

MATHEMATICAL ESSENTIALS OF QUANTUM COMPUTING

JUANJO RUÉ* AND SEBASTIAN XAMBÓ

ABSTRACT. The purpose of this expository article is to phrase the essential notions of quantum computation in purely mathematical terms. In particular we will define the notions of q -computation, q -measurement, q -procedure, q -computer and q -algorithm, and each of them will be illustrated with several examples. In addition to some low level q -algorithms, we discuss in detail a good sample of the most relevant discovered in the last years. These include q -algorithms for the Fourier transform, for telling which alternative occurs for a Boolean function that is known to be constant or balanced (Deutsch), for database searching (Grover), for estimating the phase of an eigenvalue of a unitary operator (Kitaev), for finding the multiplicative order of an integer modulo another integer (Shor) and for factoring an integer (Shor). The possible physical realizations of the model, and its potential use to obtain gains with respect to classical algorithms (sometimes even exponential gains), are analyzed in terms of a standard axiomatic formulation of (finite dimensional) quantum theory.

Contents

- Introduction
- 1. Preliminaries
- 2. q -Computations
- 3. q -Measurements and q -Procedures
- 4. q -Computers and q -Algorithms
- 5. Deutsch's and Grover's q -Algorithms
- 6. Phase estimation
- 7. Modular Order of an Integer
- 8. Shor's Factoring q -Algorithm
- 9. Physical Interpretations
- 10. Remarks and Proofs
- References

* Supported by a JAE-DOC grant from the Junta para la Ampliación de Estudios (CSIC), the MTM2011-22851 grant (Spain) and the ICMAT Severo Ochoa Project SEV-2011-0087 (Spain).

INTRODUCTION

The mathematical side of quantum processing, which we will call q -processing, will be presented as a suitable rephrasing of mathematical notions, most notably complex linear algebra and basic notions of elementary probability theory. Our aim is to cover from the most basic concepts up to the expression and analysis of a good sample of the remarkable q -algorithms discovered in the last twenty five years.

Since the link to physics is not addressed until a late section, our approach might be judged as a vacuous game by physicists and engineers, and perhaps even as an inconsequential story by mathematicians. Yet in our experience the approach turns out to be surprisingly powerful and illuminating, and we much hope that this appreciation will be shared by other people as well. Actually, the phrasing of our scheme is crafted in such a way that the tacit physical meaning will be apparent for physicists and, we expect, a reliable basis for mathematicians to appreciate the key physical ideas with minimal effort.

At the earliest stages, the most visible reason for the robustness of the paradigm, and perhaps also for its esthetical appeal, is its close relation with Boolean algebra, the mathematical side of classical computing. This relation is rooted in the fact that the basic playground of q -processing is the complex space $\mathbf{H}^{(n)}$ generated by the set \mathbf{B}^n of binary vectors of length n , which is the basic arena of classical computation.

Later, when the q of q -processing is interpreted as genuine quantum feature, the scheme delivers its full meaning as a mathematical model of interesting physical phenomena that are being intensively explored in labs around the world and which hold a broad range of scientific and technological possibilities for the years to come.

It may be worth reflecting that if computing with classical bits has brought about the ‘digital era’, dominated by information theory and computer science, together with all the enabling technologies, the long term development of the much more comprehensive q -processing is likely to be even mightier and certainly not less interesting.

1. PRELIMINARIES

Let us begin by declaring a number of symbols and conventions that will be used throughout.

- n , a positive integer. We will refer to n as the *number of q -bits*.
- j, k, \dots positive integers in the range $0, \dots, 2^n - 1$.

- $j = j_1j_2 \cdots j_n$, the binary expression of j (and similarly for k). In other words, $j_1, \dots, j_n \in \{0, 1\}$ and $j = j_12^{n-1} + j_22^{n-2} + \cdots + j_{n-1}2 + j_n2^0$.

Binary vectors. Let $\mathbf{B} = \{0, 1\}$, the set of *binary digits* (*bits*). Then the set of binary vectors of length n is \mathbf{B}^n . Its elements are usually written as strings of bits. For example,

$$\mathbf{B}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}.$$

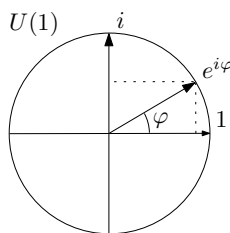
Henceforth the vectors $0 \cdots 0$ and $1 \cdots 1$ will be denoted $\mathbf{0}_n$ and $\mathbf{1}_n$, respectively.

The binary representation of integers allows us to identify the binary vectors of length n with the integers in the range $0, \dots, 2^n - 1$:

$$j \leftrightarrow j_1j_2 \cdots j_n.$$

Since (classical) information can be represented by elements of \mathbf{B}^n , the sets \mathbf{B}^n are the playing ground for classical computations. Actually, a *classical computation of order n* can be seen as a map $f : \mathbf{B}^n \rightarrow \mathbf{B}^n$. If this map is one-to-one, the computation is said to be *reversible*. Since every classical computation can be embedded in a reversible one ($\triangleright \mathbf{1}$),¹ classical computations will be assumed to be reversible unless it is said otherwise explicitly.

Real and complex numbers. The fields of real and complex numbers will be denoted \mathbb{R} and \mathbb{C} , respectively. Given a complex number $a = \alpha + \beta i$ ($\alpha, \beta \in \mathbb{R}$), we write \bar{a} to denote its conjugate: $\bar{a} = \alpha - \beta i$. We have $\bar{a}a = \alpha^2 + \beta^2 = |a|^2$, where $|a|$ denotes the *modulus* or *norm* of a . The complex numbers of modulus 1 have the form $\cos \varphi + i \sin \varphi = e^{i\varphi}$, $\varphi \in \mathbb{R}$ (φ is defined up to integer multiples of 2π). The set $U(1)$ of unit complex numbers is a multiplicative group.



q -Vectors. We will write $\mathbf{H}^{(n)} = \mathbb{C}^{2^n}$ and we will say that this is the space of *q -vectors of order n* . These spaces are the playing ground for q -procedures, just as the sets \mathbf{B}^n are the playing ground for classical computations. In standard mathematical notation, the elements \mathbf{a} in $\mathbf{H}^{(n)}$ are complex column vectors of *dimension* or *length* 2^n :

¹ $\triangleright n$ refers to the note number n in section 10 (Remarks and Proofs), p. 43.

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2^n-1} \end{bmatrix}, \quad a_j \in \mathbb{C}.$$

The q -vector \mathbf{a} can be written in the form

$$\mathbf{a} = a_0 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + a_1 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + a_{2^n-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

This can be compressed as $\mathbf{a} = \sum_j a_j \mathbf{u}_j$, where \mathbf{u}_j is the q -vector whose j -th component is 1 and with all other components 0.

Dirac's notation. Instead of \mathbf{u}_j we will write $|j\rangle$. Thus we have $\mathbf{a} = \sum_j a_j |j\rangle$. Regarding j as a binary vector of length n , we see that $\mathbf{H}^{(n)}$ is the complex vector space with basis \mathbf{B}^n .

Example 1.1 ($n = 1$: One q -bit).

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = a_0|0\rangle + a_1|1\rangle = a_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Example 1.2 ($n = 2$: Two q -bits).

$$\begin{aligned} \mathbf{a} &= a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle \\ &= a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle \\ &= \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = a_{00} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + a_{01} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + a_{10} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + a_{11} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

Here the 0 in $|0\rangle$ is an integer and those in $|00\rangle$ are bits. This abuse of notation is basically harmless, as we can infer the meaning from the context. Note also that $|1\rangle$ (1 an integer) and $|01\rangle$ (0 and 1 are bits) refer to the same basis vector, and that it is required to write the bit 0 on the left.

Example 1.3 (Hadamard q -vector of order n). The *Hadamard q -vector of order n* is defined as

$$\mathbf{h}^{(n)} = \rho^n (|0\rangle + |1\rangle + \cdots + |2^n - 1\rangle),$$

where $\rho = 1/\sqrt{2}$ (see the table on page 49 for a list of frequently used symbols).

Linear maps. Let us recall that a \mathbb{C} -linear map $T : \mathbf{H}^{(n)} \rightarrow \mathbf{H}^{(n)}$ (also called an *operator*) is determined by the 2^n images $\mathbf{t}_j = T|j\rangle$, for

$$T \left(\sum_j a_j |j\rangle \right) = \sum_j a_j T(|j\rangle) = \sum_j a_j \mathbf{t}_j.$$

Moreover, given any set of q -vectors $\{\mathbf{t}_j\}_{0 \leq j < 2^n}$, there is a (unique) linear map $T : \mathbf{H}^{(n)} \rightarrow \mathbf{H}^{(n)}$ such that $T|j\rangle = \mathbf{t}_j$. In the sequel, this observation will be the basic method used to prescribe operators. Properties of T are usually a reflection of properties of the \mathbf{t}_j . For example, the map T is bijective if and only if the vectors \mathbf{t}_j are linearly independent.

Scalar product and norm. If \mathbf{a} and \mathbf{b} be q -vectors, the *bracket product*, or just the bracket, of \mathbf{a} and \mathbf{b} , $\langle \mathbf{a} | \mathbf{b} \rangle$, is defined by

$$\langle \mathbf{a} | \mathbf{b} \rangle = \sum_j \bar{a}_j b_j.$$

It is linear in \mathbf{b} and conjugate-linear in \mathbf{a} . If $\langle \mathbf{a} | \mathbf{b} \rangle = 0$, we say that \mathbf{a} and \mathbf{b} are *orthogonal*. Note that $\langle \mathbf{a} | \mathbf{a} \rangle = |\mathbf{a}|^2$, where

$$|\mathbf{a}|^2 = |a_0|^2 + |a_1|^2 + \dots + |a_{2^n-1}|^2$$

(the *norm squared* of \mathbf{a}). If $|\mathbf{a}| = 1$, we say that \mathbf{a} is a *unit vector*. If \mathbf{x} is any non-zero q -vector, $\mathbf{x}/|\mathbf{x}|$ is a unit q -vector. This vector is denoted $\hat{\mathbf{x}}$ or $\mathbf{u}(\mathbf{x})$.

Note also that $\langle \mathbf{u}_j | \mathbf{u}_k \rangle = \delta_{ij}$. This means that the \mathbf{u}_j are pairwise orthogonal unit vectors (this property is expressed by saying that the $\{\mathbf{u}_j\}$ form an *orthonormal basis*). In Dirac's notation this is written as $\langle j | k \rangle = \delta_{jk}$.

Tensor product. Let $\mathbf{a} \in \mathbf{H}^{(n)}$ and $\mathbf{a}' \in \mathbf{H}^{(n')}$. Then the *tensor product* of \mathbf{a} and \mathbf{a}' , denoted $\mathbf{a} \hat{\otimes} \mathbf{a}'$, is the vector in $\mathbf{H}^{(n+n')}$ whose components are $a_j a'_{j'}$, with (j, j') in lexicographic order.

For example,

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \hat{\otimes} \begin{bmatrix} a'_0 \\ a'_1 \end{bmatrix} = \begin{bmatrix} a_0 a'_0 \\ a_0 a'_1 \\ a_1 a'_0 \\ a_1 a'_1 \end{bmatrix}.$$

Proposition 1.4. If $\mathbf{a} = \sum_j a_j |j\rangle$ and $\mathbf{a}' = \sum_{j'} a'_{j'} |j'\rangle$, then

$$\mathbf{a} \hat{\otimes} \mathbf{a}' = \sum_{j, j'} a_j a'_{j'} |j \cdot 2^{n'} + j'\rangle.$$

Proof. The index $j \in \{0, \dots, 2^n - 1\}$ of the entry a_j of a vector \mathbf{a} is the number of entries preceding it (these entries are a_0, a_1, \dots, a_{j-1}). Now the claim follows because the number of entries preceding the entry $a_j a'_{j'}$ in $\mathbf{a} \hat{\otimes} \mathbf{a}'$ is $j' + j \cdot 2^{n'}$. \square

If we think of j and j' as binary numbers, then $j \cdot 2^{n'} + j'$ is just the binary number obtained by concatenating the binary representations of j and j' . In practice this number is simply written as jj' , and so we get the following expression for the tensor product:

$$\mathbf{a} \widehat{\otimes} \mathbf{a}' = \sum_{j, j'} a_j a'_{j'} |jj'\rangle.$$

The tensor product $|j\rangle \widehat{\otimes} |j'\rangle$ is also denoted $|j\rangle |j'\rangle$, so that we have

$$|j\rangle \widehat{\otimes} |j'\rangle = |j\rangle |j'\rangle = |jj'\rangle.$$

In particular we can write

$$|j_1\rangle \widehat{\otimes} |j_2\rangle \widehat{\otimes} \cdots \widehat{\otimes} |j_n\rangle = |j_1\rangle |j_2\rangle \cdots |j_n\rangle = |j_1 j_2 \cdots j_n\rangle$$

Example 1.5. Let $\mathbf{h}^{(n)}$ be the Hadamard q -vector of order n (see Example 1.3). Then

$$\mathbf{h}^{(n)} = \rho^n (|0\rangle + |1\rangle) \widehat{\otimes} (|0\rangle + |1\rangle) \widehat{\otimes} \cdots \widehat{\otimes} (|0\rangle + |1\rangle).$$

In fact, the right hand side is equal to

$$\rho^n \sum_{j_1, \dots, j_n \in \mathbf{B}} |j_1\rangle \cdots |j_n\rangle = \rho^n \sum_{j \in \mathbf{B}^n} |j\rangle = \mathbf{h}^{(n)}.$$

Remark 1.6. The map $\mathbf{H}^{(n)} \times \mathbf{H}^{(n')} \rightarrow \mathbf{H}^{(n+n')}$, $(\mathbf{a}, \mathbf{a}') \mapsto \mathbf{a} \widehat{\otimes} \mathbf{a}'$ is bilinear, and hence induces a linear map

$$\mathbf{H}^{(n)} \otimes \mathbf{H}^{(n')} \rightarrow \mathbf{H}^{(n+n')}, \quad \mathbf{a} \otimes \mathbf{a}' \mapsto \mathbf{a} \widehat{\otimes} \mathbf{a}'.$$

This map is an isomorphism, as it sends the basis $\{|j\rangle \otimes |j'\rangle\}$ of $\mathbf{H}^{(n)} \otimes \mathbf{H}^{(n')}$ to the basis $\{|jj'\rangle\}$ of $\mathbf{H}^{(n+n')}$. In particular we have an isomorphism

$$\mathbf{H}^{(n)} \simeq \mathbf{H}^{(1)} \otimes \cdots \otimes \mathbf{H}^{(1)}.$$

Because of the isomorphism given by $\mathbf{a} \otimes \mathbf{a}' \mapsto \mathbf{a} \widehat{\otimes} \mathbf{a}'$, from now on we will write \otimes instead of $\widehat{\otimes}$.

Remark 1.7. A q -vector of order n of the form $\mathbf{a}_1 \otimes \cdots \otimes \mathbf{a}_n$, $\mathbf{a}_l \in \mathbf{H}^{(1)}$, is said to be *decomposable*. The basis vectors $|j_1 j_2 \cdots j_n\rangle = |j_1\rangle |j_2\rangle \cdots |j_n\rangle$ are examples of decomposable vectors. In general, however, q -vectors are not decomposable. For example, it is easy to check that $|00\rangle + |11\rangle \in \mathbf{H}^{(2)}$ cannot be written in the form $(a_0|0\rangle + a_1|1\rangle)(b_0|0\rangle + b_1|1\rangle)$. Using a terminology originated in physics (see page 39), non-decomposable q -vectors are said to be *entangled* and decomposable q -vectors are also called *composite*.

2. q -COMPUTATIONS

If $U = [u_{jk}]$ is a matrix, its *transpose* is $U^T = [u_{kj}]$ and its *adjoint* is

$$U^\dagger = [\bar{u}_{kj}] = \overline{U^T}.$$

A q -computation of order n is a unitary matrix U of dimension 2^n . This means that

$$U = [u_{jk}]_{0 \leq j, k < 2^n}, \quad u_{jk} \in \mathbb{C}$$

and

$$UU^\dagger = I_{2^n},$$

where I_{2^n} denotes the identity matrix of dimension 2^n .

The set of unitary matrices of dimension 2^n , $\mathbf{U}(2^n)$, will be denoted $\mathbf{U}^{(n)}$. With the standard multiplication of matrices, $\mathbf{U}^{(n)}$ is a group. In detail, this means the following:

- (*Identity*) $I_{2^n} \in \mathbf{U}^{(n)}$. In words, the identity matrix of dimension 2^n is a q -computation of order n .
- (*Composition*) If $U, V \in \mathbf{U}^{(n)}$, then $VU \in \mathbf{U}^{(n)}$. Thus the composition of two q -computations of order n is a q -computation of order n .
- (*Reversibility*) If $U \in \mathbf{U}^{(n)}$, then $U^{-1} \in \mathbf{U}^{(n)}$ (notice that $U^{-1} = U^\dagger$). The inverse of a q -computation of order n is a q -computation of order n .

If $\mathbf{a} \in \mathbf{H}^{(n)}$ is a unit vector and U a q -computation, then $\mathbf{b} = U\mathbf{a}$ is also a unit vector. As in classical computations, we say that \mathbf{b} is the q -output of U with q -input \mathbf{a} .

Example 2.1. If $U \in \mathbf{U}^{(n)}$ and $U' \in \mathbf{U}^{(n')}$, define the map

$$U \otimes U' : \mathbf{H}^{(n+n')} \rightarrow \mathbf{H}^{(n+n')}$$

such that $|jj'\rangle = |j\rangle|j'\rangle \mapsto U|j\rangle U'|j'\rangle$. Then $U \otimes U' \in \mathbf{U}^{(n+n')}$. This map actually satisfies $|\mathbf{a}\rangle|\mathbf{a}'\rangle \mapsto U|\mathbf{a}\rangle U'|\mathbf{a}'\rangle$ for all $\mathbf{a} \in \mathbf{H}^{(n)}$ and $\mathbf{a}' \in \mathbf{H}^{(n')}$.

Similarly, if $U \in \mathbf{U}^{(1)}$, then we can define $U^{\otimes n} \in \mathbf{U}^{(n)}$ by the relation

$$U^{\otimes n}|j\rangle = U|j_1\rangle U|j_2\rangle \cdots U|j_n\rangle.$$

Example 2.2 (Classical reversible computations). Given a classical computation of order n , $f : \mathbf{B}^n \rightarrow \mathbf{B}^n$, we can define a linear map $U_f : \mathbf{H}^{(n)} \rightarrow \mathbf{H}^{(n)}$ by the relations

$$U_f|j\rangle = |f(j)\rangle.$$

If f is reversible, then U_f is a q -computation, as it maps the orthonormal basis $\{|j\rangle\}$ to the orthonormal basis $\{|f(j)\rangle\}$ (a permutation of the former). We will say that U_f is the q -computation associated to the classical reversible computation f .

Some of the examples below are just special cases of this general example.

Remark 2.3. The q -computations of order n are vastly more abundant than the classical reversible computations of order n . This is already clear for $n = 1$, for in this case NOT is the only classical reversible computation of order 1 different from the identity, whereas the q -computations of order 1 depend on four continuous parameters (cf. Example 2.4).

Graphical representation. A q -computation U of order n is often represented by a diagram like this:

$$n \left\{ \begin{array}{c} \boxed{U} \\ \hline \hline \hline \hline \hline \hline \end{array} \right.$$

The n horizontal lines are called q -wires.

If we want to represent the q -input \mathbf{a} and q -output \mathbf{b} , the diagram can be modified as follows:

$$n \left\{ \begin{array}{c} \boxed{U} \\ \hline \hline \hline \hline \hline \hline \end{array} \right. \begin{array}{l} \mathbf{a} \\ \mathbf{b} \end{array}$$

In the case where $\mathbf{a} = |j_1\rangle \cdots |j_n\rangle$, the input is represented as follows:

$$\begin{array}{l} |j_1\rangle \text{ —————} \\ |j_2\rangle \text{ —————} \\ \vdots \\ |j_{n-1}\rangle \text{ —————} \\ |j_n\rangle \text{ —————} \end{array}$$

Example 2.4 ($n = 1$). It is easy to check that the matrix

$$U = e^{i\alpha} \begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix}, \quad \alpha \in \mathbb{R}, \quad u_0, u_1 \in \mathbb{C}, \quad u_0\bar{u}_0 + u_1\bar{u}_1 = 1$$

is a q -computation of order 1. Actually any $U \in \mathbf{U}^{(1)}$ has this form. Indeed, we may write $U = e^{i\alpha}U'$ with $\det(U') = 1$ (this follows from $UU^\dagger = I_2$, so that $\det(U)$ is a unit complex) and then $U' \in \mathbf{U}^{(1)}$ has necessarily the form

$$\begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix} \text{ if its inverse has to coincide with its adjoint.}$$

$$\text{—} \boxed{U} \text{—}$$

Remark 2.5. For later reference, let us consider a more explicit construction of the q -computations of order 1.² With the notations of the previous example,

²Here, and in the remaining of the section, we closely follow [15], Section 4.2.

the relation $u_0\bar{u}_0 + u_1\bar{u}_1 = 1$ implies that there is a unique $\theta \in [0, \pi]$ such that $|u_0| = \cos \frac{\theta}{2}$ and $|u_1| = \sin \frac{\theta}{2}$. It follows that $u_0 = e^{-i\lambda} \cos \frac{\theta}{2}$ and $u_1 = -e^{i\mu} \sin \frac{\theta}{2}$, $\lambda, \mu \in \mathbb{R}$ (the choice of signs is for later convenience) and so

$$\begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix} = \begin{bmatrix} e^{-i\lambda} \cos \frac{\theta}{2} & -e^{i\mu} \sin \frac{\theta}{2} \\ e^{-i\mu} \sin \frac{\theta}{2} & e^{i\lambda} \cos \frac{\theta}{2} \end{bmatrix}$$

If we do the replacements $\lambda = (\beta + \gamma)/2$, $\mu = (\gamma - \beta)/2$ ($\beta, \gamma \in \mathbb{R}$), then we can write

$$\begin{bmatrix} e^{-i\lambda} \cos \frac{\theta}{2} & -e^{i\mu} \sin \frac{\theta}{2} \\ e^{-i\mu} \sin \frac{\theta}{2} & e^{i\lambda} \cos \frac{\theta}{2} \end{bmatrix} = R_z(\beta)R_y(\theta)R_z(\gamma)$$

where we define

$$R_z(\varphi) = \begin{bmatrix} e^{-i\varphi/2} & 0 \\ 0 & e^{i\varphi/2} \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}.$$

We therefore conclude that a general element of $SU(2)$ (the elements of $U(2)$ with determinant 1) has the form

$$U(\theta, \beta, \gamma) = R_z(\beta)R_y(\theta)R_z(\gamma).$$

The geometrical meaning of this statement, closely related to rotations in Euclidean 3-space, will be considered in Section 9.

Special cases

a) *Pauli matrices*

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The Pauli matrices are self-adjoint and unitary: $X^2 = Y^2 = Z^2 = \mathbf{1}$.

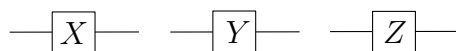
Notice that the matrix X can be defined by the relations

$$X|0\rangle = |1\rangle, X|1\rangle = |0\rangle.$$

This means that X is the q -computation corresponding to the classical Boolean operator NOT (negation):

$$X|j\rangle = |\text{NOT}(j)\rangle.$$

In terms of the Boolean sum of bits, we have $\text{NOT}(j) = 1 + j$, as $1 + 0 = 1$ and $1 + 1 = 0$. Briefly, $X|j\rangle = |1 + j\rangle$.



Remark 2.6. With the notations of Remark 2.5, we have

$$R_z(\varphi) = \cos \frac{\varphi}{2} I_2 - i \sin \frac{\varphi}{2} Z = e^{-i \frac{\varphi}{2} Z}$$

$$R_y(\theta) = \cos \frac{\theta}{2} I_2 - i \sin \frac{\theta}{2} Y = e^{-i \frac{\theta}{2} Y}.$$

This suggests defining

$$R_x(\psi) = \cos \frac{\psi}{2} I_2 - i \sin \frac{\psi}{2} X = e^{-i \frac{\psi}{2} X} = \begin{bmatrix} \cos \frac{\psi}{2} & -i \sin \frac{\psi}{2} \\ -i \sin \frac{\psi}{2} & \cos \frac{\psi}{2} \end{bmatrix}.$$

One further observation is that every $U \in \mathbf{U}^{(1)}$ can be expressed as

$$U = e^{i\alpha} A X B X C,$$

with $A, B, C \in S\mathbf{U}^{(1)}$ and $ABC = I_2$ (this will be called an *Euler decomposition* of U). Indeed, if $U = e^{i\alpha} R_z(\beta) R_y(\theta) R_z(\gamma)$, it is enough to set

$$A = R_z(\beta) R_y(\frac{\theta}{2})$$

$$B = R_y(-\frac{\theta}{2}) R_z(-\frac{\beta+\gamma}{2})$$

$$C = R_z(\frac{\gamma-\beta}{2})$$

The proof is a straightforward computation ($\triangleright \mathbf{2}$).

b) *The Hadamard matrix H*

The matrix $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is self-adjoint and $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^2 = 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. It follows that the matrix

$$H = \rho \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \begin{cases} |0\rangle \mapsto \rho(|0\rangle + |1\rangle) \\ |1\rangle \mapsto \rho(|0\rangle - |1\rangle) \end{cases}$$

is a q -computation of order 1. To write H in the form of Example 2.4, note that

$$H = (-i)(iH) \text{ and check that } iH \text{ has the form } \begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix} \text{ with } u_0 = u_1 = i.$$

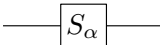
$$\text{---} \boxed{H} \text{---}$$

The q -computation $H^{\otimes n}$ (see the second part of the Example 2.1) will appear often in the following. Notice that using the notation of Example 1.3 we have the following identity:

$$H^{\otimes n}(|\mathbf{0}_n\rangle) = \mathbf{h}^{(n)}.$$

c) *Phase shift matrices*

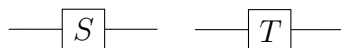
These are the matrices of the form

$$S_\alpha = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} = e^{i\frac{\alpha}{2}} \begin{bmatrix} e^{-i\frac{\alpha}{2}} & 0 \\ 0 & e^{i\frac{\alpha}{2}} \end{bmatrix}$$


In particular we define

$$S = S_{\pi/2} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \text{ and } T = S_{\pi/4} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Note that $T^4 = S^2 = X$, which is sometimes written as $S = \sqrt{\text{NOT}}$.



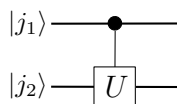
Example 2.7 ($n = 2$). Let $U \in \mathbf{U}^{(1)}$. Define $C_{12}(U) \in \mathbf{U}^{(2)}$ as follows:

$$C_{12}(U)|0j_2\rangle = |0j_2\rangle, \quad C_{12}(U)|1j_2\rangle = |1\rangle U|j_2\rangle.$$

In matrix form we have, if $U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}$,

$$C_{12}(U) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

Note that this is a form of Controlled- U , as U acts on $|j_2\rangle$ only when $j_1 = 1$.



In particular we set $N_{12} = C_{12}(X)$ (Controlled-NOT, or CNOT):

$$N_{12}|0j_2\rangle = |0j_2\rangle, \quad N_{12}|1j_2\rangle = |1\rangle|1 + j_2\rangle,$$

which can be written more compactly as

$$N_{12}|j_1j_2\rangle = |j_1\rangle|j_1 + j_2\rangle.$$

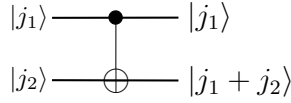
This shows that $N_{12} : \mathbf{H}^{(2)} \rightarrow \mathbf{H}^{(2)}$ is the q -computation corresponding to the classical computation $\text{CNOT} : \mathbf{B}^2 \rightarrow \mathbf{B}^2$ such that

$$00 \mapsto 00, \quad 01 \mapsto 01, \quad 10 \mapsto 11, \quad 11 \mapsto 10$$

or $\text{CNOT}(j_1, j_2) = (j_1, j_1 + j_2)$. Since its action amounts to negating the second bit if and only if the first bit is 1, it is a NOT on the second bit “controlled” by the first bit, which means that it is conditional to the first bit having the value 1.

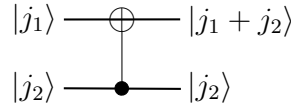
In matrix form,

$$N_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$



$C_{21}(U)$ is defined in an analogous way. For example, the matrix form of $N_{21} = C_{21}(X)$ is

$$N_{21} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



More generally, in the case of order n , we can define the q -computations $C_{r,s}(U)$, where $r, s \in \{1, \dots, n\}$ are two distinct position indices, in the same way. This q -computation acts by U on the s -th bit when the r -th bit is 1, and otherwise acts as the identity. In the special case $U = X$, we let $N_{r,s} = C_{r,s}(X)$. This negates the s -th q -bit if and only if the value of the r -th q -bit is 1. Here are some illustrations:

$$\begin{aligned} N_{4,1}|10101\rangle &= |10101\rangle, & C_{4,1}(U)|10101\rangle &= |10101\rangle \\ N_{4,1}|10111\rangle &= |00111\rangle, & C_{4,1}(U)|10111\rangle &= (U|1\rangle)|0111\rangle. \end{aligned}$$

Example 2.8. The q -computation $C_{12}(U)$ is quite different from $I_2 \otimes U$. In fact, the matrix of the latter is

$$\begin{bmatrix} u_{00} & u_{01} & 0 & 0 \\ u_{10} & u_{11} & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

Thus $(I_2 \otimes U)|00\rangle = |0\rangle U|0\rangle = u_{00}|0\rangle|0\rangle + u_{10}|0\rangle|1\rangle$, whereas $C_{12}(U)|00\rangle = |00\rangle$.

Example 2.9 (No cloning). In its most basic form, the *no-cloning theorem* is the assertion that there is no q -computation U of order 2 that satisfies

$$U(|b\rangle|0\rangle) = |b\rangle|b\rangle,$$

for $b \in \{0, 1\}$. Indeed, consider $|x\rangle = \rho(|b\rangle + |b'\rangle)$ and $b' = 1 + b$. Then we have

$$U(|x\rangle|0\rangle) = \begin{cases} |x\rangle|x\rangle = \rho^2(|b\rangle|b\rangle + |b\rangle|b'\rangle + |b'\rangle|b\rangle + |b'\rangle|b'\rangle) \\ \rho U(|b\rangle|0\rangle + |b'\rangle|0\rangle) = \rho(|b\rangle|b\rangle + |b'\rangle|b'\rangle) \end{cases},$$

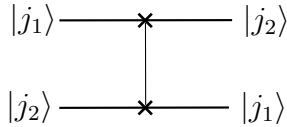
which is a contradiction.

Example 2.10. The *swap gate* is the q -computation of order 2 corresponding to the classical swap $j_1j_2 \mapsto j_2j_1$:

$$|j_1j_2\rangle \mapsto |j_2j_1\rangle$$

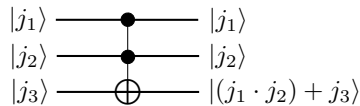
Since it leaves $|00\rangle$ and $|11\rangle$ fixed and interchanges $|01\rangle$ and $|10\rangle$, its matrix is

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



Example 2.11. The *Toffoli gate* is the q -computation of order 3 corresponding to the classical computation $j_1j_2j_3 \mapsto (j_1 \cdot j_2) + j_3$. It negates the bit j_3 precisely when $j_1 = j_2 = 1$, so it is a doubly controlled negation. It interchanges $|110\rangle$ and $|111\rangle$, leaving all other basis vectors fixed. It follows that its matrix is

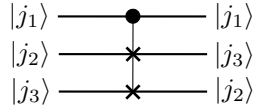
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



The Toffoli gate can be seen as the q -computation of order 3 corresponding to a classical reversible computation of order 3 that defines a reversible version of NAND: if $j_3 = 1$, then $j'_3 = \text{NAND}(j_1, j_2)$. For further details, $\triangleright \mathbf{1}$.

Example 2.12. The *Fredkin gate* is the q -computation of order 3 corresponding to the classical computation $0j_2j_3 \mapsto 0j_2j_3$ and $1j_2j_3 \mapsto 1j_3j_2$. In other words, it is a controlled-swap. It interchanges $|110\rangle$ and $|101\rangle$ and leaves all other basis vectors fixed. Hence its matrix is

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



Note that if $j_3 = 0$, then $j'_3 = \text{AND}(j_1, j_2)$. If in addition $j_2 = 1$, then $j'_2 = \text{NOT}(j_1)$. Thus the Fredkin gate can also be used to implement a reversible version of classical computation.

3. q -MEASUREMENTS AND q -PROCEDURES

In addition to the q -computations, in order to produce a mathematical model of a quantum computation we need to include the operation of measuring a set $L = \{l_1, \dots, l_r\} \subseteq \{1, \dots, n\}$ of q -bits. In the following, $\mathbf{a} \in \mathbf{H}^{(n)}$ will denote a unit q -vector. We think of this vector as the current state of a q -register of length n (the q -memory).

q -Measurements. For any binary vector of length r , say $M = m_1 \cdots m_r \in \mathbf{B}^n$, we can form the subspace $E_M \subseteq \mathbf{H}^{(n)}$ generated by the basis vectors $|j\rangle$ such that $j_L = M$, where $j_L = j_{l_1} \cdots j_{l_r}$. Its dimension is 2^{n-r} . The orthogonal projection of \mathbf{a} to E_M is the vector

$$\mathbf{a}_L^M = \sum_{j_L=M} a_j |j\rangle.$$

Since the spaces E_M , $M \in \mathbf{B}^n$, are pairwise orthogonal and $\oplus_M E_M = \mathbf{H}^{(n)}$, we conclude that

$$\mathbf{a} = \sum_{M \in \mathbf{B}^n} \mathbf{a}_L^M$$

and that

$$1 = |\mathbf{a}|^2 = \sum_M |\mathbf{a}_L^M|^2,$$

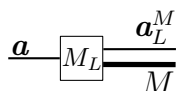
so that the numbers $p_M = |\mathbf{a}_L^M|^2$ define a probability distribution on $\{M\} = \mathbf{B}^r$. The q -measurement or q -observation of the q -bits at the positions l_1, \dots, l_r , which we will denote $M_L(\mathbf{a})$, is defined as consisting of the following two aspects:

- (i) It extracts an $M \in \mathbf{B}^n$ at random, with probability p_M ;
- (ii) it resets the q -memory state to $\mathbf{u}(\mathbf{a}_L^M)$, where $\mathbf{u}(\mathbf{x}) = \hat{\mathbf{x}}$ denotes the unit vector defined by the non-zero q -vector \mathbf{x} .

Notice that if M is the outcome of a trial, then $\mathbf{a}_L^M \neq 0$.

The q -vectors \mathbf{a}_L^M are said to be the *collapses* of \mathbf{a} with respect to the positions L .

Since a q -measurement produces a classical binary vector M and a new q -memory state, M_L will be represented graphically as



The fat line on the right represents the classical value supplied by the q -measurement and the thin one the collapsed state.

Example 3.1. As an illustration, let's look to a few special cases. Consider the case $n = 3$. If we measure the third q -bit, $M_3(\mathbf{a})$, then there are two possible values, 0 and 1, and the corresponding collapses are $\mathbf{a}_3^0 = \sum_{jk} a_{jk0} |jk0\rangle$ and $\mathbf{a}_3^1 = \sum_{jk} a_{jk1} |jk1\rangle$. Their probabilities are $p_0 = |\mathbf{a}_3^0|^2$ and $p_1 = |\mathbf{a}_3^1|^2$. Similarly, for $M_{13}(\mathbf{a})$ there are four possible outcomes and the corresponding collapses are $\mathbf{a}_{13}^{rs} = a_{r0s} |r0s\rangle + a_{r1s} |r1s\rangle$ ($rs \in \mathbf{B}^2$), with probabilities $p_{rs} = |\mathbf{a}_{13}^{rs}|^2$.

In the case when $L = \{1, \dots, n\}$, we write simply $M(\mathbf{a})$. In that case the possible outcomes are the elements $j \in \mathbf{B}^n$ and the corresponding collapses are $a_j |j\rangle$, with probabilities $|a_j|^2$. In this context, the coefficient a_j is usually called the (probability) *amplitude* of $|j\rangle$, and the probability of this result is $p_j = |a_j|^2$: *the probability is the norm squared of the amplitude*. If $n = 3$, for instance, then there are eight possible outcomes for $M(\mathbf{a}) = M_{123}(\mathbf{a})$ and the corresponding collapses are $\mathbf{a}_{123}^{rst} = a_{rst} |rst\rangle$ ($rst \in \mathbf{B}^3$) with probabilities $p_{rst} = |a_{rst}|^2$.

A *q-procedure* is a sequence of actions, each of which is either a *q-computation* or a *q-measurement*, that are applied successively to $|0 \cdots 0\rangle$ (this is the *default* initial state of the *q-memory*). Since procedures are meant to produce results, usually the last action is a *q-measurement*.

Example 3.2 (Random generator). The following *q-procedure* outputs random numbers of n bits with a uniform probability distribution:

RANDOM

$$\mathbf{a} = H^{\otimes n}|0 \dots 0\rangle = H|0\rangle \cdots H|0\rangle, \quad M(\mathbf{a}) \blacksquare$$

Indeed, from the Example 1.3 we know that \mathbf{a} is the Hadamard *q-vector* $\mathbf{h}^{(n)}$ and the amplitude of any binary vector of length n is $1/\rho^n$, and so its probability is $(1/\rho^n)^2 = 1/2^n$.

4. *q*-COMPUTERS AND *q*-ALGORITHMS

A *q-computer of order n* is a system endowed with the following operations:

1. *q-Memory*

A store capable of holding any unit *q-vector* $\mathbf{a} \in \mathbf{H}^{(n)}$. We will say that \mathbf{a} is the *q-memory state*, or simply the state.

The operations 2, 3 and 4 below, which we will call *elementary procedures*, are to be applied successively, one at a time, to the current state. We also assume that when the *q-computer* is switched on, the *q-memory* is set to the state $|0 \cdots 0\rangle = |\mathbf{0}_n\rangle$ (*default state*).

2. *One q-bit rotations* $R_l(U)$, $U \in \mathbf{U}^{(1)}$

This is the action of U on the l -th *q-bit*. More precisely, it is the *q-computation* defined as follows:

$$|\cdots j_l \cdots\rangle = |\cdots\rangle |j_l\rangle |\cdots\rangle \mapsto |\cdots\rangle U|j_l\rangle |\cdots\rangle.$$

For $n = 2$, for instance, $R_2(U) = I_2 \otimes U$ (cf. Example 2.8).

This kind of elementary *q-procedures* will be called *U-gates*. An *U-gate* will be called *restricted* if U is the Hadamard matrix H or one of the phase shift matrices $S = S_{\pi/2}$ or $T = S_{\pi/4}$.

3. *Controlled negations* $N_{r,s}$

This *q-computation* negates the s -th *q-bit* if (and only if) the r -th *q-bit* is $|1\rangle$. It is the linear map which is the identity on the basis *q-vectors* of the form

$|\cdots 0_j \cdots\rangle$ and such that

$$\begin{aligned} |\cdots 1_j \cdots 0_k \cdots\rangle &\mapsto |\cdots 1_j \cdots 1_k \cdots\rangle \\ |\cdots 1_j \cdots 1_k \cdots\rangle &\mapsto |\cdots 1_j \cdots 0_k \cdots\rangle \end{aligned}$$

This kind of elementary q -procedures will be called CNOT *gates*.

4. Measurement $M_L(\mathbf{a})$, $L = \{l_1 < \cdots < l_r\} \subseteq \{1, \dots, n\}$

This elementary procedure has been explained in detail in the previous section.

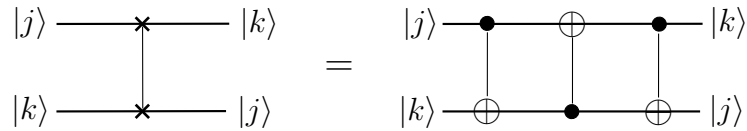
q -Algorithms. A q -algorithm is a q -procedure involving elementary q -procedures only. We will say that a q -algorithm is *internal* if it does not involve measurements. A q -algorithm (internal or not) will be called *restricted* if it only involves restricted U -gates.

As a measure of the *complexity* of a q -algorithm we take the number of elementary gates it involves. A q -algorithm is *polynomial* if its complexity is bounded by a polynomial in n .

Example 4.1 (SWAP[r, s]). This internal q -algorithm is defined as follows:

SWAP[r, s]

$N_{r,s}, N_{s,r}, N_{r,s}$ ■



The q -computation performed by this algorithm amounts to interchanging the r -th and s -th q -bits, which means that it is equal to the linear map defined by

$$|\cdots j_r \cdots j_s \cdots\rangle \mapsto |\cdots j_s \cdots j_r \cdots\rangle$$

This statement is a direct consequence of the fact that it holds for classical computations. Indeed, for any pair of bits, (x, y) , we have:

$$\begin{aligned} N_{1,2}(x, y) &= (x, x + y), \\ N_{2,1}(x, x + y) &= (x + x + y, x + y) = (y, x + y), \\ N_{1,2}(y, x + y) &= (y, y + x + y) = (y, x). \end{aligned}$$

Example 4.2 (Multiple H). Consists in applying the Hadamard gate H at any index on a given list $L \subseteq \{1, \dots, n\}$ of positions:

HADAMARD[L]

for $l \in L$ do $R_l(H)$ ■

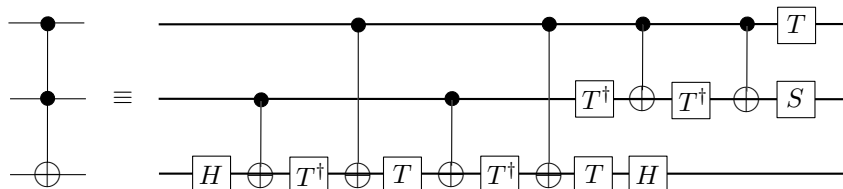
Remark that if $m \in \{1, \dots, n\}$, $\text{HADAMARD}[\{1, \dots, m\}]$ yields a q -algorithm for the q -procedure $|j\rangle \mapsto (H^{\otimes m}|j_1 \cdots j_m\rangle)|j_{m+1} \cdots j_n\rangle$. This algorithm will be denoted $\text{HADAMARD}[m]$. In the case $m = n$, it is a q -algorithm for $H^{\otimes n}$ and instead of $\text{HADAMARD}[n]$ we will simply write HADAMARD .

Similar algorithms can be devised replacing H by any $U \in \mathbf{U}^{(1)}$. For example, $U^{\otimes n}$ can be computed by the following q -algorithm:

for $l \in \{1, \dots, n\}$ do $R_l(U)$ ■

Example 4.3 (q -algorithm associated to a classical algorithm).

If $f: \mathbf{B}^n \rightarrow \mathbf{B}^n$ is a reversible computation, then there is a classical algorithm that computes f . This algorithm is a sequence of logical gates that are either a NOT or a NAND. Even more, adding auxiliary bits if necessary, we may assume that the NAND are reversible. Then the algorithm can be translated into a q -procedure consisting of q -gates that are either a Pauli X gate or a Toffoli gate. This q -procedure will become a q -algorithm if we can find a q -algorithm for the Toffoli gate. A solution to this question can be expressed by the following diagram (cf. [15], Exercise 4.24):



TOFFOLI

$R_3(H), N_{2,3}, R_3(T^\dagger), N_{1,3}, R_3(T), N_{2,3}, R_3(T^\dagger), N_{1,3}, R_3(T), R_3(H),$
 $R_2(T^\dagger), N_{2,3}, R_2(T^\dagger), N_{1,2}, R_2(S), R_1(T)$ ■

Example 4.4. In this example we show an algorithm for the q -procedure $C_{r,s}(U)$, $U \in \mathbf{U}^{(1)}$, $r, s \in \{1, \dots, n\}$ distinct position indices (see the Example 2.7 for the definition). For this it we will use an Euler decomposition of U (see Remark 2.6):

$$U = e^{i\alpha} AXBXC, \quad ABC = I_2.$$

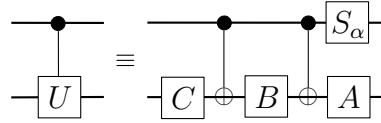
With these notations, the q -algorithm is as follows:

CONTROL $[r, s, U]$

$R_s(C), N_{r,s}, R_s(B), N_{r,s}, R_s(A), R_r(S_\alpha)$ ■

We shall assume $r = 1$ and $s = 2$, as the argument can be easily adapted to the general case. Note that if $j_1 = 0$, then the $N_{1,2}$ and $R_1(S_\alpha)$ act as the identity and hence, since $ABC = I_2$, CONTROL also acts as the identity. If $j_1 = 1$, then the action on $|j_2\rangle$ is $AXBXC|j_2\rangle$ and on $|j_1\rangle = |1\rangle$ by the phase factor $e^{i\alpha}$:

$$|1\rangle|j_2\rangle \mapsto e^{i\alpha}|1\rangle AXBXC|j_2\rangle = |1\rangle U|j_2\rangle.$$



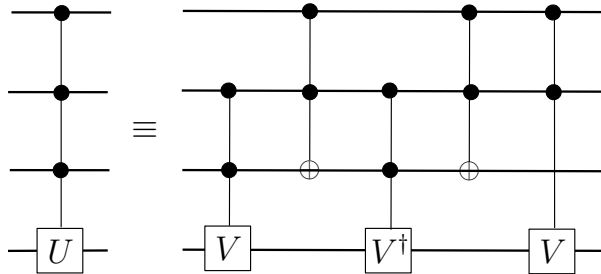
Example 4.5 (Multicontrolled U -gates). In this example we indicate how to obtain a q -algorithm for the q -procedure $C_{\{1, \dots, n\}, n+1}(U)$ defined by the relations

$$|j\rangle|j_{n+1}\rangle \mapsto \begin{cases} |\mathbf{1}_n\rangle|1 + j_{n+1}\rangle & \text{if } j = \mathbf{1}_n \\ |j\rangle|j_{n+1}\rangle & \text{otherwise} \end{cases}$$

If we take $V \in \mathbf{U}^{(1)}$ such that $U = V^2$, then the algorithm is based on the following recursive recipe:

- CONTROL $[\{1, \dots, n\}, n + 1, U]$
- CONTROL $[\{2, \dots, n\}, n + 1, V]$
- CONTROL $[\{1, \dots, n - 1\}, n, X]$
- CONTROL $[\{2, \dots, n\}, n + 1, V^\dagger]$
- CONTROL $[\{1, \dots, n - 1\}, n, X]$
- CONTROL $[\{1, \dots, n - 1\}, n + 1, V]$ ■

In other words, an n -controlled U -gate is reduced to five $(n - 1)$ -controlled U -gates. This is more easily grasped pictorially. Consider, for instance, the case $n = 3$:



If the first q -bit is $|0\rangle$, then the action on the third q -bit is $VV^\dagger = I_2$. If the second q -bit is $|0\rangle$, then the action on the third q -bit is I_2 . If the third q -bit is $|0\rangle$, and the first and second are $|1\rangle$, then the action on the third q -bit is $V^\dagger V = I_2$. Finally, if the three q -bits are $|1\rangle$, then the action on the third q -bit is $V^2 = U$.

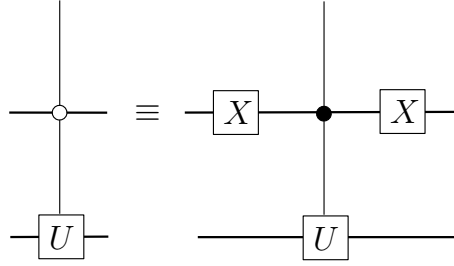
Example 4.6 (Action of $U \in S\mathbf{U}^{(1)}$ on a plane). Consider the plane $P = [|j\rangle, |k\rangle]$ spanned by two different basis vectors $|j\rangle$ and $|k\rangle$. Then we can let $U = [[a, b], [c, d]] \in S\mathbf{U}^{(1)}$ act on that plane in the obvious way: $U|j\rangle = a|j\rangle + b|k\rangle$ and $U|k\rangle = c|j\rangle + d|k\rangle$. Moreover, we can extend this action to $\mathbf{H}^{(n)}$ so that

$U|l\rangle = |l\rangle$ for all $l \neq j, k$. Since $|l\rangle$ is orthogonal to P , this action is a q -computation (we will write $U_{j,k}$ to denote it). Note, for example, that if $j = 1$ and $k = 2$, then the matrix of our q -computation is $U \oplus I_{2^{n-2}}$.

The aim of this example is to indicate how to get a q -algorithm for $U_{j,k}$. In fact, by the previous example, it will be enough to show how to resolve $U_{j,k}$ by means of simple and multicontrol U -gates. The simplest case is when $|j\rangle$ and $|k\rangle$ have the form

$$|j\rangle = |x\rangle|0\rangle|y\rangle, \quad |k\rangle = |x\rangle|1\rangle|y\rangle.$$

Indeed, in this case $U|j\rangle = a|j\rangle + b|k\rangle = |x\rangle(a|0\rangle + b|1\rangle)|y\rangle = |x\rangle(U|0\rangle)|y\rangle$, $U|k\rangle = |x\rangle(U|1\rangle)|y\rangle$ (similar computation), and therefore $U_{j,k}$ is a multicontrol U -gate, in the sense that if $|l\rangle = |x'\rangle|b\rangle|y'\rangle$, then $U_{j,k}|l\rangle = |l\rangle$ if $x \neq x'$ or $y \neq y'$ and otherwise it is equal to $|x\rangle(U|b\rangle)|y\rangle$. Note that if the controlling value of a bit is 0, then we can reduce it to the standard controlling value 1 and two X gates, as shown in the picture (the white circle is to indicate that the control value is 0):



If j and k differ in $r \geq 2$ two places, let $j' \in \mathbf{B}^n$ be such that j' differs in one position from j and in $r - 1$ positions from k . By induction we may assume that there is a q -algorithm to compute $U_{j',k}$, for the case $r = 1$ has already been established. Now a q -algorithm for $U_{j,k}$ is obtained on noticing that it coincides with $X_{j,j'}U_{j',k}X_{j,j'}$, where $X_{j,j'}$ is defined so that $X_{j,j'}|j\rangle = |j'\rangle$, $X_{j,j'}|j'\rangle = |j\rangle$ and $X_{j,j'}|l\rangle = |l\rangle$ if $l \neq j, j'$. Since $X_{j,j'}$ is a (form of) multicontrol-NOT, it can be computed by a q -algorithm, and thus so it can $U_{j,k}$.

Example 4.7 (Fourier Transform). The *Fourier transform* (FT) on $\mathbf{H}^{(n)}$ is the linear operator

$$F : \mathbf{H}^{(n)} \rightarrow \mathbf{H}^{(n)}, |j\rangle \mapsto \mathbf{f}_j = \rho^n \sum_k \xi^{jk} |k\rangle,$$

where $\xi = \xi_n = e^{i\frac{2\pi}{2^n}} = e^{i\frac{\pi}{2^{n-1}}}$.

Observe that $F \in \mathbf{U}^{(n)}$:

$$\langle \mathbf{f}_j | \mathbf{f}_{j'} \rangle = \frac{1}{2^n} \sum_k \xi^{(j'-j)k} = \delta_{jj'},$$

for, if $l \neq 0$,

$$\sum_{k=0}^{2^n-1} \xi^{lk} = \frac{(\xi^l)^{2^n} - 1}{(\xi^l - 1)} = 0.$$

Let us give an idea about how to produce a internal q -algorithm to obtain F . We have, with $\rho = 1/\sqrt{2}$,

$$\begin{aligned} F|j\rangle &= \rho^n \sum_{k=0}^{2^n-1} e^{\frac{2\pi ijk}{2^n}} |k\rangle \\ &= \rho^n \sum_{k_1, \dots, k_n \in \mathcal{B}} e^{2\pi ijk \left(\frac{k_1}{2^1} + \frac{k_2}{2^2} + \dots + \frac{k_n}{2^n} \right)} |k_1 \dots k_n\rangle \\ &= \rho^n \sum_{k_1, \dots, k_n \in \mathcal{B}} \bigotimes_{l=1}^n e^{\frac{2\pi ijk_l}{2^l}} |k_l\rangle \\ &= \rho^n \bigotimes_{l=1}^n \left(|0\rangle + e^{\frac{2\pi ij}{2^l}} |1\rangle \right). \end{aligned}$$

But

$$\frac{j}{2^l} = \frac{j_n}{2^l} + \frac{j_{n-1}}{2^{l-1}} + \dots + \frac{j_{n-(l-1)}}{2} + \left(j_l + \dots + j_1 2^{n-l-1} \right).$$

Since the part enclosed in parenthesis is an integer, the l -th tensor factor in the previous expression is equal to

$$|0\rangle + e^{i\pi \frac{j_n}{2^{l-1}}} \dots e^{i\pi j_{n-(l-1)}} |1\rangle$$

Therefore

$$\begin{aligned} F|j\rangle &= \rho^n \left(|0\rangle + e^{i\pi j_n} |1\rangle \right) \left(|0\rangle + e^{i\pi \frac{j_n}{2}} e^{i\pi j_{n-1}} |1\rangle \right) \dots \\ (*) \quad &\left(|0\rangle + e^{i\pi \frac{j_n}{2^{n-1}}} \dots e^{i\pi \frac{j_2}{2}} e^{i\pi j_1} |1\rangle \right). \end{aligned}$$

If we write this tensor product in reverse order, with one ρ for each factor,

$$\begin{aligned} &\rho \left(|0\rangle + e^{i\pi \frac{j_n}{2^{n-1}}} \dots e^{i\pi \frac{j_2}{2}} e^{i\pi j_1} |1\rangle \right) \rho \left(|0\rangle + e^{i\pi \frac{j_n}{2^{n-2}}} \dots e^{i\pi j_2} |1\rangle \right) \dots \\ &\dots \rho \left(|0\rangle + e^{i\pi j_n/2} e^{i\pi j_{n-1}} |1\rangle \right) \rho \left(|0\rangle + e^{i\pi j_n} |1\rangle \right), \end{aligned}$$

then for the l -th factor we have

$$\rho \left(|0\rangle + e^{i\pi \frac{j_n}{2^{n-l}}} \dots e^{i\pi \frac{j_{l+1}}{2}} e^{i\pi j_l} |1\rangle \right) = R_{n-l} \dots R_1 H |j_l\rangle$$

where R_s means, for the l -th bit, $C_{l+s,l}(S_{i\pi/2^s})$. So we have the following q -algorithm:

QFT

for $l \in \{1, \dots, n\}$ do

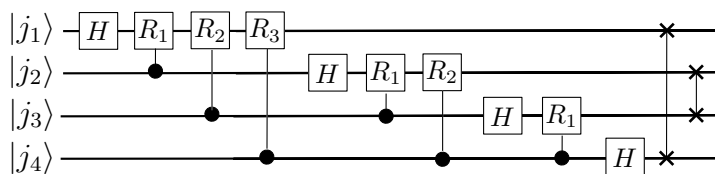
$R_l(H)$

for $s \in \{1, \dots, n-l\}$ do $C_{l+s,l}(S_{i\pi/2^s})$

for $l \in \{1, \dots, \lfloor n/2 \rfloor\}$ do SWAP $[l, n-l+1]$ ■

This shows that QFT computes F with complexity $O(n^2)$. The swaps at the end are meant to restore the original order.

Here is a diagram to illustrate the case $n = 4$.



Remark 4.8. Let us point out, for later reference, that the formula (*) can be written in the form

$$F|j\rangle = \rho^n (|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle)(|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0 \cdot j_1 j_1 \dots j_n} |1\rangle),$$

where, for binary digits b_1, b_2, \dots ,

$$0.b_1 b_2 \dots = \frac{b_1}{2} + \frac{b_2}{2^2} + \dots$$

The q -algorithms presented so far provide nice illustrations of the following general result:

Theorem 4.9 (Universality of the U and CNOT gates).

- 1) Any q -computation can be realized by an internal q -algorithm.
- 2) For any q -computation U there exists a restricted internal q -algorithm which approximates U to any wanted degree.

Proof. \triangleright **3** for 1) and \triangleright **4** for 2). □

5. DEUTSCH'S AND GROVER'S q -ALGORITHMS

In this section we present two archtypal q -algorithms. The first, due to Deutsch–Josza [5], takes a Boolean function which is known to be constant or balanced and decides *exactly* which of the two possibilities actually occurs in $O(n)$ steps. This is quite remarkable if we remember that the classical solution may require up to $2^{n-1} + 1$ steps (see below) and that q -algorithms are, in general, intrinsically probabilistic.

The second, due to Grover [6], searches for an item in an unordered list of length N in $O(\sqrt{N})$ steps. In this case the complexity gain is quite significant, as the classical complexity is $O(N)$, but the solution may be incorrect with a small probability. This fact need not be as bad as it could seem, because usually it is possible to recognize whether the returned item is the one we were searching for, with the idea that, if it is not, we can run the search again.

Deutsch's problem. Let $f : \mathbf{B}^n \rightarrow \mathbf{B}$ be a map, and assume we know that it is either *constant* or *balanced* (this means that the sets $f^{-1}(0)$ and $f^{-1}(1)$ have the same cardinal). Then the *Deutsch's problem* consists in deciding which of the two possibilities holds.

The classical solution is based on evaluating f on successive elements of \mathbf{B}^n . This process stops as soon as either we have found two different values, in which case f has to be balanced, or else the number of evaluations has exceeded 2^{n-1} , in which case f must be constant. Since the worse case requires $2^{n-1} + 1$ evaluations, the complexity of this procedure is exponential in n .

Deutsch's q -procedure. The solution to Deutsch's problem is provided by the following Deutsch–Jozsa q -procedure:

1. Initialize a q -computer of order $n + 1$ with $|\mathbf{u}_1\rangle = |0\rangle.^n \cdot |0\rangle|1\rangle$.
2. Using Example 4.2, obtain

$$|\mathbf{u}_2\rangle = H^{\otimes(n+1)}|\mathbf{u}_1\rangle = \rho^{n+1} \sum_{j=0}^{2^n-1} |j\rangle(|0\rangle - |1\rangle).$$

3. Let $U_{\bar{f}}$ be the q -computation corresponding to the classical (reversible) computation $\mathbf{B}^n \times \mathbf{B} \rightarrow \mathbf{B}^n \times \mathbf{B}$, $(x, b) \mapsto (x, b + f(x))$ and let $|\mathbf{u}_3\rangle = U_{\bar{f}}|\mathbf{u}_2\rangle$. Since

$$U(|j\rangle|b\rangle) = |j\rangle|b + f(j)\rangle$$

we clearly have

$$|\mathbf{u}_3\rangle = \rho^{n+1} \sum_{j=0}^{2^n-1} |j\rangle(|f(j)\rangle - |1 + f(j)\rangle).$$

Note that this can be written as

$$\rho^{n+1} \sum_{j=0}^{2^n-1} (-1)^{f(j)} |j\rangle(|0\rangle - |1\rangle) = \rho^{n+1} \sum_{j_r \in \mathbf{B}} (-1)^{f(j_1 \dots j_n)} |j_1 \dots j_n\rangle(|0\rangle - |1\rangle).$$

4. Compute $|\mathbf{u}_4\rangle = (H^{\otimes n} \otimes I_2) |\mathbf{u}_3\rangle$. Since

$$\begin{aligned} (H^{\otimes n} \otimes I_2) |j_1 \cdots j_n\rangle (|0\rangle - |1\rangle) &= (H|j_1\rangle) \cdots (H|j_n\rangle) (|0\rangle - |1\rangle) \\ &= \rho^n \prod_{r=1}^n (|0\rangle + (-1)^{j_r} |1\rangle) (|0\rangle - |1\rangle) \\ &= \rho^n \sum_{k_s \in \mathbf{B}} (-1)^{j \cdot k} |k_1 \cdots k_n\rangle (|0\rangle - |1\rangle), \end{aligned}$$

where $j \cdot k = j_1 k_1 + \cdots + j_n k_n$ is the scalar product of the binary vectors j and k , we find

$$|\mathbf{u}_4\rangle = \rho^{2n+1} \sum_{k_s \in \mathbf{B}} \sum_{j_r \in \mathbf{B}} (-1)^{j \cdot k + f(j_1 \cdots j_n)} |k_1 \cdots k_n\rangle (|0\rangle - |1\rangle),$$

which can be summarized as

$$\rho^{2n+1} \sum_{j,k} (-1)^{j \cdot k + f(j)} |k\rangle (|0\rangle - |1\rangle).$$

Let us look at the coefficient $a_k = \rho^{2n+1} \sum_j (-1)^{j \cdot k + f(j)}$ of $|k\rangle (|0\rangle - |1\rangle)$ in this expression. If f is constant, $a_k = \rho^{2n+1} (-1)^{f(0)} \sum_j (-1)^{j \cdot k}$, so that $a_0 = (-1)^{f(0)} \rho$ and $a_k = 0$ for $k \neq 0$. If f is balanced then $a_0 = \rho^{2n+1} \sum_j (-1)^{f(j)} = 0$, and clearly $a_k \neq 0$ for some $k \neq 0$. We can summarize these findings as follows:

$$|\mathbf{u}_4\rangle = \begin{cases} \rho |0\rangle (|0\rangle - |1\rangle) & \text{if } f \text{ is constant,} \\ \sum_{k \neq 0} a_k |k\rangle (|0\rangle - |1\rangle) & \text{if } f \text{ is balanced,} \end{cases}$$

5. The last step consists in measuring the first n q -bits. If f is constant, the result is 0 with certainty, and if f is balanced, then we obtain a non-zero integer. Hence, the algorithm decides *exactly* whether f is constant or not.

Deutsch's q -algorithm. Given a map $f : \mathbf{B}^n \rightarrow \mathbf{B}$ which is either constant or balanced, this q -algorithm returns 0 if and only if f is constant. We will work at order $n+1$ and we will let \tilde{f} denote the classical reversible computation defined by $(x, b) \mapsto (x, b + f(x))$, $x \in \mathbf{B}^n$, $b \in \mathbf{B}$.

DEUTSCH[f]

$$\begin{aligned}
 & \rightarrow |\mathbf{0}_n\rangle|0\rangle \\
 R_{n+1}(X) & \rightarrow |0\cdots 0\rangle|1\rangle \\
 \text{HADAMARD}[n] & \rightarrow \rho^{n+1} \sum_{j=0}^{2^n-1} |j\rangle(|0\rangle - |1\rangle) \\
 U_{\tilde{f}} & \rightarrow \rho^{n+1} \sum_{j=0}^{2^n-1} (-1)^{f(j)} |j\rangle(|0\rangle - |1\rangle) \\
 \text{HADAMARD}[n] & \rightarrow \rho^{2n+1} \sum_k \sum_j (-1)^{j\cdot k + f(j)} |k\rangle(|0\rangle - |1\rangle) \\
 & // \rho|\mathbf{0}_n\rangle(|0\rangle - |1\rangle) \text{ if } f \text{ is constant, and} \\
 & // \sum_{j \neq 0} a_j |j\rangle(|0\rangle - |1\rangle) \text{ if } f \text{ is balanced.} \\
 M_{\{1, \dots, n\}} & \rightarrow M \\
 & \text{if } M = 0 \text{ then Constant else Balanced } \blacksquare
 \end{aligned}$$

Grover's search

Suppose $\{j \rightarrow x_j \mid j = 0, \dots, N - 1\}$ is a database with $N = 2^n$ items. If we are to search for the j such that x_j satisfies some condition, like finding the position of a given number in a random list, in the worst case we will have to examine all the N items. In any case, to find a randomly chosen value x we will need, on the average, $N/2$ tests.

The remarkable discovery of Grover [6, 7] is that there is a q -algorithm with complexity $O(\sqrt{N/M})$ that finds an x satisfying the condition, where M is the number of solutions to the query.³

Grover's procedure. Let J_1 (J_0) be the subset of $\{0, 1, \dots, N - 1\}$ formed with the j such that x_j satisfies (does not satisfy) the condition in question. Consider the map $f : \mathbf{B}^n \rightarrow \mathbf{B}$ such that

$$f(j) = \begin{cases} 0 & \text{if } j \in J_0 \\ 1 & \text{if } j \in J_1 \end{cases} .$$

Define the unit q -vectors

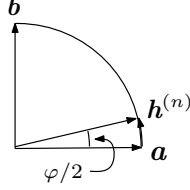
$$\mathbf{a} = \frac{1}{\sqrt{N - M}} \sum_{j \in J_0} |j\rangle \quad \text{and} \quad \mathbf{b} = \frac{1}{\sqrt{M}} \sum_{j \in J_1} |j\rangle .$$

³Grover's q -algorithm is probabilistic, in the sense that there is a small probability p that the outcome of a run does not satisfy the condition. As it is customary in such cases, running the algorithm some fixed number of times k (this does not change the complexity) will yield an answer that may be wrong with probability p^k , a value that usually is negligibly small already for small k .

The non-zero summands in \mathbf{b} (respectively \mathbf{a}) are the basis vectors corresponding to the solutions (non-solutions) of our query. Note also that

$$\mathbf{h}^{(n)} = \sqrt{\frac{N-M}{N}} \mathbf{a} + \sqrt{\frac{M}{N}} \mathbf{b} = \cos\left(\frac{\varphi}{2}\right) \mathbf{a} + \sin\left(\frac{\varphi}{2}\right) \mathbf{b},$$

where the last equality defines $\varphi \in (0, \pi)$ uniquely: $\varphi = 2 \arcsin(\sqrt{M/N})$.



Remark 5.1. Using the trigonometric formulae for the double angle we get that

$$\sin(\varphi) = \frac{2\sqrt{M}\sqrt{N-M}}{N}, \quad \cos(\varphi) = \frac{N-2M}{N}. \quad \square$$

To explain how Grover's procedure works, we need to introduce two q -computations of order n , which we denote G_f and K . The definition of G_f is as follows ($j \in \mathbf{B}^n$):

$$G_f(|j\rangle) = \begin{cases} -|j\rangle & \text{if } j \in J_1 \\ |j\rangle & \text{if } j \in J_0 \end{cases}$$

In other words, G_f is the reflexion with respect to the space spanned by the non-solutions. In particular, $G_f(\mathbf{a}) = \mathbf{a}$ and $G_f(\mathbf{b}) = -\mathbf{b}$. Therefore we also have

$$G_f(\mathbf{h}^{(n)}) = G_f\left(\cos\left(\frac{\varphi}{2}\right) \mathbf{a} + \sin\left(\frac{\varphi}{2}\right) \mathbf{b}\right) = \cos\left(\frac{\varphi}{2}\right) \mathbf{a} - \sin\left(\frac{\varphi}{2}\right) \mathbf{b}.$$

The q -computation K , which does not depend on f , is defined as

$$K(\mathbf{x}) = \sum_j (2x - x_j) |j\rangle,$$

where $x = \text{Av}(\mathbf{x}) = \frac{1}{N} \sum_j x_j$, the average of the amplitudes x_j of \mathbf{x} (we say that K is the *inversion with respect to the mean*). This linear map is indeed a q -computation, for it preserves the norm:

$$\begin{aligned} |K(\mathbf{x})|^2 &= \sum_j (2x - x_j)(2\bar{x} - \bar{x}_j) \\ &= 4Nx\bar{x} - 2\bar{x} \sum_j x_j - 2x \sum_j \bar{x}_j + \sum_j x_j \bar{x}_j \\ &= 4Nx\bar{x} - 2N\bar{x}x - 2Nx\bar{x} + |\mathbf{x}|^2 \\ &= |\mathbf{x}|^2. \end{aligned}$$

Now Grover's q -procedure can be described as follows:

1. Let $\mathbf{u}_0 = \mathbf{h}^{(n)} = \cos\left(\frac{\varphi}{2}\right)\mathbf{a} + \sin\left(\frac{\varphi}{2}\right)\mathbf{b}$.
2. For $j = 1, \dots, m = \left\lfloor \frac{\pi}{2\varphi} \right\rfloor$, define $\mathbf{u}_j = K(G_f(\mathbf{u}_{j-1}))$.
3. Return $M(\mathbf{u}_m)$.

The main reason why this procedure works is that *in the plane spanned by \mathbf{a} and \mathbf{b} the map KG_f is a rotation of amplitude φ* . Actually it is enough to show that

$$K\mathbf{a} = \cos(\varphi)\mathbf{a} + \sin(\varphi)\mathbf{b}$$

and

$$K(-\mathbf{b}) = -\sin(\varphi)\mathbf{a} + \cos(\varphi)\mathbf{b},$$

and this follows from a straightforward computation using the definition of K and the formulae in Remark 5.1 (\triangleright 5). In particular we have

$$\mathbf{u}_j = \mathbf{a} \cos \frac{2j+1}{2}\varphi + \mathbf{b} \sin \frac{2j+1}{2}\varphi.$$

This tells us that the optimal choice for the number m of iterations in step 2 is the least positive integer such that \mathbf{u}_m is closest to \mathbf{b} , and this clearly occurs when m is the nearest integer to

$$\left(\frac{\pi}{2} - \frac{\varphi}{2}\right)/\varphi = \frac{\pi}{2\varphi} - 1/2,$$

that is, when $m = \left\lfloor \frac{\pi}{2\varphi} \right\rfloor = \left\lfloor \frac{\pi}{4 \arcsin(\sqrt{M/N})} \right\rfloor$.⁴

Remark 5.2. Since $\arcsin(x) > x$ for $x \in (0, \frac{\pi}{2})$, we have

$$m \leq \frac{\pi}{4 \arcsin \sqrt{M/N}} \leq \frac{\pi}{4} \sqrt{N/M}.$$

Hence also $m \leq \left\lfloor \frac{\pi}{4} \sqrt{N/M} \right\rfloor$. Since $\frac{\pi}{4x} - \frac{\pi}{4 \arcsin(x)} < 1$ for all $x \in (0, 1)$, we also have $\left\lfloor \frac{\pi}{4} \sqrt{N/M} \right\rfloor \leq m + 1$. A more detailed study shows that when $x \rightarrow 0$ the intervals in which $\left\lfloor \frac{\pi}{4x} \right\rfloor = \left\lfloor \frac{\pi}{4 \arcsin(x)} \right\rfloor + 1$ become negligibly small compared to the intervals in which $\left\lfloor \frac{\pi}{4x} \right\rfloor = \left\lfloor \frac{\pi}{4 \arcsin(x)} \right\rfloor$. Thus if we iterate $\left\lfloor \frac{\pi}{4} \sqrt{N/M} \right\rfloor$ times the loop in step 2 of Grover's q -procedure, we would get the right number of rotations most of the time and otherwise we would go one step beyond, which in practice gives a q -vector that is almost as good as the previous one.

The probability of obtaining a right answer in a run of Grover's q -procedure is $p = \sin^2\left(\frac{2m+1}{2}\varphi\right)$, as $\sin\left(\frac{2m+1}{2}\varphi\right)\frac{1}{\sqrt{M}}$ is the amplitude in \mathbf{u}_m of any of the M solutions. Similarly, the probability of obtaining an erroneous answer is

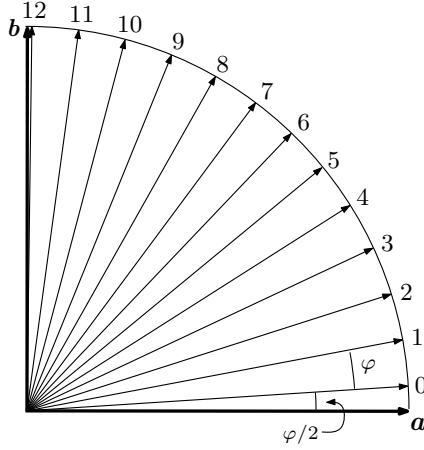
⁴We use that the nearest integer to $x - \frac{1}{2}$ is $\lfloor x \rfloor$.

$q = \cos^2(\frac{2m+1}{2}\varphi)$. Since the specification on m entails that $\frac{2m+1}{2}\varphi = \frac{\pi}{2} + \varepsilon$, with $|\varepsilon| \leq \varphi/2$, we see that

$$\begin{aligned} p &= \sin^2\left(\frac{\pi}{2} + \varepsilon\right) = \cos^2(\varepsilon) = \cos^2(|\varepsilon|) \\ &\geq \cos^2\left(\frac{\varphi}{2}\right) = \cos^2\left(\arcsin\left(\sqrt{M/N}\right)\right) = 1 - \frac{M}{N}. \end{aligned}$$

Hence also $q = 1 - p \leq M/N$.

Example 5.3. Let us illustrate the ideas so far with the case $n = 8$ and $M = 1$. We get $N = 256$, $\varphi = 7.166643^\circ$, $m = 12$. The slope of the vector \mathbf{u}_{12} is 89.583042° and the probability of success is $p = 0.999947$. Note that p is much closer to 1 than the lower bound $1 - M/N = 1 - 1/256 = 0.996094$. The probability of error is $q = 0.000053$, again much closer to 0 than the upper bound $M/N = 1/256 = 0.003906$.



Grover's algorithm. Given a map $f : \mathbf{B}^n \rightarrow \mathbf{B}$ for which we know that $M = |f^{-1}(1)| > 0$, this q -algorithm computes Grover's q -procedure for f . We will work at order $n+1$ and we will let \tilde{f} denote the classical reversible computation defined by $(x, b) \mapsto (x, b + f(x))$, $x \in \mathbf{B}^n$, $b \in \mathbf{B}$. As before, the corresponding q -computation will be denoted $U_{\tilde{f}}$. We will use the notations m and \mathbf{u}_j ($j = 0, 1, \dots, m$) from the forgoing discussion.

It is easy to phrase the sought q -algorithm $\text{GROVER}[f]$ in terms of q -algorithms $\text{GROVERG}[f]$ and GROVERK for G_f and K :

```

GROVER[f, m]
    → |0n⟩
    HADAMARD      → u0 = h(n)
    for j ∈ {1, ..., m} do
        GROVERK GROVERG[f] |uj-1⟩ → |uj⟩
    M(um)      → M ■
    
```

To describe GROVERG[f] we will also work at order $n + 1$. The last q -bit plays an ancillary role and initially it is supposed to be in the state $|1\rangle$. Since at the end it will recover this state, GROVERG[f] is revealed by looking at the final state of the q -bits other than the last.

```

GROVERG[f]
    → |x⟩|1⟩
    // Set x = x0 + x1, xi = ∑j∈Ji xj|j⟩, i = 0, 1.
    Rn+1(H) → ρ(|x0⟩|0⟩ + |x1⟩|0⟩ - |x0⟩|1⟩ - |x1⟩|1⟩)
    Uf̄      → ρ(|x0⟩|0⟩ + |x1⟩|1⟩ - |x0⟩|1⟩ - |x1⟩|0⟩)
            = (|x0⟩ - |x1⟩)(H|1⟩)
    Rn+1(H) → |Gfx⟩|1⟩ ■
    
```

```

GROVERK
    → |x⟩
    HADAMARD
    for l ∈ {1, ..., n} do
        Rl(X)
        //This loop acts as X⊗n
        C{2,...,n},1(Z)
        //Z to first q-bit controlled by all the others.
    for l ∈ {1, ..., n} do
        Rl(X)
        //X⊗n
    HADAMARD → |K(x)⟩ ■
    
```

The justification that this q -algorithm computes K is based on the following observations:

1) $K = 2P_{\mathbf{h}^{(n)}} - I_N$, where $P_{\mathbf{a}}$ denotes the orthogonal projector onto \mathbf{a} (for unit \mathbf{a} , $P_{\mathbf{a}}\mathbf{x} = \langle \mathbf{a} | \mathbf{x} \rangle \mathbf{a}$). Indeed, the claim follows immediately from the relation

$$P_{\mathbf{h}^{(n)}}\mathbf{x} = \langle \mathbf{h}^{(n)} | \mathbf{x} \rangle \mathbf{h}^{(n)} = \rho^{2n} \left(\sum x_j \right) \sum |j\rangle = \text{Av}(\mathbf{x}) \sum |j\rangle$$

and the definition of K .

2) $K = H^{\otimes n}(2P_{|\mathbf{0}_n\rangle} - I_N)H^{\otimes n}$. This is a direct consequence of the formula $UP_{\mathbf{a}}U^{-1} = P_{U\mathbf{a}}$, where U is a arbitrary q -computation and \mathbf{a} any q -vector, and the preceding formula. Note that if we apply $UP_{\mathbf{a}}U^{-1}$ to $U\mathbf{x}$ we obtain $U\mathbf{a}$ if $\mathbf{x} = \mathbf{a}$ and 0 if \mathbf{x} is orthogonal to \mathbf{a} .

3) $I_N - 2P_{|\mathbf{0}_n\rangle} = X^{\otimes n}C_{\{2,\dots,n\},1}(Z)X^{\otimes n}$. Note that $I_N - 2P_{|\mathbf{0}_n\rangle}$ changes the sign of $|\mathbf{0}_n\rangle$ and is the identity on all $|j\rangle$ with $j \neq \mathbf{0}_n$. In relation to the right hand side of the formula, observe that $C_{\{2,\dots,n\},1}(Z)$, and hence the whole composition, will do nothing on $|j\rangle$ if not all j_2, \dots, j_n are 0. If $j_2 = \dots = j_n = 0$, then $C_{\{2,\dots,n\},1}(Z)$ applies Z to $|\bar{j}_1\rangle$, and, by the definition of Z ($Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$), this action does nothing if $j_1 = 1$ and changes its sign if $j_1 = 0$.

4) The analysis of Grover's q -algorithm has to be completed with a q -algorithm for $C_{\{2,\dots,n\},1}(Z)$. But this can be obtained as indicated in the Example 4.5.

6. PHASE ESTIMATION

Let U be a q -computation of order n , and let $\mathbf{u} \in \mathbf{H}^{(n)}$ be an eigenvector of U . The corresponding eigenvalue can be written in the form $e^{2\pi i\varphi}$, with $\varphi \in [0, 1)$. Assuming that U and \mathbf{u} are known, then the *phase estimation problem* consists in obtaining r bits $\varphi_1, \dots, \varphi_r$, for a given r , of the binary expansion $0.\varphi_1\varphi_2\dots$ of φ .

The aim of this section is to phrase and analyze the interesting q -algorithm discovered by Kitaev [11] to solve this problem.

Since we need some ancillary q -bits, say m , we will work in $\mathbf{H}^{(m)} \times \mathbf{H}^{(n)}$. The algorithm assumes that we can initialize $\mathbf{H}^{(n)}$ with the q -vector \mathbf{u} and also that we are able to perform the 'controlled' q -computations $C_{m-l+1}(U^{2^{l-1}})$, for $l = 1, \dots, n$, defined on $\mathbf{H}^{(m)} \times \mathbf{H}^{(n)}$ as follows:

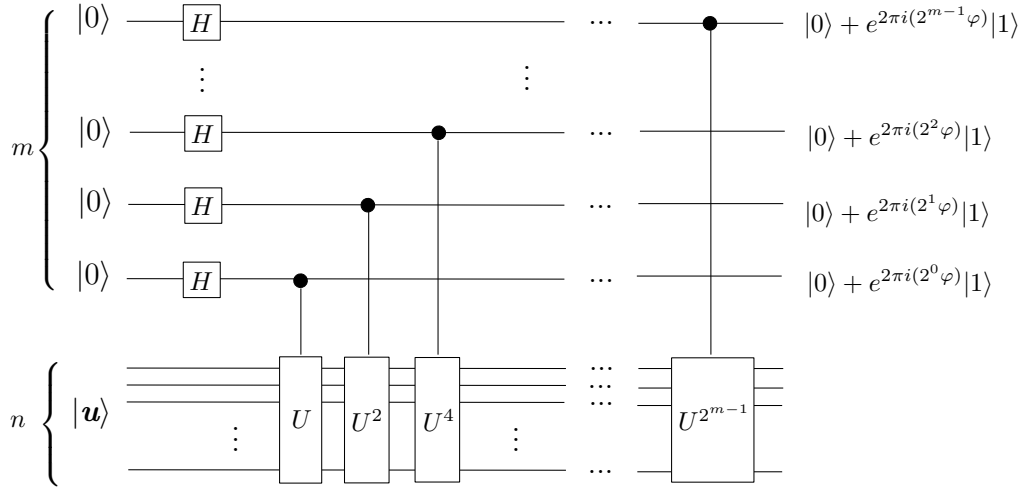
$$C_{m-l+1}(U^{2^{l-1}})(|\varphi_1 \dots \varphi_m\rangle |\mathbf{u}\rangle) = \begin{cases} |\varphi_1 \dots \varphi_m\rangle |\mathbf{u}\rangle & \text{if } \varphi_{m-l+1} = 0 \\ |\varphi_1 \dots \varphi_m\rangle (U^{2^{l-1}}|\mathbf{u}\rangle) & \text{if } \varphi_{m-l+1} = 1 \end{cases}$$

Kitaev's q -algorithm

KITAEV[U, \mathbf{u}]

0. $\rightarrow |\mathbf{0}_m\rangle|\mathbf{u}\rangle$
1. HADAMARD[m] $\rightarrow |\mathbf{h}^{(m)}\rangle|\mathbf{u}\rangle$
2. for $l \in 1..m$ do
 $C_{m-l+1}(U^{2^{l-1}})$
3. QFT † [m]
4. $M_{\{1,\dots,m\}}$

We will analyze this algorithm in two steps, denoted A and B below. In the first we will assume that $\varphi = 0.\varphi_1 \dots \varphi_m$ and in the second we will look at the general case. The following diagram illustrates the steps 0-2.



A. The action of $U^{2^{l-1}}$ only changes $|\mathbf{u}\rangle$ by a factor, either 1 or $e^{2\pi i 2^{l-1}\varphi}$ depending on whether the controlling bit is $|0\rangle$ or $|1\rangle$. Now this factor may be moved next to the controlling bit and therefore the state at the end of the loop 2 can be written in the form

$$(1) \quad \rho^m \left(|0\rangle + e^{2\pi i 2^{m-1}\varphi} |1\rangle \right) \left(|0\rangle + e^{2\pi i 2^{m-2}\varphi} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i 2^0\varphi} |1\rangle \right).$$

This, in the notation of binary expansions, takes the form

$$(2) \quad \rho^m \left(|0\rangle + e^{2\pi i 0.\varphi_m} |1\rangle \right) \left(|0\rangle + e^{2\pi i 0.\varphi_{m-1}\varphi_m} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i 0.\varphi_1 \dots \varphi_m} |1\rangle \right),$$

as $e^{2\pi i k} = 1$ for any integer k . But by the Remark 4.8, this expression is equal to $F|\varphi\rangle$, which is computed by the q -algorithm QFT (Example 4.7). Thus it is clear that we recover the state $|\varphi\rangle|\mathbf{u}\rangle$ by applying $F^\dagger \otimes I_{2^n}$, where F^\dagger is the inverse of F . We have denoted QFT † [m] the q -algorithm for F^\dagger that is obtained

by carrying out QFT in reverse order. Thus KITAEV supplies φ exactly in the case where φ can be expressed with m bits.

B. The reasoning is somewhat more involved when φ cannot be expressed using m bits. In this case, $F^\dagger \otimes I_{2^n}$ does not give the q -vector $|\varphi\rangle|\mathbf{u}\rangle$, but a superposition of the form $\sum a_l|l\rangle|\mathbf{u}\rangle$. As we will show below, this difficulty can be overcome in order to obtain the first r bits of φ provided $r \leq m$.

By expanding the product in the formula (2), we see that it can be written in the form

$$\rho^m \sum_{k=0}^{2^m-1} e^{2\pi i \varphi k} |k\rangle|\mathbf{u}\rangle.$$

The result in step 3 is

$$\begin{aligned} \rho^m \sum_{k=0}^{2^m-1} e^{2\pi i \varphi k} (F^\dagger|k\rangle)|\mathbf{u}\rangle &= \rho^{2m} \sum_{k=0}^{2^m-1} e^{2\pi i \varphi k} \sum_{l=0}^{2^m-1} e^{-\frac{2\pi i k l}{2^m}} |l\rangle|\mathbf{u}\rangle \\ &= \rho^{2m} \sum_{l=0}^{2^m-1} \left(\sum_{k=0}^{2^m-1} e^{2\pi i (\varphi - l/2^m) k} \right) |l\rangle|\mathbf{u}\rangle \\ &= \rho^{2m} \sum_{l=0}^{2^m-1} \frac{1 - e^{2\pi i (\varphi - l/2^m) 2^m}}{1 - e^{2\pi i (\varphi - l/2^m)}} |l\rangle|\mathbf{u}\rangle \end{aligned}$$

Finally the result of step 4, the measurement of the first m bits, is also clear: it will be an m -bit integer l drawn with probability⁵

$$(*) \quad p_l = \rho^{4m} \left| \frac{1 - e^{2\pi i (\varphi - l/2^m) 2^m}}{1 - e^{2\pi i (\varphi - l/2^m)}} \right|^2 = \rho^{4m} \frac{\sin^2 \pi (\varphi - l/2^m) 2^m}{\sin^2 \pi (\varphi - l/2^m)}$$

With this distribution law we can now estimate what are the chances that the first r bits of l ($0 < r \leq m$) agree with $f = \varphi_1 \cdots \varphi_r$. Indeed, using the probabilities p_l one can show (\triangleright **6**) that

$$(**) \quad p(|2^m \varphi - l| > 2^{m-r}) \leq \frac{1}{2(2^{m-r} - 2)}.$$

Therefore we can guarantee that r bits are correct with probability $1 - \varepsilon$ if $\frac{1}{2(2^{m-r} - 2)} \leq \varepsilon$, a relation that is equivalent to

$$m \geq r + \log_2 \left(2 + \frac{1}{2\varepsilon} \right).$$

⁵We use the formula $|1 - e^{i\alpha}|^2 = 4 \sin^2(\alpha/2)$, which is a consequence of $|1 - e^{i\alpha}|^2 = (1 - e^{i\alpha})(1 - e^{-i\alpha}) = 2 - (e^{i\alpha} + e^{-i\alpha}) = 2(1 - \cos \alpha)$.

7. MODULAR ORDER OF AN INTEGER

The object of this section is a presentation of Shor's q -algorithm for finding $\text{ord}_N(a)$, the order of a positive integer a modulo a positive integer N , provided $(a, N) = 1$. By definition, $r = \text{ord}_N(a)$ is the least positive integer such that $a^r \equiv 1 \pmod N$ or, in other words, the order of a seen as an element of the group \mathbb{Z}_N^* .

From a classical point of view, finding $\text{ord}_N(a)$ is related to the search of the divisors of $\phi(N)$ (where ϕ denotes the classical Euler's totient function), which has exponential complexity in terms of $n = \log_2(N)$ (see [1] for details). By contrast, Shor's q -algorithm produces a probabilistic solution which is polynomial in n .

First let us fix some notations. Set $r = \text{ord}_N(a)$ and $n = \lceil \log_2(N) \rceil$. Next define the q -computation $U_a = U_{a,N}$ of order n by the relation

$$U_a|j\rangle = \begin{cases} |aj \pmod N\rangle & \text{if } j < N \\ |j\rangle & \text{if } N \leq j < 2^n. \end{cases}$$

It is indeed a q -computation, as the map $\mathbb{Z}_N \rightarrow \mathbb{Z}_N$ such that $j \mapsto aj \pmod N$ is bijective (a permutation map). The inverse q -computation is $U_{a^{-1},N}$. Finally define, for every $s \in \{0, \dots, r-1\}$, the q -vector of order n

$$\mathbf{u}_s = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{s}{r}} |a^j \pmod N\rangle.$$

Applying the operator U_a to \mathbf{u}_s we get

$$U_a \mathbf{u}_s = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{s}{r}} |a^{j+1} \pmod N\rangle = e^{2\pi i \frac{s}{r}} \mathbf{u}_s,$$

which means that \mathbf{u}_s is an eigenvector of $U_{a,N}$ with eigenvalue $e^{2\pi i \frac{s}{r}}$.

At this point it would seem natural to apply Kitaev's q -algorithm to estimate the phase s/r of $e^{2\pi i \frac{s}{r}}$, with the idea that the information gained in this way could give us precious information about r . However this does not work, since the eigenvector \mathbf{u}_s would be known only if r were already known.

Fortunately this can be circumvented with the observation that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\mathbf{u}_s\rangle = |\mathbf{1}_n\rangle.$$

Indeed, if in Kitaev's q -algorithm we set $m = 2n + 1 + \lceil 2 + \frac{1}{2\varepsilon} \rceil$ and we let the initial state be $|\mathbf{0}_m\rangle|\mathbf{1}_n\rangle$, then, with probability $(1-\varepsilon)/r$, we will get an estimate

$\tilde{\varphi} \approx s/r$ with $2n + 1$ correct bits. Now we have that

$$\left| \frac{s}{r} - \tilde{\varphi} \right| \leq \frac{1}{2^{2n+1}} \leq \frac{1}{2r^2}$$

and hence, letting $s/r = s'/r'$ with $(s', r') = 1$, the inequality

$$\left| \frac{s'}{r'} - \tilde{\varphi} \right| \leq \frac{1}{2r'^2}$$

also holds. By a well known result in continued fractions (see [8]), s'/r' must be a convergent of $\tilde{\varphi}$. As $\tilde{\varphi}$ is a rational number, its set of convergents is finite and can be computed by the continued fraction algorithm. Summarizing, the choice of m in the phase estimation procedure assures that, with a probability of $1 - \varepsilon$, there exists a convergent $\tilde{\varphi}$ such that its denominator is either r if $(s, r) = 1$ or a divisor of r if $(s, r) \neq 1$.

If $(s, r) = 1$ then r is the order of a . This fact can be checked directly computing $a^{r_n} \bmod N$ where s_n/r_n is a convergent of $\tilde{\varphi}$. If $(s, r) \neq 1$, then $a^r \bmod N$ is not equal to 1, and we need to repeat the phase estimation algorithm in order to get an estimation such that $(s, r) = 1$. Using the prime number theorem (see [1]), one can show that repeating the algorithm $O(n)$ times, with high probability we get an estimation $\tilde{\varphi}$ with a convergent s/r such that $(s, r) = 1$ (\triangleright 7).

The number of steps of the whole q -algorithm is $O(n^4)$: the more complex step is associated to the continued fraction algorithm, with complexity $O(n^3)$, which needs to be repeated $O(n)$ times in order to assure, with high probability, a convergent s/r such that $(s, r) = 1$.

With further improvements of these ideas (see [15]) the complexity can be reduced to $O(n^3)$.

Shor's order-finding

Let $1 < a < N$ be positive integers such that $(a, N) = 1$ and $\varepsilon > 0$ a (small) real number. The algorithm described below finds $r = \text{ord}_N(a)$ with probability $1 - \varepsilon$ with an average number of iterations which is $O(n)$. The total complexity is $O(n^4)$. The algorithm `ContFrac` returns, given a rational number, the list of the denominators of its convergents (\triangleright 8).

SHOR-ORDER[a, N, ε]

$$n = \lceil \log_2(N) \rceil, m = 2n + 1 + \log_2 \left(2 + \frac{1}{2\varepsilon} \right)$$

//Working q -space: $\mathbf{H}^{(m)} \otimes \mathbf{H}^{(n)}$

- | | | |
|---------------------------------------|--|--|
| 0. | | $\rightarrow \mathbf{0}_m\rangle \mathbf{0}_n\rangle$ |
| 1. HADAMARD[m] | | $\rightarrow \rho^m \sum_{j=0}^{2^m-1} j\rangle \mathbf{0}_n\rangle$ |
| 2. $U_{a,N}$ | | $\rightarrow \frac{\rho^m}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^m-1} e^{2\pi i j \frac{s}{r}} j\rangle \mathbf{u}_s\rangle$ |
| 3. QFT † [m] | | $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left \widetilde{s/r} \right\rangle \mathbf{u}_s\rangle$ |
| 4. $M = M_{\{1, \dots, n\}}$ | | $\rightarrow \widetilde{s/r}$ |
| 5. ContFrac | | $\rightarrow D$ |
| 6. for $r' \in D$ do | | |
| if $a^{r'} \bmod N = 1$, return r' | | |
| 7. return Not-successful | | |
| | | // $r' r$, and $r' = r$ in $O(n)$ iterations ■ |

Since the condition $r' = r$ is met in $O(n)$ iterations, we will get the correct order r with an average time $O(n^4)$. This is the algorithm we need in the next Section and will be denoted SHOR-ORDER(a, N).

8. SHOR'S FACTORING q -ALGORITHM

A fundamental problem in computational number theory is the search of proper divisors of a big positive integer N . The difficulty of this problem (by classical algorithms) provides a basis for efficient criptographic algorithms [18]. Amazingly, the quantum framework permits a solution whose complexity is polynomial in $n = \log_2(N)$. The key idea is to reduce, by a well known procedure, the factoring problem to and order-finding problem and then use the q -algorithm SHOR-ORDER studied in Section 7 to solve the latter.

Thus we first review how to go from factoring to order-finding and then we will formulate the q -algorithm SHOR-FACTOR by calling SHOR-ORDER.

From order-finding to factoring. Let N be a positive integer. Since there are efficient classical algorithms to decide whether N is a primer power p^r ,⁶ henceforth we will assume that N is not a prime power. It is also harmless to assume that N is odd. It will be enough to find a proper divisor d of N , for then $N = d \cdot (N/d)$ and we can proceed recursively with the factors d and N/d .

⁶If $N = m^r$, $m > 1$, then $r \leq \log_2(N)$. For each such r , $r > 1$, let $m = \lfloor N^{1/r} \rfloor$ and check whether $m^r = N$. If the equality holds, N is an exact power and factoring N is reduced to factoring m . Otherwise N is not an exact power and hence, in particular, not a prime power.

Now the main observation is that we can obtain a proper factor of N if we are able to produce an integer $x \in \{2, \dots, N-1\}$ such that

- (1) $(x, N) = 1$;
- (2) $r = \text{ord}_N(x)$ is even.
- (3) $x^{\frac{r}{2}} + 1$ is not divisible by N .

Indeed, since by definition r is the least positive integer such that $x^r = 1 \pmod N$ (the condition 1 implies that this number exists), we see that $x^r - 1 = (x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1)$ is divisible by N . Since neither $x^{\frac{r}{2}} - 1$ nor $x^{\frac{r}{2}} + 1$ is divisible by N , it follows that a prime factor of N must divide either $x^{\frac{r}{2}} - 1$ or $x^{\frac{r}{2}} + 1$. Therefore either $\gcd(x^{\frac{r}{2}} - 1, N)$ or $\gcd(x^{\frac{r}{2}} + 1, N)$ is a proper divisor of N .

So we are led to the question of finding an x satisfying the three conditions above. As we will see, this can be accomplished, with good chances, by picking x at random in $\{2, \dots, N-1\}$. Indeed, iff $(x, N) > 1$, then $d = (x, N)$ is a proper divisor of N and we are done. So we may assume that $(x, N) = 1$, and hence that $r = \text{ord}_N(x)$ exists. How likely are we that r is even and $x^{\frac{r}{2}} + 1$ is not divisible by N ? The following proposition provides the answer we need.

Proposition 8.1. Let N be a positive integer with $m \geq 2$ distinct prime factors. Then the density of the set

$$\{x \in \mathbb{Z}_N^* \mid r = \text{ord}_N(x) \text{ is even and } x^{\frac{r}{2}} + 1 \text{ is not divisible by } N\}$$

in \mathbb{Z}_N^* is $\geq 1 - \frac{1}{2^{m-1}}$.

Proof. See [▷ 9](#). □

Example 8.2. Consider the number $N = 904279$, which is odd and not a prime power. We pick a number x at random in $\{1, \dots, N-1\}$ (say as the value of `random(N)`). In our case we got $x = 743579$. We check that $(x, N) = 1$ (`gcd(x,N)→1`), so $r = \text{ord}_N(x)$ exists. We get $r = 150396$, as the value of `order(x,N)`. This is even and $(x^{\frac{r}{2}} - 1, N)$ gives the (prime) divisor 907 and $(x^{\frac{r}{2}} + 1, N)$ gives the (prime) divisor 997. Finally we check that $N = 907 \cdot 997$.

Shor's q -algorithm for factoring integers. As explained in the preceding section, we assume that N is an odd positive integer which is not a prime power.

SHOR-FACTOR[N]

- x, r, d
1. random(N) $\rightarrow x$
 2. if $d = (x, N) > 1$, return d
 3. SHOR-ORDER(x, N) $\rightarrow r$
 4. if $r \equiv 1 \pmod{2}$, go to 1.
 5. if $d = \left(x^{\frac{r}{2}} - 1, N\right) > 1$ return d
 6. if $x^{\frac{r}{2}} + 1 \pmod{N} = 0$, go to 1.
 7. return $d = \left(x^{\frac{r}{2}} + 1, N\right)$ ■

The complexity of SHOR-FACTOR is determined by step 3, and so its average cost is $O(n^4)$, $n = \log_2(N)$.

A more detailed analysis (\triangleright 10) shows that the average number of go to in steps 4 and 6 is $O(1)$.

9. PHYSICAL INTERPRETATIONS

In this section we will present how q -computations can be interpreted in (axiomatic) physical terms. For a deeper understanding of the physics involved, see for example [20, 10, 21].

1. Quantum states. A quantum system Σ is characterized by a *complex vector space* E endowed with a Hermitian scalar product $\langle \mathbf{x} | \mathbf{y} \rangle$ (i.e., linear in \mathbf{y} and conjugate-linear in \mathbf{x}). For the purposes of quantum computation we may also assume that E has finite dimension.

The non-zero vectors $\mathbf{x} \in E$ represent (pure) *states* of Σ , and two non-zero vectors $\mathbf{x}, \mathbf{y} \in E$ represent the same state if and only if there exists $\xi \in \mathbb{C}$ such that $\mathbf{y} = \xi \mathbf{x}$.

In particular, any state can be represented by a unitary vector \mathbf{u} (determined up to a phase factor $e^{i\alpha}$). Thus the state space of Σ is the *projective space* associated to E (it is usually denoted $\mathbf{P}E$).

Following Dirac's notations, we will write $|\mathbf{u}\rangle$ to denote the state corresponding to \mathbf{u} (in projective geometry it is denoted $[\mathbf{u}]$). If $\mathbf{v} \in E$ is arbitrary, but non-zero, we have $|\mathbf{v}\rangle = |\hat{\mathbf{v}}\rangle$, where $\hat{\mathbf{v}}$ is the unit vector $\mathbf{v}/|\mathbf{v}|$.

Quantum superposition. Given two states $|\mathbf{u}\rangle$ and $|\mathbf{u}'\rangle$ and complex numbers $a = \alpha + \beta i$ and $a' = \alpha' + \beta' i$, we can form the state $|a\mathbf{u} + a'\mathbf{u}'\rangle$. Such states are said to be a *quantum superposition* of the states $|\mathbf{u}\rangle$ and $|\mathbf{u}'\rangle$ and are often

denoted, by abuse of notation, as $a|\mathbf{u}\rangle + a'|\mathbf{u}'\rangle$. In geometrical terms, they are the points on the line determined by $|\mathbf{u}\rangle$ and $|\mathbf{u}'\rangle$.

2. Observables. An *observable* of Σ is a linear map $A : E \rightarrow E$ such that

$$\langle A\mathbf{x}|\mathbf{y}\rangle = \langle \mathbf{x}|A\mathbf{y}\rangle.$$

If we express A with respect to an orthonormal basis, it is easy to check that this condition is equivalent to $A = A^\dagger$. In other words,

$$\text{observable} \equiv \text{self-adjoint operator}$$

If a_1, \dots, a_r are the distinct eigenvalues of A , then $a_1, \dots, a_r \in \mathbb{R}$ and

$$(3) \quad A = \sum_j a_j P_j,$$

where $P_j : E \rightarrow E_j$ is the orthogonal projection from E onto the space of eigenvectors of A with eigenvalue a_j , namely

$$E_j = \{\mathbf{x} \in E \mid A\mathbf{x} = a_j\mathbf{x}\}.$$

The result of an *observation* or *measure* of A when Σ is in the state $|\mathbf{u}\rangle$ is one of the eigenvalues a_j , with probability

$$p_j = |P_j\mathbf{u}|^2 = \langle \mathbf{u}|P_j\mathbf{u}\rangle,$$

and also that Σ is reset to the state $P_j\mathbf{u}$ (or, more precisely, to $|P_j\mathbf{u}\rangle$). Notice that $\mathbf{u} = \sum_j P_j\mathbf{u}$ and hence $1 = |\mathbf{u}|^2 = \sum_j |P_j\mathbf{u}|^2 = \sum_j p_j$, as the E_j are pairwise orthogonal. Note also that $\langle \mathbf{u}|P_j\mathbf{u}\rangle - |P_j\mathbf{u}|^2 = \langle \mathbf{u}|P_j\mathbf{u}\rangle - \langle P_j\mathbf{u}|P_j\mathbf{u}\rangle = \langle \mathbf{u} - P_j\mathbf{u}|P_j\mathbf{u}\rangle = 0$, since $\mathbf{u} - P_j\mathbf{u}$ is orthogonal to E_j by definition of P_j .

In particular, if $\mathbf{u} \in E_j$, then the observation yields a_j with certainty and Σ remains in the state $|\mathbf{u}\rangle$.

Example 9.1. If F is a linear subspace of E , the orthogonal projection $P_F : E \rightarrow F$ is an observable with eigenvalues 1 and 0: $E_1 = F$ and $E_0 = F^\perp$. The observables of this form are called *propositions* or *eventualities*. Note that the probability of measuring 1, if the system is in the state $\mathbf{u} \in E$, is $|P_F(\mathbf{u})|^2$.

The formula (3) shows that any observable is a linear combination with real coefficients (the values a_j) of eventualities (the projector P_{a_j} is the eventuality corresponding to the space $E_j = E_{a_j}$). Seen in this light, an observable can be identified with a list of pairs $\{(a_1, E_1), \dots, (a_r, E_r)\}$, where $a_1, \dots, a_r \in \mathbb{R}$ (the possible values of the observable) and $E_1, \dots, E_r \subseteq E$ are linear subspaces of E such that $E = E_1 \oplus \dots \oplus E_r$ and $E_j \perp E_k$ for $j \neq k$. The non-zero vectors of E_j represent states for which the measured value is a_j with certainty and the non-zero vectors in the orthogonal $E_j^\perp = \bigoplus_{k \neq j} E_k$ represent states for which the measured value is $\neq a_j$ with certainty. Note that the probability of obtaining a_j agrees with the probability of observing 1 for the eventuality defined by E_j .

3. Unitary dynamics. If Σ lies in a non-reactive environment (i.e., the environment is not affected by Σ) in the time interval $[0, t]$, there exists a unitary operator

$$U : E \rightarrow E$$

such that

$$\mathbf{b} = U\mathbf{a}$$

represents the state of Σ at time t if $\mathbf{a} \in E$ represents the state of Σ at time 0.

Example 9.2. If H is an observable, the operator

$$U = e^{iHt}$$

is unitary, as $U^\dagger = e^{-iH^\dagger t} = e^{-iHt} = U^{-1}$. It is customary to say that $U = e^{iHt}$ is the time evolution defined by the *Hamiltonian* H .

4. Entanglement. If Σ' is a second quantum system with associated space E' , then the associated space of the composite system $\Sigma + \Sigma'$ is $E \otimes E'$, with the natural hermitian bracket defined by the relation

$$\langle \mathbf{x} \otimes \mathbf{x}' | \mathbf{y} \otimes \mathbf{y}' \rangle = \langle \mathbf{x} | \mathbf{y} \rangle \cdot \langle \mathbf{x}' | \mathbf{y}' \rangle.$$

If $\mathbf{u} \in E$ and $\mathbf{u}' \in E'$ are unit vectors, the state $|\mathbf{u} \otimes \mathbf{u}'\rangle$ is also denoted $|\mathbf{u}\rangle|\mathbf{u}'\rangle$, or $|\mathbf{u}\mathbf{u}'\rangle$, and it is thought as the state of the composite system corresponding to the state $|\mathbf{u}\rangle$ of Σ and $|\mathbf{u}'\rangle$ of Σ' . We will say that they are *composite states*. Note, however, that in general the states of the composite system are not composite states. A simple example is given by $|\mathbf{u}_1 \otimes \mathbf{u}'_1\rangle + |\mathbf{u}_2 \otimes \mathbf{u}'_2\rangle$ if $\mathbf{u}_1, \mathbf{u}_2 \in E$ (respectively $\mathbf{u}'_1, \mathbf{u}'_2 \in E'$) are orthogonal unit vectors (\triangleright 11). Since the non-composite states are superposition of composite states (any vector in $E \otimes E'$ is a sum of composite vectors), the non-composite states are called *entangled states*.

Example 9.3 (q-bits). The states of a spin- $\frac{1}{2}$ particle (system $\Sigma^{(1)}$) can be thought as points lying on the sphere S^2 of radius 1 (in suitable units).

The complex space associated to this system according to axiom 1 is \mathbb{C}^2 (*spinor space*).

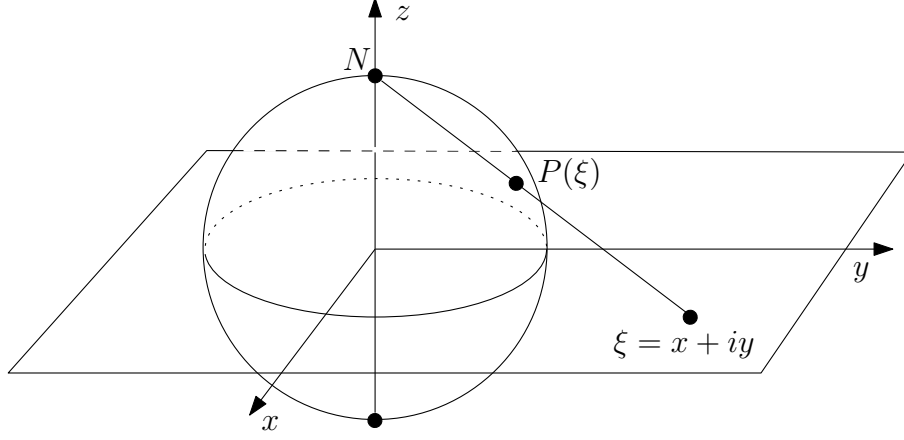
This fact can be argued as follows.

- Identify $\xi = x + iy \in \mathbb{C}$ with the point $(x, y, 0) \in \mathbb{R}^3$ and consider the point $P = P(\xi)$ of

$$S^2 = \{(x, y, z) \in \mathbb{R}^3 | x^2 + y^2 + z^2 = 1\}$$

obtained by stereographic projection from $N = (0, 0, 1)$:

$$P = \left(\frac{2x}{x^2 + y^2 + 1}, \frac{2y}{x^2 + y^2 + 1}, \frac{x^2 + y^2 - 1}{x^2 + y^2 + 1} \right).$$



Setting $P(\infty) = N$, we get a bijection between $\widehat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ and S^2 . The inverse map is given by

$$(x, y, z) \mapsto \frac{x}{1-z} + i \frac{y}{1-z}, \text{ for } z < 1,$$

and $N = [0, 0, 1] \mapsto \infty$, for $z = 1$.

- On the other hand we also have

$$\widehat{\mathbb{C}} \simeq \mathbf{P}\mathbb{C}^2 = \mathbf{P}_{\mathbb{C}}^1,$$

for any element $[\xi_0, \xi_1] \in \mathbb{C}^2$ is proportional to a unique vector of the form $[1, \xi]$ when $\xi_0 \neq 0$, and to $[0, 1]$ if $\xi_0 = 0$. Thus we have a bijective map

$$\widehat{\mathbb{C}} \rightarrow \mathbf{P}_{\mathbb{C}}^1, \quad \xi \mapsto [1, \xi], \quad \infty \mapsto [0, 1].$$

The inverse map is given by

$$[\xi_0, \xi_1] \mapsto \begin{cases} \xi = \xi_1/\xi_0 & \text{if } \xi_0 \neq 0 \\ \infty & \text{if } \xi_0 = 0 \end{cases}$$

These considerations indicate that we may take \mathbb{C}^2 as the space associated to $\Sigma^{(1)}$.

Note. The sphere S^2 , with the structure of $\mathbf{P}_{\mathbb{C}}^1$, is called the *Riemann sphere*. It is the simplest compact Riemann surface. In quantum computation references, it is often called the *Bloch sphere* or even the *Poincaré–Bloch sphere*.

Remark 9.4. Let $P = (x, y, z)$ be a point of S^2 and define φ as the argument of $x + iy$ and θ as the angle between OP and ON , where O is the center of the sphere. The relation between the spherical coordinates (φ, θ) and the cartesian coordinates (x, y, z) is given by the formulas

$$(4) \quad x = \sin \theta \cos \varphi, \quad y = \sin \theta \sin \varphi, \quad z = \cos \theta.$$

The point in $\widehat{\mathbb{C}}$ corresponding to $P(x, y, z)$ is

$$\xi = \frac{x}{1-z} + i \frac{y}{1-z} = \frac{\sin \theta \cos \varphi}{1 - \cos \theta} + i \frac{\sin \theta \sin \varphi}{1 - \cos \theta} = \frac{\sin \theta}{1 - \cos \theta} e^{i\varphi} = e^{i\varphi} \cot \frac{\theta}{2}.$$

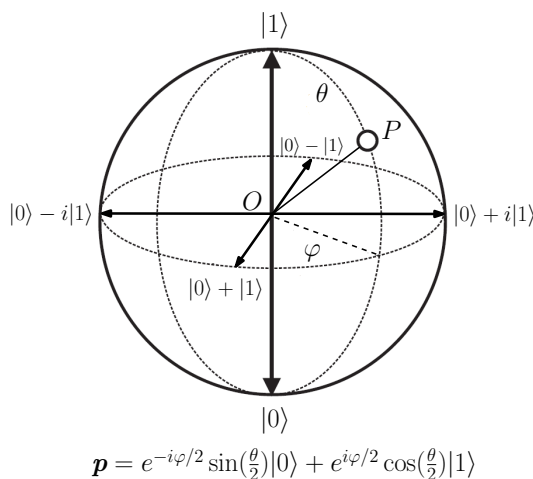
Since this corresponds to the point

$$[1, e^{i\varphi} \cot \frac{\theta}{2}] \sim [e^{-i\varphi/2} \sin \frac{\theta}{2}, e^{i\varphi/2} \cos \frac{\theta}{2}] \in \mathbf{P}_{\mathbb{C}}^1,$$

we conclude that

$$(5) \quad \mathbf{p} = e^{-i\varphi/2} \sin \frac{\theta}{2} |0\rangle + e^{i\varphi/2} \cos \frac{\theta}{2} |1\rangle \in \mathbf{P}_{\mathbb{C}}^1$$

is the point corresponding to P under the identification $S^2 \simeq \mathbf{P}_{\mathbb{C}}^1$. The picture below illustrates this relation and also shows some special cases (up to a normalization factor).



Remark 9.5. The formula (5) shows that $R_z(\alpha)(\mathbf{p})$ corresponds to $\rho_z(\alpha)(P)$, where $\rho_z(\alpha)$ denotes the rotation about the axis Oz of amplitude α . This is a special case of a well known relation between matrices $U \in SU(1)$ and rotations of S^2 . This relation can be explained as follows.

We can view a matrix $U = \begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix} \in SU(1)$ as a linear map $\mathbb{C}^2 \rightarrow \mathbb{C}^2$:

$$\begin{bmatrix} \xi_0 \\ \xi_1 \end{bmatrix} \mapsto U \begin{bmatrix} \xi_0 \\ \xi_1 \end{bmatrix}. \text{ This map induces a projective map of } \mathbf{P}_{\mathbb{C}}^1,$$

$$[\xi_0, \xi_1] \mapsto [u_0\xi_0 + u_1\xi_1, -\bar{u}_1\xi_0 + \bar{u}_0\xi_1]$$

and hence a map $\widehat{U} : \widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$,

$$\xi \mapsto \frac{\bar{u}_0\xi - \bar{u}_1}{u_1\xi + u_0}, \quad \infty \mapsto \bar{u}_0/u_1.$$

This map induces, in turn, the map $\tilde{U} : S^2 \rightarrow S^2$ such that $\tilde{U}(P(\xi)) = P(\widehat{U\xi})$.

If we take U to be one of the matrices

$$R_z(\varphi) = \begin{bmatrix} e^{-i\varphi/2} & 0 \\ 0 & e^{i\varphi/2} \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & \sin \frac{\theta}{2} \\ -\sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad R_x(\psi) = \begin{bmatrix} \cos \frac{\psi}{2} & -i \sin \frac{\psi}{2} \\ -i \sin \frac{\psi}{2} & \cos \frac{\psi}{2} \end{bmatrix}$$

(see Section 2, p. 9) then it turns out that $\rho_z(\varphi) = \widetilde{R_z(\varphi)}$, $\rho_y(\theta) = \widetilde{R_y(\theta)}$ and $\rho_x(\psi) = \widetilde{R_x(\psi)}$ are the rotations of amplitude φ , θ and ψ about the axes z , y and x , respectively. This, together with the relations

$$\mathbf{p} = e^{-\varphi/2} \sin \frac{\theta}{2} |0\rangle + e^{\varphi/2} \cos \frac{\theta}{2} |1\rangle = R_z(\varphi)R_y(\theta)|1\rangle,$$

show that, in terms of S^2 ,

$$P = \rho_z(\varphi)\rho_y(\theta)N,$$

whose geometric content is clear by the definitions of φ and θ . Conversely, this relation, together with the interpretation of R_z and R_y , provides a proof of the formula for \mathbf{p} .

Example 9.6 (q -Registers). By Axiom 4 (Entanglement) and the formula $\mathbf{H}^{(n)} \simeq \mathbf{H}^{(1)} \otimes \dots \otimes \mathbf{H}^{(1)}$, the space $\mathbf{H}^{(n)}$ is the associated space of $\Sigma^{(n)} = \Sigma^{(1)} + \dots + \Sigma^{(1)}$ (n summands), the system composed of n q -bits. By analogy with the classical bit registers, it is called a *q-register* of order n .

Now Axiom 3 (Unitary dynamics) tells us that the time evolution of $\Sigma^{(n)}$ is given by a unitary matrix of order 2^n . In other words, the time evolution of $\Sigma^{(n)}$ is a q -computation.

Finally, Axiom 2 (Observables) indicates that the [optional] operation $M(\mathbf{b})$ at the end of q -programs corresponds to the operation of measuring the (diagonal) observable

$$L = \sum_j |j\rangle\langle j| \quad (\text{that is, } L|k\rangle = |k\rangle \text{ for all } k)$$

when $\Sigma^{(n)}$ is in the state $|\mathbf{b}\rangle$. Note that $(\mathbf{H}^{(n)})_j = \mathbb{C}|j\rangle$, hence $P_j \mathbf{b} = b_j |j\rangle$ and $p_j = |b_j|^2$.

Quantum computers. From the preceding observations it follows that it is sufficient, in order to execute q -programs of order n on a physical support, to have a quantum register and “implementations” of the operations

$$\begin{aligned} &M(\mathbf{b}) \\ &R_j(U) \text{ [with } U \in \{H, U_{\pi/2}, U_{\pi/4}\} \text{ in the restricted case]} \\ &C_{j,k} \end{aligned}$$

A *quantum computer* (of order n) is a quantum register $\Sigma^{(n)}$ endowed with such implementations.

Its main beauty is that such a computer allows us to perform (or approximate) any q -computation.

An interesting feature of a quantum computer is that it supports *quantum parallelism*. This stems from the possibility of initializing the q -computation in states such as the Hadamard q -input

$$\mathbf{h}^{(n)} = \frac{1}{\sqrt{2^n}} (|0\rangle + |1\rangle + \cdots + |2^n - 1\rangle),$$

which has the following features:

- It contains (actually is the normalized sum of) all numbers of n bits.
- Hence, any operation of the quantum computer acts on all numbers simultaneously. This “explains” why the quantum computer can be much faster than a classical computer.
- In general, the usefulness of the algorithms (as Shor’s order-finding, for instance) is based in the fact that after its execution the amplitudes of “useful numbers” are high and the others are small.

Let us mention here the “problem of decoherence”, which arises from the fact that interactions with the environment can quickly “perturb” the states of $\Sigma^{(n)}$ (uncontrolled entanglement between states of the environment and states of $\Sigma^{(n)}$). Such problems in the road of building quantum computers are of a physical and technological nature. Research in many labs around the world is focussed on those questions, with continuous progress and in many directions (▷ 12).

More references. In addition to the references given so far, here are a few more books, ordered by year of publication, that the reader may find interesting to delve more deeply into the study of quantum computing or, more generally, quantum information processing: [17], [12, 13], [19], [3, 4], [14], [9], [2].

10. REMARKS AND PROOFS

► 1 (p. 3, p. 14). It is a well known fact that any classical computation can be resolved into a sequence of *logical gates* that are either NOT acting on a single bit or NAND acting on two bits (instead of NAND we could use any of a number of 2-bit gates, like AND, OR or XOR; the choice of NAND is just a convenience for our presentation). Therefore it is enough to embed NAND into a reversible computation f . This can be accomplished with $f : \mathbf{B}^3 \rightarrow \mathbf{B}^3$ that interchanges 110 and 111 and is the identity otherwise. Indeed, in the following table of f ,

x	000	001	010	011	100	101	110	111
$f(x)$	000	001	010	011	100	101	111	110

we see the NAND embedded by looking at the boldfaced bits. Notice that this embedding involves the four 3-bit vectors ending with 1, and that for these f acts as $ij1 \mapsto ij\text{NAND}(i, j) = ij(1 + i \cdot j)$.

► **2** (p. 10). For the proof, we follow the indications given in [16], Section 2.2.4. We will use the relations

$$XR_y(\theta)X = R_y(-\theta), \quad XR_z(\varphi)X = R_z(-\varphi).$$

Notice that XM (respectively MX) interchanges the rows (columns) of M . The claim follows from this and the definitions of $R_y(\theta)$ and $R_z(\varphi)$. Hence we can write (the φ in the third equality is an arbitrary auxiliary angle):

$$\begin{aligned} R_z(\beta)R_y(\theta)R_z(\gamma) &= R_z(\beta)R_y(\theta/2)R_y(\theta/2)R_z(\gamma) \\ &= R_z(\beta)R_y(\theta/2)XR_y(-\theta/2)XR_z(\gamma) \\ &= R_z(\beta)R_y(\theta/2)XR_y(-\theta/2)R_z(\varphi)R_z(-\varphi)XR_z(\gamma) \\ &= R_z(\beta)R_y(\theta/2)XR_y(-\theta/2)R_z(\varphi)XR_z(\varphi + \gamma) \\ &= AXBXC \end{aligned}$$

with

$$A = R_z(\beta)R_y(\theta/2), \quad B = R_y(-\theta/2)R_z(\varphi), \quad C = R_z(\varphi + \gamma).$$

Finally, since $ABC = R_z(2\varphi + \beta + \gamma)$, it is enough to choose $\varphi = -(\beta + \gamma)/2$, which means that

$$A = R_z(\beta)R_y\left(\frac{\theta}{2}\right), \quad B = R_y\left(-\frac{\varphi}{2}\right)R_z\left(-\frac{\beta+\gamma}{2}\right), \quad C = R_z\left(\frac{\gamma-\beta}{2}\right),$$

as claimed.

► **3** (p. 22). This result is not used in this article, but its proof is interesting and so we outline it.

Let $U = [u_{jk}] \in \mathbf{U}^{(n)}$ and set $N = 2^n$. Then $U = e^{i\alpha}U_1U_2 \cdots U_{N-1}$, with $\alpha \in \mathbb{R}$ and where $U_l = U_{l,l+1} \cdots U_{l,N}$, with $U_{l,j}$ an element of $S\mathbf{U}^{(1)}$ acting on the plane $[|l\rangle, |j\rangle]$ in the standard form (using the reference $|j\rangle$ and $|k\rangle$) and acting as the identity on any $|k\rangle$ such that $k \neq l, j$. This expression of U can be constructed as follows. The matrix $U_{1,2}$ is taken as the identity if $u_{21} = 0$ and otherwise as

$$\begin{bmatrix} u_{11}/\rho & -\bar{u}_{21}/\rho \\ u_{21}/\rho & \bar{u}_{11}/\rho \end{bmatrix}, \quad \rho = \sqrt{|u_{11}|^2 + |u_{12}|^2},$$

so that the entry 21 of the matrix $U_{1,2}^\dagger U$ is 0. Defining $U_{1,3}, \dots, U_{1,N}$ in a similar way, we achieve that all entries of the first column of $U' = U_{1,N}^\dagger \cdots U_{1,2}^\dagger U$, other than the entry 11, are 0. Since U' is unitary, so that any two of its columns are

orthogonal, all entries of the first row of U' , other than the entry 11, are also 0. Since the entry 11 of U' is a unit complex number, we see that there is $\alpha_1 \in \mathbb{R}$ such that $e^{-i\alpha_1} U_{1,N}^\dagger \cdots U_{1,2}^\dagger U$ has the form

$$\begin{bmatrix} 1 & \mathbf{0}_n \\ \mathbf{0}_n^\dagger & V \end{bmatrix}, \quad V \in \mathbf{U}^{(N-1)}.$$

Now, by induction, $V = e^{i\beta} U_2 \cdots U_{N-1}$, with $\beta \in \mathbb{R}$ and where $U_l = U_{l,l+1} \cdots U_{l,N}$ with $U_{l,j}$ an element of $S\mathbf{U}^{(1)}$ acting on the plane $[|l\rangle, |j\rangle]$ and as the identity on any $|k\rangle$, $k \neq l, j$. Finally the claim follows by defining $\alpha = \alpha_1 + \beta$ and $U_1 = U_{1,2} U_{1,3} \cdots U_{1,N}$. Note that the number of the $U_{l,j}$ different from the identity is at most $N(N-1)/2$.

The proof can be completed on noticing that in the Example 4.6 we established that the $U_{l,j}$ can be expressed as a product of U -gates and $N_{r,s}$ -gates.

► 4 (p. 22). We refer to Section 4.5.3 of [15] for a sketch of how the proof goes. But even in this encyclopedic book we read that providing all the details “is a little beyond our scope” (p. 198). A more complete proof, including the more subtle mathematical details, can be found in [16]. In particular it contains a full proof of the key fact that if $\cos \alpha = \cos^2(\pi/8)$, then α/π is irrational (Lemma 3.1.8).

► 5 (p. 27). Since $\text{Av}(\mathbf{a}) = \frac{1}{N} \frac{N-M}{\sqrt{N-M}} = \sqrt{N-M}/N$,

$$\begin{aligned} K(\mathbf{a}) &= \sum_{j \in J_0} (2\sqrt{N-M}/N - 1/\sqrt{N-M}) |j\rangle + \sum_{j \in J_1} \frac{2\sqrt{N-M}}{N} |j\rangle \\ &= \sum_{j \in J_0} \frac{N-2M}{N\sqrt{N-M}} |j\rangle + \sum_{j \in J_1} \frac{2\sqrt{M}\sqrt{N-M}}{N\sqrt{M}} |j\rangle \\ &= \cos(\varphi) \mathbf{a} + \sin(\varphi) \mathbf{b}. \end{aligned}$$

Similarly, since $\text{Av}(\mathbf{v}) = M/N\sqrt{M} = \sqrt{M}/N$,

$$\begin{aligned} K(\mathbf{b}) &= \sum_{j \in J_0} \left(\frac{2\sqrt{M}}{N} \right) |j\rangle + \sum_{j \in J_1} \left(\frac{2\sqrt{M}}{N} - \frac{1}{\sqrt{M}} \right) |j\rangle \\ &= \sum_{j \in J_0} \left(\frac{2\sqrt{M}\sqrt{N-M}}{N} \frac{1}{\sqrt{N-M}} \right) |j\rangle + \sum_{j \in J_1} \left(\frac{2M-N}{N\sqrt{M}} \right) |j\rangle \\ &= \sin(\varphi) \mathbf{a} - \cos(\varphi) \mathbf{b}. \end{aligned}$$

Observe that the relations above imply that K is, on the space spanned by \mathbf{a} and \mathbf{b} , the symmetry with respect to $\mathbf{h}^{(n)} = \cos(\varphi/2) \mathbf{a} + \sin(\varphi/2) \mathbf{b}$. Indeed, let $\mathbf{u}_\alpha = \cos(\alpha) \mathbf{a} + \sin(\alpha) \mathbf{b}$ (thus $\mathbf{h}^{(n)} = \mathbf{u}_{\varphi/2}$) and let R_φ denote the rotation

of amplitude φ . Then $K = R_\varphi G$ (from $KG = R_\varphi$ and $G^2 = Id$) and

$$K(\mathbf{h}^{(n)}) = R_\varphi(G(\mathbf{u}_{\varphi/2})) = R_\varphi(\mathbf{u}_{-\varphi/2}) = \mathbf{u}_{\varphi/2} = \mathbf{h}^{(n)},$$

while, if $\mathbf{k}^{(n)} = -\sin(\varphi/2)\mathbf{a} + \cos(\varphi/2)\mathbf{b} = \mathbf{u}_{\varphi/2+\pi/2}$, then

$$\begin{aligned} K(\mathbf{k}^{(n)}) &= R_\varphi G R_{\pi/2} \mathbf{u}_{\varphi/2} = R_\varphi R_{-\pi/2} G \mathbf{u}_{\varphi/2} \\ &= R_{\varphi-\pi/2} \mathbf{u}_{-\varphi/2} = \mathbf{u}_{\varphi/2-\pi/2} = -\mathbf{k}^{(n)}. \end{aligned}$$

► 6 (p. 32). The probability $p(|2^m\varphi - l| > 2^{m-r})$ is equal to

$$\sum_{l=-2^{m-1}+1}^{-(2^{m-r}+1)} pl + \sum_{l=(2^{m-r}+1)}^{2^{m-1}} pl.$$

Now the bound (**), p. 32, can be derived from the explicit expression (*) for p_l (page 32). We refer to [15] for further details.

► 7 (p. 34). The Prime Number Theorem asserts that the number of primes which are smaller than r is asymptotically equal to $\frac{r}{\log(r)}$. Hence, the probability of choosing (uniformly) a random prime number $0 < s < r$ is asymptotically equal to

$$p(0 < s < r, s \text{ is prime}) = p \sim \frac{1}{\log r} > \frac{1}{\log N}.$$

Then the expected number of iterations in order to find a prime number $s < r$ is equal to:

$$\sum_{i=1}^{\infty} i(1-p)^{i-1}p = p \sum_{i=1}^{\infty} i(1-p)^{i-1} = \frac{p}{(1-(1-p))^2} = \frac{1}{p} \sim \log(r) < \log(N).$$

Hence, after not more than $\log(N) = O(n)$ choices, we expect to choose a value of s which is prime with r .

► 8 (p. 34). The continuous fraction representation of a rational number x is a vector of integers $[x_0, x_1, \dots, x_n]$, with $x_j > 0$ for $j = 1, \dots, n$. The relation between x and $[x_0, x_1, \dots, x_n]$ can be displayed as a ‘continuous fraction’:

$$x = x_0 + \frac{1}{x_1 + \frac{1}{x_2 + \frac{1}{x_3 + \frac{1}{x_4 + \frac{1}{x_5 + \frac{1}{x_6 + \frac{1}{x_7 + \frac{1}{x_8 + \frac{1}{x_9 + \frac{1}{x_{n-1} + \frac{1}{x_n}}}}}}}}}}}}}}$$

By abuse of notation we will also write $x = [x_0, x_1, \dots, x_n]$. In these terms the continuous fraction can be expressed by the recursive formula

$$[x_0, x_1, \dots, x_n] = x_0 + \frac{1}{[x_1, \dots, x_n]}$$

The rational numbers $c_j = [x_0, x_1, \dots, x_j]$, $j = 0, 1, \dots, n$, are called the convergents of the number x . The list of denominators $\{d_0, d_1, \dots, d_n\}$ of these convergents can be computed recursively as follows:

$$d_0 = 1, \quad d_1 = x_1, \quad d_j = x_j d_{j-1} + d_{j-2} \quad (j = 2, \dots, n)$$

Actually it is easy to prove by induction that $c_j = m_j/d_j$, where

$$m_0 = x_0, \quad m_1 = x_1 x_0 + 1, \quad m_j = x_j m_{j-1} + m_{j-2} \quad (j = 2, \dots, n)$$

It follows that the list $\{d_0, d_1, \dots, d_n\}$ can be computed by the following algorithm:

```

ContFrac(x) :=
a=floor(x), k=1, d={0,1}
while x!=a and j<n do
  x=1/(x-a)
  a=floor(x)
  d=d|{a*d.(j-1)+d.(j-2)}
  j=j+1
return tail(d)
    
```

► 9 (p. 36). We prove that

$$p \left(x \in \mathbb{Z}_N^* \mid r = \text{ord}_N(x) \text{ is odd or } x^{\frac{r}{2}} + 1 \text{ is divisible by } N \right) \geq \frac{1}{2^m}.$$

We start writing $N = p_1^{\alpha_1} \dots p_m^{\alpha_m}$, where p_1, \dots, p_m are distinct prime numbers. Then, $\mathbb{Z}_N^* = \mathbb{Z}_{p_1^{\alpha_1}}^* \times \dots \times \mathbb{Z}_{p_m^{\alpha_m}}^*$. Write x_j for the reduction of $x \pmod{p_j^{\alpha_j}}$, and r_j for the order of x_j in $\mathbb{Z}_{p_j^{\alpha_j}}^*$. Denote by d_j the biggest exponent such that 2^{d_j} divides r_j . Denote by d the biggest exponent such that 2^d divides r . Then, it is easy to show that if r is odd or if r is even and $x^{\frac{r}{2}} \equiv -1 \pmod{N}$, then $d_j = d$ for all d .

To conclude, we use that if 2^{d_j} is the largest power of 2 dividing $\varphi(p_j^{\alpha_j})$, then

$$p \left(x \in \mathbb{Z}_N^* \mid 2^{d_j} \text{ divides } \text{ord}_{p_j^{\alpha_j}}(x) \right) = \frac{1}{2}$$

► 10 (p. 37). Denote by p the probability of the event $\{x \in \mathbb{Z}_N^* \mid r = \text{ord}_N(x) \text{ is even and } x^{\frac{r}{2}} + 1 \text{ is not divisible by } N\}$, when x is chosen uniformly

at random at \mathbb{Z}_N^* . Then, the expected number of iterations of the algorithm is equal to:

$$\sum_{i=1}^{\infty} i(1-p)^{i-1}p = p \sum_{i=1}^{\infty} i(1-p)^{i-1} = \frac{1}{p} \leq \frac{2^{m-1}}{2^{m-1}-1} = 1 + \frac{1}{2^{m-1}-1}.$$

As $m > 1$, the expected number of iterations is $O(1)$.

► **11** (p. 39). If $\mathbf{u}_1, \dots, \mathbf{u}_n$ is an orthonormal basis of E and If $\mathbf{u}'_1, \dots, \mathbf{u}'_{n'}$ an orthonormal basis of E' , then a general vector of $E \otimes E'$ has the form $\sum_{j,j'} a_{j,j'} \mathbf{u}_j \otimes \mathbf{u}_{j'}$. On the other hand, the composite vector $\mathbf{x} \otimes \mathbf{x}'$ has the form $\sum_{j,j'} a_j a_{j'} \mathbf{u}_j \otimes \mathbf{u}_{j'}$ if $\mathbf{x} = \sum_j a_j \mathbf{u}_j$ and $\mathbf{x}' = \sum_{j'} a_{j'} \mathbf{u}_{j'}$. Now this vector may not be equal to $\mathbf{u}_1 \otimes \mathbf{u}'_1 + \mathbf{u}_2 \otimes \mathbf{u}'_2$, for this would in particular require that the inconsistent relations $a_0 a'_0 = 1, a_1 a'_1 = 1, a_0 a'_1 = 0$ were satisfied.

► **12** (p. 43). There is an explosion of activity in the last years, and especially since 2006, as seen, for example, in

http://en.wikipedia.org/wiki/Timeline_of_quantum_computing

http://en.wikipedia.org/wiki/Quantum_computer

In the latter, in particular, there are over a dozen lines of inquiry aiming at the realization of a quantum computer.

REFERENCES

- [1] T. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer-Verlag, 1976.
- [2] F. Benatti, M. Fannes, R. Floreanini, and D. (editors) Petritis. *Quantum information, computation and cryptography. An introductory survey of theory, technology and experiments*, volume 808 of *Lecture Notes in Physics*. Springer, 2010.
- [3] Giuliano Benenti, Giulio Casati, and Giuliano Strini. *Principles of quantum computation and information*, volume I: Basic cocepts. World Scientific, 2004.
- [4] Giuliano Benenti, Giulio Casati, and Giuliano Strini. *Principles of quantum computation and information*, volume II: Basic tools and special topics. World Scientific, 2007.
- [5] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc Roy Soc Lond A*, 439:553–558, October 1992.
- [6] L. K. Grover. A fast quantum mechanical algorithm for database search. Annual ACM Symposium on Theory of Computing, pages 212–219, 1996.
- [7] L. K. Grover. From Schrödinger’s equation to quantum search algorithm. *American Journal of Physics*, 69(7):769–777, 2001.
- [8] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 2008 (6th edition, revised by D. R. Heath-Brown and J. H. Silvermann; 1st edition published in 1938).
- [9] G. Jaeger. *Quantum Information –An overview*. Springer, 2007.
- [10] A. Kitaev, A. H. Shen, and M. N. Vyalıy. *Classical and quantum computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

- [11] A. Y. Kitaev. Quantum measurements and the abelian stabilizer problem. *Electr. Coll. Comput. Complex.*, 3:article no. 3, 22 pp., 1995.
- [12] S. J. Jr. Lomonaco, editor. *Quantum computation: A grand mathematical challenge for the twenty-first century and the millenium*, volume 58 of *Proceedings of Symposica in Applied Mathematics*. American Mathematical Society, 2002.
- [13] S. J. Jr. Lomonaco and H Brand, editors. *Quantum computation and information*, volume 305 of *Contemporary Mathematics*. American Mathematical Society, 2002.
- [14] D. Mermin. *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.
- [15] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000 (5th printing 2005).
- [16] K. R. Parthasarathy. *Lectures on Quantum Computation, Quantum Error Correcting Codes and Information Theory*. Narosa Publishing House, 2006. (For the Tata Institute of Fundamental Research, international distribution by AMS).
- [17] A. O. Pittenger. *An Introduction to Quantum Computing Algorithms*. Progress in Computer Science and Applied Logic. Birkhuser, 2000.
- [18] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
- [19] Joachim Stolze and Dieter Suter. *Quantum Computing: a Short Course from Theory to Experiment*. Physics Textbook. Wiley-VCH, 2008. Second, updated and enlarged edition.
- [20] A. Sudbery. *Quantum mechanics and the particles of nature. An outline for mathematicians*. Cambridge University Press, 1988 (reprinting, with corrections, of 1986 edition).
- [21] Steven Weinberg. *Lectures on quantum mechanics*. Cambridge University Press, 2013.

List of frequently used symbols

$\mathbf{0}_n$		$0 \dots 0$
$\mathbf{1}_n$		$1 \dots 1$
$\langle \mathbf{a} \mathbf{b} \rangle$	Scalar product of \mathbf{a} and \mathbf{b}	$\sum_j a_j \bar{b}_j$
$\mathbf{a} \widehat{\otimes} \mathbf{b}, \mathbf{a} \otimes \mathbf{b}$	Tensorial product of \mathbf{a} and \mathbf{b}	
\mathbf{B}	Set of binary digits	$\{0, 1\}$
$C_{12}(U)$	Controlled- U	
$ j\rangle$	Ket of j	
F	Quantum Fourier Transform	
I_{2^n}	Identity matrix of dimension 2^n	
$\mathbf{H}^{(n)}$	Space of q -vectors of order n	\mathbb{C}^{2^n}
$\mathbf{h}^{(n)}$	Hadamard q -vector of length n	$\rho^n \left(\sum_{j=0}^{2^n-1} j\rangle \right)$
H	Hadamard matrix	$\rho \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
$M_L(\mathbf{a})$	q -measurement of \mathbf{a} at the positions $\{l_1, \dots, l_r\}$	
$N_{r,s}$	Controlled negation	$C_{r,s}(X)$
ρ		$1/\sqrt{2}$

$R_l(U)$	One q -bit rotation, $U \in \mathbf{U}^{(1)}$	$\left[\begin{array}{cc} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{array} \right]$ $\left[\begin{array}{cc} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{array} \right]$ $\left[\begin{array}{c} 1 \\ 1 \end{array} \right]$ $\left[\begin{array}{c} -i \\ i \end{array} \right]$ $\left[\begin{array}{c} 1 \\ -1 \end{array} \right]$ $\left[\begin{array}{cc} 1 & 0 \\ 0 & e^{i\alpha} \end{array} \right]$
$R_y(\varphi)$		
$R_z(\theta)$		
$\sigma_x = X$	Pauli matrix x	
$\sigma_y = Y$	Pauli matrix y	
$\sigma_z = Z$	Pauli matrix z	
S_α	Shift matrix of angle α	
SWAP[r, s]	Transposition of bits r and s	
$\mathbf{U}^{(n)}$	Group of unitary matrices of dimension 2^n	
$U \otimes U'$	Tensorial product of matrices	
U^\dagger	Adjoint matrix	$\overline{U^T}$
$U^{\otimes n}$		$U \otimes \dots \otimes U$

INSTITUTO DE CIENCIAS MATEMÁTICAS-CSIC, CALLE NICOLÁS CABRERA 13-15, CAMPUS CANTOBLANCO UAM, 28049 MADRID, SPAIN.

MATEMÀTICA APLICADA II, UNIVERSITAT POLITÈCNICA DE CATALUNYA, EDIFICI OMEGA, C/ JORDI GIRONA 1-3, 08034 BARCELONA, SPAIN.

E-mail address: juanjo.rue@icmat.es, sebastia.xambo@upc.edu