

Computación Cuántica Profesional desde Cero

Cuántica para no científicos

Alejandro Mata Ali, Daniel García Guijo

Ricardo Quispe Mendizábal, Aitor Moreno de Leceta

QuantumQuipu & Ayesa

0. Introducción al libro

0.1 ¿Por qué computación cuántica?

Desde hace décadas, la humanidad se ha dado cuenta de la gran potencia que tienen las nuevas tecnologías asociadas a la computación. Gracias a ello ha podido crear nuevas soluciones capaces de explorar nuevos horizontes, tanto en la mejora de la industria, como en la investigación básica, como a nivel usuario.

Sin embargo, el afán de obtener mejores soluciones nos ha llevado a una barrera de la computación clásica tradicional, con problemas tan grandes y complejos que su resolución sería tan costosa que tendríamos que esperar edades del universo o consumir niveles de memoria infinitos. Estos problemas van desde algo tan complicado como la simulación de sistemas con muchas dimensiones hasta obtener la forma óptima de almacenar paquetes en un contenedor. Estos problemas intentan muchas veces resolverse de forma aproximada con nuevos algoritmos, como los algoritmos genéticos o algoritmos heurísticos. Aun así, tienen diversas limitaciones, como depender de heurísticas o el poder quedarnos con soluciones subóptimas.

Aquí surge la idea de la computación cuántica. Los sistemas cuánticos no solo son capaces de almacenar cantidades exponenciales de información eficientemente, sino que además son capaces de realizar operaciones extremadamente complejas y costosas clásicamente aprovechando propiedades cuánticas como el entrelazamiento. Esto permite que ciertos problemas que llevarían millones de años resolver solo lleven unos minutos.

Esto nos lleva inevitablemente a problemas de seguridad de la información, ya que los métodos de ciberseguridad se mantienen sobre la idea de que llevaría demasiado tiempo encontrar las combinaciones correctas. Pero como vimos, con la computación cuántica esto podría ser posible. Por ello mismo surgen la Distribución de Claves Cuánticas (Quantum Key Distribution QKD) y la Criptografía Cuántica (Quantum Cryptography). Estas se aprovechan de propiedades de los sistemas cuánticos, como el teorema de no clonación, para poder proteger información y mensajes de atacantes externos.

No obstante, debido a la condición actual de los ordenadores cuánticos, los cuales siguen siendo muy pequeños, terriblemente ruidosos y con errores, han surgido otras tecnologías, las Quantum-Inspired, inspiradas en procesos cuánticos que permiten mediante técnicas clásicas o con

otros sistemas cuánticos realizar nuevos algoritmos notablemente más eficientes que sus homólogos clásicos aprovechando propiedades de los sistemas cuánticos a los cuales imitan.

Toda esta carrera va a revolucionar el mundo y la industria los próximos años, por lo que es importante prepararse para cuando llegue.

0.2 Cómo usar este libro

El objetivo principal de este libro es poder acercar a cualquier persona las claves de los algoritmos cuánticos, independientemente de sus conocimientos previos en matemáticas, física o ciencias en general, de forma que puedan dedicarse de manera profesional a la computación cuántica. Lo hemos diseñado para que sea lo más autocontenido posible de forma que pueda adentrarse en este mundo una persona que haya cursado la educación secundaria. Las bases matemáticas necesarias son explicadas en la primera parte del libro, la cual no es totalmente necesaria para lectores ya versados en la matemática planteada. Además, explicaremos con un detalle importante tanto los conceptos como las operaciones aplicadas.

Un lector novel sin conocimiento alguno debería empezar por la parte 1, por la introducción matemática para poder comprender los conceptos básicos que van a ser usados posteriormente. Tras ello, consideraremos que tiene una base matemática suficiente para equiparse con un estudiante de física.

En este nivel, el siguiente paso es el capítulo de la introducción física, que hará que entendamos los elementos básicos que utilizaremos posteriormente. Esta es la conexión con el mundo físico y la interpretación matemática, por lo que debería ser consultado por cualquiera que venga fuera del mundo de la cuántica. A partir de aquí, el conocimiento útil es cercano al obtenido por un graduado en física, especializado en cuántica, o alguien que haya participado en cursos y workshops al respecto. El siguiente capítulo será un capítulo de matemática más avanzada al que llamaremos posteriormente cuando sea necesario un conocimiento más amplio, así que puede ser omitido por ahora.

El siguiente paso será ir a la parte 2, en la que exploraremos los primeros algoritmos introductorios con los que nos familiarizaremos con la matemática y la forma de pensar cuántica. Empezaremos con las puertas cuánticas, nuestra pieza clave, continuaremos con algunos ejemplos de circuitos para crear estados e iremos ya al primer caso útil, la teleportación cuántica. Tras ello, veremos técnicas para medir valores esperados o aplicar ciertos operadores diferentes a nuestros estados. Acabaremos este bloque explicando cómo medir fases de operadores y cómo crear funciones cuánticas, los oráculos. Algunas de estas son las herramientas base con las que trabajaremos en las siguientes partes, mientras que otros son casos que utilizaremos para interiorizar conceptos. Con esto, tendríamos un conocimiento similar al de un iniciado en la computación cuántica.

Nuestro próximo objetivo será la parte 3, donde ya empezaremos con algoritmos útiles utilizando lo aprendido en la parte 2, introduciendo más herramientas. Estos algoritmos nos permitirán buscar en bases de datos, resolver sistemas de ecuaciones lineales, obtener el periodo de funciones o resolver problemas de optimización combinatoria. Con ello ya tendremos unos conocimientos dignos de alguien que empieza a trabajar en computación cuántica y a desarrollar algoritmos.

Llegados aquí, vamos a la parte 4, donde aprenderemos algoritmos más avanzados, para optimización y quantum machine learning. Este nivel ya es el más avanzado y con ello ya podríamos desarrollar proyectos.

Ahora exploraremos la seguridad cuántica en la parte 5, donde miraremos protocolos como el BB84 para Quantum Key Distribution y daremos unos toques a la criptografía cuántica. Con esto podremos tener un primer contacto con este campo.

Finalmente, tendremos la parte 6 optativa, referente a temas extra. El primero son las tecnologías Quantum-Inspired, donde destacamos el Quantum Annealing, que ya tiene resultados actuales, y las tensor networks, que marcan una línea importante no solo en el apartado Quantum-Inspired.

El segundo tema será todo el tema relacionado con los ordenadores reales, como la computación cuántica topológica, los tipos de ordenadores cuánticos, la corrección de errores y la Cuántica de escala intermedia ruidosa (noisy intermediate-scale quantum NISQ). Como último, analizaremos posibles casos de uso y de negocio para todo esto que hemos aprendido anteriormente.

Con estos conocimientos esperamos que el lector pueda pasar de ser un profano en la computación cuántica a poder tener una visión clara de cómo funciona y de cómo desarrollar algoritmos propios o utilizar los ya existentes para diferentes casos.

Contents

0	Introducción al libro	3
0.1	¿Por qué computación cuántica?	3
0.2	Cómo usar este libro	4

I

Parte 1: Introducción teórica

1	Introducción matemática	15
1.1	Conceptos previos	15
1.1.1	Símbolos básicos	15
1.1.2	Números	16
1.1.3	Índices	16
1.1.4	Sumatorio	17
1.1.5	Productorio	19
1.1.6	Mapeos y funciones	20
1.1.7	Gráficas	21
1.2	Números complejos	23
1.2.1	Unidad imaginaria	23
1.2.2	Número complejo	23
1.3	Vectores	26
1.3.1	Definición	26
1.3.2	Vectores base	27
1.3.3	Producto escalar	29
1.4	Matrices	31
1.4.1	Definición	31
1.4.2	Productos matriciales	32

1.4.3	Interpretación geométrica	34
1.4.4	Características matriciales	34
1.4.5	Matrices relevantes	35
1.4.6	Matriz inversa	37
1.4.7	Matriz unitaria	37
1.4.8	Matriz hermítica	37
1.4.9	Matrices de Pauli	38
1.5	Tensores	40
1.5.1	Definición	40
1.5.2	Operaciones	40
1.5.3	Tensores importantes	42
2	Introducción física	45
2.1	Estados físicos	45
2.2	El qubit: superposición y probabilidad	45
2.3	Ejemplos físicos de qubits	45
2.4	Puertas cuánticas y operadores unitarios	45
2.5	Estados cuánticos de varios qubits	45
2.6	Entrelazamiento	45
2.7	Valores esperados y operadores hermíticos	45
2.8	Puertas multiqubit	45
2.9	Medida y colapso del estado	45
2.10	Errores cuánticos	45
3	Matemática avanzada	47
3.1	Álgebra modular básica	47
3.2	Álgebra de Lie básica	47
3.3	Cálculo diferencial básico	47
3.4	Operadores no unitarios	47
3.5	Cuantización	47
3.6	Matrices densidad	47
3.7	Proyecciones	47
3.8	Problemas QUBO	47
3.9	Problema de Ising	47

II

Parte 2: Algoritmos cuánticos

4	Algoritmos introductorios	51
4.1	Explorando las puertas	51
4.2	Creando entrelazamiento	51
4.3	Teleportación cuántica	51
4.4	Obteniendo información colapso	51

4.5	Aplicando operadores	51
4.6	Obteniendo fases	51
4.7	Oráculos y superposiciones	51
5	Algoritmos intermedios	53
5.1	Buscando datos con Grover	53
5.2	Resolviendo sistemas de ecuaciones	53
5.3	Obteniendo periodos	53
5.4	Aproximaciones variacionales	53
5.5	Optimización de sistemas	53
5.6	Evolución en tiempo imaginario	53
6	Algoritmos avanzados	55
6.1	Optimización combinatoria	55
6.2	Factorización de números	55
6.3	Quantum Singular Value Decomposition	55
6.4	Quantum Support Vector Machine	55
6.5	Quantum Neural Networks	55
6.6	Quantum Autoencoder	55

III

Parte 3: Seguridad Cuántica

7	Distribución de claves cuánticas	59
7.1	Motivación	59
7.2	Teorema de no clonación	59
7.3	Protocolo BB84	59
7.4	Protocolo E91	59
7.5	Decoy State	59
7.6	Otros protocolos	59
8	Criptografía cuántica	61
8.1	Motivación	61
8.2	Dinero Cuántico	61

IV

Parte 4: Computación Cuántica avanzada

9	Ordenadores cuánticos	67
9.1	Criterios de DiVincenzo	67
9.2	Qubits físicos	68
9.2.1	Superconductores	68
9.2.2	Átomos neutros	68

9.2.3	Iones atrapados	68
9.2.4	Fotones	68
9.2.5	Computación cuántica topológica	68
9.3	NISQ	68
9.3.1	Computación híbrida	68
9.4	Qubits lógicos	68
9.4.1	Corrección de errores	68
10	Tecnologías Quantum-Inspired	69
10.1	Quantum Annealing	69
10.2	Tensor Networks	69
11	Casos de uso	71
12	Out	73
	Bibliography	75
	Books	75
	Articles	75

Parte 1: Introducción teórica

1 Introducción matemática 15

- 1.1 Conceptos previos
- 1.2 Números complejos
- 1.3 Vectores
- 1.4 Matrices
- 1.5 Tensores

2 Introducción física 45

- 2.1 Estados físicos
- 2.2 El qubit: superposición y probabilidad
- 2.3 Ejemplos físicos de qubits
- 2.4 Puertas cuánticas y operadores unitarios
- 2.5 Estados cuánticos de varios qubits
- 2.6 Entrelazamiento
- 2.7 Valores esperados y operadores hermíticos
- 2.8 Puertas multiqubit
- 2.9 Medida y colapso del estado
- 2.10 Errores cuánticos

3 Matemática avanzada 47

- 3.1 Álgebra modular básica
- 3.2 Álgebra de Lie básica
- 3.3 Cálculo diferencial básico
- 3.4 Operadores no unitarios
- 3.5 Cuantización
- 3.6 Matrices densidad
- 3.7 Proyecciones
- 3.8 Problemas QUBO
- 3.9 Problema de Ising

En esta parte del libro abordaremos los conocimientos básicos necesarios para poder empezar a aprender acerca de los diferentes algoritmos cuánticos y las matemáticas que los rodean. Empezaremos con la matemática básica, que será el lenguaje con el que expresaremos todas las ideas y realizaremos cálculos, continuaremos con una introducción física a los conceptos de la computación cuántica como el qubit, las puertas cuánticas y la superposición. Finalmente terminaremos con un capítulo complementario de matemáticas más avanzadas a las que iremos refiriendo en los diferentes algoritmos.

Esta introducción es vital para cualquiera desconocedor tanto de las matemáticas como de la notación cuántica. Incluso siendo conocedor de las mismas, es recomendable repasar esta parte a fin de homogeneizar la notación para las posteriores partes.

1. Introducción matemática

Aunque pueda parecer una barrera para un profano en el campo, las matemáticas básicas utilizadas son muy simples, siendo la mayoría álgebra lineal con vectores, matrices, tensores y números complejos. Vamos darles un vistazo enfocándonos en las partes más relevantes para la computación cuántica sin perdernos en conceptos no útiles o complementarios.

Primero haremos una introducción de conceptos previos sencillos y luego vamos a empezar con los conceptos de álgebra lineal y los números complejos.

1.1 Conceptos previos

1.1.1 Símbolos básicos

Empezamos con los símbolos matemáticos básicos que necesitaremos posteriormente. Lo primero que recordaremos es que usualmente utilizaremos letras para nombrar ciertas cosas. Algunas letras suelen tener un significado constante, como π o Σ , pero normalmente su significado debe deducirse de la circunstancia en la que se estén usando. Por ejemplo, la letra i sirve para denotar el número $\sqrt{-1}$, pero también suele usarse para denotar la posición en una lista.

Todas las fórmulas y ecuaciones estarán formuladas siempre de manera que dos conceptos diferentes tengan nombres diferentes. Por ejemplo, si queremos decir i veces cierto elemento de una lista a , usaremos el subíndice j en vez de i para no confundirnos. Si el número i no aparece en la ecuación, podemos usar el subíndice i sin problema, ya que será evidente su significado.

Vamos a hacer una pequeña lista de expresiones que son relevantes:

- $\{a, b, c\}$: conjunto de elementos a , b y c .
- $[a, b]$: intervalo cerrado entre a y b . Los valores que hay entre a y b , ambos incluidos.
- (a, b) : intervalo abierto entre a y b . Los valores que hay entre a y b , ambos excluidos.
- $[a, b)$: intervalo semiabierto entre a y b . Los valores que hay entre a (incluido) y b (excluido).
- $a \in b$: a está contenido o es parte de b .
- $a \in [b, c]$: a está en el intervalo $[b, c]$.
- $a := b$: a se define como b .
- $a \Rightarrow b$: a implica b .
- $a \Leftrightarrow b$: a implica b y b implica a . a sí y solo sí b .

- $\forall a \in b$: para todos los a en b .
- $\exists a$: existe a .
- $a | b$: a tal que b . a que cumple b .

1.1.2 Números

Haremos un repaso de los tipos de números que hay y la notación que se usa para expresarlos.

Definition 1.1.1 — Números enteros. Los números enteros son básicamente los números que no tienen decimales. Estos son $-3, -2, -1, 0, 1, 2, 3, \dots$. Estos incluyen tanto los números positivos como los negativos y el cero.

Notation 1.1. Si un número a es entero, diremos que $a \in \mathbb{Z}$.

Definition 1.1.2 — Números naturales. Los números enteros positivos y el cero. Estos son $0, 1, 2, 3, \dots$

Notation 1.2. Si un número a es natural, diremos que $a \in \mathbb{N}$.

Definition 1.1.3 — Números racionales. Los números que se pueden expresar como la división entre dos números enteros. Están incluidos los números enteros, debido a que el denominador puede ser 1. También se pueden ver como cualquier número con un número de decimales finito o con unos decimales que se repitan a partir de algún punto. Estos son $\frac{1}{1}, \frac{2}{1}, \frac{1}{2}, \frac{1}{3}, \frac{3}{1}, \frac{4}{1}, \frac{3}{2}, \frac{2}{3}, \frac{1}{4}, \dots$

Notation 1.3. Si un número a es racional, diremos que $a \in \mathbb{Q}$.

Definition 1.1.4 — Números irracionales. Los números que no se pueden expresar como la división entre dos números enteros. También se pueden ver como cualquier número con un número de decimales infinito y que no se repitan en ningún momento. Estos son por ejemplo $\pi = 3.1416\dots, \sqrt{2} = 1.4142\dots, e = 2.7182\dots, \dots$

Notation 1.4. Si un número a es irracional, diremos que $a \in \mathbb{I}$. (Esta notación no es universal)

Definition 1.1.5 — Números reales. Los números con decimales. Incluyen a los racionales y a los irracionales.

Notation 1.5. Si un número a es real, diremos que $a \in \mathbb{R}$.

1.1.3 Índices

La notación de índices es algo relativamente sencillo y muy necesario para expresar de una manera compacta y comprensible las diferentes operaciones matemáticas necesarias para nuestros cálculos. Estos se basan en expresar una operación repetitiva de manera abstracta para poder comprimirla en un mismo término que va variando un cierto valor.

Un índice viene a ser una etiqueta que nos permite localizar o identificar cierto valor o elemento. Estos toman diversos valores enteros en un cierto rango, como posiciones en una lista. Vamos a ver un ejemplo.

Notation 1.6. Dado un conjunto ordenado de elementos a y un número natural i inferior o igual al número de elementos de a , llamaremos a_i al elemento i -ésimo del conjunto a . Esto es, el elemento que aparezca en la posición i en el conjunto a . i se denominará el **subíndice** (o índice a secas) del elemento.

■ **Example 1.1 — Índices en una lista.** Tenemos la lista a , cuyos elementos son $3, 5, 8, 4$. Normalmente diremos que $a = \{3, 5, 8, 4\}$. Por otra parte, también podemos decir que $a_1 = 3$, $a_2 = 5$, $a_3 = 8$ y $a_4 = 4$. Así su primer elemento es a_1 , el segundo es a_2 , el tercero es a_3 y el cuarto a_4 . De forma comprimida, podemos decir que a_i es el elemento i -ésimo de la lista a , siendo i el subíndice que indica la posición en la lista. Este índice es un número entero que va desde 1 hasta 4. Esto matemáticamente sería

$$a = \{a_i \mid a_i \in a \ \forall i \in [1, 4]\} \quad (1.1)$$

donde vemos que creamos la lista de forma compacta. Para programadores, esto puede verse como usar un bucle for iterando sobre i . ■

■ **Example 1.2 — Lista de elementos pares.** Si por ejemplo quisiéramos que los elementos fueran los números pares entre 2 y 10, sería

$$b = \{b_i \mid b_i = 2 \cdot i \ \forall i \in [1, 5]\} = \{2, 4, 6, 8, 10\} \quad (1.2)$$

donde vemos que el índice no va de 1 a 10, sino que va hasta la mitad y cada elemento viene dado por una fórmula dependiente del índice, esto es, de la posición en la lista. ■

Esta notación es muy importante para vectores, matrices y tensores, por lo que habrá que familiarizarse con ella. Es importante notar que a veces el hecho de que un nombre tenga subíndice no implica que esto sea un subíndice numérico, sino que puede ser parte del nombre. Por ejemplo, si tenemos N_0 y N_f , esto puede significar que N_0 es el número inicial de algo y N_f el número final de algo. En las siguientes secciones se entenderá mejor.

1.1.4 Sumatorio

El sumatorio es importante para comprimir las largas sumas de elementos que pueden aparecer de forma abstracta. Esta notación es extraordinariamente buena para los casos en los que hay una fórmula con índices que nos dice el valor de cada elemento que sumaremos. También cuando el número de elementos que sumamos es variable o cuando son muchos.

Notation 1.7. La suma (o sumatorio) de los elementos desde la posición N_0 hasta la posición N_f de un conjunto a se denominará como

$$S = \sum_{i=N_0}^{N_f} a_i. \quad (1.3)$$

El sumatorio de 1.7 está compuesto de la siguiente manera

$$\sum_{\substack{\text{índice}=\text{valor inicial} \\ \text{Valor final}}}^{\text{Elemento}_{\text{índice}}}. \quad (1.4)$$

En casos en los que el valor inicial o el valor final sean completamente conocidos o evidentes, es usual omitirlos.

Veremos unos cuantos ejemplos para acabar de comprenderlo.

■ **Example 1.3 — Suma de enteros.** Imaginemos que queremos sumar los números enteros entre 1 y 100. Esta suma sería $S = 1 + 2 + 3 + \dots + 100$. Podemos expresarla de forma más comprimida usando el sumatorio

$$S = \sum_{i=1}^{100} i, \quad (1.5)$$

que significa ‘los valores i para cada i desde 1 hasta 100’. Esto también es equivalente al factorial de 100, denotado como $100!$.

Si quisieramos sumar los números enteros desde el -3 al 24 tendríamos

$$S = \sum_{i=-3}^{24} i. \quad (1.6)$$

Vamos a poner un ejemplo más complejo para entender bien la parte del elemento a sumar.

■ **Example 1.4 — Suma de números pares.** Si queremos sumar los elementos pares desde el 2 hasta el 100, podemos hacer un sumatorio del 2 al 100 donde solo sumemos si tenemos un número par. Sin embargo, también podemos sumar los elementos de la lista de los números pares $b = \{2, 4, 6, \dots, 100\}$. La construcción de esta lista la vimos en (1.2), por lo que si el sumatorio sería

$$S = \sum_{i=1}^{50} b_i = \sum_{i=1}^{50} 2 \cdot i \quad (1.7)$$

Vemos que de esta forma hemos comprimido la fórmula y la hemos adaptado para nuestro caso particular de forma abstracta.

En este caso, como tenemos un factor común en todos los elementos, el 2, podemos extraerlo del sumatorio, de forma que

$$S = \sum_{i=1}^{50} (2 \cdot i) = 2 \sum_{i=1}^{50} i. \quad (1.8)$$

■ **Example 1.5 — Suma de fracciones.** Ahora queremos sumar todos los números del tipo $1/n$, siendo n un número entero. Esto es, $1/1 + 1/2 + 1/3 + \dots$. En este caso, nuestra fórmula para los elementos será $1/i$, por lo que el sumatorio será

$$\sum_{i=1}^N 1/i, \quad (1.9)$$

donde N es el número de elementos que vamos a querer sumar. Esto es, sumaremos desde 1 hasta $1/N$.

■ **Example 1.6 — Suma de potencias.** Vamos a sumar los elementos de un conjunto compuesto por un cierto número b elevado a la posición en el conjunto. Esto es, $a = \{b^1, b^2, b^3, \dots, b^N\}$. Su sumatorio será

$$\sum_{i=1}^N b^i. \quad (1.10)$$

Si tenemos dos sumatorios que tienen los mismos límites, podemos juntarlos. Por ejemplo,

$$\sum_{i=1}^{10} \pi^i + \sum_{i=1}^{10} i^2 = \sum_{i=1}^{10} (\pi^i + i^2), \quad (1.11)$$

donde es importante colocar los paréntesis para dejar claro que el sumatorio actúa sobre los dos términos. Lo mismo se puede realizar de manera inversa.

1.1.5 Productorio

El siguiente paso después de sumar es multiplicar, por lo que definiremos un equivalente al sumatorio, pero para los productos: el productorio. Este será usado para cuando, en vez de sumar N elementos, queremos multiplicar N elementos.

Notación 1.8. *El producto (o productorio) de los elementos desde la posición N_0 hasta la posición N_f de un conjunto a se denominará como*

$$P = \prod_{i=N_0}^{N_f} a_i. \quad (1.12)$$

La composición del productorio es exactamente la misma que la del sumatorio. Vamos a ver algunos ejemplos.

■ **Ejemplo 1.7 — Producto de enteros.** Imaginemos que queremos multiplicar los números enteros entre 1 y 100. Esta suma sería $S = 1 \times 2 \times 3 \times \dots \times 100$. Podemos expresarla de forma más comprimida usando el productorio

$$P = \prod_{i=1}^{100} i. \quad (1.13)$$

Si quisiéramos multiplicar los números enteros desde el -3 al 24 tendríamos

$$P = \prod_{i=-3}^{24} i. \quad (1.14)$$

■

Algo importante es que, si los elementos del productorio son multiplicaciones de números, podemos dividir el productorio en varios productorios. Veremos un caso en los siguientes ejemplos.

■ **Ejemplo 1.8 — Producto de números pares.** Si queremos multiplicar los elementos pares desde el 2 hasta el 100, podemos hacer un productorio del 2 al 100 donde solo sumemos si tenemos un número par. Sin embargo, también podemos multiplicar los elementos de la lista de los números pares $b = \{2, 4, 6, \dots, 100\}$. La construcción de esta lista la vimos en (1.2), por lo que si el productorio sería

$$P = \prod_{i=1}^{50} b_i = \prod_{i=1}^{50} (2 \cdot i) \quad (1.15)$$

Vemos que de esta forma hemos comprimido la fórmula y la hemos adaptado para nuestro caso particular de forma abstracta.

En este caso, como tenemos que cada elemento es el producto de 2 por i podemos hacer

$$P = \prod_{i=1}^{50} (2 \cdot i) = \prod_{j=1}^{50} 2 \cdot \prod_{i=1}^{50} i. \quad (1.16)$$

■

■ **Ejemplo 1.9 — Producto de productorios.** Si queremos multiplicar dos productorios y tienen los mismos límites, podemos unirlos en el mismo productorio. Por ejemplo,

$$P = \prod_{i=1}^{50} 3^i \cdot \prod_{j=1}^{50} (i+4) = \prod_{i=1}^{50} (3^i(i+4)). \quad (1.17)$$

■

1.1.6 Mapeos y funciones.

Los mapeos vienen a ser una relación entre dos conjuntos. Esto es, una máquina a la que le das un elemento de uno de los conjuntos como llave y te devuelve el elemento que le corresponde en el otro conjunto. El ejemplo más simple es el mapeo índice-conjunto para un conjunto a de elementos a_i . Este mapeo hace que dado un número del índice i recuperemos el valor a_i asociado.

Una versión más especializada es una función, la cual ya viene dada por una fórmula y suele ser graficable. Las funciones suelen expresarse con la forma $f(x)$, donde la función f devuelve un valor para un valor del x de entrada dado. A x la llamaremos la variable independiente.

Notación 1.9. Llamaremos función a un mapeo desde un conjunto o espacio A a un conjunto o espacio B . Para una función f con valores de entrada x , esta se notará como

$$f(x). \quad (1.18)$$

Es importante darnos cuenta de que para cada valor de x en el conjunto de entrada, hay un valor $f(x)$ asociado. Vamos a verlo con unos ejemplos.

■ **Example 1.10 — Función producto.** Vamos a hacer una función que multiplique el número de entrada por 2. Esta función se expresa como

$$f(x) = 2 \cdot x. \quad (1.19)$$

Para el valor $x = 3$, $f(x) = 2 \cdot 3 = 6$, por ejemplo.

Nuestra función también puede depender de dos (o más) variables independientes. Un ejemplo sería una función que multiplique dos números de entrada

$$f(x, y) = x \cdot y. \quad (1.20)$$

Para $x = 3$ e $y = 7$, $f(x, y) = 3 \cdot 7 = 21$.

Nuestra función también puede tener más de una salida, pudiendo arrojar un conjunto de valores.

■ **Example 1.11 — Función productos por parejas.** Vamos a hacer una función que multiplique cada pareja de valores de entrada. Esta función se expresa como

$$f(x, y, z) = \{x \cdot y, x \cdot z, y \cdot z\}. \quad (1.21)$$

Para los valores $x = 3, y = 2, z = 5$, $f(x, y, z) = \{3 \cdot 2, 3 \cdot 5, 2 \cdot 5\} = \{6, 15, 10\}$, por ejemplo.

Una función también puede expresarse con más de una fórmula, de manera que para diferentes conjuntos de valores de entrada, las fórmulas a aplicar sean diferentes.

■ **Example 1.12 — Función por partes.** Queremos hacer la función valor absoluto, que devolverá el valor de entrada sin el signo. Esta sería

$$f(x) = \begin{cases} x, & \text{si } x \geq 0 \\ -x, & \text{si } x < 0 \end{cases} \quad (1.22)$$

Aquí vemos que, en función del signo de x , aplicamos una función u otra para suprimirlo.

También podemos hacer funciones de funciones. Esto es, hacer que la entrada de una función sea otra función. Esto es la composición de funciones y se denota como $f(g(x))$ o $f \circ g(x)$.

■ **Example 1.13 — Función suma producto.** Queremos hacer una función que primero sume 3 a la entrada y luego lo multiplique todo por 4. Esto lo podemos hacer definiendo dos funciones.

$$f(x) = 4 \cdot x$$

$$g(x) = x + 3.$$

De esta forma, la función que queremos es

$$h(x) = f(g(x)) = 4 \cdot (x + 3) \quad (1.23)$$

Aquí vemos que las funciones se aplican de dentro a fuera. ■

1.1.7 Gráficas.

Las gráficas son representaciones visuales de situaciones matemáticas ampliamente utilizadas. Estas vienen dadas por unos ejes, que indican los valores que toman las variables que vamos a representar, y las figuras a representar, puntos, líneas, planos, etc. Vamos a poner un ejemplo en 2 dimensiones.

■ **Example 1.14 — Gráfica de puntos.** Queremos representar 3 puntos en el espacio, el $O = (0, 0)$, el $P = (2, 1)$ y el $Q = (-1, -2)$. Los puntos son lugares en el espacio cuya posición viene dada por las coordenadas, que usualmente son (x, y) . Vamos a ver la gráfica de estos puntos en la Fig. 1.1.

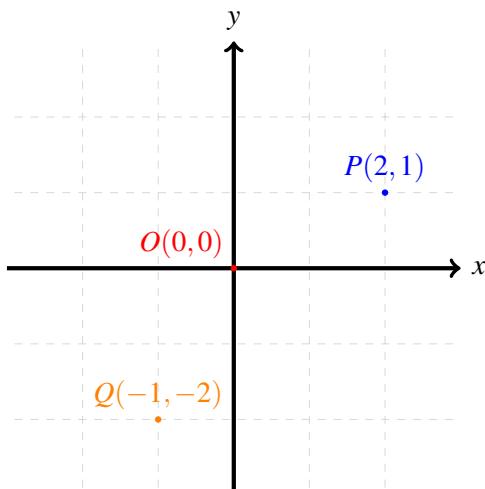


Figure 1.1: Gráfica de los puntos $O = (0, 0)$, $P = (2, 1)$ y $Q = (-1, -2)$.

Podemos observar que el punto O está en el centro de los ejes, mientras que para llegar el punto P tendríamos que movernos 2 unidades en la dirección x y 1 en la dirección y . Además, para llegar al punto Q tenemos que movernos 1 unidad en la dirección x negativa y 2 en la dirección y negativa. ■

Así, vemos que los puntos vienen dados por cuantas unidades hay que moverse sobre cada uno de los ejes de la gráfica para llegar hasta ellos desde el centro.

Las funciones también se pueden representar en las gráficas, mediante el uso de los puntos $(x, f(x))$, donde hacemos que el punto esté a una altura dada por la función aplicada a su posición horizontal.

■ **Example 1.15 — Funciones en gráficas.** Si queremos graficar la función $f(x) = 0.5 \cdot x$, tendremos que tomar todos los x posibles en el intervalo de la gráfica y evaluarlos con la función $f(x)$. Esto representado sería la Fig. 1.2

Podemos ver cómo a cada x le corresponde una y en la gráfica, dada por $f(x)$. ■

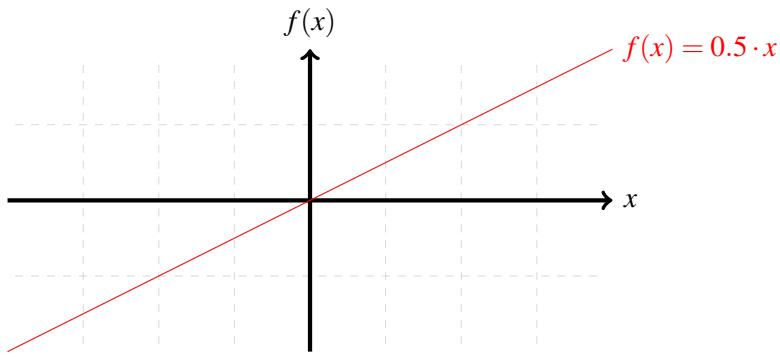


Figure 1.2: Gráfica de la función $f(x) = 0.5 \cdot x$ para $x \in [-4, 4]$.

Hemos visto el caso bidimensional, pero podemos ir a espacios con más dimensiones. Por ejemplo, en 3 dimensiones podemos hacer gráficas donde un punto P tenga 3 elementos, el primero cuánto nos desplazamos en la dirección x , el segundo cuánto en la dirección y y el tercero cuánto en la dirección z . Podemos observar en la Fig. 1.3 cómo se vería el punto $P = (0.9, 2.5, 2.4)$, producto de avanzar 0.9 unidades en la dirección x , 2.5 en la dirección y y 2.4 en la dirección z .

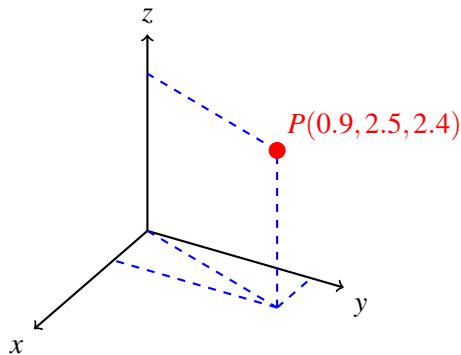


Figure 1.3: Punto $P = (0.9, 2.5, 2.4)$.

En caso de tener puntos en 4 dimensiones, estos serían de la forma $P = (P_1, P_2, P_3, P_4)$, con 4 elementos. Para un número general N de dimensiones, nuestro punto estará compuesto por N elementos. Cada uno de estos elementos lo llamaremos una componente del punto P . Como es evidente, no podemos visualizar más de 3 dimensiones, así que lo dejaremos como un aspecto abstracto.

1.2 Números complejos

Una parte muy importante en todos los cálculos que realizaremos en computación cuántica es la de los números complejos. Si queremos resumir, podemos decir que un número complejo es un número que se compone de dos partes, tales que se cumplen ciertas operaciones. No obstante, vamos a optar por una introducción más cuidadosa al concepto, paso por paso.

1.2.1 Unidad imaginaria

Lo primero que tenemos que estudiar es la unidad imaginaria. Si recordamos de las clases de matemáticas del instituto, la raíz cuadrada y de un número x es un número tal que $y^2 = x$. Siempre se nos enseñó que la raíz cuadrada solo está definida para números positivos, de forma que si tenemos $\sqrt{-|x|}$, esta no puede existir. De ahí parte el concepto de la unidad imaginaria.

La unidad imaginaria i se define como el número que precisamente al multiplicarlo por sí mismo obtenemos $\sqrt{-1}$. Esto es, ya que a primera vista no existe, lo definimos a mano, aunque veremos que tiene muchas aplicaciones ‘reales’. Por tanto, si tenemos un número x positivo, siempre podremos decir que $y = \sqrt{-x} = \sqrt{-1}\sqrt{x} = i\sqrt{x}$. O sea, que la raíz cuadrada de todo número negativo será i veces la raíz cuadrada de su módulo. Así mismo, $(ix) \cdot (iy) = -xy$.

1.2.2 Número complejo

Una vez definida la unidad imaginaria, podemos abordar el número complejo como tal. Un número complejo es un número compuesto por dos partes, la parte real y la parte imaginaria, de forma que tenga la forma

$$z = x + iy, \quad (1.24)$$

también expresado a veces como $z = (x, y)$ en analogía a un vector de dos dimensiones, como veremos en la siguiente sección. Esta es la llamada ‘representación binómica’. Sin embargo, ya podemos ir obteniendo una visión de lo que es un número complejo como un punto en un plano bidimensional. Aquí, x es la parte real del número complejo z , lo cual se expresa como $x = \Re(z)$, mientras que y es la parte imaginaria de z , expresado como $y = \Im(z)$. De esta manera, el eje x del plano sería el eje real, mientras que el eje y sería el eje imaginario. Así, podemos ver algunos números complejos representados como puntos en la Fig. 1.4.

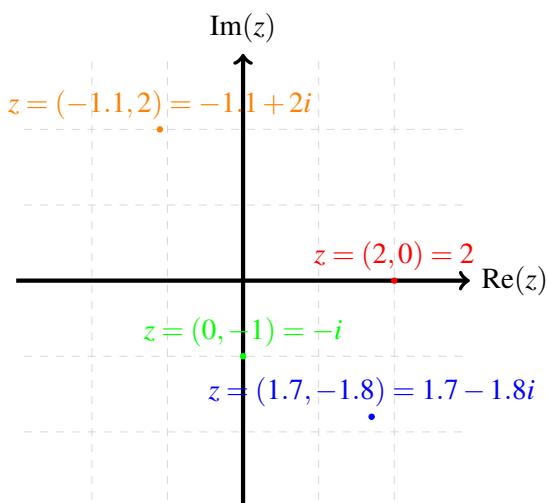


Figure 1.4: Gráfica de los números $z = (2, 0) = 2$, $z = (1.7, -1) = 1.7 - 1.8i$, $z = (-1.1, 2) = -1.1 + 2i$ y $z = (0, -1) = -i$.

Un número complejo tiene diversas propiedades que lo caracterizan, las cuales pueden llegar a facilitarnos bastantes cálculos con ellos. Para ello, lo primero que debemos hacer es definir la operación de ‘conjugación’ de un número complejo, la cual nos dará su ‘complejo conjugado’.

Definition 1.2.1 — Complejo conjugado. Dado un número complejo $z = x + iy$, diremos que su complejo conjugado es

$$z^* = x - iy. \quad (1.25)$$

Esto es, la operación de conjugación consiste en intercambiar i por $-i$. Esta operación sería reflejar el punto con respecto al eje x , ya que le cambiamos el signo a la componente y .

Con esto en mente, la primera de estas propiedades sería el módulo del número complejo.

Definition 1.2.2 — Módulo del número complejo. Dado un número complejo $z = x + iy$, diremos que su módulo viene dado por

$$|z| = \sqrt{zz^*} = \sqrt{(x + iy) \cdot (x - iy)} = \sqrt{x^2 + ixy - ixy + y^2} = \sqrt{x^2 + y^2}. \quad (1.26)$$

Este módulo representa la distancia del punto al origen de coordenadas $(0,0)$. Veremos en la sección de vectores que esto es equivalente al módulo de un vector de 2 dimensiones.

La segunda propiedad de un número complejo es su fase o argumento.

Definition 1.2.3 — Fase de un número complejo. Dado un número complejo $z = x + iy$, diremos que su fase viene dada por

$$\theta = \text{Arg}(z) = \arctan 2(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{si } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{si } y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{si } y < 0, x < 0 \\ \frac{\pi}{2} & \text{si } y > 0, x = 0 \\ -\frac{\pi}{2} & \text{si } y < 0, x = 0 \\ \text{indefinido} & \text{si } y = 0, x = 0 \end{cases} \quad (1.27)$$

definido en los 4 cuadrantes y sus límites. La fase nos da el ángulo de la línea que conecta el punto con el origen y la línea del eje horizontal.

Con estas dos propiedades definidas, podemos definir un número complejo como el movernos una distancia $|z|$ en la dirección con ángulo θ respecto al eje real. Esto lo podemos observar en la Fig. 1.5. Esta es la representación polar del ‘número complejo’.

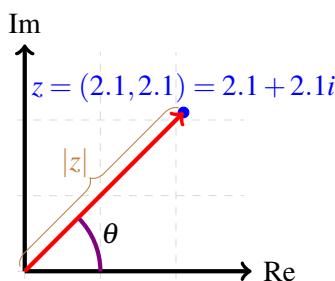


Figure 1.5: Módulo y fase de un número complejo.

Definition 1.2.4 — Representación polar. Dado un número complejo $z = x + iy$, diremos que su representación polar será

$$z = |z|e^{i\theta}, \quad (1.28)$$

siendo $|z| = \sqrt{x^2 + y^2}$ y $\theta = \arctan 2(y, x)$.

Para recuperar las componentes x e y utilizaremos una fórmula extremadamente importante en este campo, la fórmula de Euler. Esta fórmula puede demostrarse fácilmente, mediante un método llamado ‘expansión en serie de Taylor’, pero la omitiremos por no ser necesaria.

Theorem 1.2.1 — Fórmula de Euler. Dado un $x \in \mathbb{R}$, se cumple que

$$e^{ix} = \cos x + i \sin x. \quad (1.29)$$

La principal ventaja que nos ofrece la forma polar es que podemos hacer uso de una de las propiedades más importantes de la función exponencial:

$$e^a e^b = e^{a+b}. \quad (1.30)$$

Por tanto, si tenemos dos números complejos en forma polar $z = |z|e^{i\theta_z}$ y $w = |w|e^{i\theta_w}$, entonces su producto será

$$zw = |z|e^{i\theta_z}|w|e^{i\theta_w} = |z||w|e^{i\theta_z}e^{i\theta_w} = |z||w|e^{i(\theta_z+\theta_w)}, \quad (1.31)$$

mucho más sencillo de calcular e interpretar que

$$zw = (x + iy)(a + ib) = (xa - yb) + i(ya + xb). \quad (1.32)$$

En el caso de que ambos números complejos tengan un módulo unidad, el producto entre dos números complejos se interpreta como una suma de sus fases. Esto será especialmente importante para algoritmos que veremos, como la estimación cuántica de fase o el QAOA.

1.3 Vectores

Empezaremos con el concepto matemático más básico y fundamental que necesitaremos en los temas posteriores, los vectores. El vector es la representación básica que utilizaremos para todos los sistemas físicos, ya que son una colección de cantidades con un significado dado y relacionadas entre sí a través de rotaciones. Veremos en profundidad los diversos aspectos que rodean a los vectores sin irnos a conceptos innecesarios.

1.3.1 Definición

La forma más sencilla de ver un vector es gráficamente como una flecha que apunta a un punto dado en el espacio desde el centro de este. De esta forma, normalmente se llama a un vector de la misma forma que al punto hacia el que apunta. Por ejemplo, un vector que apuntase al punto $P = (1.2, 3, 2.1)$ sería

$$\vec{v} = (1.2, 3, 2.1). \quad (1.33)$$

Un vector general de N dimensiones viene representado igual que un punto de N dimensiones, de forma que sería

$$\vec{v} = (v_1, v_2, v_3, \dots, v_N). \quad (1.34)$$

Ahora bien, aunque los vectores se definen casi igual para el caso de tener componentes complejas en lugar de reales, trabajaremos con componentes reales con el fin de simplificar la interpretación geométrica de los conceptos.

Podemos ver la representación gráfica del vector \vec{v} de 1.33 en la Fig. 1.6, viendo los caminos que habría que recorrer para llegar al punto.

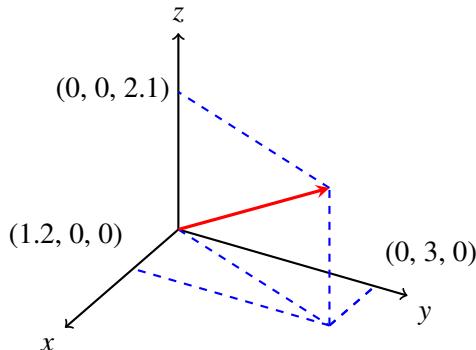


Figure 1.6: Vector $\vec{v} = (1.2, 3, 2.1)$ en coordenadas cartesianas.

Como vemos, un vector tiene una **dirección** (la línea sobre la que se encuentra), un **sentido** (hacia donde está la punta) y una **longitud** (el módulo, que es la distancia del punto al origen). La longitud del vector se calcula mediante el uso del teorema de Pitágoras.

Definition 1.3.1 — Módulo del vector. Dado un vector de N dimensiones $\vec{v} = (v_1, v_2, \dots, v_N)$, su módulo o longitud $|\vec{v}|$ viene dado por la distancia del origen al punto al que apunta, la cual es

$$|\vec{v}| = \sqrt{\sum_{i=1}^N |v_i|^2} = \sqrt{|v_1|^2 + |v_2|^2 + \dots + |v_N|^2}. \quad (1.35)$$

Una vez definido el módulo del vector, podemos descomponer el propio vector en dirección y módulo mediante

$$\vec{v} = |\vec{v}| \cdot \hat{v}, \quad (1.36)$$

siendo \hat{v} un vector de módulo 1 que apunta en la misma dirección y sentido que \vec{v} . Este será el llamado **vector unitario** de \vec{v} .

Definition 1.3.2 — Vector unitario. Dado un vector \vec{v} , diremos que su vector unitario asociado será el vector de módulo 1 que apunte en misma dirección y sentido que este. Este será

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|}. \quad (1.37)$$

También diremos que este vector \hat{v} está normalizado.

1.3.2 Vectores base

Podemos ver en la Fig. 1.6 que las direcciones x , y y z se pueden considerar vectores en sí mismas. Si queremos conservar el significado en unidades de distancia que usábamos hasta ahora, podemos definir cada una de las direcciones como un vector unitario, de forma que tendríamos \hat{x} , \hat{y} y \hat{z} . Usando estos 3 vectores unitarios podemos construir todo los posibles vectores de 3 componentes. Los llamaremos **vectores base**. En este caso, esta la llamaremos **base estándar**. Así, todo vector en 3 dimensiones se puede describir como

$$\vec{v} = v_1\hat{x} + v_2\hat{y} + v_3\hat{z}. \quad (1.38)$$

Ahora bien, podemos ver construir los puntos que describen un vector moviéndonos cierta distancia en cada una de las direcciones descritas por estos vectores base. Sin embargo, es fácil darse cuenta de que a cada punto se puede llegar por una infinidad de caminos diferentes. Por ejemplo, al punto $P = (1, 2, 3)$ podemos llegar moviéndonos 1 unidad en la dirección x , 2 en la dirección y y 3 en la dirección z , pero también podemos llegar al mismo punto moviéndonos en diagonal, en la dirección del vector unitario de $\vec{v} = (1, 2, 3)$, tanta distancia como la longitud de \vec{v} . Por tanto, podríamos escoger otros vectores base para describir a los vectores del espacio en el que estamos. Veamos este ejemplo en la Fig. 1.7, donde hemos graficado el mismo vector, pero rotando los vectores base.

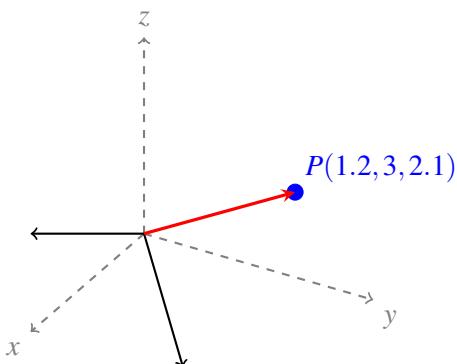


Figure 1.7: Vector $\vec{v} = (1.2, 3, 2.1)$ en una base rotada $\{(1.2, 3, 2.1), (1, 0, 0.5), (0.7, 1, -1.4)\}$. En esta base $\vec{v} = (1, 0, 0)$.

Los requisitos básicos para que un conjunto de vectores pueda considerarse un conjunto de vectores base o una **base vectorial** son:

1. **Todo vector del espacio** puede describirse mediante sus vectores base. Esto es, puedes llegar a cualquier punto del espacio moviéndote en esas direcciones.
2. Todos los vectores de la base son **linealmente independientes**. Esto es, ningún vector base puede construirse usando los demás vectores base.

Vamos a poner un ejemplo de lo que no sería un conjunto que cumpla cada condición:

■ **Example 1.16 — Base incompleta.** Imaginemos la base $\{\vec{b}_1 = (1, 0, 0), \vec{b}_2 = (0, 1, 0)\}$, donde el subíndice aquí indica el elemento de la base, no la componente del vector. Vemos que evidentemente no podemos crear el vector $\vec{v} = (0, 0, 1)$ combinando \vec{b}_1 y \vec{b}_2 . Por tanto, no puedes crear el espacio en 3 dimensiones con solo 2 direcciones base, ya que solo generan un plano. Podemos completar la base con el vector que falta, de forma que $\{\vec{b}_1 = (1, 0, 0), \vec{b}_2 = (0, 1, 0), \vec{b}_3 = (0, 0, 1)\}$ es una base completa y válida. Por este motivo, para un espacio en N dimensiones necesitaremos N vectores base. ■

■ **Example 1.17 — Base no independiente.** Ahora imaginemos una base de 3 componentes que sea $\{\vec{b}_1 = (1, 0, 0), \vec{b}_2 = (0, 1, 0), \vec{b}_3 = (0, 2, 0)\}$. Vemos que aunque tengamos 3 vectores, el tercero es dos veces el segundo, por lo que no aporta una dirección nueva. Esto es, el segundo y el tercer vector son linealmente dependientes.

Podemos ver otro ejemplo menos simple, con la base $\{\vec{b}_1 = (1, 0, 0), \vec{b}_2 = (0, 1, 0), \vec{b}_3 = (0.5, 0.75, 0)\}$. Esta base tampoco es válida ya que el tercer vector es $0.5\vec{b}_1 + 0.75\vec{b}_2$, por lo que son linealmente dependientes. ■

Estas dos condiciones se aseguran de que la base da suficientes direcciones válidas como para llegar a todas partes, pero no tantas como para ser redundantes. Además, estos vectores base no tienen porqué tener módulo 1, pero normalmente desearemos que así sea. Una vez creada una nueva base $b = \{\vec{b}_1 = (b_{1,1}, b_{1,2}, b_{1,3}), \vec{b}_2 = (b_{2,1}, b_{2,2}, b_{2,3}), \vec{b}_3 = (b_{3,1}, b_{3,2}, b_{3,3})\}$, podemos describir cualquier vector en 3 dimensiones como

$$\vec{v} = v_1\hat{x} + v_2\hat{y} + v_3\hat{z} = v'_1\vec{b}_1 + v'_2\vec{b}_2 + v'_3\vec{b}_3, \quad (1.39)$$

donde los coeficientes v'_i y v_i están relacionados por la relación entre las bases. Para ver cómo es el cálculo, usemos un caso simple en 2 dimensiones.

■ **Example 1.18 — Cambio de base en 2 dimensiones.** Tenemos el vector $\vec{v} = (1, 2)$ en la base estándar y queremos ver sus coeficientes en la nueva base $b = \{\vec{b}_1 = (1, 0.5), \vec{b}_2 = (-2, 1)\}$. Lo primero que necesitaremos será poner los vectores base \hat{x} e \hat{y} en función de la nueva base \vec{b}_1 y \vec{b}_2 . La forma de la base ya nos da las siguientes 2 ecuaciones lineales:

$$\vec{b}_1 = 1\hat{x} + 0.5\hat{y}$$

$$\vec{b}_2 = -2\hat{x} + 1\hat{y}$$

cuya solución es

$$\hat{x} = 0.5\vec{b}_1 - 0.25\vec{b}_2 \quad (1.40)$$

$$\hat{y} = 1\vec{b}_1 + 0.5\vec{b}_2 \quad (1.41)$$

Ahora sustituimos (1.40) en la ecuación $\vec{v} = 1\hat{x} + 2\hat{y}$:

$$\vec{v} = 1\hat{x} + 2\hat{y} = 0.5\vec{b}_1 - 0.25\vec{b}_2 + 2(\vec{b}_1 + 0.5\vec{b}_2) = 2.5\vec{b}_1 + 0.75\vec{b}_2 \quad (1.42)$$

por lo que en la nueva base, el vector sería expresado como $\vec{v} = (2.5, 0.75)$. Es importante recordar que ambos vectores $\vec{v} = (1, 2)$ y $\vec{v} = (2.5, 0.75)$ son el mismo, pero expresado en dos bases diferentes. Por ello mismo, cuando usamos dos bases diferentes, es mejor utilizar la notación de la ecuación (1.39). ■

Así, en general podemos decir que un vector de N dimensiones se puede expresar como

$$\vec{v} = \sum_{i=1}^N v_i \vec{b}_i. \quad (1.43)$$

Ahora bien, vemos que si calculamos el módulo de los dos \vec{v} obtenemos 2.24 y 2.61 respectivamente. Sin embargo, por el hecho de cambiar la base (las direcciones básicas) no debería cambiar la longitud del vector. En otras palabras, cambiando la manera de medir no debería cambiar aquello que medimos. Esto es debido a que nuestra definición del módulo no es completa. Vamos a explicar un concepto antes de meternos en esto.

1.3.3 Producto escalar

Todos tenemos claro cómo multiplicar dos números, pero ¿cómo multiplicamos dos vectores?. Hay dos maneras de multiplicar vectores, la escalar (que devuelve un número) y la vectorial (que devuelve otro vector). Nos centraremos en la escalar, que es la más útil para lo que nos ocupa.

El producto escalar es una operación que toma dos vectores y devuelve un número. Este número se puede interpretar como el ‘parecido’ que tienen dichos vectores. Para obtener el producto escalar de un vector expresado en la base estándar, solo tendremos que multiplicar cada componente del primer vector con su correspondiente del segundo y sumar el total:

Definición 1.3.3 — Producto escalar. Dado un vector \vec{v} y un vector \vec{w} expresados en la base estándar, su producto escalar $\vec{v} \cdot \vec{w}$ viene dado

$$\vec{v} \cdot \vec{w} = \sum_{i=1}^N v_i w_i. \quad (1.44)$$

Si un vector es el conjugado complejo del otro, tenemos

$$\vec{v} \cdot \vec{v}^* = \vec{v}^* \cdot \vec{v} = \sum_{i=1}^N |v_i|^2 = |\vec{v}|^2, \quad (1.45)$$

por lo que el producto escalar de un vector con su conjugado complejo nos permite obtener el módulo al cuadrado de nuestro vector. En el caso de componentes reales, ambos vectores son el mismo.

Si existe un ángulo θ entre ambos vectores, el producto escalar también se puede expresar como:

$$\vec{v} \cdot \vec{w} = |\vec{v}| |\vec{w}| \cos \theta. \quad (1.46)$$

Podemos ver esta segunda forma en la Fig. 1.8.

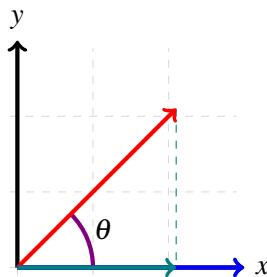


Figure 1.8: Vectores $\vec{v} = (2.1, 2.1)$ y $\vec{w} = (3, 0)$ y la proyección de \vec{v} en la dirección de \vec{w} .

Esto se puede entender mejor si descomponemos cada vector en módulo y dirección:

$$\vec{v} \cdot \vec{w} = |\vec{v}| |\vec{w}| \hat{v} \cdot \hat{w}, \quad (1.47)$$

donde vemos que el producto escalar entre dos vectores unitarios da el ángulo entre ellos. Esto es, la coincidencia entre sus direcciones. Si tenemos $\vec{v} = (1, 0)$ y $\vec{w} = (0, 1)$, su producto escalar será

0, por ser direcciones totalmente diferentes, mientras que si tenemos $\vec{v} = \vec{w} = (1, 0)$, el producto escalar será 1, ya que es la misma dirección.

Por ello, si comparamos dos vectores, su producto escalar nos da una idea de cuánto se parecen ambos vectores, aumentado por el módulo de ambos. Para obtener el ángulo entre estos vectores, haremos el siguiente cálculo:

$$\cos \theta = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} = \frac{\sum_{i=1}^N v_i w_i}{|\vec{v}| |\vec{w}|}. \quad (1.48)$$

Podemos ver entonces que, si tenemos dos vectores que sea perpendiculares (con direcciones totalmente diferentes), su producto escalar será cero.

Ahora bien, tenemos que pensar en la forma de (1.43). Para una base general, el producto escalar será

$$\vec{v} \cdot \vec{w} = \left(\sum_{i=1}^N v_i \vec{b}_i \right) \cdot \left(\sum_{j=1}^N w_j \vec{b}_j \right) = \sum_{i=1}^N \sum_{j=1}^N v_i w_j \vec{b}_i \cdot \vec{b}_j \quad (1.49)$$

donde vemos que, a menos de que los \vec{b}_i sean todos perpendiculares entre sí, añaden términos cruzados, ya que hay que hacer el producto escalar del \vec{b}_1 con el \vec{b}_2 , por ejemplo, multiplicado por $v_1 w_2$. Además, siempre añaden el factor de su propio módulo. De aquí surge la discrepancia comentada en la subsección anterior. Si tenemos este hecho en cuenta, da igual con qué base trabajemos, en todos los casos el módulo de cualquier vector va a ser el mismo en cualquier base que elijamos. A partir de este punto, hablaremos siempre de bases donde todos los vectores base son perpendiculares entre sí y de módulo uno (base ortonormal).

Definition 1.3.4 — Base ortonormal. Una base de N vectores $\{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_N\}$ se dirá que es ortonormal si para todo vector de la base se cumple que

$$|\vec{b}_i| = 1, \forall i \in [1, N] \quad (1.50)$$

y para toda pareja de vectores base se cumple que

$$\vec{b}_i \cdot \vec{b}_j = 0, \forall i \neq j \in [1, N] \quad (1.51)$$

Por tanto, con todo esto vamos a definir totalmente un vector.

Definition 1.3.5 — Vector. Un vector \vec{v} de dimensión N es un objeto descrito por N componentes asociadas a N vectores base. En el caso de que dichos vectores base $\{\hat{b}_i | i \in [1, N]\}$ formen una base ortonormal, el vector podrá expresarse como

$$\vec{v} = \sum_{i=1}^N v_i \hat{b}_i \quad (1.52)$$

o, por otro lado, en función a su módulo $|\vec{v}|$ y dirección \hat{v} , como

$$\vec{v} = |\vec{v}| \cdot \hat{v}. \quad (1.53)$$

Siempre podremos pasar de una base arbitraria a una ortonormal, y pasar de una base ortonormal a otra se puede considerar como una rotación del espacio en el que vive nuestro vector, como si lo mirásemos desde otro punto. Esto nos lleva directamente al siguiente punto, que son las matrices.

1.4 Matrices

1.4.1 Definición

Las matrices son representaciones en 2 dimensiones de datos. Como ejemplo de una matriz tenemos la Tabla 1.1, donde tendríamos una matriz 3×3 . Una matriz viene dada por 2 dimensiones, el número de filas N y el número de columnas M , diciéndose que es una matriz $N \times M$.

$$\begin{pmatrix} 0.1 & 1.2 & -3.1 \\ -2 & -3 & 2.1 \\ 0 & 1.1 & 2 \end{pmatrix}$$

Table 1.1: Representación de una matriz 3×3 .

Podemos ver en la Tabla 1.2 un ejemplo de matriz representando una tabla de datos.

Código	Distrito	Precio
2001	3	3.5
102	1	2
31038	1	7
3113	2	3.1

Table 1.2: Matriz de tabla de datos 4×3 .

Ahora bien, las matrices aparte de servir como representaciones de datos, pueden servir como representaciones de tensores, los cuales veremos en la siguiente sección. Lo importante con lo que tenemos que quedarnos es con que a partir de ahora vamos a llamar ‘matrices’ a los ‘2-tensores’.

Entonces, ¿qué es una matriz (2-tensor)? Una matriz es un objeto que tiene elementos con 2 subíndices, uno para indicar la fila a la que nos referimos y otro para la columna. Al igual que en los vectores, cada índice está asociado a un vector base \hat{b}_i , y tendrá elementos A_{ij} asociados a estos vectores base.

Definition 1.4.1 — Matriz (2-tensor). Una matriz, referido a un 2-tensor, de dimensiones $N \times M$, asociada a una base de vectores ortonormales $\{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_N, \dots, \vec{b}_M\}$, viene expresada como

$$A = \sum_{i=1}^N \sum_{j=1}^M A_{ij} \hat{b}_i \hat{b}_j. \quad (1.54)$$

De manera gráfica, los elementos A_{ij} pueden representarse como una tabla de N filas y M columnas, donde el elemento A_{ij} estaría situado en la fila i y la columna j . Esto es, se colocarían como

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{21} & A_{22} & \cdots & A_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NM} \end{pmatrix}. \quad (1.55)$$

A diferencia del caso de los vectores, al tener dos índices, existe la posibilidad de realizar una operación de intercambio de índices. Esta operación de intercambio se denomina ‘trasposición’ y nos da la ‘matriz traspuesta’.

Definition 1.4.2 — Traspuesta de una matriz. Dada una matriz $A = \sum_{i=1}^N \sum_{j=1}^M A_{ij} \hat{b}_i \hat{b}_j$, diremos que su matriz traspuesta viene dada por $A^T = \sum_{i=1}^N \sum_{j=1}^M A_{ji} \hat{b}_i \hat{b}_j$. Esto es equivalente a

intercambiar filas por columnas, de forma que ahora tengamos

$$\begin{pmatrix} A_{11} & A_{21} & \cdots & A_{M1} \\ A_{12} & A_{22} & \cdots & A_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1N} & A_{2N} & \cdots & A_{MN} \end{pmatrix}. \quad (1.56)$$

Podemos utilizar la notación de la traspuesta para los vectores, indicando que nuestro vector pasa de ser un vector fila a un vector columna.

Al igual que los vectores, las componentes de una matriz pueden ser números complejos, de manera que podemos definir una matriz ‘traspuesta conjugada’ o ‘dagger’, la cual es traspuesta y compleja conjugada a la vez.

Definition 1.4.3 — Matriz traspuesta conjugada o hermítica conjugada. Dada una matriz $A = \sum_{i=1}^N \sum_{j=1}^M A_{ij} \hat{b}_i \hat{b}_j$, diremos que su matriz traspuesta conjugada o hermítica conjugada viene dada por $A^\dagger = \sum_{i=1}^N \sum_{j=1}^M A_{ji}^* \hat{b}_i \hat{b}_j$. Esto es equivalente a intercambiar filas por columnas y hacer el complejo conjugado de cada componente, de forma que ahora tengamos

$$\begin{pmatrix} A_{11}^* & A_{21}^* & \cdots & A_{M1}^* \\ A_{12}^* & A_{22}^* & \cdots & A_{M2}^* \\ \vdots & \vdots & \ddots & \vdots \\ A_{1N}^* & A_{2N}^* & \cdots & A_{MN}^* \end{pmatrix}. \quad (1.57)$$

La operación de conjugación compleja y la de traspuesta pueden realizarse en cualquier orden, de manera que

$$A^\dagger = (A^*)^T = (A^T)^*. \quad (1.58)$$

Como es evidente, operaciones relacionadas con el cambio de base, estudiado en la Sección 1.3, son exactamente iguales para las matrices. Las matrices pueden expresarse en diferentes bases, cambiando el valor de sus elementos, pero siendo el mismo objeto.

1.4.2 Productos matriciales

De manera algebráica, las matrices sirven como mapas lineales entre dos espacios vectoriales. ¿Qué significa esto? Las matrices permiten transformar unos vectores en otros de manera lineal. Para entenderlo, vamos a explorar el producto entre una matriz y un vector. Es importante tener en cuenta que la siguiente explicación se enfoca en entender esta operación, no en realizar un análisis riguroso de la misma.

El producto entre una matriz A , de dimensiones $N \times M$ y un vector \vec{v} de dimensión M , se expresa como

$$A\vec{v} = \left(\sum_{i=1}^N \sum_{j=1}^M A_{ij} \hat{b}_i \hat{b}_j \right) \cdot \left(\sum_{k=1}^M v_k \hat{b}_k \right). \quad (1.59)$$

Aquí vemos que estamos haciendo un producto escalar entre la matriz A y el vector \vec{v} , por lo que, reordenando los términos, tenemos

$$A\vec{v} = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^M (A_{ij} v_k \hat{b}_i \hat{b}_j \cdot \hat{b}_k), \quad (1.60)$$

donde vemos que vamos a realizar productos escalares entre los vectores base de A y \vec{v} , que si suponemos que forman una base ortonormal, serán 0 si $j \neq k$ y 1 si $j = k$. Por tanto, tendremos

que en el sumatorio solo sobrevivirán los términos donde $j = k$, reduciendo ambos sumatorios a un único sumatorio en j y sustituyendo toda k por una j :

$$A\vec{v} = \sum_{i=1}^N \sum_{j=1}^M (A_{ij} v_j \hat{b}_i). \quad (1.61)$$

Si reordenamos los paréntesis, podemos ver que el resultado es un nuevo vector \vec{w} de dimensión N ,

$$\vec{w} = A\vec{v} = \sum_{i=1}^N \left(\sum_{j=1}^M A_{ij} v_j \right) \hat{b}_i = \sum_{i=1}^N w_i \hat{b}_i, \quad (1.62)$$

donde vemos que los elementos del vector resultante son $w_i = \sum_{j=1}^M A_{ij} v_j$.

Por tanto, una matriz lo que hace es tomar un vector de dimensión M de entrada y devolver un vector de dimensión N de salida, con elementos relacionados de manera lineal. También podemos realizar un producto por la izquierda de la matriz, de forma que obtengamos un vector

$$\vec{w} = \vec{v}^T A = \sum_{j=1}^M \left(\sum_{i=1}^N v_i A_{ij} \right) \hat{b}_j = \sum_{j=1}^M w_j \hat{b}_j, \quad (1.63)$$

donde vemos que la dimensión de entrada es N y la de salida M . Sin embargo, lo más usual es el producto por la derecha.

Definition 1.4.4 — Producto matriz-vector. El producto entre una matriz A de dimensiones $N \times M$ y un vector \vec{v} de dimensión M , ambos expresados en la misma base ortonormal $\{\hat{b}_i\}$ con elementos A_{ij} y v_j , viene dado por un vector

$$\vec{w} = A\vec{v} = \sum_{i=1}^N w_i \hat{b}_i \quad | \quad w_i = \sum_{j=1}^M A_{ij} v_j. \quad (1.64)$$

La pregunta que nos podemos hacer ahora es cómo multiplicar dos matrices entre sí. Esto es completamente análogo al caso con los vectores. Dadas dos matrices A de dimensiones $N \times M$ y B de dimensiones $M \times P$, su producto será

$$AB = \left(\sum_{i=1}^N \sum_{j=1}^M A_{ij} \hat{b}_i \hat{b}_j \right) \cdot \left(\sum_{k=1}^M \sum_{q=1}^P B_{kq} \hat{b}_k \hat{b}_q \right) = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^M \sum_{q=1}^P (A_{ij} B_{kq} \hat{b}_i \hat{b}_j \cdot \hat{b}_k \hat{b}_q). \quad (1.65)$$

Al igual que antes, los sumatorios en j y k se unen en uno en j y las k pasan a ser j .

$$AB = \sum_{i=1}^N \sum_{j=1}^M \sum_{q=1}^P (A_{ij} B_{jq} \hat{b}_i \hat{b}_q) = \sum_{i=1}^N \sum_{q=1}^P \left(\sum_{j=1}^M A_{ij} B_{jq} \right) \hat{b}_i \hat{b}_q, \quad (1.66)$$

donde podemos ver que tenemos una matriz resultante de dimensión $N \times P$

$$C = AB = \sum_{i=1}^N \sum_{q=1}^P C_{iq} \hat{b}_i \hat{b}_q, \quad (1.67)$$

con elementos son $C_{iq} = \sum_{j=1}^M A_{ij} B_{jq}$.

Definition 1.4.5 — Producto matriz-matriz. El producto entre una matriz A de dimensiones $N \times M$ y otra matriz B de dimensiones $M \times P$, ambas expresadas en la misma base ortonormal

$\{\hat{b}_i\}$ con elementos A_{ij} y B_{jk} , viene dado por una matriz

$$C = AB = \sum_{i=1}^N \sum_{k=1}^P C_{ik} \hat{b}_i \hat{b}_k \quad | \quad C_{ik} = \sum_{j=1}^M A_{ij} B_{jk}. \quad (1.68)$$

Es importante notar que no es lo mismo AB que BA , ya que $BA = \sum_{i=1}^N \sum_{q=1}^P (\sum_{j=1}^M A_{ji} B_{qj}) \hat{b}_i \hat{b}_q$, resultando en elementos de matriz completamente diferentes para C . Se dice que las matrices no conmutan. Debido a ello, podemos definir dos operaciones muy importantes entre parejas de matrices: el conmutador y el anticomutador.

Definition 1.4.6 — Conmutador de dos matrices. Dadas dos matrices A y B , se dirá que el conmutador de la matriz A con la matriz B es

$$[A, B] = AB - BA. \quad (1.69)$$

Esta operación dará como resultado una matriz de ceros en el caso de que ambas matrices conmuten. Por tanto, esta operación nos da cuenta de ‘cuanto conmutan’ dos matrices. Esto tiene unas consecuencias capitales que veremos en la introducción física.

Para el conmutador se cumplen las siguientes identidades.

- $[B, A] = -[A, B]$, ya que $[B, A] = BA - AB = -(AB - BA) = -[A, B]$.
- $[AB, C] = A[B, C] + [A, C]B$, ya que $ABC = A[B, C] + ACB$ y $CAB = [C, A]B + ACB$, similar a la regla de la cadena.

Definition 1.4.7 — Anticomutador de dos matrices. Dadas dos matrices A y B , se dirá que el anticomutador de la matriz A con la matriz B es

$$\{A, B\} = AB + BA. \quad (1.70)$$

Vemos que esta operación es conceptualmente opuesta a la del conmutador. Si el anticomutador de dos matrices es nulo, se dirá que dichas matrices ‘anticomutan’ entre sí.

Con estas dos operaciones definidas, un producto de matrices siempre podrá expresarse como

$$AB = \frac{1}{2} ([A, B] + \{A, B\}). \quad (1.71)$$

Esto debido a que $[A, B] + \{A, B\} = AB - BA + AB + BA = 2AB$.

Algo a destacar es que $(AB)^T = B^T A^T$, debido al intercambio de orden de todos los índices. A su vez, $(AB)^\dagger = B^\dagger A^\dagger$. Con ello, es importante destacar también que si tenemos un vector $\vec{w} = A\vec{v}$, al hacer productos escalares con él a la izquierda, tendremos que realizarle una traspuesta. Esto es,

$$\vec{w} \cdot \vec{a} = \vec{w}^T \vec{a} = \vec{v}^T A^T \vec{a}. \quad (1.72)$$

Lo único que significa esto es que para expresar el producto escalar de manera similar al producto matricial, necesitamos cambiar el vector de la izquierda por su traspuesta. O sea, cambiar el vector fila por un vector columna.

1.4.3 Interpretación geométrica

RELEVAR

1.4.4 Características matriciales

Con estas dos operaciones definidas, podemos definir muchas características de una matriz.

Definition 1.4.8 — Autovalores y autovectores de una matriz. Dada una matriz A de dimensiones $N \times N$, esta siempre podrá ser expresada en una base ortonormal en la cual solo tenga elementos diagonales diferentes de cero. Los vectores $\hat{\lambda}_i$ de dicha base se llamarán autovectores y los elementos λ_i diagonales de dicha matriz en esta base se llamarán autovalores. De esta manera, una matriz siempre se podrá expresar como

$$A = \sum_{i=1}^N \lambda_i \hat{\lambda}_i \hat{\lambda}_i, \quad (1.73)$$

donde vemos que cada autovalor tiene su autovector asociado. La forma gráfica de ver dicha matriz en esta base será

$$\begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}. \quad (1.74)$$

Definition 1.4.9 — Traza de una matriz. Dada una matriz A de dimensiones $N \times N$, su traza vendrá dada como la suma de sus elementos diagonales. Esto es,

$$Tr(A) = \sum_{i=1}^N A_{ii}. \quad (1.75)$$

La traza es independiente de la base escogida para expresar la matriz. Tiene las siguientes propiedades relevantes:

- Dadas dos matrices de misma dimensión A y B , entonces $Tr(A + B) = Tr(A) + Tr(B)$.
- Dada una matriz A y un escalar r , entonces $Tr(rA) = rTr(A)$.
- $Tr(A^T) = Tr(A)$.
- $Tr(AB) = Tr(BA)$.
- Dada una tercera matriz C , $Tr(ABC) = Tr(BCA) = Tr(CAB) \neq Tr(ACB)$.
- $Tr(A) = \sum_i \lambda_i$, siendo λ_i el i -ésimo autovalor de A .

Definition 1.4.10 — Norma de Frobenius. La norma de Frobenius de una matriz A de dimensiones $N \times M$ viene dada por

$$\|A\|_F = \sqrt{\sum_{i=1}^N \sum_{j=1}^M |A_{ij}|^2} = \sqrt{Tr(A^\dagger A)} \quad (1.76)$$

Esta norma sería el equivalente matricial al módulo vectorial.

1.4.5 Matrices relevantes

Ahora bien, hay unas ciertas matrices que son interesantes de conocer en este punto inicial. Estas matrices serán:

Identidad

Matriz identidad \mathbb{I} , con elementos diagonales iguales a 1 y el resto 0.

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}. \quad (1.77)$$

Esta matriz actuando sobre un vector devuelve el mismo vector. Es el elemento neutro de la multiplicación.

Diagonal

Matriz diagonal D , con elementos diagonales iguales a sus autovalores y el resto 0.

$$\begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}. \quad (1.78)$$

Esta matriz actuando sobre un vector, reescala sus componentes, sin mezclarlas entre si. La identidad es un caso particular.

Constante

Matriz constante, con todos sus elementos iguales.

$$\begin{pmatrix} a & a & \cdots & a \\ a & a & \cdots & a \\ \vdots & \vdots & \ddots & \vdots \\ a & a & \cdots & a \end{pmatrix}. \quad (1.79)$$

Simétrica

Matriz simétrica, tal que es igual a su traspuesta: $A = A^T$.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{12} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1N} & a_{2N} & \cdots & a_{NN} \end{pmatrix}. \quad (1.80)$$

Antisimétrica

Matriz antisimétrica, tal que es igual a menos su traspuesta: $A = -A^T$.

$$\begin{pmatrix} 0 & a_{12} & \cdots & a_{1N} \\ -a_{12} & 0 & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{1N} & -a_{2N} & \cdots & 0 \end{pmatrix}. \quad (1.81)$$

Evidentemente, para ser igual a su traspuesta con el signo cambiado, ya que sus elementos diagonales no cambian, deben ser nulos: $a_{ii} = -a_{ii} \Rightarrow a_{ii} = 0$.

1.4.6 Matriz inversa

Una operación matricial muy importante es la inversión de matrices. La matriz inversa A^{-1} de una matriz A es una matriz tal que

$$A^{-1}A = AA^{-1} = \mathbb{I}. \quad (1.82)$$

Esto es, una matriz multiplicada por su inversa nos dará la matriz identidad. Es el equivalente matricial a multiplicar un número a por su inverso $\frac{1}{a}$, lo cual resultará en 1. Es especialmente útil en la resolución de un gran número de problemas que veremos en las siguientes partes del libro.

La matriz inversa A^{-1} se puede expresar fácilmente en la base de autovectores de A . Si $A = \sum_{i=1}^N \lambda_i \hat{\lambda}_i \hat{\lambda}_i$, entonces $A^{-1} = \sum_{i=1}^N \frac{1}{\lambda_i} \hat{\lambda}_i \hat{\lambda}_i$. Podemos comprobar que es su inversa fácilmente, ya que

$$AA^{-1} = \sum_{i=1}^N \lambda_i \frac{1}{\lambda_i} \hat{\lambda}_i \hat{\lambda}_i = \sum_{i=1}^N 1 \hat{\lambda}_i \hat{\lambda}_i = \mathbb{I}. \quad (1.83)$$

Como podemos observar en su definición, si alguno de los autovalores de A fuese 0, el autovalor correspondiente de A^{-1} sería infinito, por lo que la matriz A no tendría inversa.

Otra propiedad interesante es que $(AB)^{-1} = B^{-1}A^{-1}$, ya que $B^{-1}A^{-1}AB = B^{-1}B = \mathbb{I}$.

1.4.7 Matriz unitaria

Un tipo de matriz extremadamente interesante, y que definirá todas nuestras operaciones cuánticas, es el de las matrices unitarias. Estas matrices cumplen la condición

$$UU^\dagger = U^\dagger U = \mathbb{I}. \quad (1.84)$$

Esto es, para estas matrices, su inversa es simplemente su traspuesta conjugada. Esto permite realizar muchos cálculos relacionados con ellas muy sencillos. A su vez, dichas matrices siempre son invertibles, lo cual implica que siempre podemos deshacer su acción.

Un tipo de matrices relacionado es el de las matrices ortogonales, que cumplen

$$UU^T = U^T U = \mathbb{I}. \quad (1.85)$$

Esta relación la podemos encontrar en todas las matrices unitarias de coeficientes reales, ya que en ese caso $U^* = U$.

La interpretación de dichas matrices unitarias es la rotación de vectores sin modificar su módulo.

Proposition 1.4.1 — Rotación de vectores. Dados un vector \vec{v} y una matriz unitaria U , el vector $\vec{w} = U\vec{v}$ tendrá el mismo módulo que \vec{v} , pudiendo modificar solamente su dirección.

Demostración:

$$|\vec{w}|^2 = \vec{w}^* \cdot \vec{w} = \vec{v}^{*T} U^\dagger U \vec{v} = \vec{v}^{*T} \vec{v} = |\vec{v}|^2. \quad (1.86)$$

Por este mismo motivo, todos los autovalores de las matrices unitarias son números complejos de módulo 1, ya que sino, en la base de los autovectores, los valores de alguna componente del vector por el que la multiplicamos cambiaría de módulo.

1.4.8 Matriz hermítica

Las matrices hermíticas también son otra clase de matrices muy importantes en cuántica, ya que serán las que representen las cantidades físicas medibles. Estas matrices cumplen la propiedad

$$A = A^\dagger. \quad (1.87)$$

Esto es, son equivalentes a su traspuesta conjugada. Es una extensión del concepto de matriz simétrica. Esto lleva a que tengan propiedades especiales muy útiles e interpretables físicamente.

- Todos sus autovalores son reales.
- Se pueden descomponer como $A = B + iC$, siendo B una matriz real simétrica y C una matriz real antisimétrica.
- Su inversa también es hermítica. Esto ya que sus autovalores son reales, y las inversas de estos también son reales.
- Su diagonal principal solo tiene números reales, debido a su descomposición.

1.4.9 Matrices de Pauli

Un grupo de matrices de gran utilidad y muy recurrentes en el mecanismo cuántico son las matrices de Pauli. Estas matrices surgen en el contexto del momento angular intrínseco de las partículas (spin) y están relacionadas con las álgebras de Lie, en este caso con $SU(2)$. Todo esto lo veremos en más detalle en sus secciones correspondientes, ahora nos centraremos en la base matemática simple de las mismas.

Las matrices de Pauli son las siguientes:

$$\sigma_X = \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Y = \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_Z = \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.88)$$

En el contexto de los circuitos cuánticos, a cada una de estas matrices se las suele denominar matrices X , Y y Z , obviando el símbolo σ . A pesar de que existen diferentes representaciones de las matrices básicas del álgebra de Lie $SU(2)$, esta es la representación usualmente utilizada.

Las propiedades que deben cumplir dichas matrices base son:

- $[\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k$, siendo ϵ_{ijk} el símbolo de Levi-Civita. Esta propiedad da cuenta de que la conmutación entre dos matrices de Pauli diferentes es igual a la otra con un signo dependiendo del orden, mientras que si son iguales, el conmutador va a ser nulo.
- $\{\sigma_i, \sigma_j\} = 2\delta_{ij}\mathbb{I}$, siendo que anticommutan entre ellas.
- $\sigma_X^2 = \sigma_Y^2 = \sigma_Z^2 = \mathbb{I}$, de manera que son sus propias inversas.
- Su determinante es -1 .
- Su traza es 0 .
- En esta representación, son matrices unitarias y hermíticas.

Una cadena de Pauli de longitud N es el producto tensorial de N matrices de Pauli

$$\sigma_i \otimes \sigma_j \otimes \cdots \otimes \sigma_k. \quad (1.89)$$

Estas cadenas de Pauli son a su vez, matrices hermíticas y unitarias.

Un aspecto muy importante de las matrices de Pauli es que toda matriz hermética 2×2 puede representarse como una combinación lineal de las tres matrices de Pauli y la identidad, de forma que

$$A = \sum_{i=1}^3 a_i \sigma_i + a_4 \mathbb{I}, \quad a_i \in \mathfrak{R} \quad (1.90)$$

Esto se puede demostrar fácilmente de la siguiente manera.

$$A^\dagger = \sum_{i=1}^3 a_i^* \sigma_i^\dagger + a_4^* \mathbb{I}^\dagger \quad (1.91)$$

Como los a_i son reales, $a_i^* = a_i$, y como las matrices de Pauli son hermíticas, $\sigma_i^\dagger = \sigma_i$. Además, la identidad también es hermética, por lo que recuperamos la forma inicial.

Para matrices herméticas $2^N \times 2^N$ se puede demostrar de manera similar que siempre se pueden representar como una combinación lineal de cadenas de Pauli. Por tanto, se podrá expresar como

$$A = \sum_{i,j,\dots,k} a_{i,j,\dots,k} \sigma_i \otimes \sigma_j \otimes \cdots \otimes \sigma_k. \quad (1.92)$$

Esta propiedad será vital para una gran cantidad de algoritmos cuánticos, ya que nos permitirán conectar matrices hermíticas generales con los operadores unitarios que sí podemos utilizar en nuestros circuitos.

1.5 Tensores

Con todo lo aprendido hasta ahora, vamos a explorar una extensión de estas ideas: los tensores.

1.5.1 Definición

Un tensor es una extensión de los conceptos de escalar, vector y matriz que hemos estudiado hasta ahora. En vez de empezar con una definición dura usual, vamos a intentar aproximarnos al concepto de una manera más amigable. Además, debido a que no tendrá relevancia en lo que trataremos de computación cuántica, omitiremos la diferencia entre covariante y contravariante.

Un tensor de orden n (o n -tensor) es un objeto cuyos elementos tienen n índices y están asociados a n vectores base. De hecho, un escalar es un tensor de orden 0, ya que no tiene índices, un vector es un tensor de orden 1, ya que tiene un índice, y una matriz es un tensor de orden 2, ya que tiene 2 índices.

Por tanto, un 3-tensor será aquel que tenga 3 índices. Este sería

$$T = \sum_{i,j,k} T_{ijk} \hat{b}_i \hat{b}_j \hat{b}_k. \quad (1.93)$$

Para un n -tensor general tendremos

$$T = \sum_{i_1, i_2, \dots, i_n} T_{i_1 i_2 \dots i_n} \hat{b}_{i_1} \hat{b}_{i_2} \dots \hat{b}_{i_n}. \quad (1.94)$$

De esta manera, vemos que un tensor tiene una estructura igual que la estructura que hemos visto para vectores y matrices, pero aumentando el número de índices y vectores base. Como es evidente, al hacer un cambio de base, como los vistos en la Sección 1.3, el tensor se transformará sustituyendo las fórmulas del cambio de base en la fórmula (1.94). Esto es lo que define la transformación tensorial bajo un cambio de base. Y de ahí que ‘un tensor es aquello que se transforma como un tensor’.

Definition 1.5.1 Un tensor T de orden n y dimensión D_j para su índice j -ésimo, definido para una base $\{\hat{b}_i\}$ y con elementos T_{i_1, i_2, \dots, i_n} , viene dado por la expresión

$$T = \sum_{i_1, i_2, \dots, i_n} T_{i_1 i_2 \dots i_n} \hat{b}_{i_1} \hat{b}_{i_2} \dots \hat{b}_{i_n}. \quad (1.95)$$

Dicho tensor modifica sus elementos tensoriales al realizar un cambio de base de manera que recupere una forma con la estructura de (1.95), tras sustituir las fórmulas de cambio de base en su expresión.

1.5.2 Operaciones

Las operaciones principales de los tensores son las contracciones. Una contracción es la generalización del producto que hemos visto en los casos vectorial y matricial. Para entenderlo, expondremos la fórmula general y veremos cómo se reduce a las obtenidas anteriormente. Por simplificar, consideraremos que todo está expresado en las mismas bases ortonormales, para no perdernos con métricas.

Definition 1.5.2 — Contracción de dos tensores. Dados dos tensores T y R , de orden n y m respectivamente, con elementos $T_{i_1 i_2 \dots i_n}$ y $R_{j_1 j_2 \dots j_m}$ y en la misma base ortonormal $\{\hat{b}_i\}$, diremos que el tensor E es la contracción de ambos tensores por sus últimos q índices cuando

$$E = \sum_{i_1 i_2 \dots i_{n-q}, j_1 j_2 \dots j_{m-q}} E_{i_1 i_2 \dots i_{n-q}, j_1 j_2 \dots j_{m-q}} \hat{b}_{i_1} \hat{b}_{i_2} \dots \hat{b}_{i_{n-q}} \hat{b}_{j_1} \hat{b}_{j_2} \dots \hat{b}_{j_{m-q}}, \quad (1.96)$$

siendo sus elementos

$$E_{i_1 i_2 \dots i_{n-q}, j_1 j_2 \dots j_{m-q}} = \sum_{k_1 k_2 \dots k_q} T_{i_1 i_2 \dots i_{n-q}, k_1 k_2 \dots k_q} R_{j_1 j_2 \dots j_{m-q}, k_1 k_2 \dots k_q}. \quad (1.97)$$

De esta definición podemos sacar que al contraer dos tensores por sus últimos q índices, lo que hacemos es sumar para todos los valores de estos últimos índices el producto de los elementos de un tensor con el otro, para un valor fijo en cada uno de los índices que no estamos contrayendo. Hay formas diferentes de contracciones, algunas implican más de dos tensores, con índices que conectan entre ellos de forma más general, un índice repetido en más de dos tensores o incluso repetidos en un mismo tensor. Suena lioso, pero vamos a poner unos ejemplos que hagan que esto tenga más sentido.

■ **Example 1.19 — Producto escalar de vectores.** El producto escalar de dos vectores \vec{v} y \vec{w} viene dado como

$$\vec{v} \cdot \vec{w} = \sum_i v_i w_i. \quad (1.98)$$

Esto sería una contracción de dos vectores a través de su único índice. ■

■ **Example 1.20 — Producto matriz-vector.** El producto de un vector \vec{v} y una matriz A viene dado como

$$A\vec{v} = \sum_j A_{ij} v_j \hat{b}_i. \quad (1.99)$$

Esto sería una contracción de una matriz y un vector a través de su último índice. ■

■ **Example 1.21 — Producto matriz-matriz.** El producto de una matriz A y una matriz B viene dado como

$$AB = \sum_k A_{ik} B_{kj} \hat{b}_i \hat{b}_j \quad (1.100)$$

Esto sería una contracción de dos matrices a través de sus índices segundo y primero. ■

■ **Example 1.22 — Traza de una matriz.** La traza de una matriz A viene dada como

$$Tr(A) = \sum_i A_{ii} \quad (1.101)$$

Esto sería una contracción de una matriz consigo misma a través de sus índices repetidos. ■

Debido a esta operación, podemos ver que un tensor también sirve como mapeo entre dos espacios vectoriales. Esto es, un tensor recibe otro tensor y devuelve un tercer tensor. Podemos exponer en la Tabla 1.3.

Como podemos observar, por cada índice contraído estamos mezclando componentes del tensor de entrada, reduciendo en 1 el orden del tensor de salida. Por otro lado, los índices que no han sido contraídos aumentarán en 1 el orden del tensor de salida.

Otra operación importante en los tensores es el cambio de orden, usualmente llamado redimensionado o ‘reshape’ en lenguaje de arrays. Esto consiste en realizar un mapeado de los n índices del tensor actual a un conjunto de m índices diferente, de forma que cada combinación de los índices iniciales tenga una y solo una combinación de índices finales dada. Además, deben estar presentes todos los elementos al final. Por tanto, necesitamos una función que podamos deshacer y recuperar el tensor inicial a partir del nuevo.

Orden tensor	Orden de entrada	N° índices contraídos	Orden de salida
0 (escalar)	0 (escalar)	0 (producto de escalares)	0 (escalar)
1 (vector)	0 (escalar)	0 (producto por escalar)	1 (vector)
1 (vector)	0 (escalar)	1 (suma de componentes)	0 (escalar)
1 (vector)	1 (vector)	0 (producto tensorial)	2 (matriz)
1 (vector)	1 (vector)	1 (producto escalar)	0 (escalar)
2 (matriz)	0 (escalar)	0 (producto por escalar)	2 (matriz)
2 (matriz)	0 (escalar)	1 (suma de componentes)	1 (vector)
2 (matriz)	0 (escalar)	2 (traza)	0 (escalar)
2 (matriz)	1 (vector)	0 (producto tensorial)	3
2 (matriz)	1 (vector)	1 (producto vector-matriz)	1 (vector)
2 (matriz)	1 (vector)	2 (producto vector-matriz y suma de componentes)	0 (escalar)
2 (matriz)	2 (matriz)	0 (producto tensorial)	4
2 (matriz)	2 (matriz)	1 (producto matricial)	2 (matriz)
2 (matriz)	2 (matriz)	2 (producto matricial y traza)	0 (escalar)
n	0 (escalar)	0 (producto por escalar)	n
n	1 (vector)	0 (producto tensorial)	n+1
n	1 (vector)	1	n
n	2 (matriz)	0 (producto tensorial)	n+2
n	2 (matriz)	1	n+1
n	2 (matriz)	2	n

Table 1.3: Tabla de tensores obtenidos tras contraer con otro tensor.

■ **Example 1.23 — Reshape matriz a vector.** Dada una matriz A de $N \times M$, podemos crear un vector \vec{v} de dimensión NM tal que contenga sus elementos a modo de reshape. Por poner un ejemplo, consideremos la matriz

$$A = \begin{pmatrix} 2 & 3 & 4 \\ -1 & 8 & 0 \end{pmatrix}. \quad (1.102)$$

Un reshape simple de dicha matriz sería un vector

$$\vec{v} = (2, 3, 4, -1, 8, 0), \quad (1.103)$$

donde se ve que cada elemento de la matriz A tiene una y solo una posición en el vector \vec{v} y están todos presentes. ■

Como caso particular, todo tensor de orden n y dimensiones D_i para el índice i -ésimo puede ponerse como un vector de dimensión $\prod_{i=1}^n D_i$. Esto se suele llamar aplanamiento o ‘flatten’. Normalmente los tensores suelen redimensionarse a matrices, para su interpretación sencilla. Esto lo veremos más en detalle con las puertas multiqubit en su sección correspondiente.

1.5.3 Tensores importantes

Además de los tensores que veremos en la siguiente parte como puertas multiqubit, algunos tensores importantes son:

Delta de Kronecker

La Delta de Kronecker $\delta_{i,j,\dots,k}$ es un tensor que generaliza el concepto de la identidad a un mayor número de índices. Todos sus elementos son nulos, salvo aquellos en los cuales tengamos que todos sus índices son iguales, en los cuales tendremos 1.

Este tensor será muy importante, ya que cada vez que aparezca en un conjunto de sumatorios, hará que todos los índices compartidos con él tomen el mismo nombre, borrando todos salvo uno de los sumatorios asociados.

Símbolo de Levi-Civita

Este es un objeto cuyas componentes vienen dadas por permutaciones, y es especialmente útil en cálculos con rotacionales.

Sus elementos serán nulos cuando 2 o más de sus índices coincidan, 1 cuando sean todos diferentes y haya un número par de permutaciones entre ellos y -1 cuando haya un número impar de permutaciones. Para un símbolo de Levi-Civita de orden n tendremos que los índices estarán en el intervalo $[1, n]$. Veámoslo con unos ejemplos.

■ **Example 1.24 — Levi-Civita 2×2 .**

$$\epsilon_{ij} = \begin{cases} 1 & \text{si } (i, j) \text{ es } (1, 2) \\ -1 & \text{si } (i, j) \text{ es } (2, 1) \\ 0 & \text{si } i = j \end{cases} \quad (1.104)$$

■

■ **Example 1.25 — Levi-Civita 3×3 .**

$$\epsilon_{ijk} = \begin{cases} 1 & \text{si } (i, j, k) \text{ es } (1, 2, 3), (2, 3, 1), (3, 1, 2) \\ -1 & \text{si } (i, j, k) \text{ es } (2, 1, 3), (3, 2, 1), (1, 3, 2) \\ 0 & \text{si } i = j \text{ o } i = k \text{ o } j = k \end{cases} \quad (1.105)$$

■

2. Introducción física

En este capítulo realizaremos una introducción a los conceptos e interpretaciones físicas necesarias para la computación cuántica. Nos enfocaremos solamente en lo necesario para la algoritmia, por lo que omitiremos muchos conceptos de física y cuántica, como las cuantizaciones, las integrales de camino o el espacio de momentos. Nos centraremos en los aspectos más básicos y aplicables, como el qubit, las puertas cuánticas, el colapso de la función de onda o el entrelazamiento.

- 2.1 Estados físicos**
- 2.2 El qubit: superposición y probabilidad**
- 2.3 Ejemplos físicos de qubits**
- 2.4 Puertas cuánticas y operadores unitarios**
- 2.5 Estados cuánticos de varios qubits**
- 2.6 Entrelazamiento**
- 2.7 Valores esperados y operadores hermíticos**
- 2.8 Puertas multiqubit**
- 2.9 Medida y colapso del estado**
- 2.10 Errores cuánticos**

3. Matemática avanzada

En este capítulo vamos a abordar matemáticas más específicas, especializadas para los algoritmos que vamos a explicar en las siguientes partes. Nos referiremos a este capítulo en las secciones correspondientes.

- 3.1 Álgebra modular básica**
- 3.2 Álgebra de Lie básica**
- 3.3 Cálculo diferencial básico**
- 3.4 Operadores no unitarios**
- 3.5 Cuantización**
- 3.6 Matrices densidad**
- 3.7 Proyecciones**
- 3.8 Problemas QUBO**
- 3.9 Problema de Ising**

Parte 2: Algoritmos cuánticos

4	Algoritmos introductorios	51
4.1	Explorando las puertas	
4.2	Creando entrelazamiento	
4.3	Teleportación cuántica	
4.4	Obteniendo información colapso	
4.5	Aplicando operadores	
4.6	Obteniendo fases	
4.7	Oráculos y superposiciones	
5	Algoritmos intermedios	53
5.1	Buscando datos con Grover	
5.2	Resolviendo sistemas de ecuaciones	
5.3	Obteniendo periodos	
5.4	Aproximaciones variacionales	
5.5	Optimización de sistemas	
5.6	Evolución en tiempo imaginario	
6	Algoritmos avanzados	55
6.1	Optimización combinatoria	
6.2	Factorización de números	
6.3	Quantum Singular Value Decomposition	
6.4	Quantum Support Vector Machine	
6.5	Quantum Neural Networks	
6.6	Quantum Autoencoder	

4. Algoritmos introductorios

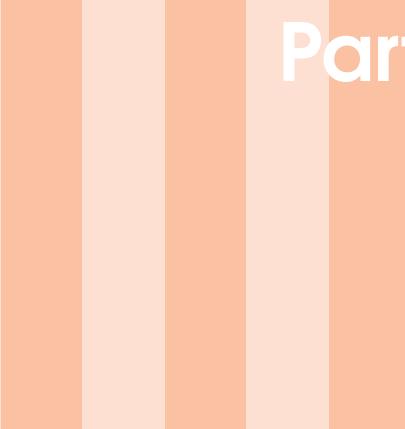
- 4.1 Explorando las puertas
- 4.2 Creando entrelazamiento
- 4.3 Teleportación cuántica
- 4.4 Obteniendo información colapso
- 4.5 Aplicando operadores
- 4.6 Obteniendo fases
- 4.7 Oráculos y superposiciones

5. Algoritmos intermedios

- 5.1 Buscando datos con Grover**
- 5.2 Resolviendo sistemas de ecuaciones**
- 5.3 Obteniendo periodos**
- 5.4 Aproximaciones variacionales**
- 5.5 Optimización de sistemas**
- 5.6 Evolución en tiempo imaginario**

6. Algoritmos avanzados

- 6.1 Optimización combinatoria**
- 6.2 Factorización de números**
- 6.3 Quantum Singular Value Decomposition**
- 6.4 Quantum Support Vector Machine**
- 6.5 Quantum Neural Networks**
- 6.6 Quantum Autoencoder**



Parte 3: Seguridad Cuántica

7 Distribución de claves cuánticas 59

- 7.1 Motivación
- 7.2 Teorema de no clonación
- 7.3 Protocolo BB84
- 7.4 Protocolo E91
- 7.5 Decoy State
- 7.6 Otros protocolos

8 Criptografía cuántica 61

- 8.1 Motivación
- 8.2 Dinero Cuántico

7. Distribución de claves cuánticas

7.1 Motivación

7.2 Teorema de no clonación

7.3 Protocolo BB84

7.4 Protocolo E91

7.5 Decoy State

7.6 Otros protocolos

8. Criptografía cuántica

8.1 Motivación

8.2 Dinero Cuántico

Parte 4: Computación Cuántica avanzada

9	Ordenadores cuánticos	67
9.1	Criterios de DiVincenzo	
9.2	Qubits físicos	
9.3	NISQ	
9.4	Qubits lógicos	
10	Tecnologías Quantum-Inspired	69
10.1	Quantum Annealing	
10.2	Tensor Networks	
11	Casos de uso	71
12	Out	73
	Bibliography	75
	Books	
	Articles	

En esta última parte del libro, veremos como los conceptos teóricos explicados hasta el momento se llevan a cabo en el mundo real. En el primer capítulo, comenzaremos explicando lo que es un ordenador cuántico, así como las diversas formas de implementar qubits y puertas cuánticas físicamente, incluyendo como corregir los errores que aparecen en estos sistemas. Seguidamente, introduciremos el concepto de tecnologías Quantum-Inspired, explicando las ventajas de estos métodos que, pese a no ser puramente cuánticos, pueden ofrecer múltiples ventajas frente a otros más tradicionales. Finalmente, haremos un repaso de las aplicaciones más prometedoras de la computación cuántica, detallando los distintos casos de uso y las ventajas que ofrece frente a la computación clásica.

9. Ordenadores cuánticos

En las tres primeras partes de este libro, hemos introducido los conceptos matemáticos y físicos básicos para entender la computación cuántica, y utilizando estos conceptos, hemos visto cómo diseñar algoritmos cuánticos que permitan superar a los clásicos de diferentes formas. Sin embargo, hay una pregunta que aún no hemos respondido. Esta pregunta es posiblemente la primera que se nos hace a todos aquellos que nos desempeñamos de una u otra forma en el campo de la computación cuántica, y quizás muchos de nosotros no hayamos respondido tan correctamente como creemos a pesar de la sencillez de la misma. Después de todo, ¿qué es un ordenador cuántico?

9.1 Criterios de DiVincenzo

Desde que a principios de los ochenta Yuri Manin y, posteriormente, Richard Feynman propusieran la idea de un simulador cuántico, comenzaron a aparecer multitud de métodos que, utilizando los principios de la mecánica cuántica, permitían computar de forma más eficiente que cualquier método clásico. Al menos, sobre el papel. Veinte años más tarde, en el año 2000, el físico teórico David DiVincenzo publicó una serie de criterios necesarios para implementar físicamente cualquiera de estos algoritmos. Estos cinco criterios de computación, más otros dos de comunicación, se conocen hoy en día como Criterios de DiVincenzo. En esta sección, enumeraremos dichos criterios, incluyendo el razonamiento detrás de los mismos, sirviendo como base para el resto de secciones de este capítulo.

Criterio 1

Un ordenador cuántico debe ser un sistema físico escalable con qubits bien caracterizados.

Criterio 2

Un ordenador cuántico debe poder ser inicializado en un estado fiduciario simple, como el $|000\dots\rangle$.

Criterio 3

Un ordenador cuántico debe poseer tiempos de decoherencia suficientemente largos, mayores que los tiempos de operación de las puertas cuánticas.

Criterio 4

Un ordenador cuántico debe ser capaz de implementar un conjunto universal de puertas cuánticas.

Criterio 5

Un ordenador cuántico debe tener la capacidad de medir qubits específicos.

9.2 Qubits físicos**9.2.1 Superconductores****9.2.2 Átomos neutros****9.2.3 Iones atrapados****9.2.4 Fotones****9.2.5 Computación cuántica topológica****9.3 NISQ****9.3.1 Computación híbrida****9.4 Qubits lógicos****9.4.1 Corrección de errores**

10. Tecnologías Quantum-Inspired

10.1 Quantum Annealing

10.2 Tensor Networks

11. Casos de uso

12. Out

Theorem 12.0.1 — Name of the theorem. In $E = \mathbb{R}^n$ all norms are equivalent. It has the properties:

$$|||\mathbf{x}|| - ||\mathbf{y}||| \leq ||\mathbf{x} - \mathbf{y}|| \quad (12.1)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (12.2)$$

R The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

Corollary 12.0.2 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

Proposition 12.0.3 — Proposition name. It has the properties:

$$|||\mathbf{x}|| - ||\mathbf{y}||| \leq ||\mathbf{x} - \mathbf{y}|| \quad (12.3)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (12.4)$$

Proposition 12.0.4 Let $f, g \in L^2(G)$; if $\forall \varphi \in \mathcal{D}(G)$, $(f, \varphi)_0 = (g, \varphi)_0$ then $f = g$.

Exercise 12.1 This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ■

Problem 12.1 What is the average airspeed velocity of an unladen swallow?

Vocabulary 12.1 — Word. Definition of word.

Bibliography

Books

Articles

