p_i |\psi_i\rangle\langle\psi_i| \quad (39)$$

where  $p_i$  is the probability that the quantum system is in state  $|\psi_i\rangle$  and  $\sum_i p_i = 1$  and  $\hat{\rho}_i = |\psi_i\rangle\langle\psi_i|$  is the pure state density operator.

In reality, we are, however, more interested in measurables than in the quantum state itself (remember: only the measurement is really real). Any measurable is, of course, defined by its measurement operator  $\hat{A}$  and can be characterized by expectation value  $\langle\hat{A}\rangle$ , which is defined as:

$$\langle\hat{A}\rangle = \sum_j p_j \langle\hat{A}\rangle_j = \sum_j p_j \text{Tr}(\hat{\rho}_j \hat{A}) = \text{Tr}\left(\sum_j p_j \hat{\rho}_j \hat{A}\right) = \text{Tr}(\hat{\rho} \hat{A}) \quad (40)$$

Where  $\text{Tr}(\cdot)$  is the trace operator, i.e. the sum of the diagonal elements of the density matrix. We don't show this relation here, please look it up if you are interested.

It is noteworthy that  $\hat{\rho}$  is Hermitian (being a sum of obviously Hermitian  $\hat{\rho}_i$  with real factors) and thus can always be decomposed into eigenstates and appropriate eigenvalues, such that:

$$\hat{\rho} = \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i| \quad (41)$$

which is called the spectral decomposition of the density matrix. For example, a light source may emit 50% horizontally polarized photons and 50% diagonally upwards polarized photons, thus:

All notes subject to change, no guarantee to correctness, corrections welcome.

$$\begin{aligned}
 \hat{\rho} &= \frac{1}{2} |h\rangle\langle h| + \frac{1}{2} |u\rangle\langle u| \\
 &= \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
 &= \frac{1}{4} (2 + \sqrt{2}) \begin{bmatrix} \frac{1 + \sqrt{2}}{\sqrt{4 + 2\sqrt{2}}} & \frac{1}{\sqrt{4 + 2\sqrt{2}}} \\ \frac{1}{\sqrt{4 + 2\sqrt{2}}} & \frac{1}{\sqrt{4 + 2\sqrt{2}}} \end{bmatrix} \begin{bmatrix} \frac{1 + \sqrt{2}}{\sqrt{4 + 2\sqrt{2}}} \\ 1 \\ \frac{1}{\sqrt{4 + 2\sqrt{2}}} \end{bmatrix} \\
 &\quad + \frac{1}{4} (2 - \sqrt{2}) \begin{bmatrix} \frac{1 - \sqrt{2}}{\sqrt{4 + 2\sqrt{2}}} & \frac{1}{\sqrt{4 + 2\sqrt{2}}} \\ \frac{1}{\sqrt{4 + 2\sqrt{2}}} & \frac{1}{\sqrt{4 + 2\sqrt{2}}} \end{bmatrix} \begin{bmatrix} \frac{1 - \sqrt{2}}{\sqrt{4 + 2\sqrt{2}}} \\ 1 \\ \frac{1}{\sqrt{4 + 2\sqrt{2}}} \end{bmatrix} \\
 &= \frac{1}{4} (2 - \sqrt{2}) \left\{ \frac{1 - \sqrt{2}}{\sqrt{4 + 2\sqrt{2}}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \frac{1}{\sqrt{4 + 2\sqrt{2}}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right\}
 \end{aligned} \tag{42}$$

Which means that the spectrally decomposed version of this are again linear states of light. And this is surprisingly cumbersome.

Also note that:

$$\text{Tr}(\hat{\rho}) = 1 \tag{43}$$

And furthermore, for any quantum state  $|\psi\rangle$ , we get:

$$\langle \psi | \hat{\rho} | \psi \rangle \geq 0 \tag{44}$$

i.e. the density operator is always positive. For pure quantum state vectors the density matrix reduces to a projection operator  $|\psi_i\rangle\langle\psi_i|$ , for which the relation  $\hat{\rho}^2 = \hat{\rho}$  is readily shown. This relation is useful as it allows us to quantify the “degree of mixedness”, i.e. the state purity:

$$\text{Purity}(\hat{\rho}) = \text{Tr}(\hat{\rho}^2) \tag{45}$$

The reader can readily verify that  $\text{Purity}(|\psi_i\rangle\langle\psi_i|) = 1$  for a pure state and  $\text{Purity}(\hat{\rho}_N) = \hat{1}/N$  for a completely mixed state of dimension  $N$ .

Note that the type of uncertainty here is a different one from the uncertainty introduced by the quantum measurement process. Each of these effects may in fact be fully quantified and measured, this may just be practically impossible or impractical to deal with. Also note that each of the effects, which contribute to some kind of statistical uncertainty are themselves subject to the laws of quantum physics (even your co-worker is!). They derive from a pure state and if the system is large enough, they are unaffected by external noise. Thus, any mixed state can be purified into a pure state of a larger system. We won't show the mathematical proof here.

### 2.4.1 Entropy in Quantum Physics

In classical physics there is an intricate relation between the notion of Entropy and Information in a System. If you are more interested in that please consult the seminal works by Landauer. We'll just summarize here: the more entropy a system has, the more information it contains. I typically think about the room of my kids: if there are toys lying around everywhere there is lots of information in the

room (e.g. to describe which toy is where takes a loooong time), whereas if the room is cleaned up you can describe it with a single piece of info: everything is where it belongs<sup>2</sup>.

We would now like extend this concept to quantum physics and the idea that a pure quantum state is a minimum information/entropy state a little more formally. For a pure state, where we have complete information of the preparation procedure, we expect a measure describing disorder (if you're from a physics background) or information content (if you're a telecom engineering background) to be minimized. The von Neumann entropy is the extension of the concept of entropy from classical thermodynamics (Gibbs entropy) or information theory (Shannon entropy) to the quantum realm. It is defined as:

$$S(\hat{\rho}) = -\text{Tr}\{\hat{\rho} \text{Log}(\hat{\rho})\} \quad (46)$$

It is straightforward to verify that the von Neumann entropy<sup>3</sup> of a physical system prepared in any pure quantum state  $|\psi\rangle$  is zero:

$$S(|\psi\rangle\langle\psi|) = 0 \quad (47)$$

With the pure quantum states thus corresponding to minimum information. The state of maximum confusion, i.e. the opposite of a pure state, is the maximally mixed state in which each eigenstate of the system  $|i\rangle$  appears with equal likelihood:

$$\hat{\rho}_M = \frac{1}{N} \sum_i |i\rangle\langle i| = \frac{\hat{1}}{N} \quad (48)$$

where  $\hat{1}$  is the unit operator and  $N$  is the dimension of the state space. This is the state of maximum entropy in a Hilbert space of dimension  $N$ :

$$S(\hat{\rho}_M) \propto \log(d) \quad (49)$$

Hence you can see that the concept of the impurity of the state is closely related to the entropy of a quantum system. When you think about this for a while you can come to a few nifty conclusions on the relation of entropy, information, and the nature of coincidences:

**There are two *distinguishable* types of randomness in a quantum measurement:** If you make measurements on a mixed state you have two contributions to the statistics of the measurement: the statistics of the quantum measurement process and the classical ensemble statistics that comes from the mixed'ness of the states. While the latter does contribute to the entropy the former does not. So, there is a conceptual difference between the two classes of randomness. Only classic-statistical randomness is attributed to entropy. The reason is: the quantum randomness can be reduced to zero by virtue of choosing a measurement operator, where the quantum state is an eigenstate, e.g.  $\hat{A} = |\psi\rangle\langle\psi|$ . The selection of the (virtual) measurement operator, however, should not contribute to the systems' entropy.

**Quantum states have a fixed entropy when not measured:** A pure state does not have entropy. Any quantum operation that does not affect the purity of a state thus does not increase entropy. We know from the postulates that Unitary operators/gates  $\hat{U}$  leave the purity of a quantum state unaffected. In

---

<sup>2</sup> My colleagues tell me this example shows more than anything else, that I am German. Alas.

<sup>3</sup> in the following the entropy is commonly defined in terms of the base-2 logarithm, so that a maximally mixed state of a two-level system corresponds to one bit of entropy.

other words: unless you measure a quantum system, its entropy does not increase by its intrinsic evolution or by the application of gates. Quantum circuits do not produce entropy and are thus reversible.

**A measurement operation *can* induce entropy and is thus irreversible if the measurement outcome was not yet known:** As an example take a  $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  state, which is measured with the  $S_x = |0\rangle\langle 0| - |1\rangle\langle 1|$  operator (we shall later see, that this e.g. corresponds to a diagonally polarized photon measured with an HV-polarization beamsplitter). The result is a  $|0\rangle$  or a  $|1\rangle$  state, each with 50% probability, thus a mixed state with  $\hat{\rho} = 1/2(|0\rangle\langle 0| + |1\rangle\langle 1|)$  and an entropy of  $S(\hat{\rho}) = \text{Tr}(\hat{\rho} \log \hat{\rho}) = \log(2) = 1$  (e.g. this is a maximum entropy state). As we have increased the entropy we have made an irreversible operation.

**A measurement operation *does not need* to induce entropy and may thus be reversible if the measurement outcome was known to begin with:** If the measurement had been in parallel with the state, then we would have gotten one answer with certainty and retained a pure state. This operation is thus NOT irreversible.

Thus: **If the measurement apparatus extracts information from the quantum system. It must thus increase the quantum systems entropy:** If the entire system (measurement apparatus plus quantum system) is closed, then the overall entropy of the system cannot have been changed by the measurement. Thus, the measurement must have reduced the entropy of the measurement apparatus (by increasing that of the measured system). In other words: the measurement has transferred a certain degree of order from the quantum system to the measurement apparatus (its quantum information being measured leaves the measurement apparatus in a more well-defined state as before; e.g. it shows a specific reading and not just noise) but the apparatus must likewise transfer disorder to the quantum system. In this respect the measurement process in quantum physics may be a bit less mysterious: it's "simply" the random dephasing of a highly ordered state, when it gets in contact with a thermal bath of a large apparatus.

## 3 From Single Qubits to Circuits

As we now have a fundamental understanding of how the world works on a quantum level, we shall dive deeper into the realm of quantum information. We do so by dumbing down all the concepts from the last chapter until nothing is left but the simplest quantum system, that you can still righteously call a quantum system. A quantum system which is composed of two modes and only two modes: the Qubit.

### 3.1 The Qubit

In the classical case we can encode information in any physical system that has at least two clearly distinguishable states – a bit. Such states may be a low or high voltage; a light being turned on or off or an apple having a bite taken out of it or not. In any case we can give these two specific states logical representations and call them:

$$|0\rangle, |1\rangle \quad (50)$$

Note that the formal similarity to quantum states is at this case purposefully selected but not yet obvious. Let's call these the *computational basis states* (CBS). We can of course also use two basis states of an arbitrary quantum system as the physical representation for a bit, these basis states are also distinguishable with an appropriate measurement. Since we are in the realm of quantum physics, we now have the possibility of introducing general superposition states, called *qubit states*

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (51)$$

which is something, that one, of course, cannot do with a classical bit. Such physical system is thus called a Qubit.

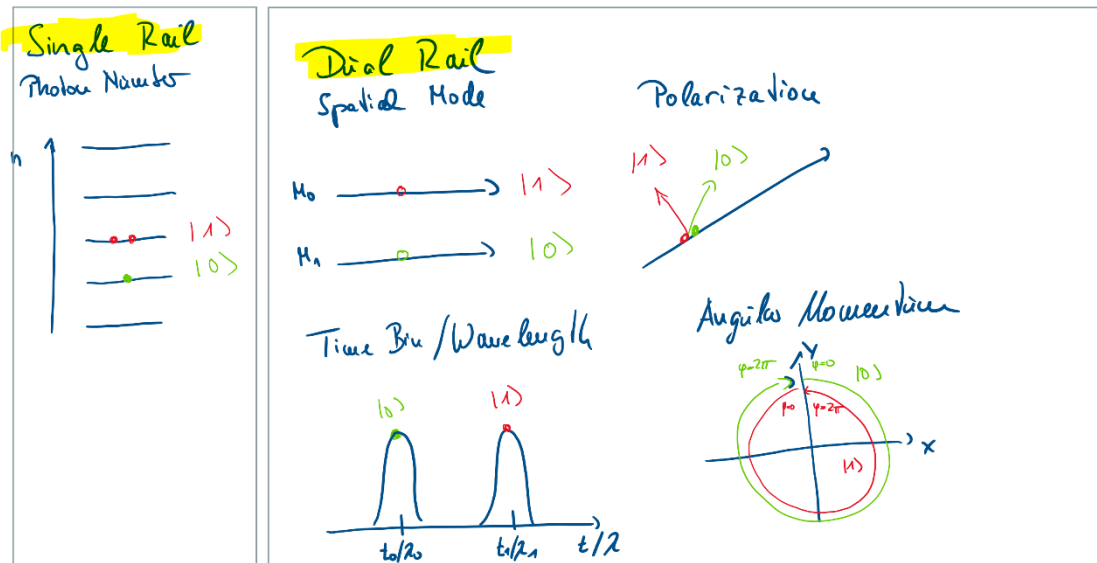


Figure 10: Some important classes of photonic Qubits.

How can we physically realize Qubits? The first option, only available if we use Bosons, is to encode the qubit in the number state of a single fixed mode with index  $i$ , which we shall call  $\hat{a}_i$ . This is known as the *single-rail qubit* representation and one possible implementation would be to differentiate between the excited and non-excited states of the field in this particular mode:

$$\begin{aligned} |0\rangle &\equiv |n_i = 1\rangle = \hat{a}_i^\dagger |\text{vac}\rangle \\ |1\rangle &\equiv |n_i = 2\rangle = \hat{a}_i^\dagger \hat{a}_i^\dagger |\text{vac}\rangle \end{aligned} \quad (52)$$

Note that we have changed the notation of the number-States somewhat (they are now called  $|n_i = 1\rangle$ ), to differentiate between them (and the vacuum-state) and the CBS. That is, the computational basis state  $|1\rangle$  corresponds to a state of the field with a two bosons in mode  $\hat{a}_i$  and the state  $|0\rangle$  corresponding to a state with one boson. Keep in mind the specific numbers are chosen completely arbitrary, in fact we are not even fixed to the notion of Fock states, should we not feel comfortable with them.

The problem with single-rail qubit encoding in optics is that loss will affect the qubit state in the sense of that it changes its value. Moreover, it requires a handle on detectors and even more so on devices and sources that create and/or mix different number states at will. This is indeed difficult. Moreover, if you want to implement operations which work differently, depending on the state of the qubit you'll have to resort to highly nonlinear elements and that's generally a bugger. They are nevertheless used quite frequently in quantum computation, e.g. superconducting Qubits are most frequently single-rail, i.e. Transmon qubits they use two different excitation states of an anharmonic electronic resonator circuit.

The second way is to fix the number-state and use a pair of orthogonal field modes  $M_i$  and  $M_j$  to encode the qubit. If we use photons, we may employ orthogonal polarized photonic modes, Gauss-Laguerre modes of different order or azimuthal phase, different modes of a single waveguide or modes of different waveguides, or different wavelengths or different time-bins or anything that you can imagine. Other systems are also frequently used: different excited states in atoms and ions. Different

topological states<sup>4</sup>. This is called the *dual-rail qubit* representation. If photons are used, the Fock state is then typically fixed to a single photon state – everything else is complicated enough already:

$$\begin{aligned} |0\rangle &\equiv |n_i = 1, n_j = 0\rangle = \hat{a}_i^\dagger |\text{vac}\rangle \\ |1\rangle &\equiv |n_i = 0, n_j = 1\rangle = \hat{a}_j^\dagger |\text{vac}\rangle \end{aligned} \quad (53)$$

To make things less abstract, let's take these modes to be orthogonal polarization modes. Two particularly popular polarization modes are the linear horizontal  $|H\rangle$  and linear vertical  $|V\rangle$  polarization (typically in reference to an optical table or a polarizing beam splitter):

$$\begin{aligned} |0\rangle &\equiv |H\rangle = \hat{a}_H^\dagger |\text{vac}\rangle \\ |1\rangle &\equiv |V\rangle = \hat{a}_V^\dagger |\text{vac}\rangle \end{aligned} \quad (54)$$

But again, we will only use that to exemplify the physical meaning of what we discuss here, and you can take any kind of qubit and apply the following discussion, because it's nice and abstract.

### 3.2 The Bloch Sphere

The first thing we do is a bit of bookkeeping. We have introduced the expansion coefficient  $\alpha$  and  $\beta$  which both are, of course complex numbers. However, this in – in fact – a bit overly complex (unintended pun!) and we can describe the entire state space with only two real numbers, which represent the latitude and longitude of an imaginary sphere, according to:

$$|\psi\rangle = \alpha|H\rangle + \beta|V\rangle = \cos 2\theta |H\rangle + e^{i\phi} \sin 2\theta |V\rangle \quad (55)$$

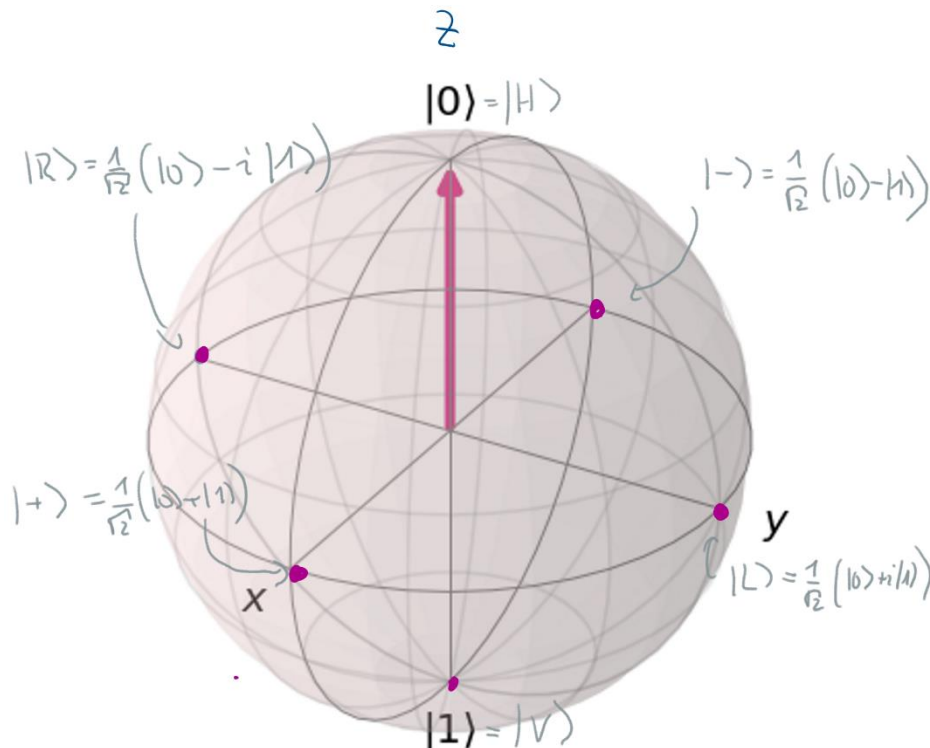


Figure 11: Representation of a qubit on the Poincarè sphere.

<sup>4</sup> In fact, we need not limit ourselves to two basis vectors but could take more. These states are then called qu-dit states.

**QISKIT Code to Plot the Vector corresponding to the  $|0\rangle$  state on a Poincaré Sphere.**

```
from qiskit_textbook.widgets import plot_bloch_vector_spherical
coords = [0,0,1] # [Theta, Phi, Radius]
plot_bloch_vector_spherical(coords) # Bloch Vector with spherical coordinates
```

Where we have used the fact that  $\alpha^2 + \beta^2 = 1$  as a justification to introduce the polar angle  $\Theta$  (longitude) and the azimuthal angle  $\phi$  (latitude) and have also utilized the fact that a cumulative phase is irrelevant (this is true for any qubit system: the total phase is irrelevant and cannot be measured). It thus becomes clear that the state of any polarization qubit and therefore ANY qubit state altogether can be represented as a point on the surface of a sphere; the infamous Bloch sphere according to the equation:

$$\begin{aligned} x &= r \sin \Theta \cos \phi \\ y &= r \sin \Theta \sin \phi \\ z &= r \cos \Theta \end{aligned} \quad (56)$$

Where  $r = 1$  (we'll see later, that  $r \neq 1$ ) also has a physical meaning.

### 3.3 Single Qubit Gates, Rotations, Universality

On the Bloch sphere the state  $|0\rangle = |H\rangle$  is represented by the north pole and  $|1\rangle = |V\rangle$  is represented by the south pole, e.g. the CBS are exclusively along the z-axis of the Bloch sphere. The other axes have a profound meaning, too: The points on the x-axis, e.g. those on the equator facing the viewer or point straight away also belong to linear polarization, namely to the diagonal basis vectors  $|+\rangle$  and  $|-\rangle$ , which can be constructed using the Hadamard operator  $\hat{H}$ :

$$\begin{bmatrix} |+\rangle \\ |-\rangle \end{bmatrix} = \hat{H} \begin{bmatrix} |0\rangle \\ |1\rangle \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} |0\rangle \\ |1\rangle \end{bmatrix} \quad (57)$$

Here the Hadamard operator is given in its matrix representation (with the CBS as an expansion basis) as:

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (58)$$

In other words:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (59)$$

In the case of photons, we will later see that find that such an action can be connected to a Half-Wave-Plate with its fast axis rotated 22.5 degrees with respect to the horizontal.

Another set of special points on the Bloch sphere are those, where the sphere intersects the y-axis. This is where the left-handed and right-handed circular basis states  $|L\rangle$  and  $|R\rangle$  (sometimes also called  $|\oslash\rangle$  and  $|\ominus\rangle$ ) are located. They can also be constructed from  $|H\rangle$  and  $|V\rangle$  according to:

$$\begin{bmatrix} |L\rangle \\ |R\rangle \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} |H\rangle \\ |V\rangle \end{bmatrix} = \mathcal{S}\hat{H} \begin{bmatrix} |H\rangle \\ |V\rangle \end{bmatrix} \quad (60)$$

In other words:

$$|L\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \quad (61)$$

The procedure makes use of the now well-established Hadamard Gate  $\hat{H}$  and the phase gate  $\hat{S}$ :

$$\hat{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad (62)$$

It is obvious that  $\hat{S}$  is a bitwise selective phase-shifter, i.e. it shifts the phase of the  $|1\rangle$  component of the state by  $\frac{\pi}{2}$  and does nothing to the  $|0\rangle$  component. Quite logically this gate is called the quarter-pi gate (no joke).

Note that  $\hat{H}$  as well as  $\hat{S}$  are unitary operators (which can be easily seen, by multiplication of their matrix with the conjugated adjoint matrices). At this point it makes sense to introduce a third unitary gate, the  $\pi/8$  or  $T$ -gate, as:

$$\hat{T} = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\frac{\pi}{4}) \end{bmatrix} \quad (63)$$

Please note the somewhat strange notation as a  $\frac{\pi}{8}$  gate, even though the phase shift is  $\frac{\pi}{4}$ . This was done because, if you come from a quantum physics background, it makes sense to introduce a symmetrized version of the gate with  $\pm\frac{\pi}{8}$  phase shift.

Each of these operators has a distinct effect on a quantum state, which can most easily be described in terms of how the state's Bloch-Vector moves over the Bloch-Sphere. Keep in mind that any unitary operator must be representable by its action on the Bloch-Sphere because there is a one-to-one connection between the sphere's surface and any possible state of a Qubit. For the three introduced operators  $\hat{H}, \hat{S}, \hat{T}$  we have the following situation:

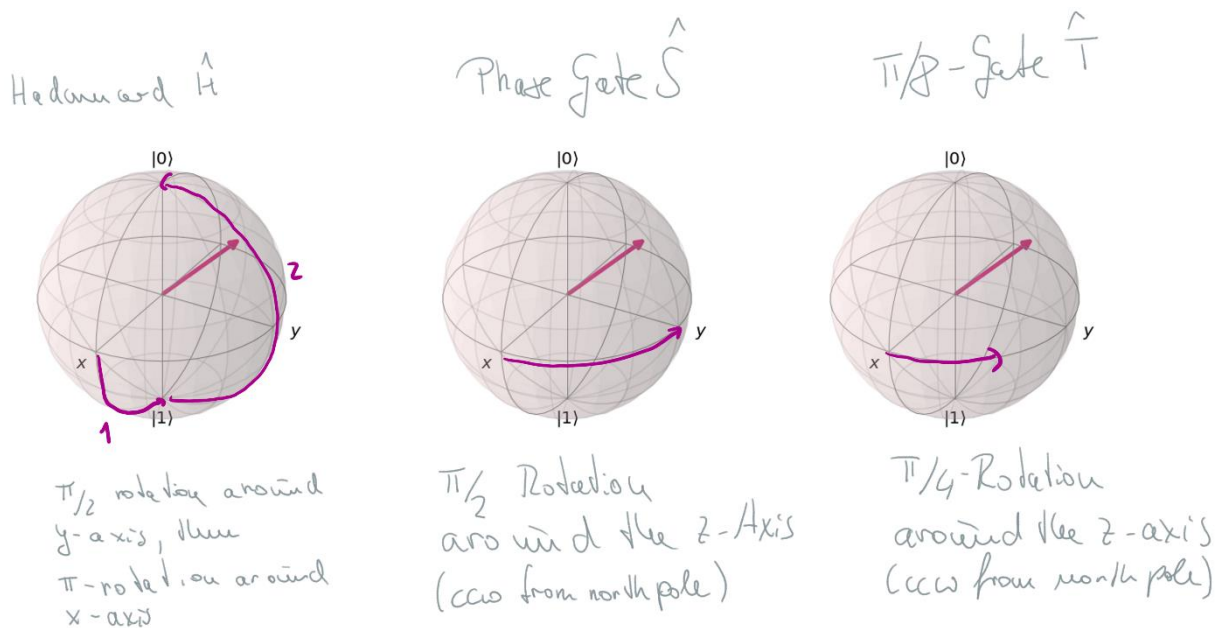


Figure 12: Actions of the  $\hat{H}, \hat{S}, \hat{T}$ -gates on a quantum state, interpreted as rotations on the Bloch-sphere.



### 3.3.1 Universality

Of course, there is an infinite number of unitary gates  $\hat{U}$ . However, any  $\hat{U}$  (except for an unimportant phase  $\alpha$ ) can be described as a rotation on the Bloch-sphere around a specific unit  $\vec{n}$  vector with an angular distance  $\theta$  that tells you how far to rotate, i.e.:

$$\hat{U} = e^{i\alpha} \hat{R}_{\vec{n}}(\theta) \quad (64)$$

There is another interesting feature: if  $\theta$  is an irrational number, we will never get back to the same angle (modulus  $2\pi$ ) if we apply the  $\hat{R}_{\vec{n}}(\theta)$  operator repeatedly. Moreover, one can show that for any target angle  $\theta'$  there is a number  $n < N$  which minimizes the distance between the rotation by the target angle  $\theta'$  and the repeated application of a rotation by the given angle  $n\theta \bmod 2\pi$ . The minimum distance roughly scales as  $\min_{n < N} E(\hat{R}_{\vec{n}}(\theta)^n, \hat{R}_{\vec{n}}(\theta')) = \mathcal{O}(N^{-1})$ .<sup>5</sup> This means, that you can approximate a rotation around a fixed axis by any (irrational angle)  $\theta'$  with around the same axis but a fixed angle  $\theta$  with a worst case error, which scales as  $1/N$ , for a maximum number of repeated rotations  $N$ .

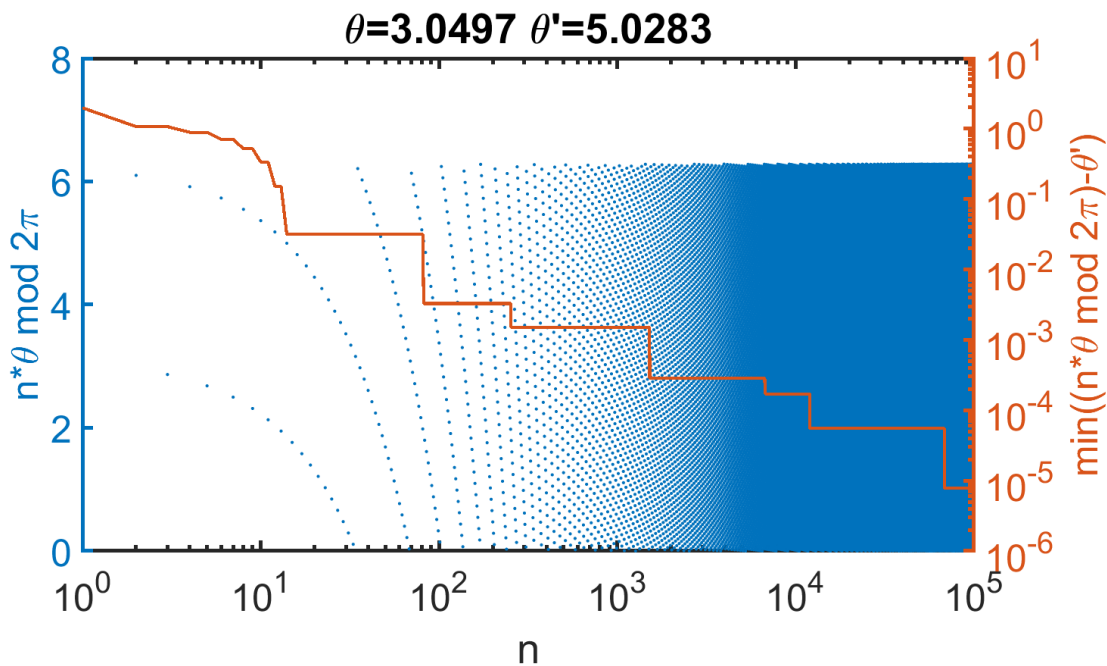


Figure 13: Approximation of a rotation of  $\theta' = 5.0283$  by the multiple application of a rotation with  $\theta = 3.0497$  with a maximum repetition number of  $N = 10^5$ . The resulting angle is plotted in blue, the precision of the best approximation is plotted in orange. Precision scaling is roughly  $\mathcal{O}(N^{-1})$

Two important rotation matrices with irrational rotation angled can be constructed from the Hadamard and the  $\frac{\pi}{8}$ -gate. The first is:

$$\hat{R}_{\vec{n}_a}^{(b)}(\theta_0) = \hat{T} \hat{H} \hat{T} \hat{H} \quad \cos\left(\frac{\theta_0}{2}\right) = \cos^2 \frac{\pi}{8} \quad \vec{n}_a = \left( \cos \frac{\pi}{8}, \sin \frac{\pi}{8}, \cos \frac{\pi}{8} \right) \quad (65)$$

The second rotates by the same angle but around a different axis:

$$\hat{R}_{\vec{n}_b}^{(b)}(\theta_0) = \hat{H} \hat{R}_{\vec{n}_a}^{(a)}(\theta_0) \hat{H} \quad \vec{n}_b = \left( \cos \frac{\pi}{8}, -\sin \frac{\pi}{8}, \cos \frac{\pi}{8} \right) \quad (66)$$

<sup>5</sup> The  $E(\dots) < \varepsilon$  notation means that all measurables of the two operators will give at maximum  $\varepsilon$  different probabilities for any type of measurement.

This together with the identity from above means that from the Hadamard gate  $\hat{H}$  and the  $\frac{\pi}{8}$ -gate  $\hat{T}$  we can construct arbitrary rotations around the unit vectors  $\vec{n}_a$  and  $\vec{n}_b$  with high precision via a multiple application of the  $\hat{R}_{\vec{n}_a}^{(a)}(\theta_0)$  and the  $\hat{R}_{\vec{n}_b}^{(b)}(\theta_0)$  gate.

This in and by itself is not more than a mathematical oddity. It becomes interesting however, when we take another matrix identity, which we shall not prove here: suppose that you are not free to choose  $\vec{n}$  but instead have two fixed, arbitrary but non-parallel unit vectors  $\vec{n}_a$  and  $\vec{n}_b$  given, then you can still construct any single qubit gate using a series of three rotations around these two axes, except for a trivial phase factor<sup>6</sup>:

$$\hat{U} = e^{i\alpha} \hat{R}_{\vec{m}_1}(\beta) \hat{R}_{\vec{m}_2}(\gamma) \hat{R}_{\vec{m}_1}(\delta) \quad (67)$$

If we take all of this together, we have seen that:

*Theorem 3: Universality of  $\hat{H}$ ,  $\hat{S}$ ,  $\hat{T}$  for single Qubit gates:  
 The set of  $\hat{H}$ ,  $\hat{S}$ ,  $\hat{T}$  gates is an efficient universal set for single qubit operations. This means that we can approximate any single qubit gate  $\hat{U}$  by a series of  $N$  of these three gates with an overall error that scales not worse than  $\frac{1}{N}$ .*

### 3.4 Observables and the Pauli-Matrices

Now that we have investigated the evolution dynamics of single qubit states, we shall focus on their measurement. For the sake of simplicity we shall identify the basis vectors as the eigenstates of the respective projection operators and construct measurement operators from the individual projectors, with measurement values 1, for the first basis vector and measurement value  $-1$  for the second basis vector. The construction is particularly simple for the CBS set

$$\hat{\sigma}_3 = \hat{\sigma}_z = |0\rangle\langle 0| - |1\rangle\langle 1| = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (68)$$

Where the matrix representation is done in the computational basis state. The operator is termed the Pauli-z or first Pauli operator, and the alphabetic naming takes its name from the corresponding axis of the Bloch sphere.

Of course, we can construct similar measurement operators from the other two sets of basis vectors, namely:

$$\begin{aligned} \hat{\sigma}_1 = \hat{\sigma}_x &= |+\rangle\langle +| - |-\rangle\langle -| \\ &= \frac{1}{2} [(|0\rangle + |1\rangle)(\langle 1| + \langle 0|) - (|0\rangle - |1\rangle)(-\langle 1| + \langle 0|)] \\ &= \frac{1}{2} [|0\rangle\langle 1| + |0\rangle\langle 0| + |1\rangle\langle 1| + |1\rangle\langle 0| - |0\rangle\langle 0| - |1\rangle\langle 1| + |1\rangle\langle 0|] \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \hat{\sigma}_2 = \hat{\sigma}_y &= |R\rangle\langle R| - |L\rangle\langle L| \\ &= \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix} \end{aligned} \quad (69)$$

---

<sup>6</sup> If you want to prove these two identities, you can just go about and plug in the matrices for the operators and see that their products lead to a matrix of the type:  $\hat{U} = \begin{bmatrix} a & ic \\ -ic & b \end{bmatrix}$  With  $a, b, c \in \mathbb{R}$  which is the most general form for a unitary  $2 \times 2$  matrix there is.

Frequently there is a fourth Pauli-Operator  $\hat{\sigma}_0 = \hat{1}$  introduced, which is the unit matrix. All four of these are obviously Hermitian, e.g.:

$$\hat{\sigma}_i = \hat{\sigma}_i^\dagger \quad (70)$$

We also note that:

$$\hat{\sigma}_i \hat{\sigma}_j = \delta_{ij} \mathbb{1} + i \epsilon_{ijk} \hat{\sigma}_k \quad (71)$$

where  $\epsilon_{ijk}$  is the Levi-Civita-Symbol or antisymmetric epsilon tensor.

Any linear operator  $\hat{M}$  on the qubit state space (e.g. any operator that acts on a two-dimensional Hilbert space and whose result still is in that space) can be constructed from a superposition of the Pauli-Operators:

$$\hat{A} = \sum_{i=0\dots3} a_i \hat{\sigma}_i \quad (72)$$

If the expansion coefficients are real, then the resulting operator  $\hat{M}$  is Hermitian, i.e. it belongs to a measurement. In other words: any quantum measurement you can make on a qubit is a superposition of the Pauli measurement operators, or, from an optics point of view a polarization measurement.

The three types of basis state sets are mutually unbiased. You can see this relation by looking at the commutation relation of their observables  $\hat{\sigma}_{1,2,3}$ , for which the relation

$$[\hat{\sigma}_i, \hat{\sigma}_j] = 2i \epsilon_{ijk} \hat{\sigma}_k \quad (73)$$

where  $\epsilon_{ijk}$  is the Levi-Civita-Symbol or antisymmetric epsilon tensor, holds. You can compare this with the uncertainty relation of chapter 2.2.3 and you will find, that the Pauli operators are mutually *complementary*, in the sense of that complete knowledge about the result of a measurement of the first means that we have absolutely no knowledge of the measurement outcome of the second.

In other words: if you decide to measure your Qubit  $|\psi\rangle$  in the CBS (which from now on in shall mean that we apply the  $\hat{\sigma}_z$  operator) then there is absolutely no information left of the qubit, which you could measure in any of the other bases. Or, to put it in an even more blunt language:

*Although the state of a qubit is characterized by two real numbers (e.g., the latitude and longitude on the Bloch-Sphere) you can only ever hope to extract a single bit of information from them.*

This is a profound finding, which cannot be stressed enough, because it limits the power of computational machines quite drastically. Although we have seen from above that quantum systems have this super high-dimensional and complex internal dynamics that we can utilize for computation, they still only give very simple answers. We may never even hope to extract their full internal state as an answer to our algorithmic problems. This is a profound difference to Turing-Machines, where you can – after the machine is finished – easily inspect the complete tape. Therefore, a large part of the difficulty in designing quantum algorithms derives from the challenge to formulate sufficiently simple “questions” that you can ask your quantum state or. In other words: quantum algorithms require the design of useful observables.

### 3.5 Mixed Single-Qubit States

In chapter 2.4 we have introduced mixed states as a representation for the statistical uncertainty of a quantum field. Of course, such kind of uncertainty may also be attributed to the state of a qubit. It may

e.g. be initialized into the  $|1\rangle$  state but might – over time – flip into the  $|0\rangle$  with probability  $p$ . The physical implementation of quantum computers aims to reduce this probability, but these probabilities are still a significant issue; they may be caused by thermal noise, vibrations, decoherence and many other effects. The final state, however, can be described by the mixed state:

$$\hat{\rho} = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1| \quad (74)$$

If the state was pure, e.g.  $p = 0$ , then the density matrix would correspond to the pure state  $|1\rangle$  and its representative point on the pole of the Bloch-Sphere. The same is true for  $p = 1$ . The mixed state above can thus be thought of as lying on the connection line between the  $|1\rangle$  and the  $|0\rangle$  point, with a fraction of  $p$  of the way from  $|1\rangle$  to  $|0\rangle$  complete. Thus, mixed states lie inside the Bloch sphere and the center of the sphere at  $\hat{\rho}_{\text{Unpol}} = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$  is the maximally mixed state, i.e. completely mixed (unpolarized).

It is also obvious that any point inside the Bloch-Sphere may be reached with multiple mixtures. As one example,  $\hat{\rho}_{\text{Unpol}} = \frac{1}{2}|R\rangle\langle R| + \frac{1}{2}|L\rangle\langle L| = \frac{1}{4}|R\rangle\langle R| + \frac{1}{4}|L\rangle\langle L| + \frac{1}{4}|+\rangle\langle +| + \frac{1}{4}|-\rangle\langle -|$  may be decomposed into mixtures of left- and right handed circular states or mixtures of left- and right handed and up- and down-polarized states, etc...

A density matrix decomposition of any point on inside the Bloch-sphere is therefore never unique. It is, however, conceptually simple to use the three orthogonal axes to define the position of any point, which we have seen above are defined by the Pauli-Matrices. Thus, one can define any mixed polarization state (and thus any mixed Qubit state) according to:

$$\hat{\rho} = \frac{1}{2}(1 + \vec{s} \cdot \hat{\sigma}) \quad (75)$$

where  $\vec{s}$  ist the so-called *Stokes-Vector*, with each entry  $s_i \in (-1,1)$ . We immediately note that  $\text{Tr}(\hat{\rho}) = 1$  is automatically fulfilled and the expectation value for a polarization measured along the axis  $i$  is given as

$$\text{Tr}(\hat{\rho}\hat{\sigma}_i) = \frac{1}{2}\text{Tr}(\hat{\sigma}_i + 2s_i) = s_i \quad (76)$$

Which is just, what we expected, i.e. if we measure any type of polarized light (pure or mixed) with a polarization beam splitter along the axis  $j$ , then we will get the value of the appropriate stokes vector entry as an average measurement result.

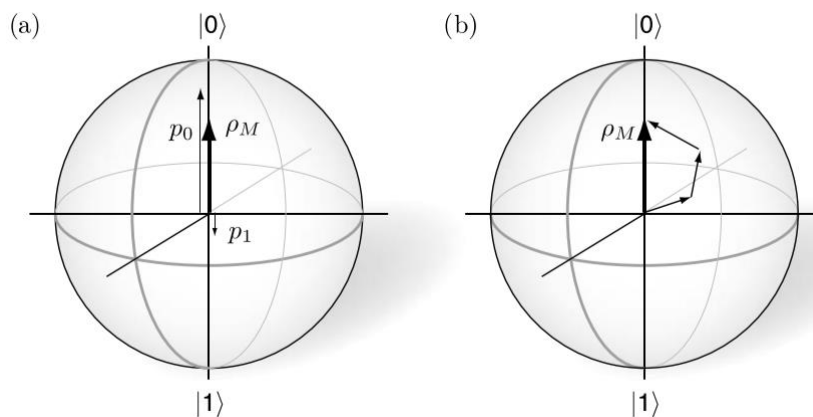


Figure 14: A mixed state is represented by a Point inside the Poincaré-sphere. (a) Representation of the state as  $\hat{\rho}_M = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$  and (b) as an alternative but equally viable mixture. (stolen from Lovett/Kok)

### 3.6 The Circuit Representation

We shall pretty soon see that Quantum Algorithms may be composed of fairly complicated sequences of gates and measurement operations. It is therefore altogether fitting to introduce a representation, which is both instructive but also concise and precise. We take flow-charts in classical computers as a role model and note that quantum flow charts are actually even more simple, because quantum algorithms must be composed of reversible quantum gates, hence the number of topologies in quantum flow charts is somewhat more restricted. But I digress.

Quantum circuits are generally composed of four types elements, with the flow of time from the left to the right.

1. Qubits (or sets thereof) are represented as solid lines, with a marker for the initial state at the left.
2. Classical bits (or sets thereof) are represented as double lines. A marker for the initial state may be omitted because their values may be overwritten (Unitarity does not apply to them).
3. Gates (Unitary Operations) are represented by squares with inputs at the left and outputs to the right. The type of gates is marked in the box. Gate parameters may be controlled by a (sequence of) classical bits. This is indicated by an extra input wire. Gates may operate on a single qubit or multiple qubits.
4. Measurement operators are marked similar to gates but indicated with a gauge symbol. They also have a classical output bit (or sequence thereof), which stores the result of the measurement.

Of course, this is extremely theoretical and we shall start with the simplest example:

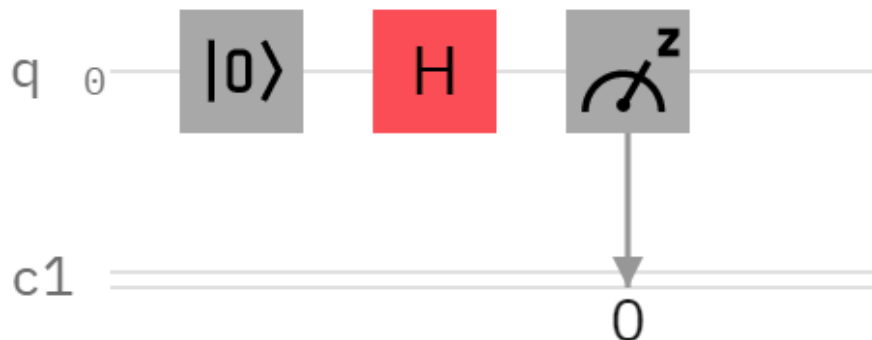


Figure 15: Circuit Representation of a simple 1-gate circuit. The circuit starts from the left with a qubit (named  $q_0$ ) in the  $|0\rangle$  state. A Hadamard operator is then applied. The qubit is then measured (in the computation basis, hence the  $z$ -notation of the measurement operator!), the result is stored in the classical bit  $c_1$ .

Because people are lazy it has become somewhat customary to skip initialization step and the measurement unless the measured result is explicitly needed in a downstream part of the code.



Figure 16: Same as Figure 15, in a simplified notation however. The initialization and measurement steps are left out for brevity.

We can, of course, also compose more complicated types of circuits, even if there is only a single qubit floating around. Here are a few examples, taken from the chapters above. We always start with a  $|0\rangle$ -state and we end up with a few of the Pauli-Basis states discussed above:

Circuit	Result	Comment
	<p>Amplitude</p> <p>Computational basis states</p> <p>Phase 0</p>	$ 0\rangle \rightarrow  +\rangle$
	<p>Amplitude</p> <p>Computational basis states</p> <p>Phase 0</p>	$ 0\rangle \rightarrow  L\rangle$
	<p>Amplitude</p> <p>Computational basis states</p> <p>Phase 0</p>	$ 0\rangle \rightarrow  -\rangle$


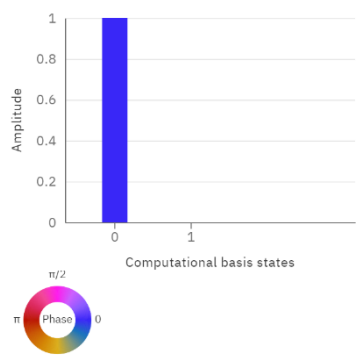
Circuit	Result	Comment
		$\hat{H}^2 = \mathbb{I}$

Figure 17: A few examples for 1-bit quantum circuits.

We shall later see, how we can use this scheme to implement and represent complex quantum circuits.

## 4 Multiple Qubits, Entanglement, and Universality

So far, we have only discussed individual Qubits. Most protocols in Quantum Information Processing rely explicitly on composite systems of multiple Qubits. Imagine a physical system, which consists of multiple qubits, say for example multiple photons, which we shall number from 1 to  $N$ . Thus, the state of this qubits must be given by

$$|\psi_i\rangle = \alpha_0^{(i)} |0_i\rangle + \alpha_1^{(i)} |1_i\rangle \quad (77)$$

In other words: each Qubit's Basis spans its own two-dimensional Hilbert-Space  $\mathcal{H}_i$ . A system of  $N$  Qubits is must there span a Hilbert space  $\mathcal{H}$ :

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_N \quad (78)$$

Which means that the Hilbert space is spanned by the basis vectors composed of all possible combinations of individual computational basis vectors for the individual basis states  $|b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_N\rangle$ , where  $|b_i\rangle \in \{0,1\}$ . Thus, any state in the complete system is given by

$$\begin{aligned} |\psi\rangle &= \sum_{b_1=0}^1 \sum_{b_2=0}^1 \dots \sum_{b_N=0}^1 \alpha_{b_1 b_2 \dots b_N} |b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_N\rangle \\ &= \sum_{b_1=0}^1 \sum_{b_2=0}^1 \dots \sum_{b_N=0}^1 \alpha_{b_1 b_2 \dots b_N} |b_1 b_2 \dots b_N\rangle \end{aligned} \quad (79)$$

Where  $\sum_{b_1=0}^1 \sum_{b_2=0}^1 \dots \sum_{b_N=0}^1 |\alpha_{b_1 b_2 \dots b_N}|^2 = 1$  must hold for reasons of normalization. The second line differs from the first in just the fact that the tensorial product of the basis vectors has been written in a shorthanded notation. To make this more obvious:  $|b_1 b_2 \dots b_N\rangle$  is the state, where each Qubit  $i$  is in the state  $|b_i\rangle$ ; e.g.  $|000\rangle$  is a three Qubit system in a state where all Qubits have value zero, e.g. they are all horizontally polarized. These basis vectors  $|b_1 b_2 \dots b_N\rangle$  are called the computational basis states (CBS) of the composite system. If the composite system  $|\psi\rangle$  is in a product state

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle \quad (80)$$

then the relation of the quantum amplitudes is simply:

$$\alpha_{b_1 b_2 \dots b_N} = \alpha_{b_1}^{(1)} \cdot \alpha_{b_2}^{(2)} \cdot \dots \cdot \alpha_{b_N}^{(N)} \quad (81)$$

However, most states in the combined system cannot be rewritten in terms of individual product states, as defined above, which becomes immediately clear from simple combinatorial arguments. Assume that you have a  $N$ -Qubit system, then you require  $2^N$  quantum amplitudes  $\alpha_{b_1 b_2 \dots b_N}$  to describe any possible state of that system. If you, however, have  $N$  individual states there are just  $2N$  individual quantum amplitudes  $\alpha_i, \beta_i$ . To make that more obvious: assume you have a three-qubit system. There are eight possible combinations of the individual qubit states  $|b_i\rangle$  and thus eight possible basis states  $|b_1 b_2 b_3\rangle$  with eight expansion coefficients  $\alpha_{b_1 b_2 b_3}$ , running from  $\alpha_{000}$  to  $\alpha_{111}$ . If the state was composed of individual states there were only six  $\alpha_0^{(1)} \dots \alpha_0^{(3)}$  and  $\alpha_1^{(1)} \dots \alpha_1^{(3)}$

From this simple argument you immediately see that multi-qubit systems have a much larger complexity than all of their composite systems individually. Moreover, the difference scales exponentially and it is exactly that exponential scaling of the number of internal degrees of freedom, which is leveraged in a quantum computers to make complex calculations. We can use this revised understanding to try and refine Definition 3:

*Definition 5: A quantum computer is a device, which makes use of the exponential scaling of the degrees of freedom of a multipartite quantum system (typically of multiple qubits) as a resource in solving computational tasks.*

## 4.1 Two-Qubit States and Entanglement

Let's now focus on a system composed of two Qubits, to elaborate on the nature of the internal degrees of freedom inherent in a multipartite quantum system and some of its consequences.

### 4.1.1 Product States and Non-Correlation

So far, we have used the computational basis states:  $|00\rangle, |01\rangle, |10\rangle, |11\rangle$  and superpositions thereof to describe any state of the quantum system  $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$ .

If the quantum system in question is in any product state  $|\psi\rangle = |\psi_1\rangle|\psi_2\rangle$ , then we can be certain, that any measurement (i.e. a polarization measurement) on the first Qubit does not affect the outcome of the measurement on the second Qubit, whatsoever. Nor does it produce any information on the state of the second Qubit. To show this we assume an arbitrary measurement on Qubit one  $\hat{A}_1$ , which we shall describe by its two orthogonal projection operators and measurement results of  $\pm 1$ . The basis states of the projection operators shall be called  $|a_1\rangle$  and  $|a_2\rangle$  without loss of generality

$$\hat{A}_1 = |a_1\rangle\langle a_1| - |a_2\rangle\langle a_2| \quad (82)$$

We can decompose the state of the first qubit into the basis states of the first measurement operator, according to  $|\psi_1\rangle = \cos \theta |a_1\rangle + \sin \theta \exp(i\phi) |a_2\rangle$ :

$$|\psi_1\rangle|\psi_2\rangle = \cos \theta |a_1\rangle|\psi_2\rangle + \sin \theta \exp(i\phi) |a_2\rangle|\psi_2\rangle \quad (83)$$

The measurement then collapses the first Qubit onto  $|a_1\rangle$  with probability  $\cos^2 \theta$  resulting in a joint state of  $|a_1\rangle|\psi_2\rangle$  and onto  $|a_2\rangle$  with probability  $\sin^2 \theta$  resulting in a joint state of  $|a_2\rangle|\psi_2\rangle$ . The result is classically random ensemble and must therefore be treated in the mixed state formalism with a density matrix:

$$\begin{aligned} \hat{\rho} &= \cos^2 \theta |a_1\rangle|\psi_2\rangle\langle\psi_2|\langle a_1| + \sin^2 \theta |a_2\rangle|\psi_2\rangle\langle\psi_2|\langle a_2| \\ &= (\cos^2 \theta |a_1\rangle\langle a_1| + \sin^2 \theta |a_2\rangle\langle a_2|) \otimes |\psi_2\rangle\langle\psi_2| \\ &= \hat{\rho}_1 \otimes |\psi_2\rangle\langle\psi_2| \end{aligned} \quad (84)$$



From this result you can clearly see, that the measurement procedure has neither extracted any information from the second qubit, nor has it affected the second qubit in any tangible way or form. The measurement results are thus uncorrelated. Moreover, the result has left Qubit 2 in a pure state.

Altogether this seems like a rather classical result: a measurement on Qubit 1 does not affect Qubit 2 and it also does not produce any prior information on Qubit 2. Or to put it in other terms: product states behave like classically independent systems, they are thus kind of boring.

#### 4.1.2 Non-Product States, Correlation, and Entanglement

We shall now see that this classicality is not maintained for non-product states. For this we shall introduce a new basis set for the two-qubit system as an alternative to the CBS  $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ . Among the many possible set of basis states, one, which stands out particularly, is the set of maximally entangled Bell-States  $|\Psi/\Phi^\pm\rangle$ :

$$\begin{aligned} |\Psi^\pm\rangle &= \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle) \\ |\Phi^\pm\rangle &= \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle) \end{aligned} \quad (85)$$

Let's repeat our measurement experiment for any of these, say  $|\psi\rangle = |\Phi^+\rangle$

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0_1 0_2\rangle + |1_1 1_2\rangle) \quad (86)$$

We measure the first Qubit in an arbitrary observable, which is defined by its projection-based measurement operator. As a reminder this operator is

$$\hat{A}(\theta, \phi)_1 = |a_1\rangle\langle a_1| - |b_1\rangle\langle b_1| \quad (87)$$

The measurement corresponds to some arbitrary basis (not necessarily the CBS), which can be represented by a point on the Bloch sphere for  $|a_1\rangle$  and a point on the opposite side for  $|a_2\rangle$ , which we can describe by the two angles  $\theta$  and  $\phi$  according to the equations:

$$|a_1\rangle = \cos \theta |0_1\rangle + \sin \theta \exp i\phi |1_1\rangle \quad |b_1\rangle = \sin \theta \exp(-i\phi) |0_1\rangle - \cos \theta |1_1\rangle_1 \quad (88)$$

This simply means, that  $\theta$  represents how far away on the Bloch-Sphere we are from the CBS. Here  $\theta = 0$  and  $\theta = \frac{\pi}{2}$  represent measurements in the CBS basis and  $\theta = \pm \frac{\pi}{4}$  represent measurements on the equator of the Bloch-Sphere, e.g. the  $|\pm\rangle$  or the  $|L/R\rangle$  bases or superpositions thereof. The specific choice of factors also automatically ensures that  $|a_1\rangle$  and  $|b_1\rangle$  are orthonormal, i.e. they are a valid basis set.

As we must expand the CBS in which the initial state was defined into these states anyway it makes sense to expand the basis states into the eigenstates of the observable:

$$|0_1\rangle = \cos \theta |a_1\rangle + \sin \theta \exp i\phi |b_1\rangle \quad |1_1\rangle = \sin \theta \exp(-i\phi) |a_1\rangle - \cos \theta |b_1\rangle \quad (89)$$

At any rate, we can now describe the first Qubit state as a superposition of the measurement basis and we get:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{2}}[(\cos \theta |a_1\rangle + \sin \theta \exp(i\phi) |b_1\rangle)|0_2\rangle + (\sin \theta \exp(-i\phi) |a_1\rangle - \cos \theta |b_1\rangle)|1_2\rangle] \\ &= \frac{1}{\sqrt{2}}[(\cos \theta |0_2\rangle + \sin \theta \exp(-i\phi)|1_2\rangle)|a_1\rangle + (\sin \theta \exp(i\phi) |0_2\rangle - \cos \theta |1_2\rangle)|b_1\rangle] \end{aligned} \quad (90)$$

The measurement then collapses the first Qubit and each of the terms has a certain probability of being the resulting state after collapse. The probabilities are:

$$\begin{aligned}
 p(A_1 = +1) &= \langle \psi | \hat{P}_a | \psi \rangle \\
 &= \langle \psi | a_1 \rangle \langle a_1 | \psi \rangle \\
 &= \frac{1}{2} [(\cos \theta \langle 0_2 | + \sin \theta \exp(i\phi) \langle 1_2 |)] [(\cos \theta |0_2\rangle + \sin \theta \exp(-i\phi) |1_2\rangle)] \\
 &= \frac{1}{2} [\cos^2 \theta + \sin^2 \theta] \\
 &= \frac{1}{2} \\
 p(A_1 = -1) &= \langle \psi | \hat{P}_b | \psi \rangle = \frac{1}{2}
 \end{aligned} \tag{91}$$

The states after the measurement are:

$$\begin{aligned}
 |\psi | A_1 = +1 \rangle &= (\cos \theta |0_2\rangle + \sin \theta \exp(-i\phi) |1_2\rangle) |a_1\rangle \\
 |\psi | A_1 = -1 \rangle &= (\sin \theta \exp(i\phi) |0_2\rangle - \cos \theta |1_2\rangle) |b_1\rangle
 \end{aligned} \tag{92}$$

Here we note the first curious thing. The resulting probability distributions of Qubit 1 do not at all depend on the type of measurement applied. From the single particle picture, you would expect that a quantum particle must have one specific observable, where the result is fixed. Or to put it more bluntly: by now you have accepted that it may not be clear what property a Quantum Particle may have, but you would surely expect that it should have some fixed property. Yet, any possible measurement, which you can apply on Qubit 1 gives the same result. It seems like Qubit 1 has become a particle without properties. This also means that there is no point on the Bloch Sphere, which describes the state of Qubit 1.

In a sense Qubit 1 has ceased to exist as an independent particle. Instead, it has gone into a state, in which it does not make sense to think about the properties of Qubit 1 without resolving its connection with Qubit 2. Both Qubits have become ENTANGLED.

That said, let's explore the status of the joint system after the measurement on Qubit 1. As it is in a mixed state it must be described using the density matrix approach, where we can simply read off the entirety of the density operator from the table above

$$\hat{\rho} = \frac{1}{2} [|\psi | A_1 = +1 \rangle \langle \psi | A_1 = +1 | + |\psi | A_1 = -1 \rangle \langle \psi | A_1 = -1 |] \tag{93}$$

Which is clearly not factorizable in the same way, as the non-correlated state from above. Let's elaborate on this a bit more in-depth by explicitly calculating the state of the second Qubit. This is done by calculating the partial trace over the first Qubit (e.g. a hypothetical measurement with the identity operator for Qubit 1).

$$\begin{aligned}
 \hat{\rho}_2 &= \text{Tr}_1 \hat{\rho} = \sum_i \langle a_i | \hat{\rho} | a_i \rangle \\
 &= \frac{1}{2} (\cos \theta |0_2\rangle + \sin \theta \exp(-i\phi) |1_2\rangle) (\cos \theta \langle 0_2| + \sin \theta \exp(i\phi) \langle 1_2|) \\
 &\quad + \frac{1}{2} (\sin \theta \exp(i\phi) |0_2\rangle - \cos \theta |1_2\rangle) (\sin \theta \exp(-i\phi) \langle 0_2| - \cos \theta \langle 1_2|) \\
 &= \frac{1}{2} (\cos^2 \theta + \sin^2 \theta) |0_2\rangle \langle 0_2| + \frac{1}{2} (\cos^2 \theta + \sin^2 \theta) |1_2\rangle \langle 1_2| \\
 &\quad + \frac{1}{2} (\cos \theta \sin \theta \exp(i\phi) - \cos \theta \sin \theta \exp(i\phi)) |0_2\rangle \langle 1_2| \\
 &\quad + \frac{1}{2} (\cos \theta \sin \theta \exp(-i\phi) - \cos \theta \sin \theta \exp(-i\phi)) |1_2\rangle \langle 0_2| \\
 &= \frac{1}{2} [|0_2\rangle \langle 0_2| + |1_2\rangle \langle 1_2|] \tag{94}
 \end{aligned}$$

This is not just any mixed state but a maximally mixed state according to the definition in chapter 2.4.1. This means that a measurement in Qubit 1 does not only increase the information content (entropy) of Qubit 1 it also increases the entropy of Qubit 2. Indeed, this is much weirder than you would initially expect. Let's set this aside for a second and use this finding to define the entangledness of a quantum system:

*Definition 6: The degree of Entanglement of a two-Qubit quantum system in a joined state  $|\psi\rangle$  is measured by testing the purity of the state of Qubit 2 after a measurement  $A_1$  is applied onto Qubit 1, i.e. let  $\hat{\rho}$  be the state of the joint system after application of measurement  $A_1$  then the entanglement  $E$  is calculated using  $E = 2 \text{Tr}[(\text{Tr}_1 \hat{\rho})^2]$ .  $E \in [0,1]$  with  $E = 0$  indicating non-entanglement and  $E = 1$  indicating maximum entanglement.*

*The specific kind of measurement of Qubit 1 does not matter. A generalization with larger systems is straightforward.*

Let's return to the weirdness of entangled systems. Previously, we had seen that Quantum Systems are transferred from a pure into a mixed state by measurement only. But we have never even touched Qubit 2. We have only measured Qubit 1. Still, in the process we have transformed Qubit 2 into a mixed state. This means we must have made implicitly made some sort of measurement with Qubit 2. Let's find this out and do so by applying the observable  $\hat{A}(\theta, -\phi)_2$ , onto Qubit 2 (this is the same as for Qubit 1, with the only exception that the phase shift between the two measurement bases is reversed, e.g. the sense of the chirality is flipped).

We rewrite the state of the second Qubit system into two parts, according to the measurement outcome of  $A_2$  (we could proceed with the complete  $\hat{\rho}$  from above but then the equations get somewhat lengthy):

$$\begin{aligned}
 |\psi_2 |_{A_1 = +1} \rangle &= \cos \theta |0_2\rangle + \sin \theta \exp(-i\phi) |1_2\rangle \\
 |\psi_2 |_{A_1 = -1} \rangle &= \sin \theta \exp(i\phi) |0_2\rangle - \cos \theta |1_2\rangle \tag{95}
 \end{aligned}$$

Let's now apply the same measurement (let's call it  $A_2$ ), which have applied to the first Qubit on the second qubit. We now calculate the probabilities of  $A_2$  by noting that  $p(A_2 = a_q | A_1 = a_r) = \text{Tr}(\hat{\rho}_2(A_1 = a_r) |a_q\rangle \langle a_q|) = \sum_i \langle a_i | \hat{\rho}_2(A_1 = a_r) |a_q\rangle \langle a_q | a_i \rangle$ . We read them off as:

$$\begin{aligned}
 p(A_2 = +1|A_1 = +1) &= |\langle a_1 | \psi_2 | A_1 = +1 \rangle|^2 \\
 &= |\langle a_1 | (\cos \theta |0_2\rangle + \sin \theta \exp(-i\phi) |1_2\rangle) \rangle|^2 \\
 &= |\cos^2 \theta + \sin^2 \theta|^2 \\
 &= 1 \\
 p(A_2 = -1|A_1 = +1) &= |\langle a_2 | \psi_2 | A_2 = +1 \rangle|^2 & (96) \\
 &= |\cos \theta \sin \theta \exp(-i\phi) - \cos \theta \sin \theta \exp(-i\phi)|^2 \\
 &= 0 \\
 p(A_2 = +1|A_1 = -1) &= 0 \\
 p(A_2 = -1|A_1 = -1) &= 1
 \end{aligned}$$

Note, that we have explicitly shown, how the first solution is obtained and then just given the result for the second to fourth. We now group the four cases into two classes. The situation  $(A_2 = +1|A_1 = +1)$  and  $(A_2 = -1|A_1 = -1)$  mean that the measurements on Qubit Number 2 will yield the SAME result as the measurement on Qubit Number 1 (correlation). The other two situations correspond to measurements with different results (anticorrelation). We find that both members in both of the classes are equal and they are 1 and 0 exclusively.

This result is profound: a measurement of Qubit 1 with observable  $A_1$  with any result will force Qubit 2 to instantly collapse into the same resulting state for observable  $A_2$ . The results are perfectly correlated. Moreover, and this is an important point: the correlation is maintained irrespective of the measurement basis! The two Qubit give the same results, irrespective of what you measure, as long as you make the same measurement.

Or in other words, the observable in the  $A_1$  measurement basis is perfectly correlated to the observable in the same basis, with a flipped phase as represented by the observable  $A_2$ . Here we have only discussed this relation for an initial two-Qubit system in the  $|\Phi^+\rangle$  state but one can show that for the other three Bell-States there is a correlated Basis for Qubit 2 for any possible measurement of Qubit 1, too (there is relation is just a slight bit more complicated than just a flip of the  $\phi$ -phase). This leads us to an alternative definition of entanglement:

*Definition 7: Two Qubits are completely entangled, if for any basis set for Qubit 1 there exists a corresponding basis set for Qubit 2, in which a measurement is guaranteed to yield the identical result. The degree of entanglement can be quantified by the maximum degree of correlation between a measurement in a basis set in Qubit 1 and the most correlated basis set in Qubit 2.*

In other words: measurements in entangled systems produce correlated results, irrespective of the measurement!

In fact, one can show, that such a behaviour produces a stronger correlation than could be constructed for any kind of classical interaction. This is done by generalizing our analysis to measurements on Qubit 1 and Qubit 2 into combinations of three different bases and the derivation of a quantity  $E$  (not to be confused with the degree of entangledness), which expresses the correlations of these different measurements. It can be shown, that there is a range of values for  $E$  which can be reached by Quantum Systems but not by classical systems; the resulting inequality is the so-called CHSH-Version of Bell's inequalities. They can be tested for experimentally (which has been done first by a team around A. Aspect in 1984, see below) and it was indeed shown that two entangled Qubits exhibits correlations, which cannot be explained with classical particles; this is generally considered a resounding proof that quantum physics is required to describe nature properly.

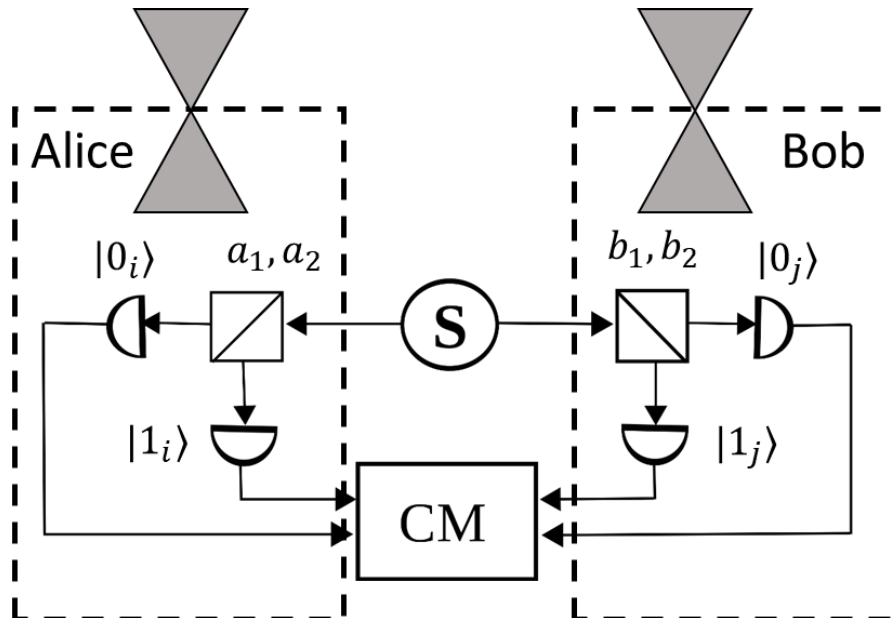


Figure 18 Schematic of a two-channel Bell-test with polarization splitters using two time-like separated observer Alice and Bob. The test has to be re-run at least four times for all combinations of two different settings  $A_a, A_b$  of the left and  $B_a, B_b$  of the right polarizer. The strongest deviation for the classical prediction of  $S = 2\sqrt{2}$  can be found at  $a=0^\circ, b=45^\circ$  and  $c=22.5^\circ, d=67.5^\circ$  (Tsirelson's bound)

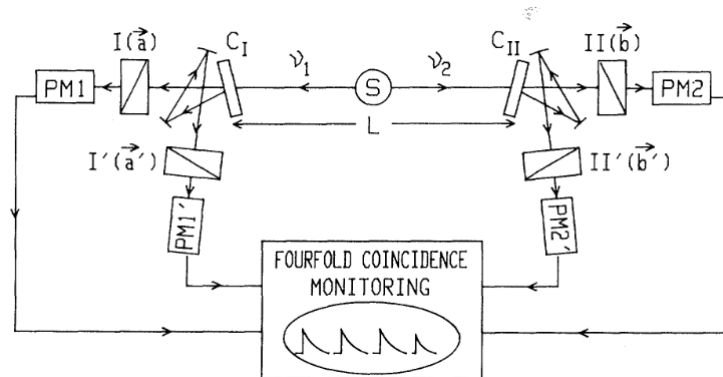


Figure 19 Scheme of the Aspect-experiment, the first to successfully demonstrate  $S > 2$ . PRL **49** 1804 (1982).

## 4.2 Controlled Operations on a single Qubit

In one of the last chapters we discussed single Qubit gates, in depth. Although there is a surprising amount stuff to learn there, it is of course not enough to build a quantum computer, the same way, that single bit operation are not enough to build an ordinary. One particular operation of a classical computer, that inherent requires two bit gates are controlled operations, i.e. operation in which the action on one bit depends on the value of another.

### 4.2.1 The CNOT Operation

The simplest (and as we shall soon see the only one which is really required) is the controlled NOT or CNOT operation; typically abbreviated as  $\widehat{CX}$ . The CNOT operation has two inputs, dubbed the control Qubit  $|c\rangle$  and the target Qubits  $|t\rangle$ . The state of the target Qubit is supposed to flip, if the control Qubit is in state  $|0\rangle$ . You can quite easily see that the CNOT is in principle the quantum version of an EXOR or a half-adder, e.g.

$$\widehat{CX}(|c\rangle|t\rangle) = |c\rangle|c\oplus t\rangle \quad (97)$$

We can, of course, also write the gate as a superposition of projectors:

$$\begin{aligned} \widehat{CX} &= |0_c\rangle\langle 0_t| + |1_t\rangle\langle 1_t|\langle 0_c| + |1_c\rangle\langle 0_t| + |1_t\rangle\langle 0_t|\langle 1_c| \\ &= |0_c\rangle\mathbb{I}_t\langle 0_c| + |1_c\rangle(1 - \mathbb{I}_t)\langle 1_c| \end{aligned} \quad (98)$$

Or we can write it as a matrix:

$$\widehat{CX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (99)$$

And there is, as you probably expected also a specific symbol, which is used in the circuit-model notation:

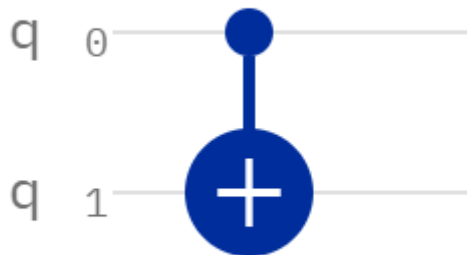


Figure 20: Circuit Representation of a CNOT register.

While the CNOT-Gates seems rather trivial there are a lot of fancy things that you can do with CNOTs and just a few other gates. The first fancy thing to note, is that for CNOT operation the roles of the control and the target Qubit are largely interchangeable, indeed, we find that:

$$\hat{H}_c \hat{H}_t \widehat{CX} (\hat{H}_c |c\rangle \hat{H}_t |t\rangle) = \widehat{CX} (|t\rangle |c\rangle) = |c \oplus t\rangle |t\rangle \quad (100)$$

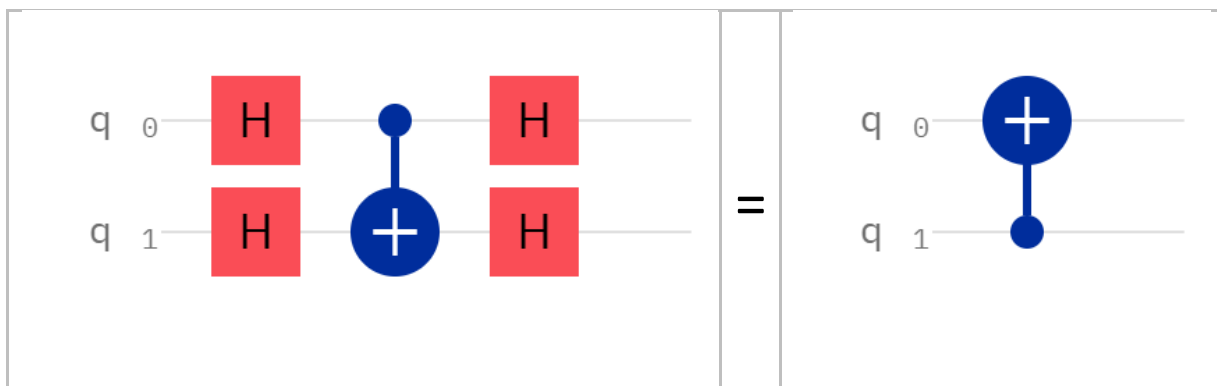


Figure 21: Two equivalent representations of the CNOT gates with the roles of the control and target Qubit interchanged.

#### 4.2.2 Bell State Creation and Measurement

Another fancy use of the CNOT-gate is the construction and measurement of Bell states from CBS-states. Indeed, we find the simple relation:

$ c\rangle$	$ t\rangle$	$CX(\hat{H} c\rangle t\rangle)$
$ 0\rangle$	$ 0\rangle$	$ \Phi^+\rangle$
$ 0\rangle$	$ 1\rangle$	$ \Phi^-\rangle$
$ 1\rangle$	$ 0\rangle$	$ \Psi^+\rangle$
$ 1\rangle$	$ 1\rangle$	$ \Psi^-\rangle$

We can exploit the invertibility of quantum circuits to also map Bell states onto CBS and thereby creating measurement systems for Bell states.

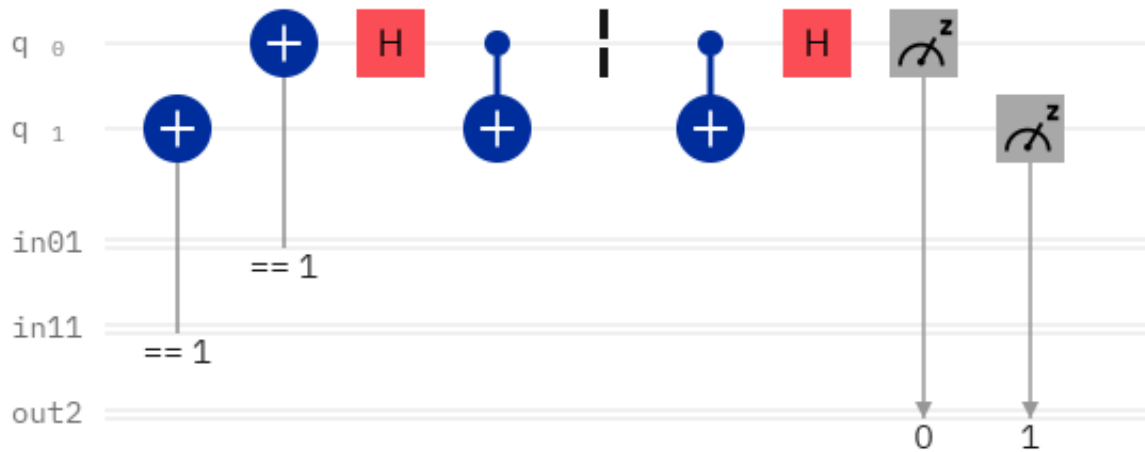


Figure 22: A Bell-State creator (left of the Barrier), which creates any of the four Bell states according to the input bits  $in_0$  and  $in_1$  and a Bell state measurement circuit, which measures the Bell state and outputs the result (0,1,2,3) into the classical two-bit register  $out_2$ .

Here is an alternative circuit, which creates a random Bell-states and then measures it:

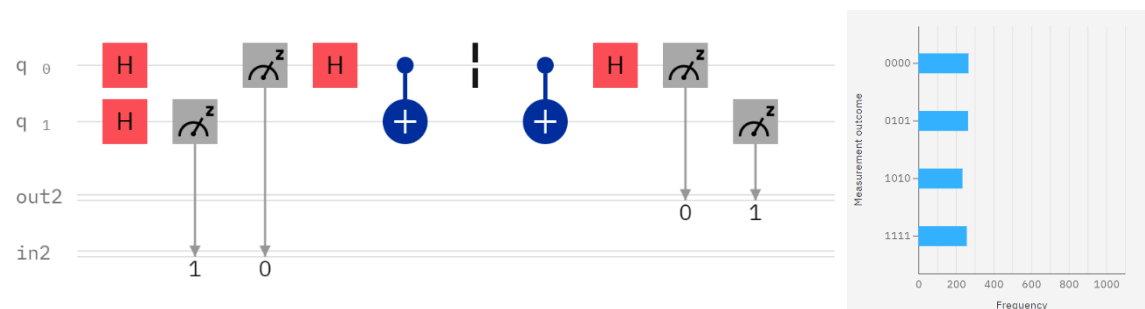


Figure 23: A Bell-State creator (left of the Barrier), which first creates two Qubits with a random distribution of  $|0\rangle$  and  $|1\rangle$  states using a Hadamard operator and a measurement. Right of the Barriers is the Bell State Measurement operator. Results are plotted to the right for 1000 runs. Note that only those measurements occur, in which the pairs of bits (in and out) are equal. This show that the algorithm does its job.

### 4.2.3 Quantum Teleportation and Related Protocols

Bell state measurements are fancier than you may think, as they allow us to implement a lot of awesome operations. While these are not at the core subject of this lecture, we shall here introduce an algorithm for quantum teleportation. The synopsis is as follows. We create two Qubits ( $q_1$  and  $q_2$ ) in a  $|\Phi^+\rangle$ -state and a third qubit  $q_0$  in a random state  $|\psi\rangle$ . Then we apply Bell State measurement on Qubit  $q_0$  and  $q_1$ . The resulting two classical bits will drive (or not drive) unitary operations in Qubit 1. After these, Qubit  $q_2$  will be in state  $|\psi\rangle$ . Hence, we have transported the quantum state  $|\psi\rangle$  from Qubit  $q_0$  to Qubit  $q_2$  (keep in mind, the state is destroyed in Qubit  $q_0$ ), hence the name quantum teleportation.

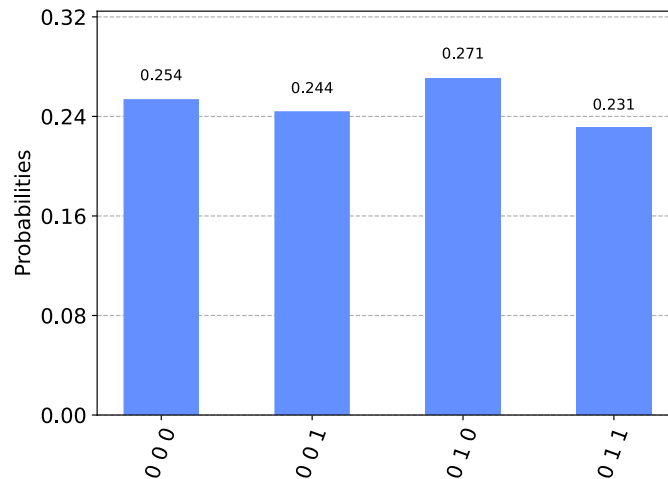
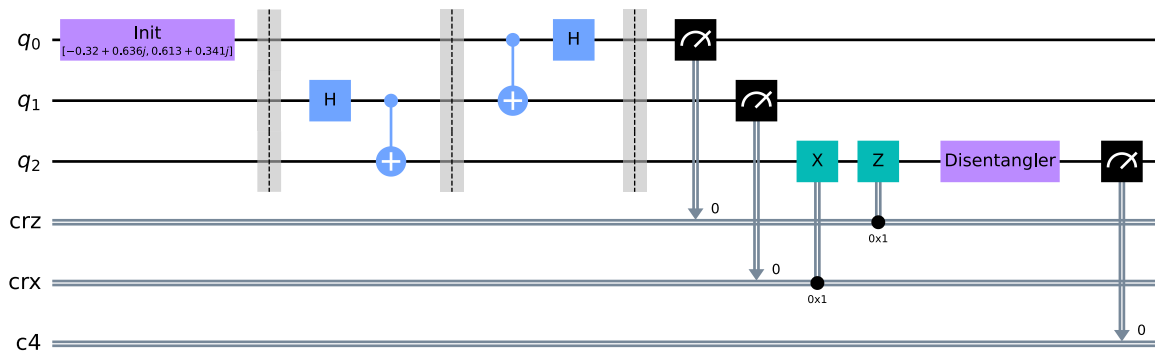


Figure 24: A Quantum Teleportation Circuit, which teleports an  $|+\rangle$  state from  $Q0$  to  $Q2$ . Note the initial state is defined in a random state using the “init” Gate. The disentangle-Gate is simply the inverse of the init-gate. The application of the Disentangler-Gate always produces the  $|0\rangle$  state as shown in the result (the leading bit is always zero) irrespective of the equally distributed output of the BS-measurement.

#### 4.2.4 Controlled U-Operations

As a next step we shall expand the scope of controlled operations. So far, we have only discussed the CNOT gate. We shall now expand the discussion to a controlled- $\hat{U}$  operation, that is, an operation that applies a single-qubit gate  $\hat{U}$  onto a target bit, if the control bit is in the  $|1\rangle$ -state and does nothing otherwise.

The common notation is:

$$\hat{C}\hat{U}(|c\rangle|t\rangle) = |c\rangle\hat{U}^c|t\rangle \quad (101)$$

Note that the controlled- $\hat{U}$  operation is not a binary do-something or do-nothing operation, unless the control bit is in a CBS. In the more general state this will enact a superposition of application and non-application of  $\hat{U}$  on the target qubit and thus leave the qubit pair in an entangled state.

From a practical point of view the question arises: how can we implement controlled- $\hat{U}$  operations? Are they new or can we break them down into well-known gates, which we have discussed prior. To do so, we need a corollary, which extends the discussion in single qubit gates from chapter 3.3.

Assume we have an arbitrary unitary gate  $\hat{U}$  then we can find unitary operators  $\hat{A}$ ,  $\hat{B}$ , and  $\hat{C}$  and  $\hat{A}\hat{B}\hat{C} = \mathbb{I}$  and a phase factor  $\alpha$  such that

$$\hat{U} = \exp(i\alpha) \hat{A}\hat{X}\hat{B}\hat{X}\hat{C} \quad (102)$$



Where  $\hat{X} = \hat{S}^2$  is the NOT gate  $\hat{X} = |0\rangle\langle 0| - |1\rangle\langle 1|$ . We will not give a proof here but the simplest way is to find specific angles in the triple-rotation theorem from chapter 3.3 that yield this result. If you read this for the first time you are guaranteed to find the corollary quite mysterious, but indeed it is very helpful, for the construction of controlled U-gates from CNOTs. Indeed we find, that

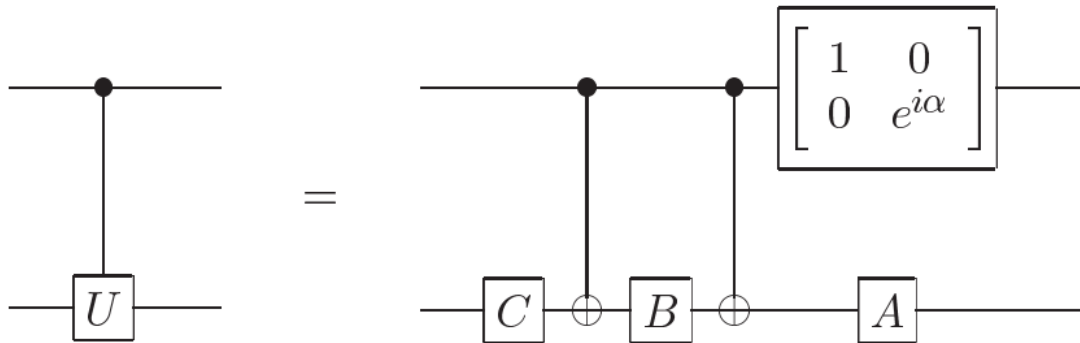


Figure 25: A controlled  $\hat{U}$ -operation in circuit notation and the equivalent gate composed of single-gate operations and CNOTs.

We verify by this in two steps. The first is phase kickback relation depicted in Figure 26:

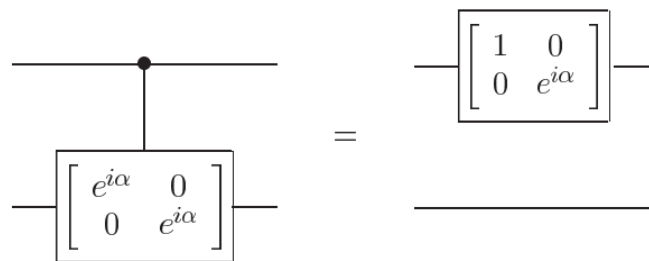


Figure 26: Phase kickback for two Qubits.

You can do so by noting that both sides map the CBS-states according to the following rules

$$|0_c 0_t\rangle \rightarrow |00\rangle \quad |01\rangle \rightarrow |01\rangle \quad |10\rangle \rightarrow \exp(i\alpha) |10\rangle \quad |11\rangle \rightarrow \exp(i\alpha) |11\rangle \quad (103)$$

The rest of the relation in Figure 25 can be shown by plugging in  $|0\rangle$  into the control qubit, which leaves  $\hat{A}\hat{B}\hat{C} = \mathbb{I}$ . If you instead plug in a  $|1\rangle$  then you get  $\exp(i\alpha) \hat{A}\hat{X}\hat{B}\hat{X}\hat{C}$ , which we have constructed to be  $\hat{U}$ .

Hence, we have seen that we can construct any single-qubit controlled U operation from CNOT and single qubit operations.

#### 4.2.5 Multiple Controls

We might require the use of multiple (bitwise connected) control operations. E.g., a control-bit may only go into the active states if multiple criteria are matched (AND) or if any of a number of criteria is met (OR). Moreover, we may require operations which do fire on a control bit being  $|0\rangle$  as opposed to being  $|1\rangle$ .

Let's begin with the AND case. Assume we have  $n$  control bits and we desire to operate  $\hat{U}$  on the  $n + 1^{th}$  qubit, if all control bits are in the  $|1\rangle$  state. This controlled gate is accordingly called the  $C^n U$  gate and its notation is:

$$\overline{C^n U}(|c_1 \dots c_n\rangle|t\rangle) = |c\rangle \hat{U}^{c_1 \dots c_n} |t\rangle \quad (104)$$

In case of  $U$  being the  $NOT$  operation we write

$$\overline{C^n X}(|c_1 \dots c_n\rangle|t\rangle) = |c\rangle |t \otimes c_1 c_2 \dots c_n\rangle \quad (105)$$

Let's start with  $n = 2$  by introducing an operator  $\hat{V}$  with  $\hat{V}^2 = \hat{U}$ . Such an operator is in principle easy to construct. It is the same rotation as  $\hat{U}$  with half the angle or you can decompose  $\hat{U}$  into eigenvector-eigenvalue pairs and divide the (phase only) eigenvalues by two. Then we can show the following relation:

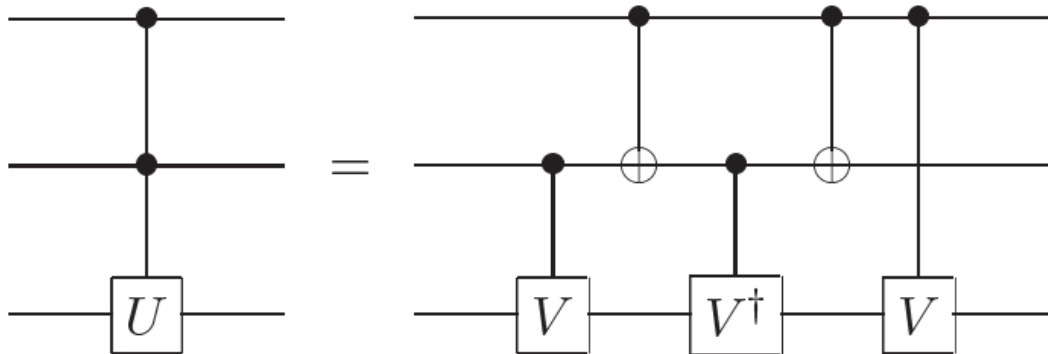


Figure 27: Decomposition of a double controlled  $U$  operation into CNOTs and square root operators.

We can easily verify this relation by plugging in all four combinations of the CBS bases into the two control lines. In the  $|00\rangle$ -case nothing ever happens. In the  $|01\rangle$  case we do nothing to the control bits and apply a  $\hat{V}\hat{V}^\dagger = \mathbb{I}$ . In the  $|10\rangle$  case we apply a double NOT to the second Qubit and a  $\hat{V}^\dagger\hat{V} = \mathbb{I}$  to the target qubit. In the  $|11\rangle$  we apply a  $\hat{V}\hat{V} = U$ .

The most important double-controlled operation is the double CNOT for which we only need to find the proper  $\hat{V}$ , which is the so-called root swap gate  $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$ :

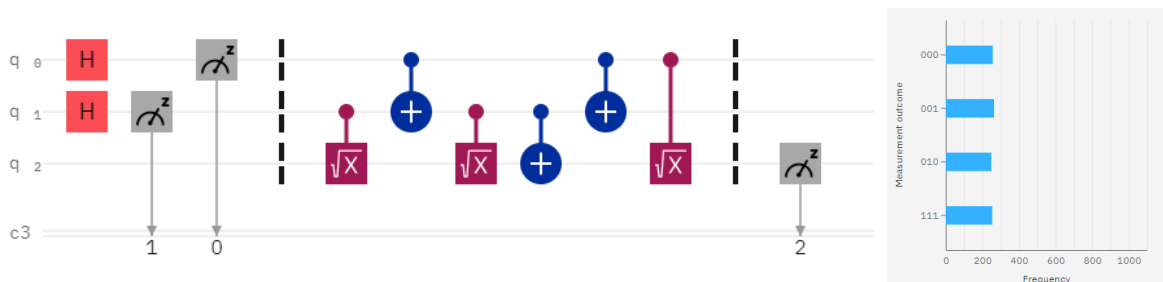


Figure 28: A double CNOT gate using composed of CNOT and SQRT-NOT gates only. The target bit is initially in the  $|0\rangle$  state and the control bit are initialized in a random manner. Note the result of target qubit  $q_2$  is always 0 unless  $q_0 = q_1 |1\rangle$ , then it is in the one  $|1\rangle$  state just as expected. Also note that QISKIT does not support the inverse of the controlled SQRT-NOT so this is implemented a controlled NOT plus a controlled SQRT-NOT (three quarters clockwise instead of a one quarter counterclockwise).

We can now generalize to an arbitrary number of inputs at the expense of a few ancilla qubits, and give a construction for an  $n = 3$  case, which may serve as a generic example:

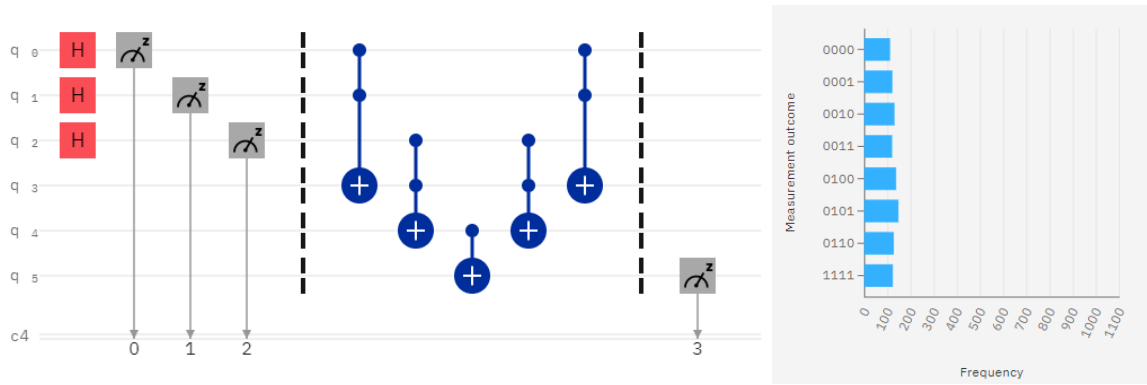


Figure 29: A triple CNOT gate using three control qubits  $q_0 \dots q_2$  two ancilla qubits  $q_3, q_4$  and a target qubit  $q_5$ . The target bit is initially in the  $|0\rangle$  state and the control bits are initialized in a random manner. Note the result of qubit  $q_5$  is always 0 unless  $q_0 = q_1 = q_2 = |1\rangle$ , then it is in the one  $|1\rangle$  state just as expected.

To construct more complex logic we introduce the inverted CNOT gate, which is defined as:

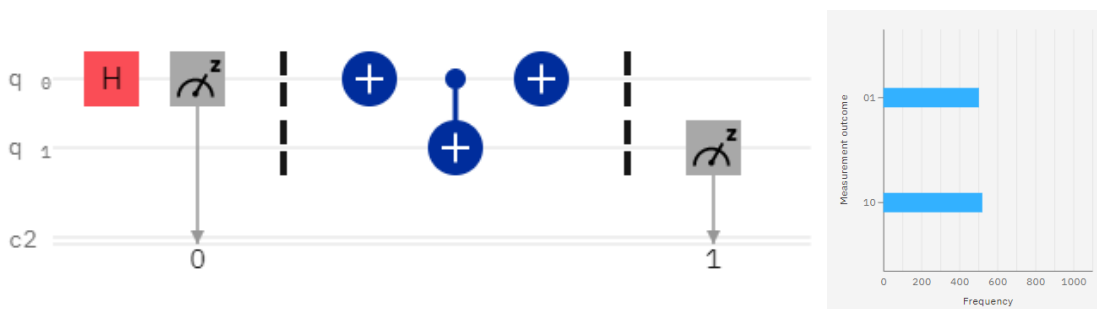


Figure 30: A  $|0\rangle$  active CNOT gate using. The target bit is initially in the  $|0\rangle$  state and the control bit is initialized in a random manner. Note the result of the target bit is only flipped if the control bit is in the  $|0\rangle$  state.

### 4.3 Classic Computation on a Quantum Computer

Quantum Algorithms are frequently used to solve problems which are formulated in the language of classical algorithms and we must find a method to make these problems accessible on a quantum computer. This issue is much more profound as you might think, because classical computers are based on irreversible operations, which you cannot – by definition – implement on a Quantum Computer.

Nevertheless classical computers are subject to the laws of quantum physics, so it would come as a great surprise, if we could not implement logical operations on a quantum computer and hence classical computation in a more general sense.

We shall tackle the issue in a two-pronged approach, by first introducing quantum logic gates (i.e. the quantum equivalents of binary logic gates) and then have a look at the consequences of superposition on classical algorithms.

#### 4.3.1 Implementing Logical Operations on a Quantum Computer

We start by taking the NAND-operation as an example. It has the following truth-table:

A	B	A NAND B
0	0	1
1	0	1
0	1	1
1	1	0

You can immediately see that this operation is not reversible, i.e. you cannot “uncompute” the A NAND B, if the result is 1 because three different inputs will produce the same result. We have previously

discussed that irreversibility leads to an increase of entropy and indeed there is a series of groundbreaking results by Landauer from the 1960s, which identify the deletion of information as the action, which increases the entropy by  $kT$ . Strange enough, this puts a lower limit on the power consumption of classical computers, but this is still a few orders of magnitude lower than the power consumption that we see today (but not so many). But I am digressing.

We chose the NAND-example for a fundamental reason, because the NAND is a universal gate for classical computers, i.e. you can construct a Turing-complete computer only from NAND-gates. Hence, in principle we only need to find a reversible implementation of the NAND gate in the quantum language to be able to port any classical algorithm into the quantum domain.

The trick with reversibility is easily achieved, by retaining in input qubits in their initial state and constructing the gate in a way that the output bit is loaded with a predefined state  $|0\rangle$ . To implement the NAND-gate we resort to the double controlled NOT gate (the toffoli-gate) from chapter 4.2.5 and turn it into a NAND by application of a NOT.

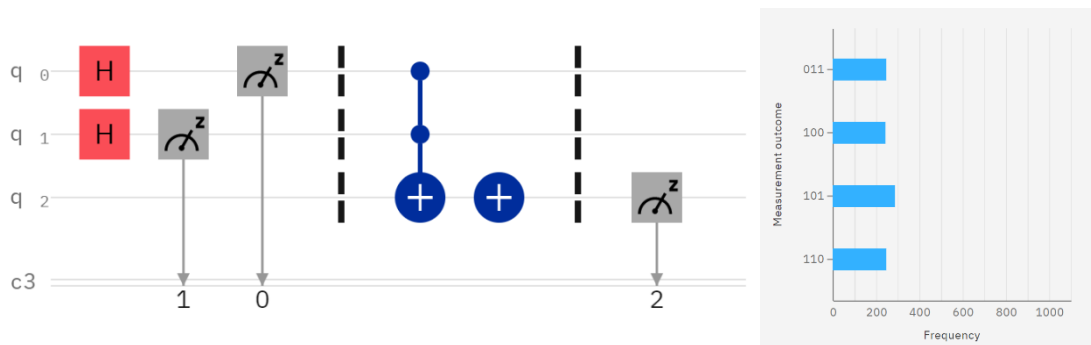


Figure 31: A reversible NAND-Gate.

We can also define an OR operator by simply using the fact that  $A \text{ OR } B = \text{NOT}((\text{NOT } A) \text{ AND } (\text{NOT } B))$ , thus we find the following layout for a quantum OR-gate:

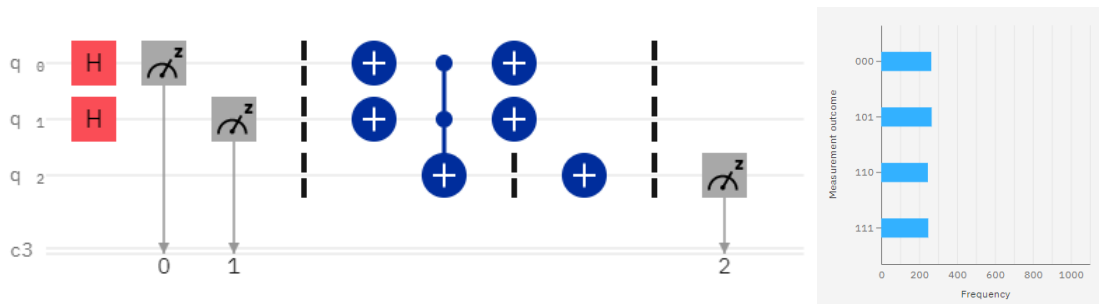


Figure 32: A reversible OR-gate. The target bit is initially in the  $|0\rangle$  state and the control bits are initialized in a random manner. Note the result of target qubit  $q_2$  is 0 if and only if  $q_0 = q_1 = |0\rangle$ . Otherwise it is flipped in the one  $|1\rangle$  state just as expected.

The last of the bunch, which is frequently employed is the XOR-gate, which the only one in the common set of logic operations that is reversible, if one input state is know, hence, we can implement it in two possible ways, the usage of which depends on the question if we need the second input for further processing:



Figure 33: Two possible ways of implementing a quantum XOR-gate. (left) Stores the result of the XOR directly in Qubit1. (right) Stores the result in Qubit2 and requires Qubit2 to be initialized in the  $|0\rangle$  state.

### 4.3.2 Classical Algorithms on Quantum Computers

Logical gates implement a complete basis set for classic computers, we therefore conclude that we can use to this to cast any classical algorithm in a quantum form. What does this mean? This means that the algorithm behaves exactly like the classical one, if we run it with CBS as an input (where the CBS are supposed to be read like binary numbers for the classical input). However, we can also run the algorithm with an entangled superposition state and retain a true quantum result. As a example we take an entirely fictitious algorithm that is fed with a four bit number. We run it twice, once with the input number "12" and once with the input "6". Suppose the algorithm us VERY hard to compute then we would, after a long wait compile the following table:

Input	Ouput
1100 (12)	1110 (14)
0110 (6)	0011 (3)

We can now turn the algorithm into a quantum version by replacing all its NANDs with their quantum equivalents and we would be guaranteed to get:

Input	Ouput
$ 1100\rangle$ (12)	$ 1110\rangle$ (14)
$ 0110\rangle$ (6)	$ 0011\rangle$ (3)

This is certainly not an improvement. However, because the quantum algorithm is necessarily linear (it is a unitary matrix!) this means that if we input a superposition of CBS-states we obtain a result, which is superimposed of both classical runs:

Input	Ouput
$\alpha 1100\rangle + \beta 0110\rangle$	$\alpha 1110\rangle + \beta 0011\rangle$

Of course, we can generalize this to all possible superpositions, if we wanted to. We are now in a position, where we can run a classical algorithm with all possible classical inputs at once! This is, however, not really useful, because upon a simple measurement in the CBS we would still collapse onto ONE particular solution of the algorithm, and we would not even know which one. So, we can only get a real advantage out of this, if what we are really looking for are not individual solutions but specific properties, which come from superpositions of solutions. Think Averaging. Think statistics. Think Fourier transformations.

### 4.3.3 A word on Uncomputation

As we have discussed there is no easy way to delete data in quantum computation, because all operations must be reversible. We have also seen that many operations require the usage of ancilla qubits

to store intermediate states of the calculation. Some quantum algorithms, as we shall see later, however require the return of all ancillas into a clean state, e.g. the rely on you to clean up all the intermediate values of your calculation, except for the final result. This process is called “uncalculation”.

For operations, which are based on the CNOT-gates, this is, in general, not more complicated than the original calculation, because you can uncalculated by recalculating the result, e.g assume that you have an intermediate QuBit, which is created by a CNOT operation. A second application will restore the ancilla bit into the previous state, because  $\hat{C}X(\hat{C}X(|c\rangle|t\rangle)) = |c\rangle|t \oplus c \oplus c\rangle = |c\rangle|t\rangle$ . Thus can be used as discussed in the image below:

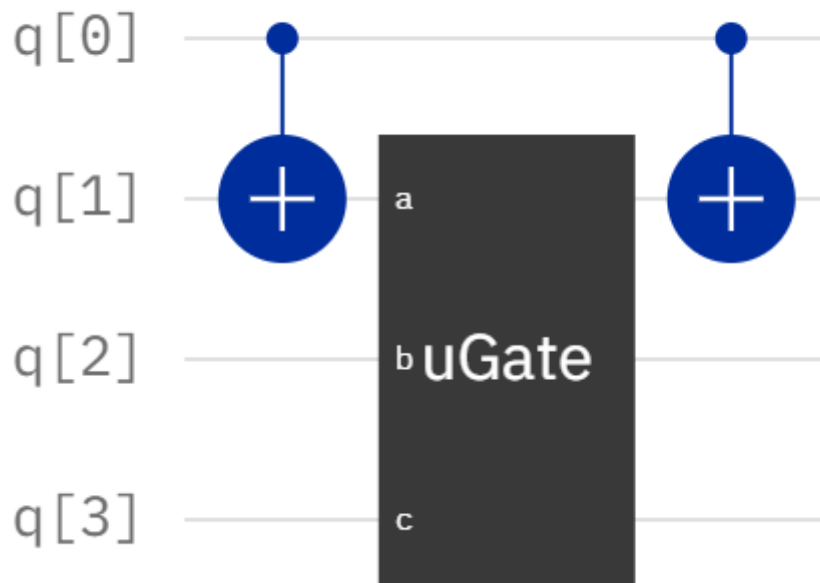


Figure 34: Some Quantum operation  $\hat{U}$  relies on the ancilla QuBit  $q_1$  for the input. It will create a result in  $q_2$  and/or  $q_3$ . After the operation is carried out, we can uncalculated  $q_1$  by recalculating it.

We shall see in the next chapter that this approach is indeed universal because CNOTs are universal.

#### 4.4 Generic Operations and Universality

So far, we have restrained ourselves to operations, which are active on a single qubit only. In general, this is not the case, and we can, of course, define gates which are active on any number of qubits or any number or combinations of parts of qubits. We shall spend this chapter to show, that even such complicated operations can be broken down into a series of single Qubit operations and CNOTs. This will then conclude the universality proof, with the result, that we can decompose any possible quantum operations in a series of CNOTs, Hadamard, and Phase Shift Gates.

We shall however see that this construction is not terribly efficient. On the other hand, it should not be terribly efficient, because we can efficiently simulate those three gates on a classical computer and if we could decompose any quantum operation efficiently into them, we could efficiently simulate a complete Quantum Computer and there would be not much to learn from them.

Any arbitrary gate is characterized by its unitary operator  $\hat{U}$ , which can be represented by a matrix. Here we shall restrain ourselves to a 3x3 matrix and introduce an algorithm, which we can use to re-

duce the  $3 \times 3$  matrix by one dimension into a series of  $2 \times 2$  matrices. The algorithm can straightforwardly be extended to any number of  $N \times N$  and by consecutive application we can use it to reduce the  $N \times N$  into a series of  $N - 1 \times N - 1$  and so on, until we are again at a series of  $2 \times 2$  matrices. The matrix is given as:

$$\hat{U} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \quad (106)$$

We next find three  $2 \times 2$  matrices  $\hat{U}_1, \hat{U}_2,$  and  $\hat{U}_3,$  such that  $\hat{U}_3 \hat{U}_2 \hat{U}_1 \hat{U} = \mathbb{I}$  and hence  $\hat{U} = \hat{U}_1^\dagger \hat{U}_2^\dagger \hat{U}_3^\dagger$ . We use the first 2 the submatrices to produce zeros in the first column below the top-left diagonal element and then the last one to produce zeros in the top row, again ignoring the top left. We start by choosing:

$$\hat{U}_1 = \begin{bmatrix} \frac{a^*}{\sqrt{|a|^2 + |b|^2}} & \frac{b^*}{\sqrt{|a|^2 + |b|^2}} & 0 \\ \frac{b}{\sqrt{|a|^2 + |b|^2}} & -\frac{a}{\sqrt{|a|^2 + |b|^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (107)$$

Which produces a zero in the first row:

$$\hat{U}_1 \hat{U} = \begin{bmatrix} a' & d' & g' \\ 0 & e' & h' \\ c' & f' & i' \end{bmatrix} \quad (108)$$

Then we set:

$$\hat{U}_2 = \begin{bmatrix} \frac{a'^*}{\sqrt{|a'|^2 + |c'|^2}} & 0 & \frac{c'^*}{\sqrt{|a'|^2 + |c'|^2}} \\ 0 & 1 & 1 \\ \frac{c'}{\sqrt{|a'|^2 + |c'|^2}} & 0 & -\frac{a'}{\sqrt{|a'|^2 + |c'|^2}} \end{bmatrix} \quad (109)$$

Which produces a second zero in the first row:

$$\hat{U}_2 \hat{U}_1 \hat{U} = \begin{bmatrix} 1 & d'' & g'' \\ 0 & e'' & h'' \\ 0 & f'' & i'' \end{bmatrix} \quad (110)$$

Since  $\hat{U}_2, \hat{U}_1,$  and  $\hat{U}$  are all unitary  $\hat{U}_2 \hat{U}_1 \hat{U}$  must be unitary, too it follows that  $d'' = g'' = 0$ . This also implies that the submatrix composed of  $e'', h'', f'', i''$  is by itself unitary and therefore  $f'' = -h''^*$ . Thus we have:

$$\hat{U}_2 \hat{U}_1 \hat{U} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e'' & h'' \\ 0 & -h''^* & i'' \end{bmatrix} \quad (111)$$

We can then simply define  $\hat{U}_3$  as a Hermitian conjugate of unitary submatrix, which we know is the inverse due to the unitarity:

$$\hat{U}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e''^* & -h''^* \\ 0 & h'' & i''^* \end{bmatrix} \quad (112)$$

And we can therefore guarantee that:

$$\hat{U}_3 \hat{U}_2 \hat{U}_1 \hat{U} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbb{I} \quad (113)$$

Thus, we have successfully reduced the  $3 \times 3$  gate into a series of three  $2 \times 2$  gates! A word of caution: we have, so far ignored the case, when  $b = 0$  or  $c' = 0$ . In such cases you can simply skip the step and set  $\hat{U}_1 = \mathbb{I}$  or  $\hat{U}_2 = \mathbb{I}$ .

If you have a larger  $N \times N$  matrix you can generalize the algorithm to work on the first column, then the second column, until all of the subdiagonal elements (except for a  $2 \times 2$  matrix) are zero. This requires  $\mathcal{O}(N^2)$  operations. If the operation spans  $n$  qubits then we have  $N = 2^n$  and thus we  $\mathcal{O}(2^{2n})$  operations. Again, this is not very efficient, but as we have discussed this is necessarily the case. Nevertheless, there are a few important subclasses, were a decomposition is in fact quite efficient; we shall discuss them in the following chapter.

There is one piece missing in the completeness proof. Although we have decomposed an  $n$ -Qubit unitary into  $\mathcal{O}(2^{2n}) = \mathcal{O}(4^n)$   $2 \times 2$  matrices  $\hat{U}_i$  this does not yet mean that we have decomposed it onto  $\mathcal{O}(2^{2n})$  single Qubit Gates, because the matrices will in general span any possible combination of CBS (e.g. they may operate on the subspace spanned by the  $|01\rangle$  and the  $|10\rangle$  CBS, which belong to two different qubits). To map that onto single qubit operations we must implement a swapping scheme first, map the two states onto the states of one specific qubit, enact the single qubit operation  $\hat{U}$  on that specific qubit and then swap everything back into place. For swapping we use controlled NOT operations in a specific manner.

Assume we have a three Qubit system and we want to implement the following operation on it:

$$\hat{U} = \begin{pmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & 0 & d \end{pmatrix} \quad (114)$$

Obviously, this operation is a  $2 \times 2$  matrix  $\hat{U}' = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ , which acts on the states  $|000\rangle$  and  $|111\rangle$ . Let's call the three qubits by the names  $q_0$ ,  $q_1$ , and  $q_2$ . Let's further write down a sequence of qubit-wise swapping operations, which transforms the  $|000\rangle$  state into the  $|011\rangle$  state, which shares the same qubit with  $|111\rangle$ , in the sense that these are the CBS of  $q_2$ . This sequence is:

Operation	Action	Explanation	where is $ 000\rangle$ the amplitude after the operation
Swap 1	Swap $ 000\rangle$ with $ 001\rangle$	Not $q_0$ under the condition that $q_2$ and $q_1$ are in the $ 0\rangle$ -state	$ 001\rangle$
Swap 2	swap $ 001\rangle$ with $ 011\rangle$	Not $q_1$ under the condition that $q_2$ is in the $ 0\rangle$ -state and $q_0$ is in the $ 1\rangle$ -state	$ 011\rangle$
Apply $\hat{U}'$		Apply $\hat{U}'$ on $q_2$ under the condition that $q_0$ and $q_1$ are in the $ 1\rangle$ -state	unaffected



Unswap2	swap $ 011\rangle$ with $ 010\rangle$	Not $q_1$ under the condition that $q_2$ is in the $ 0\rangle$ -state and $q_0$ is in the $ 1\rangle$ -state	$ 001\rangle$
Unswap1	swap $ 001\rangle$ with $ 000\rangle$	Not $q_0$ under the condition that $q_2$ and $q_1$ are in the $ 0\rangle$ -state	$ 000\rangle$

We can of course also write this as a quantum circuit:

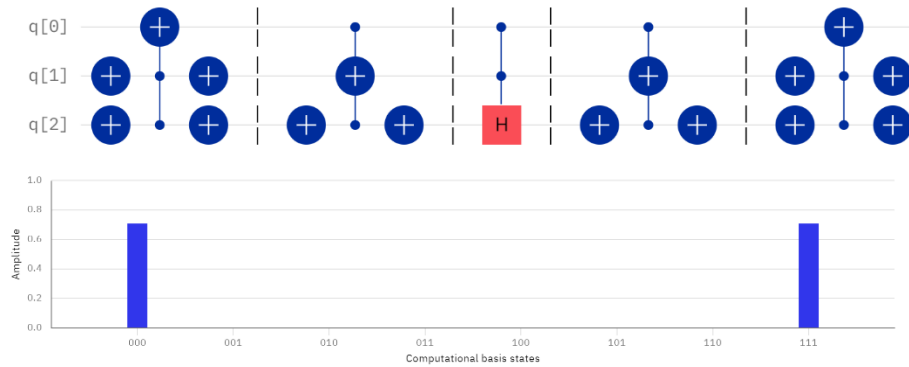


Figure 35: A swap-based implementation of an arbitrary rank-2 operator  $\hat{U}'$  acting on  $|000\rangle$  and  $|111\rangle$  based on CNOTs and single Qubit operations only. See resulting state for the illustration of the result. The barriers represent the different steps of the algorithm (Swap 1, Swap 2, Apply, Unswap 2, Unswap 1).

For any possible  $\hat{U}'$  in an  $n$ -Qubit system we may require up to  $2(n - 1)$  CNOT operations, which can be implemented with  $\mathcal{O}(n)$  operations, using only single qubits and 2-qubit CNOTs. Thus, we require up to  $\mathcal{O}(n^2)$  elementary operation to implement the entire swap sequence. Together with the previous result, we therefore conclude that we can implement an arbitrary unitary operation on an  $n$ -Qubit state with  $\mathcal{O}(n^2 4^n)$  elementary operations.

We therefore conclude:

*Summary: Any possible gate on an  $n$ -Qubit System can be implemented with a series of Hadamard,  $\frac{\pi}{8}$ , and two Qubit CNOT gates. Any universal quantum computer can be constructed if these three gates can be implemented.*

A word of caution, which should not go unmentioned. The conclusion is actually not completely true, because of quantum errors. First of all, we have discussed in section 3.3.1, we can only ever hope to approximate single Qubit gates, yielding an approximation error for every gate. Since the construction of multi-qubit gates heavily relies on replacing a few complex operations with a lot of single-qubit operations, this means that approximation errors will occur many, many times in a quantum circuit constructed from fundamental gates. Moreover, any realistic Quantum computer will add external noise sources, which will add an intrinsic error over time. The source for these errors are complicated and involved but most can be understood in the context of decoherence, which means that after a certain decoherence time, quantum interference is no longer observable and all entanglement is lost. Since any gate requires a certain process time, this means that any execution of a gate will also introduce noise-based errors.

While this seems rather bleak, there is also a beacon of hope, in the form of the error accumulation theorem. This means that a sequence of  $N$  imperfect gates  $\hat{U}_i$ , each of which produces an error  $\epsilon$ , will produce a total error that scales no worse than  $\mathcal{O}(\epsilon N)$ . We conclude that Quantum Errors are, more or less, additive and a reduction of the per-gate-error of  $\epsilon$  yields a linear increase in the number of

possible gates  $N$ , which can be implemented, before the results of the algorithm get killed of by accumulating errors.

The way to more powerful quantum computers therefore involves:

- the implementation of more Qubits
- the reduction of gate errors
- the direct implementation of more complicated gates
- the reduction of the number of gates using more efficient algorithms
- the reduction of statistical noise using error correction (which requires more Qubits)

## 5 Quantum Algorithms

In this chapter we shall return to the canonical circuit model and use it to introduce, discuss, and understand a few key algorithms in Quantum Computing. Up until three or four years ago this would have comprised an almost complete list of the algorithms which have been found and discussed on Quantum Computers but, lo and behold, the number of algorithms available for Quantum Computers grows as quickly as does their computational power.

We shall nevertheless stick to the traditional basics for two reasons. The first reason is that these algorithms are incredibly well-understood, including their limitations but also including the impact of noise on such algorithms. This is very important from an application point of view and also for the development of Quantum Computers: these classical algorithms are near-ideal to test and characterize the power of real world implementations of Quantum Computers. The second reason is that these algorithms nicely highlight some of the specific feature, which make Quantum Computers particularly powerful. As such, that can serve as a good starting point to design novel quantum algorithms. If you, like me, have are accustomed to writing classical computer programs you will see that quantum software does not seem to naturally come about. A proper analysis may give us the kind of natural understanding of the strengths of Quantum Computers and the essential building blocks of quantum software such that we may hope to end of with the ability to come up with novel way of applying Quantum Computers.

### 5.1 Josza-Deutsch's Algorithm: a Case of Useless but Powerful

The first algorithm which we will discuss was also the first algorithm even to be developed specifically for Quantum Computers. To be more precise: it was custom-designed as a demonstration for Quantum Advantage, i.e. it gives the solution to a very artificial problem, which scales much more efficiently on a Quantum Computer as opposed to a classical computer.

Assume the following problem: you play a game with a friend of yours, that is located somewhere in a small village in rural Thuringia. It's one of those places, where mobile reception is nil; cable-based internet keeps on breaking down constantly and you can't travel because it's Corona-lockdown. Again. So, you have to resort to writing letters back and forth (you know: pieces of written paper stuck in an envelope, like they used to do in the 19<sup>th</sup> century), which is slow and expensive.

The friend of yours has invented a mathematical function  $f$ , which inputs a (binary) number  $x$  from, say  $x \in \{1, 2^n\}$  and returns a single bit, e.g.  $f(x) \in \{0, 1\}$ . The function is guaranteed to be either constant or even. Constant means that either  $\forall x: f(x) = 0$  or  $\forall x: f(x) = 1$ . Even means that there exist exactly  $2^{n-1}$  distinct values for  $x$  for which  $f(x) = 0$  and equally many for which  $f(x) = 1$  but you don't know in advance which ones.

Your part in the games is to find out as fast as possible: is the function constant? Or is it even? There is a catch, however. You are only allowed to ask the result for one specific input in a single letter. E.g. one letter may read  $x = 63$  and the answer would be  $f(x = 63) = 0$ .

As you see: we have constructed the game in such a way that the evaluation of the function is very expensive (in terms of time) because you have to write a separate letter for each evaluation and wait for the answer to arrive. This is a bit artificial but in reality, the function may just be very hard and expensive to compute or  $n$  may just be a very large number under which circumstances the number of letters may even be too much to handle for a very fast postal service. What matters more is: we are not interested in the specifics of a particular game, rather in the cost that a solution to this game would incur as a function of the number of bits  $n$  in general.

The classical solution to this problem is indeed quite simple. You start with some value of choice, say  $n = 0$  and ask your friend the result  $f(x = 0)$ . Then you go on and ask  $f(x = 1)$  et cetera and compare the results. If  $f(0) \neq f(1)$  then you know the function is not constant and thus even. However, if  $f(0) = f(1)$  you can't make a statement because the function may be constant or it may be even and you have just happened to select two specific values of  $x$  that produce the same result. You would then go on to  $x = 2, 3, 4 \dots$  and so on. If you keep on getting the same results you end up stuck in the same dilemma as you cannot guarantee that the function is constant unless you have checked more than half of the possible inputs, e.g. until you have progressed to  $x = 2^{n-1} + 1$ . Thus, we find that the solution to the algorithm may require  $\mathcal{O}(2^n)$  (expensive) steps for a solution and is thus very, very inefficient.

Keep in mind that the inefficiency experience above is of an extremely annoying type. We have to make a shitload of function evaluations and we don't even care about any of the specific results. All we care about is a – to some degree – averaged result over a large subset of possible inputs. If you remember the last paragraphs of chapter 4.3, you may start to feel that Quantum Computers may be a good thing to apply here. If you don't, then just bear with me anyway.

First we'll construct the quantum equivalent of the function  $f$  to be evaluated by turning it a unitary operation  $\hat{U}_f$ , which operated on the set of input Qubits  $|\mathbf{x}\rangle$  and the result Qubit  $|y\rangle$ . Keep in mind that  $|\mathbf{x}\rangle = |x_1 \dots x_n\rangle$  is a number of qubits, equivalent to the number of bits that may be input into the function  $f$ . As we may encounter this quite frequently, we will try and use the boldface notation whenever we feel that it is required for notational clarity. And also keep in mind  $\oplus$  is the XOR operation which can be implemented using a simple CNOT.

$$\hat{U}_f(|\mathbf{x}\rangle|y\rangle) = |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle \quad (115)$$

We start the algorithm with the initial CBS state:

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle \quad (116)$$

This, however, is not helpful for the computation, which we would like to carry out, as it would simply evaluate the function  $f(x = 0)$  at one specific value; i.e. it would do the same thing as a classical computer would do. The same is true for any other CBS on the  $|\mathbf{x}\rangle$ -part of the Qubit. Instead we are looking for a compound property; i.e. we would like to evaluate the function at as many input bits, as we possibly can, and this can be done by transforming  $|\mathbf{x}\rangle$  into a balanced superposition of all possible CBS. Luckily this is a simple task, which can be achieved by applying Hadamard-Gates onto each and every Qubit of  $|\mathbf{x}\rangle$ . For good measure we also apply the Hadamard onto the result qubit.

$$|\psi_1\rangle = (\hat{H}|0\rangle)^{\otimes n} \hat{H}|1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \left(\sum_{x=1\dots 2^n} \frac{|x\rangle}{\sqrt{2^n}}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \quad (117)$$

Next we evaluate the function on this register and obtain:

$$\hat{U}_f |\psi_1\rangle = \left(\sum_{x=1\dots 2^n} (-1)^{f(x)} \frac{|x\rangle}{\sqrt{2^n}}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \quad (118)$$

Because  $(|0\rangle - |1\rangle) \oplus 1 = |1\rangle - |0\rangle = -(|0\rangle - |1\rangle)$ . This is a noteworthy result in its own right because this means that the result of the evaluation of  $\hat{U}_f$  is not stored in the Qubit value of  $|y\rangle$  but in the phases of the computation basis states  $|x\rangle$  of the input register. This, somewhat unexpected intermediary result, sheds light on a rather fundamental property of Quantum Computation: compared to classical computation there is no differentiation of input and output registers, whatsoever. This is due to the global nature of the wavefunction and the reversibility of the computational paradigm.

Of course, we can't measure the phases of the input registers directly. So what to do with this result? Think physics: the generic way of measuring phases is by measuring interference, using beam splitters. The Quantum Computer equivalent is the application of the Hadamard operator and this is just what we do: we  $\hat{H}$  to all of the input register Qubits again:

$$|\psi_2\rangle = \hat{H}^{\otimes n} \left(\sum_{x=1\dots 2^n} (-1)^{f(x)} \frac{|x\rangle}{\sqrt{2^n}}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \quad (119)$$

The calculation of the result is a tiny bit cumbersome and we'll do it separately by each of the elements of the sum. Keep in mind that  $|x\rangle$  is any possible CBS, e.g.  $|x\rangle = |27\rangle = |0001\ 1011\rangle = |0\rangle|0\rangle|0\rangle|1\rangle|1\rangle|0\rangle|1\rangle|1\rangle$ . From this we get:

$$\hat{H}^{\otimes n} |x\rangle = \sum_{x'=1\dots 2^n} (-1)^{x \cdot x'} \frac{|x'\rangle}{\sqrt{2^n}} \quad (120)$$

Where  $x \cdot z$  is the bitwise inner product modulo 2 of  $x$  and  $z$ , e.g. if  $x = 27$  and  $x' = 15$  we have  $27 \cdot 15 = 0001\ 1011 \cdot 0000\ 1111 = (0 + 0 + 0 + 0 + 1 + 0 + 1 + 1) \bmod 2 = 3 \bmod 2 = 1$ . We can now evaluate what happens to our wavefunction:

$$|\psi_2\rangle = \left(\sum_{x'=1\dots 2^n} \sum_{x=1\dots 2^n} (-1)^{x \cdot x' + f(x)} \frac{|x'\rangle}{2^n}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \quad (121)$$

Now we observe the query register. This will force the superposition state to collapse into any of the CBS-states. Let's check for the probability of the  $|0\rangle^{\otimes n}$  state first, e.g. we are looking for the amplitude with  $x' = 0$ . It's amplitude is:

$$\sum_{x=1\dots 2^n} (-1)^{f(x)} \frac{1}{2^n} = \begin{cases} -1 \Leftrightarrow f(x) = 0 \\ +1 \Leftrightarrow f(x) = +1 \\ 0 \Leftrightarrow f(x) \text{ is even} \end{cases} \quad (122)$$

Thus if the function  $f$  is constant a result of  $|0\rangle$  is observed with  $p = 1$ . If, however, the function is even then one of the resulting Qubits is certain to produce a nonzero result. So we can answer the initial question by just checking, whether the result is zero, then we have a constant function or if it is nonzero, then we have an even function.

Keep in mind that in the procedure we have only applied the function  $f$  once and thus we have found a Quantum Algorithm that solves Deutsch-Josza's problem with  $\mathcal{O}(1)$  evaluations of  $f$  and  $\mathcal{O}(n)$  quantum gates altogether. This is a tremendous speedup if compared to the  $\mathcal{O}(2^n)$  for the classical solution and showcases the power of the Quantum Computer.

Of course, we shall also give you a proper circuit diagram and have it run on a (simulated) Quantum Computer and we'll start with two implementations of the constant case:

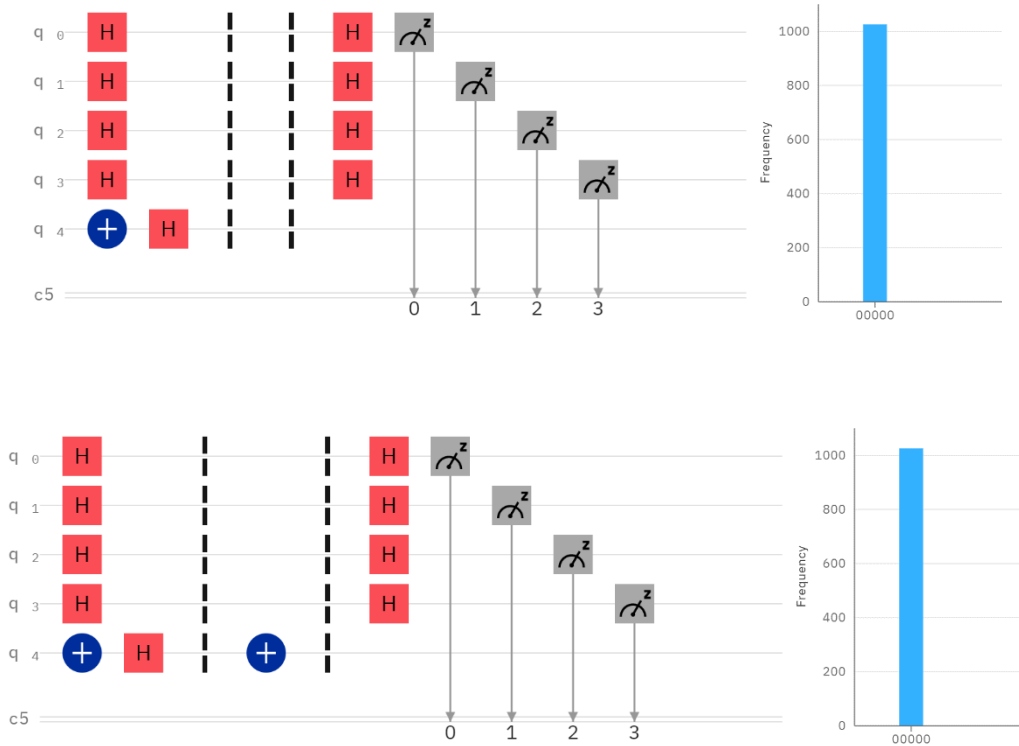


Figure 36: Josa-Deutsch with a constant function  $f$ . At the top the function is  $f(x) = 0$ , whereas at the bottom it is  $f(x) = 1$ . As expected the results are exclusively  $|00000\rangle$ .

Let's now move to the even case, by flipping the resulting qubit, if the last input register is in the  $|1\rangle$  state, which happens in exactly 50 of the cases. We'll do so first for a error-free quantum simulator and then for a real quantum computer which has exactly  $N = 5$  Qubit and a fairly low error rate.

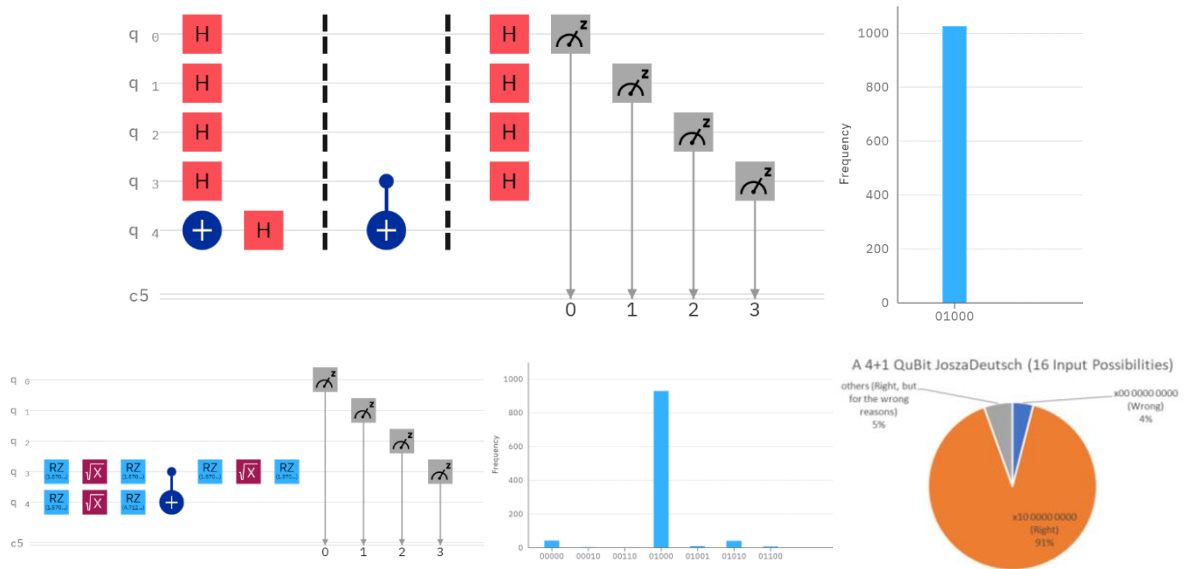


Figure 37: Josa-Deutsch with an even function  $f$ . Where  $f = 0$  if  $q_3 = |0\rangle$  and  $f = 1$  if  $q_3 = |1\rangle$ . At the top is the plain vanilla implementation and the result on a quantum simulator. As expected, the results are never  $|0000\rangle$ . At the bottom is the transpiled version, which was run on a proper QC (IBM Santiago) and the results which are correct roughly 93% of the time.

Using 4 input register Qubits we have therefore run through  $2^4 = 16$  possibilities and of course the speedup is still very...minimal. Let's take this to the next level and use publicly available QC with the largest number of Qubits that is available at the moment; this one has 15 Qubits but the gates are not of particularly high quality. At and rate we still calculate a task, which does otherwise require  $2^{14} = 16653$  individual queries.

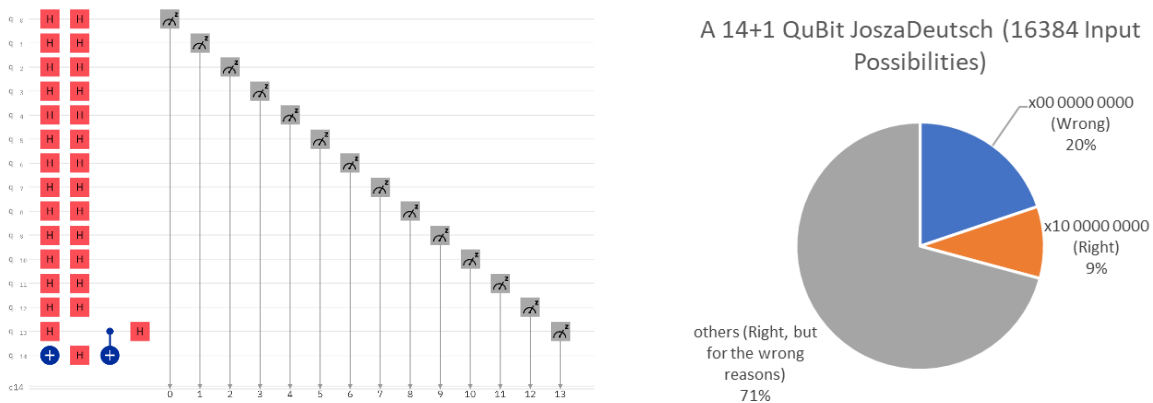


Figure 38: Even Josa-Deutsch on 15 Qubit machine (IBM\_melbourne). The machine has a fairly low Quantum Volume and hence the algorithm produces wrong results most of the times. Only 9% of the runs produce correct results.

The results are pretty disappointing; we get the proper results in only 9% of the cases and we have not even used a particularly complicated function  $f$ ; Josa-Deutsch is of course particularly interesting if exactly this is case; namely if  $f$  is difficult to compute.

After marvelling on the tremendous speedup I'd like to add two afterthoughts. While the classical solution I have presented you above is the most straightforward one, it is not the most elegant so to say. An arguably more elegant classical approach would be to simply calculate the average over all possible solutions  $\frac{1}{2^n} \sum_x f(x)$ . If the solution is equal to 0 or 1 then we know the function is constant; if the solution is anything else, then the function is even. Keep in mind that we have previously discussed that Quantum Parallelism is very good for the calculation of momenta of functions and the average is,

of course, the simplest moment. Hence, what we have done is somewhat related to a quantum version of calculating the average.

The second thought that comes to mind is that one way of calculating the average is the Fourier transformation. The if  $\tilde{f}(k)$  is the Fourier transform of  $f(x)$  then we know, that  $\tilde{f}(k = 0)$  is the average. As the fourier transform is a tremendously powerful tool in mathematics, this begs the question: is there an (efficient) quantum version of the Fourier transformation? And if yes, what can we use it for?

## 5.2 Quantum Fourier Transformation: Divide et Conquera

So let's dive right in, after this flawless transition into the Quantum Fourier Transformation Algorithm. Because we are in the realms of QuBits we shall, of course, think strictly about the discrete fourier transformation of function with  $N = 2^n$  entries:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jk/N} x_j = 0 \quad (123)$$

The difference here being that the numbers  $x_j$  and  $y_k$  are supposed to be the amplitudes of the corresponding CBS  $|j\rangle$  and  $|k\rangle$  before and after the application of the Fourier transform operator, e.g.

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle \quad (124)$$

The operator thus must act on the CBS in the way:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{jk}{N}} |k\rangle = 0 \quad (125)$$

While this is all nice and well it is a pretty useless formulation for a quantum computer, because it is written in terms of sums of phase shifts that have to be acquired for individual CBS and, as we now know, this is not well-implemented in a QC. If this is no obvious from the equation above, you can also write down the matrix  $\hat{U}$  for a, say three Qubit QFT, which is:

$$\hat{U} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{pmatrix} \quad (126)$$

Were we have introduced the abbreviation  $\omega = \exp\left(\frac{2\pi i}{8}\right)$ .

As from the last chapters, we know, that this may not be an operation, which can be easy or efficiently implemented in a Quantum Computer. So, does that mean, that Fourier transformations are per-se not efficiently implementable on a Quantum computer? Actually, this could not be further from the truth. But what we really need to do is to reformulate the above equation in a way, that we can implement into a series of controlled controlled and single QuBit operations, where each operation acts on the basis states  $|0_k\rangle$  and  $|1_k\rangle$ , whereas the details of the operation in question may depend on the specific computational state, represented by the index  $j$  (in the sense of some type of control). This means, we must strive to reformulate the equation into a type of equation that looks like:

All notes subject to change, no guarantee to correctness, corrections welcome.

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \bigotimes_{k=1}^n \hat{U}_k(|0_k\rangle, |1_k\rangle, j) \quad (127)$$

To achieve this, it is helpful to think of the CBS-indicies  $j$  and  $k$  as binary numbers composed of the binary digits  $j_{1\dots n}$ , e.g.  $j = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_{n-1} 2^1 + j_n 2^0$ . We shall also introduce the binary fraction representation of the index  $j$  as  $0.j_1 j_{l+1} \dots j_m = \frac{j_l}{2} + \frac{j_{l+1}}{4} + \dots + \frac{j_m}{2^{m-l+1}}$ . Using these notations we find that we can indeed rewrite the fourier transformation into a series of controlled single Qubit operations, using the following series of transformations:

$$\begin{aligned} |j\rangle &\rightarrow \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j \frac{k_1 2^{n-1} + k_2 2^{n-2} + \dots + k_n 2^0}{2^n}} |k_1 k_2 \dots k_n\rangle \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j (\sum_l k_l 2^{-l})} |k_1 k_2 \dots k_n\rangle \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 \bigotimes_{l=1}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} \left( \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 |0_1\rangle \bigotimes_{l=2}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle + \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j 2^{-1}} |1_1\rangle \bigotimes_{l=2}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right) \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} (|0_1\rangle + e^{2\pi i j 2^{-1}} |1_1\rangle) \sum_{k_2=0}^1 \dots \sum_{k_n=0}^1 \bigotimes_{l=2}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} (|0_1\rangle + e^{2\pi i j 2^{-1}} |1_1\rangle) \left( \sum_{k_3=0}^1 \dots \sum_{k_n=0}^1 |0_2\rangle \bigotimes_{l=3}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle + \right. \\ &\quad \left. \sum_{k_3=0}^1 \dots \sum_{k_n=0}^1 e^{2\pi i j 2^{-2}} |1_2\rangle \bigotimes_{l=3}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right) \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} (|0_1\rangle + e^{2\pi i j 2^{-1}} |1_1\rangle) (|0_2\rangle + e^{2\pi i j 2^{-2}} |1_2\rangle) \sum_{k_3=0}^1 \dots \sum_{k_n=0}^1 \bigotimes_{l=3}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\ &\rightarrow \frac{1}{2^{\frac{n}{2}}} (|0_1\rangle + e^{2\pi i j 2^{-1}} |1_1\rangle) \dots (|0_n\rangle + e^{2\pi i j 2^{-n}} |1_n\rangle) \end{aligned} \quad (128)$$

If we now also decompose the index  $j$  into its bitwise representation and we note that  $e^{2\pi i j}$  is periodic in the non-fractional parts of  $j$  (e.g.  $e^{2\pi i(1.5)} = e^{2\pi i(0.5)}$ ) we come to the following useful expression

$$|j_1 \dots j_n\rangle \rightarrow \frac{1}{2^{\frac{n}{2}}} (|0_n\rangle + e^{2\pi i 0.j_n} |1_n\rangle) \dots (|0_1\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1_1\rangle) \quad (129)$$

Keep in mind that, just to confuse you, we have swapped the order of the factors to adhere with the standard notations of having the lowest index Qubits to the right of the equation. It is useful because the representation is in a product form that tells us exactly, what we have to do to each qubit in order to implement the quantum fourier transform. The approach is quite simple; we

1. Apply a Hadamard operator  $\hat{H}$  to each qubit to construct the transformation  $|j_k\rangle \rightarrow \frac{1}{\sqrt{2}} (|0_k\rangle \pm |1_k\rangle)$



- Apply a phase shift of  $\exp(2\pi i/2^{l-k})$  to the  $|1\rangle$  state using the operator  $\hat{R}_{l-k} = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^{l-k}} \end{bmatrix}$  if any of the of higher Qubits  $|l > k\rangle$  are in the  $|1\rangle$  state (or in other words, conditionally on  $|l\rangle$ ).
- Repeat for all Qubits

Here's the circuit representation for a four qubit case:

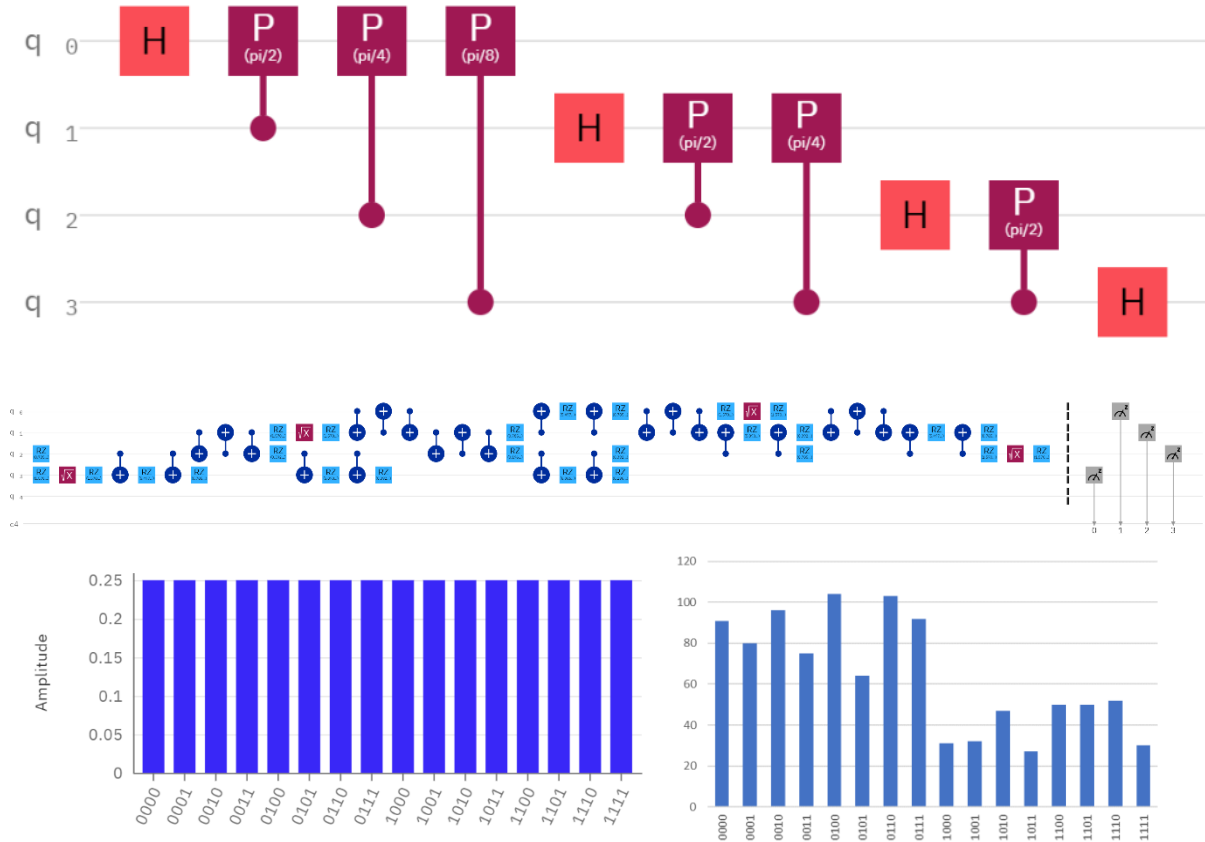


Figure 39: (Top) Circuit Representation of a 4 QuBit QFT. (Middle) Transpiled circuit on the 5-Qbit IBM\_Athens machine. (bottom, left) Results on a Quantum Simulator. As expected the  $|0000\rangle$ -state, which is equivalent to a single  $\delta$ -peak at  $x = 0000$ , is transformed into a an equal superposition of all plane waves. (bottom, right) Result from the 5-Qbit IBM\_Athens machine. Note that the circuit depth is way beyond that what the QC can do and the results are more or less random.

Now let's sit back, relax and have some fun. We'll use a 6-bit version of the QFT to create sine waves on  $N = 2^6 = 64$  positions by superimposing two  $\exp(i \dots)$  functions. We can do so in the high bits to create low frequency waves:

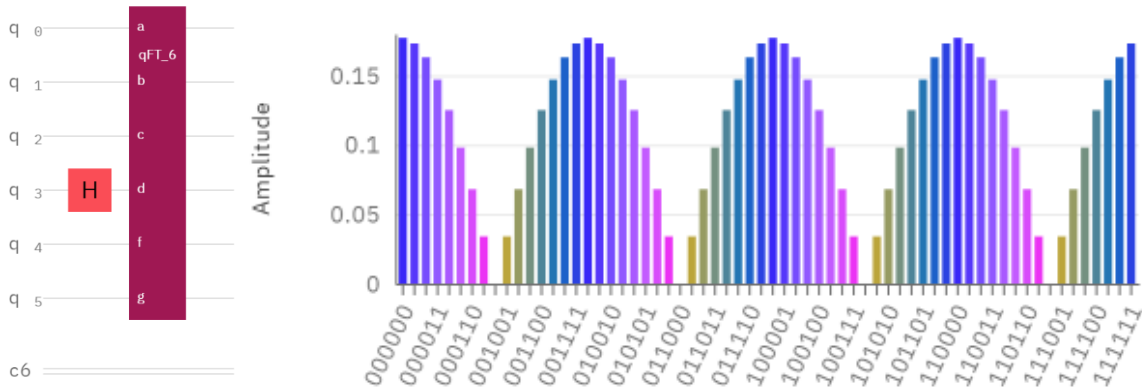


Figure 40: (left) Circuit Representation of a 6-Qubit QFT used to create a low-frequency sine wave by seeding the input of the QFT with two delta peaks  $|0\rangle|0\rangle(|0\rangle + |1\rangle)|0\rangle|0\rangle|0\rangle$ . Note that the red box contains the complete QFT logic. (right) Probability amplitudes showing the sine behaviour as expected (note: color=phase).

Or in the low bits to create high frequency waves:

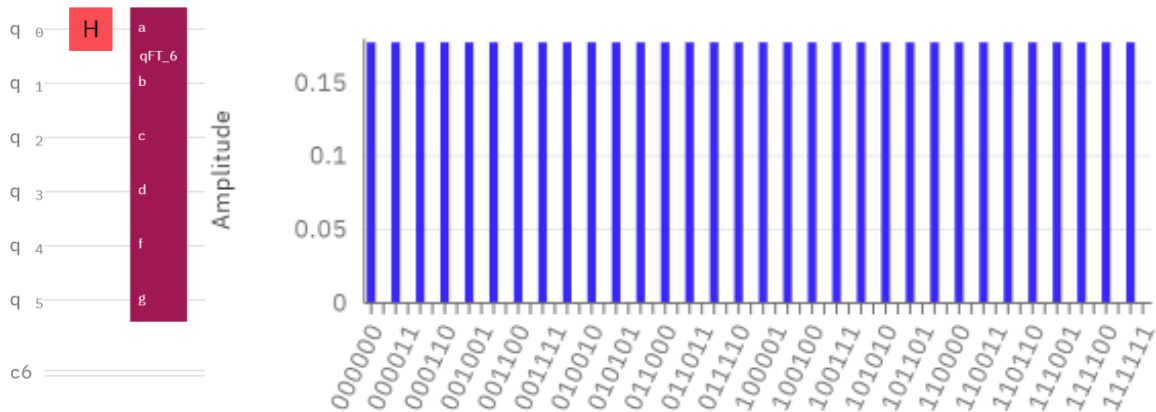


Figure 41: (left) Circuit Representation of a 6-Qubit QFT used to create a high-frequency sine wave by seeding the input of the QFT with two delta peaks  $|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle(|0\rangle + |1\rangle)$ . Note that the red box contains the complete QFT logic. (right) Probability amplitudes showing the sine behaviour as expected (note: color=phase).

We can also superimpose an equal superposition of waves to retain a  $\delta(x = 0)$ -peak:

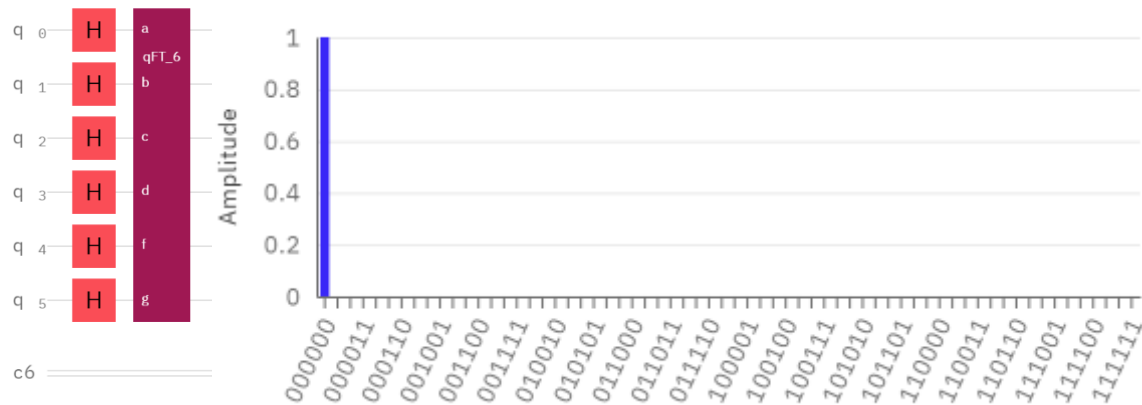


Figure 42: (left) Circuit Representation of a 6-QuBit QFT used to create a delta-peak sine wave by seeding the input of the QFT with an balanced superposition of all possible waves  $(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)$ . Note that the red box contains the complete QFT logic. (right) Probability amplitudes showing the peak-behaviour as expected (note: color=phase).

Now, let's take a look at efficiency. Keep in mind this is important. Nobody gives a damn, if Quantum Computers can calculate a QFT, as a normal FFT is already very efficient, namely it requires  $\mathcal{O}(N \log N)$  computations steps to Fourier-Transform a function with  $N$  elements. The QFT operates on  $n$  QuBits requires  $\mathcal{O}(n)$  Hadamard operations and  $\mathcal{O}(n^2)$  controlled rotation operations. As per chapter 4.2.4 each of these operations requires four single Qubit operations and three CNOTs, so the altogether required number of gates is  $\mathcal{O}(n^2)$ . Keep in mind that with  $n$  QuBits we can describe a function which has  $2^n$  entries and thus  $n = \log N$  and therefore, the entire QFT scales as  $\mathcal{O}(\log^2 N)$ . The QFT thus provides an exponential speedup over the FFT, which is a tremendous result.

There is a (major) catch, however. While the FFT produces the result of the discrete Fourier transform as a series of numbers, we here have the result only in the quantum amplitude. A measurement would collapse the result onto a single CBS and the measurement of the entire Fourier transform would require many, many measurements and an equal number of computations of the QFT. We therefore cannot straight up use quantum parallelism to replace all FFTs with QFTs and end up with a tremendous performance boost. Instead, we must use the QFT as an intermediate step, which is then mapped onto a specific observable, which is of interest in the context of specific algorithms. These must make sure to concentrate the entire amplitude of the QFT in a single (or a few) CBS and the solution to the algorithm must boil down to the question: "which CBS" is the entire wavefunction concentrated in. In physical words: we must create algorithms in such a way, that the solutions are embedded in resonances of the algorithmic structure; then we can use the QFT to find locate these resonances precisely.

### 5.3 Quantum Phase Estimation: Eigenvalue where Art Thou?

Before we finally make the move towards the infamous algorithms of Shor and Grover we shall discuss a rather nifty mathematical problem, which has gazillions of applications, particularly in physics. The name of the Algorithm is quantum phase estimation, but this is really all about eigenvalue decomposition.

We'll start with the (somewhat arbitrary and, as you shall soon see, also unnecessary) assumption that we know an eigenstate  $|u\rangle$  to a Unitary operator  $\hat{U}$ , but we don't know the eigenvalue. Of course, with the  $\hat{U}$  being unitary, we can guarantee that the eigenvalue is located somewhere on the complex unit circle and that we can represent it with a phase  $\varphi$ , e.g. the eigenvalue takes the form  $\exp(2\pi i \varphi)$ . Hence the name "phase estimation".

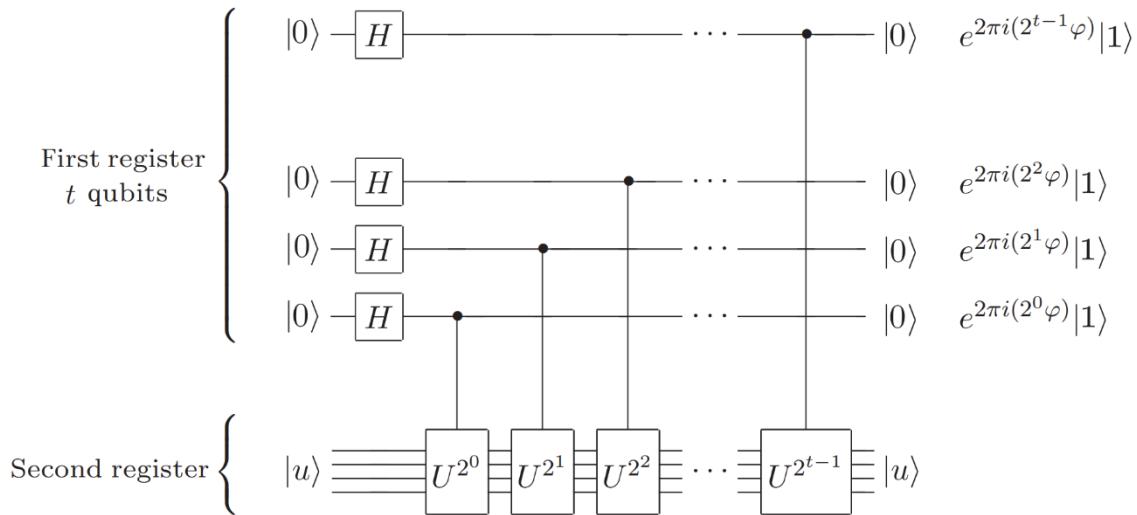


Figure 43: Abstract circuit representation of the Quantum Phase Estimation algorithm.

The algorithm requires two registers of Qubits. The first register of  $t$  Qubits shall be initialized in the  $|0\rangle$ , whereas the second register is prepared in the state  $|u\rangle$  and requires as many Qubits as are needed to hold this eigenstate and to operate  $\hat{U}$ . In a first step we apply Hadmard operators  $\hat{H}$  onto each of the  $t$  Qubits of the first register, bringing this into the state:

$$|\psi_1\rangle = (\hat{H}|0\rangle)^{\otimes t} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes t} \quad (130)$$

We have omitted the state of the  $|u\rangle$  register from this notation, as there is no appreciable impact of this operation on the  $|u\rangle$ -state.

In the second step we apply a series of controlled  $\hat{U}^{2^j}$  operations onto the  $|u\rangle$  register. Here  $j$  is running from 0 to  $t - 1$  and for each step the  $j^{\text{th}}$  Qubit acts as the control register. Keep in mind that  $|u\rangle$  is an eigenvalue to  $\hat{U}$  and applying  $\hat{U}^{2^j}$  thus does nothing but changing the phase of  $|u\rangle$ , e.g.

$$\hat{U}^{2^j}|u\rangle = \exp(2\pi i 2^j \varphi) |u\rangle \quad (131)$$

However, we don't just apply  $\hat{U}^{2^j}$ , we use the  $j^{\text{th}}$  Qubit as the control for the application. This enacts the phase kickback effect discussed in chapter 4.2.4 and in Figure 26. This means that the phase acquired by the  $|u\rangle$ -register is transferred onto the  $|1\rangle$ -state of the control register, whereas the  $|u\rangle$ -Register is again unchanged. Thus, we end up with the first register in the state:

$$|\psi_2\rangle = \left(\frac{|0\rangle + e^{2\pi i 2^{t-1} \varphi} |1\rangle}{\sqrt{2}}\right) \dots \left(\frac{|0\rangle + e^{2\pi i 2^1 \varphi} |1\rangle}{\sqrt{2}}\right) \left(\frac{|0\rangle + e^{2\pi i 2^0 \varphi} |1\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i \varphi \cdot k} |k\rangle \quad (132)$$

This is clearly a plane wave with phase gradient  $\varphi$  and we make use of the Quantum Fourier transformation on the first register to retain a  $\delta(x = \varphi)$ -function whose location and thus phase we can measure with certainty. What does this mean in terms of Qubits-however? To understand this a bit better, it makes sense to decompose  $\varphi$  into a binary fraction, e.g.  $\varphi = 0.\varphi_1\varphi_2\dots\varphi_t$ . Note that we can guarantee that  $\varphi < 1$  without loss of generality because of the  $2\pi$  ambiguity of phases. Then our state is simply:

$$|\psi_2\rangle = \left( \frac{|0\rangle + e^{2\pi i 0 \cdot \varphi_t} |1\rangle}{\sqrt{2}} \right) \dots \left( \frac{|0\rangle + e^{2\pi i 0 \cdot \varphi_2 \dots \varphi_t} |1\rangle}{\sqrt{2}} \right) \left( \frac{|0\rangle + e^{2\pi i 0 \cdot \varphi_1 \dots \varphi_t} |1\rangle}{\sqrt{2}} \right) \quad (133)$$

The inverse Fourier transformation can be retained from the Fourier transformation in the last chapter, just with the order of the gates reversed and all the phases negated. And the resulting state after the Fourier transformation is:

$$|\psi_3\rangle = |\varphi_t \dots \varphi_1\rangle |u\rangle \quad (134)$$

When we then measure the first register we are guaranteed to find the register in the CBS, which corresponds to the phase  $\varphi$ .

There are two important generalizations, which we have to make for this algorithm to be useful. The first is related to precision. Obviously, the phase  $\varphi$  is not guaranteed to have a value that can be written in a sufficiently short binary fraction to be completely representable with a given number of Qubits  $t$ . Assume for example that we have  $t = 3$ , thus a phase of  $\varphi = \frac{1}{2}$  can be represented as  $\varphi = 0.100$ , whereas  $\varphi = \frac{1}{2} + \frac{1}{16}$  cannot because it sits right in the middle between  $\varphi = 0.100$  and  $\varphi = 0.101$ . In this case the resulting wavefunction will be in a weighed superposition of  $|100\rangle$  and  $|101\rangle$  and you will measure either  $\varphi = 0.100$  or  $\varphi = 0.101$  depending on your luck. Or more general, one can show that the algorithm is likely to produce a good estimate  $\tilde{\varphi}$  to the real solution  $\varphi$ . The estimate is accurate to  $n$  bits with a success probability of at least  $1 - \epsilon$ , if

$$t = n + \log\left(2 + \frac{1}{2\epsilon}\right) \quad (135)$$

Which means that for any given success probability  $1 - \epsilon$  an increase in  $t$  goes one-to-one into an exponential increase in precision.

The second generalization is related to the requirement to beforehand know  $|u\rangle$ , which is quite useless because if you need to calculate eigenstates on a classical computer you usually get the eigenvalue for free. Assume that we don't know any eigenvector and just supply the QC with a random input state  $|\psi\rangle = \sum_u c_u |u\rangle$ , which can be, of course, decomposed into a superposition of eigenstates. The algorithm itself is linear so one can show that the resulting state is a superposition of the CBS-states, which belong to the eigenvector's phases, e.g.

$$|\psi_3\rangle = \sum_u c_u |\tilde{\varphi}_u\rangle |u\rangle \quad (136)$$

If we then measure the state of the first register, we will collapse onto a random  $|\varphi_u\rangle |u\rangle$  and thus measure this specific phase. So, even if you do not know any eigenstate, you are guaranteed to observe one specific eigenvalue after running the code, you just cannot predict, which value you will observe and you can't (completely) measure the eigenstate either.

Nevertheless, the algorithm is quite useful and supremely efficient. Finding an eigenvalue to an (unknown) eigenstate requires  $\mathcal{O}(2^{2n})$  operations on a classical computer, where  $n$  is the number of vector dimensions. Here we require  $\mathcal{O}(t^2)$  operations for the QFT and  $\mathcal{O}(t^2)$  controlled  $\hat{U}$ -operations. Assuming a more or less even distribution of eigenvalues on the unit circle we should probably aim for a precision of much more than  $1/n$ , e.g. we should choose  $t$  such that  $t \gg \log(n)$ , yielding a total of  $\mathcal{O}(\log^2 n)$  controlled  $\hat{U}$ -operations. If we can implement these efficiently (and in many cases we can), then this is a massive speedup.

Let's try on a Quantum simulator and see, what the actual circuit looks like:

All notes subject to change, no guarantee to correctness, corrections welcome.

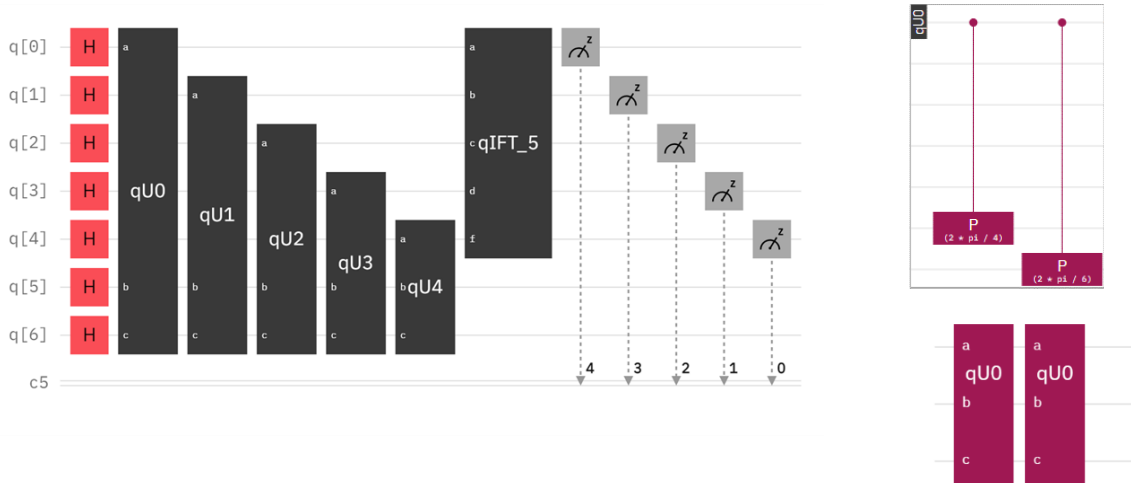

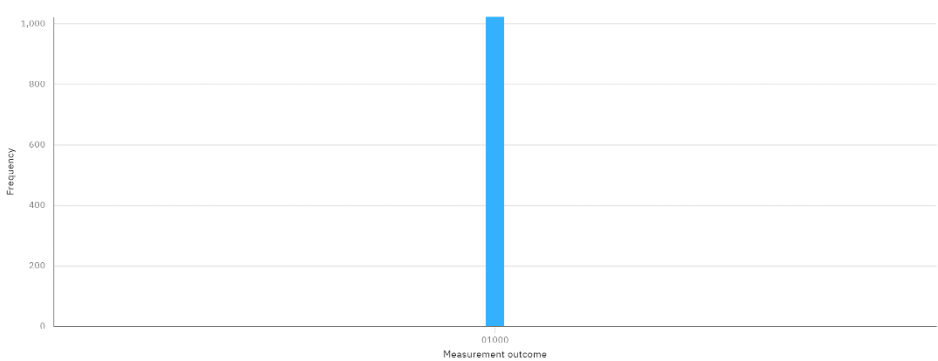

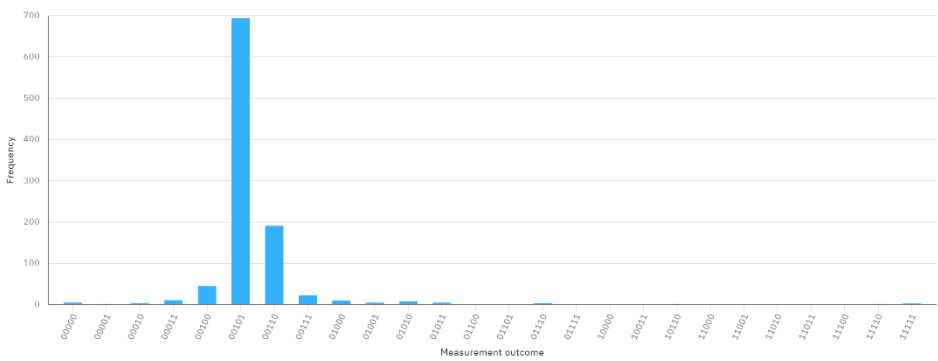


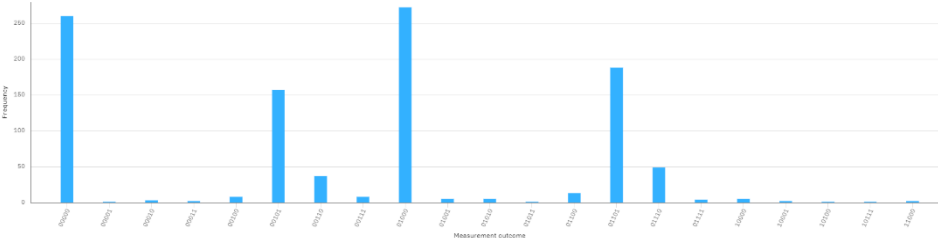


Figure 44: (left, top) Circuit representation of the Quantum Phase Estimation Algorithm for a 2 Qubit Operator  $\hat{U}$  and a 5-Qubit eigenvalue estimator. Note that the  $i$ QFT operator is the same as the QFT but with inverted order and inverted phases. (right, top) The Operator  $\hat{U}$  has a four eigenvalues  $\frac{2\pi}{4}, \frac{2\pi}{6}, 0, 2\pi(\frac{1}{4} + \frac{1}{6})$ . (right, bottom) The  $\hat{U}^2$  operator is composed of two repetitions of  $\hat{U}$ .

Note that for the given operator  $\hat{U} = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/6} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/4} \end{bmatrix}$  we have the following four eigenvector eigenvalue pairs:

Eigenvector	Eigenphase [ $2\pi$ ]	Eigenphase [ $2\pi$ ] decimal
$ 00\rangle$	0	0
$ 01\rangle$	$\frac{1}{4}$	0.25
$ 10\rangle$	$\frac{1}{6}$	0.1666
$ 11\rangle$	$\frac{1}{4} + \frac{1}{6}$	0.4166

Let's now see, if we can find the appropriate eigenvalues if we initialize to specific eigenvectors and let's also see what happens, if we initialize into a balanced superposition of all possible eigenstates:

$ u\rangle$	Result
q 4 —  q 5 — $ 01\rangle$	 <p><math>\frac{1}{4} = 0.25</math>; Correct</p>
q 4 — q 5 —  $ 10\rangle$	 <p><math>\frac{1}{8} + \frac{1}{32} = 0.15625</math>; off by 0.01 (6 bit equivalent)</p>
q 4 —  q 5 —  $ 00\rangle +  01\rangle +  10\rangle +  11\rangle$	 <p>0; Correct  <math>\frac{1}{4} = 0.25</math>; Correct  <math>\frac{1}{8} + \frac{1}{32} = 0.15625</math>; off by 0.01 (6 bit equivalent)  <math>\frac{1}{4} + \frac{1}{8} + \frac{1}{32} = 0.40625</math>; off by 0.01 (6 bit equivalent)</p>

As you can see, the algorithm does just what it is supposed to do. If the initialization is perfectly on an eigenvector and the eigenvalue is representable by a binary fraction we get exactly the correct value out. If we hit an eigenvector but the eigenstate is not representable by a binary fraction, we get within to the resolution of the binary fraction (in this case to within  $1/32=0.03$ ) and we get the correct answer most of the times. If we just guess the initial state, we still get peaks at the probability distribution at the values of the eigenvalues and we can, after a few runs, find all eigenvectors, no matter what.

### 5.3.1 A Graphic Interpretation

From above we have seen that the phase estimator makes use of the phase-kickback in conjunction with the inverse QFT algorithm. The phase-kickback acts on the initial state  $|\psi_1\rangle$  of the  $t$ -register, which is a balanced superposition with zero phase. Such as this one here:

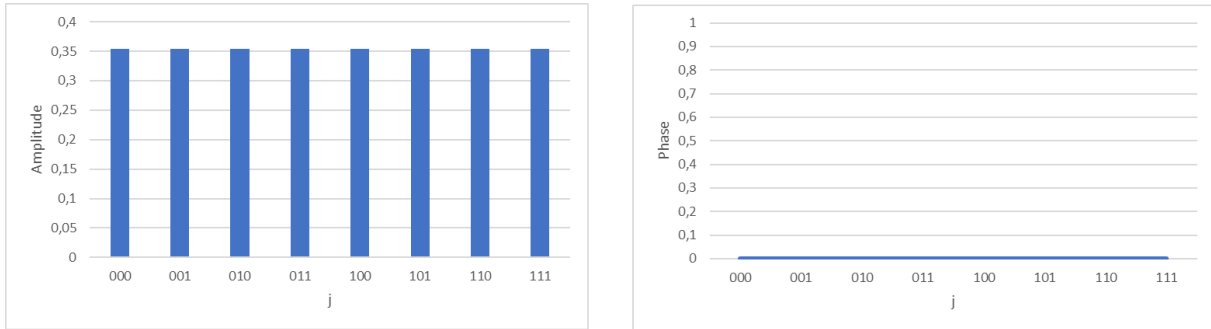


Figure 45: Initial state  $|\psi_1\rangle$  of the  $t$ -register.

The application of the  $\hat{U}^{2^j}$ -gates never changes the amplitudes of the  $t$ -register only the phases are changed. It makes sense to look at the phases starting with the most significant digit. If the most significant digit is 1 then  $\varphi^{2^{t-1}}$  is applied. In the phase graph this means that the right half is elevated by  $\varphi^{2^{t-1}}$ . In the next step half of this phase step is applied if the second-most digit is equal to one, e.g. on the right half of each of the half, such that the initial box is now a four-step staircase. The process goes on. By each step the stair is filled with twice as many smaller boxes and made smoother and smoother until a perfectly regular staircase with a step-size of  $\varphi$  is created.

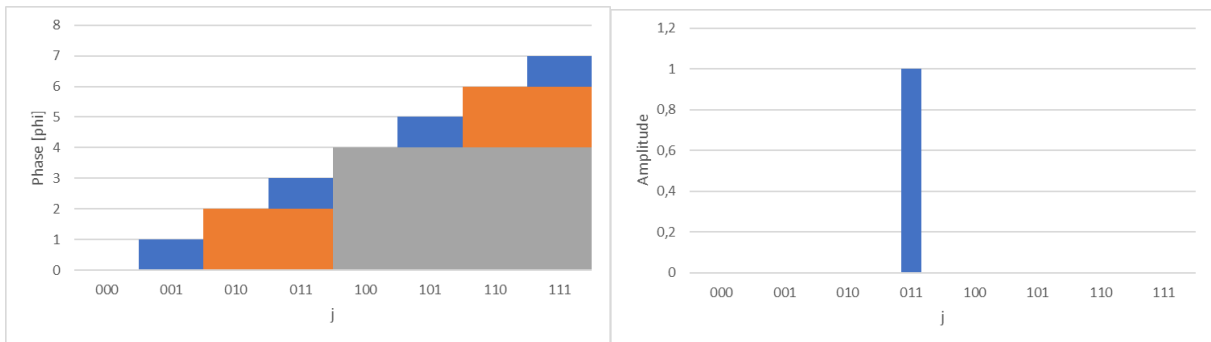


Figure 46: (left) Phase of the  $t$ -register after the application of the series of discrete power-controlled phase-shifts. Coloured boxes indicates contributions of (gray) most significant bit ( $4\varphi$ ), (orange) middle bit ( $2\varphi$ ), and (blue) least significant bit ( $\varphi$ ). (right) state of  $t$ -register after application of the  $i$ QFT. The specific location of the delta-peak is indicative of the value of  $\varphi$  and can be retrieved from a single. the CBS-measurement.

This is nothing but a plane wave with a slope of  $\varphi$ . Since  $\varphi$  itself is the number we are actually looking for, we must now just measure the slope. This is where the  $i$ QFT comes on handy. We know that the fourier transformation of a plane wave is a delta-function located at the position of the slope, hence the  $i$ QFT transforms the staircase into a delta-peak (e.g. a perfect CBS!), whose value indicates its slope and hence  $\varphi$ . We must know just measure once and the resulting CBS-code is the sought-after slope.

## 5.4 Shor's Algorithm: The Internet will Hate You

The arguably most famous algorithm for Quantum Computers is the algorithm published in 1997 by Peter Shor. The algorithm uses our profound knowledge on QFTs and Quantum Phase Estimation to create an extremely efficient solution to the problem of number factoring.

### 5.4.1 Classic Number Factoring

Assume that we have an integer number  $N$  and we would like to decompose this number into its prime factors. As an example we know that  $39 = 13 \cdot 3$ , which appears to be quite simple. However, this problem is harder than it may appear, because instead of 39 I might just give you a very large number, e.g.  $N = 7\,906\,198\,969$ . The most efficient classic solutions to this problem is to take a table of all known random numbers  $n$  starting from 2 and dividing  $N$  by all of these number until you have one



division, where the remainder (called the modulus) is 0, e.g.  $N \bmod n = 0$ . Using a Turing machine, in the worst case you'd have to try all prime number up to  $\mathcal{O}(\sqrt{N})$ . You may hope the density of prime numbers within the set of positive integers would eventually drop for large numbers but this in fact not the case, e.g. even for large integers the density of prime numbers only drops logarithmically, e.g. the number of prime up to  $\sqrt{N}$  is roughly  $\sqrt{N}/\log N$ . Assume that integer division is asymptotically as complex as integer multiplication (which is not proven to my understanding, but seems a reasonable conjecture) we know from section 1.2, that an  $n$  digit number requires  $\mathcal{O}(n \log n)$  operations to carry out a single division on a Turing machine. Thus, the grand total, given  $N = \exp(n)$  is  $\mathcal{O}\left(n^{-\frac{1}{2}} \log n \exp \frac{n}{2}\right) \sim \mathcal{O}\left(\exp \frac{n}{2}\right)$ .

To cut a long story short; if I were to give you a large number, splitting it up into primes is really hard. To the contrary, the inverse problem is really simple. Multiplying two prime numbers to get a large number is an  $\mathcal{O}(n \log n)$  operation, as you can tell my proving that  $7\,906\,198\,969 = 103\,643 \cdot 76\,283$ .

Note that most of the statements here have no hard mathematical proof. There may be more efficient approaches on a Turing machine, that would solve the prime factoring problem which we simply do not know. However, prime factoring is a mathematical problem dates back to the ancient Greek and possibly before that, and ever since the time of Euclid until the seminal paper by Shor we have not found a more efficient algorithm than the above-mentioned number sieve (actually the above one is not the number sieve but it's sufficiently close).

#### 5.4.2 Connection to Cryptography

The prime factorization problem this seems to belong to a class of problems, which are called "trap-door" functions. E.g. there are fairly easy to compute but very hard to "uncompute" (if you find this wording suggestive in the context of quantum computers, it is on purpose: Quantum Computers are reversible and thus we may expect that at least some trapdoors functions should be able to run more efficiently on Quantum Computers). Trap-door functions are not just a mere oddity, they are of paramount importance to our digitally connected world. You may be aware that pretty much all data-communication in the internet is encrypted using some type of encryption algorithm.

An encryption algorithm take a message and turns it into unintelligible garbage using a specific key. The recipient of the message can turn the garbage into the message using the same key and known algorithm. A particularly simple example is the letter shifting algorithm (caesarian cipher, named after Julius Caesar). Assume the key  $k = 5$ , which means that we shift each letter in the message by five positions (modulus 26) in the alphabet. E.g. "HELLO WORLD"  $\rightarrow$  "MJQQT BTWQI". The recipient can undo encoding using a shift of  $k = -5$  and retain "HELLO WORLD". These type if encryption algorithms are called "symmetric ciphers" because the secret key is required on both sides. Modern algorithms such as AES or RC6 are probably rather secure and are used to encrypt everything from digital money transfers, to WhatsApp messages, from power grid controls to interconnected sensors in hospitals, from warehouse databases to nuclear weapons codes.

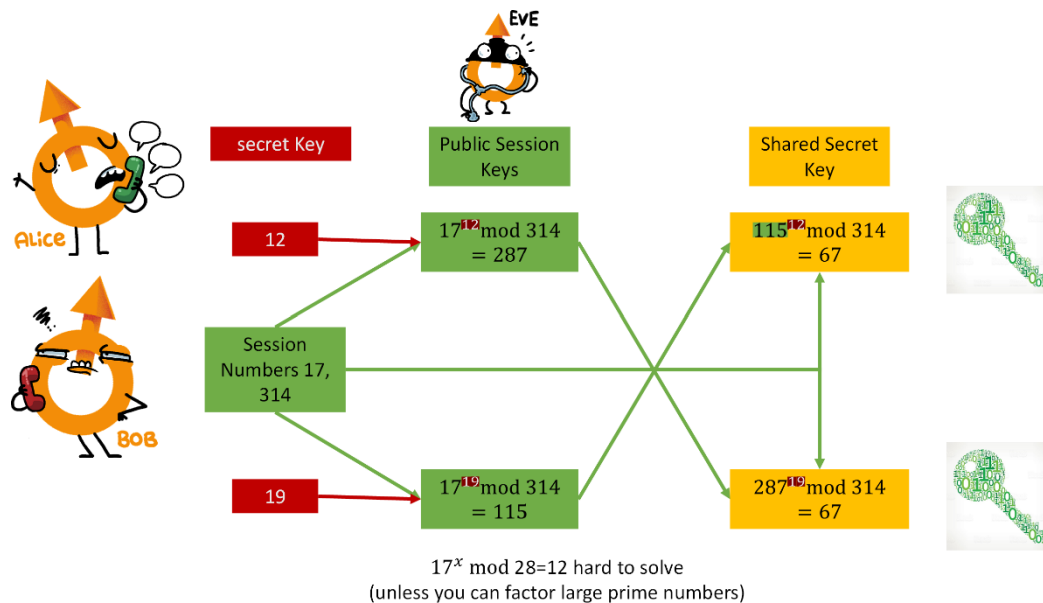


Fig. 1: The Diffie-Hellman-Scheme uses asymmetric encryption to establish a shared secret, i.e. it distributes keys, using trap-door type functions. The most common trap-door is the discrete log  $l = a^b \bmod c$ , where  $l$  is easy to calculate if  $a, b, c$  are given but  $b$  is hard to calculate if  $a, c, l$  are known. (red) Secret Data, (green) public data, (yellow) shared secret.

The real challenge and weak point of such systems is the secret key. The symmetric cipher is useless if you can't guarantee that the key is identical and secret at both sides. For some application an offline exchange of secret keys may be feasible (e.g. in TAN number systems for bank transfer) but this is generally cumbersome. You really want to be able to establish a secret between two parties over a public channel and indeed, using trapdoor functions, you can do just that. The most famous method here is the Diffie-Hellman-algorithm (Diffie-Hellman-Merkle).

I will not discuss the entire algorithm here but just sketch its outline. Assume Alice and Bob want to generate a shared secret. They start by picking an individual secret key each (called  $a$  and  $b$ ), which they will never share with anyone. Moreover, they agree publicly on a shared prime number  $p$  and a small publicly known integer  $g$ . They then generate a public key  $A, B$  each, e.g.  $A = g^a \bmod p$ , a method which is known as a discrete logarithm,

The public keys are virtually impossible to uncompute because  $g^a \bmod p$  is a trapdoor function. Its uncomputation does require a fast algorithm for number factoring (the connection here is not discussed). Therefore the public keys can be exchanged safely over an unsecured line. Alice then takes Bob's public key  $B$  and calculates  $K_a = B^a \bmod p$ . Bob takes Alice's public key and calculates  $K_b = A^b \bmod p$ . One can show that  $K_a = K_b = K$  and this Alice and Bob are guaranteed to have the shared secret  $K$ , which is only known to them because the last step of the computation requires the knowledge of the individual secret keys. This key can then be used as a key for a fast symmetric cipher.

### 5.4.3 An Alternative Approach to Prime Number Factoring

Before we can harness the power of the Quantum Computers we must reformulate the number factoring problem. Assume we have a number  $N$ , which we would like to decompose into prime factors. We then follow the following procedure:

1. Select another positive integer  $x$  with  $1 < x < N$ .
2. Check if the greatest common divisor  $\gcd(x, N)$  of  $n$  and  $x$  is larger than 1. You can do so quite efficiently e.g. with Euklid's algorithm. If  $\gcd(x, N) > 1$  then you are one hell of a lucky

student, because you have found one prime factor to  $n$  and you can terminate the algorithm successfully.

3. Find the period of the function  $x^r \bmod N$ , e.g. the smallest number  $r_0$  for which  $x^{r_0} \bmod N = 1$ . Note that  $r_0$  is also a positive integer. Also note that one can show that  $r_0 \leq N$ .
4. If  $r_0$  is uneven or if  $x^{r_0/2} \bmod N = N - 1$  then you are really unlucky. Select a different  $x$  and restart the algorithm.
5. Calculate  $n_1 = \gcd(x^{r_0/2} - 1, N)$  and  $n_2 = \gcd(x^{r_0/2} + 1, N)$ . Both are prime factors of  $N$ .

Note that this algorithm, if run on a classical computer, is not more efficient than, e.g. a number sieve. Also note, that the difficult step is the period finding step (3). It turns out that quantum computers are very good at period finding. So the key of implementing this on a QC is implementing Step 3; everything else can be run on an ordinary computer very efficiently. Before we turn to the implementation, I want to prove some of the key points here.

First let's look at the function  $x^r \bmod N$  with the example of  $x = 3$  and  $N = 35$

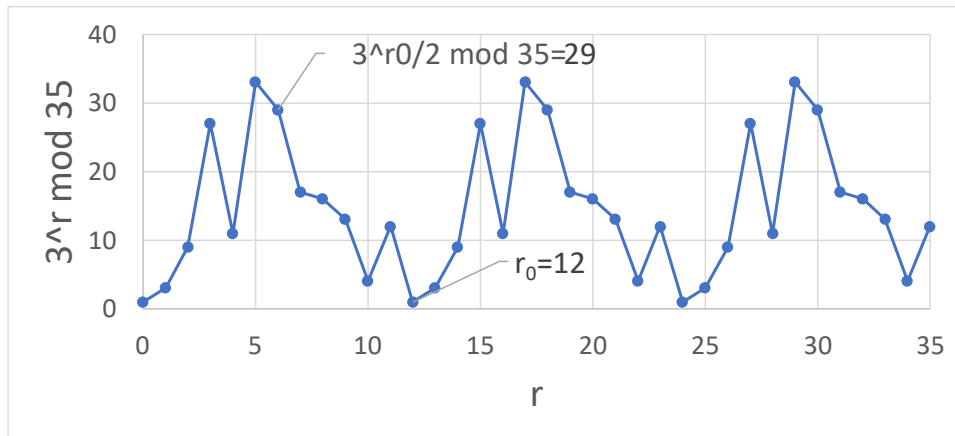


Figure 47:  $x^r \bmod n$ , with  $x = 3$  and  $n = 35$ . Clearly the period is  $r_0 = 12$ .

We find that every six calculations the return value is 1, therefore the period of the function is  $r_0 = 12$ , e.g. we see that  $x^{12} \bmod N = 1$ . We also pass the test in the fourth step of the algorithm, e.g. the period is even and  $3^6 \bmod 35 = 29$ , which is not  $35 - 1$ . We can therefore proceed by calculating  $x^{r_0/2} - 1 = 28$ . In the last step we calculate  $\gcd(28, 35)$  and find  $n_1 = 7$ , which is one prime factor of 21. The second prime factor can be determined easily by dividing  $\frac{35}{7} = 5$ . We can also find  $\gcd(30, 35) = 5$ , same thing. Let me stress one more time that the gcd algorithm is efficient; you can for example use Euclid's algorithm (check Wikipedia if you like).

The magical step to understand is obviously, why  $\gcd(x^{r_0/2} - 1, N)$  and  $\gcd(x^{r_0/2} + 1, N)$  should be a prime factor of  $N$ . What we can do, is to simply multiply:

$$\left(x^{\frac{r_0}{2}} - 1\right)\left(x^{\frac{r_0}{2}} + 1\right) = x^{r_0} - 1$$

Then we take  $\bmod N$  on both sides:

$$\left(x^{\frac{r_0}{2}} - 1\right)\left(x^{\frac{r_0}{2}} + 1\right) \bmod N = (x^{r_0} - 1) \bmod N$$

We first manipulate the right hand side, using our knowledge of  $x^{r_0} \bmod N = 1$  and  $(x^{r_0} - 1) \bmod N = x^{r_0} \bmod N - 1$ , unless  $x^{r_0} \bmod N = N - 1$ , which is not the case because  $x^{r_0} \bmod N = 1$ .

$$\left(x^{\frac{r_0}{2}} - 1\right)\left(x^{\frac{r_0}{2}} + 1\right) \bmod N = 0$$

This means, that  $\left(x^{\frac{r_0}{2}} - 1\right)\left(x^{\frac{r_0}{2}} + 1\right)$  is a multiple of  $N$  (say  $mN$ , where the specific value of  $m$  does not matter) and hence:

$$\left(x^{\frac{r_0}{2}} - 1\right)\left(x^{\frac{r_0}{2}} + 1\right) = mN = m_1 m_2 n_1 n_2$$

Where  $m = m_1 m_2$  and  $m_1$  and  $m_2$  are some unimportant integers and  $n_1$  and  $n_2$  are the unknown prime factors of  $N = n_1 n_2$ . Because the factors of the left hand side are integers, too, we can guarantee, that the two factors on the left hand each contain an  $n_1$  and an  $n_2$ , e.g.  $\left(x^{\frac{r_0}{2}} - 1\right) = m_1 n_1$  and  $\left(x^{\frac{r_0}{2}} + 1\right) = m_2 n_2$ . We can guarantee this distribution of factors because any other distribution (e.g.  $n_1$  and  $n_2$  being a part of the same factor) would imply  $x^{r_0/2} \bmod N = N - 1$ , which we have tested for in step number 4.

Therefore, we know, that both  $x^{\frac{r_0}{2}} - 1$  and also  $x^{\frac{r_0}{2}} + 1$  contain nontrivial factors of  $N$ , which we can find by simply taking  $n_1 = \gcd(x^{\frac{r_0}{2}} - 1, N)$  and  $n_2 = \gcd(x^{\frac{r_0}{2}} + 1, N)$ . Note that both  $x^{\frac{r_0}{2}} - 1$  and  $x^{\frac{r_0}{2}} + 1$  are both typically very large numbers, which may be hard to calculate but you may just use  $(x^{\frac{r_0}{2}} \pm 1) \bmod N$  to begin with, because this is calculated in the first step of Euklid's gcd-Algorithm anyway and it's a number that you have already calculated in the period-finding part of the algorithm.

Now that we understand the connection between period-finding and prime-number factoring, it is time for two remarks:

1. The algorithm in the classical sense is very inefficient. The only thing you know a-priori is that  $r_0 \leq N$  and therefore the algorithm requires up to  $\mathcal{O}(\exp n)$  operations to complete for an  $n$  digit prime number (e.g.  $N = \exp n$ ).
2. If you find the algorithm weird, then keep in mind that it is nothing more than a generalization to the divide by 10 rule that you learn in school. (e.g.  $N = 90 = 9 \cdot 10$ ). Obviously, this rule works for base  $x = 10$  because we write down numbers in this format and there is a certain compatibility with 10 and 90.

The algorithm generalizes this rule to off-by-one-pairs:  $99 = 9 \cdot 11$  and  $9999 = 99 \cdot 101$ , which can be easily factored using the binomial rule. The algorithm then looks at these numbers 99, 9 999, 999 999, 99 999 999 and sees if they are a multiples of  $N$ , which can then be easily factored. It does so in a modular sense to keep the numbers small.

#### 5.4.4 Quantum Order Finding

We shall now use the Quantum Phase Estimation algorithm to efficiently solve the Quantum Order Finding problem. The overall structure of the algorithm looks like this:

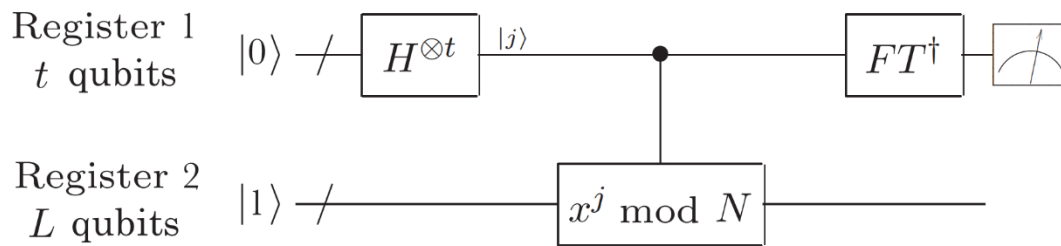


Figure 48: The structure of the quantum period finding algorithm. Stolen from Nielsen and Chuangs magnificent book.

For this we simply need to define an appropriate operator and show what kind of eigenstates /eigen-vectors this operators has. The operator is simply:

$$\hat{U}|y\rangle = |xy \text{ mod } N\rangle$$

Where  $N$  is of course the number we would like to factor and  $x$  is the random base integer as introduced above. Here  $y \in \{0,1\}^L$  are all the numbers CBS states of the eigenstate register. Note that when  $y > N$  we just use the convention that  $xy \text{ mod } N = y$ , in other words: the register only acts nontrivially, only up to a CBS with number  $N$  but unless you fuck up the initialization this will never happen. I personally find this a bit tough to grasp so we'll go by an example taking  $x = 3$  and  $N = 35$ , starting with  $y = 1$ .

$$\begin{aligned} \hat{U}^0|1\rangle &= |1\rangle \\ \hat{U}^1|1\rangle &= |3\rangle \\ \hat{U}^2|1\rangle &= |3 \cdot 3 \text{ mod } 35\rangle = |9\rangle \\ \hat{U}^3|1\rangle &= |9 \cdot 3 \text{ mod } 35\rangle = |27\rangle \\ \hat{U}^4|1\rangle &= |27 \cdot 3 \text{ mod } 35\rangle = |11\rangle \\ &\dots \\ \hat{U}^{r-1}|1\rangle &= |12\rangle \\ \hat{U}^r|1\rangle &= |1\rangle \end{aligned}$$

It should also be clear that a balanced superposition  $|u_0\rangle$  of this cycle is an eigenstate of  $\hat{U}$ , e.g.

$$\hat{U}|u_0\rangle = \hat{U} \frac{1}{\sqrt{r_0}} \sum_{k=0}^{r_0-1} |x^k \text{ mod } N\rangle = \frac{1}{\sqrt{r_0}} \sum_{k=0}^{r_0-1} |x^k \text{ mod } N\rangle$$

This can be seen in the following representation:

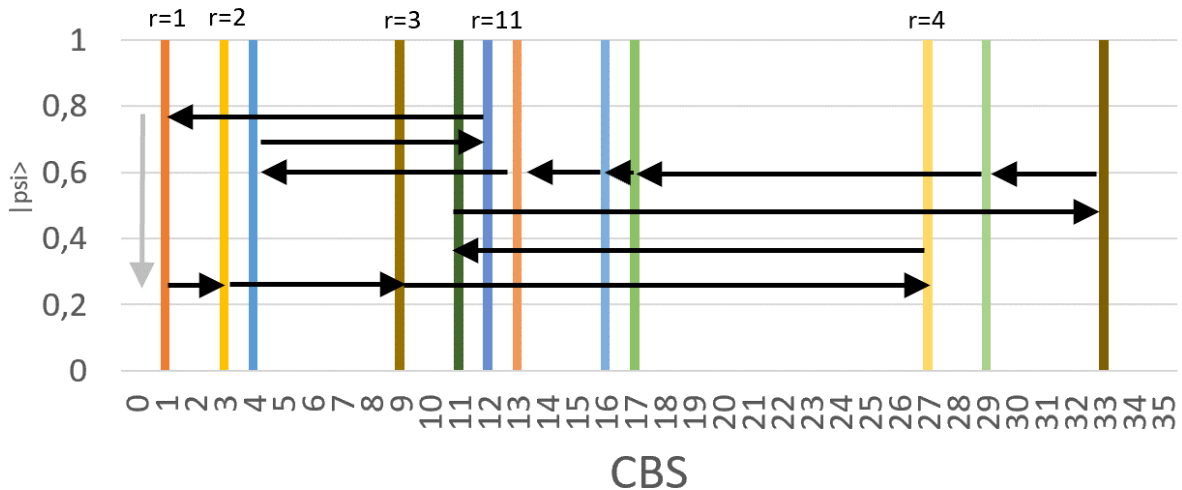


Figure 49: The representation of the  $|u_0\rangle$  eigenstate of the problem with  $x = 3$  and  $N = 35$ . The colors are just a guide to the eye. The arrows indicate the succession states, if seeded with any of the CBS which are a part of this series (e.g.  $|x^r \bmod N\rangle$ ).

This figure displays the quantum amplitudes of the eigenstate  $|u_0\rangle$  with  $x = 3$  and  $N = 35$ , with the colors as a guide to the eye to indicate the succession of CBS states, if the  $|x^r \bmod N\rangle$  was seeded with  $|0\rangle$ . It is obvious that the displayed state is an eigenstate of  $\hat{U}$ , because its application merely shifts the quantum amplitude from the  $r = 1$  (orange) to the  $r = 2$  (yellow) CBS, and the amplitude from  $r = 2$  (yellow) to the  $r = 3$  (brown) CBS and so on. The last at  $r = 11$  (blue) then fills the void left behind at  $r = 1$  (orange). The shifting of quantum amplitudes has thus occurred in a completely circular manner and nothing has changed globally.

However,  $|u_0\rangle$  is a pretty boring eigenstate because its eigenphase is  $\varphi_0 = 0$ . However, we are free to make the balanced superpositions with phases, where we just have to make sure to distribute the  $r_0$  phases on the complex unit circle evenly and over exactly  $s$  revolutions. Thus, we find that for any integer  $0 \leq s \leq r_0 - 1$  that the states

$$|u_s\rangle = \frac{1}{\sqrt{r_0}} \sum_{k=0}^{r_0-1} \exp\left[-\frac{2\pi i s k}{r_0}\right] |x^k \bmod N\rangle$$

are also eigenstates to the operator  $\hat{U}$  since

$$\begin{aligned} \hat{U}|u_s\rangle &= \frac{1}{\sqrt{r_0}} \sum_{k=0}^{r_0-1} \exp\left[-\frac{2\pi i s k}{r_0}\right] |x^{k+1} \bmod N\rangle \\ &= \frac{1}{\sqrt{r_0}} \exp\left[\frac{2\pi i s}{r_0}\right] \sum_{k=0}^{r_0-1} \exp\left[-\frac{2\pi i s (k+1)}{r_0}\right] |x^{k+1} \bmod N\rangle \\ &= \exp\left[\frac{2\pi i s}{r_0}\right] |u_s\rangle \end{aligned}$$

with the eigenphase  $\varphi_s = s/r_0$ , where  $r_0$  is the sought after periodicity. This is a quite remarkable finding. We know that the dimensionality of the part of the operator  $\hat{U}$  that is of interest is  $r_0$  and we have found  $r_0$  eigenstates. So we know, that we have now found all eigenstates. Therefore, we can guarantee that the phase estimation procedure will return an eigenvalue which is guaranteed to contain the sought-after periodicity  $r_0$ .

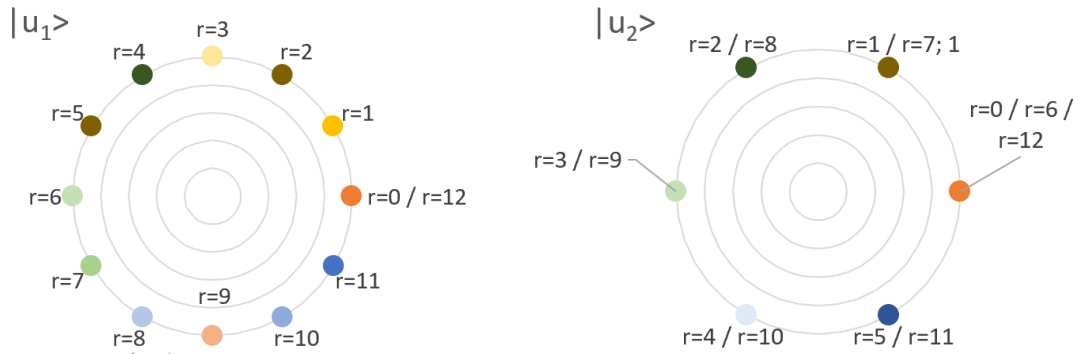


Figure 50:  $|u_1\rangle$  and  $|u_2\rangle$  eigenstates on the complex plane for  $N = 35$  and  $x = 3$ . Note that the respective eigenphases are  $\varphi_1 = 2\pi/12$  and  $\varphi_2 = 2 \cdot 2\pi/12$ , respectively. Coloring more or less identical to figure from above. Note that all eigenstates have a probability amplitude at  $\varphi = 0$  ( $r = 0$ ).

This leaves us with four problems;

- 1) we need to find good initial conditions
- 2) we need to determine a sensible number of qubits  $t$  in register one
- 3) we need to implement the  $\hat{U}^{2^j}$  operations efficiently
- 4) we need to extract  $r$  from the eigenvalues.

The first problem is the missing initial state of  $|u_s\rangle$ , (which is unknown because it depends on  $r_0$ ) can be solved by simply taking the initial state as  $|y\rangle = |1\rangle$ . This works because:

$$|1\rangle = \frac{1}{\sqrt{r_0}} \sum_{s=0}^{r_0-1} |u_s\rangle$$

Which means that irrespective of the unknown  $r_0$ , the  $|1\rangle$  state is guaranteed to be in a superposition of the eigenstates of the operator  $\hat{U}$  and thus we know that the phase estimator will collapse onto one specific  $\varphi_s = s/r_0$ .

More specifically, it follows that we have to use  $t = 2L + 1 + \lceil \log(2 + \frac{1}{2\varepsilon}) \rceil$  qubits in the first register to obtain, for each  $s$  in the range of 0 to  $r - 1$ , an estimate of the phase  $\varphi \approx s/r$  accurate to  $2L + 1$  bits with probability  $p = (1 - \varepsilon)/r$ . A quick note: we need the  $2L + 1$  bit precision to avoid ambiguities. E.g. if your prime number  $N$  is in the range of say  $1^9$  then  $L \approx 30$  and you can accept a fail rate of  $\frac{1}{4}$  then you need roughly  $t = 61 + \log(2 + 2) = 63$  qubits. So altogether you need a bit more than  $3L$  qubits for the entire circuit.

Then we need to implement a series of  $\hat{U}^{2^j}$  gates, which do the following (keep in mind, we have again introduced a binary digit representation  $j = j_1 2^{t-1} + j_2 2^{t-2} + \dots + j_{t-1} 2^1 + j_t 2^0$ ):

$$\begin{aligned} |j\rangle|y\rangle &\rightarrow |j\rangle \hat{U}^{j_t 2^{t-1}} \dots \hat{U}^{j_1 2^0} |y\rangle \\ &= |j\rangle |x^{j_t 2^{t-1}} \dots x^{j_1 2^0} y \bmod N\rangle \\ &= |j\rangle |x^j y \bmod N\rangle \end{aligned}$$

I shall not go into the details on how to implement this operation and there are indeed quite a few solutions for this problem but will just sketch the outline here. We shall first calculate  $x^j \bmod N$  by introducing a third register. This register holds  $x^2 \bmod N$  (calculated by squaring  $x \bmod N$ ). Then we calculate  $x^4 \bmod N$  by squaring this and then  $x^8 \bmod N$  etc.  $x^j \bmod N$  is then calculated by using  $x^j \bmod N$  by multiply the factors according to the bit pattern of  $j$ . E.g. if  $j = 6$  we multiply



$(x^4 \bmod N)(x^2 \bmod N)$ . The result is then multiplied onto  $|y\rangle$  (modulus  $N$ ) and then uncomputed in the third register. All this can be done quite efficiently, albeit at the cost of a few more qubits.

This leaves only the last problem. Can we obtain  $r_0$  from the phase  $s/r_0$  without knowing  $s$  (e.g. the wavefunction will collapse on one possible  $s/r_0$  and we don't know a priori, which  $s$  this belongs to)? Moreover, we only know  $s/r_0$  with a precision of  $2L + 1$  bits but what we know is that both  $s$  and  $r_0$  are integer numbers. Surprisingly, this problem can be solved uniquely and efficiently, if  $s$  and  $r_0$  have no common denominator. The algorithm can be implemented on a classical computer and is called the continuous fraction expansion. Look it up on Wikipedia.

If  $\gcd(s, r_0) > 1$  then you just had back luck and you have to try again. There is also an iterative version to solve this problem even more efficiently than just trying again by just running the phase estimate a few times and observing that with exponentially growing likelihood we can get a pair of the  $s_1$  to  $s_n$  with no common denominator.

Now is the algorithm efficient? The inverse FT requires  $\mathcal{O}(L^2)$  gates. The modal expansion requires  $\mathcal{O}(L^3)$ . The continuous fraction requires  $\mathcal{O}(L^3)$  (classical) operations, including the  $s_1$  to  $s_n$  extension. The algorithm can deal with numbers up to  $N \leq 2^L$  so the total cost is  $\mathcal{O}(\log^3 N)$ , which is an exponential speedup over all classically known solutions.

## 5.5 Grover's Algorithm: Whacking the Oracle

In the last chapter we have discussed trapdoor function, or at least one specific type of trapdoor function in detail. There we have found that we could speed up the solution of the trapdoor function prime number decomposition and discrete logarithm exponentially with a quantum computer. Do Quantum Computers always behave that way? No, they don't. In general, there is no known algorithm to exponentially speedup trapdoor function and in detail there is no known algorithm to exponentially speed up any NP-complete problem, which would amount to the same thing.

### 5.5.1 Overview

In fact, for many computationally hard to solve problems, the best solution is systematically test possible solution candidates until a proper solution is found. This approach is said to use an "oracle" function: a function  $f(x)$ , which return  $f(x) = 0$  for all  $x$ , which are not solutions to the problem ("incorrect guess/solution") and  $f(x) = 1$  for proper solutions to the problem ("correct guess/solution"). In general one may assume that there are only very few correct solutions (say  $M$  of them) among a very large number of correct solutions (say  $N$  of them), i.e.  $M \ll N$ .

The name "oracle" is altogether fitting because it contains two central ideas to this solution approach

1. the answer is always correct but in the most cases of very minimal use, and
2. asking the oracle is connected to some cost (the difference of the 21<sup>st</sup> century and the 2<sup>nd</sup> century BC is that this cost is due in computational resources and not in sacrificial goats)

Using our understanding of reversible computation (i.e. the ideas discussed in chapter 4.3) we can, of course, create a quantum version of the oracle  $\hat{O}$ , i.e.:

$$\hat{O}|x\rangle|q\rangle = |x\rangle|q \oplus f(x)\rangle$$

Where  $|x\rangle$  is the register of input Qubits and  $|q\rangle$  is the single output qubit. It is helpful to initialize the output Qubit into a superposition state  $|q\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$  because this implies that:

$$\hat{O}|x\rangle \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = (-1)^{f(x)}|x\rangle \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle)$$

All notes subject to change, no guarantee to correctness, corrections welcome.



Which is a rather funny result, because it tells us that the result qubit does not change its state irrespective of the input being a solution or not but that the quantum oracle rather marks solutions to search problems by shifting their phase by  $\pi$ . Since the oracle Qubit never changes its state, we shall omit it from the rest of the discussion  $|q\rangle$  and only concern ourselves with the oracle qubit  $|x\rangle$ .

Let's further assume that we have  $N$  possible input states and  $M \ll N$  correct solutions to the problems, e.g.  $M$  distinct CBS  $|x\rangle$ , which are correct answers to the problems. The size  $N$  of the search state is, of course, connected to the number of input Qubits by  $N = 2^n$ . Classically you don't have much choice but to test  $\mathcal{O}(N/M)$  solutions to get a "correct" answer from the oracle but it turns out that the quantum version of the oracle can facilitate the same with  $\mathcal{O}(\sqrt{N/M})$  queries to the quantum oracle. While this is not an exponential speedup, which would be required to be able to make the claim of being able to solve NP-complete problems efficiently, this is still a ridiculous speedup considering the generality of the oracle and the lack of prescribed internal mathematical structures thereof.

The algorithm was first described by Lov Grover in 1996 and consists of an iterative application of the iteration operator  $\hat{G}$  onto the input register  $|x\rangle$ . The operator itself can be decomposed into four steps:

1. Apply the oracle operator  $\hat{O}$
2. Apply the Hadamard-Operator  $\hat{H}^{\otimes n}$  onto all Qubits
3. Perform the conditional phase shift of  $\pi$ , onto every computational basis state except the state  $|0\rangle$ , i.e.  $|x\rangle \rightarrow -(-1)^{\delta_0} |x\rangle$ .
4. Apply the Hadamard-Operator  $\hat{H}^{\otimes n}$  onto all Qubits, again

Note that steps 2 to 4 are very similar to the Josza-Deutsch algorithm and indeed it turns out that Josza-Deutsch can both be described as a special case of the QFT-based class of quantum algorithms, as well as, a special case of the quantum search based algorithms.

### 5.5.2 Analysis of the Grover Iteration Step

In a next step we rewrite step 3 into an operator form  $\hat{P}$ , i.e.

$$\hat{P}|x\rangle = 2|0\rangle\langle 0| - \hat{I}$$

Where  $\hat{I}$  is the identify operator. With this understanding we can concatenate steps 2,3,4, into a single operator, the so-called inversion about the mean operator  $\hat{M}$ :

$$\hat{M} = \hat{H}^{\otimes n}(2|0\rangle\langle 0| - \hat{I})\hat{H}^{\otimes n} = 2|\psi\rangle\langle\psi| - \hat{I}$$

Where  $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle$ , is the equally weighted superposition of all CBS  $|x\rangle$ . The entire grover operator can then be written as:

$$\hat{G} = (2|\psi\rangle\langle\psi| - \hat{I})\hat{O}$$

But what does the operator actually do? To answer that question, it is useful to imagine the entire search space as a high-dimensional vector space. This space is spanned by two linearly independent subspaces, the  $M$ -dimensional subspace of "correct" answers  $|x'\rangle$  and the  $N - M$  dimensional subspace or incorrect answers  $|x''\rangle$ . Within both of these subspaces we define normalized superposition states:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x''} |x''\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x'} |x'\rangle$$

The initial state of the input register  $|\psi\rangle$ , produced by an equal superposition of all CBS  $|\psi\rangle = \hat{H}^{\otimes n}|0\rangle$  is then simply:

$$|\psi\rangle = \sqrt{\frac{N-M}{N}} |\alpha\rangle + \sqrt{\frac{M}{N}} |\beta\rangle$$

$$= \cos \frac{\theta_0}{2} |\alpha\rangle + \sin \frac{\theta_0}{2} |\beta\rangle$$

Where we have introduced the angle  $\theta_0$  such that  $\cos \theta_0/2 = \sqrt{(N-M)/N}$ . For a balanced superposition of all possible states the angle is fixed but we shall see that it is exactly this angles which changes during the grover iteration and

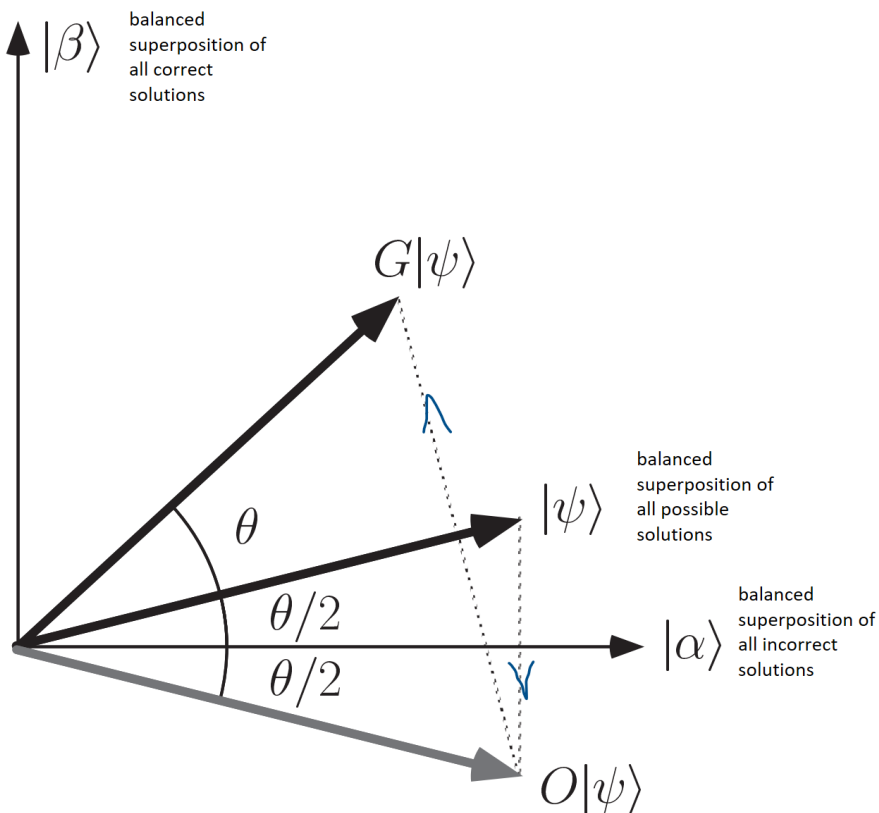


Figure 51: Geometric interpretation of the Grover iteration, as a rotation in the plane spanned by the balanced superposition of all correct solutions  $|\beta\rangle$  and the balanced superposition of all incorrect solutions  $|\alpha\rangle$ . The two sub-steps amount to a reflection at  $|\alpha\rangle$  and then reflection at the balanced superposition of all possible solutions  $|\psi\rangle$  yielding a counterclockwise rotation of the state by  $\theta_0$  (marked as  $\theta$  in the sketch). Note that  $|\psi\rangle$  is very close to  $|\alpha\rangle$  for the common case, when  $M \ll N$ .

To understand, what the Grover algorithm actually does, we need to understand, what the Oracle  $\hat{O}$  and the inversion about mean operator  $\hat{M}$  actually do to the state from above. We start with the oracle:  $\hat{O}$  performs a reflection about the incorrect average state vector  $|\alpha\rangle$  in the plane defined by  $|\alpha\rangle$  and  $|\beta\rangle$ , i.e.

$$\hat{O}|\psi(\theta)\rangle = \cos\frac{\theta}{2}|\alpha\rangle - \sin\frac{\theta}{2}|\beta\rangle = |\psi(-\theta)\rangle$$

Then we apply the inversion about the mean operator  $\hat{M}$  and notice that it, too, performs a reflection, also in the plane defined plane defined by  $|\alpha\rangle$  and  $|\beta\rangle$  but this time about the average vector  $|\psi\rangle$  (which is also in the  $|\alpha\rangle$ - $|\beta\rangle$ -plane). The average vector  $|\psi\rangle$  is tilted with respect to the superposition of all false answers  $|\alpha\rangle$  by  $\theta_0/2$ .

Because both reflections operate in the same plane, we know that for all  $k$  the state of the system will remain in the  $|\alpha\rangle$ - $|\beta\rangle$ -plane, so we can visualize the entire operation in this plane. Moreover, we know that a double reflection is a rotation and indeed we can express the action of the entire Grover operator  $\hat{G} = \hat{M}\hat{O}$  as a single angle rotation.

$$\hat{G}|\psi(\theta)\rangle = \cos\frac{2\theta_0 + \theta}{2}|\alpha\rangle + \sin\frac{2\theta_0 + \theta}{2}|\beta\rangle = |\psi(2\theta_0 + \theta)\rangle$$

Because we start the first iteration at  $\theta_0$  we find the state of the system after the  $k^{\text{th}}$  application of  $\hat{G}$  in the state:

$$\hat{G}^k|\psi\rangle = |\psi((2k + 1)\theta_0)\rangle$$

The geometric interpretation is indeed quite simple. The application of  $\hat{G}$  has rotates the state  $|\psi(\theta)\rangle$  by  $2\theta_0$  counterclockwise from the  $|\alpha\rangle$  towards the  $|\beta\rangle$  direction and has thus increased the relative quantum amplitude and therefore the likelihood of observing a correct answer.

### Action of the Grover Iteration Step on the Quantum Amplitudes

Apart from the canonical interpretation as a rotation in the  $|\alpha\rangle$ - $|\beta\rangle$  plane we can also interpret the action of the Grover iteration directly on the quantum amplitudes. We will do so assuming  $M = 1$  and only for the first Grover Iteration. Note that I have stolen the picture from the QISKIT book. We start in the balanced superposition  $|\psi\rangle$  and we will mark the amplitudes of the CBS states as the height of gray boxes, like this:

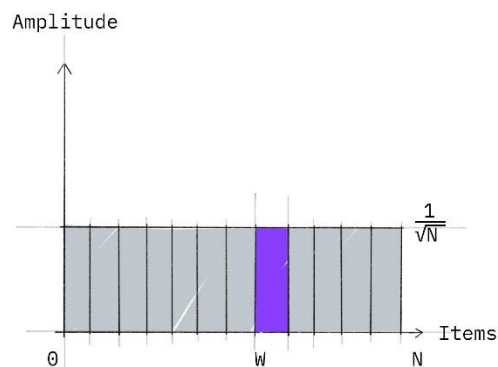


Figure 52: Initial distribution of the amplitudes at the beginning of the first Grover step. The correct solution is marked in purple.

We have marked the correct solution in purple as a guide to the eye. We then apply the oracle operator  $\hat{O}$ , which marks the correct solution by negating its phase.

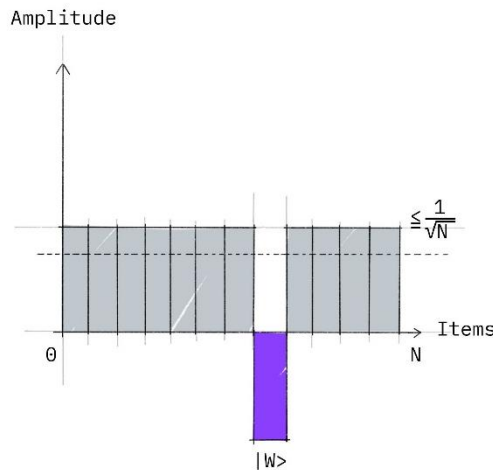


Figure 53: Distribution of the amplitudes after application of the oracle function  $\hat{O}$ . The dashed line at the top marks the average of the amplitudes.

As you can see the oracle has two effects. The first is that it flips the state of the correct solution and thereby it reduces the average of the amplitudes to below  $1/\sqrt{N}$ , which is marked by the dashed line in the image. The next step is the application of the inversion about the mean operator  $\hat{M}$ , which mirrors the amplitudes at the aforementioned dashed line. Thereby the gray boxes shrink, whereas the purple box flips towards the positive and grows in size, yielding:

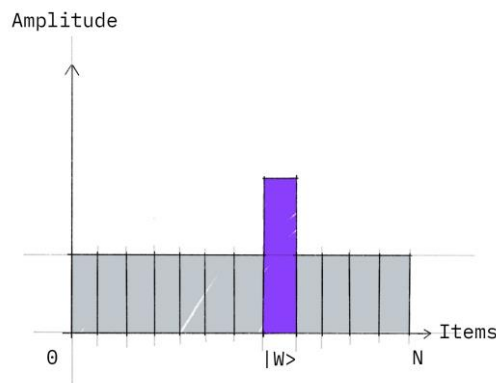


Figure 54: Distribution of the amplitudes after application of the inversion about the mean operator  $\hat{M}$ . The observation of the correct answer is now more likely than before.

As is obvious, the likelihood of observing a correct answer has increased. By repeated application we can drop the size of the gray boxes almost to zero and concentrate all the quantum amplitude in the purple box.

### 5.5.3 Termination Conditions and Optimality

It is now a mere question of when to terminate, such that we do not “overrotate” and again decrease the likelihood of the system being in any of the CBS, which comprise  $|\beta\rangle$ . Quantitatively we must terminate the algorithm, when  $\frac{2k+1}{2}\theta_0 = \frac{\pi}{2}$  this means that:

$$k = \frac{\pi}{2\theta_0} - \frac{1}{2}$$

Let's assume for simplicity that  $M \ll N$ , e.g. that a correct answer is a fairly rare event. Note that this approximation is an exclusively technical approximation. As long as  $M < N/2$  this just impacts the termination condition and if  $M > N/2$  then we can just add one (or more) idle input Qubit to the Oracle to artificially increase the  $N$  by factors of 2 to retain a situation where  $M < N/2$ . If  $M \ll N$  then we can approximate  $\theta \approx 2\sqrt{M/N}$  and thus we get:

$$k = \frac{\pi}{4} \left(\frac{N}{M}\right)^{1/2} - \frac{1}{2} \approx \frac{\pi}{4} \left(\frac{N}{M}\right)^{1/2}$$

Here we see the algorithm terminates with  $\mathcal{O}(\sqrt{N})$  operations, which is, of course the scaling behaviour that we desire.

Because we can only choose an integer number of  $k$  we however also see that, unless we get really lucky, we will never get  $|\beta\rangle$  exactly and angle of  $\theta = \frac{\pi}{2}$  will never be obtained exactly. We can, however, estimate the maximal error of failure, because the maximal angle deviation is at most  $\Delta\theta \leq 2\sqrt{M/N}$  giving a failure probability of not more than  $\varepsilon = M/N \ll 1$ . In practice this is not a problem, because we can run the oracle once more for any attained solution and check if it really is a solution and run the entire algorithm again if we have failed. Since the probability of failure is low (per construction it is always  $< 1/2$ ) the probability of repeated failure drops exponentially.

This still does not solve the problem that it seems like we must know  $M$  to be able to run the algorithm. Without giving a proper mathematical proof I am just gonna state here, that we can just run the algorithm in a series assume that  $M = 1, 2, 4, 8, 16, \dots$  e.g. we run it for  $k = \frac{\pi}{4}\sqrt{N}, k = \frac{\pi}{4}\sqrt{N/2}, k = \frac{\pi}{4}\sqrt{N/4}, \dots$  and then we test the retained result for all iterations. One can show that the overall failure probability remains at  $\varepsilon = M/N \ll 1$  and the overall number of Grover iterations  $K$  is then  $K = \frac{\pi}{4}\sqrt{N}$  which is just the same as if  $M = 1$ .

#### 5.5.4 A Physical Model

Let's attempt to physically understand, where the  $\sqrt{N}$  dependence comes from. Assume we have an array of  $N$  waveguides, which are all identical except for one, which is different. Your job is to find out, which one is different, by propagating light down the waveguide array. However, you can only launch light into a single fixed waveguide. The waveguides are arranged in a line and each waveguide is coupled evanescently to both neighbours; i.e. if you launch light into one waveguide it will couple to the neighbouring waveguides. You have a second "perfect" waveguide array, which you can use as an (interferometric) source of reference.

The classical approach is to take incoherent light (e.g. white light). Incoherent light behaves classically, i.e. it does not exhibit interference. Its spreading through the waveguide array follows the laws of classical statistics (light does a classical random walk) and after length of  $L$  your light has spread over  $N \approx \mathcal{O}(\sqrt{L/L_0})$  individual waveguides (note:  $L_0$  is some characteristic length that corresponds to the interwaveguide coupling). The type of diffraction is called diffusive.

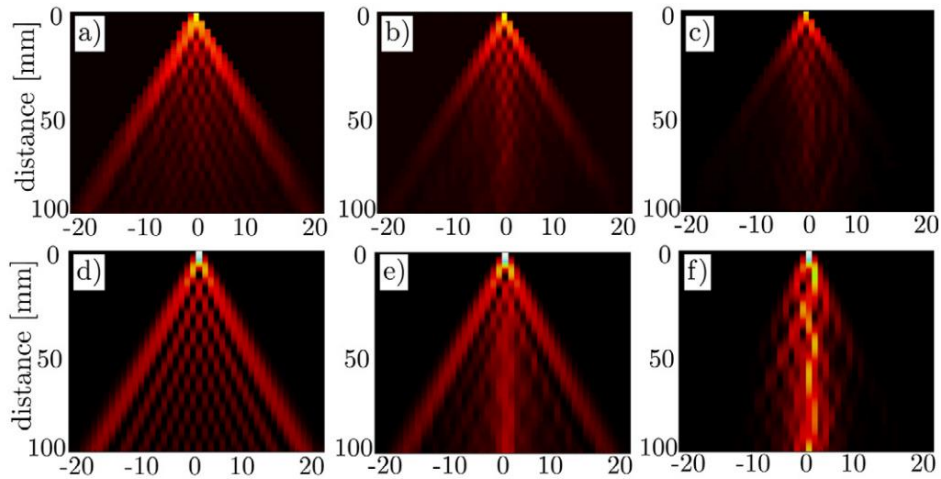


Figure 55: Numerical and experimental observations of (left) ballistic and (right) diffusive diffraction patterns in a waveguide array. Note that in this case the transition is enforced by a variation of waveguide perturbations and not by incoherent illumination. The effects is the same. Stolen from U. Naether et al 2013 *New J. Phys.* 15 013045.

The quantum approach is to take coherent light. Coherent light exhibits interference. Its spreading through the waveguide array follows the laws of quantum statistics (light does a so-called quantum random walk) and after length  $L$  you light has spread over  $N \approx \mathcal{O}(L)$  individual waveguides. This type of spreading is called ballistic. You can therefore find the solution waveguide with a square-root enhanced efficiency.

Also note that there is quite an interesting work by Anderson from already 1958, which shows that even very small (random) perturbations to such a kind of array eventually stops the ballistic spreading of the quantum wave altogether (the effect is called Anderson Localization). This can be transferred one-to-one to quantum computers: noise in quantum computers limit your search space exponentially.

### 5.5.5 Example: Solving $a+b=17$ in QISKIT

Will be discussed in the lecture. Material is available in moodle.

## 5.6 Quantum Error Correction

## 6 Quantum Galore

This lecture only serves to introduce the basics of quantum computers, shine some light on the implementation of physical gates, discuss a few key quantum algorithms and give some super-simple circuit examples. And it really is just that: an appetizer. All of the mentioned fields of science have undergone dramatic and self-accelerating improvements in the last years and new devices and methods are invested as we speak.

Some developers have 1000+ Qubits on single chips on their roadmaps until 2025; particularly superconducting Qubit system seem to lead the way in scaling of the sheer number of Qubits. Ion-Trap computers on the other hand are a close contender and they seem to move more towards better Gate qualities, complementing Quantum Computers. Photonic QCs have moved heavily towards Boson-sampling, which is a subset of Quantum Computing, but can draw from the amazing quality with which we can scale photonic circuits and the precision with which we can manipulate them.

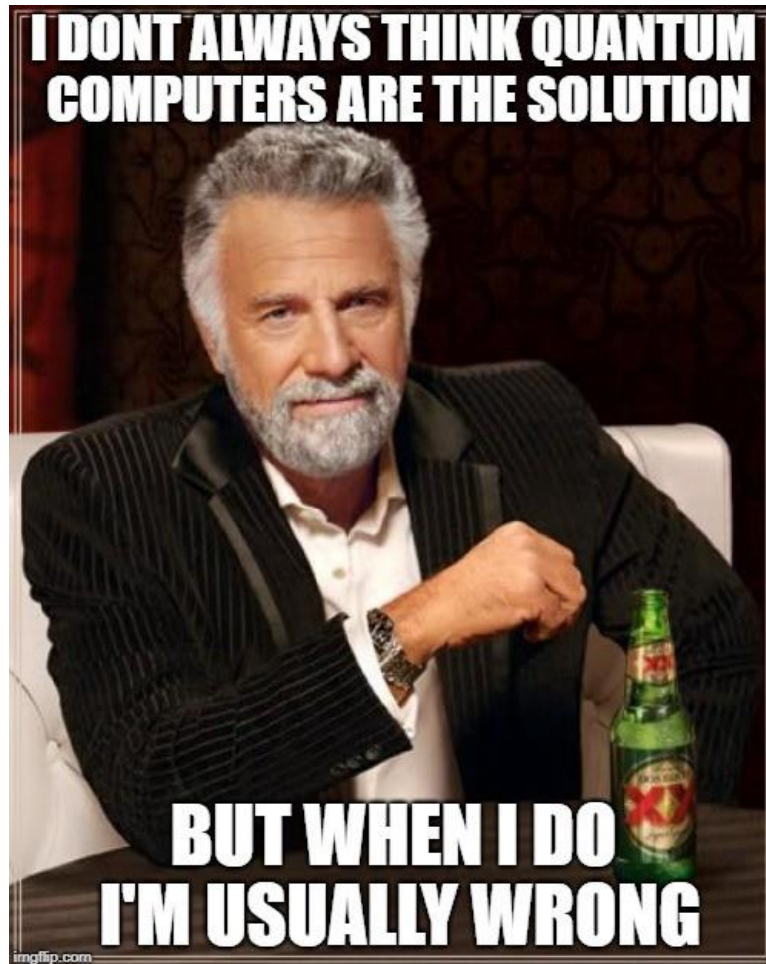


Figure 56: Don't ask me about Quantum Computers. Ask the world's most interesting man.

The development of algorithms and their applications are also going through the roof and am not even trying to given an overview here. What is probably most noteworthy, is that quantum computers are now so easily accessible that the q-software community starts to become disconnected from the q-hardware community. Historically this was one of the points of ignition for classical computers: the ability to develop software without the requirement to understand the hardware in all details.

There are also many more and exciting algorithms, which we have not discussed here. Some should find their mention here. The QFT can be extended to the HHL algorithm, with which linear systems of equations can be solved; the amount of applications is gargantuan. A field that we have not touched base on is the field of Quantum Simulation; this field is particularly interesting: the simulation of the structure of atoms and molecules is one where we don't have good classical algorithms at all, were an approximate solution is often acceptable and were the connectivity of contemporary quantum chips is often compatible with the problem: both field simulation but also Variational Quantum Eigensolvers (VQE) are important classes of algorithms here. A related problem is the search for minima of function and the solution of optimization problems, which is done with Quantum Approximate Optimization Algorithms (QAOA). All of these above can be arbitrarily confined with hype topics: QAOA+Neural Networks = Quantum Machine Learning. Or how about Quantum Image processing?

The future has just begun...



## 7 Alternative Computational Models

### 7.1 Measurement-based Quantum Computing

We have now established that a universal quantum computer can be constructed using a certain minimal set of single-qubit and two-qubit gates, and have seen that – depending on the choice of these gates - we may require a large number of quantum gates to produce a desired transformation. Not all of these gate operations will be equally accessible from an experimental point of view - in particular, as we shall see in the following, multi-qubit controlled gates are technologically challenging. The practical implementation of quantum computers thus requires careful consideration of the types of resources involved in a computation, along with methods that will allow overcoming inevitable experimental imperfections in physical systems. Over the years several alternatives to the quantum circuit model have been proposed, and in the following we shall discuss one that has gained particular traction in photonics platforms: measurement-based quantum computing and the cluster state model. Other approaches, such as the adiabatic model of quantum computing<sup>7</sup> are beyond the scope of this lecture series.

In their seminal 1999 paper<sup>8</sup>, Gottesmann and Chuang proposed a variant of quantum computing in which quantum gates are applied to quantum states via quantum teleportation. They proved that single-qubit unitary gates, Bell state measurements, and entangled resource states are sufficient to construct a universal quantum computer. In essence, their approach was to substitute multi-qubit control gates with entangled resource states and multi-qubit Bell state measurements – an ingenious feat that has since become known as the “teleportation trick”. The benefit that this entails might not be immediately obvious - after all we have seen that a Bell state measurement and entanglement may be achieved using Hadamard operations and CNOT gates, the latter being exactly the type of gate we would hope to avoid. The key point to note is that the preparation and detection of a particular entangled state can be substantially easier to realize than a multi-qubit gate that works for the most general multi-qubit input state. This makes it practical for implementations where gates cannot be applied directly, such as optical quantum computing, where single-qubit operations and Bell state measurements based on quantum interference and photodetection are substantially easier to realize than general qubit-controlled operations. The teleportation trick was a breakthrough for linear optical quantum computing and the starting point for the KLM approach to universal photonic quantum computing<sup>9</sup> and more general measurement-based approaches, such as the one-way cluster-state model.

#### 7.1.1 Quantum Teleportation

To understand the Gottesmann and Chuang “teleportation trick”, let us cycle back and appreciate the quantum teleportation protocol in a little more detail. The quantum teleportation protocol is typically discussed in the context of quantum communication where a key challenge is to get quantum information from Alice (A) to Bob (B). Quantum Teleportation was initially conceived to facilitate the transfer quantum states over a noisy quantum communication channel<sup>10</sup>. Teleportation allows Alice to send an unknown quantum bit (*oblivious* protocol) to Bob with the help of an entangled resource state and classical communication. In other words, Alice can transmit a quantum state to Bob without physically

---

<sup>7</sup> [https://en.wikipedia.org/wiki/Adiabatic\\_quantum\\_computation](https://en.wikipedia.org/wiki/Adiabatic_quantum_computation)

<sup>8</sup> Gottesman, D. & Chuang, I. L. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature* 402, 390:393 (1999).

<sup>9</sup> E. Knill, R. Laflamme, and G.J. Milburn, *Nature* 409, 46 (2001).

<sup>10</sup> Bennett, Charles H., et al. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels." *Physical review letters* 70.13 (1993): 1895.



transmitting the information carrier on which the qubit is encoded. We can express quantum teleportation in the following communication resource inequality:

$$[qq] + 2[c \rightarrow c] \geq [q \rightarrow q] \quad (137)$$

where  $[qq]$  is an entangled resource state shared between Alice and Bob, and  $[c \rightarrow c]$  denotes a single use of a classical bit channel. The teleportation protocol consumes shared entanglement and two uses of classical bit channel to transmit a quantum state  $[q \rightarrow q]$ . In quantum computing we are less concerned about communication resources, and usually more worried about the required number of gate operations and circuit depth. This is contrast to a quantum communication setting, where local operations, no matter how complex are usually considered “free” resources (for more on such resource inequalities and distributed quantum protocols, you’d best sign up for the quantum communications lecture).

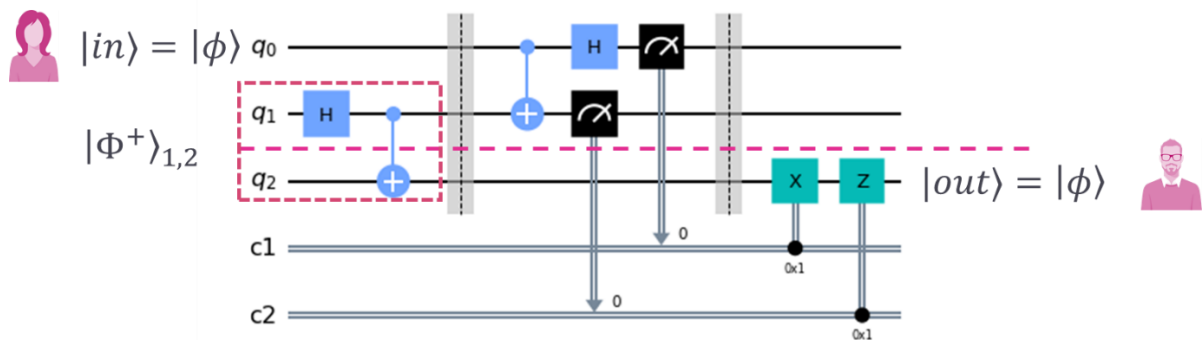


Figure 57: Quantum Teleportation in the circuit model. Alice and Bob are a remnant of the protocol’s origins in quantum communication. As we shall see in the following, the protocol is equally useful in a quantum computing setting.

The sequence of gates required to implement the teleportation protocol is illustrated in the now familiar circuit gate mode in the figure above. Qubit 0 is initialized in an unknown qubit in the state

$$|\phi\rangle_0 = \alpha|0\rangle + \beta|1\rangle \quad (138)$$

and Alice and Bob share an entangled state on qubits 1 and 2,

$$|\Phi^+\rangle_{1,2} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (139)$$

This state is can be obtained by applying the Hadamard and CNOT operations to qubits 1 and 2 (pink box). At the first barrier, the joint system of qubits 0,1,2 is thus described by the quantum state:

$$|\phi\rangle_0 |\Phi^+\rangle_{1,2} = \alpha|0\rangle + \beta|1\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \beta|100\rangle + \alpha|011\rangle + \beta|111\rangle) \quad (140)$$

Next, using the fact that the Bell-state basis is a complete basis,

$$|00\rangle = \frac{1}{\sqrt{2}}(|\Phi^+\rangle + |\Phi^-\rangle)|01\rangle = \frac{1}{\sqrt{2}}(|\Psi^+\rangle + |\Psi^-\rangle) \quad (141)$$

$$|10\rangle = \frac{1}{\sqrt{2}}(|\Psi^+\rangle - |\Psi^-\rangle)|11\rangle = \frac{1}{\sqrt{2}}(|\Phi^+\rangle + |\Phi^-\rangle) \quad (142)$$

we can rewrite this in terms of the Bell state Basis on qubits 0 and 1, as:

$$\begin{aligned}
 |\phi\rangle_0 |\Phi^+\rangle_{1,2} &= \#(143) \\
 &= \frac{1}{2} [\alpha(|\Phi_{0,1}^+\rangle + |\Phi_{0,1}^-\rangle)|0\rangle_2 + \beta(|\Psi_{0,1}^+\rangle - |\Psi_{0,1}^-\rangle)|0\rangle_2 + \alpha(|\Psi_{0,1}^+\rangle - |\Psi_{0,1}^-\rangle)|1\rangle_2 \\
 &\quad + \beta(|\Phi_{0,1}^+\rangle - |\Phi_{0,1}^-\rangle)|1\rangle_2] \#(144) \\
 &= \frac{1}{2} [|\Phi_{0,1}^+\rangle(\alpha|0\rangle_2 + \beta|1\rangle_2) + |\Phi_{0,1}^-\rangle(\alpha|0\rangle_2 - \beta|1\rangle_2) + |\Psi_{0,1}^+\rangle(\alpha|1\rangle_2 + \beta|0\rangle_2) \\
 &\quad + |\Psi_{0,1}^-\rangle(\alpha|1\rangle_2 - \beta|0\rangle_2)] \\
 |\phi\rangle_0 |\Phi^+\rangle_{1,2} &= \frac{1}{2} [|\Phi_{0,1}^+\rangle|\phi\rangle_2 + |\Phi_{0,1}^-\rangle Z|\phi\rangle_2 + |\Psi_{0,1}^+\rangle X|\phi\rangle_2 + |\Psi_{0,1}^-\rangle XZ|\phi\rangle_2]
 \end{aligned}$$

We see that, depending on the Bell State of qubits 0 and 1 the state, which was initially on qubit 0, now re-appears on qubit 2, up to some corrective Z and X operations. To execute the teleportation protocol, Alice performs a Bell State measurement on the qubits 0,1. She then sends two classical bits which indicate the outcome of her Bell state measurement (00:  $\Phi^+$ , 01:  $\Phi^-$ , 10:  $\Psi^+$ , 11:  $\Psi^-$ ) to Bob. Bob then applies the corresponding a corrective operation to his qubit. To recover the initial state on qubit  $|\phi\rangle_2$

Bob's State	Bits Received	Gate Applied
$(\alpha 0\rangle + \beta 1\rangle)$	00	$I$
$(\alpha 1\rangle + \beta 0\rangle)$	01	$X$
$(\alpha 0\rangle - \beta 1\rangle)$	10	$Z$
$(\alpha 1\rangle - \beta 0\rangle)$	11	$ZX$

Notice that if Bob does not apply the corrective Pauli operation to qubit 2, then the teleported state is identical up to the corresponding Pauli gate, i.e.  $X^i Z^j |\phi\rangle_2$ . The “feedforward” of the classical bits to apply the corrective Pauli gates is not strictly necessary - as long as the impact of the additional Pauli gates that are consequently incurred by the state can be tracked throughout all subsequent processing steps they can be undone at the end of the computation.

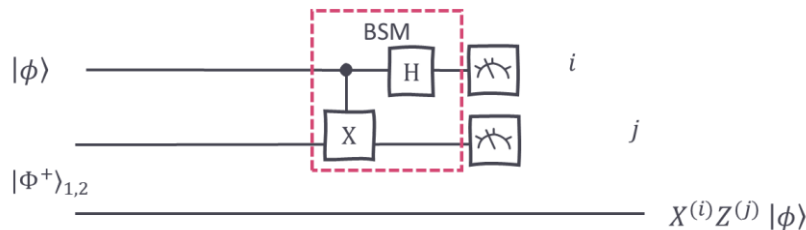


Figure 58: Quantum Teleportation without the corrective Pauli operations results in a modified output state.

### 7.1.2 The teleportation trick; or: teleporting a qubit “through a gate”

The teleportation protocol consumes a particular entangled state and transfers the state of qubit 0 to qubit 2, up to a corrective Pauli Gate. But what if we are provided with a different entangled state? In the above example we used  $|\Phi^+\rangle$ , but we might equally have used a maximally entangled  $|\Phi^-\rangle$  state to run the protocol. In this case the output state would be mapped as follows:

$$|\phi\rangle_0 |\Psi^+\rangle_{1,2} = |\phi\rangle_0 X|\Phi^+\rangle_{1,2} = \frac{1}{2} [|\Phi_{0,1}^+\rangle X|\phi\rangle_2 + |\Phi_{0,1}^-\rangle XZ|\phi\rangle_2 + |\Psi_{0,1}^+\rangle|\phi\rangle_2 + |\Psi_{0,1}^-\rangle Z|\phi\rangle_2] \quad (145)$$

In other words, if the entangled resource state is changed, then so does the mapping of Bell-state measurement (BSM) outcomes to gate operations. Gottesmann and Chuang key result was to notice that this could be used to perform targeted manipulation of the teleported state. To see how this

works, we check what happens to the teleported state when we manipulate  $U|\phi\rangle$  - with our hope being that this will give us a result that  $U|\phi\rangle:U|\phi\rangle$ :as illustrated below:

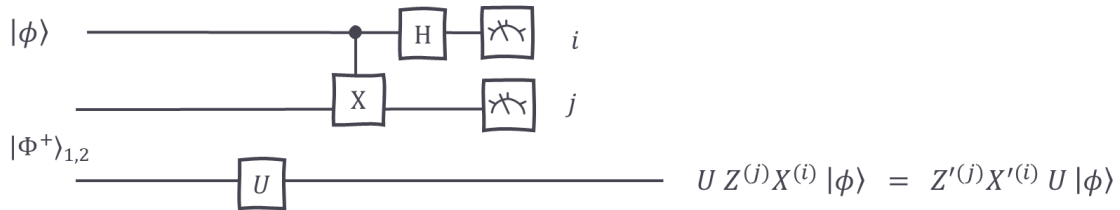


Figure 59: Quantum Gate Teleportation. Acting on the entangled resource state results in transformed output state. The unitary should act directly on the state, which can be accomplished by commuting the unitary through the Pauli gates.

Since the circuit has no gate connecting qubits 0 and 1 with qubit 2, we can simply apply the unitary operator to the teleported state<sup>11</sup>, i.e.  $U Z^{(j)} X^{(i)} |\phi\rangle$ . This looks quite good already, but we still have the additional Pauli gates between the unitary and the state to be acted on. Fortunately, for a large class of unitaries the Pauli gates can be commuted to the front without adding other gates, i.e.  $U Z^{(j)} X^{(i)} |\phi\rangle = Z'^{(j)} X'^{(i)} U |\phi\rangle$ . To illustrate by means of example, consider the Hadamard operator:  $H Z^{(j)} X^{(i)} = -1^{i+j} Z^{(j)} X^{(i)} H$ . At this point the teleportation trick may seem no more than a conjuring trick; in the end, what have we really gained from this –we need to act on a quantum state with a unitary operation in either case. This is undoubtedly true, however there is a marked practical difference between a perfect gate that acts on an unknown quantum state (that is embedded in a larger computational process) and applying the same operation to a known resource state. The entangled resource state may be prepared beforehand, and independent of any quantum data to be acted on. The benefit becomes even more obvious when we consider qubit-control gates, where can simply duplicate the teleportation trickery. In the example below we have applied the CNOT operation to the resource state, and again verify that the Pauli gates can be commuted to the front, leaving us with a CNOT acting directly on the teleported qubits.

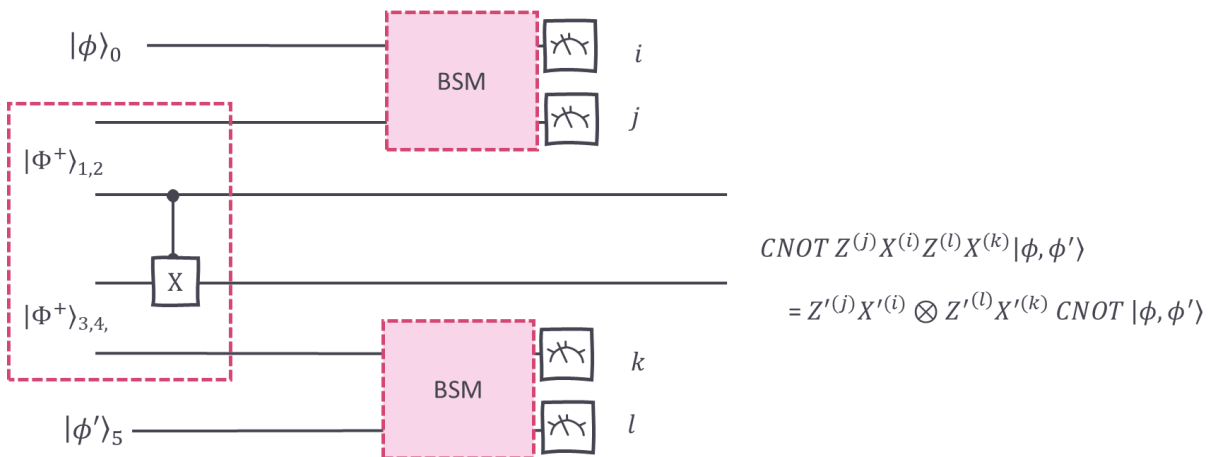


Figure 60: Quantum Gate Teleportation with a CNOT operation. The box contains the required resource state, that can be constructed offline.

<sup>11</sup> The fact that we can apply the unitary to the state "after" teleportation, even though we may have applied the unitary to qubit 2 well before the Bell state measurement, might seem counter-intuitive. It is a result of the linearity of quantum theory - since qubit 2 and qubits 1 and 2 are not connected by any multi-qubit gates the order in which an experimentalist applies the unitary operation and the Bell state measurement is irrelevant.

The required resource state can also be generated via Bell State measurements and single qubit unitaries acting on a larger entangled state (Figure 61). We can see this as follows:

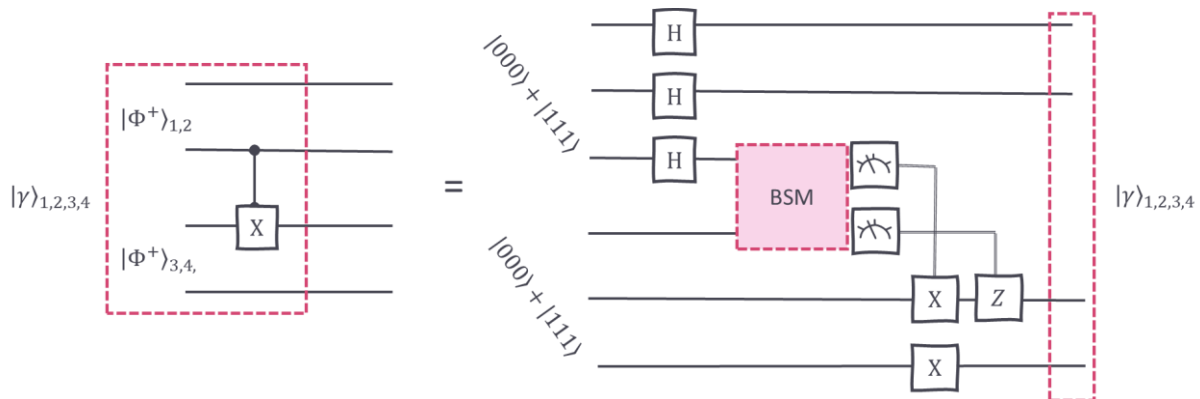


Figure 61: Construction of the resource state for CNOT from two 3-qubit entangled states.

In summary, this shows by means of example that 1- and 2-qubit gates can be applied to unknown input states by teleporting the state through the circuit. To achieve this, we need only single qubit-unitaries for correction, entangled resource states, and Bell state measurements. Strictly speaking, here we have only shown it for gates that can be commuted through Pauli gates at no additional resource (unitaries of this type are part of the so-called Clifford Group) – the reader is referred to the original article by Gottesmann et al. for a proof that the scheme also works for general unitary operators. The teleportation trick was thus the starting point for what has since become known as measurement-based or one-way quantum computing.

## 7.2 One-Way Quantum Computing

The concept of one-way quantum computing was introduced by Raussendorf and Briegel forms an alternative to the circuit/gate model. The central resource in this architecture is a large, highly entangled resource state, a so-called “Cluster State”. The computation is moved along by a sequence of single qubit measurements on the cluster state. This is a very powerful tool in part because the cluster state can be prepared beforehand or grown on the fly – it lends itself to both implementations that have limited success rate in creating large cluster states or systems with limited coherence time (i.e. shallow circuit depth). Before we introduce these cluster states we shall illustrate the basics of the measurement based quantum computing approach by mapping some simple circuits to measurements on entangled resource states.

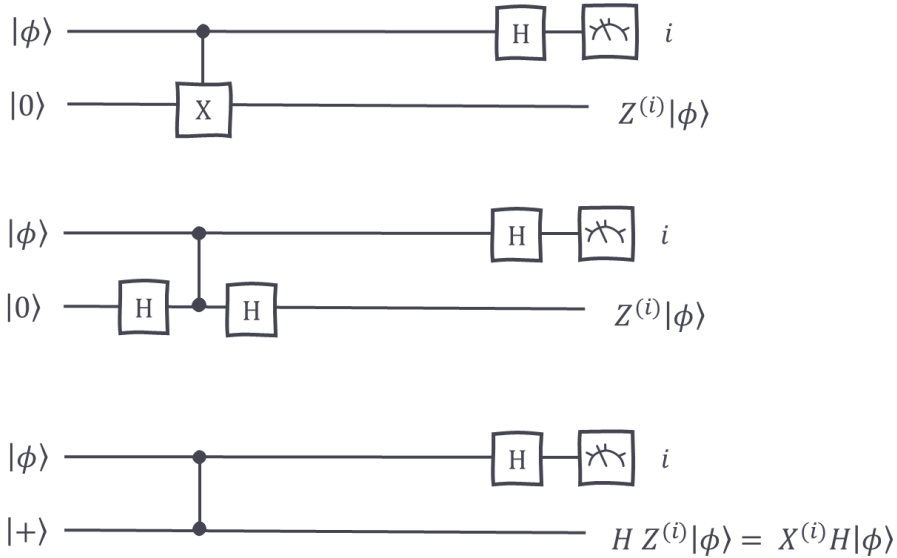
To do so, let us first consider a simplification of the teleportation circuit (Figure 62), which is also known as a local teleportation circuit. Formally this local teleportation can be described as follows:

$$CNOT(\alpha|0\rangle + \beta|1\rangle)|0\rangle = \alpha|00\rangle + \beta|11\rangle = |+\rangle(\alpha|0\rangle + \beta|1\rangle) + |-\rangle(\alpha|0\rangle - \beta|1\rangle)$$

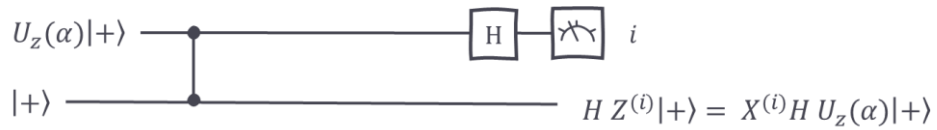


Figure 62: Simplified teleportation circuit. While very similar to “regular” teleportation, the approach is not terribly useful if we consider quantum communication setting, since it requires a control-gate that acts on the input and output qubits, i.e. they have to be at the same location.

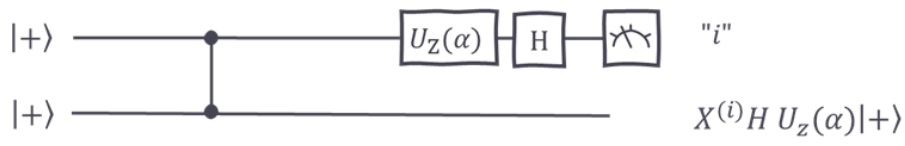
While this two-qubit circuit accomplishes the same task as the regular teleportation (it transfers the state of qubit 1 to qubit 2 up to an extra Pauli gate), there are also key differences: i) only one corrective Pauli gate is required; ii) the Bell state measurement is replaced with a single-qubit measurement; iii) the approach is not applicable in a quantum communication setting since it requires a control-gate that acts on both teleported and the teleporter, which implies the in- and output qubits be at the same location. For reasons that will become clearer in the following, we will re-wire the circuit in Figure 62 using some basic identities:



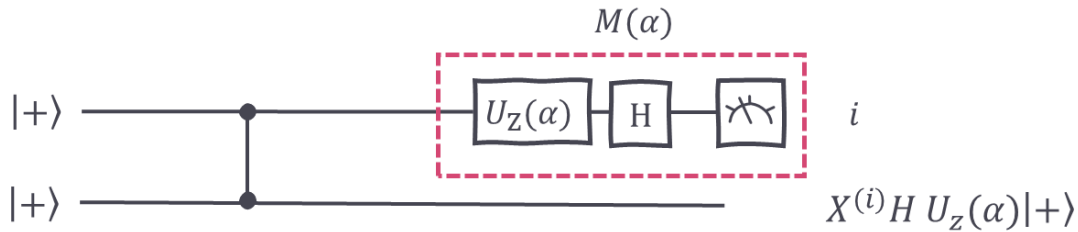
In the last step we used the fact that  $XHX=Z$ , i.e.  $XH=HZ$ . Again, and following the same line of inquiry as in the preceding section, we now check what happens when we apply a unitary to the qubit in the input state. Without loss of generality, we consider the state to be of the form  $|\phi\rangle = U_z(\alpha)|+\rangle$ , where  $U_z(\alpha) = \exp(-\frac{i\alpha}{2}Z)$  is a rotation around the z-axis of the Bloch sphere.



Since the Z-rotation commutes with the Z gate, we can commute it through the control-phase gate and get:

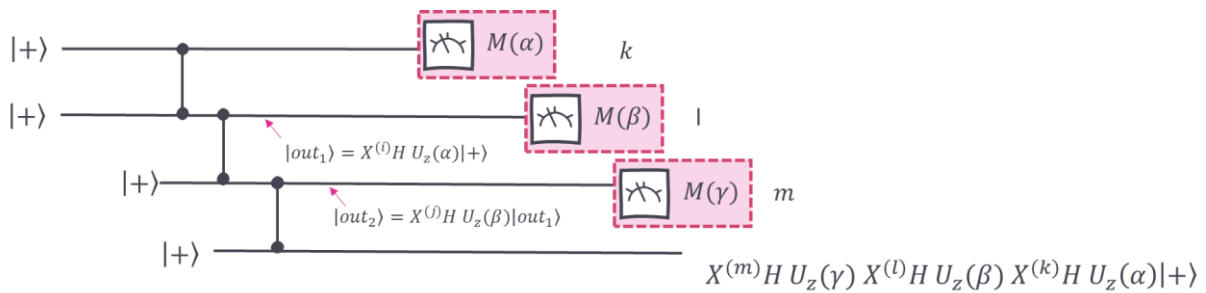


We see that the desired unitary operation is transferred to the output (up to a corrective Pauli gate and a Hadamard) by performing a single-qubit measurement in a modified basis  $M(\alpha)$  :



We can interpret this circuit as a preparation of an entangled resource state CZ  $|+, +\rangle$  followed by a measurement  $M(\alpha)$ , which implements the desired transformation  $X^{(i)}H U_Z(\alpha)|+\rangle$ .

The reader can verify that the measurement  $M(\alpha)$  has the Eigenstates  $|\pm\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm e^{-i\alpha}|1\rangle)$ , and thus corresponds to a projective measurement in the equatorial plane of the Bloch sphere. This shows that we can implement rotations about the Z-axis of the Bloch sphere, but what about general single-qubit unitaries? As you might guess, arbitrary single qubit unitaries – i.e. general rotations on the sphere can be accomplished by concatenation of the above procedure.



The state after three-fold concatenation of the teleportation procedure is given by:

$$|out\rangle = X^{(m)}H U_Z(\gamma) X^{(l)}H U_Z(\beta) X^{(k)}H U_Z(\alpha)|+\rangle$$

Where we notice that the corrective Pauli operations that need to be applied depend on the measurement outcome of the previous measurements. In the following, we see that this will be taken into account by adjusting the measurements conditional on the outcome of previous measurements (so-called feedforward). To show that this three-fold concatenation is in fact sufficient to accomplish the most general rotation on the Bloch sphere, we commute the corrective Pauli gates to the front of the expression. After some basic algebra, and using the identities  $U_Z(\alpha) = \exp\left(-\frac{i\alpha\hat{Z}}{2}\right) = \cos\left(\frac{\alpha}{2}\right)\hat{1} - i\sin\left(\frac{\alpha}{2}\right)\hat{Z}$  and  $U_Z(\alpha)X = XU_Z(-\alpha)$ ,  $XH=HZ$ , the reader can verify that

$$X^{(m)}H U_Z(\gamma) X^{(l)}H U_Z(\beta) X^{(k)}H U_Z(\alpha) \rightarrow X^{(m)}Z^{(l)}X^{(k)}H U_Z((-1)^l\gamma) H U_Z((-1)^k\beta) H U_Z(\alpha)$$

With this we have split into an overall corrective Pauli gates, that can be applied at the end of the computation (those appearing at the beginning of the expression) and gate operations for which the time-ordering is in fact relevant. The direction of rotation induced by the second measurement  $U_Z((-1)^k\beta)$  depends on the outcome “k” of the first. Likewise, the rotation  $U_Z((-1)^l\gamma)$  depends on the outcome “l” of the second measurement. To show how this all plays together in implementing an arbitrary rotation, we re-write the term after the final corrective Pauli gates

$$H U_Z((-1)^l\gamma) H U_Z((-1)^k\beta) H U_Z(\alpha)$$

using

$$HU_Z(\beta)H = H[\cos(\frac{\beta}{2})\hat{1} - i\sin(\frac{\beta}{2})\hat{Z}]H = [\cos(\frac{\beta}{2})\hat{1} - i\sin(\frac{\beta}{2})\hat{X}] = U_x(\beta)$$

Which we recognize (up to an additional Hadamard gate) the well-known Euler decomposition of general rotations:

$$H U_z((-1)^l\gamma) U_x((-1)^k\beta) U_z(\alpha)$$

In conclusion, to achieve a desired arbitrary rotation

$$U_z(\gamma) U_x(\beta) U_z(\alpha)$$

we must simply adjust the measurement basis of the previous measurements  $(k,l)$  to account for the sign changes which would otherwise be incurred. For this reason, the scheme is called “one-way” quantum computing. We can regard this as a sequence of measurements acting on an entangled resource state, a so-called cluster state or graph state. These states are conveniently represented by a graph where vertices denote CZ gates acting on neighboring qubits. In the 4-qubit example above, we have a linear cluster state:

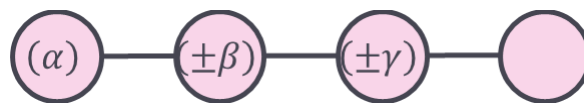


Figure 63 Graphical representation of a linear cluster state of 4-qubits: vertices denote physical qubits and edges denote that the qubits are connected via a CZ operation. Measurements on the linear cluster state propagate the unitary transformation from left to right. Linear cluster states are sufficient to perform any single-qubit unitary.

the unitary transformation is propagated from left to right by performing measurements  $M(\cdot)$ . The linear cluster state is sufficient to implement any possible single-qubit unitary.

To show that the one-way quantum computing is indeed universal, we also need two-qubit gates, which can be achieved by using different resource states. For example, consider we want to implement  $CZ|\alpha\rangle|\beta\rangle$  on qubits  $|\alpha\rangle = U_z(\alpha)|+\rangle$  and  $|\beta\rangle = U_z(\beta)|+\rangle$ . The corresponding circuit is depicted in Figure 64 can be achieved using the “Horseshoe” cluster in Figure 65.

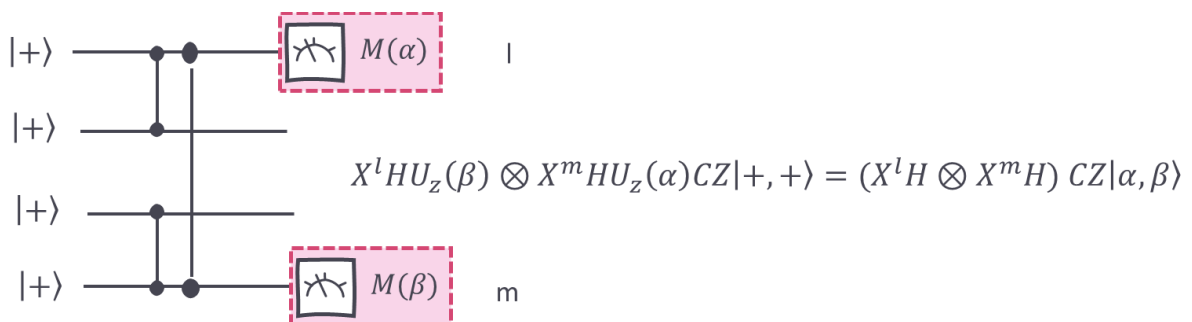


Figure 64: Mapping two-qubit operations from the circuit model to a corresponding cluster state

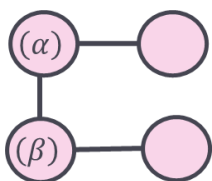


Figure 65 Horseshoe cluster state

In conclusion, the one-way measurement-based scheme is an alternative approach to universal quantum computing. To perform a particular computational task, we need a resource state resource

All notes subject to change, no guarantee to correctness, corrections welcome.

state with a similar degree of connectivity as the corresponding circuit. The challenge of implementing general multi-qubit gates in the circuit model is thus translated into the challenge of preparing large and potentially highly connected resource states.

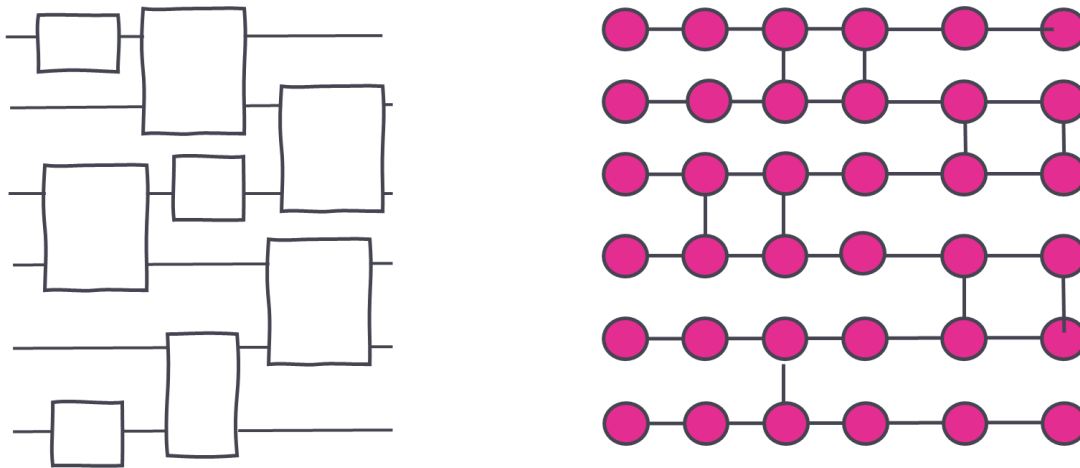


Figure 66: Illustration of quantum circuit and cluster state of similar connectivity. A particular computational task can always be mapped from the circuit model to measurements on a resource state of similar topology. The calculation proceeds from left to right via a sequence of single-qubit measurements, whereby the outcome of preceding measurements have to be considered in the next measurement (feedforward).

### 7.2.1 Definition of Cluster states

We conclude this section with a brief discussion of some of the defining features of cluster states and their graphical representations. To understand some of their features, it is instructive to consider how cluster states can be constructed from unentangled qubits: A general  $n$ -qubit cluster state can be constructed by initializing all  $n$  qubits in the state  $|+\rangle$  and applying pairwise CZ operations to certain qubits. Formally we can write this as:

$$|C\rangle = \prod CZ_{i,n(i)} |+\rangle^{\otimes N}$$

Where  $n(i)$  denotes the neighborhood of qubit  $i$ , that is, all those qubits that are connected to it via CZ operations. Since the CZ operations commute, the order in which these are applied is irrelevant and we can represent the cluster by means of a graph (see preceding examples). A very compact description of cluster states is possible using the stabilizer formalism. Cluster states are positive eigenstate of a group of operators  $S_j |C\rangle = |C\rangle$ , where  $S_j = X_j \prod_{n \in (j)} Z_n$  denotes the stabilizer operator for qubit  $n$  of the cluster state. A detailed discussion can be found in e.g. Kok and Lovett (pages 54-56, and pages 67 ff).



## A 1 The No-Cloning Theorem

The commutation rules for the Pauli-operators have some serious consequences on the type of operations, which can be implemented in a two-Qubit system.

Remember that the state of a (pure) qubit is represented by an arbitrary point on the Poincaré-sphere, depending on the chosen basis vectors and the coefficients  $\alpha$  and  $\beta$  or equally by the angles  $\Theta$  and  $\phi$ . If you attempt to measure its state, you must choose a certain basis in which to measure. This basis is represented by a specific Pauli-Operator or a superposition thereof. However, we have learned, that these operators are complementary, which in essence means, that you only ever get one chance of measuring your polarization state (with a result of  $\pm 1$ ), without permanently and irrevocably destroying the specific state.

If you knew the specific basis in which the qubit was operated, then you'd be quite fine (in the sense of, that you'd only have to determine on which side of the sphere your state is). In general, however, you end up in a situation, where you have absolutely no chance of measuring the complete state of your qubit, unless you have a lot of advance knowledge. Full stop.

To make it simple: a qubit may be any point on the Poincaré-Sphere, i.e. it's defined by two real numbers, but you only ever get to measure on which side of the globe it (most likely) was. And as you cannot copy, what you cannot measure, you end up in a situation that in most of the cases you cannot clone a qubit.

This idea can be proven rigorously, with the two-Qubit notation, which will introduce in the next chapter<sup>12</sup>. Suppose that we have a cloning operator  $\hat{U}$ , which operates on two combined qubits with states  $|\phi\rangle$  and  $|k\rangle$ , such that it copies the state of  $|\phi\rangle$  onto  $|k\rangle$ , i.e.:

$$\hat{U}(|\phi\rangle \otimes |k\rangle) = |\phi\rangle \otimes |\phi\rangle \quad (146)$$

As a cloning-operator  $\hat{U}$  must of course work in the same way for any other state  $|\psi\rangle$ , too, i.e.

$$\hat{U}(|\psi\rangle \otimes |k\rangle) = |\psi\rangle \otimes |\psi\rangle \quad (147)$$

Needless to say, that  $U$  must be connected to a physical process and thus must be unitarian. Let's now compare the two results by taking their scalar product:

$$\begin{aligned} \langle \hat{U}(|\phi\rangle \otimes |k\rangle) | \hat{U}(|\psi\rangle \otimes |k\rangle) \rangle &= \langle \phi \otimes \phi | \psi \otimes \psi \rangle \\ \langle \hat{U}(|\phi\rangle \otimes |k\rangle) | \hat{U}(|\psi\rangle \otimes |k\rangle) \rangle &= \langle \phi \otimes k | \hat{U}^\dagger \hat{U} | \psi \otimes k \rangle = \langle \phi \otimes k | \psi \otimes k \rangle \end{aligned} \quad (148)$$

The first line is simply taken from the definition of the cloning operator, whereas the last line utilized the fact that  $\hat{U}$  is unitarian. Thus we find:

$$\langle \phi \otimes \phi | \psi \otimes \psi \rangle = \langle \phi \otimes k | \psi \otimes k \rangle \quad (149)$$

Because the tensor and the scalar product can be exchanged, we simplify both sides of the equation to:

$$\langle \phi | \psi \rangle \langle \phi | \psi \rangle = \langle \phi | \psi \rangle \langle k | k \rangle \quad (150)$$

Because  $\langle k | k \rangle = 1$  we get:

$$\langle \phi | \psi \rangle^2 = \langle \phi | \psi \rangle \quad (151)$$

<sup>12</sup> In fact, this works for any type of quantum system; qubit or not.

This result is crucial. It can either be fulfilled if  $\langle \phi | \psi \rangle = 1$ , which means that  $|\phi\rangle = |\psi\rangle$ , which is trivial or if  $\langle \phi | \psi \rangle = 0$ , which means that  $|\phi\rangle$  is orthogonal to  $|\psi\rangle$ . In other words: if you have found a cloning operator that works on one specific quantum state (e.g. a Qubit), it can only work on orthogonal quantum states as well but it will not work for arbitrary quantum states. Full stop. Thus, if you cannot find a cloning operator, i.e. any physical process, that copies quantum states, then you cannot copy a quantum state. As long as you have to stick to the laws of nature, that is.

The central argument for the derivation of the no-cloning theorem is obviously the unitarity of  $\hat{U}$ . In terms of time evolution unitarity is equivalent to time-reversibility and thus to a constant entropy: In other words quantum operations must not destroy information in an irrecoverable manner. The supposed cloning-operator, however, would just do that: it would destroy any information of the prior state  $|k\rangle$  of the target system upon it being overwritten with  $|\phi\rangle$ . Thus cloning, from a thermodynamic point of view, is an irreversible process and quantum mechanics just does not provide any means to do that.<sup>13</sup>

---

<sup>13</sup> Note that if you replace  $|k\rangle$  with a many-body thermal bath, then you can “hide” the reversibility in the huge state-space and the fact that most of these states are in reality very hard to differentiate. Reversibility this thus practically impossible.