Universidad Nacional Autónoma de México

Facultad de Ciencias

Errores unitarios locales en el algoritmo de Shor

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
FÍSICO

PRESENTA:
JOSÉ FRANCISCO MORALES HERNÁNDEZ

DIRECTOR DE TESIS:
CARLOS FRANCISCO PINEDA ZORRILLA

2013

1. Datos del alumno
   Morales
   Hernández
   José Francisco
   55 16 88 03 56
   Universidad Nacional Autónoma de México
   Facultad de Ciencias
   Física
   409099906

2. Datos del tutor
   Dr
   Carlos Francisco
   Pineda
   Zorrilla

3. Datos del sinodal 1
   Dr
   Luis
   de la Peña
   Auerbach

4. Datos del sinodal 2
   Dr
   Ramón
   López
   Peña

5. Datos del sinodal 3
   Dr
   Thomas Henry
   Seligman
   Schurch

6. Datos del sinodal 4
   Dr
   Pablo
   Barberis
   Blostein

7. Datos del trabajo escrito
   Errores unitarios locales en el algoritmo de Shor
   44 p
   2013

Agradecimientos

Recibi apoyo de los proyectos CONACyT 153190 y UNAM-PAPIIT IA101713.

II

Resumen

En computación cuántica ciertos algoritmos pueden ser realizados en un tiempo exponencialmente más rápido que su realización clásica, uno de estos es el algoritmo de factorización de Shor. Una de las partes fundamentales del algoritmo de Shor es la aplicación de la *transformada cuántica de Fourier* inversa (QFT inversa).

En este trabajo se hace un estudio numérico de como se modifica la probabilidad de éxito del algoritmo cuando se perturban las compuertas de dos qubits en la QFT inversa bajo un modelo de *teoría de matrices aleatorias*.

Idealmente se tiene que mientras mayor es el número de qubits la probabilidad de éxito del algoritmo es mayor pero se encontró que esto no es cierto cuando hay perturbaciones. De esta manera se pudo concluir que el comportamiento de la probabilidad de éxito del algoritmo tiene semejanza con el de la fidelidad de la QFT inversa y que hay un número de qubits óptimo para el cual la probabilidad de éxito es máxima.

Índice general

Capítulo 1

Introducción

El descubrimiento más espectacular en computo cuántico hasta la fecha es que las computadoras cuánticas pueden realizar eficientemente algunas tareas que no son factibles en una computadora clásica. Un ejemplo es el algoritmo de factorización de Shor [1]. El algoritmo de Shor es un procedimiento que nos ayuda a encontrar los factores de un número que es un producto de dos números primos. Esto ha motivado que experimentalmente se construyan dispositivos que manipulen qubits [2] (bits cuánticos) en diversos sistemas. Es critico para poder aprovechar el poder de las computadoras cuánticas tener la habilidad de controlar y manipular precisamente estados cuánticos por medio de compuertas cuánticas [3], y éstas son implementadas con dispositivos físicos. En la realidad, sin embargo, estos dispositivos no son perfectos principalmente por dos motivos: la inevitable interacción con el medio ambiente y la imprecisión de los dispositivos físicos en la aplicación de las compuertas cuánticas. La interacción con el medio ambiente se conoce como *decoherencia* [4] y causa una perdida de información sobre el estado cuántico.

En el presente trabajo estudiaremos numéricamente cuáles son las consecuencias, sobre la probabilidad de éxito del algoritmo de Shor, de considerar perturbaciones en la parte cuántica del algoritmo. Las perturbaciones que estudiamos son errores unitarios locales que representan errores debido a la imprecisión de los dispositivos en la aplicación de las compuertas. No consideramos errores debido a decoherencia pues es más difícil modelar este tipo de errores. Sin embargo, en algunos casos decoherencia puede ser descrita por una evolución unitaria aleatoria sin necesidad de considerar el medio ambiente [5], es decir, de la misma forma en la que modelamos los errores debido a la imprecisión de los dispositivos en la aplicación de las compuertas.

A pesar de lo dicho anteriormente ya existe una computadora cuántica. En de Mayo de 2011 D-Wave Systems anunció el D-Wave One, una

computadora cuántica integrada con un procesador de 128 qubits y basada en computo cuántico adiabático[1]. El procesador usado realiza una sola operación matemática llamada optimización discreta, es decir no cuenta con la universalidad de una computadora clásica.

En el capítulo 2 damos los conceptos necesarios para entender el algoritmo de Shor; conceptos como la transformada cuántica de Fourier, el algoritmo de estimación de fase y la exponenciacion modular [6]. Esta sección está basada principalmente en la referencia [7]. También introducimos el concepto de fidelidad [8, 9] como medida de la estabilidad de un operador cuántico ante perturbaciones.

En el capítulo 3 hablamos de nuestro modelo para estudiar las perturbaciones en el algoritmo. Este modelo consiste en perturbar las compuertas de dos qubits en la transformada cuántica de Fourier inversa, debido a que las compuertas de un qubit pueden ser llevadas acabo experimentalmente con una precisión lo suficientemente grande para la implementación de protocolos cuánticos de corrección [10]. Aunque tanto la QFT inversa como la exponenciacion modular son las operaciones más importantes en el algoritmo de Shor, sólo nos ocupamos en perturbar la QFT inversa por simplicidad. Estudiamos dos tipos de perturbaciones: una estática y otra dinámica.

En el capítulo 4 presentamos los resultados obtenidos de considerar el algoritmo de Shor perturbado con ambos tipos de perturbaciones. Con las simulaciones observamos es la probabilidad de éxito de la factorización tiene un comportamiento similar al comportamiento de la fidelidad de la QFT inversa. Un resultado importante es que cuando el algoritmo es perturbado, un mayor número de qubits no implica necesariamente una probabilidad de éxito mayor en la factorización; esto es sobresaliente porque en el caso del algoritmo no perturbado un mayor número de qubits sí implica una probabilidad de éxito mayor.

Finalmente en el capítulo 5 presentamos las conclusiones y sugerimos direcciones para trabajo futuro.

[1]Debido al método de computo cuántico utilizado en D-Wave One se especula si esta en realidad puede sobrepasar el poder de computo de una computadora clásica.

Capítulo 2

Algoritmo de Shor

El algoritmo de factorización de Shor es uno de los principales ejemplos en el cual una computadora cuántica demuestra un enorme poder sobrepasando a su contraparte clásica [7]. Pero, ademas de esto, ¿por qué es interesante este algoritmo? Esto es debido a que un sistema ampliamente usado para codificar y decodificar mensajes (un ejemplo se da en A.2) está basado en el hecho de que factorizar números grandes es muy difícil. Este es el sistema criptográfico de clave publica RSA[2], por lo que una factorización rápida es ventajosa.

El algoritmo de Shor es casi idéntico al clásico excepto por un paso, el cual es remplazado por un algoritmo cuántico donde tanto la *transformada cuántica de Fourier (QFT)* como la *exponenciacion modular* son fundamentales. Antes de adentrarnos más en el algoritmo hay conceptos que se tienen que saber para comprenderlo mejor.

A continuación se hablará de las *transformadas integrales cuánticas* para encaminarnos hacia una definición de la QFT, la cual a su vez nos ayudará a comprender un procedimiento conocido como el *algoritmo de estimación de fase* y finalmente discutiremos el algoritmo de Shor. También se hablará de la *fidelidad* como medida de estabilidad de un operador cuántico pues en el próximo capítulo nos ocuparemos de perturbar el algoritmo de Shor y nos interesa conocer cual es el efecto de esta perturbación sobre la probabilidad de tener una factorización exitosa.

[2]RSA debido a las iniciales de los autores que describieron el algoritmo en 1977, Ron Rivest, Adi Shamir y Leonard Adleman. En la sección A.2 se muestra un ejemplo de RSA.

2.1. Transformadas integrales cuánticas

Sea U una matriz unitaria que actúa sobre un espacio de n qubits $H = (\mathbb{C}^2)^{\otimes n}$. Supón que U actúa en un vector base $|x\rangle$ como [7]:

$$U|x\rangle = \sum_{y=0}^{D-1} K(y,x)|y\rangle \tag{2.1}$$

donde K es el kernel de alguna transformada integral discreta y $D = 2^n$ para simplificar la notación.

Entonces U actúa en una combinación lineal de los elementos de la base como:

$$U\left[\sum_{x=0}^{D-1} f(x)|x\rangle\right] = \sum_{y=0}^{D-1} \tilde{f}(y)|y\rangle \tag{2.2}$$

donde $\tilde{f}(y) = \sum_{x=0}^{D-1} K(y,x)f(x)$.

Nótese que la matriz unitaria U realiza la transformada integral discreta $\tilde{f}(y)$ para todas las variables y en una sola operación si actúa en el estado $\sum_{x=0}^{D-1} f(x)|x\rangle$. Hay un numero exponencialmente grande (2^n) de números ys para un registro de n qubits, y este hecho provee a una computadora cuántica con un poder de computo exponencialmente más rápido para cierta clase de operaciones comparado con alternativas clásicas.

La matriz unitaria U que implementa una transformada integral discreta como en la Ec. (2.2) es llamada la *transformada integral cuántica*.

2.1.1. Transformada cuántica de Fourier (QFT)

La QFT es una de las transformadas integrales cuánticas más importantes, es un algoritmo cuántico eficiente para realizar una transformada de Fourier de las amplitudes cuánticas de una combinación lineal de los elementos de la base. No acelera la tarea clásica de realizar transformadas de Fourier de datos clásicos puesto al realizar la transformada sobre la densidad de probabilidad del estado y al colapsar, esta información se pierde. El kernel de la QFT esta definido como:

$$K(x,y) = \frac{1}{\sqrt{D}} e^{\frac{2\pi i x y}{D}} \tag{2.3}$$

y la transformada integral discreta con este kernel es llamada la *transformada discreta de Fourier*:

$$\tilde{f}(y) = \sum_{x=0}^{D-1} \frac{1}{\sqrt{D}} e^{\frac{2\pi i x y}{D}} f(x). \tag{2.4}$$

Con un poco de álgebra la transformada cuántica de Fourier puede dar la siguiente *representación de producto* (ver A.1):

$$|j_1, ..., j_n\rangle \rightarrow \frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1\cdots j_n}|1\rangle\right)}{2^{n/2}}. \quad (2.5)$$

Como se verá adelante esta representación de producto es muy útil.

2.1.2. Circuito de la QFT

La representación de la Ec. (2.5) nos permite construir un circuito cuántico eficiente que realice la transformada cuántica de Fourier. Esta es una prueba de que la transformada cuántica de Fourier es unitaria y proporciona información sobre los algoritmos basados en la QFT.

Pero, ¿Por qué nos interesa conocer este circuito? Para obtener el estado final que se muestra en la ecuación 2.5 se necesita manipular el estado inicial. La manipulación de estados cuánticos, en el modelos de computo cuántico usado aquí, se realiza mediante operaciones unitarias de uno y dos qubits a las que llamaremos compuertas, y estas a su vez representan operaciones físicas que como se verá adelante son susceptibles a errores. En este trabajo se modelarán dichos errores colocando una perturbación en las compuertas, por lo que es necesario conocer el circuito.

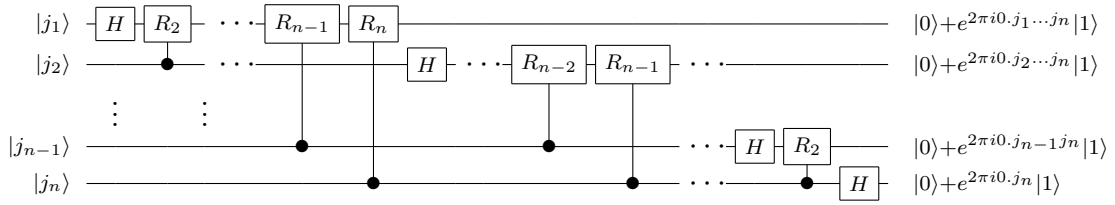

Figura 2.1: Circuito eficiente para la QFT. Este circuito es derivado fácilmente de la representación de producto (2.5). No se muestran ni las compuertas Swap al final del circuito, las cuales invierten el orden de los qubits $|a_1, a_2, ..., a_n\rangle \rightarrow |a_n, a_{n-1}, ..., a_1\rangle$, ni los factores de normalización $\frac{1}{\sqrt{2}}$ en la salida.

Donde el *operador de rotación de fase* R_k denota la transformación unitaria

$$R_k \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \quad (2.6)$$

y la compuerta *Hadamard* H

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (2.7)$$

Si se tiene un operador U entonces el circuito para una operación controlada Ctrl-U es

cuya representación matricial es:

$$\text{Ctrl-}U \equiv \begin{pmatrix} \mathbb{1} & 0 \\ 0 & U \end{pmatrix}. \tag{2.8}$$

En el circuito tenemos operaciones Ctrl-R_k.

La QFT es realizada en $O(n^2)$ operaciones. Si se quiere ver como es que el este circuito realiza la QFT vease el apéndice A.1.

2.2. Algoritmo de estimación de fase

La transformada de Fourier es la clave para un proceso general conocido como *estimación de fase*, el cual a su vez es la clave para muchos otros algoritmos cuánticos. Supón que un operador unitario U tiene un eigenvector $|u\rangle$ con eigenvalor $e^{2\pi i\varphi}$, donde el valor de φ es desconocido. El objetivo del algoritmo de estimación de fase es estimar φ [3].

El proceso cuántico de estimación de fase usa dos registros. El primer registro contiene n qubits inicialmente en el estado $|0\rangle$. La forma en la que elegimos n depende de dos cosas: el numero de dígitos de precisión que deseamos tener en nuestra estimación de φ, y con que probabilidad deseamos que el proceso de estimación de fase sea exitoso.

El segundo registro comienza en el estado $|u\rangle$, y contiene tantos qubits como sea necesario para almacenar $|u\rangle$. La estimación de fase es realizada en dos etapas. Primero, aplicamos el circuito mostrado en la figura 2.2. El estado final del primer registro es:

$$\frac{1}{2^{\frac{n}{2}}}\big(|0\rangle + e^{2\pi i 2^{n-1}\varphi}|1\rangle\big)\big(|0\rangle + e^{2\pi i 2^{n-2}\varphi}|1\rangle\big)...\big(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\big) = \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{2\pi i\varphi k}|k\rangle. \tag{2.9}$$

Supóngase que φ puede ser expresado exactamente con n bits, si expresamos a $\varphi = \frac{\varphi_1}{2^1}+\frac{\varphi_2}{2^2}+\cdots+\frac{\varphi_n}{2^n}$ $(\varphi_l=0,1)$ en representación binaria se tiene que $\varphi = 0.\varphi_1...\varphi_n$. Entonces el estado (2.9) resultante de la primera etapa de la estimación de fase puede ser reescrito como

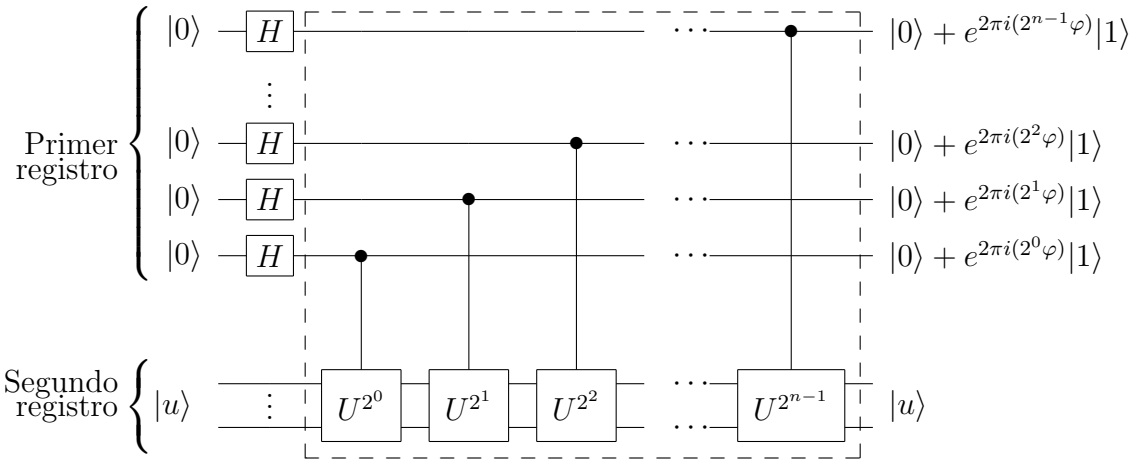

Figura 2.2: Primera etapa del proceso de estimación de fase. Los factores de normalización $\frac{1}{\sqrt{2}}$ han sido omitidos a la derecha. La operación en el recuadro puede ser vista como una operación U_f sobre todo el espacio.

$$\frac{1}{2^{\frac{n}{2}}}\big(|0\rangle + e^{2\pi i 0.\varphi_n}|1\rangle\big)\big(|0\rangle + e^{2\pi i 0.\varphi_{n-1}j_n}|1\rangle\big)...\big(|0\rangle + e^{2\pi i 0.\varphi_1 j_2...\varphi_n}|1\rangle\big). \quad (2.10)$$

La segunda etapa de la estimación de fase es aplicar al primer registro la inversa de la QFT, esta es obtenida de invertir el orden del circuito de la figura 2.1. Comparando la ecuación previa con la representación producto de la QFT, ecuación (2.5), se puede ver que el estado de salida de la segunda etapa es el estado producto $|\varphi_1...\varphi_n\rangle$. Por lo tanto una medición en la base computacional nos da exactamente φ. En otro caso solo se puede obtener una aproximación de esta fase ($\tilde{\varphi}$).

Para obtener exitosamente φ precisa a k bits con probabilidad de éxito de al menos $1-\epsilon$ elegimos el numero de qubits del primer registro de la siguiente manera [3]

$$n = k + \left\lceil \log\left(2 + \frac{1}{2\epsilon}\right)\right\rceil. \quad (2.11)$$

Esta relación es importante pues nos dice que mientras más qubits haya en el primer registro mayor es la probabilidad de obtener φ. Cuando hagamos la simulación del algoritmo perturbado con distintos números de qubits veremos si ésta se sigue satisfaciendo.

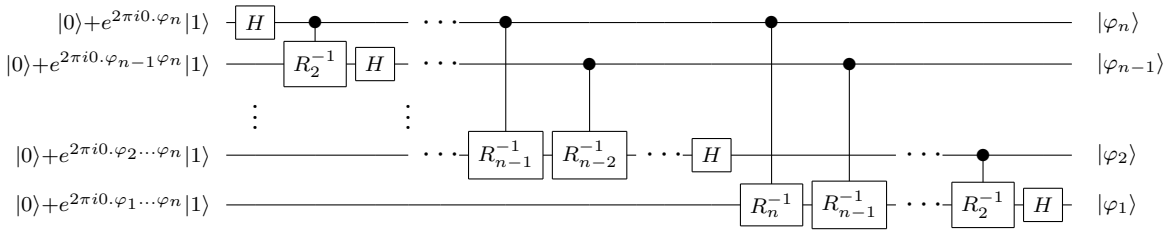

Figura 2.3: Segunda etapa del proceso de estimación de fase. Circuito cuántico para realización de la QFT inversa, $QFT^{\dagger}$. Los factores de normalización $\frac{1}{\sqrt{2}}$, a la izquierda, como las operaciones Swap, a la derecha, han sido omitidos.

2.3. Algoritmo de Shor

Hasta ahora solo se ha dado una introducción a conceptos necesarios de computo cuántico. Son necesarios pues como ya se dijo el algoritmo de Shor consta de dos partes, una clásica y otra cuántica. A continuación hablaremos del algoritmo de factorización usado en el algoritmo de Shor y después la parte cuántica del algoritmo de Shor.

2.3.1. Algoritmo de Shor

El problema de factorización que analizaremos en el trabajo es el siguiente. Sean p y q números primos y sea $N = pq$. Queremos expresar N como un producto de p y q. El mejor algoritmo clásico conocido para realizar esta tarea es el *general number field sieve*, el cual trabaja en un tiempo sub-exponencial[3] realizando $O(e^{(\log\ N)^{1/3}\ (\log\ \log\ N)^{2/3}})$ pasos antes de que p y q sean encontrados. Este algoritmo sigue siendo ineficiente para factorizar numero grandes. El siguiente algoritmo es el más adecuado para nuestro propósito [7].

PASO 1: Se toma aleatoriamente un entero positivo m menor que N y mayor que 1. Si $\gcd(m, N) \neq 1$, entonces m es p o q. Supón que m y N son coprimos, es decir, $\gcd(m, N) = 1$; entonces proseguimos al siguiente paso[4].

PASO 2: Se define[5] $f_N : \mathbb{N} \to \mathbb{Z}/N\mathbb{Z}$ por $a \to m^a \bmod N$. Ahora, debemos

[3]El termino sub-exponencial es usado para expresar que el tiempo de computo del algoritmo puede crecer más rápido que cualquier tiempo polinomial pero es significativamente menor que uno exponencial.

[4]El máximo común divisor.

[5]$\mathbb{Z}/N\mathbb{Z}$ representa el conjunto de clases de equivalencia en donde x y $x + kN$ $(k \in \mathbb{Z})$ son equivalentes. Claramente, podemos tomar x que satisfaga $0 \leq x \leq N - 1$ como un representante de cada clase de equivalencia.

encontrar el $r \in \mathbb{N}$ más pequeño tal que

$$f_N(r) \equiv 1 \text{mod } N. \tag{2.12}$$

El numero r es llamado el *periodo*. Este es el único pasa en el que se necesitará una computadora cuántica.

PASO 3: Si r es impar regresar al paso 1 pues no servirá para la factorización. Si r es par, seguir al paso 4.

PASO 4: Se cumple que

$$(m^{r/2} - 1)(m^{r/2} + 1) = m^r - 1 \equiv 0 \text{ mod } N. \tag{2.13}$$

Si $m^{\frac{r}{2}} + 1 \equiv 0 \text{ mod } N$ regresar al paso 1. Si $m^{\frac{r}{2}} + 1 \neq 0 \text{ mod } N$ entonces $\gcd(m^{\frac{r}{2}} - 1, N)$ es p o q, y la factorización está hecha.

Ejemplo de factorización clásica

Con la ayuda de *Mathematica* realizamos los cálculos para este ejemplo de factorización clásica. El número que queremos expresar como un producto de dos números primos es $N = 471088873$. Seguimos el procedimiento explicado.

PASO 1: Tomamos aleatoriamente un entero positivo m menor que N y mayor que 1. Supongamos que el resultado es $m = 426342592$, el cual es coprimo con N.

PASO 2: Tenemos que encontrar el periodo r de la función

$$f_N(x) = m^x \text{ mod } N = 426342592^x \text{ mod } N.$$

El periodo resultante de la computación es $r = 235522730$.

PASO 3: Debido a que r es par podemos continuar al siguiente paso.

PASO 4: Se tiene que $m^{\frac{r}{2}} + 1 \neq 0 \text{ mod } N$ entonces $p = \gcd(m^{\frac{r}{2}} - 1, N) = 22031$ y $q = \frac{N}{p} = 21383$ son los coprimos tales que $N = pq$, y la factorización esta realizada.

En el paso 2 el periodo r se encontró probando que números menores que N cumplían con la condición $m^r = 1\text{mod } N$ y eligiendo al menor. El tiempo de computo depende de que tan grande es el periodo comparado con el numero que se quiere factorizar y sabemos que r siempre es menor que $N/2$ para cualquier coprimo [11]. Encontrar el periodo tomó un tiempo de 62.3 minutos en mi computadora personal. Éste ya es un tiempo considerable y el número no es tan grande por lo que para números no mucho mayores a éste el tiempo será un problema. En la sección A.3 se muestra el algoritmo utilizado en este ejemplo.

2.3.2. Parte cuántica del algoritmo de Shor

Debe enfatizarse que un proceso cuántico solo es requerido en el paso 2 del algoritmo de Shor. Esta parte está basada en el algoritmo de estimación de fase y el objetivo es encontrar el periodo r de la función $f_N(x)$, es decir el numero r más pequeño que cumpla con la ecuación 2.12.

El procedimiento que seguiremos para encontrar este periodo r es el siguiente:

1. Primero hay tenemos que es establecer el estado inicial en ambos registros. El estado inicial es
$$|\psi_0\rangle = |0\rangle|0\rangle.$$

2. Se aplica la compuerta Hadamard en cada qubit del primer registro, creando una superposición.
$$|\psi_1\rangle = (H^{\otimes n} \otimes I)|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|0\rangle.$$

3. Suponemos que la compuerta unitaria U_f actúa como
$$U_f|x\rangle|0\rangle = |x\rangle|f_N(x)\rangle = |x\rangle|m^x \bmod N\rangle.$$
Esta etapa recibe el nombre de *exponenciación modular*. Para realizar esta operación en términos de compuertas de uno y dos qubits es necesario un registro temporal con $2n+1$ qubits [6]. Debido a la necesidad de un registro adicional y de que las operaciones necesarias son $O(n^3)$ [6] consideramos que es muy difícil realizar la simulación numérica de esta etapa, por lo que la consideraremos como ideal en el resto del trabajo a pesar de que es la operación más importante [12] del algoritmo perturbado. Si se quiere ver como se realiza esta operación en términos de operaciones elementales se puede consultar [6, 7].

   Así, tras la aplicación de U_f el estado es
$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f_N(x)\rangle.$$

4. Se aplica la QFT inversa en el primer registro
$$|\psi_3\rangle = (QFT^\dagger \otimes I)|\psi_2\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} e^{-2\pi i x y/2^n} |y\rangle|f_N(x)\rangle$$
$$= \frac{1}{2^n} \sum_{y=0}^{2^n-1} |y\rangle|\gamma(y)\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \||\gamma(y)\rangle\||y\rangle \frac{|\gamma(y)\rangle}{\||\gamma(y)\rangle\|},$$

donde

$$|\gamma(y)\rangle = \sum_{x=0}^{2^n-1} e^{-2\pi i x y/2^n} |f_N(x)\rangle.$$

5. Se mide el primer registro. El resultado y es medido con una probabilidad

$$\text{Prob}(y) = \frac{\||\gamma(y)\rangle\|^2}{2^{2n}},$$

y, al mismo tiempo, el estado colapsa a

$$|y\rangle \frac{|\gamma(y)\rangle}{\||\gamma(y)\rangle\|}.$$

En [7] se analiza esta probabilidad y se llega a que

$$\text{Prob}(y) = \begin{cases} \frac{r' \ \sin^2(\frac{\pi r y}{Q}(\frac{Q_0}{r}+1))+(r-r') \ \sin^2(\frac{\pi y Q_0}{Q})}{Q^2 \ \sin^2(\frac{\pi r y}{Q})} & (ry \neq 0 \bmod Q) \\ \frac{r'(Q_0+r)^2+(r-r')Q_0^2}{Q^2 r^2} & (ry = 0 \bmod Q), \end{cases} \tag{2.14}$$

donde $Q = 2^n = rq + r'$, $(0 \leq r' < r)$ y $Q_0 = rq$.

6. Después de medir el primer registro obtenemos y. Con este resultado, r puede ser obtenido eficientemente con la expansión en fracciones continuas (A.4) de $\frac{y}{2^n}$. Se elige el m-convergente más cercano [1] a $\frac{y}{2^n}$ y cuyo denominador sea menor[6] que N. Con cierta probabilidad este denominador es el periodo r de la función f_N.

2.4. Fidelidad

En el próximo capítulo nos ocuparemos de perturbar la parte cuántica del algoritmo de Shor por lo que debemos de tener un método por medio del cual conocer si el sistema es estable o no. Como criterio de estabilidad usaremos la fidelidad $F(t)$, definida como una distancia entre un estado obtenido por una evolución con un algoritmo ideal $|\psi(t)\rangle$ y uno perturbado obtenido por una evolución perturbada $|\psi^\delta(t)\rangle$:

$$F(t) = |\langle \psi^\delta(t) | \psi(t) \rangle| \tag{2.15}$$

[6]En [11] toman el m-convergente cuyo denominador sea menor que $N/2$, pues prueban que el periodo r es menor que este número. Por esta razón tienen una probabilidad mayor de obtener r que con el método usado aquí.

donde $|\psi(t)\rangle = U(t)|\psi(0)\rangle$ y $\left|\psi^{\delta}(t)\right\rangle = U^{\delta}(t)|\psi(0)\rangle$. A lo largo del trabajo se asumirá que el tiempo t es una variable discreta y entera que denota una unidad de tiempo del algoritmo, por ejemplo una compuerta, así $U(t) = U_t$. Debido a como se definió la fidelidad esta tendrá un valor de entre uno y cero, uno cuando $|\psi(t)\rangle = \left|\psi^{\delta}(t)\right\rangle$[7] y cero cuando sean completamente diferentes, este caso es cuando los estados son ortogonales.

Generalmente, puede ser útil promediar estadísticamente la fidelidad sobre un ensamble de estados iniciales $|\psi_k\rangle$, así:

$$F(t) = \frac{1}{l}\sum_{k=1}^{l} |\langle\psi_k^{\delta}(t)|\psi_k(t)\rangle| \tag{2.16}$$

donde $|\psi_k(t)\rangle = U_t|\psi_k\rangle$, $\left|\psi_k^{\delta}(t)\right\rangle = U_t^{\delta}|\psi_k\rangle$ y l es el tamaño del ensamble.

Así una medida del éxito del algoritmo completo es la fidelidad $F(t)$ a $t = T$, donde T denota el numero de compuertas del algoritmo.

A lo largo del trabajo este ensamble se considerará un ensamble de estados aleatorios. Ademas de éste, otro posible en ensamble es el de los estados producidos en el paso 3 de la parte cuántica del algoritmo de Shor.

Si en lugar de la relación 2.16 tomamos

$$f(t) = \frac{1}{l}\sum_{k=1}^{l} \langle\psi_k^{\delta}(t)|\psi_k(t)\rangle,$$

podemos relacionar la fidelidad con una traza de la siguiente forma:

$$f(t) = \mathrm{Tr}\left[\frac{1}{2^n}U_t U_t^{\delta}\right].$$

Aunque esta relación no la utilizaremos vale la pena mencionarla pues es común en la literatura.

[7] O cuando $|\psi(t)\rangle = e^{i\alpha}\left|\psi^{\delta}(t)\right\rangle$ pero este caso es equivalente pues son proporcionales por una fase global.

Capítulo 3

Modelo teórico

Los errores que se producen durante una operación cuántica se deben tanto a la interacción del sistema principal con el medio ambiente como a la imprecisión de los dispositivos físicos en la aplicación de compuertas y medidas cuánticas.

A la interacción del sistema principal con el medio ambiente se le denomina *decoherencia*. Ésta causa que generalmente la evolución del sistema principal no sea unitaria por lo que debe de considerarse como un sistema abierto.

Los errores ocasionados por la imprecisión de los dispositivos físicos en la aplicación de compuertas y medidas cuánticas son, en principio, más fáciles de modelar que los errores de decoherencia por lo que en este trabajo solo trataremos con errores de este tipo. En un el principio se trabajó con la hipótesis de que los errores en cada qubit son independientes [13]. Esta hipótesis es realista para los errores debido a la imprecisión de los dispositivos en la aplicación de compuertas y medidas cuánticas de un qubit [14] pero no para compuertas de dos qubits que son en los que nos concentraremos en este trabajo. Estos errores se consideran independientes y se modelan mediante operadores unitarios cercanos al operador identidad, que se denominan operadores de error. Se asume que el error que se produce al aplicar una compuerta cuántica afecta de modo directo exclusivamente a los qubits sobre los que actúa [14]. Por ejemplo, si se aplica una compuerta cuántica, de dos qubits, al primer y tercer qubit el operador de error será de la forma $E_{1,3} \otimes I_{2,4,\cdots,n}$. Esta notación se usa para indicar que el operador de error solo actúa sobre el primer y tercer qubit y sobre los demás actúa la identidad.

En este capítulo hablaremos sobre como modelamos estos errores en las compuertas del algoritmo de Shor. El modelo de perturbación esta basado en la *teoría de matrices aleatorias* por lo que a continuación discutiremos un poco de ello, pero antes discutiremos la similitud de nuestro sistema con el

de un sistema abierto.

Como se puede observar en la figura 3.1 en la parte cuántica del algoritmo de Shor se utilizan dos registros. Nosotros estamos interesados en el primero, que es donde se realizan las mediciones. Esto hace que el estado final del primer registro se pueda ver como un estado mixto debido a que el estado final no es un estado separable[8].

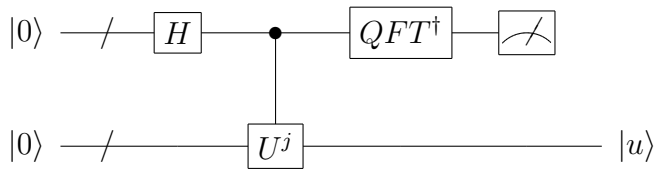

Figura 3.1: Esquema resumido de la parte cuántica del algoritmo de Shor.

Así el operador de densidad del primer registro se puede ver como

$$\rho' = \sum_k p(k)\rho_k, \tag{3.1}$$

donde $p(k)$ es la probabilidad de que el sistema se encuentre en el estado k, y ρ_k está determinada tanto por el enlazamiento con el registro 2 como por los efectos de las perturbaciones en la QFT inversa, y por lo tanto tiene aleatoriedad como en el caso de una interacción con el medio ambiente. Esto tiene mucha similitud con el tratamiento de un sistema abierto en decoherencia. Ademas de esto, en algunos casos decoherencia puede ser descrita por una evolución unitaria aleatoria sin necesidad de considerar el medio ambiente [5], es decir, de la misma forma en la que modelamos los errores debido a la imprecisión de los dispositivos en la aplicación de las compuertas.

3.1. Teoría de Matrices Aleatorias (RMT)

Originalmente la teoría de matrices aleatorias (RMT[9]) fue aplicada por Wigner para tratar con la estadística de eigenvalores y eigenfunciones de sistemas cuánticos complejos de muchos cuerpos. Ha sido aplicada exitosamente a las propiedades de fluctuaciones espectrales del núcleo atómico y se han estudiado las fluctuaciones estadísticas de los procesos de scattering en tales sistemas [15].

[8] No puede ser visto como un producto tensorial entre un estado puro del primer registro con otro del segundo.

[9] Random Matrix Theory.

El modelo introducido por Wigner difiere en una forma fundamental de la aplicación estándar de conceptos estadísticos en la física. En la mecánica estadística estándar, uno considera un ensamble de sistemas físicos idénticos, todos gobernados por el mismo hamiltoniano pero con condiciones iniciales distintas, y calcula funciones termodinámicas promediando sobre este ensamble. Wigner procedió diferente: Él consideró ensambles de sistemas dinámicos gobernados por hamiltonianos *diferentes* con alguna propiedad de simetría en común. El modelo enfoca su atención en las propiedades que son comunes en todos los miembros del ensamble y que están determinadas por las simetrías fundamentales sobresalientes.

RMT ha experimentado un desarrollo rápido e inesperado en el ultimo cuarto de siglo, ha sido aplicada exitosamente a una gran variedad de problemas físicos. RMT ha llegado a ser una herramienta importante en el estudio de sistemas que son aparentemente bastante diferentes de sistemas de muchos cuerpos. Por ejemplo: Sistemas cuánticos desordenados, sistemas cuánticos caóticos y gravedad en dos dimensiones [15].

3.1.1. Ensambles gaussianos

En RMT, se empieza con un ensamble de Hamiltonianos que es definido por un ensamble de matrices aleatorias, donde cada miembro describe una dinámica diferente. En la practica se usan matrices Hamiltonianas de dimensión finita D, aunque para resultados analíticos a menudo se toma el limite $D \to \infty$.

Dyson encontró tres ensambles relevantes en mecánica cuántica, definidos en términos de las propiedades de simetría del hamiltoniano [15].

(i) El ensamble ortogonal gaussiano (GOE[10]). Son sistemas invariantes ante inversión temporal con simetría rotacional. Para tales sistemas, la matriz hamiltoniana puede ser elegida real y simetrica.

(ii) El ensamble unitario gaussiano (GUE[11]). Son sistemas no invariantes ante reversión temporal. Para tales sistemas, las matrices hamiltonianas son hermitianas y los elementos de matriz aleatorios H_{nm} son complejos.

(iii) El ensamble simplectico gaussiano (GSE[12]). Son sistemas invariantes ante inversión temporal, sin simetría rotacional y con espín semi-entero.

En los tres casos la densidad de probabilidad de encontrar una matriz particular esta dada por una función $P_D(H)$. Las funciones $P_D(H)$ son invariantes bajo transformaciones ortogonales, unitarias, y simplecticas del hamiltoniano, respectivamente.

[10]Gaussian orthogonal ensemble
[11]Gaussian unitary ensemble
[12]Gaussian symplectic ensemble

Para estos ensambles considerados por Wigner, las densidades de probabilidad $P_D(H)$ se escogen con forma gaussiana. Que sean densidades de probabilidad gaussianas se debe a un argumento de máxima entropía en las densidades de probabilidad de los ensambles, ademas de fijar $\mathrm{tr}(H^2)$ a una constante [16].

3.1.2. Ensamble GUE

El ensamble GUE es el que usamos en las simulaciones del algoritmo de Shor perturbado por lo que lo describiremos con mayor profundidad.

La matriz hamiltoniana es hermitiana y sus elementos son variables aleatorias complejas con una distribución de probabilidad gaussiana. Salvo por los requisitos de simetría, los elementos de matriz son estadísticamente independientes y tienen media cero.

La función de peso para el ensamble GUE es de la forma

$$\begin{aligned} P_D(H) \propto & \exp\left(-\frac{2D}{\lambda^2}\mathrm{tr}H^2\right) = \exp\left(-\frac{2D}{\lambda^2}\left(\sum_i |H_{ii}|^2 + 2\sum_{i<j}|H_{ij}|^2\right)\right) \\ = & \exp\left(-\frac{2D}{\lambda^2}|H_{11}|^2\right)\cdots\exp\left(-\frac{2D}{\lambda^2}|H_{nn}|^2\right)\exp\left(-\frac{2D}{\lambda^2}2|H_{12}|^2\right) \\ & \cdots\exp\left(-\frac{2D}{\lambda^2}2|H_{n-1\ n}|^2\right), \end{aligned}$$

donde λ es una constante de normalización independiente de D. Como se puede en la función de peso, la distribución de los elementos diagonales de la matriz difiere de la de los elementos no diagonales. Con esto en mente nosotros construimos las matrices del ensamble GUE de la siguiente forma:

$$H = \frac{A + A^\dagger}{\sqrt{2}}, \tag{3.2}$$

donde A es una matriz con elementos aleatorios, independientes y con una distribución de probabilidad gaussiana estándar $\mathcal{N}(0,1)$. Debido a esto los elementos diagonales de H tiene una distribución gaussiana $\mathcal{N}(0,2)$ y los elementos no diagonales $\mathcal{N}(0,1)$.

3.2. Descripción del sistema

El objetivo de este trabajo es analizar como se comporta el algoritmo de Shor cuando perturbamos la parte cuántica del algoritmo. Con este fin,

construimos operadores unitarios cercanos a la identidad para modelar errores pequeños en las compuertas cuánticas de dos qubits en la QFT inversa. La intensidad de la perturbación se controla con el parámetro δ al que llamaremos *fuerza de perturbación*. Así, la compuerta perturbada esta dada por [8]:

$$U_j^\delta = U_j \exp(-i\delta V_j), \tag{3.3}$$

donde U_j representa una compuerta de dos qubits y V_j es un operador hermitiano perteneciente al ensamble GUE.

Las perturbaciones que estudiaremos se tratarán como errores internos y no se considerará que el sistema interactúe con el medio ambiente (decoherencia) puesto que es más difícil de modelar. Debe enfatizarse que las perturbaciones actúan únicamente sobre los qubits en los actúan las compuertas ideales U_j.

La razón por la que solo se perturban las compuertas de dos qubits es debido a que experimentalmente en las compuertas de un qubit ya se ha alcanzando un error menor a 10^{-4} en diversos sistemas y errores tan pequeños permite que se puedan implementar protocolos cuánticos de corrección [10]. Por otro lado, en compuertas de dos qubits no se ha podido alcanzar errores de esta magnitud. En la figura 3.2 se muestra el circuito para tres qubits de la $QFT^\dagger$ perturbada.

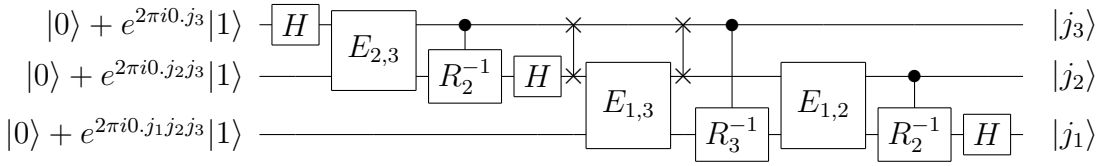

Figura 3.2: Circuito para tres qubits de la $QFT^\dagger$ perturbada, donde $E_{i,j} = \exp(-i\delta V_t)$. La operación entre H y E_2 es la operación Swap e intercambia los qubits involucrados, estas operaciones Swap no son realizadas numéricamente pero han sido puestas en el circuito para enfatizar que E_2 solo actúa sobre los qubits en los que actúa Ctrl-R_3^{-1}. Al igual que en los otros circuitos se omiten los factores $\frac{1}{\sqrt{2}}$ (a la izquierda).

A lo largo de las simulaciones se considerarán dos tipos de perturbaciones:

- Perturbación estática: El operador V_j no cambia a lo largo del circuito de la QFT inversa, es decir $V_j = V$, aunque sí cambia cuando se realiza el algoritmo nuevamente. Esta perturbación puede representar, por ejemplo, un error sistemático asociado al método usado en la implementación de las compuertas que no puede ser corregido.

- Perturbación dinámica: Se considera que el operador V_j cambia a lo largo del circuito de la QFT inversa, es decir $V_j \neq V_k$. Esta perturbación puede representar una imprecisión en los dispositivos que cambia con el tiempo.

3.3. Trabajos previos

En los trabajos [17, 18] se estudia la estabilidad de la QFT debido a los efectos de errores externos, es decir, la fidelidad. Estos errores actúan sobre todo el espacio de Hilbert y son modelados por una matriz aleatoria hermitiana de un ensamble GUE. La fidelidad se analiza bajo la teoría de respuesta lineal [8], la cual puede ser escrita de la forma

$$F(t) \approx 1 - \delta^2 \sum_{t_1,t_2=1}^{t} C(t_1, t_2), \tag{3.4}$$

donde la función de correlación de la perturbación es

$$C(t_1, t_2) = \big\langle V_{t_1}(t_1) V_{t_2}(t_2) \big\rangle - \big\langle V_{t_1}(t_1) \big\rangle \big\langle V_{t_2}(t_2) \big\rangle \tag{3.5}$$

siendo $V_j(t) = U^\dagger(t) V_j U(t)$ la perturbación de la j-esima compuerta en la representación de Heisemberg. En estos se muestra que es ventajoso usar la QFT mejorada [18] (IQFT), para la cual los errores crecen como $\sim n^2$, mientras que para la QFT normal los errores crecen como $\sim n^3$. Ademas, al igual que en [8], se hace referencia a la idea de que mientras más lento decae la fidelidad más rápido decaen las correlaciones.

En los trabajos [12,19] se estudia la probabilidad de éxito del algoritmo de Shor bajo perturbaciones en la exponenciacion modular, mientras la QFT se considera perfecta. La QFT no es perturbada debido a que en la exponenciacion modular se involucran muchas más operaciones y resulta más importante cuando hay una imperfecciones. Las perturbaciones son sobre todo el espacio y son debido a un acoplamiento residual entre los qubits.

Debido a que las perturbaciones que consideramos actúan solo sobre subespacios de dos qubits y a que las perturbaciones no son del mismo tipo que en [12, 19] sólo podriamos hacer comparaciones cualitativas.

Capítulo 4

Resultados

El método de estudio de nuestro sistema es el numérico. Con la ayuda del lenguaje de programación C++ se realizan simulaciones del sistema descrito en el capítulo anterior con cada tipo de perturbación, y obtenemos datos tanto de la probabilidad de éxito de obtener el período como de la fidelidad de la QFT inversa. En las simulaciones analizamos la factorización de los números 21, 39 y 55; desde 8 y hasta 15 qubits en la parte cuántica.

4.1. Distribución de probabilidad

Antes de pasar a los resultados obtenidos con la perturbación de nuestro sistema hay cosas que decir sobre el caso cuando no hay perturbaciones de ningún tipo. En la figura 4.1 se muestra un ejemplo de la distribución de probabilidad de las mediciones y que se habló en la parte cuántica del algoritmo de Shor. Esta gráfica fue obtenida mediante la simulación del sistema cuando la fuerza de la perturbación δ es cero, es decir para el caso ideal, y coincide a la perfección con la probabilidad de la ecuación 2.14. Los puntos rojos son las y que tras el procedimiento de fracciones continuas nos llevan al periodo correcto.

Las figuras 4.2 muestran las probabilidad de obtener el periodo r con el método explicado en la parte cuántica del algoritmo de Shor. Como se observa en las figuras ademas del periodo se obtienen otros resultados con cierta probabilidad, esto es debido a que no todos los números y nos sirven para encontrar r. Ademas se puede ver que mientras más grande es el número de qubits que se utilizan en el primer registro, la probabilidad de encontrar el período correcto aumenta. Esto coincide con lo establecido cualitativamente en la ecuación 2.11.

A partir de ahora solo nos interesará conocer la probabilidad de obtener el

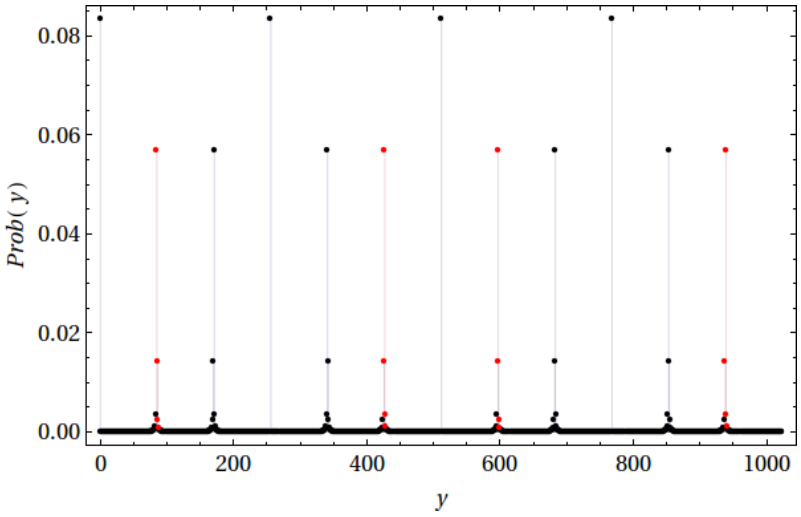

Figura 4.1: Distribución de probabilidad para $\delta = 0$, $N = 39 = 13 \times 3$, $m = 37$ y $n = 10$. Los puntos rojos indican las y que conducen a encontrar el periodo correcto.

periodo correcto y centraremos nuestra atención en como cambia ésta cuando analicemos el caso perturbado.

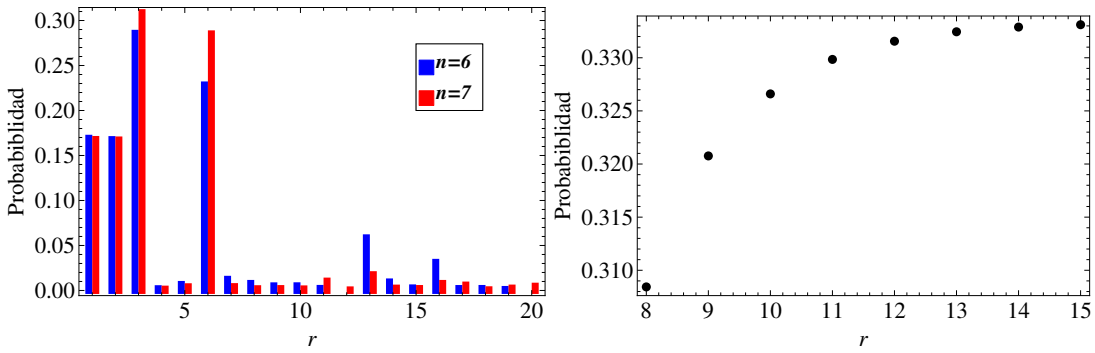

Figura 4.2: Probabilidad de obtener el periodo para $\delta = 0$, $N = 21 = 7 \times 3$ y $m = 2$. El periodo correcto es $r = 6$. A la izquierda se muestan los posibles r para $n = 6$ y $n = 7$, y a la derecha se muestra la probabilidad de r en función de n.

4.2. Fidelidad de la QFT inversa

La única etapa que es perturbada en la parte cuántica del algoritmo de Shor es la de la aplicación de la QFT inversa, por lo que analizaremos la estabilidad de ésta ante las perturbaciones. Estamos interesados en su

estabilidad ya que mientras más estable menos cambiará la probabilidad de obtener el periodo r al haber perturbaciones.

La medida de estabilidad que usamos es la fidelidad F promediando sobre un ensamble de estados iniciales aleatorios complejos $|\psi_k(0)\rangle$ con distribución de probabilidad gaussiana estándar $\mathcal{N}(0,1)$, usaremos la ecuación 2.16:

$$F = \frac{1}{l}\sum_{k=1}^{l} |\langle \psi_k^\delta | \psi_k \rangle| \tag{4.1}$$

donde $|\psi_k\rangle = QFT^\dagger |\psi_k(0)\rangle$, $|\psi_k^\delta\rangle = QFT_\delta^\dagger |\psi_k(0)\rangle$, $QFT_\delta^\dagger$ es la QFT inversa perturbada donde todas las compuertas de error tiene la misma fuerza de perturbación δ, y l es el tamaño del ensamble.

4.2.1. Fidelidad como función de δ

Como ya se dijo el método de estudio de nuestro sistema es el numérico. Obtenemos resultados de la fidelidad de la QFT inversa tanto para la perturbación estática como la dinámica, desde 8 hasta 15 qubits.

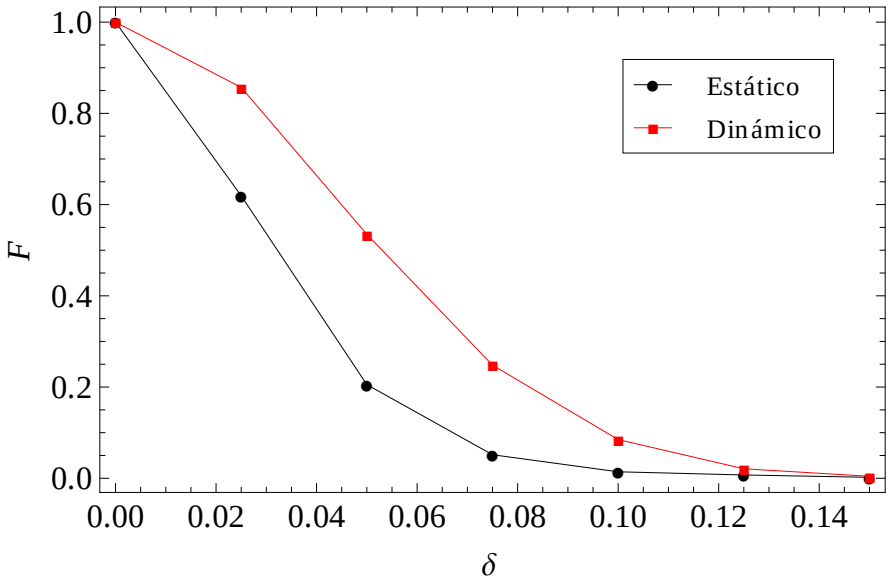

Figura 4.3: Comportamiento de la fidelidad con perturbación estática y dinámica en función de la fuerza de perturbación para $n = 12$ qubits y $l = 265$. Este comportamiento es el mismo para todo n.

En la figura 4.3 se puede observar que la fidelidad para la perturbación dinámica es más estable que la estática, es decir, la fidelidad decae más rápido

para la perturbación estática. Esto se esperaba ya que, aunque en nuestro sistema las perturbaciones actúan solo en el subespacio en el que actúan las compuertas, ha sido observado en otros trabajos donde las perturbaciones actúan sobre todo el espacio [20, 21].

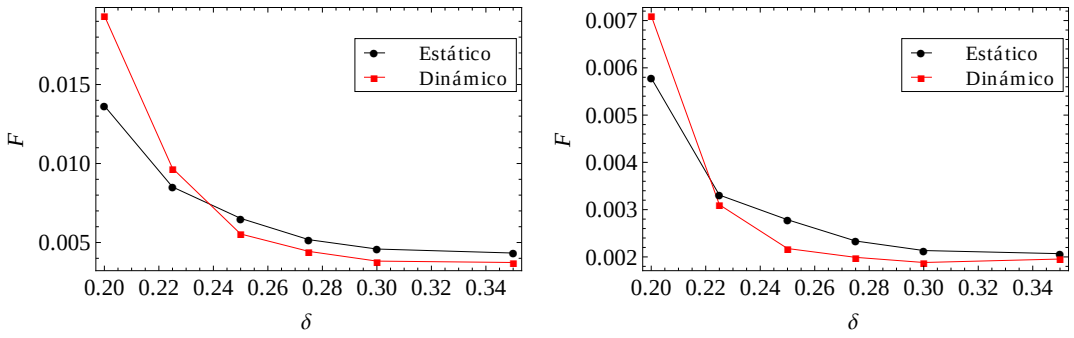

Figura 4.4: Comportamiento de la fidelidad con perturbación estática y dinámica en función de la fuerza de perturbación para $n = 8$, $l = 1800$ (izquierda) y $n = 9$, $l = 900$ (derecha). Se puede observar que para cierta fuerza de perturbación las gráficas se intersectan.

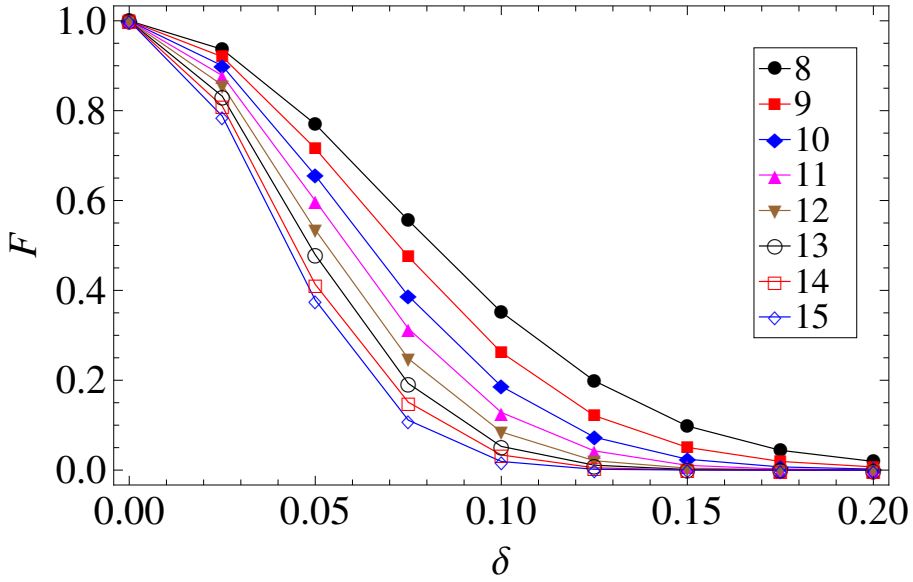

Figura 4.5: Comportamiento de la fidelidad con perturbación dinámica. Se observa que mientras más grande es el número de qubits la fidelidad decae más rápido.

Algo que no se alcanza a apreciar en la figura 4.3 es que para una fuerza de perturbación δ lo suficientemente grande el comportamiento entre las gráficas

se intercambia. En cierto punto las fidelidad decae más rápido en el caso de la perturbación dinámica como se observa en la figura 4.4.

En la figura 4.5 se muestra que la fidelidad decae más rápido mientras mayor es el número de qubits, esto ocurre para ambos tipos de perturbación. Este comportamiento de cierta forma es lógico pues mientras mayor es número de qubits mayor es el número de operaciones que involucran compuertas perturbadas.

4.2.2. Ajuste para la fidelidad

La expresión para la fidelidad en la aproximación de respuesta lineal (3.4) a su vez se puede aproximar [8,9,17,18] como $F = \mathrm{Exp}(-\delta^2 f(n))$ donde $f(n)$ es un polinomio que depende de n.

Encontramos que esta expresión para la fidelidad ajusta bien para las perturbaciones dinámicas pero no para las estáticas. Llegamos a esta conclusión puesto que lo primero que hacemos es calcular el coeficiente para δ que se ajusta mejor a los datos numéricos. Esto se hace para cada número de qubits y para la perturbación estática el coeficiente es diferente en cada caso, mientras que para el dinámico varia muy poco de 2. Así, para la perturbación dinámica se tiene que dependencia de la fidelidad en el número de qubits n y en la fuerza de perturbación δ es:

$$F_{din}(n,\delta) = \mathrm{Exp}(-\delta^2\ (2.482\ n^2 - 15.27\ n + 67.488)). \tag{4.2}$$

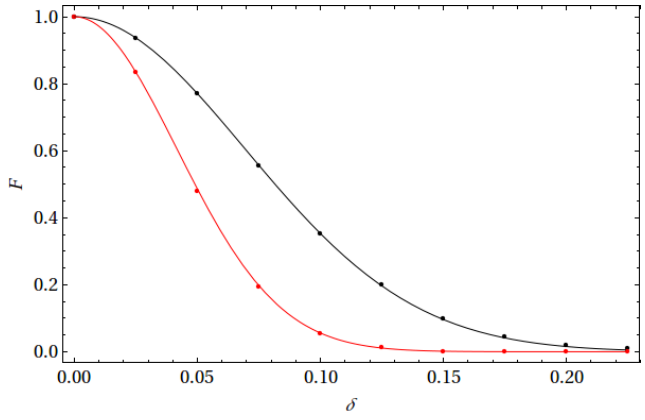

Figura 4.6: Ajuste para la fidelidad con perturbación dinámica. Los puntos corresponden a valores obtenidos mediante la simulación del sistema y la curva es el ajuste $F_{est}(n,\delta)$, la gráfica negra es para $n = 8$, $l = 1800$ y roja para $n = 13$, $l = 160$.

Obtuvimos una dependencia en n^2 mientras que en [18] se obtiene una dependencia con n^3 para el decaimiento de la fidelidad de la QFT. La función 4.2 se ajusta muy bien salvo para una fuerza de perturbación muy grande. Cuando la fuerza de perturbación δ es grande la fidelidad de ambos tipos de ruido convergen a una constante que depende del número de qubits [9]:

$$\lim_{\delta\to\infty} F(n,\delta) = \frac{1}{2^n}. \tag{4.3}$$

En la figura 4.7 se observa que la fidelidad para ambos tipos de perturbaciones se comporta de acuerdo a la ecuación anterior.

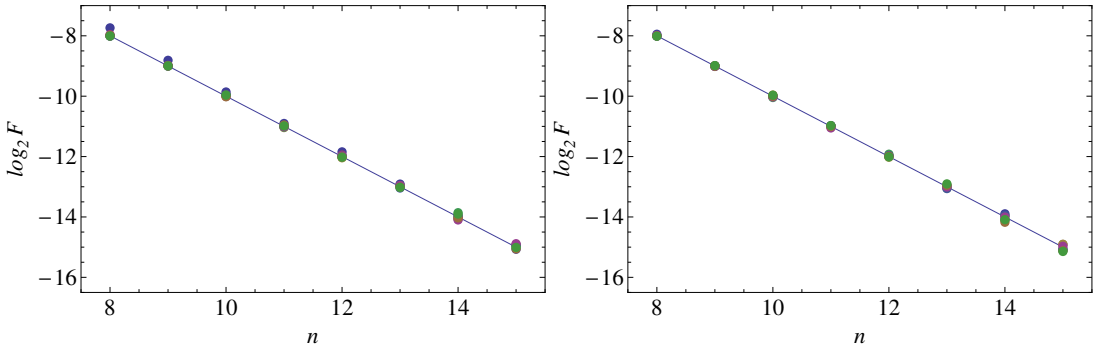

Figura 4.7: Comportamiento de la fidelidad cuando la fuerza de la perturbación es grande. Tanto en el caso de la perturbación estática (izquierda) como de la dinámica (derecha) la fidelidad se comporta como la ecuación 4.3. Los puntos corresponden a diferentes valores de $\delta = \{,3,,4,\ldots,1\}$ y la linea solida corresponde a la ecuación 4.3.

4.2.3. Distribución del ensamble $|\langle\psi_k^\delta|\psi_k\rangle|$

Como ya se dijo antes la fidelidad la calculamos promediando sobre un ensamble de estados iniciales aleatorios $|\psi_k(0)\rangle$. Usamos la ecuación 4.1:

$$F = \frac{1}{l}\sum_{k=1}^{l} |\langle\psi_k^\delta|\psi_k\rangle|$$

donde $|\psi_k\rangle = QFT^\dagger|\psi_k(0)\rangle$, $|\psi_k^\delta\rangle = QFT_\delta^\dagger|\psi_k(0)\rangle$, $QFT_\delta^\dagger$ es la QFT inversa perturbada donde todas las compuertas de error tiene la misma fuerza de perturbación δ, y l es el tamaño del ensamble.

A continuación discutiremos como es la densidad de probabilidad del valor $|\langle\psi_k^\delta|\psi_k\rangle|$ al calcular la fidelidad con solo un estado aleatorio. Las distribuciones que se muestran en la figura 4.8 se obtuvieron haciendo un histograma

sobre los datos que se utilizan para obtener la fidelidad y las lineas punteadas representan la ubicación del promedio de la distribución. Se muestran cuatro casos que nos ayudaran a inferir el comportamiento general de esta distribución:

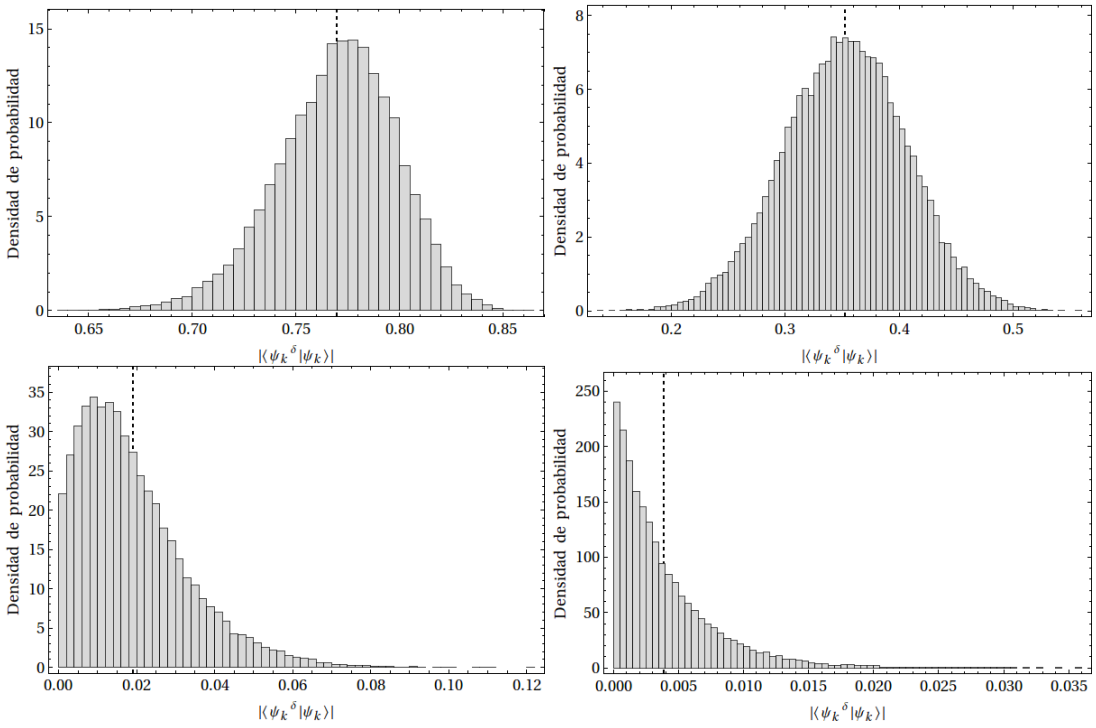

Figura 4.8: Distribución de probabilidad de aparición de los datos $|\langle\psi_k^{\delta}|\psi_k\rangle|$. Hay cuatro casos que se pueden diferenciar conforme la fuerza de la perturbación δ aumenta: F grande (arriba izquierda), F intermedia (arriba derecha), F pequeña (abajo izquierda) y $\lim_{\delta\to\infty} F$ (abajo derecha). Las lineas punteadas marcan el valor promedio de la distribución.

- La distribución de arriba y a la izquierda es una distribución obtenida cuando la fidelidad es grande. Mientras más grande sea F la distribución estará más ubicada hacia la derecha.
- La distribución de arriba y a la derecha es una distribución obtenida cuando la fidelidad es intermedia. Se puede observar que esta distribución es una distribución gaussiana.
- La distribución de abajo y a la izquierda es una distribución obtenida cuando la fidelidad es pequeña. Se puede observar que es como si la distribución se hubiera aplastado del lado izquierdo de la distribución y por esta razón el promedio no coincide con los valores más probables.

- La distribución de abajo y a la derecha es una distribución obtenida en el caso $\lim_{\delta\to\infty} F$. Esta distribución es la distribución que se observa en los datos de la figura 4.7. El valor promedio de la fidelidad es el de la ecuación 4.3 y al igual que el caso anterior este valor no esta cerca de los valores más probables.

Resumiendo, cuando la fidelidad es grande la distribución está ubicada muy hacia la derecha y mientras disminuye la distribución se mueve hacia la izquierda.

El comportamiento de la distribución es el mismo para ambas perturbaciones. Sin embargo, para la perturbación estática el movimiento de la distribución, al principio, ocurre más rápido. Otras cosas que se observaron al hacer el análisis de la fidelidad son:

- Mientras mayor es el número de qubits menos datos $|\langle\psi_k^\delta|\psi_k\rangle|$ se necesitan para obtener un buen promedio, a este fenómeno se le llama auto-promedio. Esto se debe a que mientras mayor es el número de qubits, mayor es el número de operaciones en las que se involucran compuertas perturbadas y esto hace que se auto-promedie a lo largo del proceso.

- Para obtener un buen promedio se necesitaron menos datos en la perturbación estática que en la dinámica para el mismo número de qubits.

4.3. Probabilidad de éxito

A continuación estudiaremos el efecto de perturbar la QFT inversa en la parte cuántica del algoritmo de Shor sobre la probabilidad de éxito de encontrar el periodo y sobre la probabilidad de éxito de la factorización considerando sólo los números coprimos m con N. Como veremos estas probabilidades no son las mismas.

4.3.1. Probabilidad de éxito de encontrar el período

La probabilidad de éxito de encontrar el periodo está dada de la siguiente forma:

$$R = \frac{1}{\phi(N)-1} \sum_{m\in\mathbf{C}\setminus\{1\}} R_m \tag{4.4}$$

donde $\phi(x) = |\mathbf{C}|$ es la función ϕ de Euler y nos dice cuántos números $m < N$ son coprimos con N, $\mathbf{C} = \{y \in \mathbb{N} | y < N \wedge \gcd(N, y) = 1\}$; y

$$R_m = \frac{1}{z} \sum_{i=1}^{z} R_m^{(i)}$$

es la probabilidad promedio de obtener el periodo correcto para un numero coprimo m donde z el tamaño del ensamble, $R_m^{(i)}$ es la probabilidad de encontrar el periodo en una realización del algoritmo para un coprimo m, un número de qubits n y número a factorizar N dados.

En la figura 4.9 se muestra el comportamiento de la probabilidad R como función de la fuerza de perturbación δ para ambos tipos de perturbaciones. Se observa que al principio, al igual que en la fidelidad, la probabilidad en el caso de la perturbación estática decae más rápido que para la perturbación dinámica y en algún punto se cruzan. El cruce de las gráficas es mucho más marcado que para la fidelidad y mientras mayor es el número de qubits el cruce ocurre antes. Este comportamiento es el mismo para los tres números analizados 21, 39 y 55.

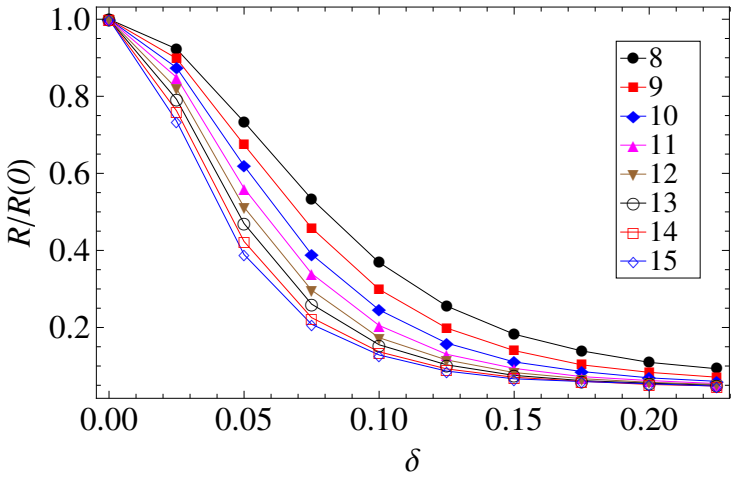

Figura 4.9: Comportamiento de la probabilidad R con perturbación estática para $N = 55$. Cada gráfica está normalizada a la probabilidad cuando $\delta = 0$. Se observa que mientras más grande es el número de qubits R decae más rápido.

En la figura 4.10 se puede observar que la probabilidad R decae más rápido mientras mayor es el número de qubits. En la figura cada gráfica se normaliza a la probabilidad cuando $\delta = 0$ precisamente con el objetivo de ver cuando decae con mayor rapidez. Esto sucede tanto para la perturbación

estática como para la dinámica y para los tres números estudiados. Esta gráfica difiere de las mostradas en la figura 4.11 donde mostramos como es la probabilidad R sin normalizar las gráficas a la probabilidad cuando $\delta = 0$. La ecuación 2.11,

$$n = k + \left[\log\left(2 + \frac{1}{2\epsilon}\right)\right],$$

sugiere que en el caso ideal ($\delta = 0$) mientras más qubits la probabilidad de éxito es mayor pero como se puede observar en la figura 4.11 esto solo es cierto para δ pequeña. Cuando δ es grande hay un número de qubits óptimo para el cual la probabilidad de éxito es mayor.

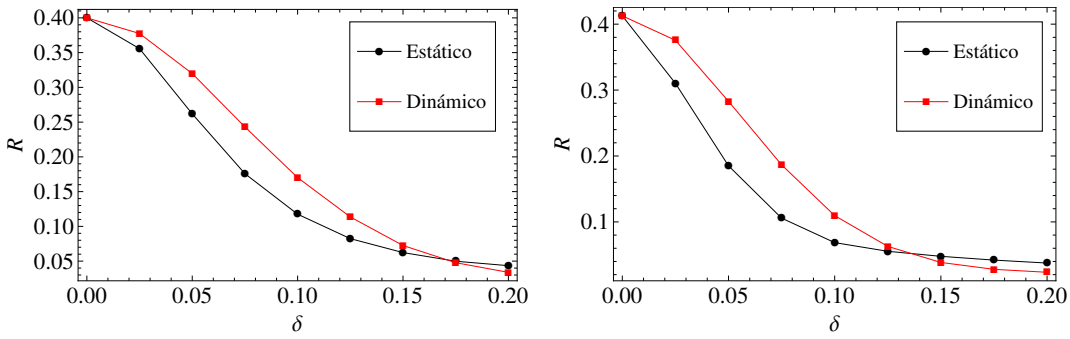

Figura 4.10: Comportamiento entre la probabilidad R para la perturbación estática y dinámica como función de la fuerza de perturbación. Gráficas para el número $N = 39$ para $n = 10$ promediada sobre $z = 75$ (izquierda) y $n = 15$ promediada sobre $z = 10$ (derecha).

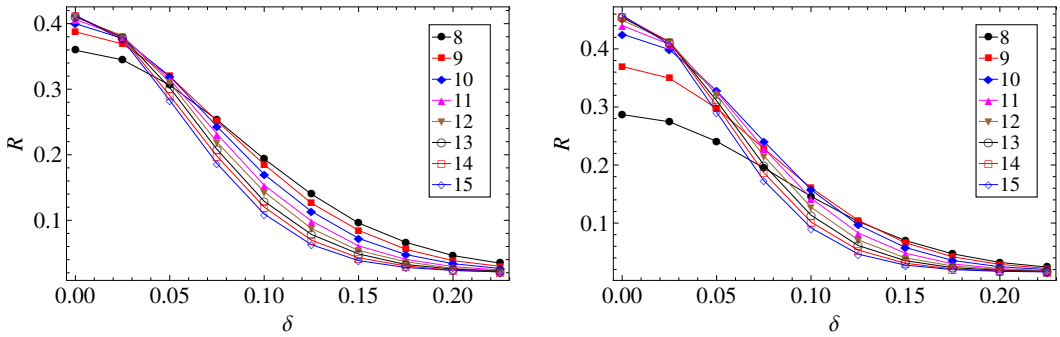

Figura 4.11: Comportamiento de la probabilidad R con perturbación dinámica para $N = 39$ (izquierda) y $N = 55$ (derecha).

4.3.2. Probabilidad de éxito de la factorización

La probabilidad de éxito de la factorización difiere de la probabilidad de la sección anterior pues en este caso no consideramos a todos los números m coprimos, sino sólo aquellos que cumplen con las condiciones del *algoritmo de Shor (2.3.1)* para conseguir la factorización. Así, la probabilidad de éxito de la factorización esta dada por:

$$P = \frac{1}{\phi(N) - 1} \sum_{m \in \zeta} R_m, \tag{4.5}$$

donde ϕ es la función ϕ de Euler y ζ es el conjunto $\zeta = \{m \in \mathbf{C}\backslash\{1\} | r$ cumple con las condiciones del *algoritmo de Shor* (2.3.1)$\}$.

En esta definición de P no estamos considerando a los números que no son coprimos con N porque solo estamos interesados en el caso donde interviene la etapa cuántica. Pero, si se quiere conocer este número se debe tomar en cuenta la elección aleatoria del paso 1 del algoritmo de Shor:

$$P' = \frac{N - \phi(N) - 1}{N - 2} + \frac{1}{N - 2} \sum_{m \in \zeta} R_m. \tag{4.6}$$

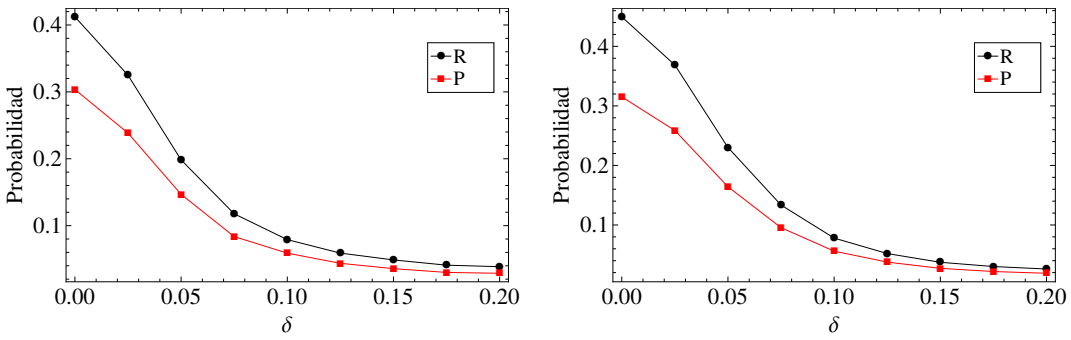

Figura 4.12: Comparación entre las probabilidades R y P para $N = 39$, $n = 14$ (izquierda) y $N = 55$, $n = 12$ (derecha).

La probabilidad P tiene las mismas características que la probabilidad R pero es más interesante pues estamos viendo la probabilidad de éxito de la factorización; P siempre es menor que R pues el conjunto ζ tiene menos elementos que el conjunto $\mathbf{C}\backslash\{1\}$. En la figura 4.12 se muestran dos ejemplos.

4.3.3. Distribución del ensamble $R_m^{(i)}$

Cuando realizamos el experimento de factorización tomamos un número coprimo m con N y calculamos su periodo r. En una realización del experimento obtenemos este periodo con una probabilidad $R_m^{(i)}$. En seguida estudiaremos como se comporta este ensamble.

La forma de esta distribución es como se muestra en la gráfica de arriba y a la izquierda en la figura 4.13. Aparecen varios picos gaussianos donde cada una refleja la distribución para un grupo de coprimos m que tiene la misma probabilidad de éxito. En general se tiene que cuando la fuerza de la perturbación aumenta los picos se mueven hacia la izquierda, pues la probabilidad de éxito disminuye, y se juntan más.

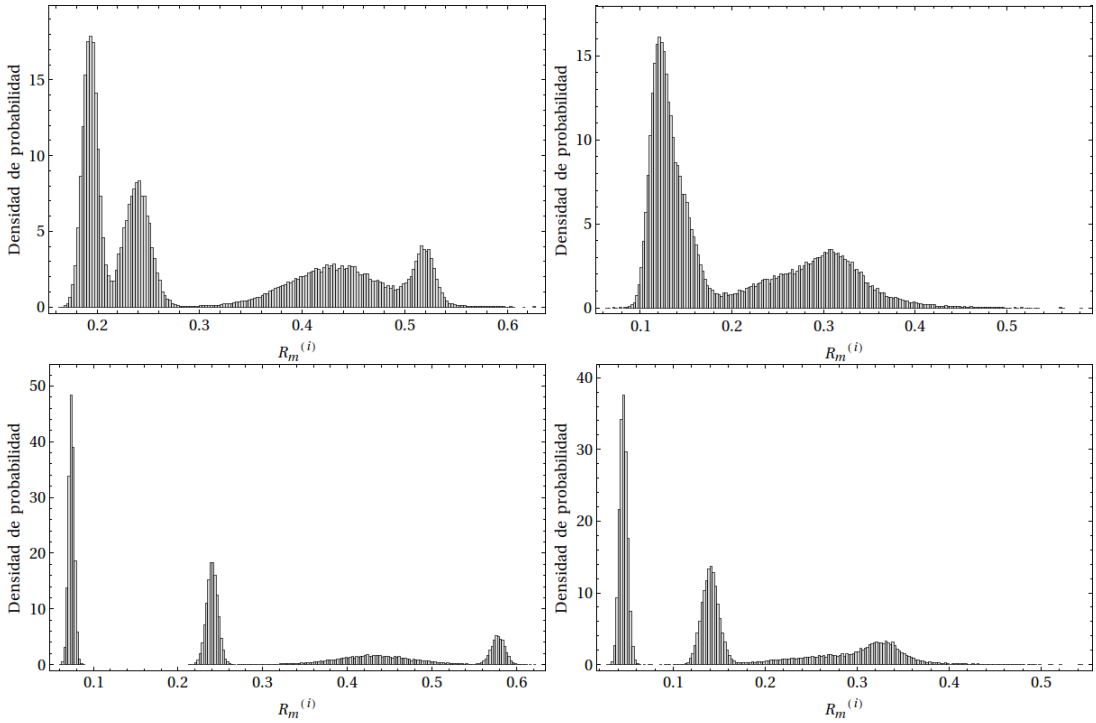

Figura 4.13: Distribución de probabilidad de los datos $R_m^{(i)}$. Gráficas para $n = 8$, $N = 39$ (arriba), $N = 55$ (abajo), $\delta = 0{,}05$ (izquierda) y $\delta = 0{,}1$ (derecha). Los picos en las distribuciones corresponden a las distribuciones de los coprimos que tienen las mismas probabilidades de éxito de encontrar el periodo.

Cuando la fuerza de la perturbación es cero la distribución solo corresponde a la probabilidad asociada con cada coprimo. En la figura 4.14 se muestran dos ejemplos.

Las gráficas de la figura 4.13 son considerando a todos los coprimos m con N. Pero, si consideramos solo a los coprimos con los que se consigue la

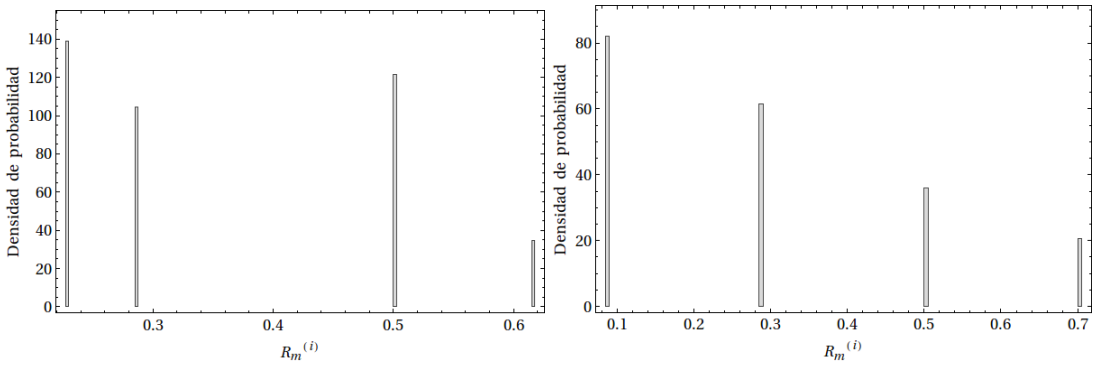

Figura 4.14: Distribución cuando $\delta = 0$ en $N = 39$ (izquierda) y $N = 55$ (derecha). Los picos tienen ancho cero pues en realidad son proporcionales a la función delta .

factorización ocurre algo curioso. En las gráficas de la figura 4.15 se observa que cuando solo tomamos en cuenta los coprimos con los que se logra una factorización exitosa el pico que tiene la probabilidad mayor desaparece. Esto pasa en todos los casos que se estudiaron.

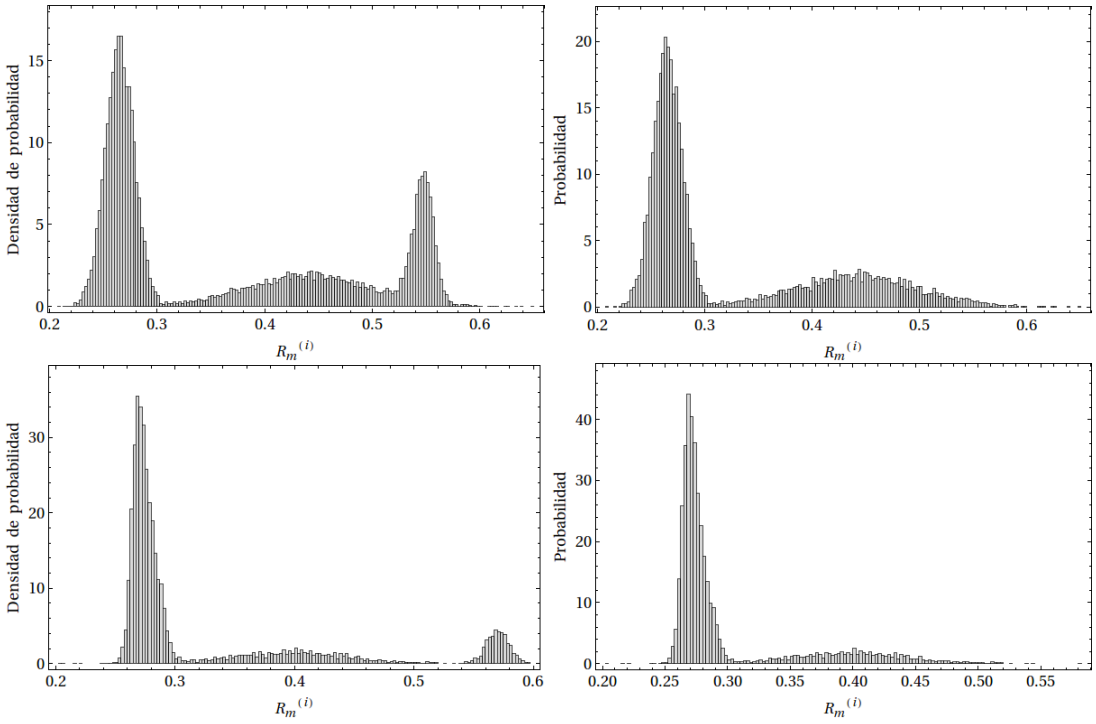

Figura 4.15: Distribución con todos los coprimos (izquierda) y con los que se logra una factorización exitosa (derecha) para $\delta = 0{,}05$, $n = 8$, $N = 21$ (arriba) y $N = 55$, $n = 11$ (abajo).

4.4. Comparación entre fidelidad y probabilidad de éxito

A continuación vemos como se comporta la probabilidad de éxito de la factorización contra la fidelidad. En la gráfica 4.16 se observa que la probabilidad P dinámica decae más rápido que la estática. Esto nos dice que si se logra implementar la QFT inversa con la misma fidelidad para ambos tipos de ruido la probabilidad será más alta para la perturbación estática. Esto ocurre para los tres números estudiados N y para todos los qubits n. Este comportamiento no puede ser inferido fácilmente de la gráficas anteriores.

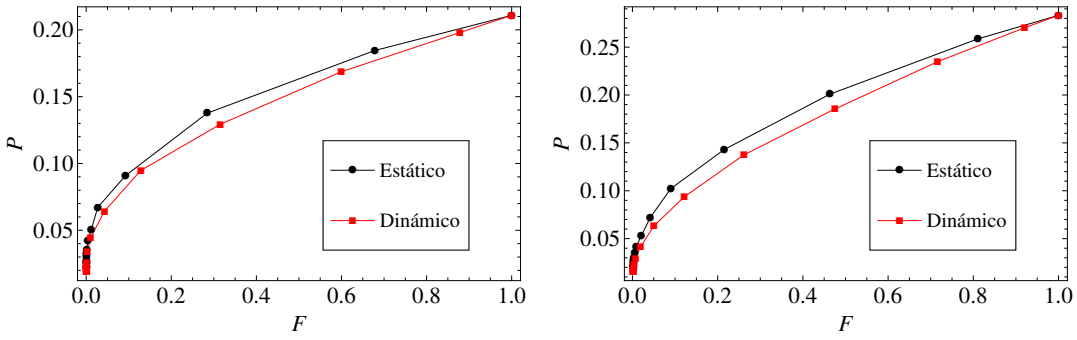

Figura 4.16: Comparación entre las probabilidades P para la perturbación estática y dinámica como función de la fidelidad F para $N = 21$, $n = 11$ promediada sobre $t = 60$ (izquierda) y $N = 39$, $n = 9$ promediada con $t = 105$ (derecha).

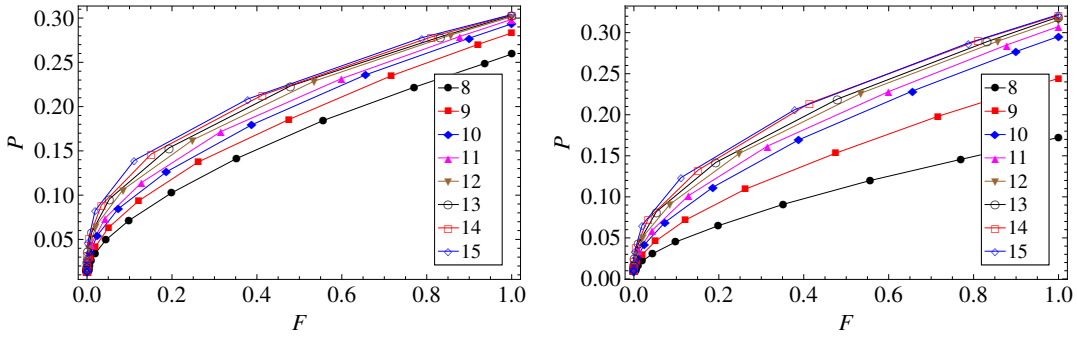

Figura 4.17: Comportamiento de la probabilidad P para la perturbación dinámica en el número de qubits como función de la fidelidad. Gráficas para $N = 39$ (izquierda) y $N = 55$ (derecha).

Otra cosa que se observa es que tanto para la probabilidad P con perturbación estática como con perturbación dinámica en los tres números analizados se tiene que la probabilidad sí es mayor mientras mayor es el número de

qubits cuando fijamos la fidelidad. En la figura 4.17 se muestran dos ejemplos para el caso de probabilidad P con perturbación dinámica pero tambien ocurre para la perturbación estática.

Parece ser que lo establecido en la ecuación 2.11 es cierto cuando fijamos la fidelidad. Sin embargo, no es posible conseguir esto experimentalmente pues si con un método se consigue que la QFT inversa para n qubits tiene una fidelidad dada, entonces, la fidelidad para la QFT inversa con $n + n'$ qubits tendrá una fidelidad menor pues hay más operaciones implicadas.

4.5. Número de qubits óptimo

Como ya se había dicho en la sección 4.3.1 cuando tenemos una fuerza de perturbación δ grande existe un número de qubits óptimo para el cual la probabilidad de éxito en la factorización es la máxima. A este número de qubits le llamaremos Q. En las figuras 4.18, 4.19 y 4.20 se muestra el número de qubits óptimo para $N = 21, 39$ y 55, respectivamente.

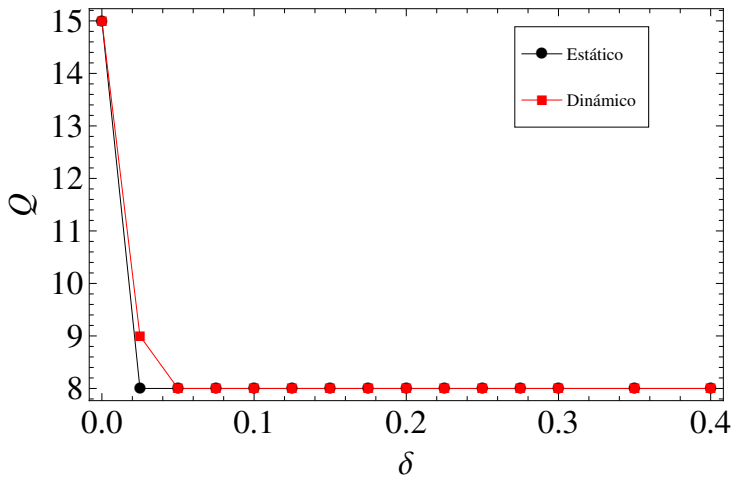

Figura 4.18: Número de qubits óptimo para $N = 21$.

Tanto para $N = 39$ como para $N = 55$ se observa un comportamiento parecido al de la fidelidad como para las probabilidad de éxito. Cuando δ es pequeña Q decae más rápido para la perturbación dinámica y después, para cierto valor de δ, Q es mayor para la perturbación estática.

Que la probabilidad de éxito de la factorización sea la máxima para un número de qubits que no es el mayor es un resultado importante pues nos hace ver que si la perturbación es grande no importa que nuestra computadora

cuántica tenga muchos qubits, la tarea no se realizará lo más eficientemente posible. En algún punto el menor número de qubits estudiado $n = 8$ resulta más eficiente que el mayor $n = 15$. Las gráficas que se muestran se realizaron sólo tomando en cuenta a los qubits estudiados, es decir, desde $n = 8$ hasta $n = 15$. Para $\delta = 0$ el valor de Q será el del n máximo al que se tiene acceso.

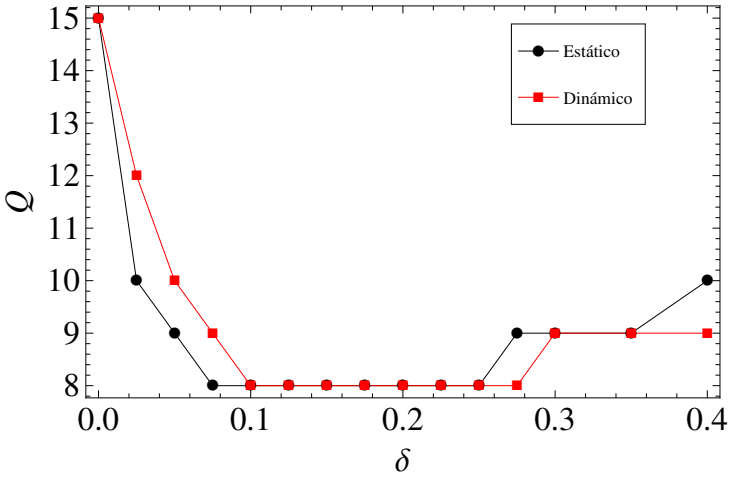

Figura 4.19: Número de qubits óptimo para $N = 39$.

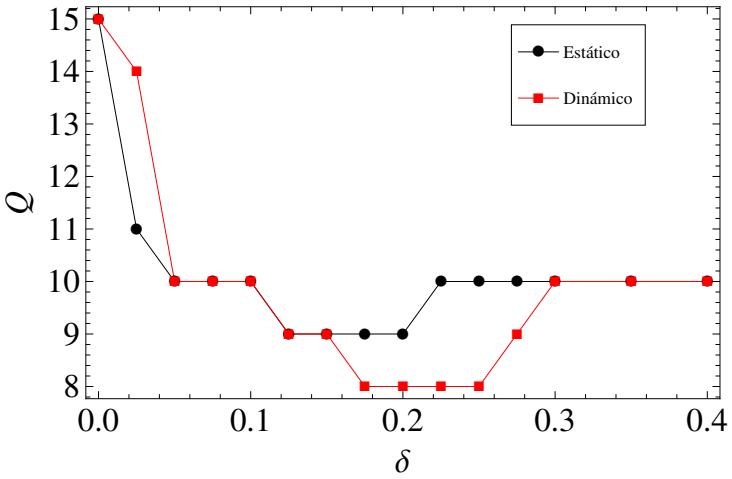

Figura 4.20: Número de qubits óptimo para $N = 21$.

Capítulo 5

Conclusiones

A lo largo del trabajo se estudiaron las implicaciones de perturbar las compuertas de dos qubits en la parte cuántica del algoritmo de factorización de Shor sobre la probabilidad de éxito del algoritmo, así como en la fidelidad de la QFT inversa.

Se determinó que la fidelidad como función de la fuerza de perturbación δ decae más rápido en el caso de perturbaciones estáticas que en el de perturbaciones dinámicas. Pero, cuando la fidelidad es pequeña el comportamiento se invierte y decae más rápido para la perturbación dinámica. Se estableció que la fidelidad decae más rápido mientras mayor es el número de qubits con los que se trabajan.

El comportamiento de la probabilidad de éxito en la factorización como función de la fuerza de perturbación es similar que el de la fidelidad de la QFT inversa en cuanto al decaimiento con perturbación estática o dinámica, ademas decae más rápido mientras mayor es el número de qubits. Un resultado importante es que existe un número de qubits para el cual la probabilidad de éxito es la máxima y no es necesariamente el número de qubits más grande al que se tiene acceso. Así, en el caso perturbado, existe un número de qubits óptimo para el cual la probabilidad de éxito es la máxima dependiendo de la fuerza de perturbación.

Otro resultado sobresaliente es que la probabilidad de éxito en la factorización, como función de la fidelidad, decae más rápido para la perturbación dinámica. Ademas, al igual que el caso ideal, si fijamos la fidelidad mientras mayor es el número de qubits mayor es la probabilidad de éxito del algoritmo.

Como trabajo futuro hay varias opciones. Para extender el trabajo se puede: incrementar el número de qubits máximo con ayuda de el lenguaje de programación CUDA; estudiar más números a factorizar N para tratar de establecer una relación entre estos; implementar la QFT mejorada (IQFT) y comparar su estabilidad con la de la QFT ante este tipo de errores; y realizar

el algoritmo de exponenciacion modular y perturbarlo.

Una forma en la que se podría mejorar la eficiencia del algoritmo de factorización es hacer una implementación semiclásica de la QFT en la que solo se usa un qubit [22], esto hace que podamos acceder a la factorización de números más grandes con el mismo número de qubits y así estudiar números N mayores.

Apéndice A

Complemento al capítulo 2

En este apéndice se presentan comentarios e información que complementan a la del capítulo 2.

A.1. Circuito de la QFT

A continuación se presenta de desarrollo algebraico para obtener la relación mostrada en la ecuación 2.5. Utilizando la equivalencia en la representación producto [3] $|j_1, ..., j_n\rangle = |j\rangle$ y las ecuaciones 2.1, 2.3

$$|j\rangle \rightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle$$

Separamos en sumas de los estados de los qubits:

$$= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j \left(\sum_{l=1}^{n} k_l 2^{-l}\right)} |k_1 ... k_n\rangle$$

Separamos en un producto tensorial de estados de los qubits:

$$\begin{aligned}
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right] \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right]
\end{aligned}$$

$$= \frac{\big(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\big)\big(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\big)\cdots\big(|0\rangle + e^{2\pi i 0.j_1\cdots j_n}|1\rangle\big)}{2^{n/2}}.$$

A continuación se analiza como trabaja el circuito de la QFT mostrado en la figura 2.1. Consideremos que la QFT actúa sobre el estado inicial $|j_1, ..., j_n\rangle$. Aplicar la compuerta Hadamard sobre el primer qubit produce

$$\frac{1}{2^{1/2}}\big(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\big)|j_2, ..., j_n\rangle,$$

ya que $e^{2\pi i 0.j_1} = -1$ cuando $j_1 = 1$ y $+1$ cuando $j_1 = 0$. Aplicar la compuerta Ctrl-R_2 produce el estado

$$\frac{1}{2^{1/2}}\big(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\big)|j_2, ..., j_n\rangle.$$

Se continúan aplicando las compuertas Ctrl-R_3, R_4 hasta R_n, y cada una de estas agrega un bit en la fase de $|1\rangle$. Al final de este procedimiento se tiene el estado

$$\frac{1}{2^{1/2}}\big(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\big)|j_2, ..., j_n\rangle.$$

Ahora se realiza un procedimiento similar en el segundo qubit. Se aplica la compuerta Hadamard sobre el segundo qubit y produce el estado

$$\frac{1}{2^{2/2}}\big(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\big)\big(|0\rangle + e^{2\pi i 0.j_2}|1\rangle\big)|j_3, ..., j_n\rangle,$$

y las compuertas Ctrl-R_2 hasta R_{n-1} producen el estado

$$\frac{1}{2^{2/2}}\big(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\big)\big(|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle\big)|j_3, ..., j_n\rangle.$$

Se realiza este procedimiento en cada qubit y se obtiene el estado

$$\frac{1}{2^{n/2}}\big(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\big)\big(|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle\big)...\big(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\big).$$

Para finalizar se realiza la aplicación de las compuertas Swap, que se omitieron en la figura 2.1 por motivos de claridad y puesto que esta operación se puede realizar clásicamente con facilidad, se obtiene el estado final

$$\frac{1}{2^{n/2}}\big(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\big)\big(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\big)\cdots\big(|0\rangle + e^{2\pi i 0.j_1\cdots j_n}|1\rangle\big),$$

el cual es el estado final que se muestra en la ecuación 2.5.

A.2. Criptografía RSA

Aquí se presenta un ejemplo hecho en *Mathematica* de como el método RSA codifica y decodifica mensajes mediante el producto de dos números primos.

Bob le quiere enviar a Alice un mensaje mediante un vía publica. Para hacer esto Bob codifica el mensaje y Alice será la única persona capaz de decodificarlo.

1. Alice prepara dos números primos grandes p y q, los cuales mantiene en secreto y solo publica el producto de ambos N. En este ejemplo se usarán:

   $$p = 61427839512211$$

   y

   $$q = 61427839542533,$$

   para los cuales

   $$N = 377337946900056589610737046 3.$$

   Toma un tiempo bastante largo (para número más grandes) factorizar N como un producto de p y q. Alice también prepara un número llamado *exponente* $e(< N)$, el cual es primo relativo a $(p-1)(q-1)$. Ella puede encontrar fácilmente el número:

   $$e = 901,\ \gcd[e,(p-1)(q-1)] = 1,$$

   por ejemplo, donde $\gcd[a,b]$ es el máximo común divisor entre a y b. Este número también es publicado junto con N. Ahora calcula en inverso modular d de e mod $(p-1)(q-1)$:

   $$de \equiv 1 \bmod (p-1)(q-1) \rightarrow$$
   $$d = 728710352503970132113792381.$$

   Alice mantiene a d en secreto.

2. Bob le quiere enviar el mensaje "hola", por ejemplo. Con cierto método (ASCII, etc.) se transforma este mensaje a una secuencia de números decimales y el resultado es menor que N. Supón que el método transforma el mensaje a

   $$\text{adiós} \rightarrow 234005670089100,$$

por ejemplo. Él codifica su mensaje como adiós^e mod$N$ y lo envía a Alice a través de un canal abierto:

$$\begin{aligned}\text{mensaje} &= \text{adiós}^e \bmod N \\ &= 447305466323772068994836794.\end{aligned}$$

3. Usando d Alice decodifica el mensaje que recibió

$$\text{mensaje}^d \bmod N = 234005670089100.$$

Así Alice obtiene el mensaje.

A.3. Factorización clásica

El siguiente procedimiento es el realizado *Mathematica* para el algoritmo de factorización clásica que se usa en el algoritmo de Shor. Está hecho para para factorizar un número N como un producto de dos números.

```
FactorizacionClasica[N_] := Module[{m, r, p, q},
 Label[uno];
 m = Random[Integer, {2, N - 1}];
 If[GCD[m, N] === 1, Goto[dos], Goto[uno]];
 Label[dos];
 r = Timing[Catch[Table[If[PowerMod[m, i, N] === 1, Throw[i]],
                 {i, 2, N/2}]]];
 If[EvenQ[r[[2]]], Goto[tres], Goto[uno]];
 Label[tres];
 If[Mod[m^{r[[2]]/2} + 1, N] === 0, Goto[uno]];
 p = GCD[m^{r[[2]]/2} - 1, N];
 q = N/p;
 {m, r, p, q}]
```

El procedimiento solo considera números m que son coprimos con N. El resultado es una lista con m, el periodo y su tiempo de computo $r = \{tiempo, periodo\}$, los factores p y q.

A.4. Algoritmo de fracciones continuas

El objetivo del algoritmo de fracciones continuas es expresar números reales solo en términos de enteros, usando expresiones de la forma [3]

$$[a_0, ..., a_M] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ldots + \frac{1}{a_M}}}}, \tag{A.1}$$

donde $a_0, ..., a_M$ son enteros positivos, para los propósitos del algoritmo de Shor es conveniente tomar $a_0 = 0$ pues φ es menor que 1. Definimos el *m-convergente* $(0 \leq m \leq M)$ como $[a_0, ..., a_m]$. El *algoritmo de fracciones continuas* es un método para determinar la expansión en fracciones continuas de un numero real arbitrario.

Bibliografía

[1] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[2] B. Schumacher, "Quantum coding," *Phys. Rev. A*, vol. 51, pp. 2738–2747, Apr 1995.

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge University Press, 2000.

[4] M. Schlosshauer, "Decoherence, the measurement problem, and interpretations of quantum mechanics," *Rev. Mod. Phys.*, vol. 76, pp. 1267–1305, Feb 2005.

[5] J. Helm and W. T. Strunz, "Quantum decoherence of two qubits," *Phys. Rev. A*, vol. 80, p. 042108, Oct 2009.

[6] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Phys. Rev. A*, vol. 54, pp. 147–153, Jul 1996.

[7] M. Nakahara and T. Ohmi, *Quantum computing: from linear algebra to physical realizations*. CRC Press, 2008.

[8] T. Prosen and M. Znidaric, "Stability of quantum motion and correlation decay," *Journal of Physics A: Mathematical and General*, vol. 35, no. 6, p. 1455, 2002.

[9] M. Žnidarič, *Stability of Quantum Dynamics*. PhD thesis, University of Ljubljana, June 2004.

[10] K. R. Brown, A. C. Wilson, Y. Colombe, C. Ospelkaus, A. M. Meier, E. Knill, D. Leibfried, and D. J. Wineland, "Single-qubit-gate error below $\mathbf{10^{-4}}$ in a trapped ion," *Phys. Rev. A*, vol. 84, p. 030303, Sep 2011.

[11] E. Gerjuoy, "Shor's factoring algorithm and modern cryptography. an illustration of the capabilities inherent in quantum computers," *American Journal of Physics*, vol. 73, no. 6, pp. 521–540, 2005.

[12] I. García-Mata, K. M. Frahm, and D. L. Shepelyansky, "Effects of imperfections for shor's factorization algorithm," *Phys. Rev. A*, vol. 75, p. 052311, May 2007.

[13] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, Oct 1995.

[14] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation: error models and thresholds," *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 365–384, 1998.

[15] T. Guhr, A. M. Groeling, and H. A. Weidenmaller, "Random-matrix theories in quantum physics: common concepts," *Physics Reports*, vol. 299, no. 4â6, pp. 189 – 425, 1998.

[16] C. Pineda, *One, Two, and n Qubit Decoherence*. PhD thesis, Universidad Nacional Autónoma de México, December 2007.

[17] G. L. Celardo, C. Pineda, and M. Znidaric, "Stability of quantum fourier transformation on ising quantum computer," *International Journal of Quantum Information*, vol. 3, p. 441, 2005.

[18] T. Prosen and M. Znidaric, "Can quantum chaos enhance the stability of quantum computation?," *Journal of Physics A: Mathematical and General*, vol. 34, no. 47, p. L681, 2001.

[19] I. García-Mata, K. M. Frahm, and D. L. Shepelyansky, "Shor's factorization algorithm with a single control qubit and imperfections," *Phys. Rev. A*, vol. 78, p. 062323, Dec 2008.

[20] G. Benenti, G. Casati, S. Montangero, and D. L. Shepelyansky, "Efficient quantum computing of complex dynamics," *Phys. Rev. Lett.*, vol. 87, p. 227901, Nov 2001.

[21] M. Terraneo and D. L. Shepelyansky, "Imperfection effects for multiple applications of the quantum wavelet transform," *Phys. Rev. Lett.*, vol. 90, p. 257902, Jun 2003.

[22] R. B. Griffiths and C.-S. Niu, "Semiclassical fourier transform for quantum computation," *Phys. Rev. Lett.*, vol. 76, pp. 3228–3231, Apr 1996.