Generating images with a quantum computer

BATXILLERAT'S RESEARCH WORK

IES MIQUEL TARRADELL

Author:

Tomàs Ockier Poblet
2nd Batxillerat
ockier1@gmail.com

Tutor:

Tomàs Ockier Poblet

Institut Miquel Tarradell

September 19, 2021
Barcelona, Barcelona

Index

Chapter 1

Introduction

Desde hace más de un año, me he dedicado a estudiar computación cuántica durante mi tiempo libre. Buscaba investigar un campo relacionado con la mecánica cuántica, pero sin que sea muy complicado, que se pueda entender a un nivel teórico y que me entusiasme.

La Computación Cuántica encaja perfectamente con esos criterios. Es más sencilla que la mecánica cuántica debido a que no está basada en cálculo o ecuaciones diferenciales, se basa en la álgebra lineal, utilizando valores discretos, vectores y matrices. Además si se trabaja a un nivel teórico sencillo, no se tienen en consideración las interpretaciones físicas, lo cual simplifica mucho las cosas. Cuanto más me adentraba, más ganas tenía de seguir.

Mi parte favorita de este campo es el Quantum Machine Learning que consiste en diseñar y aplicar conceptos de Machine Learning a los ordenadores cuánticos, como por ejemplo implementar cuánticamente las famosas Redes Neuronales, que están detrás de la mayoría de inteligencias artificiales que vemos hoy en día [1].

QML es un campo de investigación joven y en crecimiento debido a que sus algoritmos son ideales para implementarlos con los ordenadores cuánticos actuales, los cuales no son muy potentes. Ejemplos de estas implementaciones serían [insertar aplicaciones aquí], etc.

De entre todos los tipos de algoritmos me he centrado en las Redes Neuronales Cuánticas, análogas cuánticas de las Redes Neuronales tan utilizadas

hoy en día para hacer gran variedad de tareas. Me he interesado particularmente en ellas debido a que tenía experiencia en el pasado con las RNs clásicas y había visto que existen frameworks de software para trabajar con ellas como TensorFlow Quantum [2] que me podían ayudar.

Para adentrarme en el campo de QML, he tenido que adquirir conocimientos en álgebra lineal, cálculo y física. Dentro de QML en concreto me he dedicado a leer papers que me interesan y en un par de ocasiones intentar implementar los algoritmos detallados en esos papers. Puede parecer algo imposible en principio debido a que no tengo acceso directo a un ordenador cuántico, no obstante estos no son necesarios debido a que las operaciones cuánticas pueden ser simuladas en un ordenador corriente de escritorio (con ciertas limitaciones). Pero puedo tener acceso a ordenadores cuánticos ya que IBM permite acceder a los suyos mediante IBM Quantum Experience [3], aunque nunca he dado uso de ello debido a que no lo veía necesario.

En este trabajo de investigación me he propuesto implementar mediante código uno de los algoritmos que he visto en un paper, una Red Adversaria Generativa Cuàntica (GAN, en inglés) [4] que genera imágenes a partir de un circuito cuántico [5]. Como objetivo tengo verificar una sugerencia que hacen los autores del paper: implementar una función no-lineal en una parte del algoritmo que podría mejorar el rendimiento de este. Mi hipótesis al igual que los autores (aunque ellos lo comentan muy brevemente) es que el algoritmo va a reducir ligeramente el número de interacciones que son necesarias para llegar a su punto óptimo. Es decir, el modelo con la función no-lineal va a necesitar menos operaciones que lo entren conseguir los mismos resultados que el modelo sin la función.

Chapter 2

Theoretical Framework

2.1 Linear Algebra

When I start looking into quantum computation, I realized quickly that I needed more mathematical knowledge to understand the mathematical notation and concepts that were on the quantum computation textbooks. By that time I came across the wonderful video lectures on lineal algebra -which is the branch of mathematics that is the background for quantum computing- by Prof. Gilbert Strang at MIT [6, 7]. I watched nearly all of them along a part of the complementary videos that went through the homework.

Those lectures really helped me to understand the math on *Quantum Computation and Quantum Information* [8] and *Quantum Computing: A Gentle Introduction* [9]. Little by little I learned the math notation used in quantum mechanics, the Dirac notation.

In the current section I will be going through the basic concepts of linear algebra, to form the mathematical background used along this work.

2.1.1 Vectors and Vector Spaces

The basic objects of linear algebra are vector spaces. A vector space is the set of all the vectors that have the same dimensions. For example $\mathbb{R}^3$ would be the vector space of all 3 dimensional vectors that can be used to represent all possible

coordinates on a 3D space. In quantum computing I a special kind of vector spaces are used: Hilbert spaces, in other words, an inner product space [10]. Hilbert spaces follow a set of products and have certain properties, on section 2.1 I'll be going through a part of those products and properties, the amount that is compulsory. Keep in mind that Hilbert spaces are much more complicated than what is presented here. Keep in mind that when mentioning a vector space in this work, that vector space is a complex Hilbert space, unless it's specified otherwise.

Vector spaces are defined by their bases, a set of vectors $B = \{|v_1\rangle, \ldots, |v_n\rangle\}$ is a valid base for a vector space V if any vector $|v\rangle$ in it, can be written as $|v\rangle = \sum_i a_i |v_i\rangle$ for $|v_i\rangle \in B$. Every vector in B are linearly independent to each other.

The standard notation for linear algebra concepts in quantum mechanics is the Dirac notation, which represents a vector as $|\psi\rangle$. Where ψ is the label of the vector. A vector $|\psi\rangle$ with n dimensions can be also represented as a column matrix with the form of:

$$|\psi\rangle = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix}$$

With the complex numbers $(z_1, z_2, \ldots, z_{n-1}, z_n)$ as its elements. A vector written as $|\psi\rangle$ is named *ket*.

Addition of a pair of the vectors on a Hilbert space it defined by [1]:

$$|\psi\rangle + |\varphi\rangle = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_n \end{bmatrix} + \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_n \end{bmatrix}$$

Moreover, there is a scalar multiplication defined by:

$$z |\psi\rangle = z \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_n \end{bmatrix} = \begin{bmatrix} z\psi_1 \\ \vdots \\ z\psi_n \end{bmatrix}$$

---

[1]The vectors on this definition have their elements represented with their label and a subscript e.g. the vector $|\psi\rangle$ as an element ψ_i in it and its first element is ψ_1. This notation is followed from now on.

Where z is a scalar and $|\psi\rangle$ a vector. Note that each element of the vector is multiplied by the scalar.

Because Hilbert spaces are complex spaces they have a conjugate defined for scalars as: For a complex scalar $z = a + bi$ its conjugate z^* is equal to $a - bi$. This notion can be extended for both vectors and matrices taking the conjugate of all the entries:

$$|\psi\rangle^* = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_n \end{bmatrix} * = \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_n^* \end{bmatrix}$$

$$A^* = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}^* = \begin{bmatrix} A_{11}^* & \cdots & A_{1n}^* \\ \vdots & \ddots & \vdots \\ A_{m1}^* & \cdots & A_{mn}^* \end{bmatrix}$$

With $|\psi\rangle$ being a vector and A being a matrix of dimension $m \times n$.

Another important concept is the transpose represented with a superscript T that 'flips' a vector or a matrix. A column vector with dimension $n, 1$ becomes a row vector with dimension $1, n$ [2]:

$$|\psi\rangle^T = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_n \end{bmatrix}^T = \begin{bmatrix} \psi_1 & \ldots & \psi_n \end{bmatrix}$$

The same holds for matrices, a transposed $m \times n$ matrix becomes a $n \times m$ matrix. For example:

$$A^T = \begin{bmatrix} 2 & 3 \\ 6 & 4 \\ 2 & 5 \end{bmatrix}^T = \begin{bmatrix} 2 & 6 & 2 \\ 3 & 4 & 5 \end{bmatrix}$$

The combination of the complex conjugate and the transpose is named the Hermitian conjugate, written as a superscript †. For a vector $|\psi\rangle$ its Hermitian conjugate $|\psi\rangle^\dagger$ is:

$$|\psi\rangle^\dagger = (|\psi\rangle^*)^T = \begin{bmatrix} \psi_1^* & \ldots & \psi_n^* \end{bmatrix} = \langle\psi|$$

This equivalence works both ways: $|\psi\rangle^\dagger = \langle\psi|$ and $\langle\psi|^\dagger = |\psi\rangle$.

---

[2]In reality column vectors are matrices of dimension n,1 but I have been omitting the 1. When referring to the dimensions of any vector I am going to say just a number, nonetheless, I will also specify the type of vector -a column or a row one-.

The Hermitian conjugate of a column vector $|\psi\rangle$ is named *bra* or dual vector. On Dirac notation is written as $\langle\psi|$.

2.1.2   Linear Operators

To operate with vectors and do operations on them matrices are used. Matrices are also named linear maps or linear operators, which are names that illustrate better what they do. The formal definition of a linear operator can be quite complicated, thus, in this section I am going to explained in more mundane terms.

Put it simply, a linear operator transforms a vector onto another vector, those two vectors can be on different spaces or not [11]. More formally, for a vector $|v\rangle$ on the space V and a vector $|w\rangle$ on the space W, a linear operator A between this two vectors, performs the action:

$$A\,|v\rangle = |w\rangle$$

In other words, it maps an element of the vector space V to an element of the vector space W. Linear operators must preserve the following operations:

1. Vector addition:
   For the vectors $|\psi\rangle$ and $|\varphi\rangle$ on the same vector space, and the linear operator A:
   $$A(|\psi\rangle + |\varphi\rangle) = A\,|\psi\rangle + A\,|\varphi\rangle$$

2. Scalar multiplication:
   For the vector $|\psi\rangle$, the scalar z and the linear operator A:
   $$A(z\,|\psi\rangle) = zA\,|\psi\rangle$$

These affirmations have to be true for all vectors and all scalars on the spaces that the linear operator acts upon. Note that a linear operator doesn't have to be a matrix necessarily, for example, derivatives and integrals are linear operators, you can check easily that they follow the criteria specified above. However, they are usually applied to functions, although, it is possible to make derivatives and integrals with matrices.[3]

Matrices are just the matrix representation of a linear operator [12].

---

[3]Don't you worry, I am definitely going to do that, and I hope you are as excited as I am :D .

2.1.2.1 Types of linear operators On the current section, I am going to cover the basic types of linear operators that are indispensable for the following theory on this chapter and the experimental work.

1. **Zero operator**

   The zero operator Every vector space has a zero vector that in Dirac notation is represented by 0, since, $|0\rangle$ is belong to something completely different on QC and QI. The zero vector is such that for any other vector $|\psi\rangle$ and any scalar z: $|\psi\rangle + 0 = |\psi\rangle$ and $z0 = 0$. Note that the zero operator is also represented as 0 and its defined as the operator that maps all vectors to the zero vector: $0 |\psi\rangle = 0$.

2. **Inverse matrix**

   A squared matrix[4] A is called invertible if there exist a matrix A^{-1} such that $AA^{-1} = A^{-1}A$. A^{-1} is the inverse matrix of A. The fastest way to see if a square matrix is invertible is to check if it has a non-zero determinant.

3. **Identity operator**

   For any vector space V there exist an identity operator I that is defined as $I |\psi\rangle = |\psi\rangle$, this operator doesn't do anything to the vectors that it is applied to. Note that for any matrix A and its inverse A^{-1} is true that $AA^{-1} = I$.

4. **Unitary operator**

   A unitary operator is any operator that doesn't change the norm of the vectors that it is applied to, moreover, a matrix A is unitary if $AA^{\dagger} = I$. To convert any operator to unitary all its entries are divided by the norm of the operator.

5. **Hermitian operators**

   A Hermitian operator or self-adjoint operator is any operator that its Hermitian conjugate is itself $A = A^{\dagger}$. Another thing to point out, is that there exist a unique operator A on a Hilbert space. Such that for any vectors $|\psi\rangle$ and $|\varphi\rangle$:

   $$\langle\psi| (A |\varphi\rangle) = (A^{\dagger} \langle\psi|) |\varphi\rangle$$

   This operator is known as the adjoint or Hermitian conjugate of A.

---

[4]A square matrix is a matrix with dimensions $n \times n$.

2.1.3 Inner Product and Outer Product

2.1.3.1 Inner product A dual vector $\langle\psi|$ and a vector $|\varphi\rangle$ combined form the inner product $\langle\psi|\varphi\rangle$ which is an operation that takes two vectors of the same space as input and produces a complex number as output:

$$\langle a|b\rangle = a_1 b_1 + a_2 b_2 + ... + a_{n-1}b_{n-1} + a_n b_n = z$$

With $z, a_i, b_i \in \mathbb{C}$. When referring to an inner product I will often say 'the inner product of two vectors' when in reality is an operation between a dual vector and a vector.

The equivalent to this product on a 2 dimensional real space $\mathbb{R}^2$ is the dot product, which is also expressed as:

$$\langle a|b\rangle = \||a\rangle\|_2 \cdot \||b\rangle\|_2 \cos\theta \tag{2.1}$$

With $\|\cdot\|_2$ being the ℓ^2 norm defined by $\||\psi\rangle\|_2 = \sqrt{\psi_1^2 + ... + \psi_n^2}$ and θ being the angle between the vectors $|a\rangle$ and $|b\rangle$. As I said the equation (2.1) is equivalent to the inner product, however, from what I have seen, it is not widely used when working in high dimensional spaces because interpreting θ as an angle doesn't make sense when referring to vectors that have more than 2 dimensions. Instead, I have seen more often the inner product presented in its geometrical representation[5] as the product between one row vector and one column vector:

$$\langle a|b\rangle = \begin{bmatrix} a_1...a_n \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

I have already defined the ℓ_2 norm as the square root of the sum of the squared entries of a vector:

$$\||a\rangle\|_2 = \sqrt{\sum_i a_i^2}$$

---

[5]The exact details of the geometrical representation of matrices and vectors are out of the scope of this work, even so, it gives a clear and intuitive way of looking at vectors and vector spaces [citation].

Nonetheless, the more common definition is based upon the inner product. As you can see the inner product of a vector by itself is the sum of the squared entries:

$$\langle a|a\rangle = a_1 a_1 + ... + a_n a_n = a_1^2 + ... + a_n^2 = \sum_i a_i^2$$

Thus, the norm can be defined as the square root of the inner product of a vector:

$$\||a\rangle\|_2 = \sqrt{\langle a|a\rangle} \tag{2.2}$$

When the norm is applied to a 2 dimensional vector you can see that is the same as the length of that vector, that is because norm and length are the same concepts, however, the norm is the generalized length that can be applied to a vector of any dimension.

From what I understand some properties of the length of a two dimensional vector do not hold with the norm of a vector that has more than 2 dimensions. In other words, the norm behaves in similar ways like the distance from the origin (which is the length), thus they are not the exact same thing. Moreover, there are different types of norm[6] that are used in different types of scenarios. That is why I am referring to a ℓ^2 norm, a specific type of norm that is also named Euclidean norm which is used to define the ℓ^2 distance or Euclidean distance, widely used to measure the distance of two points in a 2D space or a 3D space in high school [13].

2.1.3.2 Properties of the Inner Product

The basic properties of the inner product are as follows:

1. Is linear in the second argument $(z_1 \langle a| + z_2 \langle c|)|b\rangle = z_1 \langle a|b\rangle + z_2 \langle c|b\rangle$

2. Conjugate symmetry $\langle a|b\rangle = (\langle b|a\rangle)^*$

3. $\langle a|a\rangle$ is non-negative and real, except in the case of $\langle a|a\rangle = 0 \Leftrightarrow |a\rangle = 0$

2.1.3.3 Orthonormal and orthogonal vectors

From the concept of norm comes the concepts a pair of orthogonal vectors and a pair of orthonormal vectors[7].

---

[6]But not different types of length, that I know of at least.

[7]Funny note, when I encountered these two terms for the first time in QC and QI [citation], I taught they were the same thing and a week passed until I realized. It was such a difficult mess to understand everything else with these two terms confused.

Looking at the equation (2.2) we can see that if the inner product of a vector is one, the norm of this vector is also one. A vector that has norm one is named a unit vector. Therefore, if the inner product of a vector is one, that vector is a unit vector.

A pair of non-zero vectors are orthogonal if their inner product is zero. For two non-zero 2 dimensional vectors, if their inner product is equal to zero, you can see that they are perpendicular to each other by looking at equation (2.1):

For $|a\rangle$ and $|b\rangle \neq 0$:
$$\text{If } \langle a|b\rangle = 0 \text{ then: } \||a\rangle\|_2 \cdot \||b\rangle\|_2 \cos\theta = 0$$
Because $|a\rangle$ and $|b\rangle$ are non-zero vectors, their norms can't be zero.
Thus the remainder term $\cos\theta$ is equal to zero.
Therefore, the angle θ as to be $\dfrac{\pi}{2}$.

However, thinking that perpendicularity and orthogonality are the same concepts is a mistake, since, it only holds when looking at 2 dimensional vectors. As with norm and length, orthogonality is the generalized concept of perpendicularity that works for high dimensional vectors.

When we mix the concepts of unit vector and orthogonal vectors we arrive at the term orthonormality [10]. A pair of non-zero vectors are orthonormal when both are unit vectors and there are orthogonal to each other:

$$|a\rangle \text{ and } |b\rangle \text{ are othornormal if } \begin{cases} \langle a|b\rangle = 0 \\ \langle a|a\rangle = 1 \\ \langle b|b\rangle = 1 \end{cases}$$

Orthonormal vectors are important, they are broadly used in quantum computation as well as quantum mechanics because they form the basis for the vector spaces on which the quantum states are located.
One thing to point out is that I have been talking about a pair of vectors when referring to orthonormal vectors, however, orthonormality can be extended to a set of vectors. If a set has all unit vectors and the vectors are orthogonal to each other, the set is orthonormal. The set of vectors $B = \{|\beta_1\rangle, |\beta_2\rangle, ..., |\beta_{n-1}\rangle |\beta_n\rangle\}$ is

maybe introduce them before if you talk about basis later on

orthonormal if $\langle \beta_i | \beta_i \rangle = \delta_{ij} \forall i, j$ [10] where δ_{ij} is the Kronecker delta defined as :

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

2.1.3.4   Outer product   The outer product is a function that takes two vectors -expressed as $|a\rangle \langle b|$, with $|a\rangle$ and $|b\rangle$ being vectors- and produces an linear operator as output. Unlike the inner product, there is no analog for the outer product on the mathematics taught in high school[8], and it is a bit difficult to understand it as it can take two vectors from different spaces as input. It is defined as follows:

For a vector $|v\rangle$ and $|v'\rangle$ of dimensions m and a vector $|w\rangle$ of dimension n. The output is a linear operator A of dimensions $m \times n$ in the space $M_{m \times n}$:

$$|v\rangle \langle w| = A \text{ with } A \in \text{Mat}_{m \times n}.$$

Whose action is defined by:

$$(|v\rangle \langle w|) |v'\rangle \equiv |w\rangle \langle v|v'\rangle = \langle v|v'\rangle |w\rangle \tag{2.3}$$

From equation (2.3) the usefulness and meaning of the outer product are hard to comprehend, so I will look at the way to compute it next to clarify how it works. For two vectors $|a\rangle$ and $|b\rangle$ of dimensions m and n respectively, their outer product is computed multiplying each element of $|a\rangle$ by each element of $|b\rangle$ forming a matrix of size $m \times n$:

$$|a\rangle \langle b| = \begin{bmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m b_1 & a_m b_2 & \cdots & a_m b_n \end{bmatrix}$$

The usefulness of the outer product will be shown in future sections.

2.1.4   Tensor product

The last product to mention is the tensor product, represented with the symbol $\otimes$. This product is used to create larger vector spaces by combining smaller vector

---

[8]The analog of the inner product would be the dot product.

spaces. The formal explanation of this concept is quite difficult, so I will focus on explaining the way to compute it by using the matrix representation of this product, named the Kronecker product.

For a $m \times n$ matrix A and a $p \times q$ matrix B the output of their Kronecker product [14] is a $pm \times qn$ matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}$$

Note that $a_{ij}B$ is a scalar multiplication by a matrix, with a_{ij} being the scalar and B being the matrix.

Here is a clearer example with two 2×2 matrices, note that each entry of the first matrix is multiplied by the second matrix:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 1 \times 0 & 1 \times 5 & 2 \times 0 & 2 \times 5 \\ 1 \times 6 & 1 \times 7 & 2 \times 6 & 2 \times 7 \\ 3 \times 0 & 3 \times 5 & 4 \times 0 & 4 \times 5 \\ 3 \times 6 & 3 \times 7 & 4 \times 6 & 4 \times 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

One important piece of notation to take into account is $\otimes$, used to represent the equivalent of the sum (noted with $\sum$), but instead of addition the Kronecker product is used. In other words $\otimes$ denotes the Kronecker product of a finite number of terms. To clarify here's an example with the identity matrix:

With $\mathbb{I}$ as the matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and n as a power of 2:

$$\mathbb{I}_n = \bigotimes^{\log_2 n} \mathbb{I}$$

Here is the case for $n = 8$:

$$\mathbb{I}_8 = \bigotimes^{\log_2 8} \mathbb{I} = \bigotimes^{3} \mathbb{I} = \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The Kronecker product also works for vectors in the same way, with a scalar-vector multiplication:

For the vectors $|\psi\rangle$ and $|\varphi\rangle$ of dimensions n and m respectively:

$$|\psi\rangle \otimes |\varphi\rangle = \begin{bmatrix} \psi_1 \, |\varphi\rangle \\ \psi_2 \, |\varphi\rangle \\ \vdots \\ \psi_m \, |\varphi\rangle \end{bmatrix} = \begin{bmatrix} \psi_1 \varphi_1 \\ \psi_1 \varphi_2 \\ \vdots \\ \psi_1 \varphi_m \\ \vdots \\ \vdots \\ \psi_n \varphi_1 \\ \psi_n \varphi_2 \\ \vdots \\ \psi_n \varphi_m \end{bmatrix}$$

Note that the Kronecker product can also be taken between a vector and a matrix or vise-versa, however this form isn't as common as the other two.

2.1.4.1 Properties of the Tensor Product The basic properties of the tensor product are as follows [15, 16]:

1. Associativity:

$$A \otimes (B + C) = A \otimes B + A \otimes C$$
$$(zA) \otimes B = A \otimes (zB) = z(A \otimes B)$$
$$(A \otimes B) \otimes C = A \otimes (B \otimes C)$$
$$A \otimes 0 = 0 \otimes A = 0$$

2. Non-commutative [9]:

$$A \otimes B \neq B \otimes A$$

2.1.5 Trace

The trace of a matrix is just the sum of the elements on the main diagonal, the one that goes from top to bottom and left to right.

Here's a matrix A with its main diagonal marked:

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

And its trace, denoted by $\mathrm{Tr}[A]$ is:

$$\mathrm{Tr}[A] = 1 + 1 + 1 = 3$$

More formally, the trace of a n-dimensional squared matrix A is:

$$\mathrm{Tr}[A] = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}$$

The trace of a matrix as the following properties:

---

[9]One cool thing is that $A \otimes B$ and $B \otimes A$ are permutation equivalent: $\exists P, Q \Rightarrow A \otimes B = P(B \otimes A)Q$ where P and Q are permutation matrices.

1. Linear operator:
   Because the trace is a linear mapping, it follows that:
   $\mathrm{Tr}[A + B] = \mathrm{Tr}[A] + \mathrm{Tr}[B]$ and $\mathrm{Tr}[zA] = z\,\mathrm{Tr}[A]$, for all squared matrices A and B and all scalars z.

2. Trace of a Kronecker product:
   $\mathrm{Tr}[A \otimes B] = \mathrm{Tr}[A]\,\mathrm{Tr}[B]$

3. Transpose has the same trace:
   $\mathrm{Tr}[A] = \mathrm{Tr}\left[A^T\right]$

4. Trace of a product is cyclic:
   For a $m \times n$ matrix A and a $n \times m$ matrix B:
   $\mathrm{Tr}[AB] = \mathrm{Tr}[BA]$

One very useful way to compute the trace of an operator is through the Gram-Schmidt procedure[10] and an outer product. Using Gram-Schmidt to represent the unit vector $|\psi\rangle$ with an orthonormal basis $|i\rangle$ which includes $|\psi\rangle$ as the first element, is true that:

$$\mathrm{Tr}[A\,|\psi\rangle\,\langle\psi|] = \sum_i \langle i|\,A\,|\psi\rangle\,\langle\psi|i\rangle = \langle\psi|\,A\,|\psi\rangle$$

2.2 Quantum Computation

After some amount of math theory, it is time to start talking about quantum mechanics, in this chapter I am going to introduce the basic concepts of quantum computation and quantum information.

Quantum mechanics is a mathematical framework or rather a set of theories used to describe and explain the physical properties of atoms, molecules, and subatomic particles. It is the framework of all quantum physics including quantum information science. The right way of presenting quantum computation is through the formal quantum mechanics postulates because, with them, the statements made in quantum computation do not seem to come from anywhere [17]. However, to not complicate this section more than it is, I will do my best to explain the concepts and math of quantum computing just on their own, without presenting more generalized concepts from quantum mechanics, unless it is totally necessary to do so.

---

[10]See A.1 for the definition of the Gram-Schmidt procedure.

2.2.1 Quantum State and Superpositions

To describe how physical systems evolution through time, you need to represent them in some way. On quantum computing those quantum physical systems are represented through quantum states, which are some form of probably distribution for the possible outcomes of a measurement on a quantum system [18].

Imagine that you have a pen, however, you do not know which color is it, but you know that it can be red or blue. To figure out the color, you would do a measurement, in other words, write something to see the color of the ink. You also know that there is a 50% change of being blue, and a 50% of being red[11]. So far you have your quantum system (the pen), a way to measured it (writing), and a list of all the possible outcomes (50% blue and 50% red). Now you need your quantum state, the way to represent everything mathematically. Let's try to store the information that we know about the pen in a vector.

If we put the probability of each possible outcome in an entry, we have the following vector:

$$\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Where the first entry is the probability of painting blue and the second one of painting red, to make it more easy let me color the numbers:

$$\begin{bmatrix} \color{blue}0.5 \\ \color{red}0.5 \end{bmatrix}$$

Note that this vector is normalized went using the ℓ_1 norm, defined as the sum of the entries of a vectors[12], in other words, the ℓ_1 norm of this vector is 1.

So, how can we extract the probability about a possible outcome from the vectors, by a mathematical operation; I mean. We how that we need to get a real number out of this operation, so let's try with an inner product. As we know a vector in a space is a linear combination of space basis:

$$\color{blue}0.5 \color{black}\ket{0} + \color{red}0.5 \color{black}\ket{1} = \begin{bmatrix} \color{blue}0.5 \\ \color{red}0.5 \end{bmatrix}$$

---

[11]This is starting to sound a lot like US politics.
[12]With $\ket{a}$ being a vector, the ℓ_1 norm, denoted as $\|\cdot\|_1$ is $\|\ket{a}\|_1 = \sum_i a_i$.

Where $|0\rangle$ is the vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and, $|1\rangle$ is the vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Thus to get the scalar, you take the inner product between the basis corresponding to that scalar and the vector, like so:

$$\langle 0|v\rangle = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = 0.5$$

$$\langle 1|v\rangle = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = 0.5$$

You can see that this is quite easy. For example, to represent a pen with 6 colors with a random possibility of writing in a specific color, a valid vector to represent this pen would be[13]:

$$|w\rangle = \begin{bmatrix} 0.25 \\ 0.3 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.05 \end{bmatrix}$$

Then we can measure the state with an inner product:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.25 \\ 0.3 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.05 \end{bmatrix} = 1$$

Note that we use the corresponding basis for the 3th element, the green number.

Leaving the pens aside, now we are going to substitute the pens by quantum systems, for example, with a photon. Photons have certain properties that can be measured, like polarization[14], when looking at photon as waves in the electromagnetic field, the polarization is the geometrical orientation of the oscillation of the wave. This can be interpreted as an angle respect to the propagation direction.

insert figure here

We can set the basis states of the polarization as vertical and horizontal, denoted with the vectors $|\rightarrow\rangle$ and $|\uparrow\rangle$, respectively. And we can set a state that is a

---

[13]The entries in the vector are colored for clarity, but don't take it to seriously.

[14]More concretely, its a property of traverse waves, which electromagnetic waves are.

superposition of both, denoted as $|\nearrow\rangle$ [19]. The superposed state will be:

$$|\nearrow\rangle = \alpha |\rightarrow\rangle + \beta |\uparrow\rangle$$

Where α and β are complex numbers. The superpose state simply is an angle that is not 0 nor $\frac{\pi}{2}$. Note that we do not have to worry about the exact math of all these vectors[15], because we are only worry about the information that they carry. Which is taken through the measurements, like with the pen. To know[16] the state of polarization, e.i. to measure it, you need to pass the photon through several polarizing filters, which let a photon pass completely through it or absorbed completely on a probabilistic matter.

Let me explain, what the filter does is to collapse the photon in two possible states of polarization in either the state on which the filter is oriented or on the state that is perpendicular to it. For example, a filter oriented in the horizontal direction will either pass a photon oriented vertically or block entirely. The matter on which the collapse is decided is probabilistic, a photon oriented on the horizontal direction has a 100% probability of being absorbed, while a photon polarized vertically has a 0% chance of being absorbed, and finally, a photon that has a $\frac{\pi}{4}$ angle of polarization respect to the filter, meaning that is exactly in between these two perpendicular states, has a 50% probability of being absorbed by the filter, thus it has a 50% chance of passing through.

However, the polarization of the photons are a concrete physical system, in quantum information everyone works with qubits, instead of physical systems[17]. Went presenting algorithms for example, researchers use qubits.

2.2.2 Qubits

Modern computers represent information through strings of zeros and ones named bits. We encode and decode bits using binary code, through standards like UTF-8 [20] that can encode any character from the Unicode standard, that is all letters and emojis that exist.

---

[15]The mathematical description of the polarization of the photon is not necessary to know went talking about quantum computation, unless you are working with a quantum computer that represents the quantum states as photons (using their polarization or something else.). Then you would be talking about quantum optics which is itself a complex part of quantum mechanics, that I do not want to learn about yet.

[16]Kind of, you will see later.

[17]Unless you are working with a concrete quantum system, and you want to represent the information in the system's terms.

2.2.3 Quantum Measurement

Turning back to the photons and their polarization, to predict on which basis the photon is going to collapse, we need to employ quantum measurements. On this case to know the probability of the collapse to a specific basis is enough.

Where the vector $|\rightarrow\rangle$ expresses the polarization of the filter and $|\uparrow\rangle$ its perpendicular complement. A photon that is polarized in the following state [19]:

$$|\nearrow\rangle = \alpha\,|\rightarrow\rangle + \beta\,|\uparrow\rangle$$

The probability of being absorbed is α^2, note that α is the coefficient that corresponds to the basis $|\rightarrow\rangle$, which is the polarization of the filter. In other words, α^2 indicates the probability of the collapse of the polarization onto the state $|\rightarrow\rangle$.

The coefficients α and β can be expressed as function of an angle θ[18]:

$$|\nearrow\rangle = \cos\theta\,|\rightarrow\rangle + \sin\theta\,|\uparrow\rangle \tag{2.4}$$

And the expected probabilities become $\cos^2\theta$ and $\sin^2\theta$, respectively [19]. Now we can prove the previous statements. In the case of a $\pi/2$ angle we see that:

$$p(|\rightarrow\rangle) = \cos^2\theta = \cos^2\frac{\pi}{2} = \left(\frac{1}{\sqrt{2}}\right)^2$$
$$= 0.5$$
$$p(|\uparrow\rangle) = \sin^2\theta = \sin^2\frac{\pi}{2} = \left(\frac{1}{\sqrt{2}}\right)^2$$
$$= 0.5$$

---

[18]For more information on the polarization of a photon, see appendix C.

Chapter 3

Experimental Work

Chapter 4

Conclusions

Appendices

Appendix A

More Linear Algebra

I have wrote many pages of linear algebra theory, but that wasn't enough. So here we go, I guess.

A.1   Gram–Schmidt Procedure

The Gram–Schmidt procedure is a method used to produce orthonormal basis for a vectors space [10]. For a d-dimensional inner product vector space V with a basis vectors set $|v_1\rangle, \cdots, |v_d\rangle$, we can define a new orthonormal basis set $\{|u\rangle\}$. The first element of that set is $|u_1\rangle = |v_1\rangle / \||v_1\rangle\|$, with the following element $|v_{k+1}\rangle$ being:

$$|u_{k+1}\rangle = \frac{|v_{k+1}\rangle - \sum_{i=1}^{k} \langle u_i|v_{k+1}\rangle |u_i\rangle}{\left\||v_{k+1}\rangle - \sum_{i=1}^{k} \langle u_i|v_{k+1}\rangle |u_i\rangle\right\|}$$

For k in the interval $1 \leq k \leq d-1$.

If we follow the above for each k in $1 \leq k \leq d-1$, we obtain the new vector set $|u_1\rangle, \cdots, |u_d\rangle$ that is a valid orthonormal basis for the space V. The created vector set must have the same span[1] as the previous one to be a valid basis for the space V:

$$\text{span}(\{|v\rangle\}) = \text{span}(\{|v\rangle\}) = V$$

---

[1]The span of a set of vectors is all the possible linear combinations that come from those vectors.
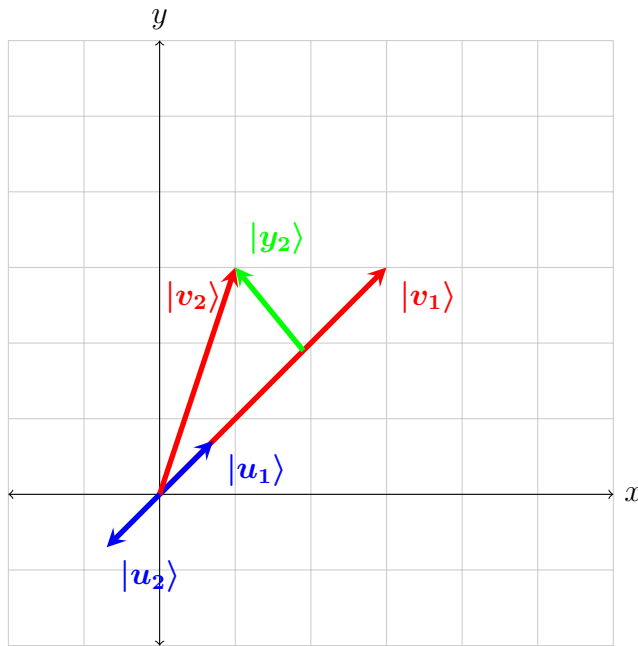
Note that the span of the basis set is the definition of the space. In other words, every vector in V can be represented as a linear combination of the basis vectors.

To get at some intuition let's look at the case for $k = 1$:

$$|u_2\rangle = \frac{|v_2\rangle - \langle u_1|v_2\rangle\, |u_1\rangle}{\| |v_2\rangle - \langle u_1|v_2\rangle\, |u_1\rangle \|}$$

We can see that the vector $|u_2\rangle$ is defined by the subtraction of the vector $|v_2\rangle$ by the projection[2] of $|v_2\rangle$ onto $|v_1\rangle$. Which is equivalent for the projection of $|v_2\rangle$ onto $|u_1\rangle$, remember that $|u_1\rangle$ is just $|v_1\rangle$ normalized. Therefore we have: _____ finish the graph

The proof that is an orthonormal basis is quite simple: We can see immediately that the components of $\{|u\rangle\}$ are unit vectors because they are normalized (the vectors $|v_{k+1}\rangle - \sum_{i=1}^{k} \langle u_i|v_{k+1}\rangle\, |u_i\rangle$ are divided by their norm). And we can se that they are orthogonal by checking that the inner product of non-equal vectors in the set is 0:

For $k = 1$:

$$|u_2\rangle = \frac{|v_2\rangle - \langle u_1|v_2\rangle\, |u_1\rangle}{\| |v_2\rangle - \langle u_1|v_2\rangle\, |u_1\rangle \|}$$

---

[2]The way to project the vector $|a\rangle$ onto $|b\rangle$ is: $\langle b|a\rangle\, |b\rangle$.

Then the inner product with $|v_1\rangle$ is:

$$\langle u_1|u_2\rangle = \langle u_1| \left(\frac{|v_2\rangle - \langle u_1|v_2\rangle\,|u_1\rangle}{\||v_2\rangle - \langle u_1|v_2\rangle\,|u_1\rangle\|} \right)$$

$$= \frac{\langle u_1|v_2\rangle - \langle u_1|v_2\rangle\,\langle u_1|u_1\rangle}{\||v_2\rangle - \langle u_1|v_2\rangle\,|u_1\rangle\|}$$

$$= 0$$

By induction we can see that for $j \leq d$, with d being the dimension of the vector space:

$$\langle u_j|u_{n+1}\rangle = \langle u_j| \left(\frac{|v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,|u_i\rangle}{\||v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,|u_i\rangle\|} \right)$$

$$= \frac{\langle u_j|v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,\langle u_j|u_i\rangle}{\||v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,|u_i\rangle\|}$$

$$= \frac{\langle u_j|v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,\delta_{ij}}{\||v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,|u_i\rangle\|}$$

$$= \frac{\langle u_j|v_{n+1}\rangle - \langle u_j|v_{n+1}\rangle}{\||v_{n+1}\rangle - \sum_{i=1}^{n} \langle u_i|v_{n+1}\rangle\,|u_i\rangle\|}$$

$$= 0$$

All of this doesn't seem straightforward at first but remember that the inner product of two orthogonal vectors is zero, and the inner product between the same unit vector is one.

A.2 Dirac Notation Crash Course

On the following table there is a quick summary important mathematical concepts of linear algebra [21] expressed in Dirac Notation[3].

---

[3]The notation used for the complex vector spaces and complex number space are not standard Dirac Notation, but I included them in the table to explain what they mean.

Notation	Description
z	Complex number
z^*	Complex conjugate of the a complex number z. $(a+bi)^* = (a-bi)$
$\lvert\psi\rangle$	Vector with label ψ. Known as *ket*
$\lvert\psi\rangle^T$	Transpose of vector $\lvert\psi\rangle$
$\lvert\psi\rangle^\dagger$	Hermitian conjugate of vector. $\lvert\psi\rangle^\dagger = (\lvert\psi\rangle^T)^*$
$\langle\psi\rvert$	Dual vector to $\lvert\psi\rangle$. $\lvert\psi\rangle = \langle\psi\rvert^\dagger$ and $\langle\psi\rvert = \lvert\psi\rangle^\dagger$. Known as *bra*
$\langle\varphi\lvert\psi\rangle$	Inner product of vectors $\langle\varphi\rvert$ and $\lvert\psi\rangle$
$\lvert\varphi\rangle\langle\psi\rvert$	Outer product of vectors $\langle\varphi\rvert$ and $\lvert\psi\rangle$
$\lvert\psi\rangle \otimes \lvert\varphi\rangle$	Tensor product of vectors $\lvert\varphi\rangle$ and $\lvert\psi\rangle$
0	Zero vector and zero operator
$\mathbb{I}_n$	Identity matrix of dimension n
$\mathbb{C}_n$	Complex vector space of dimension n
$\mathbb{C}_1$ or $\mathbb{C}$	Complex number space

A.3 More on the Partial Trace

Appendix B

Quantum Computation vs Quantum Mechanics

On the introduction I mentioned that one of the reasons for with I started learning and researching quantum computing is because it is easy, on this appendix I am going to present why this is the case with a practical example.

On quantum mechanics the more general way to represent quantum state, like the orbitals of an atom of hydrogen, are wavefunctions, not statevectors. Wavefunctions are extremely useful, however, working with them adds a hole new level of complexity because they are continuous functions that depend on time. Compare them with vectors, which are discrete packets of information that evolve through discrete amounts of time.

B.1   Normalizing

Because the probabilistic interpretation of both statevectors and wavefunctions, these two objects have to be normalize upon measurement, thus making the total sum of probabilities 1. How to normalize these objects is a perfect example to illustrate the difference in complexity that I see between quantum mechanics and quantum computation.

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi$$

Figure B.1: **Schrödinger's Equation**. Where $\hbar$ is $h/2\pi$, and V is the potential energy function.

For a wavefunction Ψ that represents a particle, the probability of finding the particle in a point x is $|\Psi(x,t)|^2$. Then, the wavefunction has to be normalized like the following:

$$\int_{-\infty}^{+\infty} |\Psi(x,t)|^2 \, dx = 1 \tag{B.1}$$

Since the wavefunctions evolves over time with the Schrödinger equation, see Fig.B.1, any solution this equation has to be normalized. Luckily, if the wavefunction is normalized at time $t = 0$ it stays this way, in other words, the Schrödinger equations preserves the normalization of the wavefunction [22].

We can prove that the equation, preserves B.1, starting from the trivial equality:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{-\infty}^{+\infty} |\Psi(x,t)|^2 \, dx = \frac{\partial}{\partial t}\int_{-\infty}^{+\infty} |\Psi(x,t)|^2 \, dx$$

By the product rule we have that[1]:

$$\frac{\partial}{\partial t}|\Psi|^2 = \frac{\partial}{\partial t}(\Psi\Psi^*) = \Psi^*\frac{\partial\Psi}{\partial t} + \frac{\partial\Psi^*}{\partial t}\Psi$$

Now the Schrödinger equation says that

$$\frac{\partial\Psi}{\partial t} = \frac{i\hbar}{2m}\frac{\partial^2\Psi}{\partial x^2} - \frac{i}{\hbar}V\Psi$$

then by taking the complex conjugate we have that

$$\frac{\partial\Psi^*}{\partial t} = -\frac{i\hbar}{2m}\frac{\partial^2\Psi^*}{\partial x^2} + \frac{i}{\hbar}V\Psi^*$$

so

$$\frac{\partial}{\partial t}|\Psi|^2 = \frac{i\hbar}{2m}\left(\Psi^*\frac{\partial^2\Psi}{\partial x^2} - \frac{\partial^2\Psi^*}{\partial x^2}\Psi\right) = \frac{\partial}{\partial x}\left[\frac{i\hbar}{2m}\left(\Psi^*\frac{\partial\Psi}{\partial x} - \frac{\partial\Psi^*}{\partial x}\Psi\right)\right]$$

finally we can evaluate the integral from the start:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{-\infty}^{+\infty} |\Psi(x,t)|^2 \, dx = \frac{i\hbar}{2m}\left(\Psi^*\frac{\partial\Psi}{\partial x} - \frac{\partial\Psi^*}{\partial x}\Psi\right)\Big|_{-\infty}^{+\infty}$$

---

[1]From now on I am going to write $\Psi(x,t)$ as simply Ψ to not make the equations to complicated.

Because $\Psi(x,t)$ has to go to zero when x goes to either infinity, is true that:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{-\infty}^{+\infty} |\Psi(x,t)|^2 \, dx \; = 0$$

Thus the integral is constant and went Ψ is normalized at $t = 0$, it stays that way for any t (positive of course).

falta la prueba para la normalización de un statevector

Appendix C

Polarization of a photon

On equation 2.4 I have excluded the concept of phase, which determinates the type of polarization that the photon has. There are three types of polarization:

1. **Linear:** A photon is linear polarized when the phase angles α_x, α_y of both basis $|x\rangle$, $|y\rangle$ states are equal:

$$|\nearrow\rangle = \cos(\theta)e^{i\alpha_x}|x\rangle + \cos(\theta)e^{i\alpha_y}|y\rangle$$
$$= [\cos(\theta)|x\rangle + \sin(\theta)|y\rangle]e^{i\alpha}$$

   Where $\alpha = \alpha_x = \alpha_y$.

2. **Circular:** Where the angles α_x, α_y are apart exactly $\frac{\pi}{2}$ and the amplitude of both basis states is the same:

$$|\nearrow\rangle = \frac{1}{\sqrt{2}}\cos(\theta)e^{i\alpha_x}|x\rangle \pm i\frac{1}{\sqrt{2}}\sin(\theta)e^{i\alpha_y}|y\rangle$$
$$= [\cos(\theta)e^{i\alpha_x}|x\rangle \pm i\sin(\theta)e^{i\alpha_y}|y\rangle]\frac{1}{\sqrt{2}}$$

   Where the $\pm$ sign indicates the difference between right and left circular polarization, $+$ and $-$, respectively.

3. **Elliptical:** Where just the phase angles α_x, α_y differ in some amount:

$$|\nearrow\rangle = \cos(\theta)e^{i\alpha_x}|x\rangle + \sin(\theta)e^{i\alpha_y}|y\rangle$$

   This is the more general case.

Bibliography

[1] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13:2567–2586, 2014. [Online; accessed 19/09/2021: Link].

[2] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J. Martinez, Jae Hyeon Yoo, Sergei V. Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, Evan Peters, Owen Lockwood, Andrea Skolik, Sofiene Jerbi, Vedran Dunjko, Martin Leib, Michael Streif, David Von Dollen, Hongxiang Chen, Shuxiang Cao, Roeland Wiersema, Hsin-Yuan Huang, Jarrod R. McClean, Ryan Babbush, Sergio Boixo, Dave Bacon, Alan K. Ho, Hartmut Neven, and Masoud Mohseni. Tensorflow quantum: A software framework for quantum machine learning. 2021. [Online; accessed 19/09/2021: Link].

[3] IBM. Ibm quantum experience, 2021. [Online; accessed 19/09/2021: Link].

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2014. [Online; accessed 19/09/2021: Link].

[5] He-Liang Huang, Yuxuan Du, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, and et al. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16(2), 2021. [Online; accessed 19/09/2021: Link].

[6] Gilbert Strang. Linear algebra mit ocw, 2011. [Online; accessed 19/09/2021: Link].

[7] Gilbert Strang. Matrix methods in data analysis, signal processing, and machine learning mit ocw, 2018. [Online; accessed 19/09/2021: Link].

[8] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambrindge University Press, 10 edition, 2010.

[9] Eleanor Rieffel and Wolfgang Polak. *Quantum Computing: A Gentle Introduction*. MIT Press, 1 edition, 2011.

[10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*, chapter 2, page 66. Cambrindge University Press, 10 edition, 2010.

[11] Sheldon Axler. *Linear Algebra Done Right*, chapter 3, pages 37–58. Springer, 2 edition, 1997.

[12] Sheldon Axler. *Linear Algebra Done Right*, chapter 3, pages 48–52. Springer, 2 edition, 1997.

[13] Wolfram MathWorld. L2-norm, 2021. [Online; accessed 19/09/2021: Link].

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*, chapter 2, page 74. Cambrindge University Press, 10 edition, 2010.

[15] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*, chapter 2, page 73. Cambrindge University Press, 10 edition, 2010.

[16] Wikipedia. Tensor product, 2021. [Online; accessed 19/09/2021: Link].

[17] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*, chapter 2, pages 80–96. Cambrindge University Press, 10 edition, 2010.

[18] Asher Peres. *Quantum Theory: Concepts and Methods*, chapter 2, pages 24–29. Kluwer Academic Publishers, 2002.

[19] Eleanor Rieffel and Wolfgang Polak. *Quantum Computing: A Gentle Introduction*, chapter 2, pages 12–13. MIT Press, 1 edition, 2011.

[20] Wikipedia. Utf-8, 2021. [Online; accessed 19/09/2021: Link].

[21] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*, chapter 2, page 62. Cambrindge University Press, 10 edition, 2010.

[22] David J. Griffiths. *Introduction to Quantum Mechanics*, chapter 1, pages 11–12. Prentice Hall, 1995.