

# Relación carga masa del electrón

Sergio Montoya Ramirez\* and Luisa Lopez\*\*

Universidad de los Andes, Bogotá, Colombia.

(Dated: 22 de noviembre de 2025)

TODO

## I. INTRODUCCIÓN

Uno de los puntos mas difíciles para convertir la computación cuántica en una realidad es lo sensibles que son los qubits físicos. Dada la naturaleza cuántica de estos tener un estado y actuar sobre el sin que colapse a una medición o sea alterado por variables externas (por ejemplo, temperatura, cambios electromagnéticos, etc) hace que la necesidad de manejar errores en computación cuántica sea de primera categoría. Al mismo tiempo, el teorema de no clonación causa que estrategias clásicas de corrección de errores, como la redundancia en la información sean imposibles. Sin embargo, en 1995 (y posteriormente 1996) Shor y Calderbank encontraron una manera de lidiar con esto. Lo que podían hacer era distribuir la información en varios qubits y crear un código que interpretara de manera correcta estos qubits para recuperar la información original. Este proyecto tiene como objetivo resumir muy brevemente algunos de los puntos mas importantes del campo "quantum error correction" (QEC). Para cumplir este objetivo se harán 3 paradas importantes. Iniciaremos con un ejemplo de código de corrección que nos permita darnos una idea de los patrones que se seguirán en todo el campo. Luego, miraremos a una teoría general de errores que nos dará la fundamentación matemática necesaria para entender el teorema de la distancia de hamming (que nos permite determinar si un código puede corregir o no un ruido) y el formalismo de estabilizadores. Por último, veremos ejemplos de códigos muy importantes pertenecientes a la familia de códigos estabilizadores como lo son el código de Shor y la familia de códigos CSS. Esta ruta, si bien se encuentra limitada por el espacio y tiempo con el que cuenta permitirá darse una idea estructurada de las bases de este campo.

## II. EJEMPLO: PHASE FLIP

Imaginen usted que tiene un canal con un ruido que con probabilidad  $p$  hace un *phase flip* en su qubit. Esto quiere decir que con probabilidad  $p$  convierte a  $|\psi\rangle$

en  $X|\psi\rangle$ . Para este caso entonces podemos crear nuestro primer código, básicamente cambiando a una base conocida con elementos

$$|0_L\rangle = |000\rangle; |1_L\rangle = |111\rangle.$$

Es decir, el elemento  $|\psi\rangle = a|0\rangle + b|1\rangle$  se convierte en  $a|0_L\rangle + b|1_L\rangle$ .

Ahora bien, también tenemos que acompañar esto con un procedimiento que nos permita determinar cual de todos los qubits fue en el que ocurrió el cambio. Para esto entonces vamos a medir sin hacer distinciones entre estados (de modo que no destruya nuestro qubit) con:

$$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$$

$$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$$

$$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$$

$$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$$

De esta manera podemos saber en cual de los qubits ocurrió y por tanto podemos aplicar de nuevo  $X$  para regresar al estado original.

Ahora bien, como sabemos que este procedimiento funciona mejor que el caso original. Para esto definamos entonces lo que es fidelidad de modo que podamos ver cual es el peor caso en este canal. Para un canal con acción  $\rho$  la fidelidad es

$$F = \sqrt{\langle\psi|\rho|\psi\rangle} \quad (1)$$

donde  $|\psi\rangle$  es el estado que nos interesa (normalmente es el mismo estado inicial). Entonces, note que este canal se comporta como

$$\rho = (1-p)|\psi\rangle\langle\psi| + (p)X|\psi\rangle\langle\psi|X.$$

Por lo tanto tiene una fidelidad de:

$$F = \sqrt{(1-p) + p\langle\psi|X|\psi\rangle\langle\psi|X|\psi\rangle}$$

Note que  $\langle\psi|X|\psi\rangle \geq 0$  por lo tanto, en general, la fidelidad mínima será definida por la amplitud de probabilidad en el estado que nos interesa. En nuestro caso original  $F = \sqrt{1-p}$ . Por otro lado, miremos que para el caso de nuestro código nos interesa únicamente aquellos procesos donde hayan ocurrido 1 error o menos (puesto que nuestro código solo cuenta para ese caso). La

\* Correo institucional: s.montoyar2@uniandes.edu.co

\*\* Correo institucional: usuario@uniandes.edu.co

probabilidad de que esto ocurra así es la probabilidad de que no ocurra ninguna vez (Es decir  $(1-p)^3$ ) y 3 veces la probabilidad de que ocurra una sola vez (es decir  $3p(1-p)^2$ ) por lo cual la probabilidad total es  $(1-p)^3 + 3p(1-p)^2 = 1 - 3p^2 + 2p^3$  y en consecuencia esto sería una fidelidad mínima de  $F = \sqrt{1 - 3p^2 + 2p^3}$  que es menor que la original siempre y cuando  $p < \frac{1}{2}$ .

Ahora bien, este ejemplo nos da algunas de las bases necesarias para comprender el tema de QEC. Para empezar es importante que nos da una idea, no formal, de lo que es un código. En este caso podemos ver como un subconjunto de  $\mathcal{B}^n$  que representa la información codificada. En el caso particular de los códigos de corrección de errores viene además con un procedimiento que corrige el error específico que se presenta en el canal. Con estas ideas, vamos a definir un poco más formalmente con lo que estamos trabajando.

### III. FORMALISMOS

En el caso anterior vimos un ejemplo muy concreto de un código que mejoraba la fiabilidad en un canal con un ruido concreto. Sin embargo, este ejemplo particular se aprovechaba de las características del ruido para representar la corrección del error y el aumento en la fiabilidad. Ahora nos interesa construir una teoría más general de los errores. Sea  $p$  un estado codificado en  $C \subseteq \mathcal{B}^n$ . Además, sea  $\mathcal{E}$  un procedimiento que define el error. Decimos que  $C$  corrige  $\mathcal{E}$  si existe un procedimiento  $\mathcal{R}$  tal que

$$(\mathcal{R} \circ \mathcal{E})(p) \propto p \quad (2)$$

Una nota importante que ver es que el  $\propto$  nos representa que realmente la operación de recuperación no nos devuelve exactamente al qubit original. Sin embargo, si nos debe devolver a un qubit que codifique la misma información que el anterior.

Ahora bien, como podemos saber si un código corrige un error.