Universidad Politécnica de Madrid
ETS de Ingeniería de Sistemas Informáticos

Propagación de errores
en circuitos cuánticos e isotropía

Tesis doctoral

André Luiz Fonseca de Oliveira
Ingeniero Electricista

2016

Departamento de Sistemas Informáticos
ETS de Ingeniería de Sistemas Informáticos

Propagación de errores
en circuitos cuánticos e isotropía

Tesis doctoral

André Luiz Fonseca de Oliveira
(Ingeniero Electricista)

Director: Jesús García López de Lacalle
(Doctor en Ciencias Matemáticas)

2016

Tribunal nombrado por el Sr. Rector Magfco. de la Universidad
Politécnica de Madrid, el día..... de..... de 20.....

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Suplente:

Suplente:

Realizado el acto de defensa y lectura de la Tesis el día..... de.....
de 20..... en la ETS de Ingeniería de Sistemas Informáticos.

Calificación.....

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

A mi familia

Agradecimientos

Quisiera agradecer a todos que, durante el largo tiempo que ha llevado la realización de esta investigación, han aportado en forma directa, e indirecta, al éxito de este trabajo de tesis.

Primero quiero expresar mi sincera gratitud al Dr. Jesús García López de Lacalle, excelente tutor y amigo, con quién tuve mi primer curso en el área, en Uruguay. Su incentivo, paciencia y aportes fueron fundamentales. También me gustaría mencionar al personal del Departamento de Matemática Aplicada a las Tecnologías de la Información y las Comunicaciones adscrito a la ETSISI, quienes siempre hicieron que me sintiera como en casa durante mis diversas instancias en los años que ha llevado este trabajo. Mil gracias a Gregoria Blanco Viejo, José Juan Carreño Carreño, Alfonsa García López, Francisco García Mazarío, Rafael Miñano Rubio, Ana Isabel Lías Quintero y a Ángeles Martínez Sánchez. Y mi agradecimiento especial a Beatriz Peña Muñoz y a Julio Blanco Martín.

Mi sincera gratitud a la Universidad ORT Uruguay por el apoyo brindado durante estos años durante los cuales, sumando todas mis visitas, estuve más de un año en Madrid. También a mi compañero de investigación en el área, el Dr. Efraim Buksman Hollander, con quién hemos creado el Grupo de Computación e Información Cuántica en la universidad. En el trabajo de este grupo se han desarrollado los resultados de esta tesis. En este marco quisiera reconocer la participación de los estudiantes de maestría y graduación, los cuales participaron de nuestras investigaciones en el marco del grupo, con valiosos intercambios de ideas y sugerencias. Mis gracias para Fabián Kozynski, Ismael Rodríguez Brena, Laura Gatti, Ilan Cohn y Emiliano Espíndola.

Quiero agradecer al Ministerio de Educación y Cultura del Uruguay por el apoyo brindado mediante el Programa de Desarrollo Tecnológico (PDT), mediante la beca S/C/BE/55/23, que me ha permitido mis primeras estancias en Madrid. También mi agradecimiento al Banco Santander y a la Universidad Politécnica de Madrid por el apoyo mediante la beca BSCH-UPM en la convocatoria 2013/2014.

Finalmente, pero muy especialmente, a mi familia. Gracias a mi esposa Daniela, y a mi hijo Cándido Lucas, por el apoyo y la paciencia en comprender mi ausencia en muchos momentos estos últimos años.

Resumen

Esta tesis trata del desarrollo de herramientas que faciliten el análisis de la eficiencia y deterioro de los algoritmos cuánticos en sistemas abiertos. Una de las características de los sistemas cuánticos, la correlación, es considerada como una de las principales causas de la mejora de desempeño de los algoritmos cuánticos en comparación con sus homólogos clásicos. Una de las grandes dificultades en el estudio de su influencia es la diversidad de propuestas de medidas existentes, no siempre equivalentes, y la imposibilidad de comparación entre sistemas de diferentes dimensiones utilizando estas medidas. Otro factor, la presencia de errores provenientes de diversas causales, es de fundamental análisis para la determinación de las limitaciones existentes en las propuestas de algoritmos cuánticos.

En esta tesis se propone una medida de correlación multiqubit para estados cuánticos mezcla. Esta medida es definida en forma recursiva, acumulando las correlaciones de los subespacios, simplificando el cálculo sin la necesidad del uso de regresión no lineal. A diferencia de las propuestas existentes, esta medida es continua, aditiva y refleja la dimensión del espacio, permitiendo la comparación entre estados de diferentes dimensiones.

Con la finalidad de caracterizar, y analizar, la propagación de errores en los algoritmos cuánticos se presentan índices de isotropía. En particular, se presenta un nuevo índice que separa un estado mezcla en dos componentes. El primero, o componente isótropo, cuantifica la falta de información existente en el estado. El segundo, la alineación, representa la desviación del estado respecto a un estado puro de referencia. Este doble índice se representa mediante una gráfica triangular.

Utilizando las técnicas propuestas se analizan dos ejemplos de algoritmos cuánticos de búsquedas, considerando sistemas abiertos.

En el primer estudio se analiza el cambio de desempeño del algoritmo de Grover con error modelado como un canal de despolarización del estado completo y como un error de despolarización local en cada qubit. En objetivo no es la corrección del error, sino analizar cómo se caracteriza la degradación del desempeño debido a cada tipo de error.

En el segundo se analiza un algoritmo de Tulsi modificado mediante mediciones parciales en el qubit de control. Se investiga con diferentes valores del intervalo de tiempo entre mediciones, determinando relaciones existentes entre la probabilidad de éxito y las correlaciones existentes en el estado cuántico del algoritmo. Se estima el orden en cada caso, resultando que para determinada elección de parámetros el algoritmo con mediciones parciales resulta con mejor desempeño que en un sistema cerrado.

Como parte del trabajo de tesis se ha desarrollado un simulador para computación cuántica, *QuantumLab V2.1*, utilizado en todos los trabajos presentados.

Abstract

This thesis deals with the development of tools that facilitate the analysis of the efficiency and deterioration of quantum algorithms in open systems. One of the characteristics of quantum systems, correlation, is considered as the main cause of performance improvement of quantum algorithms, compared to their classical counterparts. A great difficulty in the study of its influence is the diversity of existing correlation measures, not always equivalent, and the impossibility of comparison between systems with different dimensions using these measures. Furthermore, the presence of errors from various causes, must be analyzed in order to determine the limitations in quantum algorithms.

At first a measure of correlation for multiqubit mixed states is presented. The measure is defined recursively, accumulating the correlation of the subspaces, making it simple to calculate without the use of regression. Unlike usual measures, the proposed measure is continuous, additive and reflects the dimensionality of the state space, allowing to compare states with different dimensions.

Isotropic indexes are proposed in order to characterize, and analyze, the propagation of errors in the quantum algorithms. In particular, a novel index that separates the resultant mixed error state in two components is presented. The first component, or isotropic component, quantifies the lack of information. The second one, the dis-alignment component, represents the shift between the current state and the original pure quantum state. The Isotropic Triangle, a graphical representation that fits naturally with the proposed index, is also introduced.

Using the proposed techniques, two examples of quantum search algorithms are analyzed, considering open systems.

In the first example the effect of noise on Grover's algorithm is analyzed, modeled as a total depolarizing channel and as a local depolarizing channel in each qubit. The focus was not in error correction (e.g. by the fault-tolerant method), but to provide an insight to the kind of error, or degradation, that needs to be corrected.

In the second example a modification of Tulsi's quantum search algorithm with intermediate measurements of the control qubit is presented. In order to analyze the effect of measurements in quantum searches, a different choice of the angular parameter is used. The study is performed for several values of time lapses between measurements, finding close relationships between probabilities and correlations. The order of this modified algorithm is estimated, showing that for some time lapses the performance is improved, and becomes of order $\Theta(N)$ (classical brute-force search) when measurements are taken in every step.

As part of this thesis, a simulator for quantum computing has been developed, *QuantumLab V2.1*, and it has been used in all the former examples.

Índice general

Agradecimientos	V
Resumen	VI
Abstract	VII
I Introducción y preliminares	1
1. Introducción	2
1.1. Motivación	2
1.2. Objetivos	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	3
1.3. Organización del trabajo	3
2. Fundamentos de la computación e información cuántica	6
2.1. Postulado 1. El espacio de estados	6
2.1.1. El qubit	6
2.1.2. Operadores en el espacio de Hilbert	7
2.1.3. Matrices de densidad	9
2.2. Postulado 2. Evolución de sistemas cuánticos	11
2.3. Postulado 3. Medidas cuánticas.	11
2.3.1. Medidas proyectivas.	12
2.4. Postulado 4. Sistemas compuestos.	13
2.4.1. Traza parcial	14
2.4.2. Entrelazamiento y estados separables	14
II Correlaciones y caracterización de errores en sistemas cuánticos	16
3. Correlaciones en estados cuánticos	17
3.1. Entrelazamiento geométrico para estados puros	17
3.1.1. Medida de entrelazamiento geométrico	18
3.1.2. Ejemplos	19
3.1.3. Ejemplo 3	20
3.2. Medida de correlación cumulativa	21
3.2.1. Introducción	21
3.2.2. Definición	21
3.2.3. Propiedades de CCM	22
3.2.4. CCM para estados GHZ	23
3.2.5. CCM y detección de transiciones de fase cuántica	25

4. Errores en sistemas cuánticos	29
4.1. Errores de decoherencia	29
4.2. Representación de suma de operadores	30
4.2.1. Operadores de error	31
4.3. Errores continuos	33
4.3.1. Modelos de errores cuánticos continuos de 1 qubit	33
4.3.2. Modelos de errores continuos en n qubits	34
4.4. Errores isótropos	36
4.4.1. Errores isótropos para estados cuánticos con notación vectorial	38
4.4.2. Errores isótropos para estados cuánticos con representación de matriz de densidad	39
5. Índices de isotropía para errores cuánticos	44
5.1. Índice de isotropía geométrico	44
5.1.1. Aproximaciones numéricas	45
5.1.2. Ejemplos	46
5.1.3. Error isótropo en el entorno del estado $ 0\rangle$	46
5.2. Índice de isotropía estadístico	47
5.2.1. Descomposición de un estado mezcla	48
5.2.2. Definición del índice	48
5.2.3. Representación gráfica: triángulo de isotropía	50
5.2.4. Estado isótropo de Horodecki	53
III Algoritmos cuánticos en sistemas abiertos	55
6. Algoritmo de Grover	56
6.1. Descripción del algoritmo de Grover	56
6.2. Error de despolarización de estado completo (TDCh)	57
6.2.1. Evolución del estado	58
6.2.2. Estimación del orden	59
6.3. Error local de despolarización de canal (LDCh)	61
6.3.1. OSR para la evolución del estado con LDCh	62
6.3.2. Cotas de probabilidad	62
6.3.3. Aproximación de primer orden para el caso LDCh	63
6.3.4. Estimación del orden	64
6.4. Interpretación de la degradación utilizando el índice de isotropía	65
6.4.1. Estado original del algoritmo sin ruido como estado de referencia	65
6.4.2. Estado marcado como estado de referencia	66
7. Caminatas cuánticas	69
7.1. Algoritmo de búsqueda de Tulsi	69
7.2. Algoritmo de búsquedas cuánticas con mediciones intermedias (IMA)	71
7.3. Estudio de probabilidades y correlaciones (IMA)	72
7.3.1. Correlaciones	74
7.3.2. P_t y correlaciones	74
7.3.3. P_c y correlaciones	76
7.3.4. Número de pasos y correlación	77
7.4. Estimación del orden	78

IV Conclusiones	82
8. Conclusiones	83
8.1. Conclusiones	83
8.1.1. Correlaciones cuánticas	83
8.1.2. Isotropía	83
8.1.3. Algoritmo de Grover	84
8.1.4. Caminatas cuánticas	84
8.1.5. Simulador	84
8.2. Futuras líneas de trabajo	85
8.3. Publicaciones	86
8.3.1. Revistas científicas	86
8.3.2. Congresos	86
8.4. Agradecimientos	87
A. Simulador	88
A.1. Funciones para la definición y manejo de estados cuánticos	89
A.2. Funciones para la definición de compuertas cuánticas	109
A.3. Funciones para la definición de medidas cuánticas	118
A.4. Funciones para la definición de operadores de error en estados cuánticos	123
A.5. Funciones para la operación de circuitos cuánticos	139
A.6. Funciones complementarias (útiles)	143
A.7. Funciones para cálculo de distancias y correlaciones cuánticas	156
A.8. Funciones para el cálculo de la isotropía cuántica	161
B. Cálculos y demostraciones	166
B.1. Propiedades de la CCM	166
B.2. Cálculo eficiente del índice de isotropía estadístico	169
C. Algoritmo híbrido	170
C.1. Algoritmo algebraico	170
C.2. Algoritmo genético	171
C.3. Optimización con gradiente	172

Índice de figuras

2.1. Esfera de <i>Bloch</i>	7
2.2. Esfera de <i>Bloch</i> . Estados puros y mezclas.	10
3.1. Compuerta para el sistema C_α	19
3.2. GME. Figuras de los ejemplos 1 y 2.	20
3.3. GME. Figuras del ejemplo 3.	21
3.4. Implementación de una compuerta de Toffoli con variación continua.	21
3.5. Experiencia del ejemplo 4 con un sistema $T_{\alpha,\beta}$	21
3.6. GME. Figura del ejemplo 4.	22
3.7. Curva de correlaciones del hamiltoniano XXZ.	26
3.8. Detalle de la curva de correlaciones. Modelo XXZ.	26
3.9. Curva de correlaciones. Modelo XXZ doble.	27
3.10. Curva de correlaciones. Modelo XXZ en presencia de ruido.	28
3.11. Modelo transversal de Ising y la <i>CCM</i>	28
4.1. Modelo de error de decoherencia.	30
4.2. Operaciones de bit flip, phase flip y bit-phase flip.	32
4.3. Operación de depolarizing channel.	32
4.4. Operaciones de amplitude damping y fase damping.	33
4.5. Operación de generalized amplitude damping.	34
4.6. Error continuo. Estado original $ 0\rangle$	35
4.7. Error continuo. Estado original $(\sqrt{2}/2)(0\rangle + 1\rangle)$	35
4.8. Error continuo. Estado original $(\sqrt{5}/5)(2 0\rangle + 1\rangle)$	36
4.9. Error continuo $E = X_{\alpha_X} Y_{\alpha_Y} Z_{\alpha_Z}$ con estado original $q = 0\rangle$	36
4.10. Error continuo $E = X_{\alpha_X} Y_{\alpha_Y} Z_{\alpha_Z}$ con estado original $q = (\sqrt{5}/5)(2 0\rangle + i 1\rangle)$	37
4.11. Error continuo ($p = 0,95$).	37
4.12. Error continuo ($p = 0,99$).	37
4.13. Error isótropo. 500 puntos alrededor del estado $ 0\rangle$	39
4.14. Ejemplos de errores isótropos.	40
4.15. Error isótropo uniforme en el entorno del estado $ 0\rangle$	41
4.16. Error isótropo ocasionado por ensamble no isótropo.	42
5.1. Estados con distribución de probabilidad $f(\varphi, \theta)$	44
5.2. Diferencia entre los factores i_φ e i_θ del doble índice isótropo geométrico.	45
5.3. Error isótropo uniforme alrededor del estado de referencia $ 0\rangle$	47
5.4. Variación del índice de isotropía en función del ángulo θ_{max}	49
5.5. Triángulo de isotropía.	51
5.6. Triángulo de isotropía. Propiedades 1 y 2.	52
5.7. Triángulo de isotropía. Propiedades 3 y 4.	53
5.8. Triángulo de isotropía. Probabilidades constantes.	53
5.9. Triángulo de isotropía. Variación de la probabilidad.	54
5.10. Estado isótropo de Horodecki y triángulo.	54

6.1.	Hiperplano del algoritmo de Grover.	57
6.2.	Modelo de error TDCh para el algoritmo de Grover.	58
6.3.	Probabilidades para Grover con error TDCh.	59
6.4.	Cantidad de pasos para máxima probabilidad en Grover con error TDCh	60
6.5.	Modelo de error LDCh para el algoritmo de Grover.	61
6.6.	Árbol de análisis para Grover con LDCh.	63
6.7.	Grover con error LDCh. Cotas de probabilidad.	64
6.8.	Grover con error TDCh. Isotropía con referencia $p(k)$	66
6.9.	Grover con error LDCh. Isotropía con referencia $p(k)$	67
6.10.	Grover con error TDCh. Isotropía con estado marcado como referencia.	68
6.11.	Grover con error LDCh. Isotropía con estado marcado como referencia.	68
7.1.	Esquema del algoritmo de Tulsi.	70
7.2.	Probabilidad en el algoritmo de Tulsi.	71
7.3.	Esquema del Algoritmo con Mediciones Intermedias (<i>IMA</i>).	71
7.4.	Algoritmo IMA_T . Probabilidad acumulada.	73
7.5.	Algoritmo $IMA_{\pi/4}$. Probabilidad del elemento marcado.	73
7.6.	Algoritmo $IMA_{\pi/4}$. Probabilidad acumulada.	74
7.7.	Probabilidad P_t y correlaciones	75
7.8.	Probabilidad del qubit de control	75
7.9.	Probabilidad P_c y correlaciones	76
7.10.	Probabilidad P_t y correlaciones para evolución unitaria	77
7.11.	Comparación entre correlaciones y la estimación del orden.	78
7.12.	Comparación entre TS y $MI_{c \otimes p}$	79
7.13.	Cantidad total de pasos TS en función de N	80
7.14.	Estimación del orden utilizando TS	81
A.1.	Estructura de rutinas del simulador.	89
A.2.	Estado <i>tipo 1</i>	89
A.3.	Estado <i>tipo 2</i>	89
A.4.	Estado <i>tipo 3</i>	90
A.5.	Estado <i>tipo 4</i>	90
A.6.	Errores generados con la función $qb_err_contIsoind$	124
A.7.	Errores generados con la función $qb_err_contXYZind$	126
A.8.	Comparación de errores: $qb_err_contXYZind$ y $qb_err_contXYZcorr$	127
A.9.	Operación del depolarizing channel.	129
A.10.	Operaciones de bit flip, phase flip y bit-phase flip.	133
A.11.	Operaciones de amplitude damping y fase damping.	135
A.12.	Operación de generalized amplitude damping.	136
A.13.	Ejemplos de la esfera de Bloch para estados puros.	152
A.14.	Ejemplos de la esfera de Bloch para estados mezclas.	152
A.15.	Estados generados para el ejemplo A.8.1.	163
A.16.	Triángulos de isotropía para el ejemplo A.8.3.	165

Índice de cuadros

3.1. Valores de CCM para estados GHZ.	25
5.1. Valores de las aproximaciones numericas	48

Parte I

Introducción y preliminares

Capítulo 1

Introducción

1.1. Motivación

En 1965 Gordon Moore postuló que aproximadamente cada año se duplicaría la cantidad de transistores en un circuito integrado, predicción posteriormente corregida en 1975 para la duplicación en un período de dos años [81]. Esta afirmación conocida como *Ley de Moore* se ha cumplido en las últimas décadas y posee una fecha de caducidad expresada por su propio autor: Moore afirmó, en 2007, que su ley dejaría de cumplirse en los próximos 10 o 15 años. Esto es debido a los límites de la microelectrónica, o sea, la velocidad de la luz y la naturaleza atómica de la materia. Al día de hoy estamos muy cercanos a estos límites: en los niveles actuales de integración ya aparecen efectos cuánticos indeseados que limitan la computación clásica.

A mediados de los años 80, Richard Feynman propone la idea de la utilización de sistemas cuánticos como una herramienta de cálculo [45] [46], utilizando las leyes de la mecánica cuántica. En la misma década Paul Benioff y David Deutsch trabajan con propuestas de modelos de computadoras cuánticas, siendo Deutsch el primero en formular un algoritmo cuántico [19] [41]. Nace así la *Computación Cuántica*, un área de estudio en la intersección de las ciencias de la computación y la mecánica cuántica.

En 1994 Peter Shor propone un algoritmo para la factorización de números muy grandes que muestra ser más eficiente que cualquier algoritmo clásico conocido [105] [107]. Lov Grover, en 1996, propone un algoritmo cuántico para la búsqueda de un elemento en una base desordenada, el cuál es de orden $\Theta(\sqrt{N})$ frente a su homólogo clásico de orden $\Theta(N)$ [58]. Así, en las últimas décadas han surgido algoritmos cuánticos que han demostrado el potencial de esta nueva tecnología.

Pero todo sistema físico es vulnerable a errores y los sistemas cuánticos no son la excepción. Deben ser estudiados, analizados y controlados para poder alcanzar la eficiencia deseada. El ruido cuántico aparece debido a la interacción con el ambiente, a los errores de compuertas, a la preparación y medición de los estados. Diversas propuestas han aparecido con la finalidad de acotar, proteger y controlar la presencia de errores en los sistemas cuánticos. Los códigos cuánticos correctores y la computación tolerante a fallos son algunos de los ejemplos de las técnicas propuestas para este fin [28] [55] [56] [72] [74] [106].

Todavía existen diversas interrogantes sobre el potencial y las limitaciones de la computación cuántica. El desarrollo de herramientas que permitan profundizar en el análisis de la importancia de la correlación en los estados cuánticos y las limitaciones impuestas por la propagación de errores es fundamental para el futuro del área.

Esta es la principal motivación de este trabajo de tesis.

1.2. Objetivos

1.2.1. Objetivo general

El objetivo principal de esta tesis es la generación de herramientas que permitan analizar en forma cuantitativa, y cualitativa, la computación cuántica vista no solamente como la evolución en un sistema cerrado, sino como un sistema abierto debido a la influencia de las mediciones, la decoherencia y los diferentes modelos de errores. Así, es posible estimar la degradación, o mejora, de los algoritmos cuánticos debido a estos fenómenos.

También se ha definido como objetivo el desarrollo de un simulador para algoritmos cuánticos que fuera basado en un ambiente de cálculo numérico abierto y público. La elección, como se detalla en esta tesis, ha sido desarrollar utilizando el ambiente *Scilab*.

1.2.2. Objetivos específicos

Los objetivos específicos fueron determinados de forma que posibilitaran el desarrollo de la investigación considerando el objetivo principal. Estos son:

- definición de una medida de correlación para estados cuánticos (o entrelazamiento para estados puros) que permita la comparación entre estados de diferentes dimensiones,
- definición de una medida (índice) de isotropía para estados cuánticos que refleje la deformación geométrica ocasionada por los errores en los circuitos cuánticos,
- análisis de algoritmos cuánticos de búsquedas en sistemas abiertos, como la propagación de errores o la medición de resultados parciales, y la vinculación con el índice de isotropía propuesto y con la medida de correlación,
- y la construcción de un simulador para circuitos y algoritmos cuánticos basado en un sistema de cálculo numérico abierto, de acceso libre, que posibilite la ampliación y cooperación por parte de terceros.

1.3. Organización del trabajo

A continuación se describe la estructura de esta tesis por capítulos, comentando las principales contribuciones realizadas:

- **Parte I. Introducción y preliminares.**
 - **Capítulo 1. Introducción.**
Describe la importancia del problema a estudiar, los objetivos del trabajo de tesis y la organización de este documento.
 - **Capítulo 2. Fundamentos de la computación e información cuántica.**
En este capítulo se realiza un breve resumen introductorio de los fundamentos de la computación cuántica que sirven como base para una mejor comprensión de los trabajos realizados y del simulador desarrollado.
- **Parte II. Correlaciones y caracterización de errores en sistemas cuánticos.**
 - **Capítulo 3. Correlaciones en estados cuánticos.**
En este capítulo se desarrollan dos resultados importantes de este trabajo de tesis. En primer lugar, se muestra la flexibilidad del simulador desarrollado en la implementación de un algoritmo para la estimación de un índice de entrelazamiento de estados cuánticos puros. Este algoritmo combina las funciones del simulador con una librería de optimización que utiliza un algoritmo híbrido para regresión no lineal. Como segundo resultado,

presenta la propuesta de una medida de correlación cumulativa para estados cuánticos mezcla, que considera la cantidad de qubits del estado (dimensión del espacio de Hilbert del sistema). Esta nueva medida es utilizada en algunos ejemplos en los cuales se ilustra su utilidad para el estudio de sistemas cuánticos.

- **Capítulo 4. Errores en sistemas cuánticos.**

Como parte fundamental en este trabajo se pretende caracterizar la propagación de errores en algoritmos o sistemas cuánticos. Así, en este capítulo se presentan diversos modelos de errores cuánticos implementados en el simulador y utilizados en las experiencias realizadas, con especial atención en la definición e implementación de errores isotrópicos.

- **Capítulo 5. Índices de isotropía para errores cuánticos.**

Con la finalidad de posibilitar el análisis del efecto final del acúmulo y propagación de errores en los sistemas cuánticos, en este capítulo se introducen índices para cuantificar la deformación ocasionada en los resultados y estudiar las limitaciones existentes. En particular, el índice de isotropía estadístico propuesto en la sección 5.2, juntamente con su representación gráfica en el triángulo de isotropía, ha demostrado ser una herramienta interesante en el análisis de propagación de errores, como se ilustra en la parte III de este trabajo.

- **Parte III. Algoritmos cuánticos en sistemas abiertos.**

- **Capítulo 6. Algoritmo de Grover.**

En este primer estudio se analiza el efecto de introducir ruido al algoritmo de búsqueda cuántica propuesto por Grover, utilizando dos modelos de errores: ruido de despolarización del canal para el estado completo (TDCh) e ruido local de despolarización del canal (LDCh). El objetivo no es la corrección del error, pero sí el análisis de su propagación y las limitaciones existentes. Se analiza la degradación del orden del algoritmo con ambos modelos de errores y la explicación para sus diferencias, utilizando el índice de isotropía desarrollado en el capítulo anterior.

- **Capítulo 7. Caminatas cuánticas.**

En este capítulo se analiza el efecto de realizar mediciones parciales utilizando un algoritmo de búsqueda mediante caminatas cuánticas y basado en una modificación del algoritmo de Tulsi. Se analizan las relaciones existentes entre las probabilidades de éxito y las correlaciones existentes en el estado cuántico, o entre algunos de los subsistemas del estado cuántico. Se muestra que, en algunos casos, tener un sistema cuántico abierto (con mediciones) mejora el desempeño final del algoritmo.

- **Parte IV. Conclusiones.**

- **Capítulo 8. Conclusiones.**

En este capítulo se exponen las conclusiones del trabajo de tesis, incluyendo posibles líneas futuras de investigación.

- **Apéndices.**

- **Apéndice A. Simulador.**

La descripción completa del simulador desarrollado, QuantumLab V2.1, se encuentra en este apéndice. Este simulador ha sido utilizado en todos los trabajos de investigación en el marco de esta tesis.

- **Apéndice B. Cálculos y demostraciones.**

Las demostraciones de las propiedades de la medida de correlación cumulativa, propuesta en la sección 3.2, y la forma de cálculo eficiente del índice estadístico de isotropía, propuesto en 5.2, se encuentran en este apéndice.

- **Apéndice C. Algoritmo híbrido.**

En este apéndice se detalla el algoritmo híbrido utilizado en la sección 5.1 para implementar la búsqueda del estado puro separable más cercano a un estado cuántico puro dado. La distancia entre estos estados define una medida de entrelazamiento (medida geométrica de entrelazamiento).

Capítulo 2

Fundamentos de la computación e información cuántica

Con el objetivo de tener un marco para una mejor comprensión del desarrollo del simulador y de los capítulos posteriores se hace necesaria una introducción de los fundamentos de la mecánica cuántica, cuyos principios están basados en los cuatro postulados introducidos en las secciones a continuación. Esta introducción es un resumen extraído de [69][82][89].

2.1. Postulado 1. El espacio de estados

Postulado 2.1.1 (Espacios de Hilbert). *A un sistema físico aislado (sistema cerrado) se le asocia un espacio de Hilbert (espacio vectorial complejo con producto interno, completo bajo la norma inducida), denominado espacio de estados del sistema. Los estados del sistema son vectores unitarios del espacio y describen completamente al sistema.*

2.1.1. El qubit

La estructura básica de información en la computación cuántica es el bit cuántico ó *qubit*. Un qubit es un estado cuántico, que se representa de la forma (notación de Dirac)

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.1)$$

siendo $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$ y

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (2.2)$$

Al ser representados por estados normalizados, los *qbits* pueden ser representados como puntos en una esfera unitaria, la *Esfera de Bloch* ilustrada en la figura 2.1, en función de los vectores de la base dada por la expresión

$$|\psi\rangle = \left(\cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \right) e^{i\gamma}, \quad (2.3)$$

donde la fase global $e^{i\gamma}$, por no tener efectos observables, se puede ignorar.

La notación $\langle\psi|$ es utilizada para representar el vector transpuesto conjugado del estado $|\psi\rangle$

$$\langle\psi| = |\psi\rangle^\dagger = [\alpha^* \ \beta^*], \quad (2.4)$$

siendo el α^* el complejo conjugado de α . Con esta notación, el *producto interno* entre dos vectores $|u\rangle$ y $|v\rangle$ se puede escribir como

 Figura 2.1: Esfera de Bloch y representación del estado $|\psi\rangle$.

$$(u, v) = \langle u | v \rangle = \langle u | v \rangle. \quad (2.5)$$

2.1.2. Operadores en el espacio de Hilbert

Sean $\mathcal{H}_1$ y $\mathcal{H}_2$ dos espacios de Hilbert. Se notará $\Lambda(\mathcal{H}_1, \mathcal{H}_2)$ al conjunto de todas las aplicaciones lineales de $\mathcal{H}_1$ en $\mathcal{H}_2$. Todo operador $T \in \Lambda(\mathcal{H}_1, \mathcal{H}_2)$ puede ser representado por su matriz asociada (también notada T) de forma a que si $T(v_1) = v_2$, $v_1 \in \mathcal{H}_1$ y $v_2 \in \mathcal{H}_2$, se tiene que

$$v_2 = T v_1. \quad (2.6)$$

La composición de operadores es simplemente el producto de sus matrices asociadas.

Definición 2.1 (Operador adjunto). Dado un operador $T \in \Lambda(\mathcal{H}_1, \mathcal{H}_2)$, se define el *operador adjunto* $T^\dagger \in \Lambda(\mathcal{H}_2, \mathcal{H}_1)$ como el operador que cumple

$$(T|u\rangle, |v\rangle)_{\mathcal{H}_2} = (|u\rangle, T^\dagger|v\rangle)_{\mathcal{H}_1}, \quad (2.7)$$

siendo $(\cdot, \cdot)_{\mathcal{H}_i}$ el producto interno en $\mathcal{H}_i$. Este operador siempre existe y es único. Además dadas dos bases ortonormales en $\mathcal{H}_1$ y $\mathcal{H}_2$ la matriz asociada al operador adjunto es la matriz transpuesta y conjugada de la matriz asociada al operador original.

Propiedad 2.1 (Propiedades del operador adjunto). $\forall T, S \in \Lambda(\mathcal{H}_1, \mathcal{H}_2)$ y $\alpha \in \mathbb{C}$

- $(T + \alpha S)^\dagger = T^\dagger + \alpha^* S^\dagger$,
- $(TS)^\dagger = S^\dagger T^\dagger$,
- $(T^\dagger)^\dagger = T$,
- T es invertible si y solo si $T^\dagger$ lo es, y $(T^\dagger)^{-1} = (T^{-1})^\dagger$.

Definición 2.2 (Operador normal). Un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ es *normal* si $TT^\dagger = T^\dagger T$.

Definición 2.3 (Operador hermítico). Un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ es *hermítico* si coincide con su adjunto, o sea, $T^\dagger = T$.

Propiedad 2.2 (Propiedades del operador hermítico).

- Si un operador es hermítico es normal,
- las matrices asociadas a operadores hermíticos en bases ortonormales son matrices hermíticas independientemente de las bases elegidas, $M = M^\dagger$,

- los valores propios de un operador hermítico son reales.

Definición 2.4 (Operador unitario). Un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ es *unitario* si es un operador biyectivo isométrico, o sea, $\|T|u\rangle\| = \|u\|$.

Propiedad 2.3 (Propiedades del operador unitario).

- Si un operador es unitario es normal,
- los operadores unitarios preservan el producto interno, $(Tu, Tv) = \langle Tu|Tv\rangle = \langle u|v\rangle$,
- las matrices asociadas a operadores unitarios en bases ortonormales son matrices unitarias, $UU^\dagger = U^\dagger U = I$, siendo I la matriz identidad,
- los valores propios de un operador unitario tienen módulo uno.

Definición 2.5 (Operador positivo). Un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ es *positivo* ($T \geq 0$) si $\forall |v\rangle \in \mathcal{H}$, $\langle Tv|v\rangle \geq 0$. En el caso de $\langle Tv|v\rangle = 0$ si y solo si $|v\rangle = 0$, T es denominado definido positivo.

Observación 2.1. Sean $T, S \in \Lambda(\mathcal{H}, \mathcal{H})$. Se tiene que

- $TT^\dagger$ y $T^\dagger T$ son operadores positivos,
- Si T y S son operadores positivos, y $\lambda \in \mathbb{R} \geq 0$, entonces $\lambda T + S$ es un operador positivo.

Teorema 2.1. Si un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ tiene todos los valores propios reales no negativos, entonces T es un operador positivo.

Definición 2.6 (Producto externo). Se denomina *producto externo* a la función $(|u\rangle, |v\rangle) \rightarrow |u\rangle\langle v|$, que va de $\mathcal{H} \times \mathcal{H}$ a $\Lambda(\mathcal{H}, \mathcal{H})$.

Propiedad 2.4 (Propiedades del producto externo).

- Es lineal en la primera componente y lineal conjugada en la segunda,
- $(|u\rangle\langle v|)^\dagger = |v\rangle\langle u|$,
- $|u\rangle\langle u|$ es un operador hermítico positivo,
- dado un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$, $T|u\rangle\langle v| = |Tu\rangle\langle v|$ y $|u\rangle\langle v|T = |u\rangle\langle T^\dagger v|$.

Definición 2.7 (Operador diagonalizable). Un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ es *diagonalizable* si existe una base ortonormal de $\mathcal{H}$ conformada de vectores propios $|i\rangle$ de T , asociados a los valores propios λ_i .

Un operador diagonalizable T puede ser expresado como $T = \sum_i \lambda_i |i\rangle\langle i|$. A los operadores $|i\rangle\langle i|$ se les denomina los *proyectores* en los subespacios asociados a los valores propios λ_i .

Teorema 2.2 (Descomposición espectral). Todo operador normal en un espacio de Hilbert (de dimensión finita) es diagonalizable. Recíprocamente cualquier operador diagonalizable respecto a una base ortonormal es un operador normal.

Corolario 2.1. Cualquier operador hermítico de un espacio de Hilbert es diagonalizable. Cualquier operador diagonalizable respecto a una base ortonormal cuyos valores propios son todos reales es un operador hermítico.

Corolario 2.2. Cualquier operador unitario de un espacio de Hilbert es diagonalizable. Cualquier operador diagonalizable respecto a una base ortonormal que posee todos los valores propios con módulos iguales a uno es un operador unitario.

Definición 2.8 (Traza de un operador). Dado un operador $T \in \Lambda(\mathcal{H}, \mathcal{H})$ se define la función traza, $tr : \Lambda(\mathcal{H}, \mathcal{H}) \rightarrow \mathbb{C}$, como la suma de la diagonal de la matriz asociada al operador.

Propiedad 2.5 (Propiedades de la traza). $\forall T, S \in \Lambda(\mathcal{H}, \mathcal{H})$ y $\alpha \in \mathbb{C}$

- La traza no depende de la base ortonormal elegida para la representación con la matriz asociada al operador,
- $\text{tr}(T^\dagger) = \text{tr}(T)^*$,
- la traza es cíclica: $\text{tr}(TS) = \text{tr}(ST)$,
- dado un operador U invertible $\text{tr}(T) = \text{tr}(UTU^{-1})$,
- $\text{tr}(T|u\rangle\langle v|) = \langle v|Tu\rangle, \quad \forall |u\rangle, |v\rangle \in \mathcal{H}$,
- La suma de los valores propios de un operador T , incluyendo las multiplicidades algebraicas, es igual a $\text{tr}(T)$.

2.1.3. Matrices de densidad

Es posible formular una descripción de sistemas cuánticos utilizando los denominados *operadores de densidad*, o *matrices de densidad*. A diferencia de la descripción introducida en 2.1.1, que utiliza vectores, esta descripción es más general, pudiendo utilizarse tanto para operaciones cuánticas en sistemas cerrados como abiertos.

Una nueva formulación del primer postulado sería:

Postulado 2.1.2. *A un sistema físico se le asocia un espacio de Hilbert ($\mathcal{H}$), denominado espacio de estados del sistema. El sistema es completamente descrito por los operadores de densidad, operadores positivos de traza unitaria, actuando sobre el espacio $\mathcal{H}$.*

Definición 2.9. Dado un estado cuántico $|\psi\rangle$ se define su *matriz de densidad* asociada como el operador $\rho_\psi = |\psi\rangle\langle\psi| \in \Lambda(\mathcal{H}, \mathcal{H})$. En este caso se dice que ρ_ψ es un operador de un estado cuántico puro.

Propiedad 2.6. Si ρ_ψ es una matriz de densidad de un estado puro entonces

- el operador de densidad es hermítico,
- el operador de densidad tiene un único valor propio igual a 1 (los demás valen 0). Luego tiene traza igual a 1,
- $\rho_\psi^2 = \rho_\psi$, luego es un operador positivo.

Definición 2.10. Sea un conjunto de estados cuánticos $|\psi_j\rangle$ y una probabilidad p_j asociada a cada estado del conjunto, tal que $\sum_j p_j = 1$. Se define el *estado mezcla* ρ a partir del ensamble de estados cuánticos $\{p_j, |\psi_j\rangle\}$ como el operador

$$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|. \quad (2.8)$$

Observación 2.2. Un estado mezcla es una combinación lineal convexa de estados puros.

Teorema 2.3. *Un operador $\rho \in \Lambda(\mathcal{H}, \mathcal{H})$ es un operador de densidad asociado a un ensamble de estados cuánticos (estado cuántico mezcla) si y solo si*

- ρ es un operador hermítico,
- $\text{tr}(\rho) = 1$,
- $\rho \geq 0$.

Observación 2.3. Si el operador ρ es una matriz de densidad de un estado cuántico entonces $\text{tr}(\rho^2) \leq 1$, y vale 1 si y solo si está asociada a un estado puro.

Observación 2.4. Un operador de densidad ρ puede ser expresado como

$$\rho = \frac{1}{2} (S_I I + S_X X + S_Y Y + S_Z Z) \quad (2.9)$$

siendo I la matriz identidad, X , Y y Z los operadores de Pauli¹

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad (2.10)$$

y S_j los parámetros de Stokes, calculados como

$$S_j = \text{tr}(\rho j), \quad \text{con } j = I, X, Y, Z. \quad (2.11)$$

Los parámetros S_X , S_Y y S_Z pueden ser utilizados como coordenadas en $\mathbb{R}^3$ de un vector que representa el estado cuántico en la esfera de Bloch. Se tiene que $\sqrt{(S_X^2 + S_Y^2 + S_Z^2)} \leq 1$ y la igualdad solo es verdadera en el caso de estados puros.

Ejemplo 2.1. Sea un ensamble de los estados cuánticos puros

$$|\psi_1\rangle = \sqrt{\frac{1}{2}}|0\rangle + \sqrt{\frac{1}{2}}|1\rangle \quad \text{y} \quad |\psi_2\rangle = \sqrt{\frac{1}{2}}|0\rangle + e^{i\frac{\pi}{2}}\sqrt{\frac{1}{2}}|1\rangle, \quad (2.12)$$

cada uno con probabilidad asociada igual a 0,5. Las matrices de densidad son respectivamente

$$\rho_{\psi_1} = \begin{bmatrix} 0,5 & 0,5 \\ 0,5 & 0,5 \end{bmatrix}, \quad \text{y} \quad \rho_{\psi_2} = \begin{bmatrix} 0,5 & -0,5i \\ 0,5i & 0,5 \end{bmatrix}, \quad (2.13)$$

y la matriz de densidad del estado mezcla que representa el ensamble es

$$\rho = 0,5\rho_{\psi_1} + 0,5\rho_{\psi_2} = \begin{bmatrix} 0,5 & 0,25 - 0,25i \\ 0,025 + 0,25i & 0,5 \end{bmatrix}. \quad (2.14)$$

En este caso se tiene que $\text{tr}(\rho) = 1$ y $\text{tr}(\rho^2) = 0,75$.

Figura 2.2: Representación de estados en la esfera de Bloch. Los estados puros, vectores de módulo unitario, se representan con puntos negros. Los estados mezcla, vectores con módulo ≤ 1 , se representan con puntos blancos.

¹Los operadores de Pauli también tienen como notación σ_I , σ_X , σ_Y y σ_Z .

2.2. Postulado 2. Evolución de sistemas cuánticos

Postulado 2.2.1 (Evolución de sistemas cuánticos). *La evolución de un sistema cuántico cerrado es descrito mediante una transformación unitaria. Si el estado del sistema en el tiempo t_1 es $|\psi_{t_1}\rangle$, y en un tiempo t_2 evoluciona a $|\psi_{t_2}\rangle$, estos dos estados se relacionan mediante un operador unitario U que depende solo de t_1 y t_2 como $|\psi_{t_2}\rangle = U|\psi_{t_1}\rangle$, siendo U un operador unitario que depende solamente de t_1 y t_2 .*

Observación 2.5. En caso de expresar los estados como matrices de densidad, se tiene que la evolución puede ser calculada como $\rho_2 = U \rho_1 U^\dagger$.

Observación 2.6. La evolución de un sistema cuántico cerrado se puede describir mediante la ecuación de Schrödinger

$$i\hbar \frac{d|\psi\rangle}{dt} = \mathbb{H}|\psi\rangle, \quad (2.15)$$

o, en el caso de una descripción mediante matrices de densidad, como la ecuación de Von Neumann

$$i\hbar \frac{\partial \rho}{\partial t} = \mathbb{H}\rho - \rho\mathbb{H}, \quad (2.16)$$

siendo $\mathbb{H}$ el Hamiltoniano del sistema. La solución de estas ecuaciones son las indicadas con $U = e^{-i(t_2-t_1)H}$. Como H es hermítica se tiene que U es un operador unitario. Debido a esto toda operación cuántica cerrada es reversible mediante el operador $U^\dagger$.

Ejemplo 2.2. Un operador (o compuerta) Hadamard está representado por la matriz unitaria

$$H = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} = \frac{1}{\sqrt{2}} (X + Z). \quad (2.17)$$

Este operador, además de unitario, es hermítico, o sea, $H^\dagger = H$. Luego $H^2 = I$.

Si se aplica este operador al estado $|0\rangle$ (matriz de densidad ρ_0)

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow \rho_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad (2.18)$$

se tiene como resultado el estado $|\psi\rangle$ (matriz de densidad ρ_ψ)

$$|\psi\rangle = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \Rightarrow \rho_\psi = \begin{bmatrix} 0,5 & 0,5 \\ 0,5 & 0,5 \end{bmatrix}. \quad (2.19)$$

2.3. Postulado 3. Medidas cuánticas.

Postulado 2.3.1 (Medidas cuánticas). *Las medidas cuánticas son descritas por una colección de operadores de medida M_m . Estos operadores actúan sobre el espacio de estados del sistema medido. El subíndice m refiere a los posibles resultados que se obtienen una vez efectuada la medida.*

El conjunto de operadores de medida debe cumplir con la propiedad de completitud

$$\sum_m M_m^\dagger M_m = I. \quad (2.20)$$

Esta propiedad garantiza que la suma de las probabilidades sea igual a 1.

La probabilidad de obtener el resultado m al medir un estado cuántico $|\psi\rangle$ puede ser calculada como

$$p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle, \quad (2.21)$$

y el estado final después de realizada la medición es

$$|\psi'\rangle = \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}} \quad (2.22)$$

En el caso de trabajar con estados representados mediante matrices de densidad la probabilidad es calculada por

$$p(m) = \text{tr}(M_m^\dagger M_m \rho), \quad (2.23)$$

y el estado final después de realizada la medición es

$$\rho' = \frac{M_m \rho M_m^\dagger}{\text{tr}(M_m^\dagger M_m \rho)}. \quad (2.24)$$

2.3.1. Medidas proyectivas.

Definición 2.11 (Proyectores). Un *observable* M , operador hermítico en el espacio de estados, tiene descomposición espectral

$$M = \sum_i \lambda_i |i\rangle\langle i| = \sum_i \lambda_i P_i \quad (2.25)$$

siendo $|i\rangle$ los vectores propios de M y λ_i sus valores propios asociados. P_i son los *proyectores ortogonales* sobre los subespacios propios del observable M .

En este caso, al medir un estado $|\psi\rangle$, la probabilidad de tener un resultado en el subespacio i es

$$p(i) = \langle\psi|P_i|\psi\rangle, \quad (2.26)$$

y el estado final

$$|\psi'\rangle = \frac{P_i|\psi\rangle}{\sqrt{p(i)}}. \quad (2.27)$$

Para estados como matrices de densidad se tiene que

$$p(i) = \text{tr}(P_i \rho), \quad (2.28)$$

y

$$\rho' = \frac{P_i \rho P_i}{p(i)}. \quad (2.29)$$

Ejemplo 2.3. Sea el estado $|\psi\rangle$

$$|\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (2.30)$$

Si se utiliza los proyectores P_0 y P_1 definidos como

$$P_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{y} \quad P_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.31)$$

se tiene como resultado

$$p(0) = 0,5 \quad \text{con} \quad |\psi_0\rangle = |0\rangle \quad \text{y} \quad p(1) = 0,5 \quad \text{con} \quad |\psi_1\rangle = |1\rangle. \quad (2.32)$$

2.4. Postulado 4. Sistemas compuestos.

Postulado 2.4.1 (Sistemas compuestos). *El espacio de estados de un sistema físico compuesto es el producto tensorial de los espacios de estado componentes. Si tenemos n sistemas y el i -ésimo sistema se encuentra preparado en el estado $|\psi_i\rangle$, entonces el estado compuesto será en el estado: $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.*

Observación 2.7. Para el caso de matrices de densidad se tiene el estado compuesto $\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n$.

Definición 2.12. Sean V y W dos espacios vectoriales de dimensión n y m , respectivamente, y el espacio vectorial $V \otimes W$ de dimensión $n \times m$. El producto tensorial se define como una función $\otimes : (V \times W) \rightarrow V \otimes W$ tal que al par $(|v\rangle, |w\rangle)$ le asocia el elemento $|v\rangle \otimes |w\rangle$ que cumple las siguientes propiedades:

- La función $\otimes$ es bilineal,
- $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle$,
- $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle$,
- para $c \in \mathbb{R}$, $c(|v\rangle \otimes |w\rangle) = (c|v\rangle) \otimes |w\rangle = |v\rangle \otimes (c|w\rangle)$,
- todo elemento de $V \otimes W$ se puede escribir como combinación lineal de elementos $|v_i\rangle \otimes |w_j\rangle$,
- si $\{|v_i\rangle\}_{1 \leq i \leq n}$ y $\{|w_j\rangle\}_{1 \leq j \leq m}$ son bases de V y W , respectivamente, entonces $\{|v_i \otimes w_j\rangle\}_{1 \leq i \leq n, 1 \leq j \leq m}$ es una base de $V \otimes W$,
- $\| |v\rangle \otimes |w\rangle \| = \| |v\rangle \| \| |w\rangle \|$.

Observación 2.8. Si $\mathcal{H}_1$ y $\mathcal{H}_2$ son espacios de Hilbert, entonces $\mathcal{H}_1 \otimes \mathcal{H}_2$ es también un espacio de Hilbert.

Propiedad 2.7. Sean $\mathcal{H}_1$ y $\mathcal{H}_2$ espacios de Hilbert y los operadores $A \in \Lambda(\mathcal{H}_1, \mathcal{H}_1)$ y $B \in \Lambda(\mathcal{H}_2, \mathcal{H}_2)$. Entonces $\text{tr}(A \otimes B) = \text{tr}(A) \text{tr}(B)$.

Propiedad 2.8. Sean A y $B \in \Lambda(\mathcal{H}, \mathcal{H})$. Entonces si A y B son operadores normales (hermíticos o unitarios), $A \otimes B$ es un operador normal (hermítico o unitario).

Definición 2.13 (Producto de Kronecker). El producto de *Kronecker* es un caso particular de producto tensorial definido para el espacio vectorial de matrices complejas finitas. Dadas las matrices $A \in \mathbb{M}_{n \times m}$ y $B \in \mathbb{M}_{p \times q}$, la matriz $C = A \otimes B$, $C \in \mathbb{M}_{np \times mq}$ se define como

$$C = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{bmatrix}, \quad \text{siendo} \quad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}. \quad (2.33)$$

Propiedad 2.9. El producto de Kronecker tiene las siguientes propiedades

- Es asociativo, $(A \otimes B) \otimes C = A \otimes (B \otimes C)$,
- $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$,
- si A y C son conformables, y B y D también, entonces $(A \otimes B)(C \otimes D) = AC \otimes BD$,
- $A \otimes B$ es invertible si, y solo si, A y B lo son, y $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.

Ejemplo 2.4.

$$|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |01\rangle. \quad (2.34)$$

2.4.1. Traza parcial

Usualmente es necesario trabajar con un subsistema de un sistema cuántico compuesto por dos sistemas físicos A y B . Si se considera el estado original descrito mediante su operador de densidad ρ_{AB} , el subsistema que considera solamente el subsistema físico A queda determinado por el *operador de densidad reducido*, definido como

$$\rho_A = \text{Tr}_B (\rho_{AB}), \quad (2.35)$$

donde Tr_B es la *traza parcial* sobre el sistema B .

Definición 2.14. La traza parcial es un operador que cumple con las siguientes propiedades:

- es lineal,

$$\text{Tr}_B (\rho_{AB} + \lambda \bar{\rho}_{AB}) = \text{Tr}_B (\rho_{AB}) + \lambda \text{Tr}_B (\bar{\rho}_{AB}), \quad (2.36)$$

siendo ρ_{AB} y $\bar{\rho}_{AB}$ operadores en $\mathcal{H}_A \otimes \mathcal{H}_B$ y $\lambda \in \mathbb{C}$,

- si ρ_A y ρ_B son operadores en $\mathcal{H}_A$ y $\mathcal{H}_B$, respectivamente,

$$\begin{aligned} \text{Tr}_B (\rho_A \otimes \rho_B) &= \rho_A \text{tr}(\rho_B), \\ \text{Tr}_A (\rho_A \otimes \rho_B) &= \rho_B \text{tr}(\rho_A). \end{aligned} \quad (2.37)$$

Observación 2.9. En particular, para matrices de densidad se tiene

$$\begin{aligned} \text{Tr}_B (\rho_A \otimes \rho_B) &= \rho_A \\ \text{Tr}_A (\rho_A \otimes \rho_B) &= \rho_B. \end{aligned} \quad (2.38)$$

considerando que $\text{tr}(\rho_A) = \text{tr}(\rho_B) = 1$.

Observación 2.10. La traza parcial es un *operador completamente positivo* que preserva la traza.

Ejemplo 2.5. Sea el estado cuántico

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad (2.39)$$

con matriz de densidad

$$\begin{aligned} \rho &= \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}} \right) \left(\frac{\langle 00| + \langle 11|}{\sqrt{2}} \right) \\ &= \frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2}. \end{aligned} \quad (2.40)$$

En este caso

$$\begin{aligned} \rho_A &= \text{Tr}_B (\rho) \\ &= \frac{\text{Tr}_B (|00\rangle\langle 00|) + \text{Tr}_B (|00\rangle\langle 11|) + \text{Tr}_B (|11\rangle\langle 00|) + \text{Tr}_B (|11\rangle\langle 11|)}{2} \\ &= \frac{|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|}{2} \end{aligned} \quad (2.41)$$

$$= \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} = \frac{I}{2}. \quad (2.42)$$

Como $\text{tr}(\rho_A^2) = 1/2 < 1$, el estado resultante ρ_A es un estado mezcla.

2.4.2. Entrelazamiento y estados separables

Sean dos estados puros definidos en $\mathcal{H}^2$

$$|\psi_1\rangle = \frac{|00\rangle + |01\rangle}{\sqrt{2}} \quad \text{y} \quad |\psi_2\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \quad (2.43)$$

Mientras $|\psi_1\rangle = |0\rangle \otimes (|0\rangle + |1\rangle)/\sqrt{2}$, no existen dos estados $|\varphi_A\rangle$ y $|\varphi_B\rangle$ en $\mathcal{H}$ tales que $|\psi_2\rangle = |\varphi_A\rangle \otimes |\varphi_B\rangle$.

Definición 2.15 (Estados puros entrelazados). A un estado puro al cual no es posible encontrar una descomposición en productos tensoriales de estados de un qubit (estados en $\mathcal{H}$) se le denomina *estado entrelazado*.

Cuando un estado está entrelazado no es posible ver sus subsistemas como sistemas separados, y toda acción que afecte a una parte inevitablemente afectará la otra. La existencia de estados entrelazados (la mayoría de los estados cuánticos) es, juntamente con la propiedad de superposición, el origen de la potencia de la computación cuántica.

Definición 2.16 (Estados puros totalmente separables). Cuando un estado de n -qubits ($\mathcal{H}^n$) se puede descomponer como un producto tensorial de n estados de un qubit se le denomina *estado totalmente separable*.

Definición 2.17 (Estados mezcla totalmente separables). Como los estados mezcla son una suma convexa de estados puros, un estado mezcla es *totalmente separable* cuando *todos* los componentes puros que lo definen lo son. Como muchos ensambles de estados puros pueden tener la misma representación de matriz de densidad, y por ende definir el mismo estado mezcla, se define un estado mezcla ρ como *totalmente separable* si existe una descomposición de la forma

$$\rho = \sum_i p_i \rho_i, \quad p_i \geq 0, \quad \sum_i p_i = 1, \quad (2.44)$$

donde ρ_i son estados puros separables (no entrelazados).

Definición 2.18 (Estados no correlacionados). A un estado representado por su operador de densidad $\rho \in \Lambda(\mathcal{H}^n, \mathcal{H}^n)$ se le dice *no correlacionado* si es posible descomponer su matriz de densidad en un producto tensorial de matrices de estados de un qubit, o sea,

$$\rho = \bigotimes_{i=1}^n \rho_{qi} = \rho_{q1} \otimes \rho_{q2} \cdots \otimes \rho_{qn} \quad (2.45)$$

Observación 2.11. En el caso de estados puros los conceptos de estados separables y no correlacionados coinciden. No es así en el caso de los estados mezcla. Que un estado sea no correlacionado implica que es separable, pero no ocurre al revés: existen estados que son separables pero no son no correlacionados.

Parte II

Correlaciones y caracterización de errores en sistemas cuánticos

Capítulo 3

Correlaciones en estados cuánticos

El término entrelazamiento fue acuñado por Erwin Schrödinger, en alusión a ciertos estados que tenían propiedades de correlación inusuales en la mecánica clásica [103]. Esta propiedad fundamental de algunos estados cuánticos, está relacionada con la ventaja que tiene la computación cuántica sobre su contraparte clásica, así como la relevancia en los protocolos cuánticos de teletransportación, distribución segura de llaves entre partes y el estudio de sistemas de materia condensada a muy bajas temperaturas.

La correlación es una correspondencia o relación recíproca entre dos o más cosas. En un contexto matemático, es una medida de la dependencia existente entre variables aleatorias. Así, en el área de computación cuántica, para estados puros los términos entrelazamiento y correlación cuántica son equivalentes, siendo esta una de las propiedades fundamentales. Por esta razón, cuantificar las correlaciones es una de las tareas más importantes en el área [21].

Existen en la literatura diversas propuestas para estimar la correlación de estados cuánticos: entrelazamiento de formación, entrelazamiento geométrico, discordia cuántica y otros [64][67][80][84][131]. Mientras inicialmente los resultados eran, principalmente, para correlaciones bipartitas, actualmente existen diversas generalizaciones para la correlación multipartita, tanto para el entrelazamiento como para la discordia cuántica [83][97][126]. Esta última es importante en áreas tan diversas como ser el procesamiento de la información cuántica [31][47], la aceleración de algoritmos cuánticos [27] y el estudio de los sistemas de los muchos cuerpos (*many-body systems*) [117].

Una importante limitación actual, dada la complejidad del problema, es que la mayoría de las propuestas sólo pueden ser determinadas analíticamente para unos pocos qubits, o necesitan técnicas de regresión no lineal para su determinación, lo que limita la posibilidad del estudio de sistemas más complejos. Esta limitación, ya existente para estados puros con dimensiones altas (muchos qubits), es especialmente agravada para estados mezcla.

En la sección 3.1 se estudia la correlación para estados puros¹ utilizando una medida de correlación geométrica que utiliza un nuevo algoritmo híbrido para la búsqueda de los estados separables más cercanos. La finalidad del algoritmo es tratar el difícil problema de la regresión no lineal en problemas de espacios con dimensiones grandes.

Para estados cuánticos mezcla, en la sección 3.2 se propone una medida de correlación para n -qubits que considera la dimensión del espacio de trabajo.

3.1. Entrelazamiento geométrico para estados puros

En el caso de estados puros, como se menciona en la sección 2.4, un estado separable² se puede descomponer perfectamente como un producto tensorial de n estados de un qubit.

Esta diferencia entre un estado separable y no separable (entrelazado), se puede visualizar en un ejemplo simple de estados de dos qubits. El estado $|\psi_S\rangle$ a continuación, es un estado separable, o

¹En este caso entrelazamiento.

²En esta tesis, a menos que se especifique las particiones, se utilizará los términos *separable* de n -qubits y *totalmente separable*, o *completamente separable*, indistintamente.

sea, puede ser escrito como producto de Kronecker de estados de un qubit como

$$|\psi_S\rangle = \frac{|00\rangle + |01\rangle}{\sqrt{2}} = |0\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (3.1)$$

Por otro lado el estado

$$|\psi_E\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.2)$$

es entrelazado, ya que es imposible de escribir como un producto de dos estados de un qubit. Siendo así es interesante medir qué tan correlacionado se encuentran estos estados, o sea, que tan cercanos, o no, están de ser separables.

3.1.1. Medida de entrelazamiento geométrico

La medida geométrica de entrelazamiento (GME) [29][126], se basa en qué tan alejado está un cierto estado $|\psi\rangle$ de algún estado totalmente separable. Se define explícitamente como la distancia del estado de n -qubits $|\psi\rangle$ al estado producto de Kronecker de n factores de un qubit, o sea, el estado totalmente separable más cercano $|\varphi_s\rangle$, o sea,

$$GME(|\psi\rangle) = \min_S D(|\psi\rangle - |\varphi_s\rangle), \quad (3.3)$$

$$\{|\varphi_s\rangle \in S : |\varphi_s\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle \cdots \otimes |\varphi_n\rangle\}, \quad (3.4)$$

siendo $D(\cdot)$ una distancia entre estados cuánticos.

En esta sección se utilizará la distancia inducida a partir de la *Fidelidad* entre estados cuánticos puros, definida como

$$F = \sqrt{\langle\varphi|\psi\rangle\langle\psi|\varphi\rangle} \quad (3.5)$$

siendo que la distancia está definida como [82] como

$$D_F(|\psi\rangle, |\varphi\rangle) = \arccos\left(\sqrt{\langle\varphi|\psi\rangle\langle\psi|\varphi\rangle}\right). \quad (3.6)$$

Definición 3.1 (Medida de entrelazamiento geométrico). Considerando la métrica definida por la ecuación (3.6) se define la medida de entrelazamiento geométrico (GME) como

$$GME(|\psi\rangle) = \left(\frac{4}{\pi}\right) \min_{|\varphi_s\rangle} D_F(|\psi\rangle, |\varphi_s\rangle). \quad (3.7)$$

El factor $4/\pi$ normaliza a uno el valor de la medida para el estado *Greenberger-Horne-Zeilinger* de n -qubits, $|GHZ\rangle_n$, definido como:

$$|GHZ\rangle_n = \frac{1}{\sqrt{N}} (|00 \dots 0\rangle + |11 \dots 1\rangle). \quad (3.8)$$

Este estado $|GHZ\rangle_n$ no es el de mayor entrelazamiento (geométrico) para los casos en los cuales $n > 2$. Su elección fue basada en que es un estado de fácil comparación entre distintas dimensiones.

Observación 3.1. Para la búsqueda del estado totalmente separable más cercano, y la minimización propuesta, se utiliza un algoritmo híbrido que consta de tres partes: un algoritmo algebraico para la inicialización de la población y un algoritmo genético que es acelerado por métodos de búsqueda local utilizando el gradiente. Este algoritmo se describe en el apéndice C.

3.1.2. Ejemplos

Ejemplo 1

En este primer ejemplo se analizará la evolución de la medida de entrelazamiento en estados de 2 qubits según la variación de un parámetro $\alpha \in [0, 1]$. Los estados a comparar son:

- El estado que representa la evolución continua entre el estado $|\psi\rangle = |00\rangle$ (caso $\alpha = 0$) y el estado EPR $\beta_{00} = \frac{\sqrt{2}}{2}(|00\rangle + |11\rangle)$ (caso $\alpha = 1$) según la ecuación (3.9). A modo de notación se denominará este estado $|GHZ_\alpha\rangle_2$.

$$q = \frac{1}{\sqrt{1+\alpha^2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \alpha \end{bmatrix} \quad (3.9)$$

- Estado resultante de propagar el estado original $|\psi_{in}\rangle = \frac{\sqrt{2}}{2}(|00\rangle + |10\rangle)$ por una compuerta CNOT de giro continuo (C_α , ecuación (3.10)) según ilustra la figura 3.1. En este caso para $\alpha = 0$ se tiene un sistema identidad ($|\psi_{out}\rangle = |\psi_{in}\rangle$) y cuando $\alpha = 1$ es el *CNOT* clásico.

$$C_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\left(\frac{\alpha\pi}{2}\right) & -i\sin\left(\frac{\alpha\pi}{2}\right) \\ 0 & 0 & -i\sin\left(\frac{\alpha\pi}{2}\right) & \cos\left(\frac{\alpha\pi}{2}\right) \end{bmatrix} \quad (3.10)$$

Figura 3.1: Compuerta para el sistema C_α .

La figura 3.2(a) ilustra las medidas de entrelazamiento según las variaciones de α en cada caso. La variación debida al sistema C_α es indicada mediante la línea de a trazos mientras la línea continua indica la variación de la medida de entrelazamiento del estado $|GHZ_\alpha\rangle_2$. Es interesante notar la variación lineal de la medida para el caso de un sistema C_α .

Ejemplo 2

En este ejemplo se analiza la evolución de la medida de entrelazamiento para el caso de un estado $|GHZ_\alpha\rangle_n$ de n qubits (ecuación 3.11). Se ha elegido como ejemplo un valor de $n = 5$ para utilizar el algoritmo. La figura 3.2(b) muestra la variación de la medida de entrelazamiento en función de α . Se observa que la curva es idéntica a la del ejemplo anterior para el caso de los estados de 2 qubits $|GHZ_\alpha\rangle_2$. En realidad, las curvas son idénticas para la variación de los estados $|GHZ_\alpha\rangle_n$ (n fijo).

$$|GHZ_\alpha\rangle_n = \frac{1}{\sqrt{1+\alpha^2}} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \alpha \end{bmatrix} \quad (3.11)$$

(a) GME para los estados del ejemplo 1. La línea continua corresponde al caso $|GHZ_\alpha\rangle_2$ mientras la línea de trazos corresponde al caso C_α .

(b) GME para los estados $|GHZ_\alpha\rangle_5$.

Figura 3.2: Ejemplos 1 y 2. Variación de la medida de entrelazamiento geométrico (GME).

3.1.3. Ejemplo 3

Para el tercer ejemplo se ha estudiado la variación de la medida de entrelazamiento de estados cuánticos de 3 qubits creados a partir de la interpolación de estados GHZ ($|GHZ\rangle$) y Werner ($|W\rangle$). Esta experiencia es análoga a la reportada en [126]. Considerando los estados

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle),$$

$$|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle) \text{ y}$$

$$|WN\rangle = \frac{1}{\sqrt{3}}(|110\rangle + |101\rangle + |011\rangle),$$

se crean los siguientes estados (estados puros) mediante una interpolación según un parámetro α ($\alpha \in [0, 1]$):

$$|\psi_1\rangle = \sqrt{\alpha}|W\rangle + \sqrt{1-\alpha}|WN\rangle \quad (3.12)$$

$$|\psi_2\rangle = \sqrt{\alpha}|W\rangle + \sqrt{1-\alpha}e^{i\phi}|GHZ\rangle \quad (3.13)$$

La figura 3.3 ilustra las variaciones de las medidas de entrelazamiento geométrico (GME) en función la variación del parámetro α para los estados $|\psi_1\rangle$ y $|\psi_2\rangle$.

Ejemplo 4

A similitud del ejemplo 1 se quiere calcular la influencia de los qubits de control en una compuerta del tipo *Toffoli* en la medida de entrelazamiento del estado resultante. Con esta finalidad se ha utilizado una implementación de esta compuerta propuesta por Margolus [14] (figura 3.4). Como se observa en [108] el símbolo de congruencia indica una equivalencia entre los sistemas a menos de una multiplicación por una matriz diagonal con factores de fase. Las compuertas *CNOT* clásicas son sustituidas por compuertas C_α y C_β . Para el ejemplo se utiliza como entrada al sistema el estado cuántico de 3 qubits $|\psi_{in}\rangle = \frac{\sqrt{2}}{2}(|100\rangle + |110\rangle)$.

La figura 3.6 ilustra la variación de la medida de entrelazamiento según las variaciones de los parámetros α y β .

(a) Variación de GME para el estado $|\psi_1\rangle$ en función de la variación de α .

(b) Variación de GME para el estado $|\psi_2\rangle$ en función de la variación de α .

Figura 3.3: Ejemplo 3. Variación de la medida de entrelazamiento geométrico (GME) para los estados $|\psi_1\rangle$ y $|\psi_2\rangle$ en función de la variación de α .

Figura 3.4: Implementación de una compuerta de Toffoli de 3 qubits ($T_{\alpha,\beta}$) con variación continua de los parámetros α y β (ambos en el intervalo $[0, 1]$).

Figura 3.5: Experiencia del ejemplo 4 con un sistema $T_{\alpha,\beta}$.

3.2. Medida de correlación acumulativa

3.2.1. Introducción

En esta sección se presenta una medida de correlación total para estados mezcla multiqubits. La medida es definida en forma recursiva, acumulando la correlación de los subespacios del sistema y simplificando el cálculo sin necesidad de la utilización de métodos de regresión. La medida es continua, aditiva y refleja la dimensión del espacio de Hilbert del estado cuántico.

3.2.2. Definición

A continuación se define un índice de correlación total (clásica y cuántica) para estados multiqubits.

Figura 3.6: Ejemplo 4. Variación de la medida de entrelazamiento geométrico (GME) de $|\psi_{out}\rangle$ según las variaciones de α y β .

Definición 3.2 (Medida de correlación cumulativa (CCM)). Dado un estado de n qubits ρ , la medida cumulativa de correlación (CCM) es definida como

$$C(\rho) = \min_{\{k\}} [2^{n-2} D(\rho, \rho_{A_k} \otimes \rho_{B_k}) + C(\rho_{A_k}) + C(\rho_{B_k})] \quad (3.14)$$

siendo $D(\cdot, \cdot)$ una distancia entre estados cuánticos, y k el índice del elemento (ρ_{A_k}, ρ_{B_k}) en el conjunto de todas las posibles biparticiones del estado.

Esta medida no calcula la correlación entre partes, sino que estima la correlación total existente en el sistema.

La medida calcula la suma de la distancia entre el estado ρ y el estado producto de sus matrices reducidas (ρ_{A_k} and ρ_{B_k}), ponderado por el factor dimensional (2^{n-2}), y acumulando la medida CCM de cada subparte, $C(\rho_{A_k})$ and $C(\rho_{B_k})$. La recursion se detiene para estados de un qubit ($C(\rho_{1q}) = 0$). El resultado es el mínimo entre todas las posibles particiones.

Observación 3.2. En la definición se requiere que la distancia $D(\cdot, \cdot)$ utilizada cumpla con las siguientes propiedades:

1. Es invariante. $D(U\rho U^\dagger, U\sigma U^\dagger) = D(\rho, \sigma)$, siendo U un operador unitario.
2. Es contractiva. $D(\varepsilon(\rho), \varepsilon(\sigma)) \leq D(\rho, \sigma)$, donde $\varepsilon(\sigma)$ son operaciones cuánticas locales ($\varepsilon = \varepsilon^1 \otimes \dots \otimes \varepsilon^n$).
3. Es monótona. $D(\rho_A, \sigma_A) \leq D(\rho_{AB}, \sigma_{AB})$.

Observación 3.3. En este trabajo la entropía relativa

$$D_{RE}(\rho, \sigma) \equiv S(\rho \parallel \sigma) = \text{tr } \rho (\log \rho - \log \sigma)$$

es utilizada como la distancia en la **Definición 3.2**. Aunque no es estrictamente una distancia, dado que no es simétrica [11], siempre será utilizada en el orden definido en (3.14). La entropía relativa satisface las propiedades mencionadas en la *Observación 3.2* [82]. Como en la medida propuesta $\sigma = \rho_{A_k} \otimes \rho_{B_k}$, en este caso la entropía relativa es igual a la información mútua [63].

3.2.3. Propiedades de CCM

En los últimos años se ha discutido algunas propiedades que deben tener las medidas de correlación para multiqubits [21][26][63]. A continuación se detallarán las propiedades de CCM (las demostraciones se encuentran en el apéndice B.1).

Propiedad 3.1. La correlación es mayor o igual a cero.

$$C(\rho) \geq 0.$$

Propiedad 3.2. Los estados producto de n qubits, $\rho = \rho^1 \otimes \cdots \otimes \rho^n$, no tienen correlación.

$$C(\rho^1 \otimes \cdots \otimes \rho^n) = 0.$$

Propiedad 3.3. La correlación es invariante respecto a transformaciones unitarias locales ($U = U^1 \otimes \cdots \otimes U^n$).

$$C(U\rho U^\dagger) = C(\rho).$$

Propiedad 3.4. La correlación no es afectada si un sistema ρ es aumentado por subsistemas no correlacionados localmente.

$$C(\rho \otimes \sigma) = C(\rho),$$

siendo $\sigma = \sigma^1 \otimes \cdots \otimes \sigma^K$.

Propiedad 3.5. Operaciones locales no aumentan la correlación.

$$C(\varepsilon(\rho)) \leq C(\rho),$$

siendo $\varepsilon = \varepsilon^1 \otimes \cdots \otimes \varepsilon^n$.

Es comunmente aceptado que una buena medida de correlación debe tener las propiedades anteriores [137]. Además de estas, *CCM* cumple con las propiedades a continuación.

Propiedad 3.6. *CCM* es aditiva.

$$C(\phi \otimes \varphi) = C(\phi) + C(\varphi).$$

Propiedad 3.7. *CCM* refleja la dimensión del espacio de Hilbert del estado ($\mathcal{H}^{2^n}$). Esto hecho se expresa mediante dos condiciones:

1. La correlación máxima es no decreciente con la dimensión.

$$\max_{\{\mathcal{H}^M\}} C(\rho^M) \leq \max_{\{\mathcal{H}^N\}} C(\rho^N),$$

donde M y N son las dimensiones de los estados, y $M \leq N$.

2. Para estados *GHZ* [57] la correlación crece con la dimensión.

$$C(\rho_{GHZ_M}) < C(\rho_{GHZ_N})$$

where $M \leq N$.

Además cabe resaltar que la medida *CCM* en una medida continua.

3.2.4. CCM para estados GHZ

Una ventaja de la *CCM* consiste en que para estados del tipo Greenberger–Horne–Zeilinger (GHZ) [57] el resultado puede ser calculado analíticamente, lo que posibilita la comparación entre sistemas de distintas dimensiones. En la base canónica un estado de n qubits GHZ se puede describir como

$$\begin{aligned} |GHZ\rangle_n &= \frac{\sqrt{2}}{2} (|0 \dots 0\rangle + |1 \dots 1\rangle) = \frac{\sqrt{2}}{2} (|0\rangle + |2^n - 1\rangle), \\ \rho_{GHZ_n} &= |GHZ\rangle\langle GHZ|_n. \end{aligned} \tag{3.15}$$

Para los estados GHZ todas las matrices reducidas, resultados de un bipartición, tienen la forma

$$\begin{aligned}\rho_{A_k} &= \frac{1}{2}(|0\rangle\langle 0| + |2^{n_{A_k}} - 1\rangle\langle 2^{n_{A_k}} - 1|), \\ \rho_{B_k} &= \frac{1}{2}(|0\rangle\langle 0| + |2^{n_{B_k}} - 1\rangle\langle 2^{n_{B_k}} - 1|),\end{aligned}\tag{3.16}$$

donde $n = n_{A_k} + n_{B_k}$. Así, el estado producto en la ecuación (3.14) resulta del tipo

$$\begin{aligned}\rho_{A_k} \otimes \rho_{B_k} &= \frac{1}{4}(|0\rangle\langle 0| + |2^n - 1\rangle\langle 2^n - 1| + \dots \\ &\quad |M\rangle\langle M| + |2^n - 1 - M\rangle\langle 2^n - 1 - M|),\end{aligned}\tag{3.17}$$

siendo $M \in \{1, \dots, 2^{n-1} - 1\}$. En esencia, todos los productos resultados de biparticiones representan el mismo estado (salvo un cambio de qubits). Entonces, todas las distancias entre ρ_{GHZ} y $\rho_{A_k} \otimes \rho_{B_k}$ son iguales. Si se considera $D(\cdot, \cdot)$ normalizada para este caso (independiente de la cantidad de qubits n)

$$\begin{aligned}D_{GHZ} &\equiv D(\rho_{GHZ}, \rho_{A_k} \otimes \rho_{B_k}) = 1, \\ \Rightarrow C(\rho_{GHZ}) &= 2^{n-2} D_{GHZ} + \min_{\{k\}} [C(\rho_{A_k}) + C(\rho_{B_k})] \\ &= 2^{n-2} + \min_{\{k\}} [C(\rho_{A_k}) + C(\rho_{B_k})].\end{aligned}\tag{3.18}$$

Todas las matrices reducidas ρ_{A_k} (y ρ_{B_k}) son de la forma (3.16). Entonces, independiente de la dimensión, las distancias (normalizadas como descrita anteriormente) entre ρ_{A_k} y el producto de Kronecker de cualquiera de sus biparticiones son todas iguales, y las denotaremos por d . En este caso se simplifica la minimización de la ecuación 3.14, donde solamente es necesario conocer la cantidad de qubits n del estado GHZ. Los valores de CCM para un estado genérico GHZ puede ser calculado por el siguiente algoritmo:

$$\begin{aligned}F(x, 2) &= x, \\ F(x, 3) &= 2x + d, \\ n \geq 4 \text{ (even)} : F(x, n) &= 2^{n-2}x + 2F(d, \frac{n}{2}), \\ n \geq 5 \text{ (odd)} : F(x, n) &= 2^{n-2}x + F(d, \frac{n-1}{2}) + F(d, \frac{n+1}{2}), \\ C(\rho_{GHZ_n}) &= F(1, n).\end{aligned}\tag{3.19}$$

Cuando la distancia utilizada es la entropía relativa (D_{RE}), la constante d es igual a $1/2$. La tabla 3.1 ilustra los valores de la medida para los primeros 9 estados GHZ.

Ejemplo 3.1. Para el de los estados puros de 4 qubits ($\mathcal{H}^{2^4}$)

$$\begin{aligned}\rho_1 = |\psi_1\rangle\langle\psi_1|, \quad |\psi_1\rangle &= |GHZ_2\rangle \otimes |GHZ_2\rangle \\ &= \frac{1}{2}(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle), \\ \rho_2 = |\psi_2\rangle\langle\psi_2|, \quad |\psi_2\rangle &= |GHZ_3\rangle \otimes |0\rangle \\ &= \frac{\sqrt{2}}{2}(|0000\rangle + |1110\rangle), \\ \rho_3 = |\psi_3\rangle\langle\psi_3|, \quad |\psi_3\rangle &= |GHZ_4\rangle = \frac{\sqrt{2}}{2}(|0000\rangle + |1111\rangle),\end{aligned}$$

Number of qubits	CCM	CCM with D_{RE}
2	1	1
3	$2 + d$	2.5
4	$4 + 2d$	5
5	$8 + 4d$	10
6	$16 + 6d$	19
7	$32 + 9d$	36.5
8	$64 + 12d$	70
9	$128 + 18d$	137
10	$256 + 24d$	268

Cuadro 3.1: Valores de CCM para estados GHZ.

se tiene como resultados $C(\rho_1) = 2$, $C(\rho_2) = 2,5$ y $C(\rho_3) = 5$. En los primeros dos estados la propiedad aditiva de la medida queda explícita. Este ejemplo muestra la influencia de subespacios muy correlacionados en estados cuánticos de dimensiones altas. Cuanto mayor sea la dimensión de un subespacio correlacionado, mayor es la medida de la correlación.

3.2.5. CCM y detección de transiciones de fase cuántica

En el análisis de la teoría física de sistemas de muchos cuerpos, la correlación tiene una importancia fundamental en las transiciones de fase cuántica (quantum phase transition, QPT). En sistemas con alta correlación, las fluctuaciones cuánticas son las responsables de las transiciones de fase [98]. En los últimos años se ha mostrado que es posible detectar puntos críticos en cadenas de espines correlacionados, modelados por Hamiltonianos de Heisenberg, mediante el comportamiento de algunas medidas de correlación [5][10][35][43][86][96][130].

La capacidad de CCM de indicar las correlaciones internas en un estado cuántico se muestra útil en diversos problemas. En esta sección se estudia su utilización en la detección de puntos críticos. Los ejemplos seleccionados, el modelo de Ising (transverse Ising model) y el modelo XXZ de Heisenberg, tienen la ventaja de poseer solución exacta en una dimensión, siendo sus puntos críticos conocidos [17]. A continuación se ilustra el desempeño de CCM para estos modelos.

El modelo genérico de Hamiltoniano de espín 1/2 [9] está determinado por

$$H = - \sum_{i=1}^N (J_x \sigma_i^x \sigma_{i+1}^x + J_y \sigma_i^y \sigma_{i+1}^y + J_z \sigma_i^z \sigma_{i+1}^z + h \sigma_i^z), \quad (3.20)$$

siendo N la cantidad de espines, $\sigma^{x,y,z}$ son las matrices de Pauli y satisface la condición de borde $\sigma_1 = \sigma_{N+1}$. En este trabajo solamente se analiza la interacción con los vecinos más cercanos (índices i e $i+1$).

Detección de puntos críticos en el modelo XXZ

Sea el hamiltoniano del modelo XXZ anisotrópico [101][133]

$$H_{XXZ} = -\frac{1}{2} \sum_{i=1}^N (\sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \Delta \sigma_i^z \sigma_{i+1}^z), \quad (3.21)$$

siendo $J_x = J_y = 1/2$, $h = 0$ y $J_z = \Delta/2$ (parámetro anisotrópico) los valores utilizados en la ecuación (3.20).

El modelo presenta tres fases para el estado fundamental:

- para $\Delta \rightarrow -\infty$ la cadena es completamente antiferromagnética,

- la segunda fase se encuentra para $-1 < \Delta < 1$,
- y para $\Delta \rightarrow \infty$ es ferromagnética.

Estas tres fases están separadas por dos puntos críticos: en $\Delta = -1$ existe una QPT de orden infinito mientras que para $\Delta = 1$ existe una QPT de primer orden.

La figura 3.7 muestra los resultados de CCM y de la generalización de la medida total de correlación definida en [115], $\mathcal{T}_V$, dada por

$$\mathcal{T}_V = S(\rho \| \rho_1 \otimes \rho_2 \dots \rho_N). \quad (3.22)$$

Como se puede observar, ambas medidas tienen una discontinuidad en $\Delta = 1$, indicando un punto crítico de primer orden.

Figura 3.7: Correlaciones CCM (azul, continua) y $\mathcal{T}_V$ (roja, a trazos) para el estado fundamental de una cadena XXZ de 6 espines. Ambas detectan el punto crítico en $\Delta = 1$.

La figura 3.8(a) ilustra el detalle de la curva de CCM para $-1.5 < \Delta < 1$. Mientras $\mathcal{T}_V$ el valor es constante, CCM presenta un máximo en $\Delta = -1$ indicando el punto crítico de orden infinito. Además, el valor del punto máximo crece con la cantidad de espines en la cadena (Figure 3.8(b)).

(a) Detalle de CCM para $-1.5 < \Delta < 1$. CCM detecta el punto crítico de orden infinito (máximo) en $\Delta = -1$.

(b) Detalle de CCM (punto crítico en $\Delta = -1$) para modelos con 4, 6 y 8 espines. Para visualizar las diferencias se escala la correlación en forma logarítmica.

Figura 3.8: Detalle de la curva de CCM en el entorno del punto crítico (máximo) para $\Delta = -1$.

QPT en cadenas dobles

A diferencia de las medidas genuinas [21][53], CCM puede indicar las transiciones de fase parciales en un sistema. Sea el hamiltoniano de una cadena doble dado por

$$H_{DXXZ} = -\frac{1}{2} \sum_{i=1}^N \left(\sigma_i^x \sigma_{i+1}^x + \tau_i^x \tau_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \tau_i^y \tau_{i+1}^y + \dots + \Delta \sigma_i^z \sigma_{i+1}^z + \lambda \tau_i^z \tau_{i+1}^z \right), \quad (3.23)$$

donde $\sigma^{x,y,z}$ y $\tau^{x,y,z}$ son las matrices de Pauli de las dos cadenas. Este hamiltoniano representa un modelo XXZ doble e independiente.

Los resultados en las figuras 3.9(a) y 3.9(b) muestran que CCM es capaz de indicar los cambios parciales de fase de ambos subsistemas.

(a) Indicación de los puntos críticos de primer orden en forma. Detección cuando ocurren en forma independiente y simultánea.

(b) Detalle de CCM para el punto crítico doble de orden infinito en $\Delta = -1$ y $\lambda = -1$

Figura 3.9: Curva de correlaciones para el modelo XXZ doble e independiente.

Dinámica de sistemas abiertos y QPT

En los últimos años se ha estudiado el comportamiento del estado fundamental de una cadena de spines considerando la decoherencia generada por la interacción con el ambiente [61][128][127]. Debido a que no pueden ser completamente aislados, los sistemas cuánticos reales tienen interacciones con el ambiente. Así, es de interés analizar el comportamiento de los estados fundamentales en la cercanía de puntos críticos en la presencia de ruido ocasionado por la decoherencia. Este efecto puede ser analizado mediante la utilización de operadores de Krauss (OSR, sección 4.2). En el presente análisis se considera la evolución de los estados fundamentales con una interacción disipativa con el ambiente representada mediante los operadores de amortiguación de amplitud (*amplitude damping*, sección 4.2.1),

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{p} \end{pmatrix}. \quad (3.24)$$

En este caso, el comportamiento de la CCM es similar al de la información mutua bipartita como se estudia en [87]. La figura 3.10(a) ilustra la evolución del estado fundamental cerca de un punto crítico de orden infinito ($-1.5 < \Delta < 1$), considerando un nivel de ruido bajo ($0 < p < 0.04$). Es interesante el hecho de que en el punto crítico ($\Delta = -1$) el máximo queda más pronunciado a medida que aumenta el nivel de ruido, como ilustrado en la figura 3.10(b).

Modelo transversal de Ising

El modelo de Ising tiene una clase diferente de puntos críticos en comparación con el modelo XXZ. El modelo de Ising [101] se obtiene utilizando $J_x = 1$, $J_y = J_z = 0$ y $h = \lambda$ en la ecuación

(a) Evolución del estado fundamental cerca del punto crítico de orden infinito.

(b) Curvas de contorno para diferentes valores de p .

Figura 3.10: Evolución de la CCM para el estado fundamental del modelo XXZ en presencia de ruido.

(3.20), resultando en un hamiltoniano del tipo

$$H = - \sum_{i=1}^N (\sigma_i^x \sigma_{i+1}^x + \lambda \sigma_i^z), \quad (3.25)$$

Para $\lambda = 0$ todos los espines apuntan en la dirección x , mientras cuando $\lambda \rightarrow \infty$ todos apuntan en la dirección z . El estado fundamental presenta un punto crítico cuando $\lambda = 1$, que puede ser observado en la figura 3.11 como un mínimo en la curva de la derivada de la CCM . Otras medidas de correlación han obtenido resultados similares [12][43][85][87].

Figura 3.11: Modelo transversal de Ising. Curva de la derivada de la CCM en el entorno del estado fundamental ($\lambda = 1$).

Capítulo 4

Errores en sistemas cuánticos

El análisis y modelado de errores es fundamental en la implementación de cualquier equipo o sistema. Es así en todas las áreas de la ingeniería: la civil, la electrónica, la química, la informática y otras. Esto permite conocer la incertidumbre de los resultados e investigar técnicas de reducción de errores, aumentando la eficiencia de los mismos.

En la computación cuántica esto también es cierto. Existen diversas fuentes de error en los algoritmos cuánticos [33][42]:

- errores resultantes de que la probabilidad de la medición de una respuesta correcta es menor que uno para muchos algoritmos cuánticos [34][44][82],
- errores debido a la decoherencia [25][102][138],
- errores sistemáticos e imperfecciones en los equipos de control, procesos de medición y aplicación de compuertas [38].

En los trabajos iniciales del análisis de errores se consideraba la hipótesis de que los errores en cada qubit son independientes, utilizando una analogía con el modelo clásico [106]. Aunque esta hipótesis es realista al aplicar compuertas cuánticas de un qubit (o medir solamente un qubit), no está justificada al aplicar compuertas de dos qubits o al analizar el efecto del ambiente (decoherencia) en los sistemas de n-qubits.

Además es común suponer que los errores promedio son independientes del número de qubits [90], surgiendo de forma natural un modelo estocástico. En este modelo se trabaja con la representación de suma de operadores (*Operator-Sum Representation, OSR*), utilizando operadores discretos (bit-flip, phase-flip, etc). Estos operadores permiten construir bases para el desarrollo de los operadores generales de error [73]. Este es uno de los modelos de error utilizado con mayor frecuencia, tanto en estudios teóricos como en simulaciones numéricas de algoritmos cuánticos [100].

Expuestas las limitaciones de los estudios anteriores, se considera que para el análisis de la eficiencia de los algoritmos cuánticos es necesario disponer de diversas formas de modelos de errores. En este capítulo son presentados los modelos que serán utilizados en las experiencias de propagación de errores.

4.1. Errores de decoherencia

Una forma natural de modelar la dinámica de un sistema cuántico abierto es mediante la suposición de una interacción del *sistema de interés*, con un estado cuántico ρ , con un sistema *ambiente*, con estado ρ_{amb} , formando ambos un sistema cuántico cerrado. De esta forma el *sistema completo*, $\hat{\rho} = \rho \otimes \rho_{amb}$, evoluciona mediante una transformación unitaria.

Observación 4.1. Mediante este modelo se asume explícitamente que el *sistema completo* puede ser descrito como un sistema producto entre el *sistema de interés* y el *ambiente*, hecho que puede no ser cierto. Sin embargo, para muchos casos prácticos es razonable esta hipótesis inicial [82].

Así, considerando U el operador unitario que modela la interacción, el estado resultante puede ser calculado como

$$\rho_f = \text{Tr}_{amb} (U \hat{\rho} U^\dagger) = \text{Tr}_{amb} (U \rho \otimes \rho_{amb} U^\dagger). \quad (4.1)$$

La figura 4.1 ilustra este modelo de interacción. Mediante el formalismo de operaciones cuánticas, en la sección a continuación, se presenta una forma general para analizar evoluciones de sistemas cuánticos.

Figura 4.1: Modelo de error de decoherencia.

4.2. Representación de suma de operadores

El formalismo de operaciones cuánticas es una herramienta general que permite analizar la evolución de los sistemas cuánticos en una gran variedad de condiciones [82] [94]. Tanto la evolución unitaria de los sistemas cerrados, modelada mediante operadores unitarios, como de los sistemas abiertos, procesos de medida e interacción con el ambiente, pueden ser descritos mediante este formalismo.

Definición 4.1 (Operación cuántica). Una operación cuántica es un mapeo definido como

$$\rho' = \varepsilon(\rho). \quad (4.2)$$

En particular, la propagación de un estado cuántico ρ mediante un circuito U (transformación unitaria), se describe por la operación $\varepsilon(\rho) = U \rho U^\dagger$.

Es posible representar estas operaciones mediante el formalismo denominado *Representación de Suma de Operadores* (Operator-sum representation). Así, cada operación cuántica puede ser calculada mediante un conjunto de operadores $\{E_k\}$ de acuerdo a la fórmula

$$\varepsilon(\rho) = \sum_k E_k \rho E_k^\dagger. \quad (4.3)$$

Las operaciones que conservan la traza, evoluciones mediante operadores unitarios p. ej., se describen con operadores que cumplen la condición

$$\sum_k E_k^\dagger E_k = 1. \quad (4.4)$$

Operaciones que no conservan la traza tienen operadores para los cuales

$$\sum_k E_k^\dagger E_k \leq 1. \quad (4.5)$$

El estado resultante de la operación puede ser calculado en forma genérica como

$$\rho' = \frac{\varepsilon(\rho)}{\text{tr} [\varepsilon(\rho)]}. \quad (4.6)$$

Los operadores más comumente utilizados para modelar errores en canales cuánticos se describen a continuación.

4.2.1. Operadores de error

Los operadores a continuación preservan la traza.

Canales bit flip, phase flip y Bit-phase flip

Estos operadores modelan cambios del bit lógico o la fase en un canal.

Definición 4.2 (Bit flip). El canal bit flip modela un cambio entre los estados $|0\rangle$ y $|1\rangle$ con una probabilidad $(1 - p)$. Los operadores utilizados son:

$$E_0 = \sqrt{p}I = \begin{bmatrix} \sqrt{p} & 0 \\ 0 & \sqrt{p} \end{bmatrix}, E_1 = \sqrt{1-p}X = \begin{bmatrix} 0 & \sqrt{1-p} \\ \sqrt{1-p} & 0 \end{bmatrix}. \quad (4.7)$$

Definición 4.3 (Phase flip). Este canal modela un cambio una fase relativa en un estado $|\psi\rangle$ con una probabilidad $(1 - p)$. Los operadores utilizados son:

$$E_0 = \sqrt{p}I = \begin{bmatrix} \sqrt{p} & 0 \\ 0 & \sqrt{p} \end{bmatrix}, E_1 = \sqrt{1-p}Z = \begin{bmatrix} \sqrt{1-p} & 0 \\ 0 & -\sqrt{1-p} \end{bmatrix}. \quad (4.8)$$

Definición 4.4 (Bit phase flip). Este canal tiene como operadores:

$$E_0 = \sqrt{p}I = \begin{bmatrix} \sqrt{p} & 0 \\ 0 & \sqrt{p} \end{bmatrix}, E_1 = \sqrt{1-p}Y = \begin{bmatrix} 0 & -\sqrt{1-p}i \\ \sqrt{1-p}i & 0 \end{bmatrix}. \quad (4.9)$$

Las figuras en la ilustración 4.2 muestran el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante estos operadores. Se observa que con excepción de los estados pertenecientes a los ejes que no son afectados, el resultado para cada estado de la esfera es un estado mezcla (norma menor que uno). La última figura, 4.2(d), muestra el resultado de la aplicación de una secuencia de todos los operadores. El efecto final es el mismo del operador descrito a continuación.

Depolarizing channel

En este modelo de error el canal es despolarizado, reemplazado por el estado máximamente mezclado $I/2$, con una probabilidad p ,

$$\varepsilon(\rho) = p\frac{I}{2} + (1-p)\rho. \quad (4.10)$$

Definición 4.5 (Canal de despolarización). El canal de despolarización es definido por los operadores

$$\begin{aligned} E_0 &= \sqrt{1 - \frac{3p}{4}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & E_1 &= \sqrt{\frac{p}{4}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ E_2 &= \sqrt{\frac{p}{4}} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, & E_3 &= \sqrt{\frac{p}{4}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned} \quad (4.11)$$

Las figuras 4.3(a) y 4.3(b) ilustran el efecto de la operación del canal de despolarización para dos valores de p .

Amplitude damping y phase damping

Definición 4.6 (Amplitude damping). Es el conjunto de operadores de amortiguación de amplitud modela la disipación de energía en un sistema cuántico. Como hipótesis para la determinación de los operadores se considera que el ambiente empieza en el estado $|0\rangle$ (a temperatura cero). Los operadores son:

(a) Bit flip ($p = 0,3$). Estados se contraen con un factor $1 - 2p$ en el plano Y-Z y se mantienen en el eje X.

(b) Phase flip ($p = 0,3$). Estados se contraen con un factor $1 - 2p$ en el plano X-Y y se mantienen en el eje Z.

(c) Bit-phase flip ($p = 0,3$). Estados se contraen con un factor $1 - 2p$ en el plano X-Z y se mantienen en el eje Y.

(d) Operación en secuencia de los tres operadores ($p = 0,1$).

Figura 4.2: Efecto de la aplicación de los operadores bit flip, phase flip, bit-phase flip y la secuencia de los tres sobre los estados puros en la esfera de Bloch. Para los tres primeros casos la esfera se contrae con un factor $1 - 2p$ sobre un plano, manteniendo los estados intactos en el eje perpendicular al plano de contracción.

(a) Depolarizing channel con $p = 0,25$.

(b) Depolarizing channel con $p = 0,75$.

Figura 4.3: Efecto de la aplicación de los operadores del canal de despolarización sobre los estados puros en la esfera de Bloch.

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}. \quad (4.12)$$

En el caso de modelar el efecto de disipación a un ambiente con una temperatura finita se utiliza el conjunto de operadores conocidos como de amortiguación de amplitud generalizada (generalized amplitude damping).

$$\begin{aligned} E_0 &= \sqrt{p} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, E_1 = \sqrt{p} \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix} \\ E_2 &= \sqrt{1-p} \begin{bmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{bmatrix}, E_3 = \sqrt{1-p} \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{bmatrix}. \end{aligned} \quad (4.13)$$

En este caso el estado estacionario $\epsilon_{GAD}(\rho_\infty) = \rho_\infty$ es

$$\rho_\infty = \begin{bmatrix} p & 0 \\ 0 & 1-p \end{bmatrix}. \quad (4.14)$$

Definición 4.7 (Phase damping). Modela la pérdida de información cuántica sin disipación de energía. El conjunto de operadores es equivalente (en realidad, el mismo) al de phase flip.

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\gamma} \end{bmatrix}. \quad (4.15)$$

(a) Amplitude damping con $p = 0,5$.

(b) Phase damping con $p = 0,5$.

Figura 4.4: Efecto de la aplicación de los operadores de amplitud damping y phase damping sobre los estados puros en la esfera de Bloch.

4.3. Errores continuos

4.3.1. Modelos de errores cuánticos continuos de 1 qubit

Considerando que $U = e^A$, donde A es una matriz antihermítica, es siempre una matriz unitaria, y que la suma de matrices antihermíticas es antihermítica, es posible definir el operador unitario

$$E = e^{i(\alpha_I A_I + \alpha_X A_X + \alpha_Y A_Y + \alpha_Z A_Z)}. \quad (4.16)$$

Este operador permite controlar la posible rotación de un estado cuántico mediante los coeficientes α_I , α_X , α_Y y α_Z .

Otra forma de generar un operador de error continuo de 1 qubit sería considerar el operador

$$E = e^{i(\alpha_X A_X)} e^{i(\alpha_Y A_Y)} e^{i(\alpha_Z A_Z)}, \quad (4.17)$$

utilizando el hecho de que el resultado del producto de operadores unitarios es un operador unitario. Así se tiene el operador

 (a) Generalized amplitude damping con $p = 0,5$ y $\gamma = 0,75$.

 (b) Generalized amplitude damping con $p = 0,5$ y $\gamma = 0,5$.

Figura 4.5: Efecto de la aplicación de los operadores de generalized amplitude damping sobre los estados puros en la esfera de Bloch.

$$E = X_{\alpha_X} Y_{\alpha_Y} Z_{\alpha_Z} \quad (4.18)$$

con

$$X_{\alpha_X} = e^{i(\alpha_X A_X)} = \frac{1}{2} \begin{pmatrix} 1 + e^{i\alpha_X \pi} & 1 - e^{i\alpha_X \pi} \\ 1 - e^{i\alpha_X \pi} & 1 + e^{i\alpha_X \pi} \end{pmatrix} \quad (4.19)$$

$$Y_{\alpha_Y} = e^{i(\alpha_Y A_Y)} = \frac{1}{2} \begin{pmatrix} 1 + e^{i\alpha_Y \pi} & i(e^{i\alpha_Y \pi} - 1) \\ i(1 - e^{i\alpha_Y \pi}) & 1 + e^{i\alpha_Y \pi} \end{pmatrix} \quad (4.20)$$

$$Z_{\alpha_Z} = e^{i(\alpha_Z A_Z)} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha_Z \pi} \end{pmatrix} \quad (4.21)$$

Observación 4.2. Se puede conseguir una rotación análoga a la del operador Z_θ sobre el eje Z utilizando el operador $e^{i\theta Z}$, siendo Z la matriz de Pauli definida anteriormente. La diferencia entre el resultado de aplicar estos operadores es en una fase global en el estado cuántico. Operadores de rotación similares pueden ser definidos mediante las matrices $e^{i\theta X}$ y $e^{i\theta Y}$.

Ejemplo 4.1. En la figura 4.6 vemos el resultado de aplicar los operadores de rotación $E = X_{\alpha_X}$, $\alpha_X \in [-1, 1]$, y $E = Y_{\alpha_Y}$, $\alpha_Y \in [-1, 1]$, al estado inicial $q = |0\rangle$.

Ejemplo 4.2. Se considera ahora el estado inicial $q = \frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$. Si se aplica los operadores de rotación $E = Y_{\alpha_Y}$, $\alpha_Y \in [-1, 1]$, y $E = Z_{\alpha_Z}$, $\alpha_Z \in [-1, 1]$, se tiene como resultado lo indicado en la figura 4.7.

Ejemplo 4.3. Estado inicial $q = \frac{\sqrt{5}}{5}(2|0\rangle + i|1\rangle)$. Se aplica, en forma independiente, los operadores de rotación $E = X_{\alpha_X}$, $E = Y_{\alpha_Y}$ y $E = Z_{\alpha_Z}$. La figura 4.8 ilustra las tres rotaciones resultantes.

Ejemplo 4.4. En este ejemplo se ilustra la aplicación del operador $E = X_{\alpha_X} Y_{\alpha_Y} Z_{\alpha_Z}$. Las figuras 4.9 y 4.10 ilustran los errores continuos alrededor de los estados $q = |0\rangle$ y $q = \frac{\sqrt{5}}{5}(2|0\rangle + i|1\rangle)$, respectivamente. En ambos caso se utilizó $\alpha_X, \alpha_Y, \alpha_Z \in [-0,05; 0,05]$.

4.3.2. Modelos de errores continuos en n qubits

Se puede generalizar la idea de generación de errores continuos de 1 qubit de la ecuación (4.16) para n qubits. En el caso de 1 qubit era posible construir un operador de error de la forma

$$E = e^{i(\alpha_I A_I + \alpha_X A_X + \alpha_Y A_Y + \alpha_Z A_Z)}.$$

controlando el margen de error mediante un parámetro p . En este caso se tiene que

(a) Variaciones con modelo de error $E = X_{\alpha_X}$

(b) Variaciones con modelo de error $E = Y_{\alpha_Y}$

Figura 4.6: Error continuo con estado original $q = |0\rangle$.

(a) Variaciones con modelo de error $E = Y_{\alpha_Y}$

(b) Variaciones con modelo de error $E = Z_{\alpha_Z}$

Figura 4.7: Error continuo con estado original $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$.

- $\alpha_I = p \times \text{rand}()$, en donde $\text{rand}()$ es una variable aleatoria en el intervalo $[0, 1]$.
- α_X , α_Y y α_Z son generadas de la misma forma: $\alpha_{OTRAS} = (1 - p) \times \text{rand}()$.

Ejemplo 4.5. Las figuras a continuación ilustran la generación de errores con diferentes radios promedios controlados por el parámetro p .

Para 2 qubits es posible generar un operador de error mediante

$$\begin{aligned}
 M &= \alpha_{II}(I \otimes I) + \alpha_{IX}(I \otimes X) + \alpha_{IY}(I \otimes Y) + \alpha_{IZ}(I \otimes Z) + \dots \\
 &\quad \alpha_{XI}(X \otimes I) + \alpha_{XX}(X \otimes X) + \alpha_{XY}(X \otimes Y) + \alpha_{XZ}(X \otimes Z) + \dots \\
 &\quad \alpha_{YI}(Y \otimes I) + \alpha_{YX}(Y \otimes X) + \alpha_{YY}(Y \otimes Y) + \alpha_{YZ}(Y \otimes Z) + \dots \\
 &\quad \alpha_{ZI}(Z \otimes I) + \alpha_{ZX}(Z \otimes X) + \alpha_{ZY}(Z \otimes Y) + \alpha_{ZZ}(Z \otimes Z) \Rightarrow \\
 E &= e^{iM}, \begin{cases} \alpha_{II} = p \times \text{rand}(); \\ \alpha_{OTROS} = (1 - p) \times \text{rand}(); \end{cases} \quad (4.22)
 \end{aligned}$$

Para el caso de n qubits se generaliza la idea anterior.

(a) Variaciones con modelo de error $E = X_{\alpha_X}$

(b) Variaciones con modelo de error $E = Y_{\alpha_Y}$

(c) Variaciones con modelo de error $E = Z_{\alpha_Z}$

Figura 4.8: Error continuo con estado original $\frac{\sqrt{5}}{5}(2|0\rangle + i|1\rangle)$.

(a)

(b)

Figura 4.9: Error continuo con estado original $q = |0\rangle$. Variaciones con modelo de error $E = X_{\alpha_X} Y_{\alpha_Y} Z_{\alpha_Z}$ ($\alpha_X, \alpha_Y, \alpha_Z \in [-0,05; 0,05]$).

4.4. Errores isótropos

En [49] y [48] se propone un modelo de error isótropo temporalmente independiente, en el cual el estado resultante depende solamente de la distancia del estado original.

Figura 4.10: Error continuo con estado original $q = \frac{\sqrt{5}}{5}(2|0\rangle + i|1\rangle)$. Variaciones con modelo de error $E = X_{\alpha_X} Y_{\alpha_Y} Z_{\alpha_Z} (\alpha_X, \alpha_Y, \alpha_Z \in [-0,05; 0,05])$.

Figura 4.11: Error continuo condiderando un parámetro $p = 0,95$.

Figura 4.12: Error continuo condiderando un parámetro $p = 0,99$.

Definición 4.8 (Modelo de error isótropo). Sea $|\psi\rangle$ un estado cuántico puro de n -qubits (denominado estado de referencia). Se denomina proceso de error isótropo temporalmente independiente con respecto a $|\psi\rangle$ a todo proceso estocástico $f(|\psi\rangle, 0, t)$, $t \geq 0$, que cumpla con las siguientes propiedades:

- En $t = 0$, la distribución de probabilidad cumple que $f(|\psi\rangle, 0, 0) = \delta(|\psi\rangle)$. O sea, para $t = 0$ no hay error y el estado está en $|\psi\rangle$ con probabilidad 1.
- La distribución de probabilidad en el instante t_k , $f(|\psi\rangle, 0, t_k)$, es isótropa respecto de $|\psi\rangle$. Esto significa que la probabilidad de error es igual en todas las direcciones. Considerando $|\psi(t_k)\rangle$ la variable aleatoria en el instante t_k , la probabilidad de que sea un estado determinado $|\phi\rangle$ depende solamente de la distancia entre $|\psi\rangle$ y $|\phi\rangle$ (considerando distancias definidas para estados cuánticos [82]),

$$\mathcal{P}(|\psi(t_k)\rangle = |\phi\rangle) = F(\mathcal{D}(|\psi\rangle, |\phi\rangle)). \quad (4.23)$$

- Si $t_k < t_{k+1}$, $f(|\psi\rangle, 0, t_k)$ y $f(|\psi\rangle, t_k, t_{k+1})$ son independientes.

A continuación se utilizará el estado $|0\rangle$ como estado de referencia. Para obtener la formulación de una distribución isótropa en el entorno de un estado cualquiera $|\psi\rangle$ solo es necesario realizar una rotación en el espacio de n -qubits, de forma a llevar el vector $|0\rangle$ al $|\psi\rangle$.

4.4.1. Errores isótropos para estados cuánticos con notación vectorial

Definición 4.9. Un estado $\psi(k)$ que representa un error isótropo con relación al estado de referencia $|0\rangle$ es un vector aleatorio del tipo

$$|\psi(k)\rangle = a_0(k)|0\rangle + a_1(k)e^{i\varphi_1(k)}|1\rangle + \dots + a_{2^n-1}(k)e^{i\varphi_{2^n-1}(k)}|2^n-1\rangle, \quad (4.24)$$

donde

- $|i\rangle$, $0 \leq i \leq 2^n - 1$, son los vectores de la base canónica (notación decimal),
- $\varphi_i(k)$ son variables aleatorias con distribución uniforme en el intervalo $[0, 2\pi)$ (la distancia no depende de las fases relativas) y
- $a_i(k)$, son variables aleatorias en el intervalo $[0, 1]$ que cumplen
 - $\forall k, \sum_{i=0}^{2^n-1} a_i^2(k) = 1$,
 - $a_i(k)$, $1 \leq i \leq 2^n - 1$, poseen la misma distribución de probabilidad (igual distancia a todos los vectores de la base diferentes de $|0\rangle$).

En este caso la distancia $\mathcal{D}(|0\rangle, |\psi(t_k)\rangle)$, como también la distribución de probabilidad, dependen solamente de $a_0(k)$.

Una forma práctica de crear ensambles de estados que representen un error isótropo en el entorno del estado de referencia $|0\rangle$, es mediante el uso de variables aleatorias utilizando coordenadas esféricas (esferas multidimensionales).

Observación 4.3 (Modelo de error isótropo con fase global utilizando coordenadas esféricas). Es un modelo con $2^{n+1} - 1$ parámetros (ya considera la norma unitaria).

$$\begin{aligned} |\psi\rangle &= (\cos(\theta_0) + i \sin(\theta_0) \cos(\theta_1)) |0\rangle \\ &\quad (\sin(\theta_0) \sin(\theta_1) \cos(\theta_2) + \\ &\quad i \sin(\theta_0) \sin(\theta_1) \sin(\theta_2) \cos(\theta_3)) |1\rangle \\ &\quad \vdots \\ &\quad (\sin(\theta_0) \sin(\theta_1) \dots \sin(\theta_{2^{n+1}-5}) \cos(\theta_{2^{n+1}-4}) + \\ &\quad i \sin(\theta_0) \sin(\theta_1) \dots \sin(\theta_{2^{n+1}-4}) \cos(\theta_{2^{n+1}-3})) |2^n - 2\rangle \\ &\quad (\sin(\theta_0) \sin(\theta_1) \dots \sin(\theta_{2^{n+1}-3}) \cos(\theta_{2^{n+1}-2}) + \\ &\quad i \sin(\theta_0) \sin(\theta_1) \dots \sin(\theta_{2^{n+1}-3}) \sin(\theta_{2^{n+1}-2})) |2^n - 1\rangle \end{aligned} \quad (4.25)$$

en donde θ_0 es una variable aleatoria en el intervalo $[0, \theta_{MAX}]$ y los demás ángulos son variables aleatorias en el intervalo $[0, \pi]$.

Ejemplo 4.6. En el caso de un estado cuántico de 1 qubit se tiene

$$q = \begin{pmatrix} (\cos(\theta_0) + i\sin(\theta_0)\cos(\theta_1)) |0\rangle \\ (\sin(\theta_0)\sin(\theta_1)\cos(\theta_2) + i\sin(\theta_0)\sin(\theta_1)\sin(\theta_2)) |1\rangle \end{pmatrix}$$

Lo anterior es equivalente a aplicar el operador unitario

$$U = \begin{pmatrix} X_1 + iX_2 & -X_3 + iX_4 \\ X_3 + iX_4 & X_1 - iX_2 \end{pmatrix}, \quad \begin{cases} X_1 = \cos(\theta_0) \\ X_2 = \sin(\theta_0)\cos(\theta_1) \\ X_3 = \sin(\theta_0)\sin(\theta_1)\cos(\theta_2) \\ X_4 = \sin(\theta_0)\sin(\theta_1)\sin(\theta_2) \end{cases}$$

Definición 4.10 (Modelo de error isótropo sin fase global utilizando coordenadas esféricas). Es un modelo con $2^{n+1} - 2$ parámetros.

$$\begin{aligned} |\psi\rangle &= \cos(\theta_0)|0\rangle \\ &\quad \sin(\theta_0)\sin(\theta_1)e^{i\varphi_1}|1\rangle \\ &\quad \vdots \\ &\quad \sin(\theta_0)\dots\cos(\theta_{2^n-2})e^{i\varphi_{2^n-2}}|2^n-2\rangle \\ &\quad \sin(\theta_0)\dots\sin(\theta_{2^n-2})e^{i\varphi_{2^n-1}}|2^n-1\rangle \end{aligned} \tag{4.26}$$

En este caso se tiene que:

- $\theta_i \in [0, \frac{\pi}{2}]$, $0 \leq i \leq 2^n - 2$.
- $\varphi_j \in [0, 2\pi)$, $1 \leq j \leq 2^n - 1$.

Ejemplo 4.7. A continuación se ilustran algunos ejemplos de generación ensambles de errores con distribución isótropa.

Figura 4.13: Error isótropo. 500 puntos alrededor del estado $|0\rangle$.

4.4.2. Errores isótropos para estados cuánticos con representación de matriz de densidad

La representación de matriz de densidad de un estado mezcla, equivalente a un ensamble de estados cuánticos, es determinada por

$$\rho = \sum_{k=1}^N p(k) |\psi(k)\rangle \langle \psi(k)| = E(|\psi(k)\rangle \langle \psi(k)|) = E(\rho_{\psi(k)}) \tag{4.27}$$

(a) Error isótopo (500 puntos) alrededor del estado $\frac{\sqrt{5}}{5}(2|0\rangle + i|1\rangle)$.
 (b) Error isótopo (5000 puntos) alrededor del estado $|0\rangle$.

Figura 4.14: Ejemplos de errores isótopos.

siendo $E(\rho_{\psi(k)})$ el valor esperado de la variable aleatoria $\rho_{\psi(k)}$.

En el caso de sistemas cuánticos de n -qubits los errores isótopos generados a partir del estado inicial $|0\rangle$ como se indica en (4.24), tienen una matriz de densidad de cada suceso $|\psi(k)\rangle$ dada por

$$\rho_{\psi(k)} = \begin{bmatrix} a_0(k)^2 & a_{01}(k) & \cdots & a_{0(2^n-1)}(k) \\ a_{10}(k) & a_1(k)^2 & \cdots & a_{1(2^n-1)}(k) \\ \vdots & \vdots & \ddots & \vdots \\ a_{(2^n-1)0}(k) & a_{(2^n-1)1}(k) & \cdots & a_{2^n-1}(k)^2 \end{bmatrix}, \quad (4.28)$$

en donde, $\forall 1 \leq i, j, \leq 2^n - 1$

- $a_{0i}(k) = a_0(k)a_i(k)e^{-i\varphi_i(k)}$,
- $a_{i0}(k) = a_i(k)a_0(k)e^{i\varphi_i(k)}$,
- y $a_{ij}(k) = a_i(k)a_j(k)e^{i(\varphi_i(k)-\varphi_j(k))}$.

Considerando las hipótesis indicadas en la definición 4.9 en la sección 4.4.1 se tiene que

- $E(e^{i\varphi_i(k)}) = 0$,
- $E(a_i(k)a_j(k)e^{i(\varphi_i(k)-\varphi_j(k))}) = E(a_i(k)a_j(k)) E(e^{i\varphi_i(k)}) E(e^{-i\varphi_j(k)}) = 0$
- $E(a_0(k)) = \lambda_0$,
- $E(a_i(k)) = \lambda_1, \forall 1 \leq i \leq 2^n - 1$,
- $(2^n - 1)\lambda_1 + \lambda_0 = 1$.

En este caso la matriz de densidad $\rho_\psi = E(\rho_{\psi(k)})$ del ensamble de estados de errores tiene la expresión

$$\rho_\psi = \begin{bmatrix} \lambda_0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_1 \end{bmatrix}. \quad (4.29)$$

Ejemplo 4.8. Sea un ensemble de estados cuánticos de 1 qubit generados con una función de densidad de probabilidad uniforme $f(\theta, \varphi)$ en el entorno del estado $|0\rangle$

$$\begin{aligned} |\psi(\theta, \varphi)\rangle &= \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\varphi}|1\rangle \\ \Rightarrow \rho_{\psi(\theta, \varphi)} &= \begin{bmatrix} \cos^2(\theta/2) & \cos(\theta/2)\sin(\theta/2)e^{-i\varphi} \\ \cos(\theta/2)\sin(\theta/2)e^{i\varphi} & \sin^2(\theta/2) \end{bmatrix} \\ f(\theta, \varphi) &= \begin{cases} \frac{1}{2\pi(1-\cos(\theta_{MAX}))}, & 0 \leq \theta \leq \theta_{MAX} \\ 0, & \text{otro caso} \end{cases} \end{aligned}$$

$$\begin{aligned} E(\cos^2(\theta/2)) &= \int_0^{\theta_{MAX}} \int_0^{2\pi} \frac{\cos^2(\theta/2)}{2\pi(1-\cos(\theta_{MAX}))} \sin(\theta) d\theta d\varphi \\ &= \frac{5 - \cos(2\theta_{MAX}) - 4\cos(\theta_{MAX})}{8(1-\cos(\theta_{MAX}))}, \\ E(\sin^2(\theta/2)) &= \int_0^{\theta_{MAX}} \int_0^{2\pi} \frac{\sin^2(\theta/2)}{2\pi(1-\cos(\theta_{MAX}))} \sin(\theta) d\theta d\varphi \\ &= \frac{3 + \cos(2\theta_{MAX}) - 4\cos(\theta_{MAX})}{8(1-\cos(\theta_{MAX}))}, \\ E(\cos(\theta/2)\sin(\theta/2)e^{i\varphi}) &= 0, \\ E(\cos(\theta/2)\sin(\theta/2)e^{-i\varphi}) &= 0. \end{aligned}$$

$$\Rightarrow \rho_{\psi} = \begin{bmatrix} \frac{5-\cos(2\theta_{MAX})-4\cos(\theta_{MAX})}{8(1-\cos(\theta_{MAX}))} & 0 \\ 0 & \frac{3+\cos(2\theta_{MAX})-4\cos(\theta_{MAX})}{8(1-\cos(\theta_{MAX}))} \end{bmatrix}$$

En el caso de $\theta_{MAX} = \pi$ y $\theta_{MAX} = \pi/2$ tenemos los estados mezcla

$$\rho_{\pi} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad \rho_{\frac{\pi}{2}} = \begin{bmatrix} \frac{3}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix}. \quad (4.30)$$

La figura 4.15 ilustra los ensambles de estados para los casos del ejemplo.

Figura 4.15: Error isotrópico uniforme en el entorno del estado $|0\rangle$.

Ejemplo 4.9. Sea un estado mezcla ρ_α compuesto por un ensamble de dos estados cuánticos igualmente probables $|\psi_1\rangle = \alpha|0\rangle + \sqrt{1-\alpha^2}e^{i\varphi}|1\rangle$ y $|\psi_2\rangle = \alpha|0\rangle + \sqrt{1-\alpha^2}e^{i(\varphi+\pi)}|1\rangle$.

$$\begin{aligned}\rho_\alpha &= 0,5\rho_1 + 0,5\rho_2 \\ &= 0,5 \begin{bmatrix} \alpha^2 & \alpha\sqrt{1-\alpha^2}e^{-i\varphi} \\ \alpha\sqrt{1-\alpha^2}e^{i\varphi} & 1-\alpha^2 \end{bmatrix} + 0,5 \begin{bmatrix} \alpha^2 & \alpha\sqrt{1-\alpha^2}e^{-i(\varphi+\pi)} \\ \alpha\sqrt{1-\alpha^2}e^{i(\varphi+\pi)} & 1-\alpha^2 \end{bmatrix} \\ &= \begin{bmatrix} \alpha^2 & 0 \\ 0 & 1-\alpha^2 \end{bmatrix}\end{aligned}\quad (4.31)$$

En el caso de $\alpha = \sqrt{2}/2$ y $\alpha = \sqrt{3}/2$ las matrices de densidad son idénticas a las matrices ρ_π y $\rho_{\pi/2}$ del ejemplo 4.8, respectivamente.

Figura 4.16: Error representado por el ensamble de dos estados cuánticos igualmente probables en el entorno del estado inicial $|0\rangle$. Las matrices de densidad son las mismas del ejemplo 4.8.

Cuando dos ensambles de estados cuánticos poseen la misma matriz de densidad, son indistinguibles, y por lo tanto representan el mismo estado cuántico mezcla. Siendo así, el estado ρ_π del ejemplo (4.8) y el estado $\rho_{\sqrt{2}/2}$ del ejemplo 4.9 son el mismo estado cuántico. Luego es necesario una definición alternativa para un estado proveniente de un error isótropo.

Definición 4.11 (Estado cuántico mezcla proveniente de error isótropo). Sea $|\psi\rangle$ un estado puro de referencia y M la matriz de cambio de base tal que $|0\rangle = M|\psi\rangle$. Un estado mezcla ρ representa un error isótropo, con respecto a $|\psi\rangle$, si el estado $\hat{\rho} = M\rho M^\dagger$ tiene la forma (ecuación (4.29))

$$\begin{aligned}\hat{\rho} &= \begin{bmatrix} \lambda_0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_1 \end{bmatrix}, \\ &= \begin{cases} (2^n \lambda_1) \frac{I}{2^n} + (\lambda_0 - \lambda_1) \rho_0, & \lambda_0 \geq \lambda_1, \\ (2^n \lambda_0) \frac{I}{2^n} + (2^n - 1) (\lambda_1 - \lambda_0) \rho_{N0}, & \lambda_0 < \lambda_1 \end{cases}\end{aligned}\quad (4.32)$$

siendo $\rho_0 = |0\rangle\langle 0|$ y

$$\rho_{N0} = \frac{(I - \rho_0)}{(2^n - 1)} = \frac{1}{(2^n - 1)} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}\quad (4.33)$$

un estado mezcla que se le denomina *mezcla isotropa ortogonal del estado* ρ_0 (con el producto interno usual de matrices).

Aplicando la transformación inversa, y considerando el estado de referencia $|\psi\rangle$, un estado cuántico proveniente de un error isotrópico tiene una de las dos formas

$$\begin{cases} (2^n \lambda_1) \frac{I}{2^n} + (\lambda_0 - \lambda_1) \rho_\psi, & \lambda_0 \geq \lambda_1, \\ (2^n \lambda_0) \frac{I}{2^n} + (2^n - 1) (\lambda_1 - \lambda_0) \rho_{N\psi}, & \lambda_0 < \lambda_1, \end{cases} \quad (4.34)$$

siendo $\rho_{N\psi}$ el estado *mezcla isotropa ortogonal del estado* ρ_ψ

$$\rho_\psi = \frac{(I - \rho_\psi)}{(2^n - 1)}. \quad (4.35)$$

Capítulo 5

Índices de isotropía para errores cuánticos

Como se ha visto en el capítulo 4, en los sistemas cuánticos existen errores que afectan la información.

Siendo así es importante disponer de herramientas que posibilitan caracterizar el efecto final del acúmulo y propagación de errores, cuantificar la deformación ocasionada en los resultados y estudiar las limitaciones existentes.

Con esta finalidad, en las secciones a continuación se definen índices que caracterizan la isotropía de los estados cuánticos. Además se muestra que la componente isótropa del estado, resultante de errores isótropos, indica la pérdida de información en el circuito cuántico.

5.1. Índice de isotropía geométrico

Si se considera puro el estado original, es posible estudiar la propagación del error ocasionada por la aplicación de puertas unitarias con error de implementación: el resultado final es un conjunto de estados puros generados aleatoriamente mediante el modelo de error unitario utilizado en la caracterización de las compuertas.

Definición 5.1. Considere que el estado de referencia de 1-qubit $|0\rangle$ es afectado por el ruido en forma aleatoria caracterizada por una función de distribución de probabilidades $f(\varphi, \theta)$ (figura 5.1).

Figura 5.1: Ensamble de estados con función de distribución de probabilidad de error $f(\varphi, \theta)$ en la esfera de Bloch.

El índice de isotropía geométrico es definido por el par $i_{iso} = (i_\varphi, i_\theta)$, siendo¹

$$\begin{aligned}
 k_\varphi(\varphi) &= \lim_{\Delta\varphi \rightarrow 0} \frac{2\pi}{\Delta\varphi} \int_{\varphi-\frac{\Delta\varphi}{2}}^{\varphi+\frac{\Delta\varphi}{2}} \int_0^\pi f(\varphi, \theta) \sin(\theta) d\theta d\varphi, \\
 k_{\varphi_{max}} &= \max_{\varphi} k_\varphi(\varphi), \quad k_{\varphi_{av}} = \frac{1}{2\pi} \int_0^{2\pi} k_\varphi(\varphi) d\varphi, \\
 k_\theta(\varphi) &= \lim_{\Delta\varphi \rightarrow 0} \frac{2\pi}{\Delta\varphi} \int_{\varphi-\frac{\Delta\varphi}{2}}^{\varphi+\frac{\Delta\varphi}{2}} \int_0^\pi \theta f(\varphi, \theta) \sin(\theta) d\theta d\varphi, \\
 k_{\theta_{max}} &= \max_{\varphi} k_\theta(\varphi), \quad k_{\theta_{av}} = \frac{1}{2\pi} \int_0^{2\pi} k_\theta(\varphi) d\varphi, \\
 i_\varphi &= \frac{k_{\varphi_{av}}}{k_{\varphi_{max}}}, \quad i_\theta = \frac{k_{\theta_{av}}}{k_{\theta_{max}}}.
 \end{aligned} \tag{5.1}$$

La idea es caracterizar la isotropía utilizando un índice doble:

- i_φ , indica la simetría de la distribución en todas las direcciones,
- i_θ , indica la homogeneidad en las distancias.

Ambos índices, por definición, pertenecen al intervalo $[0, 1]$. Las figuras 5.2(a) y 5.2(b) ilustran las diferencias entre ambos factores.

(a) Distribución del error entorno del estado de referencia $|0\rangle$ con índices $i_\varphi = 1$ e $i_\theta < 1$.

(b) Distribución del error entorno del estado de referencia $|0\rangle$ con índices $i_\varphi < 1$ e $i_\theta = 1$.

Figura 5.2: Diferencia entre los factores i_φ e i_θ del doble índice isótropo geométrico.

En el caso de un estado de referencia genérico $|\psi\rangle$ se aplica un cambio de coordenadas (rotación) de forma a alinear el eje Z con el estado de referencia.

Esta definición se puede generalizar para el caso de estados multiqubits si se considera la representación ilustrada en la sección 4.4.1. En este caso el cálculo, utilizando aproximaciones numéricas realizadas mediante simulación, resulta poco práctico por la necesidad de una cantidad importante de estados ruidosos. Debido a esto se ha limitado el uso de este índice para estados de 1-qubit.

5.1.1. Aproximaciones numéricas

Para realizar aproximaciones numéricas en las simulaciones se considera una partición en la coordenada φ en M partes iguales ($\Delta\varphi_j$) de forma que

¹La definición a continuación es similar a la existente en [40]. El cambio fue para una mejor adecuación de $f(\varphi, \theta)$ como una densidad de probabilidad en el área, en lugar de en sectores angulares.

$$\sum_{j=1}^M \Delta\varphi_j = 2\pi. \quad (5.2)$$

En este caso las variables definidas en 5.1 pueden ser aproximadas por

$$\begin{aligned} k_{\varphi_j} &\approx \frac{2\pi}{\Delta\varphi} \frac{c_j}{N}, & k_{\theta_j} &\approx \bar{\theta}_j k_{\varphi_j}, \\ k_{\varphi_{max}} &= \max_j k_{\varphi_j}, & k_{\theta_{max}} &= \max_j k_{\theta_j}, \\ k_{\varphi_{av}} &= \frac{1}{M} \sum_{j=1}^M k_{\varphi_j}, & k_{\theta_{av}} &= \frac{1}{M} \sum_{j=1}^M k_{\theta_j}, \end{aligned} \quad (5.3)$$

siendo c_j la cantidad de estados en la j -ésima partición, N la cantidad total de estados y $\bar{\theta}_j$ el valor medio de la coordenada θ de los estados en la j -ésima partición. Con estos valores el doble índice $i_{iso} = (i_\varphi, i_\theta)$ es determinado como en (5.1).

5.1.2. Ejemplos

Con la finalidad de ilustrar el comportamiento del índice isótropo geométrico, en esta subsección se muestran algunos ejemplos de su cálculo analítico y aproximaciones numéricas mediante simulación, utilizando algunos de los modelos de errores continuos presentados en el capítulo 4.

5.1.3. Error isótropo en el entorno del estado $|0\rangle$

Sea una distribución de error uniforme en el entorno del estado de referencia $|0\rangle$, en un sector esférico de ángulo máximo θ_{max} :

$$f(\varphi, \theta) = \begin{cases} \frac{1}{2\pi(1 - \cos(\theta_{max}))}, & \theta \in [0, \theta_{max}], \varphi \in [0, 2\pi]. \\ 0, & \text{en otro caso.} \end{cases}$$

En este caso el doble índice geométrico puede ser calculado como

$$\begin{aligned} k_\varphi(\varphi) &= \lim_{\Delta\varphi \rightarrow 0} \frac{2\pi}{\Delta\varphi} \int_{\varphi - \frac{\Delta\varphi}{2}}^{\varphi + \frac{\Delta\varphi}{2}} \int_0^{\theta_{max}} \frac{\sin(\theta)}{2\pi(1 - \cos(\theta_{max}))} d\theta d\varphi \\ &= \lim_{\Delta\varphi \rightarrow 0} \frac{1}{\Delta\varphi(1 - \cos(\theta_{max}))} \int_{\varphi - \frac{\Delta\varphi}{2}}^{\varphi + \frac{\Delta\varphi}{2}} (1 - \cos(\theta_{max})) d\varphi \\ &= 1, \end{aligned}$$

$$\begin{aligned}
 k_\theta(\varphi) &= \lim_{\Delta\varphi \rightarrow 0} \frac{2\pi}{\Delta\varphi} \int_{\varphi - \frac{\Delta\varphi}{2}}^{\varphi + \frac{\Delta\varphi}{2}} \int_0^{\theta_{max}} \frac{\theta \sin(\theta)}{2\pi(1 - \cos(\theta_{max}))} d\theta d\varphi \\
 &= \lim_{\Delta\varphi \rightarrow 0} \frac{1}{\Delta\varphi(1 - \cos(\theta_{max}))} \int_{\varphi - \frac{\Delta\varphi}{2}}^{\varphi + \frac{\Delta\varphi}{2}} (\sin(\theta) - \theta \cos(\theta))|_0^{\theta_{max}} d\varphi \\
 &= \frac{\sin(\theta_{max}) - \theta_{max} \cos(\theta_{max})}{1 - \cos(\theta_{max})}, \\
 &\Rightarrow i = (i_\varphi, i_\theta) = (1, 1).
 \end{aligned}$$

A modo de comparación con la aproximación numérica se realiza la generación de estados aleatorios utilizando las variables aleatorias α y β con distribución uniforme en el intervalo $[0, 1]$.

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + \sin\left(\frac{\theta}{2}\right) e^{i\varphi}, \quad \begin{cases} \varphi = 2\pi\alpha, \\ \theta = \cos^{-1}(1 - \beta(1 - \cos(\theta_{max}))). \end{cases}$$

Las figuras 5.3(a) y 5.3(b) ilustran el error isótropo uniforme generado para los valores de θ_{max} de $\frac{\pi}{18}$ y $\frac{\pi}{4}$.

En la tabla 5.1 se presenta el resultado de las aproximaciones numéricas en función de los valores de θ_{max} para tres grupos de puntos aleatorios generados (1000, 10000 y 100000). Para cada valor de θ_{max} han sido realizados 100 experimentos.

Como es posible observar en la figura 5.4 los valores de los índices no varían con el ángulo θ_{max} , pero sí en función de la cantidad de puntos simulados. Para una cantidad de 100000 puntos la aproximación es buena, con ambos valores de aproximadamente 0,98, con una desviación estándar baja ($\approx 0,005$).

Figura 5.3: Error isótropo uniforme alrededor del estado de referencia $|0\rangle$.

5.2. Índice de isotropía estadístico

Como se observa en la sección 4.4 la representación de operador de densidad define completamente un ensamble de estados cuánticos. Siendo así tiene sentido calcular la isotropía teniendo como base la representación de un estado cuántico proveniente de un error isótropo introducido en la definición 4.11.

Cantidad de estados	θ_{max}	$i_{iso}(i_\varphi; i_\theta)$	Desv. est. $(i_\varphi; i_\theta)$
1000	$\pi/36$	(0,8145; 0,8006)	(0,0393; 0,0376)
	$\pi/18$	(0,8078; 0,8000)	(0,0350; 0,0386)
	$\pi/8$	(0,8100; 0,8022)	(0,0359; 0,0378)
	$\pi/4$	(0,8137; 0,8031)	(0,0389; 0,0401)
	$\pi/2$	(0,8049; 0,7969)	(0,0357; 0,0332)
10000	$\pi/36$	(0,9331; 0,9270)	(0,0145; 0,0161)
	$\pi/18$	(0,9307; 0,9270)	(0,0151; 0,0170)
	$\pi/8$	(0,9316; 0,9287)	(0,0145; 0,0156)
	$\pi/4$	(0,9337; 0,9295)	(0,0140; 0,0150)
	$\pi/2$	(0,9304; 0,9272)	(0,0116; 0,0132)
100000	$\pi/36$	(0,9780; 0,9762)	(0,0043; 0,0049)
	$\pi/18$	(0,9778; 0,9765)	(0,0049; 0,0049)
	$\pi/8$	(0,9778; 0,9764)	(0,0047; 0,0052)
	$\pi/4$	(0,9767; 0,9765)	(0,0045; 0,0046)
	$\pi/2$	(0,9779; 0,9762)	(0,0050; 0,0046)

Cuadro 5.1: Valores de las aproximaciones numericas

5.2.1. Descomposición de un estado mezcla

Sea ρ un estado mezcla cualquiera. Es siempre posible descomponer ρ como

$$\rho = p \frac{I}{2^n} + (1-p) \hat{\rho}, \quad (5.4)$$

siendo $\hat{\rho}$ una matriz de densidad con al menos un valor propio nulo. En el caso de que ρ tenga al menos un valor propio nulo se tiene que $p = 0$. Además esta descomposición es única.

Demostración. Como la matriz de densidad ρ de un estado de n qubits es una matriz hermítica, es siempre posible la diagonalización mediante una matriz de cambio de base M . Luego

$$\rho_d = M^\dagger \rho M = \lambda I + \Theta = 2^n \lambda \frac{I}{2^n} + \text{Tr}(\Theta) \hat{\rho}_d = p \frac{I}{2^n} + (1-p) \hat{\rho}_d, \quad (5.5)$$

siendo λ el menor valor propio de ρ (todos los valores propios son positivos y suman 1).

Luego

$$\rho = M \rho_d M^\dagger = p \frac{I}{2^n} + (1-p) M \hat{\rho}_d M^\dagger = p \frac{I}{2^n} + (1-p) \hat{\rho}. \quad (5.6)$$

□

5.2.2. Definición del índice

Considerando la definición 4.11 y la descomposición de la sección anterior, es posible ver que la ecuación (4.34) es un caso particular de (5.4). Luego un estado ρ es un estado cuántico mezcla proveniente de un error isótropo, respecto al estado de referencia $|\psi\rangle$, si al realizar la descomposición anterior $\hat{\rho} = \rho_\psi$ o $\hat{\rho} = \rho_{N\psi}$.

Figura 5.4: Variación de los valores del índice de isotropía en función del ángulo θ_{max} . Cada curva representa una cantidad de puntos en los experimentos (1000 (azul), 2000 (verde), 5000 (roja), 10000 (celeste), 20000 (magenta), 50000 (amarilla) y 100000 (negra)).

Definición 5.2 (Índice de isotropía (estadístico)). Se define el doble índice de isotropía (estadístico) para un estado cuántico ρ , considerando como estado de referencia el estado puro $\rho_\psi = |\psi\rangle\langle\psi|$ al par

$$Iso_\psi(\rho) \triangleq (A, p) \quad (5.7)$$

siendo

- A , la *Alineación Isótropa*, definida como

$$A = Fid(\hat{\rho}, \rho_\psi) - Fid(\hat{\rho}, \rho_{N\psi}) \quad (5.8)$$

donde Fid es la *fidelidad* entre estados cuánticos, $\hat{\rho}$ proviene de la descomposición de la ecuación (5.4) y $\rho_{N\psi} = (I - |\psi\rangle\langle\psi|)/(2^n - 1)$ (*mezcla isótropa ortogonal del estado ρ_ψ*),

- y $p = 2^n \lambda$, el *Ancho Isótropo*, siendo λ el menor autovalor de ρ .

Propiedades

Propiedad 1. Por construcción el *Ancho Isótropo* p es una probabilidad, luego tiene un valor entre 0 y 1. Cuando $p = 1$ el estado ρ es igual a $I/2^n$ y $\hat{\rho}$ pierde sentido, siendo indefinido el valor de la *Alineación Isótropa* A (se puede tomar por defecto igual a 1). En este caso ρ representa un error isótropo. En el caso de $p = 0$, ρ es un estado mezcla con, al menos, un autovalor nulo.

Propiedad 2. La *Alineación Isótropa* A tiene un valor en el intervalo $[-1, 1]$. Esto es debido a que la fidelidad entre estados cuánticos toma valores en el intervalo $[0, 1]$ y que los estados ρ_ψ y $\rho_{N\psi}$ son ortogonales. Para todo p si $A = 1$ o $A = -1$, ρ representa un error isótropo.

Propiedad 3. El índice es invariante frente a una transformación unitaria U aplicada en ambos estados, el estudiado y el de referencia, o sea, $Iso_\varphi(\rho) = Iso_\psi(\sigma)$ siendo $|\psi\rangle = U|\varphi\rangle$, $\sigma = U\rho U^\dagger$. Esto es debido a que los autovalores son invariantes frente a transformaciones unitarias, así como la fidelidad.

Propiedad 4. Dada una operación cuántica, que preserva la traza, definida mediante operadores de Kraus como

$$\varepsilon(\rho) = \sum_{i=1}^k M_i \rho M_i^\dagger, \quad \text{con} \quad \sum_{i=1}^k M_i^\dagger M_i = I, \quad (5.9)$$

el *Ancho Isótropo* p es no decreciente siempre que $\varepsilon(\rho)$ sea *unital* [23], o sea,

$$\sum_{i=1}^k M_i M_i^\dagger = I. \quad (5.10)$$

Demostración. Sea el estado ρ con su descomposición

$$\rho = p \frac{I}{2^n} + (1-p)\hat{\rho}.$$

Así

$$\varepsilon(\rho) = \frac{p}{2^n} \sum_{i=1}^k M_i M_i^\dagger + (1-p) \sum_{i=1}^k M_i \hat{\rho} M_i^\dagger = p \frac{I}{2^n} + (1-p) \sum_{i=1}^k M_i \hat{\rho} M_i^\dagger. \quad (5.11)$$

Luego el *Ancho Isótropo* es igual a $p + (1-p)\alpha$, siendo $\alpha \geq 0$ el *Ancho Isótropo* del estado dado por $\sum_{i=1}^k M_i \hat{\rho} M_i^\dagger$. Como se observa de la expresión, la condición dada por la ecuación (5.10) es suficiente, pero no necesaria.

En el caso particular de evoluciones unitarias se tiene un único operador U . En este caso los autovalores de $\hat{\rho}$ no se alteran, por lo que se mantiene el *Ancho Isótropo*. $\square$

En el apéndice B.2 se indica una forma eficiente de calcular el índice de isotropía para un estado cuántico ρ , considerando un estado de referencia puro cualquiera.

5.2.3. Representación gráfica: triángulo de isotropía

Como indicado en la Prop. (1), cuando $p = 1$ el par ordenado $Iso_\varphi(\rho) \triangleq (A, 1)$ representa el mismo punto $\forall A$. Por esto la representación gráfica más adecuada del índice es un triángulo.

Sea el par ordenado $Iso_\varphi(\rho) = (A, p)$. Para determinar las coordenadas sobre el triángulo se realiza el mapeo

$$x = (1-p)A, \quad (5.12)$$

$$y = p. \quad (5.13)$$

La figura 5.5 ilustra los estados de referencia (ρ_φ), el estado *mezcla isótropa ortogonal* ($\rho_{N\varphi}$) y el estado $I/2^n$.

Propiedades del triángulo de isotropía

Como el índice de isotropía no se altera frente a un cambio de base, a continuación se considerará el estado $|0\rangle$ como el estado de referencia.

Figura 5.5: Triángulo de isotropía. En la figura se observa la representación del doble índice de isotropía de los estados ρ_φ (cuadrado), $\rho_{N\varphi}$ (círculo) y $I/2^n$ (cruz), considerando como estado de referencia el $|\varphi\rangle$.

Propiedad 1 (Estados con probabilidad 0). Los estados que tienen probabilidad cero al utilizar el proyector del estado de referencia se encuentran en la parte inferior del triángulo, en el intervalo $\left[-1, -\sqrt{\frac{1}{2^n-1}}\right]$.

Considerando la descomposición de la ecuación (5.4), la probabilidad de proyectarse sobre el estado de referencia es como mínimo $prob = p/(2^n)$. Esto implica que para tener probabilidad nula el ancho isotropo debe ser $p = 0$ y $\rho = \hat{\rho}$. Como $Fid(\hat{\rho}, \rho_0) = 0$, ya que es la raíz cuadrada de la probabilidad, en este caso se tiene que $A = -Fid(\hat{\rho}, \rho_{N0})$. Si $\rho = \rho_{N0} \Rightarrow A = -1$. Los estados de mínima fidelidad con respecto al ρ_{N0} y probabilidad cero (ortogonales al estado de referencia $|0\rangle$), son los $2^n - 1$ estados de la base cartesiana. En este caso

$$\hat{\rho} = \rho_i = |i\rangle\langle i|, \quad Fid(\hat{\rho}, \rho_{N0}) = \sqrt{\langle i|\rho_{N0}|i\rangle} = \frac{1}{\sqrt{2^n-1}}. \quad (5.14)$$

Propiedad 2 (Isotropía de estados puros). Sea un ensamble compuesto por un único estado puro $|\psi\rangle = a_0|0\rangle + a_1e^{i\varphi_1}|1\rangle + \dots + a_{2^n-1}e^{i\varphi_{2^n-1}}|2^n-1\rangle$. Como los estados puros tienen solamente un autovalor no nulo igual a 1, $p = 0$. Para el cálculo de A se tiene que

$$Fid_0 = Fid(|\psi\rangle, |0\rangle) = a_0, \quad (5.15)$$

$$\begin{aligned} Fid_{N0} &= Fid(|\psi\rangle, \rho_{N0}) = \sqrt{\langle \psi|\rho_{N0}|\psi\rangle}, \\ &= \sqrt{\sum_{k=1}^{2^n-1} \frac{a_k^2}{2^n-1}} = \sqrt{\frac{1-a_0^2}{2^n-1}}, \end{aligned} \quad (5.16)$$

$$\Rightarrow A = Fid_0 - Fid_{N0} = a_0 - \sqrt{\frac{1-a_0^2}{2^n-1}}. \quad (5.17)$$

Luego, los estados puros se encuentran en la parte inferior del triángulo, con alineaciones en el intervalo $\left[-\sqrt{\frac{1}{2^n-1}}, 1\right]$.

Como la probabilidad de proyectarse sobre el estado de referencia es a_0^2 , todos los estados puros con la misma probabilidad tienen el mismo índice de isotropía

$$Iso_0(\rho) = \left(a_0 - \sqrt{\frac{1 - a_0^2}{2^n - 1}}, 0 \right). \quad (5.18)$$

La figura 5.6 ilustra la zona de los estados puros (círculos negros) y estados con probabilidad cero (cuadrados celestes). Es interesante notar que todos los estados puros con probabilidad nula tienen índice de isotropía en la intersección de ambas zonas, $Iso_0(\rho) = \left(-\frac{1}{\sqrt{2^n - 1}}, 0 \right)$.

Figura 5.6: Propiedades 1 y 2. Zona de índice de isotropía para estados puros (círculos negros) y estados con probabilidad cero de proyectarse sobre la referencia (cuadrados celestes).

Propiedad 3 (Estados afectados por errores de despolarización de canal). Sea un estado inicial ρ con una descomposición dada por la ecuación (5.4) que tenga un índice de isotropía dado por $Iso_0(\rho) = (A, p)$. Al aplicar un error del tipo de canal de despolarización, sección 4.2.1, se tiene que

$$\varepsilon(\rho) = \gamma \frac{I}{2^n} + (1 - \gamma) \rho, \quad (5.19)$$

para $\gamma \in [0, 1]$. En este caso se tiene que

$$\begin{aligned} \varepsilon(\rho) &= \gamma \frac{I}{2^n} + (1 - \gamma) p \frac{I}{2^n} + (1 - \gamma) (1 - p) \hat{\rho} \\ &= (\gamma + p - \gamma p) \frac{I}{2^n} + (1 - \gamma - p + \gamma p) \hat{\rho} \\ &= \alpha \frac{I}{2^n} + (1 - \alpha) \hat{\rho}. \end{aligned} \quad (5.20)$$

Luego A se mantiene constante (mismo $\hat{\rho}$) y el ancho isotrópico varía desde p ($\gamma = 0$) a 1 ($\gamma = 1$). Como se puede observar en la figura 5.7(a), las curvas son las rectas que unen el punto inicial en el triángulo al vértice superior (cuando $\rho = \frac{I}{2^n}$).

Propiedad 4 (Estados con evolución unitaria). Por la propiedad 4 del índice de isotropía, si el estado de referencia es fijo y el estado estudiado evoluciona mediante operadores unitarios, el *Ancho Isótropo* se mantiene constante. La figura 5.7(b) ilustra la evolución del índice para diferentes estados iniciales (uno de ellos el estado de referencia), considerando evoluciones unitarias aleatorias.

Observación 5.1. Aunque para una mezcla cualquiera no existe una relación directa entre la probabilidad de proyectarse sobre el estado de referencia y la alineación isotrópica, si se considera su definición se observa una correlación entre el aumento (disminución) de la alineación hacia el valor 1 (-1) y el aumento (disminución) de esta probabilidad. Esto puede ser observado en las figuras 5.8 y 5.9.

(a) Evolución del índice debido a errores de despolarización de canal (probabilidades de 0 a 1). Estado inicial: estado de referencia (círculos negros), estado aleatorio con ancho isotrópico con probabilidad cero (cuadrados celestes) y estado igual a 0.25 (cuadrados celestes) y estado aleatorio con ancho isotrópico inicial distinto de cero ancho isotrópico igual a 0.5 (cruces rojas).
(b) Evolución del índice debido a evoluciones unitarias aleatorias. Estado inicial: estado de referencia (círculos negros), estado aleatorio con ancho isotrópico con probabilidad cero (cuadrados celestes) y estado igual a 0.25 (cuadrados celestes) y estado aleatorio con ancho isotrópico inicial distinto de cero ancho isotrópico igual a 0.5 (cruces rojas).

Figura 5.7: Triángulo de isotropía. Ejemplos de las propiedades 3 y 4. Estados de 4 qubits ($n = 4$).

Figura 5.8: Zonas con probabilidades constantes iguales a 0,8 (negro), 0,6 (marrón oscuro), $1/2^n$ (marrón) y $0,25/2^n$ (marrón claro).

5.2.4. Estado isotrópico de Horodecki

Un estado isotrópico [66] es un estado bipartito de $2n$ qubits, dimensión $2^n \otimes 2^n$, que es invariante frente a transformaciones unitarias del tipo $U \otimes U^*$, siendo U una matriz unitaria cualquiera y U^* su matriz conjugada. O sea, cualquier estado que cumpla la propiedad

$$\rho = (U \otimes U^*) \rho (U \otimes U^*)^\dagger, \forall U. \quad (5.21)$$

Considerando $d = 2^n$, estos estados forman una familia determinada por un único parámetro α y se puede describir como

$$\rho = (1 - \alpha) \frac{I}{d^2} + \alpha |\phi\rangle\langle\phi|, \quad -\frac{1}{d^2 - 1} \leq \alpha \leq 1 \quad (5.22)$$

siendo

$$|\phi\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{2^n-1} |j\rangle \otimes |j\rangle \quad (5.23)$$

Figura 5.9: Variación de la probabilidad de proyectarse sobre el estado de referencia. Zonas más oscuras tienen mayor probabilidad.

con $|j\rangle$ los estados de la base del espacio de dimensión d .

A diferencia del estado proveniente de un error isotrópico, por definición un estado isotrópico siempre posee una cantidad par de qubits. A continuación se calcula el índice de isotropía.

En el caso de tener el parámetro $\alpha \in [0, 1]$, y considerando que $|\phi\rangle$ es un estado puro, es inmediato observar que el estado ya se encuentra expresado como una mezcla con una descomposición de la forma (5.4). Así, se tiene un índice $Is_{\phi}(\rho) = (1, 1 - \alpha)$, o sea una *Alineación Isótropa* igual a 1 y un *Ancho Isótropo* igual a $1 - \alpha$. En este caso es simple interpretar el estado como una despolarización de canal del estado original $|\phi\rangle$.

En el caso de $\alpha \in [-1/(d^2 - 1), 0)$, si se utiliza $p = 1 + (d^2 - 1)\alpha$, se tiene que

$$\rho = (1 - \alpha) \frac{I}{d^2} + \alpha |\phi\rangle\langle\phi| = p \frac{I}{d^2} + (1 - p) \frac{I - |\phi\rangle\langle\phi|}{d^2 - 1} = p \frac{I}{d^2} + (1 - p) \rho_{N\phi}, \quad (5.24)$$

siendo $\rho_{N\phi}$ el estado mezcla isotrópica ortogonal del estado $|\phi\rangle\langle\phi|$. Se observa que con la definición anterior $p \in [0, 1]$. Con este resultado se tiene que la *Alineación Isótropa* es siempre -1 y el *Ancho Isótropo* es igual a $1 + (d^2 - 1)\alpha$.

La figura 5.10 ilustra la evolución del índice en el triángulo en función de α .

Figura 5.10: Evolución en el triángulo de isotropía del estado isotrópico en función α : $0 \leq \alpha \leq 1$ (cruces azules) y $-1/(d^2 - 1) \leq \alpha < 0$ (círculos rojos).

Parte III

Algoritmos cuánticos en sistemas abiertos

Capítulo 6

Algoritmo de Grover

El algoritmo de Grover es un algoritmo cuántico de búsqueda [58, 59, 60] que resuelve el problema de hallar un elemento marcado en un conjunto desordenado de N elementos utilizando un sistema, denominado oráculo, que es capaz de identificar a este elemento. Presenta una mejora cuadrática en el orden en relación al clásico algoritmo de búsqueda por fuerza bruta, que se ha demostrado que es óptima [22, 135].

Sin embargo, como mencionado en el capítulo 4, los sistemas cuánticos contienen errores, sea por no estar completamente aislados del ambiente, como por la implementación de los circuitos cuánticos. Considerando esto la mejora de desempeño se ve afectada y se han realizado investigaciones de las diferentes causas y efectos [13, 32, 104, 99, 93, 51, 7, 111].

Aunque la teoría cuántica tolerante a fallos [4, 56], garantizan la corrección de errores bajo ciertas condiciones (cierto umbral de error), es un método muy costoso computacionalmente. Así, es importante analizar el desempeño de los algoritmos cuánticos con errores antes de realizar propuestas de correcciones.

En este capítulo se analiza el deterioro en la mejora del orden de la búsqueda considerando dos modelos de error: error de despolarización del estado completo (TDCh), extendiendo el trabajo realizado en [125], y error de despolarización local (LDCh), aplicado en cada qubit del sistema por separado. Se estudia la propagación del índice de isotropía en cada caso, con el objetivo de analizar como afecta cada tipo de error.

6.1. Descripción del algoritmo de Grover

El algoritmo de Grover es un algoritmo cuántico de búsqueda que resuelve el problema de hallar un elemento marcado en un conjunto desordenado de N elementos utilizando $k_{Gr} = \left\lceil \frac{\pi}{4} \sqrt{N} \right\rceil$ veces un oráculo. El problema puede ser descrito como a continuación. Sea un conjunto de $N = 2^n$ estados cuánticos en un espacio de Hilbert ($\mathcal{H} = \mathbb{C}^N$), siendo n la cantidad de qubits, y un estado desconocido en la base canónica marcado entre ellos (solución al problema). Dado un sistema, u operador, denominado oráculo que identifica el elemento marcado, el objetivo es hallar este elemento marcado (con alta probabilidad) utilizando la menor cantidad de pasos posible (consultas al oráculo).

Sea $|t\rangle$ el elemento marcado de la base y

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \quad (6.1)$$

el estado superposición uniforme de todos los estados de la base. Si se utiliza la descripción de operadores de densidad ($\mathcal{H} = \mathbb{C}^N \times \mathbb{C}^N$) se tiene $\rho_t = |t\rangle\langle t|$ y $\rho_s = |s\rangle\langle s|$, respectivamente. El algoritmo de búsqueda se describe a continuación en el **Algoritmo 1**.

Al aplicar k veces los operadores de oráculo y difusor, el estado resultante puede ser calculado como

Algoritmo 1: Algoritmo de búsqueda de Grover

1. Se inicializa con el estado inicial ρ_0 ,
 2. Se aplica el operador del oráculo $O = 2|t\rangle\langle t| - I$,
 3. Se aplica el operador de difusión $D = 2|s\rangle\langle s| - I$,
 4. Se repiten $\left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor - 1$ veces los pasos 2 y 3,
 5. Se realiza una medición proyectiva en la base canónica en cada qubit. El estado marcado es obtenido con alta probabilidad para $N \gg 1$.
-

$$\rho(k) = G^k \rho_s (G^\dagger)^k, \quad (6.2)$$

donde $G = DO$ es el operador de Grover. El estado mezcla resultante puede ser representado por el operador

$$\rho(k) = |s_k\rangle\langle s_k|, \quad (6.3)$$

siendo

$$\begin{aligned} |s_k\rangle &= \sin((2k+1)\theta) |t\rangle + \cos((2k+1)\theta) |\bar{t}\rangle, \\ |\bar{t}\rangle &= \frac{1}{\sqrt{N-1}} \sum_{\substack{i=0 \\ i \neq t}}^{N-1} |i\rangle \text{ y } \theta = \arcsin\left(\frac{1}{\sqrt{N}}\right). \end{aligned} \quad (6.4)$$

Luego, la probabilidad de éxito (obtener el elemento marcado como resultado de la medición) después de k pasos es igual a

$$p(k) = \sin^2((2k+1)\theta). \quad (6.5)$$

El estado $|\bar{t}\rangle$ es ortogonal al estado $|t\rangle$, y se puede observar que el operador de Grover G realiza una doble reflexión en el hiperplano formado por ambos estados, como en ilustrado en la figura 6.1. Es decir, si el estado $|\psi\rangle$ forma un ángulo $\theta/2$ con $|\bar{t}\rangle$, el operador del oráculo lo refleja sobre $|\bar{t}\rangle$ y operador de difusión lo vuelve a reflejar sobre el estado original $|\psi\rangle$, obteniéndose un nuevo ángulo θ con $|\bar{t}\rangle$. Luego de una cierta cantidad de pasos, el estado es cercano a $|t\rangle$ ($(2k+1)\theta \approx \pi/2$).

Figura 6.1: Hiperplano correspondiente al algoritmo de Grover, formado por $|t\rangle$ y $|\bar{t}\rangle$.

6.2. Error de despolarización de estado completo (TDCh)

En esta sección se analiza como el modelo de error de canal de despolarización, y en particular, la despolarización en todo el estado (denominado canal de despolarización total, o TDCh) afecta el resultado de la búsqueda. En este estudio, en lugar de minimizar el costo medio como se propone

en [125], se realiza una maximización de la probabilidad ya que es posible obtener una expresión para el resultado. Así es posible comparar los resultados con relación a los obtenidos al utilizar un modelo de error de canal de despolarización local (LDCh), como el presentado en la sección 6.3.

Como se indica en la subsección 4.2.1, el error de canal de despolarización total es un modelo de error que realiza una operación cuántica que transforma un estado de n qubits ρ a una mezcla del estado máximamente mezclado I/N con probabilidad γ , y el original con probabilidad $(1 - \gamma)$

$$\varepsilon(\rho, \gamma) = (1 - \gamma)\rho + \gamma \frac{I}{N}. \quad (6.6)$$

Esto se puede interpretar en el sentido probabilístico como una distribución binomial, en donde se tiene ρ (éxito) o I/N (fracaso). La figura 6.2 ilustra la aplicación del modelo de error.

Figura 6.2: Modelo de error TDCh para el algoritmo de Grover.

6.2.1. Evolución del estado

En cada paso del algoritmo se aplica el operador de Grover G seguido de un error del tipo TDCh. En este caso la matriz de densidad del estado después de k pasos es

$$\hat{\rho}(k, \gamma) = (1 - \gamma)^k \rho(k) + (1 - (1 - \gamma)^k) \frac{I}{N}, \quad (6.7)$$

siendo $\rho(k)$ el estado de la ecuación (6.2). Esto significa que las probabilidades de éxito y de fracaso son $(1 - \gamma)^k$ y $1 - (1 - \gamma)^k$, respectivamente. Así, la probabilidad de obtener el elemento marcado es

$$\hat{p}(k, \gamma) = (1 - \gamma)^k p(k) + \frac{1 - (1 - \gamma)^k}{N}, \quad (6.8)$$

siendo $p(k)$ dado por la ecuación (6.5). Al realizar una medición en la base canónica, la componente $\frac{I}{N}$ tiene una probabilidad uniforme de dar como resultado cualquier estado de la base.

Como se puede observar en las figuras 6.3(a) y 6.3(b) el primer máximo de la probabilidad $\hat{p}(k)$ (correspondiente al paso k_{max}) se mueve hacia valores menores de k a medida que γ o N aumentan [99]. Este efecto también es apreciado en la figura 6.4(a), que ilustra que el comportamiento es similar para cualquier cantidad de qubits (por ejemplo, $n = 10$).

Si se calcula el máximo de la ecuación (6.8) es posible hallar la expresión para el valor del paso en el cual se da la probabilidad máxima¹, k_{max} . Se tiene como resultado que

$$k_{max}(\gamma) = \max \left(\left\lfloor \frac{\pi - \arcsin \delta - \arcsin \left(\left[1 - \frac{2}{N}\right] \delta \right)}{4\theta} \right\rfloor, 1 \right) \quad (6.9)$$

siendo

$$\delta = \sqrt{\frac{1}{1 + \left(\frac{4\theta}{\ln(1-\gamma)} \right)^2}} \quad (6.10)$$

¹El cálculo se encuentra en Ilan Cohn; ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García López de Lacalle. *Grover's search with local and total depolarizing channel errors: complexity analysis. International Journal of Quantum Information. Vol. 14, No 2, 1650009. World Scientific. 2016.*

(a) Probabilidad en función de la cantidad de pasos para estados de $n = 10$ qubits, considerando diferentes valores de γ : 0 (A), $1/(4\sqrt{N})$ (B), $1/\sqrt{N}$ (C), $4/\sqrt{N}$ (D) and 1 (E).

(b) Probabilidad en función de la cantidad normalizada de pasos (división por la cantidad de pasos del algoritmo para cada N), considerando $\gamma = 0,01$ y para distintas cantidades de qubits n ($N = 2^n$): 10 (A), 12 (B), 14 (C) and 16 (D).

Figura 6.3: Probabilidades en función de la cantidad de pasos para el algoritmo de Grover con error del tipo TDCh.

y θ dado por la ecuación (6.4).

Utilizando la ecuación (6.9) se puede realizar dos aproximaciones para el cálculo del paso de máxima probabilidad (en ambos casos se considera $n \gg 1$, $N = 2^n$):

- si $0 \leq \gamma \ll \frac{2\pi}{\sqrt{N}}$

$$k_{max}(\gamma) \approx \left\lfloor \frac{\pi\sqrt{N}}{4} - \frac{N\gamma}{8} \right\rfloor, \quad (6.11)$$

- y si $\frac{8}{\pi\sqrt{N}} \ll \gamma \leq 1$

$$k_{max}(\gamma) \approx \max \left(\left\lfloor -\frac{1}{\ln(1-\gamma)} + \frac{1}{2} \sqrt{1 + \left(\frac{2}{\ln(1-\gamma)} \right)^2} \right\rfloor, 1 \right). \quad (6.12)$$

La aproximación de la ecuación (6.11) es válida para valores suficientemente chicos de γ , mientras la de la ecuación (6.12) vale para valores grandes. En la figura 6.4(b) se observa el comportamiento de las diferentes aproximaciones de k_{max} .

6.2.2. Estimación del orden

Si se aplica amplificación clásica y se detiene el algoritmo en el paso k , es posible determinar el orden calculando el valor esperado de la cantidad de pasos (valor esperado del uso del oráculo) con la expresión

$$MC_{k,\gamma} := MC[k, \hat{p}(k, \gamma)] = \frac{k}{\hat{p}(k, \gamma)}. \quad (6.13)$$

Se sabe que el algoritmo clásico de *fuerza bruta* (probar los elementos hasta hallar el marcado) es de orden $\Theta(N)$ con un factor constante de $1/2$.

Deteniendo el algoritmo en k_{Gr}

Si se detiene el algoritmo en el paso original, k_{Gr} , se tiene un costo estimado de

$$MC_{k_{Gr},\gamma} = \frac{k_{Gr}}{(1-\gamma)^{k_{Gr}} p(k_{Gr}) + \frac{1}{N} (1 - (1-\gamma)^{k_{Gr}})}. \quad (6.14)$$

(a) Cantidad de pasos en función de la probabilidad de error (γ) y la cantidad de qubits n .

(b) Cantidad de pasos en función de la probabilidad de error (γ), para el caso de $n = 10$ qubits utilizando diferentes aproximaciones para el cálculo de k_{max} : simulación del algoritmo (A) ecuación (6.9) (B), ecuación (6.11) (C) y ecuación (6.12) (D).

Figura 6.4: Cantidad de pasos para la máxima probabilidad (normalizada, k_{max}/k_{Gr}) en el algoritmo de Grover con error del tipo TDCh.

Considerando la hipótesis de que $N \gg 1$, se tiene que $p(k_{Gr}) \approx (1 - 1/N)$ y $k_{Gr} = \left\lfloor \frac{\pi\sqrt{N}}{4} \right\rfloor \approx \frac{\pi\sqrt{N}}{4}$, y es posible aproximar la ecuación (6.14) por

$$MC_{k_{Gr}, \gamma} \approx \frac{\frac{\pi\sqrt{N}}{4}}{(1 - \gamma)^{\frac{\pi\sqrt{N}}{4}} + \frac{1}{N} \left(1 - 2(1 - \gamma)^{\frac{\pi\sqrt{N}}{4}}\right)}. \quad (6.15)$$

Para cualquier $\gamma > 0$ fijo (N suficientemente grande)

$$MC_{k_{Gr}, \gamma} \approx \frac{\frac{\pi\sqrt{N}}{4}}{(1 - \gamma)^{\frac{\pi\sqrt{N}}{4}} + \frac{1}{N}} \approx \frac{\frac{\pi N\sqrt{N}}{4}}{N e^{\frac{\pi\sqrt{N}}{4} \ln(1 - \gamma)} + 1} \approx \frac{\pi N\sqrt{N}}{4}. \quad (6.16)$$

el orden es $\Theta(N\sqrt{N})$. O sea, el algoritmo pasa de ser mejor que el clásico, con un orden $\Theta(\sqrt{N})$ en el caso sin error, a ser peor que el clásico.

En el caso de $\gamma = f(N) = \mathbb{K}/N^\beta$, $\mathbb{K}$ constante y $\beta > 0$, se tiene que

$$\begin{aligned} MC_{k_{Gr}, f(N)} &\approx \frac{\frac{\pi N\sqrt{N}}{4}}{N e^{\frac{\pi\sqrt{N}}{4} \ln(1 - \frac{\mathbb{K}}{N^\beta})} + \left(1 - 2e^{\frac{\pi\sqrt{N}}{4} \ln(1 - \frac{\mathbb{K}}{N^\beta})}\right)} \\ &\approx \frac{\frac{\pi N\sqrt{N}}{4}}{N e^{-\frac{\mathbb{K}\pi}{4} N^{\frac{1}{2} - \beta}} + \left(1 - 2e^{-\frac{\mathbb{K}\pi}{4} N^{\frac{1}{2} - \beta}}\right)}. \end{aligned} \quad (6.17)$$

Donde se tiene los siguientes casos:

- $\beta < 1/2 \Rightarrow MC_{k_{Gr}, f(N)} \approx Nk_{Gr}$, luego el orden es $\Theta(N\sqrt{N})$ con un factor constante de $\pi/4$.
- $\beta = 1/2 \Rightarrow MC_{k_{Gr}, f(N)} \approx e^{\frac{\pi\mathbb{K}}{4}} k_{Gr}$, luego el orden es $\Theta(\sqrt{N})$ con un factor constante de $e^{\frac{\pi\mathbb{K}}{4}} \pi/4$.
- $\beta > 1/2 \Rightarrow MC_{k_{Gr}, f(N)} \approx k_{Gr}$, luego el orden es $\Theta(\sqrt{N})$ con un factor constante de $\pi/4$.

Así, para mantener el orden del algoritmo original es necesario que la probabilidad de error γ tienda a cero igual, o más rápido, que $1/\sqrt{N}$.

Este resultado es coherente con el obtenido en [125], en el que la condición hallada es $\gamma \ll 1/k_{Gr}$.

Aún así, el algoritmo de Grover con ruido puede superar al algoritmo clásico de fuerza bruta, siempre que el ancho del ruido no supere el valor $\gamma_{clasico}$ indicado a continuación. Para este valor de γ el costo medio es igual a $\frac{N}{2}$. Este valor de γ es determinado usando la ecuación 6.15

$$\gamma_{clasico} \approx 1 - \left(\frac{-1 + 2k_{Gr}}{-2 + N} \right)^{\frac{1}{k_{Gr}}}. \quad (6.18)$$

Si se considera $N \gg 1$,

$$\gamma_{clasico} \approx \frac{4 \ln \left(\frac{2\sqrt{N}}{\pi} \right)}{\pi \sqrt{N}}. \quad (6.19)$$

Deteniendo el algoritmo en k_{max}

Si se detiene el algoritmo en $k_{max}(\gamma)$, se tiene la siguiente expresión para la ecuación (6.13)

$$MC[k_{max}(\gamma), \hat{p}(k_{max}(\gamma), \gamma)] \quad (6.20)$$

En este caso los resultados al anterior, deteniendo en k_{Gr} , a diferencia de una constante.

Para todo valor fijo de $\gamma > 0$, el costo medio vale

$$MC_{k_{max}(\gamma), \gamma} \approx \frac{N}{9 - 8\gamma}. \quad (6.21)$$

Así el máximo error que deja el costo $MC[k_{max}(\gamma), \hat{p}(k_{max}(\gamma), \gamma)]$ con el mismo valor del clásico ($N/2$) es independiente de N , o sea $\gamma_{clasico} \approx 7/8$.

6.3. Error local de despolarización de canal (LDCh)

El error local de despolarización de canal (LDCh) transforma un estado ρ de n -qubits en un estado mezcla aplicando un error de despolarización de canal en forma independiente en cada qubit, o sea

$$\varepsilon(\rho) = \varepsilon_1(\rho, \alpha) \circ \varepsilon_2(\rho, \alpha) \circ \cdots \circ \varepsilon_n(\rho, \alpha), \quad (6.22)$$

donde $\varepsilon_i(\rho, \alpha)$ corresponde a la aplicación de la ecuación (6.6) en el qubit i , y α es la probabilidad de error. La figura 6.5 ilustra este modelo de error.

Figura 6.5: Modelo de error LDCh para el algoritmo de Grover.

Como realizado en la sección 6.2, se define la probabilidad de obtener el elemento marcado después de k pasos como $\hat{p}^L(k, \alpha)$, y el paso en el cual se halla el primer máximo de probabilidad como $k_{max}^L(\alpha)$.

6.3.1. OSR para la evolución del estado con LDCh

En el caso de estados de un qubit la operación de despolarización de canal puede ser calculada mediante los operadores de Kraus como

$$\varepsilon(\rho, \alpha) = \sum_{l=0}^4 M_l \rho M_l^\dagger, \quad (6.23)$$

siendo $M_0 = \sqrt{1 - \frac{3\alpha}{4}} \sigma_0$, $M_i = \sqrt{\frac{\alpha}{4}} \sigma_i$ ($i \in \{1, 2, 3\}$), y σ_μ las matrices de Pauli.

Al aplicarse errores locales en todos los n -qubits se tienen 4^n operadores (todas las posibles combinaciones), que son de la forma [82]

$$M_c = \sqrt{\left(1 - \frac{3\alpha}{4}\right)^m \left(\frac{\alpha}{4}\right)^{n-m}} \bigotimes_{q=1}^n \sigma_{i_q}, \quad (6.24)$$

con $i_q \in 0, \dots, 3$ y siendo m la cantidad de matrices identidad (σ_0) que aparecen en un M_c en particular.

Si se denomina paso a la aplicación del operador de Grover G seguida de un error LDCh, se tiene el siguiente estado después de k pasos

$$\hat{\rho}^L(k, \alpha) = \sum_{c_1}^{4^n} \cdots \sum_{c_k}^{4^n} \left[O_\Pi \rho_{ini} O_\Pi^\dagger \right], \quad (6.25)$$

siendo

$$O_\Pi = \prod_{l=1}^k (M_{c_l} G), \quad (6.26)$$

cada M_{c_l} está dado por la ecuación (6.24) y $\rho_{ini} = |s\rangle\langle s|$ es el estado inicial del algoritmo.

Para el primer paso ($k = 1$), esta probabilidad vale

$$\begin{aligned} \hat{p}^L(1, \alpha) &= \frac{1}{2^{3n-4}} \left[2^n (2^{n-1} - 1) \left(1 - \frac{\alpha}{2}\right)^n + \right. \\ &\quad \left. + \frac{1}{16} (2^n - 4)^2 \right], \end{aligned} \quad (6.27)$$

la cual para $n \gg 1$ queda

$$\hat{p}^L(1, \alpha) \approx \frac{1}{2^n} \left[2^3 \left(1 - \frac{\alpha}{2}\right)^n + 1 \right]. \quad (6.28)$$

Debido al aumento exponencial de la complejidad de la ecuación (6.25), se propone analizar $\hat{p}^L(k, \alpha)$ utilizando cotas inferior y superior utilizando el modelo TDCh, y una aproximación de primer orden.

6.3.2. Cotitas de probabilidad

El algoritmo de Grover con errores del tipo LDCh puede ser interpretado como un árbol ternario en el cual cada nivel corresponde a un paso. En el nodo inicial se tiene el estado $\rho_{ini} = |s\rangle\langle s|$, y en cada nodo se tiene los tres posibles caminos indicados en la figura 6.6. De hecho, se puede calcular la probabilidad de obtener el elemento marcado en el paso k considerando todos los posibles caminos (4^{nk} si se utilizan los operadores de Kraus).

Con la finalidad de hallar las cotas superior y inferior para las probabilidades, se realiza el estudio considerando el mejor y el peor caso, o sea $\rho_{unkown} \equiv G\rho G^\dagger$ o $\rho_{unkown} \equiv I/N$, respectivamente. Así las cotas son

$$\text{superior, } \hat{p}_u(k, \alpha) = (1 - \alpha^n)^k p(k) + \frac{1 - (1 - \alpha^n)^k}{N} \quad (6.29)$$

$$\text{inferior, } \hat{p}_l(k, \alpha) = (1 - \alpha)^{nk} p(k) + \frac{1 - (1 - \alpha)^{nk}}{N}, \quad (6.30)$$

Figura 6.6: Evolución del algoritmo con LDCh en un paso cualquiera considerado como un árbol ternario.

lo que implica

$$\hat{p}_l(k, \alpha) \leq \hat{p}^L(k, \alpha) \leq \hat{p}_u(k, \alpha), \quad (6.31)$$

$\forall k \in \mathbb{N}, \alpha \in [0, 1]$. Considerando que los resultados son de forma análoga al caso TDCh, ecuación (6.7), se definen las cotas en función de su ancho de error equivalente $\gamma(\alpha)$, o sea

$$\gamma_u(\alpha) = \alpha^n, \quad (6.32)$$

$$\gamma_l(\alpha) = 1 - (1 - \alpha)^n. \quad (6.33)$$

Aunque $\gamma_l(\alpha)$ proporciona una cota razonable para la probabilidad, no ocurre lo mismo con $\gamma_u(\alpha)$, como ilustrado en la figura 6.7. Una mejor aproximación se logra con

$$\gamma_u(\alpha) = \frac{n\alpha}{2 + n\alpha}. \quad (6.34)$$

6.3.3. Aproximación de primer orden para el caso LDCh

En esta sección se propone un método para analizar el efecto de un error del tipo LDCh en el algoritmo de Grover considerando una aproximación de primer orden (en α , como en [82]). Esta aproximación sirve tanto como una aproximación para valores suficientemente pequeños de α como una cota inferior de la probabilidad para cualquier valor de α .

Después de aplicar un error del tipo LDCh en cada qubit y en cada paso, ecuación (6.25), la probabilidad de éxito en el paso k es de la forma

$$\hat{p}^L(k, \alpha) = \sum_{i=0}^{nk} \left(1 - \frac{3\alpha}{4}\right)^{nk-i} \left(\frac{\alpha}{4}\right)^i f_i(n, k), \quad (6.35)$$

siendo i la cantidad total de errores hasta el paso k . Así, la aproximación de primer orden es

$$\hat{p}^L(k, \alpha) \approx \left(1 - \frac{3\alpha}{4}\right)^{nk} f_0(n, k) + \left(1 - \frac{3\alpha}{4}\right)^{nk-1} \left(\frac{\alpha}{4}\right) f_1(n, k), \quad (6.36)$$

en donde $f_0(n, k) \equiv p(k)$ es la probabilidad original del algoritmo de Grover sin error y $f_1(n, k)$ corresponde a todas las contribuciones que aplican un sólo error en cualquier qubit y en cualquier paso del algoritmo². Como se ilustra en la figura 6.7, la ecuación (6.36) es una cota inferior de la probabilidad debido a que todos los términos $f_i(n, k)$ son no negativos (suma de probabilidades).

²El cálculo de $f_1(n, k)$ se encuentra en *Ilan Cohn; ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García López de Lacalle. Grover's search with local and total depolarizing channel errors: complexity analysis. International Journal of Quantum Information. Vol. 14, No 2, 1650009. World Scientific. 2016.*

(a) Probabilidad evaluada en el paso k_{Gr} en función del ancho de error α , para $n = 8$ qubits and sus respectivas cotas.

(b) Detalle para α chicos.

Figura 6.7: Comparación entre las diferentes cotas de la probabilidad: simulación numérica (A), aproximación de primer orden (B), cota inferior considerando $\gamma_l(\alpha) = 1 - (1 - \alpha)^n$ (C), cota superior considerando $\gamma_u(\alpha) = n\alpha/(2 + n\alpha)$ (D) y cota superior considerando $\gamma_u(\alpha) = \alpha^n$ (E).

6.3.4. Estimación del orden

En esta sección se denota el costo del algoritmo con error del tipo LDCh como

$$MC_{k,\alpha}^L := MC^L[k, \hat{p}^L(k, \alpha)]. \quad (6.37)$$

Considerando que ambas $\hat{p}^L(k, \alpha)$ y $(\gamma_l - \gamma_u)$ son monótonas en α , y usando la ecuación (6.31), se tiene que

$$MC_{k,\gamma_u} \leq MC_{k,\alpha}^L \leq MC_{k,\gamma_l}. \quad (6.38)$$

O sea, dada una cantidad fija de pasos k , las cotas $\hat{p}(k, \gamma_u)$ y $\hat{p}(k, \gamma_l)$ inducen cotas en el costo.

Deteniendo el algoritmo en k_{Gr}

Considerando k_{Gr} como último paso, se tiene un costo medio dado por $MC_{k_{Gr},\alpha}^L$, acotado como se indica en la ecuación (6.38). En forma similar al caso TDCh, para todo $\alpha > 0$ fijo, el orden es $\Theta(N\sqrt{N})$.

Si se considera $N \gg 1$ y valores pequeños de α ($\alpha k_{Gr} \log_2 N \ll 1$) el costo medio es

$$MC_{k_{Gr}, \alpha}^L = \frac{\pi\sqrt{N}}{4} + \zeta \alpha N \pi^2 \log_2 N, \quad (6.39)$$

con ζ una constante en el intervalo $[1/32, 1/16]$. Utilizando la aproximación de primer orden, ecuación (6.36) y $f_1(n, k)$, se obtiene un valor de ζ de aproximadamente $70/2048$. Este valor es mucho más cercano a $1/32$ que a $1/16$, por lo que la probabilidad real se aproxima mejor cuando se usa γ_u que cuando se usa γ_l . Considerando la simplicidad y la aproximación lograda, se utiliza γ_u para la estimación de la probabilidad $\hat{p}^L(k, \alpha)$.

De la ecuación (6.39), se deduce que es necesario que se cumpla $\alpha \ll 2/(k_{Gr} \log_2 N)$ para mantener la mejora cuadrática.

Deteniendo el algoritmo en k_{max}

Considerando la continuidad de $\hat{p}(k, \gamma)$ respecto a γ y n , se puede determinar que las cotas inducidas para k_{max} son

$$k_{max_l} \leq k_{max}^L \leq k_{max_u}, \quad (6.40)$$

donde $k_{max_l} = k_{max}(\gamma_l)$ y $k_{max_u} = k_{max}(\gamma_u)$. Como siempre se detiene el algoritmo en el paso de máxima probabilidad, se cumplen las siguientes cotas

$$MC_{k_{max_l}, \gamma_u} \leq MC_{k_{max}^L, \alpha} \leq MC_{k_{max_u}, \gamma_l}. \quad (6.41)$$

De hecho, de las ecuaciones (6.41) y (6.28), se tiene que $k_{max}^L \rightarrow 1$ y el costo viene dado por $1/p^L(1, \alpha)$. Una vez más, para todo $\alpha > 0$ fijo, el orden es $\Theta(N)$.

Para $N \gg 1$ y valores de α tal que $0,55\alpha N \log_2 N \ll 1$, el costo es aproximadamente

$$MC_{k_{max}^L, \alpha}^L = \frac{\pi\sqrt{N}}{4} + \zeta \alpha N \log_2 N, \quad (6.42)$$

donde ζ está en el intervalo $[(\pi^2 - 4)/32, (\pi^2 - 1)/16]$. Para que el algoritmo se mantenga con el orden original $\Theta(\sqrt{N})$, es necesario que $\alpha \ll 2,5/(k_{Gr} \log_2 N)$.

Este resultado es coherente con los presentados para el modelo de error TDCh, donde para valores suficientemente pequeños de γ , la elección de detener en k_{Gr} o en k_{max} no altera el orden del algoritmo, sin embargo para cualquier valor de $\gamma > 0$ fijo, el orden cambia notablemente.

Análogamente a lo desarrollado en la sección 6.2.2, es posible hallar $\alpha_{clasico}$ considerando el máximo valor de α para el cual el costo es $N/2$. Si se toma el peor caso

$$\alpha_{clasico} \leq \frac{14}{\log_2 N}. \quad (6.43)$$

A diferencia del caso TDCh, este ancho (probabilidad de tener error) depende de N .

6.4. Interpretación de la degradación utilizando el índice de isotropía

En esta sección se estudia el efecto de la degradación del algoritmo de Grover, utilizando el índice de isotropía estadístico (sección 5.2).

6.4.1. Estado original del algoritmo sin ruido como estado de referencia

Si se considera como estado de referencia al estado original del algoritmo sin ruido $\rho(k)$, ecuación (6.3), es posible analizar cómo se aparta la evolución del estado con error en función de la evolución teórica.

Algoritmo con error del tipo TDCh

La ecuación (6.7) indica la matriz densidad obtenida luego de k pasos, donde $\rho(k)$ es la matriz densidad del algoritmo de Grover sin ruido dada por la ecuación (6.3). Por lo tanto, el ancho isótropo vale $p = (1 - (1 - \gamma)^k)$ y dado que se utiliza como estado de referencia $\rho(k)$, la alineación isótropa es $A = 1$ (para todos los pasos). Este hecho se ilustra en la figura 6.8.

Figura 6.8: Variación del índice de isotropía para diferentes valores de γ : 0 (negro); 0,025 (azul); 0,05 (verde claro); 0,1 (celeste); 0,15 (rojo); 0,2 (magenta); 0,5 (verde oscuro). Errores del tipo TDCh y estado $\rho(k)$ como referencia.

Algoritmo con error del tipo LDCh

Como se ha visto en la sección 6.3, cuando se considera el modelo de error LDCh el resultado de la búsqueda empeora notablemente. En este caso, incluso para valores pequeños de ancho isótropo p , el estado muestra una clara desviación hacia el estado ortogonal ρ_{i0} (la alineación se desvía hacia el valor -1), como puede verse en la figura 6.9. Este hecho redundará en una degradación de la probabilidad de éxito.

Esto explica la razón por la cual existe una mayor degradación del algoritmo utilizando el error modelado como LDCh respecto del modelado como TDCh, ya que en éste último la alineación no varía. Este tipo de error local desvía el estado sacándolo del plano original del algoritmo (figura 6.1).

6.4.2. Estado marcado como estado de referencia

Si se considera como referencia al elemento marcado ($\rho_0 \equiv \rho_t = |t\rangle\langle t|$) es posible interpretar el funcionamiento del algoritmo como la corrección de un estado original con error, gradualmente corregido hacia el elemento marcado.

Algoritmo con error del tipo TDCh

En este caso el índice de isotropía puede ser hallado en forma analítica. Utilizando la ecuación (6.7), se puede observar que el ancho isótropo es $p = 1 - (1 - \gamma)^k$, igual al calculado en 6.4.1. Como el estado ρ_k es puro, la alineación es

$$A = \sqrt{\langle s_k | \rho_0 | s_k \rangle} - \sqrt{\langle s_k | \rho_{i0} | s_k \rangle}, \quad (6.44)$$

donde $|s_k\rangle$ está definido en la ecuación (6.4). Luego, la alineación vale

$$A = |\sin((2k+1)\theta)| - \frac{|\cos((2k+1)\theta)|}{\sqrt{2^n - 1}}, \quad (6.45)$$

que es idénticamente nula para $k = 0$, no negativa (para $0 \leq k \leq k_{Gr}$) y tiende a 1 con $n \rightarrow \infty$ cuando $k = k_{Gr}$.

 (a) Triángulo de isotropía (alineación $\times$ ancho) para diferentes valores de α .

 (b) Variación del *ancho isotrópico* (p) en función del paso k para diferentes valores de α .

 (c) Variación de la *alineación isotrópica* (p) en función del paso k para diferentes valores de α .

 Figura 6.9: Variación del índice de isotropía para diferentes valores de α : 0 (negro); 0,025 (azul); 0,05 (verde claro); 0,1 (celeste); 0,15 (rojo); 0,2 (magenta); 0,5 (verde oscuro). Errores del tipo LDCh y estado $\rho(k)$ como referencia.

La figura 6.10 ilustra las variaciones de los índices en función del paso k para valores de $\gamma \in \{0, 0,025, 0,05, 0,1, 0,15, 0,2, 0,5\}$. En el caso del algoritmo original sin error ($\gamma = 0$), se puede observar como la alineación inicial sale del valor 0 y tiende a 1 (siempre con $p = 0$). Esto se puede interpretar como un estado con error que se corrige con la aplicación del algoritmo. En la figura 6.10(c) se nota que la alineación no depende de γ , como se indica en la ecuación (6.45).

Algoritmo con error del tipo LDCh

Como el ancho isotrópico p no depende del estado de referencia, éste es igual al visto anteriormente en la figura 6.9(b). El índice de isotropía se puede observar en la Fig. 6.11.

La figura 6.11(a) muestra que aún con un error muy pequeño ($\alpha = 0,025$, figura 6.11(b)) la tendencia a alinearse con el estado marcado desaparece. Esto significa que hay una rápida degradación del algoritmo, ya que el estado tiene una tendencia hacia el estado ortogonal ρ_{i0} . Para valores mayores de ancho de ruido α , el estado tiende rápidamente a acercarse al estado I/N .

(a) Triángulo de isotropía (alineación $\times$ ancho) para diferentes valores de γ .

(b) Variación del *ancho isotrópico* (p) en función del paso k para diferentes valores de γ .

(c) Variación de la *alineación isotrópica* (p) en función del paso k para diferentes valores de γ .

Figura 6.10: Variación del índice de isotropía para diferentes valores de γ : 0 (negro); 0,025 (azul); 0,05 (verde claro); 0,1 (celeste); 0,15 (rojo); 0,2 (magenta); 0,5 (verde oscuro). Errores del tipo TDCh y estado marcado (ρ_t) como referencia.

(a) Triángulo de isotropía (alineación $\times$ ancho) para diferentes valores de α .

(b) Detalle del triángulo de isotropía.

(c) Variación del *ancho isotrópico* (p) en función del paso k para diferentes valores de α .

(d) Variación de la *alineación isotrópica* (p) en función del paso k para diferentes valores de α .

Figura 6.11: Variación del índice de isotropía para diferentes valores de α : 0 (negro); 0,025 (azul); 0,05 (verde claro); 0,1 (celeste); 0,15 (rojo); 0,2 (magenta); 0,5 (verde oscuro). Errores del tipo LDCh y estado marcado (ρ_t) como referencia.

Capítulo 7

Caminatas cuánticas

Aunque actualmente es un tema investigación, la hipótesis más aceptada sobre la ventaja de la computación cuántica sobre la clásica se debe principalmente a la correlación cuántica (entrelazamiento, discordia cuántica, no localidad, contextualidad entre otras) [114], aunque no está clara la relación entre la correlación y el éxito del resultado [36][37].

Por su parte, la correlación cuántica se ve afectada por la medición además de la interacción con el ambiente, generalmente modelada como ruido [82]. Debido a su importancia en obtener resultados prácticos, esta influencia ha sido estudiada extensivamente en los últimos años [95][3][109][51]. Además, algunas investigaciones han sido realizadas para determinar la influencia de mediciones proyectivas parciales [113][78].

En general se cree que la decoherencia degrada la eficiencia de un algoritmo cuántico. Sin embargo, se ha descubierto que se puede mejorar la eficiencia de ciertos algoritmos si se considera la evolución del sistema como un sistema abierto con bajos niveles de ruido [70][116]. Debido a esto es interesante investigar si al considerar la evolución abierta del sistema ocasionada por mediciones parciales intermedias puede mejorar el desempeño del algoritmo cuántico.

En este capítulo se propone estudiar el efecto de mediciones parciales en un algoritmo de caminata cuántica controlada. El algoritmo estudiado es una propuesta intermedia entre dos métodos conocidos: algoritmos cuánticos unitarios, con un estado inicial puro y separable [88], donde la correlación aumenta durante una etapa de la evolución; y algoritmos cuánticos con mediciones (*Measurement Based Quantum Computation (MBQC)*-[92]), con un estado inicial con alta correlación (*cluster states*), donde se toman mediciones para obtener el resultado requerido. Este algoritmo se basa en el algoritmo de Tulsi [112].

7.1. Algoritmo de búsqueda de Tulsi

El algoritmo de Tulsi es un algoritmo de búsqueda basado en los principios de una caminata cuántica controlada, y resuelve el problema de hallar un único elemento marcado de una colección desordenada de N elementos. La posición está representada por un espacio generado por un estado cuántico de n qubits ($N = 2^n$), y los N elementos son representados mediante una distribución en una red bidimensional ($\sqrt{N} \times \sqrt{N}$). Además del espacio posición, el estado completo tiene dos qubits que hacen de moneda cuántica, que determina la dirección de movimiento sobre la grilla, y un qubit de control.

El algoritmo está constituido por dos operadores: el oráculo y la caminata cuántica condicional. El oráculo es una operación de reflexión con respecto al objetivo (elemento marcado) dado por¹

$$O = I - 2 |\delta_{ctr, u_c, t}\rangle \langle \delta_{ctr, u_c, t}| \quad (7.1)$$

donde

¹los subíndices *ctr*, *c* y *p*, indican los subespacios del control, de la moneda y de la posición, respectivamente.

- $|\delta_{ctr}\rangle$ es un operador en el espacio del qubit de control y vale $|\delta_{ctr}\rangle = -\sin \delta|0\rangle + \cos \delta|1\rangle$ (dependiente de un parámetro δ),
- $|u_c\rangle$ es un operador en el espacio de la moneda y coincide con el estado inicial de la moneda $|u_{c0}\rangle$, dado por una superposición balanceada de dos qubits $\{|i_c\rangle\} = \{\leftarrow, \rightarrow, \downarrow, \uparrow\} = \{00, 01, 10, 11\}$

$$|u_{c0}\rangle = \frac{1}{2} \sum_{i_c=0}^3 |i_c\rangle, \quad (7.2)$$

- y el estado marcado $|t\rangle$ que es un elemento de la base canónica.

El operador de la caminata condicional hace que el estado evolucione dependiendo del valor del control, según

$$W = |1\rangle\langle 1| \otimes (S \cdot C) - |0\rangle\langle 0| \otimes I_{c,p} \quad (7.3)$$

siendo

- S el operador de desplazamiento

$$S = \sum_{x,y} |\leftarrow, x+1, y\rangle\langle \rightarrow, x, y| + |\downarrow, x, y+1\rangle\langle \uparrow, x, y| + |\rightarrow, x-1, y\rangle\langle \leftarrow, x, y| + |\uparrow, x, y-1\rangle\langle \downarrow, x, y| \quad (7.4)$$

en donde el estado $|x, y\rangle$ denota la posición x, y en la grilla,

- y $C = C_0 \otimes I_p$, siendo C_0 el operador que opera en el subespacio de la moneda

$$C_0 = -I_c + 2|u_c\rangle\langle u_c|. \quad (7.5)$$

El estado inicial completo $|\psi_0\rangle$ ($ctr \otimes c \otimes p$) viene dado por

$$|\psi_0\rangle = |1, u_c, u_p\rangle = |1\rangle \otimes |u_c\rangle \otimes |u_p\rangle = |1\rangle \otimes |u_c\rangle \otimes \frac{1}{\sqrt{N}} \sum_{i_p=0}^{N-1} |i_p\rangle, \quad (7.6)$$

en donde se nota que el subespacio de la posición se inicializa con una combinación lineal balanceada de la base canónica, similar a los algoritmos originales de Grover (sección 6.1) y la caminata cuántica de AKR [8]. Partiendo de este estado y aplicando $T_\delta = [\frac{\pi}{4} \sqrt{N(\log N + \tan^2 \delta/4)}]$ veces el operador $U = W \cdot O$ (figura 7.1), el estado se acerca al estado marcado. El resultado es que la distancia entre el estado final de la evolución unitaria y el estado marcado depende de δ y N . Se si elige el parámetro δ con el valor $\cos(\delta) = 1/\log N$, la probabilidad de obtener el estado marcado, luego de la medición, queda independiente de N , por lo que el algoritmo tiene un orden de complejidad $O(\sqrt{N \log N})$.

Figura 7.1: Esquema del algoritmo de Tulsi.

En la figura 7.2 se ilustra el resultado de la probabilidad de obtener el estado marcado considerando una grilla $2^9 \times 2^9$ ($N = 2^{18}$) y tres diferentes elecciones de parámetro δ ($\delta = 0$, $\delta = \pi/4$ y el parámetro de Tulsi's $\delta = \arccos(1/\sqrt{\log N})$).

Figura 7.2: Probabilidad del elemento marcado en el algoritmo de Tulsi con $N = 2^{18}$ para distintos valores de δ : $\delta = 0$ (AKR línea de puntos), $\delta = \pi/4$ (línea de a trazos) y $\delta = \arccos(\sqrt{1/\log N})$ (línea sólida).

7.2. Algoritmo de búsquedas cuánticas con mediciones intermedias (IMA)

A diferencia del algoritmo unitario de Tulsi, en esta sección se define un algoritmo en el cual se efectúan mediciones proyectivas del qubit de control con elección de salida (figura 7.3), entre intervalos de evolución unitaria de l pasos (a l se le denomina *lapso*). La finalidad de esta propuesta es estudiar el efecto de las mediciones proyectivas parciales sobre la probabilidad de encontrar el estado marcado, y su relación con la correlación total del estado, analizando los cambios de desempeño y la robustez frente a ruido.

Figura 7.3: Esquema del Algoritmo con Mediciones Intermedias (IMA).

La información que se obtiene, después de cada medición (cada l pasos), permite terminar el algoritmo si el resultado es 0, ya que este valor del control está relacionado únicamente con el estado marcado en el subespacio de la posición. Esto es claro si se analiza la secuencia de operadores (7.1) y (7.3) a un estado de la base canónica $|i_{ctr}, i_c, i_p\rangle$. Cuando $|i_p\rangle$ es una posición cualquiera diferente del estado marcado se obtiene

$$|1, i_c, i_p\rangle \xrightarrow{O} |1, i_c, i_p\rangle \xrightarrow{W} \begin{cases} |1, i_c', i_p'\rangle \\ |1, i_c', t\rangle \end{cases} \quad (7.7)$$

En el caso que en el subespacio de la posición se encuentre el estado marcado se tiene que

$$|1, i_c, t\rangle \xrightarrow{O} \begin{cases} |1, i_c', t\rangle \\ |0, i_c', t\rangle \end{cases} \xrightarrow{W} \begin{cases} |1, i_c'', i_p\rangle \\ |0, i_c', t\rangle \end{cases}$$

Algoritmo 2: Algoritmo de búsqueda con caminatas cuánticas con mediciones intermedias

1. El sistema es inicializado en el estado inicial $|\psi_0\rangle$ (ecuación (7.6)).
 2. Se aplica l veces (pasos) el operador $U = WO$,
 3. Se mide el qubit de control.
 4. Si el resultado de la medición es 0 se detiene el algoritmo: el elemento marcado ha sido hallado. Sino, se retorna al paso 2 hasta un máximo de k_{max} pasos totales.
 5. Después de k_{max} pasos, se verifica si el subespacio de la posición corresponde al elemento marcado. En caso de no ser cierto, se empieza una vez más todo el algoritmo (del paso 1).
-

$$|0, i_c, t\rangle \xrightarrow{O} \begin{cases} |0, i_c', t\rangle \\ |1, i_c', t\rangle \end{cases} \xrightarrow{W} \begin{cases} |0, i_c', t\rangle \\ |1, i_c'', i_p\rangle \end{cases} \quad (7.8)$$

De aquí se concluye que iniciando con el estado dado en (7.6), ningún estado del tipo $|0, i_c, i_p\rangle$ puede aparecer durante la evolución si $|i_p\rangle$ es diferente del estado marcado $|t\rangle$.

Al igual que en algoritmo original de Tulsi, se considera la cantidad máxima de pasos igual a $k_{max} = (\pi/4)(\sqrt{N \log N})$, ya que este valor es aproximadamente óptimo (sección 7.3).

Como el algoritmo no es completamente unitario, en lugar de amplificación cuántica [20] se utiliza amplificación clásica (probabilidad acumulada) para lograr obtener probabilidades cercanas a uno. La probabilidad acumulada, P_c , de obtener el estado marcado debido a los múltiples intentos para cualquier lapso l ($l = 1, 2, \dots, k_{max}$) viene dada por

$$\begin{aligned} P_c(k, l) &= P_0^l + P_1^l P_0^{2l} + P_1^l P_1^{2l} P_0^{3l} + \dots + P_1^l \dots P_1^{\lfloor k/l \rfloor l} P_t(k, l) \\ &= P_0^l + (1 - P_0^l) P_0^{2l} + \dots + (1 - P_0^l) (1 - P_0^{2l}) \dots (1 - P_0^{\lfloor k/l \rfloor l}) P_t(k, l) \\ &= P_0^l + (1 - P_0^l) \left[P_0^{2l} + (1 - P_0^{2l}) \left[\dots \left[P_0^{\lfloor k/l \rfloor l} + (1 - P_0^{\lfloor k/l \rfloor l}) P_t(k, l) \right] \right] \right], \end{aligned} \quad (7.9)$$

siendo

- $P_t(k, l) = |\langle \psi | (I_{ctr} \otimes I_c \otimes |t\rangle\langle t|) | \psi \rangle|^2$ la probabilidad del estado marcado en el paso k ,
- P_0^{ml} y P_1^{ml} corresponden a las probabilidades de obtener, en un paso múltiple de l , los valores 0 y 1 en la medición del control (respectivamente),
- y $\lfloor k/l \rfloor$ es la parte entera de k/l .

Como todo término entre paréntesis que aparece en (7.9) tiene la forma $x + (1-x)a$ ($0 \leq x, a \leq 1$), $P_c(k, l)$ siempre se encuentra en el intervalo $[0, 1]$.

A continuación se estudia dos casos de interés: $\delta = \pi/4$, ($IMA_{\pi/4}$) and $\delta = \arccos(1/\sqrt{\log N})$, (IMA_T). En el caso de $\delta = 0$ el algoritmo es idéntico a la caminata AKR , y el control no afecta la búsqueda.

7.3. Estudio de probabilidades y correlaciones (IMA)

A diferencia de la evolución unitaria de Tulsi, la probabilidad de encontrar el elemento marcado varía drásticamente cuando se realizan las mediciones. A continuación se calcula la probabilidad del elemento marcado P_t y la probabilidad acumulada P_c comparando sus evoluciones en cada paso con diferentes correlaciones del estado cuántico. Ambas probabilidades dependen del lapso de tiempo l de evolución unitaria entre mediciones. En los experimentos numéricos se elige l en función del número total de estados N ($N = 2^n$, n qubits),

$$l = \sqrt{N}/2^m, \quad m = 1, 2, 3, \dots \quad (7.10)$$

Para poder comparar con el caso clásico se usa también el valor $l = 1$ en algunos experimentos.

Los resultados numéricos evidencian que para el algoritmo IMA_T , que usa el parámetro δ original de Tulsi ($\delta = \arccos(1/\sqrt{\log N})$), las mediciones intermedias reducen la probabilidad acumulada para cualquier lapso l , como se muestra en la figura 7.4. Desde este punto de vista el algoritmo original de Tulsi es óptimo, o sea, no puede ser mejorado con mediciones de este tipo.

Figura 7.4: Probabilidad acumulada para el algoritmo IMA_T para $N = 2^{18}$ y diferentes valores de $l : 2\sqrt{N}, \sqrt{N}, \sqrt{N}/2, \sqrt{N}/4, \sqrt{N}/8$. La línea de a trazos indica la evolución para el caso unitario.

Por lo contrario, si se considera un diferente elección del parámetro δ , como por ejemplo en algoritmo $IMA_{\pi/4}$ ($\delta = \pi/4$)², la probabilidad acumulada puede ser mejorada dependiendo del lapso l usado.

La probabilidad del elemento marcado en el algoritmo unitario P_t se comporta similar a la energía de un oscilador armónico [112] para cualquier δ . Como es de esperar, las mediciones sucesivas del qubit de control generan una pérdida de coherencia debido al hecho de que el control tiene correlación con el resto del estado (moneda/posición). Este hecho juntamente con que el algoritmo termina cuando se mide cero en el control, hace que la probabilidad P_t tienda a ser muy baja en la evolución (para N grandes), similar a lo que ocurre con la energía de un oscilador con disipación (figura 7.5).

Figura 7.5: Probabilidad del elemento marcado, P_t , para el algoritmo $IMA_{\pi/4}$ ($\delta = \pi/4$) con $N = 2^{18}$. Se consideran los siguientes lapsos: $l = \sqrt{N}/4$ (gris), $l = \sqrt{N}/16$ (negro) y el caso de evolución unitaria (línea a trazos).

²Esta elección del parámetro δ fue tomada porque el operador 7.1 queda simétrico.

Esto último hace que la probabilidad acumulada, P_c , tienda a un valor aproximadamente constante a tiempos largos (N grandes), como se ilustra en la figura 7.6.

Figura 7.6: Probabilidad acumulada, P_c , para el algoritmo $IMA_{\pi/4}$ ($\delta = \pi/4$) con $N = 2^{18}$. Se consideran los siguientes lapsos: $l = \sqrt{N}/2$ (gris claro), $l = \sqrt{N}/4$ (gris), $l = \sqrt{N}/16$ (negro), $l = 1$ (línea punteada) y el caso de evolución unitaria (línea a trazos).

7.3.1. Correlaciones

En años recientes varios autores han tenido el interés de determinar la relación entre el desempeño del algoritmo y las diferentes correlaciones en algoritmos de búsqueda cuántica [36]. En esta sección se analiza el desempeño, considerando la relación entre las probabilidades P_c y P_t y ciertas medidas de correlación como, la información mutua bipartita [115] (MI) y la medida de correlación cumulativa CCM (sección 3.2). La mayor ventaja de esta última medida de correlación (CCM) es que para calcularla no se necesita de un método de optimización no lineal. Esto es importante a la hora de estudiar un algoritmo de búsqueda cuántica con un número de estados N muy grande.

El algoritmo IMA tiene particiones naturales del espacio compuestas por el control (ctr), la moneda (c) y la posición (p). La correlación bipartita de la partición del espacio $\mathcal{H}_{control} \otimes \mathcal{H}_{moneda+posicion}$, denominada $MI_{ctr \otimes (c,p)}$, viene dada por

$$MI_{ctr \otimes (c,p)}(\rho) = -S(\rho) + S(\rho_{ctr}) + S(\rho_{c,p}) \quad (7.11)$$

siendo ρ la matriz de densidad del estado, ρ_{ctr} la matriz reducida del qubit de control, $\rho_{c,p}$ es la matriz reducida del subespacio moneda/posición y S es la entropía cuántica de Von Neumann [82]. En forma similar se pueden calcular otras correlaciones bipartitas como $MI_{(ctr,c) \otimes p}$ y $MI_{(p,ctr) \otimes c}$.

Luego de conocer el resultado de cada medición, el estado ρ se conserva puro, por lo que $S(\rho) = 0$. Luego, no hay diferencia entre la correlación cuántica y la correlación total (que incluye la correlación clásica) del estado, y estas medidas de correlación se pueden considerar medidas de entrelazamiento [136].

A continuación también serán consideradas otras medidas para correlaciones de estados mezcla, como la MI entre la moneda y la posición ($MI_{c \otimes p}$), el control y la moneda ($MI_{ctr \otimes c}$), y el control y la posición ($MI_{ctr \otimes p}$). Como ejemplo, la correlación $MI_{c \otimes p}$ viene dada por

$$MI_{c \otimes p}(\rho) = -S(\rho_{c,p}) + S(\rho_c) + S(\rho_p), \quad (7.12)$$

donde, en este caso, $S(\rho_{c,p}) \neq 0$.

7.3.2. P_t y correlaciones

Debido a la correlación impuesta por el oráculo (7.1), la probabilidad P_t muestra un comportamiento similar a las correlaciones que separan el qubit de control: $MI_{ctr \otimes (c,p)}$, $MI_{ctr \otimes p}$ y $MI_{ctr \otimes c}$.

Estas correlaciones tienen un valor nulo inmediatamente después de la medición del control como se ve en el detalle en la figura 7.7. La fluctuaciones que aparecen después de la medición, tanto en la probabilidad como en las correlaciones, pueden ser explicadas como ondas secundarias que aparecen comúnmente en las caminatas cuánticas [71]. Se puede apreciar del estudio, que estas fluctuaciones se vuelven pequeñas con el crecimiento del número de estados N . Con la intención de obtener una media, las curvas son suavizadas tomando un promedio para cada caso.

Figura 7.7: Probabilidad del elemento marcado (línea negra continua), P_t , comparada con la evolución de las correlaciones $MI_{ctr \otimes (c,p)}$ (línea gris a trazos), $MI_{ctr \otimes p}$ (línea gris punteada) and $MI_{ctr \otimes c}$ (línea gris con puntos y rayas). Simulación para $N = 2^{16}$ y $l = \sqrt{N}/2$, con las curvas normalizadas y suavizadas. En el detalle las curvas originales sin suavizar.

La figura 7.8 ilustra la evolución de la probabilidad de obtener el estado $|1\rangle$ en el qubit de control luego de la medición. Como se observa, con el crecimiento del número de pasos, el qubit de control $\rho_{(ctr)}$ tiende al estado puro $|1\rangle$, y se tiene que $S(\rho_{(ctr)}) \approx 0$.

Figura 7.8: Probabilidad de obtener el estado $|1\rangle$ en el qubit de control. Estudio para $N = 2^{16}$ y $l = \sqrt{N}/2$.

Si se considera la ecuación (7.11) y que ρ es un estado puro entonces

$$MI_{ctr \otimes (c,p)} = 2S(\rho_{ctr}) \approx 0. \quad (7.13)$$

Para las otras correlaciones se tiene que

$$\begin{aligned} MI_{ctr \otimes p} &= -S(\rho_{ctr,p}) + S(\rho_{ctr}) + S(\rho_p), \\ &\approx -S(\rho_{ctr,p}) + S(\rho_p), \\ &\approx 0. \end{aligned} \quad (7.14)$$

Análogamente, $MI_{ctr \otimes c} \approx 0$.

En el caso de realizar la medición del control en todos los pasos, $l = 1$, la correlaciones bipartitas $MI_{ctr \otimes (c,p)}$, $MI_{ctr \otimes p}$ y $MI_{ctr \otimes c}$ son nulas, por lo que la probabilidad P_t es muy baja y P_c se vuelve casi lineal, como se muestra en la Figura 7.6.

7.3.3. P_c y correlaciones

Del estudio, la probabilidad acumulada P_c se comporta en forma similar a las correlaciones $MI_{(p,ctr) \otimes c}$, $MI_{(ctr,c) \otimes p}$ y $MI_{c \otimes p}$. Estas correlaciones oscilan en el tiempo (pasos de evolución) hasta que se estabilizan ($\pm 5\%$) en aproximadamente $5k_{max}$ pasos, como se muestra en la figura 7.9. Este comportamiento es esperado, ya que con el crecimiento del número de pasos, baja la influencia del control (7.13). Luego, para un número grande de pasos

$$MI_{(p,ctr) \otimes c} = -S(\rho) + S(\rho_{p,ctr}) + S(\rho_c) \approx S(\rho_p) + S(\rho_c), \quad (7.15)$$

$$MI_{(ctr,c) \otimes p} = -S(\rho) + S(\rho_{ctr,c}) + S(\rho_p) \approx S(\rho_p) + S(\rho_c), \quad (7.16)$$

$$\begin{aligned} MI_{c \otimes p} &= -S(\rho_{c,p}) + S(\rho_p) + S(\rho_c) = -S(\rho_{ctr}) + S(\rho_p) + S(\rho_c) \\ &\approx S(\rho_p) + S(\rho_c). \end{aligned} \quad (7.17)$$

(a) Correlaciones $MI_{(p,ctr) \otimes c}$ (gris oscuro), $MI_{(ctr,c) \otimes p}$ (negro) y $MI_{c \otimes p}$ (gris claro). A medida que crece el número de pasos todas las correlaciones tienden a ser similares.

(b) Probabilidad acumulada del elemento marcado P_c

Figura 7.9: Probabilidad acumulada del elemento marcado, P_c , comparada con la evolución de las correlaciones. Estudio para $N = 2^{16}$ y $l = \sqrt{N}/2$.

Para el caso del algoritmo unitario original las correlaciones estudiadas tienen comportamientos similares. Se tiene un mínimo relativo cuando la probabilidad del elemento marcado llega al máximo, como se observa en la figura (7.10). Esto se debe a que el subespacio de la posición se acerca al estado marcado, quedando prácticamente separable del resto, o sea, poco correlacionado con el resto del estado cuando la probabilidad se vuelve máxima.

Este comportamiento del algoritmo unitario ha sido observado en otros algoritmos de búsqueda, como el algoritmo de Grover (capítulo 6). En [36] se determina que la correlación medida con la *concurancia* [132] es un indicador del crecimiento de probabilidad del elemento marcado. A diferencia del algoritmo unitario $IMA_{\pi/4}$, en el algoritmo de Grover la probabilidad crece a valores cercanos a uno, al mismo tiempo que la correlación decrece a aproximadamente cero.

Figura 7.10: Probabilidad del elemento marcado, P_t , comparada con las correlaciones: $MI_{(p,ctr) \otimes c}$ (línea gris a trazos), $MI_{(ctr,c) \otimes p}$ (línea gris de trazos y puntos) and $MI_{c \otimes p}$ (línea negra punteada). Estudio para el caso $N = 2^{16}$ y evolución unitaria (curvas normalizadas).

7.3.4. Número de pasos y correlación

Si se quiere una probabilidad de éxito alta, $P \approx 1$, es necesaria la realización de múltiples experimentos para poder hacer una amplificación clásica de la probabilidad. Dado un experimento con probabilidad P_0 , la cantidad de repeticiones independientes R necesarias para obtener una probabilidad P se calcula (distribución geométrica) del siguiente modo:

$$\begin{aligned} P &= 1 - (1 - P_0)^R, \\ \Rightarrow R &= \frac{\log 1 - P}{\log 1 - P_0} \end{aligned} \quad (7.18)$$

Considerando que cada experimento tiene k_{max} pasos y la probabilidad de éxito es igual a P_c , el número total de pasos TS es

$$TS = k_{max} \frac{\log(1 - P)}{\log(1 - P_c)}. \quad (7.19)$$

En el algoritmo unitario se tiene que $k_{max} = (\pi/4)\sqrt{N \log(N)}$. En el caso del algoritmo *IMA*, es interesante comparar este último valor con el valor óptimo que minimiza el número total de pasos TS . Es un echo interesante que este valor óptimo de pasos esta relacionado con la correlación del subespacio $\mathcal{H}_{ctr \otimes c}$, por lo que puede ser usado como indicador de este punto.

La figura 7.11 muestra las curvas de la información mutua, $MI_{ctr \otimes c}$, de la medida de correlación cumulativa³ (sección 3.2), $CCM(\rho_{ctr,c})$, y la curva de estimación del orden e , donde

$$e(k) = \frac{1}{k} \frac{\log(1 - P_c)}{\log(1 - P)}, \quad (7.20)$$

siendo $TS = 1/e(k_{max})$.

Finalmente, en la figura 7.12 se ilustran ambas curvas, $e(k_{max})$ y $MI_{c \otimes p}(k_{max})$, en función del lapso l , siendo cada una evaluada para valores de N fijos ($N = 2^{14}$, $N = 2^{16}$) y dos diferentes k_{max} (el óptimo y $\pi/4\sqrt{N \log N}$). Ambas curvas presentan un máximo para $l = \sqrt{N}/4$ ($m = 4$). Es interesante observar que los máximos de la correlación $MI_{c \otimes p}(k_{max})$ y del orden $e(k)$ ocurren para el mismo valor del lapso $l = \sqrt{N}/4$.

Como se puede observar, utilizar como k_{max} el valor óptimo o $k_{max} = \pi/4\sqrt{N \log N}$ dan resultados similares, por lo que se utiliza este último como aproximación del óptimo.

³En las gráficas de la figura 7.11, aparece la denominación original *CCI* (*Cumulative Correlation Index*).

Figura 7.11: Comparación entre correlaciones y la estimación del orden. CCM (línea negra a trazos), $MI_{ctr\otimes c}$ (línea gris punteada), y $e(k)$ (7.20), en función del número de pasos. Todas las curvas están normalizadas y suavizadas.

7.4. Estimación del orden

El orden de un algoritmo de búsqueda puede ser estimado por la cantidad de pasos necesarios para tener una probabilidad fija, como por ejemplo una probabilidad cercana a 1. En el caso del algoritmo AKR cuando se utiliza amplificación cuántica [24] se tiene un orden de $O(\sqrt{N} \log N)$, mientras con amplificación clásica el orden es de $O(\sqrt{N} \log^{\frac{3}{2}} N)$ [1].

Para el algoritmo de Tulsi, la probabilidad del elemento marcado es cercana a $1/\cos \delta \sqrt{\log N}$, resultando en un orden de $O(\sqrt{N} \log N)$. Para el algoritmo $IMA_{\pi/4}$ unitario (un caso particular del algoritmo de Tulsi con $\delta = \pi/4$) si se utiliza amplificación clásica el orden es idéntico al del algoritmo AKR , o sea, $O(\sqrt{N} \log^{3/2} N)$.

Considerando los resultados anteriores, en donde se ha observado que la cantidad total de pasos TS necesarios para obtener una probabilidad fija P varía con el lapso de tiempo de medición l , a continuación son estimadas cotas para el orden del algoritmo $IMA_{\pi/4}$ considerando algunos valores de l .

La figura 7.13 ilustra el número total de pasos TS en función de N , para algunos valores de l . Como es posible observar en la figura 7.13(c), cuando se divide TS por N la curva correspondiente al valor de lapso $l = 1$ es la única que converge a un valor constante. En este caso el algoritmo tiene el mismo orden que el algoritmo clásico de *fuerza bruta*, mientras las otras curvas indican un orden mejor que el del clásico.

Debido a limitaciones en los recursos necesarios de cómputo para valores grandes de N (la necesidad de memoria crece exponencialmente), es difícil realizar regresiones no lineales en las curvas

(a) Número total de pasos (como TS^{-1}) en función del lapso l ($m = \sqrt{N}/l$).

(b) Correlación $MI_{c \otimes p}$ en función del lapso l ($m = \sqrt{N}/l$).

Figura 7.12: Comparación entre TS^{-1} ($e(k_{max})$) y la correlación $MI_{c \otimes p}$, para $N = 2^{14}$ (cuadrado), $N = 2^{16}$ (círculo). Ambas TS y la correlación están suavizadas y evaluadas en el paso óptimo (línea gris a trazos), y en $k_{max} = \pi/4\sqrt{N \log N}$ (línea continua negra).

para estimar el orden. En el estudio a continuación se estima rangos para el orden en función del lapso l , mediante el análisis de la división de las curvas en la figura 7.13(b) por funciones del tipo

$$\beta \sqrt{N} (\log N)^\alpha, \quad (7.21)$$

considerando diferentes valores para α . La figura 7.14 ilustra los resultados para valores de α iguales a 1,5, 1,25, 0,9 y 0,6. Es posible observar que:

- Como se espera, la curva correspondiente al algoritmo unitario tiende a ser constante cuando se divide por la ecuación (7.21) con un valor de $\alpha = 1,5$. Esto se puede observar en la figura 7.14(a).
- En los casos de $l = \sqrt{N}$ y $l = \sqrt{N}/2$, el orden está en el rango $\alpha \in [0,9, 1,25]$, como se observa en las figuras 7.14(b) y 7.14(c).
- En los casos de $l = \sqrt{N}/4$ y $l = \sqrt{N}/8$, de las figuras 7.14(c) y 7.14(d) es posible estimar que el orden se encuentra en el rango de $\alpha \in [0,6, 0,9]$.

(a) TS en función de N . Curva con línea sólida negra con asteriscos para el caso $l = 1$ (mediciones en cada paso).

(b) Ampliación de las curvas para la evolución unitaria (sólida negra), $l = \sqrt{N}$ (sólida gris), $l = \sqrt{N}/2$ (trazos-puntos grises), $l = \sqrt{N}/4$ (trazos grises) y $l = \sqrt{N}/8$ (puntos negros). Las últimas dos curvas se superponen.

(c) TS dividido por N . La curva para $l = 1$ es la única que converge a un valor constante.

Figura 7.13: Cantidad total de pasos TS en función de N . Curvas para los casos de evolución unitaria y con mediciones con lapsos $l = 1, \sqrt{N}, \sqrt{N}/2, \sqrt{N}/4$, y $\sqrt{N}/8$.

Figura 7.14: En las curvas se estima el orden utilizando la cantidad total de pasos TS dividida por las curvas dadas por la ecuación (7.21) para diferentes valores de α . Caso unitario (sólida negra), $l = \sqrt{N}$ (sólida gris), $l = \sqrt{N}/2$ (trazos-puntos grises), $l = \sqrt{N}/4$ (trazos grises) and $l = \sqrt{N}/8$ (puntos negros).

Parte IV

Conclusiones

Capítulo 8

Conclusiones

8.1. Conclusiones

El conocimiento y análisis de las limitaciones en los sistemas cuánticos es un tema fundamental para el desarrollo eficiente de algoritmos cuánticos, siendo la propagación de errores, debido a diversas causas, uno de los principales factores en la pérdida de desempeño. En ésta tesis se describen algunas herramientas que permiten:

- analizar en forma cualitativa, y cuantitativa, esta disminución del desempeño,
- caracterizar las limitaciones debidas a la propagación de errores en los diferentes algoritmos,
- analizar el efecto de realizar observaciones parciales en algoritmos cuánticos (algoritmos con sistemas parcialmente abiertos), para mejorar su desempeño o robustez,
- simular, en una computadora clásica utilizando un ambiente de cálculo numérico abierto, circuitos cuánticos con errores.

A continuación se realiza un resumen de las conclusiones de las herramientas propuestas y los ejemplos estudiados.

8.1.1. Correlaciones cuánticas

En la sección 3.1 se ha utilizado un algoritmo híbrido que permite una regresión eficiente, con la finalidad del cálculo del entrelazamiento en estados puros de dimensiones grandes. Las principales fortalezas del algoritmo son lograr en forma algebraica puntos cercanos para el inicio de la regresión no lineal, cubrir la búsqueda de todo el espacio mediante meta-heurísticas (algoritmos genéticos) y la aceleración local mediante el uso del gradiente. La implementación de este algoritmo también sirvió para demostrar la flexibilidad del simulador.

En la sección 3.2 se ha propuesto una medida de correlación cumulativa. Esta propuesta mide la correlación total para multiparticiones de estados mezcla, teniendo características de aditividad, dependencia de la dimensión del espacio de estados y la no necesidad de la utilización de técnicas de regresión no lineal. Esta medida fue utilizada exitosamente en la detección de transiciones de fase cuántica en problemas de sistemas de muchos cuerpos. Una interesante ventaja de esta propuesta es el conocimiento del valor de la correlación en forma algebraica para estados *GHZ*, lo cual facilita la comparación entre sistemas de diferentes dimensiones.

8.1.2. Isotropía

La propagación de errores en los sistemas ocasiona falta de información. En el caso cuántico esta falta de información está directamente relacionada con la presencia de errores isotrópos en el sistema.

En la sección 5.1 se propone un doble índice geométrico que caracteriza la existencia de propagación isótropa de errores en un sistema cuántico, dado un estado inicial de referencia puro. Las principales limitaciones de este índice son su dificultad de cómputo para estados con más de 1 qubit y que no refleja la unicidad de estados cuánticos mezcla provenientes de diferentes ensambles.

Para subsanar estas limitaciones se introduce un nuevo doble índice en 5.2 que se define en forma natural en el espacio de estados cuánticos mezcla. Este índice caracteriza la propagación de errores separando el estado mezcla resultante en dos componentes: una componente inherentemente isótropa, denominada *Ancho Isótropo*, y una componente residual que indica la desalineación con el estado original de referencia, denominada *Alineación Isótropa*. La importancia de la componente isótropa es que cuantifica la falta de información, y por lo tanto no puede ser corregida.

Como forma de visualizar el grado de isotropía de un estado mezcla se ha propuesto una gráfica denominada *Triángulo de Isotropía*, en la cual los *estados provenientes de errores isótropos* ocupan una región particular. Los *estados isótropos* definidos por Horodecki son un caso particular de estos estados.

8.1.3. Algoritmo de Grover

Como ejemplo del análisis de la propagación de errores en algoritmos cuánticos, en el capítulo 6 se ha estudiado el algoritmo cuántico de búsqueda de Grover utilizando dos modelos de errores: un error de despolarización del canal total en el estado completo (TDCh) y un error de despolarización del canal local en cada qubit (LDCh).

Se ha analizado la degradación del orden del algoritmo concluyendo que el error del tipo local es más perjudicial que un error total, basándose en el ancho de error necesario, en cada caso, para mantener el orden original del algoritmo. Una explicación para esto es que el error local tiende a desalinear el estado respecto del algoritmo sin ruido, mientras que el TDCh no desalinea, lo que se ve claramente utilizando el *Triángulo de Isotropía*. Es decir, el error total sustituye al estado original por el estado máximamente mezclado (con cierta probabilidad), pero deja intacto el estado original si esto no sucede. Por otro lado, el error local saca al estado original del plano de rotación del algoritmo de Grover, resultando en una desalineación del mismo.

Cuando se utiliza el estado marcado como estado de referencia para el índice de isotropía (estadístico), se visualiza en la gráfica que este estado funciona como un atractor que se ve afectado por el error.

8.1.4. Caminatas cuánticas

La importancia de la correlación en los estados cuánticos se analiza en el capítulo 7, en el cual se estudia una caminata cuántica en un sistema que es periódicamente abierto mediante mediciones parciales del estado (qubit de control). La caminata utilizada está basada en una versión modificada del algoritmo original de Tulsi.

Se ha determinado que el desempeño tiene una fuerte vinculación con algunas medidas de correlaciones. El máximo de la *CCM* en el estado está relacionado con el paso óptimo de parada del algoritmo.

Es interesante notar que, para ciertos valores del período de medición, existe una mejora del orden del algoritmo respecto a la evolución cerrada (unitaria).

8.1.5. Simulador

Se ha realizado una implementación de un simulador de sistemas para computación cuántica utilizando un ambiente de cálculo numérico abierto y gratuito (Scilab). Este simulador permite la ampliación de sus prestaciones mediante nuevas rutinas que pueden ser agregadas a las librerías ya existentes o en nuevas librerías temáticas (algoritmos, correlaciones cuánticas y/u otras). A modo de ejemplo, fue fácilmente combinado con una librería de optimización para la implementación del algoritmo híbrido mencionado en la sección 5.1.

El simulador fue utilizado en todos los trabajos de investigación de esta tesis y muchas de sus rutinas fueron desarrolladas y mejoradas durante estos trabajos.

También por su fácil uso ha demostrado ser de utilidad en la enseñanza del área.

8.2. Futuras líneas de trabajo

En la implementación de algoritmos cuánticos, así como en los clásicos, el análisis de las fortalezas y limitaciones es un factor primordial para el conocimiento de su verdadero desempeño. Debido a esto es fundamental conocer el verdadero papel de la correlación de estados cuánticos en la mejora computacional respecto a los algoritmos clásicos. Por otro lado, la propagación de errores impone una limitación importante en esta mejora, por lo que su caracterización es necesaria.

Las futuras líneas de investigación que se pretende realizar son:

- **Medida de correlación cumulativa (CCM).**

La medida propuesta calcula la correlación total (clásica y cuántica) en estados cuánticos. Sería de interés investigar, y proponer, una medida similar que considere la dimensionalidad del espacio, pero basada exclusivamente en la correlación cuántica.

Otro tema de interés es el análisis de estados en los cuales es posible obtener fórmulas algebraicas para el valor de la correlación, como fue determinado para los estados *GHZ*.

- **Índice de isotropía.**

Así como en el caso de la correlación, es de interés hallar familias de estados cuánticos para las cuales sea posible contar con fórmulas algebraicas para el índice de isotropía (estadístico).

- **Caminatas cuánticas controladas.**

En el estudio realizado para el análisis de las caminatas cuánticas controladas fue principalmente numérico, con la finalidad principal de demostrar la potencialidad del simulador desarrollado. En un futuro análisis sería de interés trabajar con un enfoque más analítico, en especial estudiando la propagación de errores en las caminatas y la degradación del desempeño, un estudio pendiente. Queda por determinar si la elección del parámetro propuesta por Tulsí sigue siendo óptima con la presencia de errores, en el canal y en compuertas.

- **Algoritmo de Grover.**

En el estudio realizado con el algoritmo de Grover se ha analizado solamente el efecto de errores locales y totales en el canal. Se puede extender este estudio analizando los errores de compuerta.

En la evolución del algoritmo, la correlación del estado evoluciona desde cero, pasando por un máximo, y llegando a un valor bajo (idealmente nulo) en el paso final. Otro estudio pendiente es la relación entre la evolución de la correlación del estado cuántico frente a errores, y la influencia de la correlación en el desempeño de este algoritmo cuántico frente al clásico de fuerza bruta, u otros.

- **Modelos de errores.**

En el análisis de la propagación de errores en algoritmos se ha utilizado, principalmente, un modelo de error de despolarización de canal. El motivo de esta elección es la simplicidad del modelo de error con el objetivo de realizar análisis mediante simulación y analíticos. Es interesante realizar el análisis para los otros modelos de errores cuánticos introducidos en el capítulo 4, y las limitaciones en el desempeño de los algoritmos.

Otro estudio futuro es la relación entre los modelos de errores y las implementaciones físicas de los sistemas cuánticos, para determinar con más veracidad el cambio de desempeño de un algoritmo cuántico frente a una implementación particular. En especial, el análisis de la propagación en algoritmos implementados con realizaciones tolerante a fallos (fault-tolerant quantum computation).

8.3. Publicaciones

Como parte del trabajo de esta tesis doctoral fueron elaborados los siguientes artículos, presentados en revistas y congresos arbitrados:

8.3.1. Revistas científicas

- ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Ilan Cohn; Jesús García López de Lacalle. *Characterizing error propagation in quantum circuits: the Isotropic Index*. Quantum Information Processing. Vol. 16, No 2, 48. Springer. 2017. DOI: <http://dx.doi.org/10.1007/s11128-016-1507-5>.
- Ilan Cohn; ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García López de Lacalle. *Grover's search with local and total depolarizing channel errors: complexity analysis*. International Journal of Quantum Information. Vol. 14, No 2, 1650009. World Scientific. 2016. DOI: <http://dx.doi.org/10.1142/S021974991650009X>.
- Efrain Buksman; ANDRÉ L. FONSECA DE OLIVEIRA; Jesús García-López de Lacalle. *Search via quantum walks with intermediate measurements*. International Journal of Modern Physics B. Vol. 29, No 19, 1550127. World Scientific. 2015. DOI: <http://dx.doi.org/10.1142/S0217979215501271>.
- ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López de Lacalle. *Cumulative measure of correlations for multipartite quantum states*. International Journal of Modern Physics B. Vol. 28, No 7, 1450050. World Scientific. 2014. DOI: <http://dx.doi.org/10.1142/S0217979214500507>.
- ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López. *Isotropic Double Index for Quantum Errors in One Qubit*. Journal of Chemistry and Chemical Engineering, Vol 5, No 11, p: 1053 – 1058. David Publishing Company. 2011.

8.3.2. Congresos

- ANDRÉ L. FONSECA DE OLIVEIRA; Ilan Cohn; Efrain Buksman; Emiliano Espíndola; Jesús García-López. *Análisis de la propagación de errores en el código de Shor*. VIII Congreso Internacional de Computación y Telecomunicaciones, COMTEL2016, Lima, Perú. 2016.
- ANDRÉ L. FONSECA DE OLIVEIRA; Ilan Cohn; Efrain Buksman; Jesús García-López. *Enfoque alternativo al análisis de errores de despolarización en el algoritmo de Grover*. VII Congreso Internacional de Computación y Telecomunicaciones, COMTEL2015, Lima, Perú. 2015.
- Laura Gatti; ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López. *QuantumLab: simulador de código abierto para computación cuántica*. VII Congreso Internacional de Computación y Telecomunicaciones, COMTEL2015, Lima, Perú. 2015.
- A. FONSECA DE OLIVEIRA; E. Buksman; J. García-López de Lacalle; F. García Mazarío. *Caracterización de errores en circuitos cuánticos: índice de isotropía para estados de n -qubits*. XXXV Reunión Bienal de la Real Sociedad Española de Física, Gijón, España. 2015.
- Ilan Cohn; ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López de Lacalle. *Order of Grover's search algorithm with both total and local depolarizing channel errors*. XXXV Reunión Bienal de la Real Sociedad Española de Física, Gijón, España. 2015.
- ANDRÉ L. FONSECA DE OLIVEIRA; Jesús García-López de Lacalle; Efrain Buksman. *Medida de correlación cumulativa para estados cuánticos multi-qubits*. XXXIV Reunión Bienal de la Real Sociedad Española de Física, Valencia, España. 2013.

- Ismael V. Rodríguez-Brena; ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López. *Controlled quantum walk*. Workshop on Quantum Simulations , Bilbao, España. 2012.
- ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López. *Algoritmo Genético Híbrido para la Estimación de la Medida Geométrica de Entrelazamiento de Estados Cuánticos de n -qubits*. XXXVI Conferencia Latinoamericana de Informática, Asunción, Paraguay. 2010.
- ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Fabián Kozynski; Jesús García-López. *Estudo de estados quânticos puros simétricos con alto emaranhamento*. III Workshop-Escola de Computação e Informação Quântica - WECIQ 2010 , Petrópolis - RJ – Brasil. 2010.
- ANDRÉ L. FONSECA DE OLIVEIRA; Efrain Buksman; Jesús García-López. *Isotropic index for unitary quantum errors*. International Conference on Quantum Information and Computation , Stockholm, Sweden. 2010.
- F. Kozynski; A. FONSECA DE OLIVEIRA; E. Buksman Hollander; J. García-López. *Entrelazamiento geométrico de estados simétricos*. XXXII Bienal de la Real Sociedad Española de Física, Ciudad Real. 2009.

8.4. Agradecimientos

Este trabajo fue parcialmente apoyado por el Programa de Desarrollo Tecnológico (PDT, Ministerio de Educación y Cultura, Uruguay) mediante la beca S/C/BE/55/23, y por el Banco Santander y la Universidad Politécnica de Madrid mediante la beca BSCH-UPM en la convocatoria 2013/2014.

Apéndice A

Simulador

La simulación de sistemas cuánticos para el estudio y análisis de los resultados de los algoritmos es un tema fundamental en el área. Pero una de las fortalezas de la computación cuántica determina también su dificultad de simulación: la complejidad debido al aumento exponencial de los recursos necesarios para la simulación mediante la computación clásica. Este hecho ha motivado investigaciones con resultados particulares en la última década [2] [6] [30] [91] [110] [124]. En particular George Viamontes ha obtenido resultados teóricos en la eficiencia de la simulación clásica y desarrollado el simulador denominado *QuIDDP*¹ [118] [119] [120] [121] [122] [123].

En la dirección <https://www.quantiki.org/wiki/list-qc-simulators> es posible encontrar una lista exhaustiva de simuladores para sistemas cuánticos desarrollados en diferentes ambientes y lenguajes, y con los más diversos propósitos. Se resalta que muchos de los proyectos listados ya tienen sus desarrollos y mantenimiento terminados.

Considerando la existencia de los diversos simuladores mencionados, surge la pregunta:

- ¿Por qué desarrollar un nuevo simulador, en lugar de utilizar uno existente?

La respuesta a esta pregunta es múltiple:

- porque necesitábamos un simulador con diferentes enfoques que requería disponer de un sistema que funcionara bajo una plataforma de fácil y rápido desarrollo, y que dispusiera de rutinas de cálculo numérico eficientes, en particular con operaciones matriciales nativas en el ambiente,
- porque necesitábamos un sistema abierto, que posibilitase la ampliación y cooperación por parte de terceros,
- y por la preferencia por una plataforma de código abierto y disponible sin costo.

Además, el desarrollo completo de un simulador exige un mayor conocimiento para su utilización en investigación y enseñanza.

Una posible elección para el ambiente de desarrollo sería el *Matlab*², como el utilizado en los proyectos PEET [65] y QETLAB³. La limitación de esta elección es la necesidad de una licencia con costo.

Considerando las razones anteriormente expuestas, se decidió utilizar el ambiente *Scilab*⁴ para el desarrollo del simulador, denominado *QuantumLab*.

La versión del simulador utilizada en esta tesis es *QuantumLab V2.1*. La primera versión, *QuantumLab V1.0*, fue desarrollada inicialmente para la prueba del ambiente elegido [39]. Actualmente existen dos versiones operativas:

- V2.1, utilizada en este trabajo de tesis y objeto de este apéndice,

¹QuIDDP: High-Performance Quantum Circuit Simulation. <http://vlsicad.eecs.umich.edu/Quantum/qp/>

²<https://www.mathworks.com/>

³QETLAB: A MATLAB Toolbox for Quantum Entanglement. <http://www.qetlab.com>

⁴<http://www.scilab.org/>

- y la V3.0, desarrollada como tema de tesis de graduación de Ingeniería en Electrónica en la Universidad ORT Uruguay por Laura Gatti [50].

La figura A.1 ilustra la organización de rutinas en paquetes en el simulador *QuantumLab V2.1*.

Figura A.1: Estructura de rutinas del simulador.

A.1. Funciones para la definición y manejo de estados cuánticos

En el simulador existen 4 formas para trabajar con estados cuánticos:

- Estado *tipo 1*. Es una representación de un conjunto de estados puros mediante una matriz de dimensiones $2^n \times m$, siendo n la cantidad de qubits de los estados y m la cantidad de estados representados.

$$M = \begin{pmatrix} v_1^1 & v_1^2 & \dots & v_1^m \\ v_2^1 & v_2^2 & \dots & v_2^m \\ \vdots & \vdots & \dots & \vdots \\ v_{2^n}^1 & v_{2^n}^2 & \dots & v_{2^n}^m \end{pmatrix}$$

Figura A.2: Estado *tipo 1*

- Estado *tipo 2*. Es una representación de un único estado mezcla mediante sus componentes puras y sus respectivas probabilidades. Está compuesto de una lista, donde el primer elemento es una matriz (estado tipo 1) y el segundo un vector con m probabilidades (suman 1).

$$L = \left[\begin{pmatrix} v_1^1 & v_1^2 & \dots & v_1^m \\ v_2^1 & v_2^2 & \dots & v_2^m \\ \vdots & \vdots & \dots & \vdots \\ v_{2^n}^1 & v_{2^n}^2 & \dots & v_{2^n}^m \end{pmatrix}, (p_1 \ p_2 \ \dots \ p_m) \right]$$

Figura A.3: Estado *tipo 2*

$$\hat{M} = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{12^n} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{22^n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{2^n 1} & \rho_{2^n 2} & & \rho_{2^n 2^n} \end{pmatrix}$$

Figura A.4: Estado *tipo 3*

- Estado *tipo 3*. Es la representación de un único estado mezcla por su matriz de densidad. Es una matriz definida positiva, con traza 1, de dimensiones $2^n \times 2^n$, siendo n la cantidad de qubits.
- Estado *tipo 4*. Es la representación de un conjunto de estados mezclas. Es una lista de estados del tipo 3. Todos deben tener la misma dimensión (misma cantidad de qubits).

$$\hat{L} = [\hat{M}_a, \hat{M}_b, \dots \hat{M}_k]$$

Figura A.5: Estado *tipo 4*

La biblioteca de rutinas para estados está compuesta por las 17 funciones. Todas empiezan con el prefijo `qb_st_`.

■ <code>qb_st_state.</code>	■ <code>qb_st_st21dens.</code>
■ <code>qb_st_rnd_vecstates.</code>	■ <code>qb_st_lstprint.</code>
■ <code>qb_st_rnd_denstates.</code>	■ <code>qb_st_partialtrace_vec.</code>
■ <code>qb_st_simetric_states.</code>	■ <code>qb_st_partialtrace_mat.</code>
■ <code>qb_st_chkdensmatrix.</code>	■ <code>qb_st_partialtrace.</code>
■ <code>qb_st_chkvecstates.</code>	■ <code>qb_st_esf_pars.</code>
■ <code>qb_st_chkpurestate.</code>	■ <code>qb_st_state_rot.</code>
■ <code>qb_st_type.</code>	■ <code>qb_st_globalextract.</code>
■ <code>qb_st_vec2dens.</code>	

A.1.1. qb_st_state

Define de estados cuánticos vectoriales puros de n qubits (estados del tipo 1).

Uso:

```
state = qb_st_state(st [, nqb, coef])
```

st: vector con estados de la base (0, 1... 2^n-1).

nqb: cantidad de qubits (si se omite se crea con la minima necesaria)

coef: crea un solo estado utilizando el vector coef para combinar los estados indicados en st.

state: estado del tipo 1.

Ejemplos:

```
-->st1 = qb_st_state([0 1])
```

```
st1 =
```

```
1.    0.
0.    1.
```

```
-->st1 = qb_st_state([0 1], 2)
```

```
st1 =
```

```
1.    0.
0.    1.
0.    0.
0.    0.
```

```
-->st1 = qb_st_state([0 3], %nan, [1 %i])
```

```
st1 =
```

```
0.7071068
0
0
0.7071068i
```

A.1.2. qb_st_rnd_vecstates

Genera estados cuánticos vectoriales aleatorios (estados del tipo 1 ó 2).

Uso:

```
sts = qb_st_rnd_vecstates(cant [, nqb, ops])
```

cant: cantidad de estados aleatorios

nqb: Cantidad de qubits (por defecto igual a 1).

ops: booleana. %T estado mezcla del tipo 2 con probabilidades aleatorias.
 %F cant estados de tipo 1 (por defecto).

sts: estados del tipo 1. Si ops es %T un estado del tipo 2.

Ejemplos:

```
-->sts = qb_st_rnd_vecstates(2)
sts =

    0.6882285          0.8145979
    0.7229863 + 0.0602690i - 0.0228432 - 0.5795761i

-->sts = qb_st_rnd_vecstates(2, 2, %T)
sts =

    sts(1)

    0.5895394          0.2716879
    0.2674512 + 0.4783594i - 0.2202259 - 0.4875199i
- 0.1881428 - 0.0392220i    0.2540951 + 0.3160807i
    0.2309993 + 0.5116530i    0.5771376 - 0.3774274i

    sts(2)

    0.2405564    0.7594436
```

A.1.3. qb_st_rnd_denstates

Genera matrices de densidad aleatorias. (estados del tipo 3 ó 4).

Uso:

```
rhos = qb_st_rnd_denstates(cant [, nqb , ops])
```

cant: cantidad de estados aleatorios

nqb: Cantidad de qubits (por defecto igual a 1).

ops: [puros cant_mz]

 puros: booleana. %T estados puros (%F por defecto).

 cant_mz: cantidad de estados aleatorios para formar la mezcla
 (por defecto igual a 10).

rhos: si cant vale 1 es un estado de tipo 3, sino es de tipo 4.

Ejemplos:

```
-->rho = qb_st_rnd_denstates(1, %nan, [%T])
rho =

    0.2952457          - 0.4326574 + 0.1445105i
- 0.4326574 - 0.1445105i    0.7047543

-->rhos = qb_st_rnd_denstates(2)
rhos =
```

```

rhos(1)

0.6177039          0.0076865 - 0.0102672i
0.0076865 + 0.0102672i  0.3822961

```

```

rhos(2)

0.5160243          - 0.0738622 - 0.0995457i
- 0.0738622 + 0.0995457i  0.4839757

```

A.1.4. qb_st_simetric_states.

Calcula los vectores base para estados simétricos (o un estado simétrico).

Uso:

```
[sts, mat] = qb_st_simetric_states(nqb [, sts_ind, coef])
```

nqb: cantidad de qubits

sts_ind: Índice de los simétricos deseados (0 a nqb).

Si no se pasa como parámetro se retornan todos los simétricos (un estado por columna).

coef: coeficientes para la superposición de los simétricos.

Un coeficiente por índice. Genera un solo estado.

Si es %nan genera coeficientes aleatorios.

sts: estados simétricos de salida.

mat: matriz con los índices de los estados base (uno por columna, con un 1 en cada caso).

Ejemplos:

```
-->[sts, mat] = qb_st_simetric_states(3)
```

```
mat =
```

```

1.    0.    0.    0.
0.    1.    0.    0.
0.    1.    0.    0.
0.    0.    1.    0.
0.    1.    0.    0.
0.    0.    1.    0.
0.    0.    1.    0.
0.    0.    0.    1.

```

```
sts =
```

```

1.    0.          0.          0.
0.    0.5773503    0.          0.
0.    0.5773503    0.          0.
0.    0.          0.5773503    0.
0.    0.5773503    0.          0.
0.    0.          0.5773503    0.
0.    0.          0.5773503    0.
0.    0.          0.          1.

```

```
-->qb_st_lstprint(sts);

Estados puros

Estado No 1

coef = 1    base_vec = 0 [000]


Estado No 2

coef = 0.5773503    base_vec = 1 [001]

coef = 0.5773503    base_vec = 2 [010]

coef = 0.5773503    base_vec = 4 [100]


Estado No 3

coef = 0.5773503    base_vec = 3 [011]

coef = 0.5773503    base_vec = 5 [101]

coef = 0.5773503    base_vec = 6 [110]


Estado No 4

coef = 1    base_vec = 7 [111]

-->sts2 = qb_st_simetric_states(3, [1 2])
sts2 =

    0.         0.
    0.5773503    0.
    0.5773503    0.
    0.         0.5773503
    0.5773503    0.
    0.         0.5773503
    0.         0.5773503
    0.         0.

-->sts3 = qb_st_simetric_states(3, [1 2], [1 %i])
sts3 =

    0
    0.4082483
    0.4082483
    0.4082483i
    0.4082483
    0.4082483i
    0.4082483i
    0
```

A.1.5. qb_st_chkdensmatrix

Verifica si el (los) estado(s) está(n) en el formato de matriz de densidad (estado(s) del tipo 3 ó 4).

Uso:

```
res = qb_st_chkdensmatrix(st)

st: estado de entrada

res: resultado booleano.
```

Ejemplos:

```
-->st1 = qb_st_state([0 1])
st1 =

    1.    0.
    0.    1.

-->res = qb_st_chkdensmatrix(st1)
res =

F

-->st2 = qb_st_rnd_denstates(1)
st2 =

    0.3688599          0.0798794 - 0.0113406i
    0.0798794 + 0.0113406i    0.6311401

-->res = qb_st_chkdensmatrix(st2)
res =

T
```

A.1.6. qb_st_chkvecstates

Verifica si el (los) estado(s) está(n) en el formato vectorial (estado(s) del tipo 1 ó 2).

Uso:

```
res = qb_st_chkvecstates(st)

st: estado de entrada

res: resultado booleano.
```

Ejemplos:

```
-->st1
st1 =

    st1(1)

    1.    0.
    0.    1.

    st1(2)

    0.5    0.5

-->res = qb_st_chkvecstates(st1)
res =

    T

-->st2 = qb_st_rnd_denstates(1)
st2 =

    0.5299113          - 0.1363480 + 0.1161687i
    - 0.1363480 - 0.1161687i    0.4700887

-->res = qb_st_chkvecstates(st2)
res =

    F
```

A.1.7. qb_st_chkpurestate

Verifica que todos los estados en la entrada sean estados cuánticos puros.

Uso:

```
res = qb_st_chkpurestate(st)

st: estado de entrada

res: resultado booleano.
```

Ejemplos:

```
-->st1
st1 =

    st1(1)

    1.    0.
    0.    1.
```

```

        st1(2)

        0.5    0.5

-->res = qb_st_chkpurestate(st1)
res =

F

-->st2 = qb_st_rnd_denstates(1, 1, %T)
st2 =

    0.2807064          - 0.0023088 - 0.4493384i
- 0.0023088 + 0.4493384i    0.7192936

-->res = qb_st_chkpurestate(st2)
res =

T

```

A.1.8. qb_st_type

Devuelve el tipo de representación de los estados cuánticos, la cantidad de estados y la cantidad de qubits.

Uso:

```
[tp, cant, cqubits] = qb_st_type(st)
```

st: estado(s) en formato vectorial (tipo 1 ó 2).

tp: Representación vectorial

1 - matriz de estados puros (1 x columna)

2 - mezcla. Lista con matriz de estados puros y vector de probabilidades.

Representacion en matrices de densidad

3 - mezcla. Matriz de densidad.

4 - mezcla. Lista de matrices de densidad.

cant: Cantidad de estados cuánticos.

cqubits: cantidad de qubits de los estados (la misma para todos).

Ejemplo:

```
-->st1 = qb_st_state([0 2 3])
st1 =
```

```

1.    0.    0.
0.    0.    0.
0.    1.    0.
0.    0.    1.

```

```
-->[tp, cant, cqubits] = qb_st_type(st1)
cqubits =

    2.
cant =

    3.
tp =

    1.
```

A.1.1.9. qb_st_vec2dens

Cambia del formato de vectores a matrices de densidad (estados de tipo 1 ó 2 a estados de tipo 3 ó 4).

Uso:

```
Mdens = qb_st_vec2dens(st [, tbol])
```

st: estado(s) en formato vectorial (tipo 1 ó 2).

tbol: booleana. Si %F no verifica el estado de entrada
(por defecto %T).

Mdens: matrices de densidad (tipo 3 ó 4).

Ejemplos:

```
-->st1
st1 =

    st1(1)

    1.    0.
    0.    1.

    st1(2)

    0.5    0.5

-->rho = qb_st_vec2dens(st1)
rho =

    0.5    0.
    0.    0.5

-->sts = qb_st_state([0 1])
sts =

    1.    0.
    0.    1.
```

```
-->rhos = qb_st_vec2dens(sts)
rhos =
```

```
rhos(1)
```

```
1.    0.
0.    0.
```

```
rhos(2)
```

```
0.    0.
0.    1.
```

A.1.10. qb_st_st21dens

Transforma el conjunto de estados cuánticos de entrada en un único estado en formato de matriz de densidad (estados de cualquier tipo a un estado de tipo 3).

Uso:

```
Mdens = qb_st_st21dens(sts_in [, probs, tbol])
```

sts_in: estados de entrada.

probs: probabilidad de los estados cuánticos.

Solo se utiliza para estados del tipo 1 y 4 (cuando hay más de un estado).

tbol: booleana. Se utiliza si probs es %nan.

Si %T los estados son equiprobables (por defecto).

En caso de ser %F tienen probabilidad aleatoria.

Mdens: Matriz (tipo 3).

Ejemplos:

```
-->st1
```

```
st1 =
```

```
st1(1)
```

```
1.    0.
0.    1.
```

```
st1(2)
```

```
0.5    0.5
```

```
-->rho = qb_st_st21dens(st1)
```

```
rho =
```

```
0.5    0.
0.    0.5
```



```
-->sts = qb_st_state([0 1])
sts =

    1.    0.
    0.    1.

-->rho = qb_st_st2dens(sts, %nan, %F)
rho =

    0.4837839    0.
    0.          0.5162161
```

A.1.11. qb_st_lstprint

Imprime en pantalla un estado cuántico en formato lista. Utiliza la función interna *qb_st_int_st2list*.

Uso:

```
qb_st_lstprint(st)
```

st: estado cuántico en formato vectorial (tipo 1 ó 2)

Ejemplos:

```
-->st1 = qb_st_state([0 3], %nan, [1 1])
st1 =

    0.7071068
    0.
    0.
    0.7071068

-->qb_st_lstprint(st1)

Estado puro

coef = 0.7071068    base_vec = 0 [00]

coef = 0.7071068    base_vec = 3 [11]

-->st2 = qb_st_rnd_vecstates(2,2)
st2 =

    0.2297032          0.5066153
    0.4010236 - 0.3519675i - 0.1889953 + 0.2548965i
- 0.2734764 + 0.3883517i    0.2442206 - 0.2787239i
    0.0889092 - 0.6549994i    0.5773627 - 0.4146940i
```

```
-->qb_st_lstprint(st2)

Estados puros

Estado No 1

coef = 0.2297032    base_vec = 0 [00]

coef = 0.4010236-%i*0.3519675    base_vec = 1 [01]

coef = -0.2734764 +%i*0.3883517    base_vec = 2 [10]

coef = 0.0889092-%i*0.6549994    base_vec = 3 [11]

Estado No 2

coef = 0.5066153    base_vec = 0 [00]

coef = -0.1889953 +%i*0.2548965    base_vec = 1 [01]

coef = 0.2442206-%i*0.2787239    base_vec = 2 [10]

coef = 0.5773627-%i*0.4146940    base_vec = 3 [11]

-->st3 = qb_st_rnd_vecstates(2, 1, %T)
st3 =

    st3(1)

    0.4999349          0.5251171
    0.8411920 + 0.2060610i    0.8398251 - 0.1376437i

    st3(2)

    0.7988161    0.2011839

-->qb_st_lstprint(st3)

Estado mezcla

Estado No 1, probabilidad : 0.7988161

coef = 0.4999349    base_vec = 0

coef = 0.8411920+%i*0.2060610    base_vec = 1

Estado No 2, probabilidad : 0.2011839

coef = 0.5251171    base_vec = 0

coef = 0.8398251-%i*0.1376437    base_vec = 1
```

A.1.12. qb_st_partialtrace_vec

Realiza la traza parcial de estados cuánticos en formato vectorial. Utiliza la función interna *qb_st_int_ptrace_vec1*.

Uso:

```
p_trace_vec = qb_st_partialtrace_vec(st, ind_orig)
```

st: estado cuántico en formato vectorial (tipos 1 ó 2).

ind_orig: índices de los qubits que permanecen.

p_trace_vec: traza parcial.

Puede devolver el estado en formato vectorial de lista (1 estado de entrada) o en matriz de densidad (más de 1 estado a la entrada).

Ejemplos:

```
-->st1 = qb_st_state([0 3], %nan, [1 1])
```

```
st1 =
```

```
    0.7071068
```

```
    0.
```

```
    0.
```

```
    0.7071068
```

```
-->res1 = qb_st_partialtrace_vec(st1, 1)
```

```
res1 =
```

```
    res(1)
```

```
    1.    0.
```

```
    0.    1.
```

```
    res(2)
```

```
    0.5    0.5
```

```
-->st2 = qb_st_state([0 3])
```

```
st2 =
```

```
    1.    0.
```

```
    0.    0.
```

```
    0.    0.
```

```
    0.    1.
```

```
res2 = qb_st_partialtrace_vec(st2, 1)
```

```
res2 =
```

```
    res2(1)
```

```
    1.    0.
```

```
    0.    0.
```

```

        res2(2)

    0.    0.
    0.    1.

-->st3
st3 =

        st3(1)

    0.7071068    0.8944272
    0.          0.
    0.          0.
    0.7071068    0.4472136

        st3(2)

    0.8    0.2

-->res3 = qb_st_partialtrace_vec(st3, 1)
res3 =

        res3(1)

    1.    1.    0.    0.
    0.    0.    1.    1.

        res3(2)

    0.4    0.16    0.4    0.04

```

A.1.13. qb_st_partialtrace_mat

Realiza la traza parcial de estados cuánticos en formato matriz de densidad.

Uso:

```
dPT = qb_st_partialtrace_mat(dM, ind)
```

st: estado cuántico en formato matricial (tipos 3 ó 4).

ind: índices de los qubits que permanecen.

dPT: traza parcial.

Devuelve el mismo formato de la entrada.

Matriz devuelve matriz y lista de matrices devuelva lista.

Ejemplos:

```
-->st1 = qb_st_state([0 3], %nan, [1 1])
st1 =
```

```
    0.7071068
    0.
    0.
    0.7071068
```

```
-->rho1 = qb_st_vec2dens(st1)
rho1 =
```

```
    0.5    0.    0.    0.5
    0.    0.    0.    0.
    0.    0.    0.    0.
    0.5    0.    0.    0.5
```

```
-->pt1 = qb_st_partialtrace_mat(rho1,1)
pt1 =
```

```
    0.5    0.
    0.    0.5
```

```
-->st2 = qb_st_state([1 2], %nan, [1 1])
st2 =
```

```
    0.
    0.7071068
    0.7071068
    0.
```

```
-->rho2 = qb_st_vec2dens(st2)
rho2 =
```

```
    0.    0.    0.    0.
    0.    0.5    0.5    0.
    0.    0.5    0.5    0.
    0.    0.    0.    0.
```

```
-->rhoL = list(rho1, rho2)
rhoL =
```

```
    rhoL(1)
```

```
    0.5    0.    0.    0.5
    0.    0.    0.    0.
    0.    0.    0.    0.
    0.5    0.    0.    0.5
```

```

rhoL(2)

0.    0.    0.    0.
0.    0.5   0.5   0.
0.    0.5   0.5   0.
0.    0.    0.    0.

-->ptL = qb_st_partialtrace_mat(rhoL, 1)
ptL =

    ptL(1)

0.5    0.
0.     0.5

    ptL(2)

0.5    0.
0.     0.5

```

A.1.14. qb_st_partialtrace

Realiza la traza parcial de estados cuánticos. Sirve de interfaz para *qb_st_partialtrace_vec* y *qb_st_partialtrace_mat*.

Uso:

```
dPT = qb_st_partialtrace(st, ind)
```

st: estado cuántico de entrada.
ind: índices de los qubits que permanecen.

dPT: traza parcial.

Ejemplos:

```

-->st1 = qb_st_state([0 3], %nan, [2 %i])
st1 =

    0.8944272
    0
    0
    0.4472136i

-->rho1 = qb_st_vec2dens(st1)
rho1 =

    0.8    0    0 - 0.4i
    0     0    0    0
    0     0    0    0
    0.4i   0    0    0.2

```

```
-->ptA = qb_st_partialtrace(st1, 1)
ptA =

    ptA(1)

    1.    0
    0     i

    ptA(2)

    0.8    0.2

-->rhoA = qb_st_vec2dens(ptA)
rhoA =

    0.8    0
    0     0.2

-->ptB = qb_st_partialtrace(rho1, 1)
ptB =

    0.8    0
    0     0.2
```

A.1.15. qb_st_esf_pars

Calcula las coordenadas esféricas v_thetas , v_phis de los estados de entrada en formato vectorial (tipo 1).

$$st_in = \cos\left(\frac{v_theta1}{2}\right) |0\rangle + \sin\left(\frac{theta_1}{2}\right) e^{(iv_alpha1)} |1\rangle.$$

$$v_thetas \in [0, \pi], \quad v_phis \in [0, 2\pi]$$

Uso:

```
[v_thetas, v_phis] = qb_st_esf_pars(st_in)
```

st_in: estados cuánticos de entrada.

v_thetas: matriz con los ángulos resultantes (1 estado por columna).

v_phis: matriz con los ángulos resultantes (1 estado por columna).

Ejemplos:

```
-->sts
sts =

    1.    0    0.7071068    0.7071068
    0    1.    0.7071068   - 0.7071068i
```

```
-->[v_thetas, v_phis] = qb_st_esf_pars(sts)
v_phis =

    0.    0.    0.    4.712389
v_thetas =

    0.    3.1415927    1.5707963    1.5707963

-->stL
stL =

    stL(1)

    0.    0.
    1.    0.
    0.    1.
    0.    0.

    stL(2)

    0.25    0.75

-->[v_thetasL, v_phisL] = qb_st_esf_pars(stL)
v_phisL =

    0.    0.
    0.    0.
    0.    0.
v_thetasL =

    3.1415927    3.1415927
    0.          3.1415927
    3.1415927    0.
```

A.1.16. qb_st_state_rot

Gira los estados de entrada en formato vectorial de acuerdo a las coordenadas indicadas por un estado de referencia.

Uso:

```
[sts_out, M] = qb_st_state_rot(sts_in, st_ref [, p_or])
```

sts_in: estados de entrada (iniciales).

st_ref: estado de referencia para el giro.

p_or: booleana. Indica el sentido de la transformacion.

%T: Aplica a los estados sts_in el giro para llevar st_ref a $|0\rangle$ (por defecto).

%F: Aplica a los estado sts_in el giro para llevar $|0\rangle$ a st_ref.

sts_out: estados de salida (girados)

M: Matriz que aplica el giro

Ejemplos:

```
-->st_in = qb_st_state([0 1], 1, [1 -1])
st_in =

    0.7071068
   - 0.7071068i

-->st_ref = qb_st_state([0 1], 1, [1 1])
st_ref =

    0.7071068
    0.7071068i

-->st_rot = qb_st_state_rot(st_in, st_ref, %T)
st_rot =

    0.
    1.

-->st_rot = qb_st_state_rot(st_in, st_ref, %F)
st_rot =

    1.
    0.
```

A.1.17. qb_st_globalextract

Extrae la fase global de los estados de entrada. Los primeros elementos no nulos de los vectores quedan reales positivos.

Uso:

```
sts_out = qb_st_globalextract(sts_in)

sts_in: Estados de entrada (con fase global)

sts_out: Estados de salida (sin fase global)
```

Ejemplos:

```
-->st = qb_st_state([0 1], 1, [%i+2 -1])
st =

    0.8164966 + 0.4082483i
   - 0.4082483i

-->st_out = qb_st_globalextract(st)
st_out =

    0.9128709
   - 0.3651484 + 0.1825742i
```

A.2. Funciones para la definición de compuertas cuánticas

Esta biblioteca consta de 14 rutinas que posibilitan diversos tipos de compuertas cuánticas para la construcción de circuitos cuánticos en el simulador. Todas las rutinas empiezan con el prefijo `qb_cmp_`.

▪ <code>qb_cmp_I.</code>	▪ <code>qb_cmp_Za.</code>
▪ <code>qb_cmp_WalshHadamard.</code>	▪ <code>qb_cmp_Rk.</code>
▪ <code>qb_cmp_X.</code>	▪ <code>qb_cmp_Swap.</code>
▪ <code>qb_cmp_Y.</code>	▪ <code>qb_cmp_lambdaU.</code>
▪ <code>qb_cmp_Z.</code>	▪ <code>qb_cmp_binfunc.</code>
▪ <code>qb_cmp_Xa.</code>	▪ <code>qb_cmp_CNOT_k.</code>
▪ <code>qb_cmp_Ya.</code>	▪ <code>qb_cmp_Rnd.</code>

A.2.1. `qb_cmp_I`

Crea un sistema cuántico con compuertas de tipo I (identidad).

Uso:

```
M = qb_cmp_I(cant_qubits)
```

`cant_qubits`: cantidad de qubits.

`M`: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M1 = qb_cmp_I(2)
```

```
M1 =
```

```
1.  0.  0.  0.
0.  1.  0.  0.
0.  0.  1.  0.
0.  0.  0.  1.
```

A.2.2. `qb_cmp_WalshHadamard`

Crea un sistema cuántico con compuertas de tipo Hadamard.

Uso:

```
M = qb_cmp_WalshHadamard(cant_qubits)
```

`cant_qubits`: cantidad de qubits.

`M`: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M1 = qb_cmp_WalshHadamard(1)
M1 =

    0.7071068    0.7071068
    0.7071068   - 0.7071068
```

A.2.3. qb_cmp_X

Crea un sistema cuántico con compuertas de tipo X.

Uso:

```
M = qb_cmp_X(cant_qubits)

cant_qubits: cantidad de qubits.

M: Matriz que define el sistema cuántico.
```

Ejemplo:

```
-->M1 = qb_cmp_X(2)
M1 =

    0.    0.    0.    1.
    0.    0.    1.    0.
    0.    1.    0.    0.
    1.    0.    0.    0.
```

A.2.4. qb_cmp_Y

Crea un sistema cuántico con compuertas de tipo Y.

Uso:

```
M = qb_cmp_Y(cant_qubits)

cant_qubits: cantidad de qubits.

M: Matriz que define el sistema cuántico.
```

Ejemplo:

```
-->M1 = qb_cmp_Y(1)
M1 =

    0    - i
    i     0
```

A.2.5. qb_cmp_Z

Crea un sistema cuántico con compuertas de tipo Z.

Uso:

```
M = qb_cmp_Z(cant_qubits)
```

cant_qubits: cantidad de qubits.

M: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M1 = qb_cmp_Z(3)
```

```
M1 =
```

```

1.    0.    0.    0.    0.    0.    0.    0.
0.   -1.    0.    0.    0.    0.    0.    0.
0.    0.   -1.    0.    0.    0.    0.    0.
0.    0.    0.    1.    0.    0.    0.    0.
0.    0.    0.    0.   -1.    0.    0.    0.
0.    0.    0.    0.    0.    1.    0.    0.
0.    0.    0.    0.    0.    0.    1.    0.
0.    0.    0.    0.    0.    0.    0.   -1.

```

A.2.6. qb_cmp_Xa

Crea un sistema cuántico con compuertas de tipo Xa (giro continuo en X).

Uso:

```
M = qb_cmp_Xa(cant_qubits, fase)
```

cant_qubits: cantidad de qubits.

fase: parámetro de la función (fase = %pi/2 -> X)

M: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M1=qb_cmp_Xa(1,%pi/4)
```

```
M1 =
```

```

0.5 - 0.5i    0.5 + 0.5i
0.5 + 0.5i    0.5 - 0.5i

```

A.2.7. qb_cmp_Ya

Crea un sistema cuántico con compuertas de tipo Ya (giro continuo en Y).

Uso:

```
M = qb_cmp_Ya(cant_qubits, fase)
```

cant_qubits: cantidad de qubits.
 fase: parámetro de la función (fase = %pi/2 -> Y)

M: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M1 = qb_cmp_Ya(1,%pi/4)
M1 =

    0.5 - 0.5i    0.5 - 0.5i
    - 0.5 + 0.5i    0.5 - 0.5i
```

A.2.8. qb_cmp_Za

Crea un sistema cuántico con compuertas de tipo Za (giro continuo en Z).

Uso:

```
M = qb_cmp_Za(cant_qubits, fase)
```

cant_qubits: cantidad de qubits.
 fase: parámetro de la función (fase = %pi/2 -> Z)

M: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M1 = qb_cmp_Za(1,%pi/4)
M1 =

    1.    0
    0    - i
```

A.2.9. qb_cmp_Rk

Crea un sistema cuántico con compuertas de tipo Rk (cambio de fase). Similar a qb_trans_Za. Útil para la transformada cuántica de Fourier.

Uso:

```
M = qb_cmp_Rk(cant_qubits, k)
```

cant_qubits: cantidad de qubits.

k: parámetro que define la fase (divide $2\pi i$ por 2^k).

M: Matriz que define el sistema cuántico.

Ejemplos:

```
-->M1 = qb_cmp_Rk(1, 1)
```

M1 =

```
1.  0
0  -1.
```

```
-->M2 = qb_cmp_Rk(1, 2)
```

M2 =

```
1.  0
0   i
```

A.2.10. qb_cmp_Swap

Crea un sistema cuántico con una compuerta Swap (2 qubits).

Uso:

```
M = qb_cmp_Swap()
```

M: Matriz que define el sistema cuántico.

Ejemplo:

```
-->M = qb_cmp_Swap()
```

M =

```
1.  0.  0.  0.
0.  0.  1.  0.
0.  1.  0.  0.
0.  0.  0.  1.
```

A.2.11. qb_cmp_lambdaU

Crea un sistema cuántico con *cant_control* qubits que controlan el sistema definido por MU.

Uso:

```
M = qb_cmp_lambdaU(MU [, cant_control])
```

MU: Sistema cuántico a ser controlado.

cant_control: cantidad de qubits de control.

M: matriz que define el sistema cuántico con control

Ejemplos:

```
-->MCnot = qb_cmp_lambdaU(qb_cmp_X(1), 1)
MCnot =
```

```
1.  0.  0.  0.
0.  1.  0.  0.
0.  0.  0.  1.
0.  0.  1.  0.
```

```
-->MToffoli = qb_cmp_lambdaU(qb_cmp_X(1), 2)
MToffoli =
```

```
1.  0.  0.  0.  0.  0.  0.  0.
0.  1.  0.  0.  0.  0.  0.  0.
0.  0.  1.  0.  0.  0.  0.  0.
0.  0.  0.  1.  0.  0.  0.  0.
0.  0.  0.  0.  1.  0.  0.  0.
0.  0.  0.  0.  0.  1.  0.  0.
0.  0.  0.  0.  0.  0.  0.  1.
0.  0.  0.  0.  0.  0.  1.  0.
```

A.2.12. qb_cmp_binfunc

Crea un sistema cuántico con las funciones binarias dadas por la tabla Mfunc.

Uso:

```
M = qb_cmp_binfunc(Mfunc)
```

Mfunc: Tabla con las funciones binarias

2^(cant_qubits) filas

m columnas (1 X función binaria)

M: matriz que define el sistema cuántico.

Opera en un sistema con una cantidad de qubits igual

a (cant_qubits + m).

Se deben utilizar ancillas $|0\rangle$ en los últimos m qubits.

Ejemplo:

```
-->fAND = [0;0;0;1]
fAND =
```

```
0.
0.
0.
1.
```

```
-->M_AND = qb_cmp_binfunc(fAND)
M_AND =
```

```
1.  0.  0.  0.  0.  0.  0.  0.
0.  1.  0.  0.  0.  0.  0.  0.
0.  0.  1.  0.  0.  0.  0.  0.
0.  0.  0.  1.  0.  0.  0.  0.
0.  0.  0.  0.  1.  0.  0.  0.
0.  0.  0.  0.  0.  1.  0.  0.
0.  0.  0.  0.  0.  0.  0.  1.
0.  0.  0.  0.  0.  0.  1.  0.
```

```
-->st_in = qb_st_state([0:2:6],3)
st_in =
```

```
1.  0.  0.  0.
0.  0.  0.  0.
0.  1.  0.  0.
0.  0.  0.  0.
0.  0.  1.  0.
0.  0.  0.  0.
0.  0.  0.  1.
0.  0.  0.  0.
```

```
-->qb_st_lstprint(st_in)
```

Estados puros

Estado No 1

```
coef = 1    base_vec = 0 [000]
```

Estado No 2

```
coef = 1    base_vec = 2 [010]
```

Estado No 3

```
coef = 1    base_vec = 4 [100]
```


Estado No 4

coef = 1 base_vec = 6 [110]

```
-->st_out = M_AND*st_in
st_out =
```

```

1.   0.   0.   0.
0.   0.   0.   0.
0.   1.   0.   0.
0.   0.   0.   0.
0.   0.   1.   0.
0.   0.   0.   0.
0.   0.   0.   0.
0.   0.   0.   1.
```

```
-->qv_st_lstprint(st_out)
```

Estados puros

Estado No 1

coef = 1 base_vec = 0 [000]

Estado No 2

coef = 1 base_vec = 2 [010]

Estado No 3

coef = 1 base_vec = 4 [100]

Estado No 4

coef = 1 base_vec = 7 [111]

A.2.13. qb_cmp_CNOT_k

Crea un sistema cuántico con compuertas de tipo CNOT con rotación continua.

Uso:

M = qb_cmp_CNOT_k(k)

k: índice de la rotación

0: identidad

1: CNOT

M: matriz que define el sistema cuántico

Ejemplos:

```
-->M1 = qb_cmp_CNOT_k(1)
M1 =
```

```
1.  0  0  0
0   1. 0  0
0   0  0  1.
0   0  1.  0
```

```
-->M2 = qb_cmp_CNOT_k(0.5)
M2 =
```

```
1.  0  0  0
0   1. 0  0
0   0  0.5 + 0.5i  0.5 - 0.5i
0   0  0.5 - 0.5i  0.5 + 0.5i
```

A.2.14. qb_cmp_Rnd

Crea un sistema cuántico con compuertas al azar.

Uso:

```
M = qb_cmp_Rnd(cant_qubits)
```

cant_qubits: cantidad de qubits

M: matriz que define el sistema cuántico

Ejemplos:

```
--> M = qb_cmp_Rnd(1)
M =
```

```
- 0.3675779 + 0.6760389i - 0.6383757 + 0.0182865i
- 0.4546669 + 0.4484818i  0.7425255 + 0.2019851i
```

```
-->clean(M*M')
ans =
```

```
1.  0
0   1.
```

A.3. Funciones para la definición de medidas cuánticas

Esta biblioteca consta de 4 rutinas para la realización de medidas en estados cuánticos. Todas las rutinas empiezan con el prefijo `qb_ms_`.

- `qb_ms_cart_proj.`
- `qb_ms_LProjMeasures.`
- `qb_ms_ProjMeasure.`
- `qb_ms_QbsMeasure.`

A.3.1. `qb_ms_cart_proj`

Calcula los proyectores (base canónica) para medir m qubits de un estado de n qubits.

Uso:

```
Projs = qb_ms_cart_proj(nqb [, ind_qb])
```

`nqb`: cantidad de qubits totales del estado cuántico a medir.

`ind_qb`: Índices de los qubits a medir (1 a `nqb`).

Por defecto todos los qubits.

`Projs`: Lista con los proyectores.

Ejemplos:

```
-->ProjsTot = qb_ms_cart_proj(2)
ProjsTot =
```

```
ProjsTot(1)
```

```
1.    0.    0.    0.
0.    0.    0.    0.
0.    0.    0.    0.
0.    0.    0.    0.
```

```
ProjsTot(2)
```

```
0.    0.    0.    0.
0.    1.    0.    0.
0.    0.    0.    0.
0.    0.    0.    0.
```

```
ProjsTot(3)
```

```
0.    0.    0.    0.
0.    0.    0.    0.
0.    0.    1.    0.
0.    0.    0.    0.
```

```

    ProjsTot(4)

    0.    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    1.

-->Projs1qb = qb__ms_cart_proj(2, 1)
Projs1qb =

    Projs1qb(1)

    1.    0.    0.    0.
    0.    1.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.

    Projs1qb(2)

    0.    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    1.    0.
    0.    0.    0.    1.

```

A.3.2. qb_ms_ProjMeasure

Realiza una medida proyectiva.

Uso:

```
res = qb_ms_ProjMeasure(st_in, Proj, ops)
```

st_in: estado de entrada
 Proj: Proyector.
 ops: Retira los estados con probabilidad cero. (Por defecto %F)

res: Lista con los resultados.
 res(1) = vector con las probabilidades de las medidas.
 res(2) = Estados resultantes (mismo formato de st_in)
 Estados con probabilidad 0, tienen como resultado %nan (todos los elementos)

Ejemplos:

```

-->st = qb_st_state([0 1], 1, [1 2])
st =

    0.4472136
    0.8944272

```

```

-->Projs = qb_ms_cart_proj(1)
Projs =

    Projs(1)

    1.    0.
    0.    0.

    Projs(2)

    0.    0.
    0.    1.

-->res = qb_ms_ProjMeasure(st, Projs(1))
res =

    res(1)

    0.2

    res(2)

    1.
    0.

-->rho = qb_st_vec2dens(st)
rho =

    0.2    0.4
    0.4    0.8

-->res2 = qb_ms_ProjMeasure(rho, Projs(2))
res2 =

    res2(1)

    0.8

    res2(2)

    0.    0.
    0.    1.

```

A.3.3. qb_ms_LProjMeasures

Realiza un conjunto de medidas proyectivas.

Uso:

```
res = qb_ms_LProjMeasures(st_in, Projs[, ops])
```

st_in: estado de entrada

Projs: Lista con los proyectores (deben ser ortogonales).

ops: Retira los estados con probabilidad cero. (Por defecto %F)

res: Lista con resultados. Cada elemento es el resultado de un proyector en el formato dado por qb_cc_execProjMeasure.

Ejemplo:

```
-->st = qb_st_state([0 1], 1, [1 2])
```

```
st =
```

```
0.4472136
```

```
0.8944272
```

```
-->Projs = qb__ms_cart_proj(1)
```

```
Projs =
```

```
Projs(1)
```

```
1.    0.
```

```
0.    0.
```

```
Projs(2)
```

```
0.    0.
```

```
0.    1.
```

```
-->res = qb_ms_LProjMeasures(st, Projs)
```

```
res =
```

```
res(1)
```

```
res(1)(1)
```

```
0.2
```

```
res(1)(2)
```

```
1.
```

```
0.
```

```

res(2)

res(2)(1)

0.8

res(2)(2)

0.
1.

```

A.3.4. qb_ms_QbsMeasure

Mide un conjunto de qubits de un estado cuántico, obteniendo el estado resultante aleatoriamente. Utiliza medidas proyectivas en la base canónica.

Uso:

```

[res_out, st_out] = qb_ms_QbsMeasure(st_in, ind_qbs)

st_in: estado de entrada
ind_qbs: Lista de qubits a medir.

res_out: Resultado de la medida.

st_out: Estado resultante después de la medida.

```

Ejemplo:

```

-->st = qb_st_state([0 3], 2, [1 1])
st =

    0.7071068
    0.
    0.
    0.7071068

-->[res_out, st_out] = qb_ms_QbsMeasure(st, [1])
st_out =

    0.
    0.
    0.
    1.
res_out =

    1.

```

```
-->[res_out, st_out] = qb_ms_QbsMeasure(st, [1])
st_out =

    1.
    0.
    0.
    0.
res_out =

    0.
```

A.4. Funciones para la definición de operadores de error en estados cuánticos

Esta biblioteca consta de 14 rutinas para la definición de operadores de error en circuitos cuánticos. Todas las rutinas empiezan con el prefijo `qb_err_`.

■ <code>qb_err_contIsoind.</code>	■ <code>qb_err_OSR_PhaseFlip.</code>
■ <code>qb_err_contXYZind.</code>	■ <code>qb_err_OSR_BitPhaseFlip.</code>
■ <code>qb_err_contXYZcorr.</code>	■ <code>qb_err_OSR_AmpDamp.</code>
■ <code>qb_err_ISO_DepChannel.</code>	■ <code>qb_err_OSR_GenAmpDamp.</code>
■ <code>qb_err_OSR_DepChannel.</code>	■ <code>qb_err_OSR_PhaseDamp.</code>
■ <code>qb_err_OSR_DepChannelN.</code>	■ <code>qb_err_OSR_m2nQbs.</code>
■ <code>qb_err_OSR_BitFlip.</code>	■ <code>qb_err_OSR_GeneraM.</code>

A.4.1. `qb_err_contIsoind`

Genera matrices para errores continuos isótropos con las rotaciones según X,Y y Z, independientes (no correlacionados) en cada qubit.

Uso:

```
MError = qb_err_contIsoind(qubits, qubits_ind, pars, cant)
```

`qubits`: cantidad de qubits.

`qubits_ind`: índice de los qubits con error

`pars`: parámetros para la generación de errores

`pars` es un escalar => 0.1 = stdv, todos los qubits

`pars` vector (2 x 1) => $\begin{bmatrix} (0/1) \\ \text{[(stdv/max)]} \end{bmatrix}$ (normal/uniform)

`cant`: Cantidad de matrices de error.

`MError`: Si `cant` es igual a 1 (por defecto) devuelve una matriz que genera el error (aleatorio).

 Si `cant` > 1 devuelve una lista con matrices que generan el error.

Ejemplo:

```
-->st = qb_st_state(0)
st =

    1.
    0.

-->MError = qb_err_contIsoind(1, 1, [1; 0.1], 5000);
-->st_out = qb_cc_execCError(st, MError);
-->qb_ut_Bloch(st_out);
-->MError = qb_err_contIsoind(1, 1, [1; 0.2], 5000);
-->st_out = qb_cc_execCError(st, MError);
-->qb_ut_Bloch(st_out);
```

Las figuras A.6(a) y A.6(b) ilustran los errores generados en el ejemplo.

(a) 5000 estados de error alrededor del estado $|0\rangle$ generados con `qb_err_contIsoind`. Desvío máximo igual a 0,1.

(b) 5000 estados de error alrededor del estado $|0\rangle$ generados con `qb_err_contIsoind`. Desvío máximo igual a 0,2.

Figura A.6: Errores generados con la función `qb_err_contIsoind`.

A.4.2. qb_err_contXYZind

Genera matrices para errores continuos con las rotaciones según X,Y y Z, independientes (no correlacionados) en cada qubit.

Uso:

```
MError = qb_err_contXYZind(qubits, qubits_ind, pars, cant)
```

qubits: cantidad de qubits.

qubits_ind: índice de los qubits con error (por defecto en todos los qubits)

pars: parámetros para la generación de errores

pars es un escalar => $a_x = a_y = a_z = \text{stdv}$,
todos los qubits (por defecto = 0.1)

pars es un vector (1 X 3) => $[a_x, a_y, a_z]$, todos stdv

pars matriz (2 x 3) => $\begin{bmatrix} (0/1) & (0/1) & (0/1) \\ (\text{stdv}/\text{max}) & (\text{stdv}/\text{max}) & (\text{stdv}/\text{max}) \end{bmatrix}$

cant: Cantidad de matrices de error.

MError: Si cant es igual a 1 (por defecto) devuelve una matriz que genera el error (aleatorio).

Si cant > 1 devuelve una lista con matrices que generan el error.

Ejemplo:

```
-->st = qb_st_state(0)
```

```
st =
```

```
1.
```

```
0.
```

```
-->MError = qb_err_contXYZind(1, 1, [1 1 1; 0.1 0.1 0.1], 5000);
```

```
-->st_out = qb_cc_execCError(st, MError);
```

```
-->qb_ut_Bloch(st_out);
```

```
-->MError = qb_err_contXYZind(1, 1, [1 1 1; 0.2 0.2 0.2], 5000);
```

```
-->st_out = qb_cc_execCError(st, MError);
```

```
-->qb_ut_Bloch(st_out);
```

Las figuras A.7(a) y A.7(b) ilustran los errores generados en el ejemplo.

(a) 5000 estados de error alrededor del estado $|0\rangle$ generados con `qb_err_contXYZind`. Desvío máximo igual a 0,1.

(b) 5000 estados de error alrededor del estado $|0\rangle$ generados con `qb_err_contXYZind`. Desvío máximo igual a 0,2.

Figura A.7: Errores generados con la función `qb_err_contXYZind`.

A.4.3. `qb_err_contXYZcorr`

Genera matrices para errores continuos con las rotaciones según X,Y y Z, correlacionando los qubits.

Uso:

```
MError = qb_err_contXYZcorr(qubits, qubits_ind, pars, cant)
```

qubits: cantidad de qubits.

qubits_ind: índice de los qubits con error (por defecto en todos los qubits)

pars: parámetros para la generación de errores

pars es un escalar => pars = stdv (1-p) en I. p en (X, Y, Z).

Por defecto p = 0.1.

pars es un vector (1 X 2) => [a_i, a_x = a_y = a_z], todos stdv

pars es un vector (1 X 4) => [a_i, a_x, a_y, a_z], todos stdv

pars vector (2 x 1) => [(0/1)]
[(stdv/max)]

a_i = (1-pars(2,1)), a_x = a_y = a_z = pars(2,1)

pars matriz (2 x 2) => [(0/1) (0/1)]
[(stdv/max) (stdv/max)]

a_i = pars(2,1), a_x = a_y = a_z = pars(2,2)

pars matriz (2 x 4) => [(0/1) (0/1) (0/1) (0/1)]
[(stdv/max) (stdv/max) (stdv/max) (stdv/max)]

a_i = pars(2,1), a_x = pars(2,2), a_y = pars(2,3), a_z = pars(2,4)

cant: Cantidad de matrices de error.

MError: Si cant es igual a 1 (por defecto) devuelve una matriz que genera el error (aleatorio).

Si cant > 1 devuelve una lista con matrices que generan el error.

Ejemplo:

```
-->st=qb_st_state(0, 2)
st =

    1.
    0.
    0.
    0.

-->MErrorI = qb_err_contXYZind(2, [1 2], [1 1 1; 0.25 0.25 0.25], 5000);
-->st_out_ind = qb_cc_execCError(st, MErrorI);
-->st_out_ind_1qb = qb_st_partialtrace(st_out_ind, 1);
-->qb_ut_Bloch(st_out_ind_1qb);
-->MErrorC = qb_err_contXYZcorr(2, [1 2], [1; 0.25], 5000);
-->st_out_corr = qb_cc_execCError(st, MErrorC);
-->st_out_corr_1qb = qb_st_partialtrace(st_out_corr, 1);
-->qb_ut_Bloch(st_out_corr_1qb);
```

Las figuras A.8(a) y A.8(b) ilustran los errores generados en el ejemplo.

(a) Traza parcial en el primer qubit de 5000 estados de error alrededor del estado $|00\rangle$ generados con *qb_err_contXYZind*. Desvío máximo igual a 0,25.

(b) Traza parcial en el primer qubit de 5000 estados de error alrededor del estado $|0\rangle$ generados con *qb_err_contXYZcorr*. Desvío máximo igual a 0,25.

Figura A.8: Comparación de errores generados con las funciones *qb_err_contXYZind* y *qb_err_contXYZcorr* alrededor del estado $|00\rangle$. Las figuras ilustran los estados resultantes de la traza parcial quedándose con el primer qubit. Se aprecia que en el primer caso (*qb_err_contXYZind*) los estados son puros (puntos negros) ya que los errores son independientes. En el segundo caso (*qb_err_contXYZcorr*) las trazas dan como resultados estados mezcla (puntos blancos).

A.4.4. qb_err_ISO_DepChannel

Aplica una despolarización del canal en todo el estado.

Uso:

```
rho_out = qb_err_ISO_DepChannel(rho_in, p)
```

rho_in: estado de entrada.

p: Probabilidad de despolarizar el estado.

L0p: Lista con los operadores.

Ejemplo:

```
-->st = qb_st_state(0,2);
-->rho = qb_st_vec2dens(st)
rho =
```

```
1.    0.    0.    0.
0.    0.    0.    0.
0.    0.    0.    0.
0.    0.    0.    0.
```

```
-->rho_out = qb_err_ISO_DepChannel(rho, 0.25)
rho_out =
```

```
0.8125  0.    0.    0.
0.    0.0625  0.    0.
0.    0.    0.0625  0.
0.    0.    0.    0.0625
```

A.4.5. qb_err_OSR_DepChannel

Genera una lista con los operadores para la operación de despolarización del canal para 1 qubit

Uso:

```
L0p = qb_err_OSR_DepChannel(p)
```

p: Parámetro para la generación de las matrices del operador.
Probabilidad de despolarizar el estado.

L0p: Lista con los operadores.

Ejemplo:

```
-->st = qb_st_state(0,2);
-->rho = qb_st_vec2dens(st)
rho =

    1.    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.

-->L0p = qb_err_OSR_DepChannel(0.25);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.875    0    0    0
    0    0    0    0
    0    0    0.125    0
    0    0    0    0

-->rho_out = qb_cc_execOSR1qb(rho_out, 2, L0p)
rho_out =

    0.765625    0    0    0
    0    0.109375    0    0
    0    0    0.109375    0
    0    0    0    0.015625
```

Las figuras A.9(a) y A.9(b) ilustran el efecto de la operación de despolarización del canal en un estado de 1 qubit, para dos valores de p .

(a) Depolarizing channel con $p = 0,25$.

(b) Depolarizing channel con $p = 0,75$.

Figura A.9: Efecto de la aplicación de los operadores de despolarización del canal sobre los estados puros en la esfera de Bloch.

A.4.6. qb_err_OSR_DepChannelN

Genera una lista con los operadores para la operación de despolarización del canal para n qubits

Uso:

```
L0p = qb_err_OSR_DepChannelN(p [, c_qbs])
```

p: Parámetro para la generación de las matrices del operador.

Probabilidad de despolarizar el estado.

c_qbs: cantidad de qubits (por defecto 1).

L0p: Lista con los operadores.

Ejemplo:

```
-->st = qb_st_state(0,3);
-->rho = qb_st_vec2dens(st);
-->L0p = qb_err_OSR_DepChannel(0.25);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p);
-->rho_out = qb_cc_execOSR1qb(rho_out, 2, L0p)
rho_out =
```

0.765625	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0.109375	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0.109375	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.015625	0
0	0	0	0	0	0	0	0

```
-->L0p = qb_err_OSR_DepChannelN(0.25, 2);
-->L0p_out = qb_err_OSR_m2nQbs(L0p, [1 2], 3);
-->rho_out2 = qb_cc_execOSRNqb(rho, L0p_out)
rho_out2 =
```

0.8125	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0.0625	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0.0625	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.0625	0
0	0	0	0	0	0	0	0

A.4.7. qb_err_OSR_BitFlip

Genera una lista con los operadores para la operación de bit flip en 1 qubit.

Uso:

`L0p = qb_err_OSR_BitFlip(p)`

`p`: Parámetro para la generación de las matrices del operador.
Probabilidad de mantener el estado original.

`L0p`: Lista con los operadores.

Ejemplo:

```
-->st = qb_st_state(0,1);
-->rho = qb_st_vec2dens(st)
rho =

    1.    0.
    0.    0.

-->L0p = qb_err_OSR_BitFlip(0.9);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.9    0.
    0.    0.1
```

La figura A.10(a) muestra el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante el operador de bit flip.

A.4.8. qb_err_OSR_PhaseFlip

Genera una lista con los operadores para la operación de phase flip en 1 qubit.

Uso:

`L0p = qb_err_OSR_PhaseFlip(p)`

`p`: Parámetro para la generación de las matrices del operador.
Probabilidad de mantener el estado original.

`L0p`: Lista con los operadores.

Ejemplo:

```
-->st = qb_st_state([0 1],1,[1 1]);
-->rho = qb_st_vec2dens(st)
rho =

    0.5    0.5
    0.5    0.5
```



```
-->L0p = qb_err_OSR_PhaseFlip(0.9);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.5    0.4
    0.4    0.5
```

La figura A.10(b) muestra el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante el operador de phase flip.

A.4.9. qb_err_OSR_BitPhaseFlip

Genera una lista con los operadores para la operación de bit-phase flip en 1 qubit.

Uso:

```
L0p = qb_err_OSR_BitPhaseFlip(p)
```

p: Parámetro para la generación de las matrices del operador.
Probabilidad de mantener el estado original.

L0p: Lista con los operadores.

Ejemplo:

```
-->st = qb_st_state([0 1],1,[1 2*i]);
-->rho = qb_st_vec2dens(st)
rho =

    0.2    - 0.4i
    0.4i    0.8

-->L0p = qb_err_OSR_BitPhaseFlip(0.9);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.26    - 0.4i
    0.4i    0.74
```

La figura A.10(c) muestra el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante el operador de bit-phase flip.

(a) Bit flip ($p = 0,3$).

(b) Phase flip ($p = 0,3$).

(c) Bit-phase flip ($p = 0,3$).

(d) Operación en secuencia de los tres operadores ($p = 0,1$).

Figura A.10: Efecto de la aplicación de los operadores bit flip, phase flip, bit-phase flip y la secuencia de los tres sobre los estados puros en la esfera de Bloch.

A.4.10. qb_err_OSR_AmpDamp

Genera una lista con los operadores para la operación de amplitud damping en 1 qubit.

Uso:

`L0p = qb_err_OSR_AmpDamp(lamb)`

`lamb`: Parámetro para la generación de las matrices del operador.
Probabilidad de perdida.

`L0p`: Lista con los operadores.

Ejemplo:

```
-->st=qb_st_state([0 1], 1, [1 1])
st =

    0.7071068
    0.7071068
```

```
-->rho=qb_st_vec2dens(st)
rho  =

    0.5    0.5
    0.5    0.5

-->LOp = qb_err_OSR_AmpDamp(0.25);
-->rho_out = qb_cc_execOSR1qb(rho, 1, LOp)
rho_out  =

    0.625    0.4330127
    0.4330127    0.375
```

La figura A.11(a) muestra el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante el operador de amplitud damping.

A.4.11. qb_err_OSR_PhaseDamp

Genera una lista con los operadores para la operación de phase damping en 1 qubit. Son los mismos operadores de phase flip.

Uso:

```
LOp = qb_err_OSR_PhaseDamp(lamb)
```

lamb: Parámetro para la generación de las matrices del operador.
Probabilidad de perdida.

LOp: Lista con los operadores.

Ejemplo:

```
-->st=qb_st_state([0 1], 1, [1 1])
st  =

    0.7071068
    0.7071068

-->rho=qb_st_vec2dens(st)
rho  =

    0.5    0.5
    0.5    0.5

-->lamb = 0.9;
-->p = (1 + sqrt(1-lamb))/2
p  =

    0.6581139

-->LOp = qb_err_OSR_PhaseDamp(lamb);

-->rho_out = qb_cc_execOSR1qb(rho, 1, LOp)
```

```
rho_out =

    0.5      0.1581139
    0.1581139  0.5

-->L0p = qb_err_OSR_PhaseFlip(p);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.5      0.1581139
    0.1581139  0.5
```

La figura A.11(b) muestra el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante el operador de phase damping.

(a) Amplitude damping con $p = 0,5$.

(b) Phase damping con $p = 0,5$.

Figura A.11: Efecto de la aplicación de los operadores de amplitude damping y phase damping sobre los estados puros en la esfera de Bloch.

A.4.12. qb_err_OSR_GenAmpDamp

Genera una lista con los operadores para la operación de generalized amplitude damping en 1 qubit.

Uso:

`L0p = qb_err_OSR_GenAmpDamp(lamb, p)`

`lamb`: Parámetro para la generación de las matrices del operador.
Probabilidad de pérdida.

`p`: Parámetro para la generación de las matrices del operador.
Probabilidad de mantener el estado $|0\rangle$.

`L0p`: Lista con los operadores.

Ejemplo:

```
-->st=qb_st_state([0 1], 1, [1 1])
st =

    0.7071068
    0.7071068

-->rho=qb_st_vec2dens(st)
rho =

    0.5    0.5
    0.5    0.5

-->L0p = qb_err_OSR_GenAmpDamp(0.25, 1);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.625    0.4330127
    0.4330127    0.375

-->L0p = qb_err_OSR_GenAmpDamp(0.25, 0.5);
-->rho_out = qb_cc_execOSR1qb(rho, 1, L0p)
rho_out =

    0.5    0.4330127
    0.4330127    0.5
```

Las figuras A.12(a) y A.12(b) ilustran el resultado de operar cada uno de los estados puros de un qubit sobre la esfera de Bloch mediante el operador de generalized amplitude damping para dos conjuntos de parámetros.

(a) Generalized amplitude damping con $p = 0,5$ y $\lambda = 0,75$.

(b) Generalized amplitude damping con $p = 0,5$ y $\lambda = 0,5$.

Figura A.12: Efecto de la aplicación de los operadores de generalized amplitude damping sobre los estados puros en la esfera de Bloch.

A.4.13. qb_err_OSR_m2nQbs

Genera operadores de OSR para estados de dimensiones superiores. Opera con la identidad en los restantes qubits.

Uso:

```
L0p_out = qb_err_OSR_m2nQbs(L0p_in, ind_OSR, c_qbs)
```

L0p_in: Lista con los operadores de entrada.

ind_OSR: índice de los qubits a operar (misma cantidad y orden que L0p_in).

c_qbs: cantidad de qubits total;

p: Parámetro para la generación de las matrices del operador.

Probabilidad de mantener el estado $|0\rangle$.

L0p_out: Lista con los operadores para el estado completo.

Ejemplo:

```
-->L0p_in = qb_err_OSR_BitFlip(0.9)
```

```
L0p_in =
```

```
    L0p_in(1)
```

```
0.9486833    0.
0.           0.9486833
```

```
    L0p_in(2)
```

```
0.           0.3162278
0.3162278    0.
```

```
-->L0p_out = qb_err_OSR_m2nQbs(L0p_in, 1, 2)
```

```
L0p_out =
```

```
    L0p_out(1)
```

```
0.9486833    0.           0.           0.
0.           0.9486833    0.           0.
0.           0.           0.9486833    0.
0.           0.           0.           0.9486833
```

```
    L0p_out(2)
```

```
0.           0.           0.3162278    0.
0.           0.           0.           0.3162278
0.3162278    0.           0.           0.
0.           0.3162278    0.           0.
```

A.4.14. qb_err_OSR_GeneraM

Genera una lista con los operadores de OSR considerando un estado de *sqbs* qubits, un ambiente de *eqbs* qubits, siendo *U* la operación en el sistema cerrado estado/ambiente.

Uso:

```
L0p = qb_err_OSR_GeneraM(eqbs, sqbs, U)
```

eqbs: Indica la cantidad de qubits del ambiente.

En caso de ser un número considera el estado del ambiente como el $|0\rangle$.

Si es un estado puro genera los operadores considerando el estado en la interacción.

Si no esta definido o es %nan, por defecto el ambiente tiene un 1 qubit y el estado es $|0\rangle$.

sqbs: Indica la cantidad de qubits del sistema (sin ambiente).

Puede ser un número o un estado cuántico.

Si no esta definido o es %nan, por defecto vale 1.

U: Operador unitario entre sistema y ambiente.

Si no esta o es %nan, se utiliza un operador aleatorio.

L0p: Lista con los operadores.

Ejemplo:

```
-->U = qb_cmp_Rnd(2)
```

```
U =
```

```
column 1 to 3
```

```
0.5529853 - 0.2417009i - 0.0212108 + 0.0517456i 0.1209124 - 0.5073038i
- 0.401362 + 0.0470178i 0.4715172 - 0.6645642i 0.1116463 - 0.0555577i
- 0.4859936 - 0.4585986i - 0.4759108 + 0.0791393i 0.5411425 - 0.0547271i
- 0.0861950 - 0.1362125i - 0.2244127 - 0.2231217i - 0.5302727 - 0.3680381i
```

```
column 4
```

```
- 0.5914105 - 0.1044852i
- 0.3779019 - 0.1198435i
- 0.0596017 - 0.1461535i
0.2796527 - 0.6156504i
```

```
-->L0p = qb_err_OSR_GeneraM(1, 1, U)
```

```
L0p =
```

```
L0p(1)
```

```
0.5529853 - 0.2417009i 0.1209124 - 0.5073038i
- 0.4859936 - 0.4585986i 0.5411425 - 0.0547271i
```

L0p(2)

- 0.401362 + 0.0470178i 0.1116463 - 0.0555577i
 - 0.0861950 - 0.1362125i - 0.5302727 - 0.3680381i

A.5. Funciones para la operación de circuitos cuánticos

Esta biblioteca consta de 5 rutinas que facilitan la simulación de operaciones cuánticas, sirviendo de interfaz para que no sea necesario especificar el tipo de representación que está siendo utilizada para el(los) estado(s) cuántico(s). Todas las rutinas empiezan con el prefijo `qb_cc_`.

- `qb_cc_Gate.`
- `qb_cc_execCError.`
- `qb_cc_execRndCError.`
- `qb_cc_execOSR1qb.`
- `qb_cc_execOSRNqb.`

A.5.1. `qb_cc_Gate`

Calcula el resultado de pasar un estado por un circuito cuántico representado por una matriz `M` unitaria.

Uso:

```
st_out = qb_cc_execGate(st_in, M)
```

`st_in`: estado de entrada.

`M`: Matriz del circuito cuántico.

`st_out`: estado de salida.

Ejemplo:

```
-->X = qb_cmp_X(1);
-->st = qb_st_state([0 1])
st =

    1.    0.
    0.    1.

-->st_out = qb_cc_execGate(st, X)
st_out =

    0.    1.
    1.    0.
```



```
-->rho = qb_st_vec2dens(st)
rho =

    rho(1)

    1.    0.
    0.    0.

    rho(2)

    0.    0.
    0.    1.

-->rho_out = qb_cc_execGate(rho, X)
rho_out =

    rho_out(1)

    0.    0.
    0.    1.

    rho_out(2)

    1.    0.
    0.    0.
```

A.5.2. qb_cc_execCError

Calcula el resultado de ocasionar un error continuo en un estado cuántico.

Uso:

```
st_out = qb_cc_execCError(st_in, MErr)
```

st_in: estado de entrada.

MErr: Matriz, o lista, con la definición para el error continuo.

Si es una lista, cada estado inicial se multiplica por la cantidad existente en la lista.

st_out: estado de salida.

Ejemplo:

```
-->st = qb_st_state(0)
st =

    1.
    0.

-->MError = qb_err_contIsoind(1, 1, [1; 0.1], 3);
```

```
-->st_out = qb_cc_execCError(st, MError)
st_out =

    0.9999934 - 0.0035146i    0.9998948 - 0.0014143i    0.9959808 + 0.0315373i
- 0.0009359 + 0.0001747i - 0.0080153 + 0.0120079i    0.0702286 - 0.0457768i
```

A.5.3. qb_cc_execRndCError

Calcula el resultado de ocasionar errores continuos (aleatorios dentro de una lista) en un conjunto de estados cuánticos.

Uso:

```
st_out = qb_cc_execRndCError(st_in, MErr)
```

st_in: estado de entrada.

MErr: Lista con la definición para los errores continuos.

Para cada estado en st_in se elige al azar una matriz
de error de la Lista MErr.

La cantidad de estados finales es igual a la cantidad inicial.

st_out: estado de salida.

Ejemplo:

```
-->st = qb_st_state([0 0])
st =

    1.    1.
    0.    0.

-->MError = qb_err_contIsoind(1, 1, [1; 0.1], 5);
-->st_out = qb_cc_execRndCError(st, MError)
st_out =

    0.9969090          0.9961959
- 0.0101963 + 0.0779007i - 0.0752074 + 0.0440179i
```

A.5.4. qb_cc_execOSR1qb

Realiza una operación sobre un estado cuántico mediante operadores cuánticos (OSR) de 1 qubit.

Uso:

```
st_out = qb_cc_execOSR1qb(st_in, ind_qubit, LOp)
```

st_in: estado de entrada.

ind_qubit: Índice del qubit para la operación

LOp: Lista con operadores de 1 qubit.

st_out: estado de salida (siempre en el formato de matriz de densidad).

Ejemplo:

```
-->st = qb_st_state(0,2)
st =

    1.
    0.
    0.
    0.

-->L0pBF = qb_err_OSR_BitFlip(0.9);
-->rho_out1 = qb_cc_execOSR1qb(st, 1, L0pBF)
rho_out1 =

    0.9    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.1   0.
    0.    0.    0.    0.

-->rho_out2 = qb_cc_execOSR1qb(st, 2, L0pBF)
rho_out2 =

    0.9    0.    0.    0.
    0.    0.1   0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.
```

A.5.5. qb_cc_execOSRNqb

Realiza una operación sobre un estado cuántico mediante operadores (OSR) de estado completo.

Uso:

```
st_out = qb_cc_execOSRNqb(st_in, L0p)
```

st_in: estado de entrada.

L0p: Lista con operadores de N qubits (misma dimensión que el estado).

st_out: estado de salida (siempre en el formato de matriz de densidad).

Ejemplo:

```
-->st = qb_st_state(0,2)
st =

    1.
    0.
    0.
    0.

-->L0pBF = qb_err_OSR_BitFlip(0.9);
-->L0p_out = qb_err_OSR_m2nQbs(L0pBF, 1, 2);
```

```
-->rho_out = qb_cc_execOSRNqb(st, L0p_out)
rho_out =
```

```
0.9    0.    0.    0.
0.     0.    0.    0.
0.     0.    0.1   0.
0.     0.    0.    0.
```

A.6. Funciones complementarias (útiles)

Esta biblioteca consta de 13 rutinas utilitarias para otras funciones del simulador. Todas las rutinas empiezan con el prefijo `qb_ut_`.

▪ <code>qb_ut_bmatrix.</code>	▪ <code>qb_ut_CombinaPauli.</code>
▪ <code>qb_ut_OrdenaInd.</code>	▪ <code>qb_ut_Bloch.</code>
▪ <code>qb_ut_InvierteInd.</code>	▪ <code>qb_ut_BiParticion.</code>
▪ <code>qb_ut_tensorial.</code>	▪ <code>qb_ut_Spec.</code>
▪ <code>qb_ut_cambia_filas.</code>	▪ <code>qb_ut_sqrtmSD.</code>
▪ <code>qb_ut_coordXYZ.</code>	▪ <code>qb_ut_logmSD.</code>
▪ <code>qb_ut_Stokespars.</code>	

A.6.1. `qb_ut_bmatrix`

Crea una matriz con $2^{\text{cant_b}}$ números binarios ordenados por fila.

Uso:

```
Mb = qb_ut_bmatrix(cant_b)
```

`cant_b`: Cantidad de bits.

`Mb`: Matriz de salida.

Ejemplo:

```
-->M = qb_ut_bmatrix(3)
M =
```

```
0.    0.    0.
0.    0.    1.
0.    1.    0.
0.    1.    1.
1.    0.    0.
1.    0.    1.
1.    1.    0.
1.    1.    1.
```

A.6.2. qb_ut_OrdenaInd

Ordena el vector de índices, evitando repeticiones. Similar a la función *unique* pero con control de límites superior e inferior.

Uso:

```
[ind_out, res] = qb_ut_OrdenaInd(ind_in [, ind_max])
```

ind_in: Vector de índices de entrada.
ind_max: Valor máximo permitido para los índices. (Por defecto %inf).
Verifica que no hayan índices inferiores a 1.

ind_out: Vector de índices resultante.
res: Resultado booleano que indica si hay índices superiores al máximo,
o inferiores a 1 (%T indica índices correctos).

Ejemplo:

```
-->ind = [3 1 2 1 2 4 8 6];
-->[ind_out, res] = qb_ut_OrdenaInd(ind)
res =

    T
ind_out =

    1.    2.    3.    4.    6.    8.

-->[ind_out, res] = qb_ut_OrdenaInd(ind, 7)
res =

    F
ind_out =

    1.    2.    3.    4.    6.    8.
```

A.6.3. qb_ut_InvierteInd

Invierte el vector de índices, dejando los no existentes en el vector de entrada.

Uso:

```
[ind_out, res] = qb_ut_InvierteInd(ind_in [, ind_max])
```

ind_in: Vector de índices de entrada.
ind_max: Valor máximo permitido para los índices.
(Por defecto máximo en ind_in).

ind_out: Vector de índices resultante. Si res es %F la salida es [].
res: Resultado booleano de la verificación de ind_in.

Ejemplo:

```
-->ind = [1 3 5 7];
-->[ind_out, res] = qb_ut_InvierteInd(ind)
res =

    T
ind_out =

    2.    4.    6.

-->[ind_out, res] = qb_ut_InvierteInd(ind,9)
res =

    T
ind_out =

    2.    4.    6.    8.    9.

-->[ind_out, res] = qb_ut_InvierteInd(ind,5)
res =

    F
ind_out =

    []
```

A.6.4. qb_ut_tensorial

Producto tensorial de dos matrices. Necesario para matrices sparse.

Uso:

```
M = qb_ut_tensorial(M1, M2)
```

M1, M2: Matrices de entrada.

M: Matriz que define el sistema cuántico

Ejemplo:

```
-->M1 = qb_cmp_I(1);
-->M2= qb_cmp_X(1);
-->M = qb_ut_tensorial(M1,M2)
M =

    0.    1.    0.    0.
    1.    0.    0.    0.
    0.    0.    0.    1.
    0.    0.    1.    0.
```

A.6.5. qb_ut_cambia_filas

Crea una descripción matricial de un sistema (circuito) con los qubits i, j intercambiados.

Uso:

```
NC = qb_ut_cambia_filas(C, i, j)
```

C: Matriz original del operador

i,j: qubits a intercambiar

NC: Matriz del operador con los qubits cambiados

Ejemplo:

```
-->M1 = qb_cmp_I(1);
-->M2= qb_cmp_X(1);
-->MS1 = qb_ut_tensorial(M1,M2)
MS1 =

    0.    1.    0.    0.
    1.    0.    0.    0.
    0.    0.    0.    1.
    0.    0.    1.    0.

-->MS2 = qb_ut_tensorial(M2,M1)
MS2 =

    0.    0.    1.    0.
    0.    0.    0.    1.
    1.    0.    0.    0.
    0.    1.    0.    0.

-->MS3 = qb_ut_cambia_filas(MS1,1,2)
MS3 =

    0.    0.    1.    0.
    0.    0.    0.    1.
    1.    0.    0.    0.
    0.    1.    0.    0.
```

A.6.6. qb_ut_cambia_filasV

Crea una descripción vectorial de un estado con los qubits i, j intercambiados.

Uso:

```
st_out = qb_ut_cambia_filasV(st_in,i,j)
```

st_in: estado a ser cambiado

i,j: qubits a intercambiar

st_out: estado de salida

Ejemplo:

```
-->st = qb_st_state([0 1], 2, [1 1])
st =

    0.7071068
    0.7071068
    0.
    0.

-->qb_st_lstprint(st)

Estado puro

coef = 0.7071068   base_vec = 0 [00]

coef = 0.7071068   base_vec = 1 [01]

-->rho = qb_st_vec2dens(st)
rho =

    0.5    0.5    0.    0.
    0.5    0.5    0.    0.
    0.     0.     0.    0.
    0.     0.     0.    0.

-->st2 = qb_ut_cambia_filasV(st, 1, 2)
st2 =

    0.7071068
    0.
    0.7071068
    0.

-->qb_st_lstprint(st2)

Estado puro

coef = 0.7071068   base_vec = 0 [00]

coef = 0.7071068   base_vec = 2 [10]

-->rho2 = qb_st_vec2dens(st2)
rho2 =

    0.5    0.    0.5    0.
    0.     0.     0.    0.
    0.5    0.    0.5    0.
    0.     0.     0.    0.
```



```
-->qb_ut_cambia_filas(rho, 1, 2)
ans =
```

```
    0.5    0.    0.5    0.
    0.    0.    0.    0.
    0.5    0.    0.5    0.
    0.    0.    0.    0.
```

A.6.7. qb_ut_coordXYZ

Calcula la ubicación del estado (o estados) en la esfera de Bloch.

Uso:

```
[coordXYZ, coord_esf] = qb_ut_coordXYZ(sts, [op])
```

sts: n estados de 1 qubit (2 filas x n columnas).

op: %F no calcula coord_esf (por defecto).

%T calcula coord_esf.

coordXYZ: coordenadas cartesianas de salida.

coord_esf: coordenadas esféricas de salida.

Ejemplo:

```
-->sts = qb_st_rnd_vecstates(3,1)
sts =
```

```
    0.7425495    0.3602378    0.6674325
    0.6295441 - 0.2286798i    0.4704949 + 0.8055205i    - 0.3294101 - 0.6678494i
```

```
-->[coordXYZ, coord_esf] = qb_ut_coordXYZ(sts, %T)
coord_esf =
```

```
    1.    1.    1.
    1.4678552    2.404547    1.6800815
    5.9347585    1.042159    4.2541639
coordXYZ =
```

```
    0.9349353    0.3389801 - 0.4397180
- 0.3396121    0.5803579 - 0.8914888
    0.1027594 - 0.7404574 - 0.1090678
```

```
-->rhos = qb_st_rnd_denstates(3,1)
rhos =
```

```
rhos(1)
```

```
    0.720336    - 0.1235905 - 0.1264548i
- 0.1235905 + 0.1264548i    0.279664
```

```

    rhos(2)

    0.4610491          - 0.0519855 + 0.0769200i
- 0.0519855 - 0.0769200i    0.5389509

    rhos(3)

    0.4486254          - 0.0660978 - 0.1462354i
- 0.0660978 + 0.1462354i    0.5513746

-->[coordXYZ, coord_esf] = qb_ut_coordXYZ(rhos, %T)
coord_esf =

    0.5650252    0.2013590    0.3370049
    0.6762653    1.9680427    1.8806184
    2.3447396    4.1180639    1.9953088
coordXYZ =

- 0.2471809 - 0.1039710 - 0.1321956
 0.2529096 - 0.1538401  0.2924709
 0.4406720 - 0.0779019 - 0.1027491

```

A.6.8. qb_ut_Stokespars

Calcula los parámetros de Stokes de un estado cuántico.

Uso:

```
[res, vec, Matres] = qb_ut_Stokespars(sts)
```

sts: Estados cuánticos de entrada

res: Parámetros de Stokes

vec: Vector asociado (parámetros sin el primer elemento)

Matres: Matrices de combinaciones de las matrices de Pauli (I, X, Y, Z)
con la cantidad de qubits (sts) adecuada (2 qubits: II, IX, ..., ZZ).

En caso de ingresar más de un estado cuántico res y vec contienen una matriz con el resultado de cada caso en una columna.

Ejemplo:

```

-->sts = qb_st_state([0 1], 1)
sts =

    1.    0.
    0.    1.

```

```
-->[res, vec, Matres] = qb_ut_Stokespars(st)
Matres =
```

```
Matres(1)
```

```
1.    0.
0.    1.
```

```
Matres(2)
```

```
0.    1.
1.    0.
```

```
Matres(3)
```

```
0    - i
i     0
```

```
Matres(4)
```

```
1.    0.
0.   - 1.
```

```
vec =
```

```
0.    0.
0.    0.
1.   - 1.
```

```
res =
```

```
1.    1.
0     0
0     0
1.   - 1.
```

A.6.9. qb_ut_CombinaPauli

Devuelve una lista con la combinación de las matrices de Pauli para n qubits.

Uso:

```
lPauli = qb_ut_CombinaPauli(c_qbts)
```

c_qbts: cantidad de qubits.

lPauli: Lista con matrices resultantes de las combinaciones ordenadas de las matrices de Pauli (considera la identidad).

Ejemplo:

```
-->LP = qb_ut_CombinaPauli(2);
```

```
-->length(LP)
ans =

    16.

-->full(LP(8))
ans =

    0.    0.    1.    0.
    0.    0.    0.   - 1.
    1.    0.    0.    0.
    0.   - 1.    0.    0.

-->qb_ut_tensorial(qb_cmp_X(1),qb_cmp_Z(1))
ans =

    0.    0.    1.    0.
    0.    0.    0.   - 1.
    1.    0.    0.    0.
    0.   - 1.    0.    0.
```

A.6.10. qb_ut_Bloch

Representa estados de 1 qubit en la esfera de Bloch. Estados puros son representados por puntos negros, mientras las mezclas por puntos blancos.

Uso:

```
XYZ = qb_ut_Bloch(sts [, ops])
```

sts: Estados de 1 qubit.

ops: Booleana. Si %F no muestra la esfera (solo los estados).
Por defecto %T.

XYZ: Coordenadas cartesianas de los estados.

Ejemplo:

```
-->sts = qb_st_rnd_vecstates(500,1);
-->qb_ut_Bloch(sts);
-->qb_ut_Bloch(sts, %F);
-->rhos = qb_st_rnd_denstates(500,1);
-->qb_ut_Bloch(rhos);
-->qb_ut_Bloch(rhos, %F);
```

En las ilustraciones de las figuras A.13 y A.14 es posible observar los resultados del ejemplo.

(a) 500 estados puros en la esfera de Bloch. Se ilustra la esfera.

(b) 500 estados puros en la esfera de Bloch. Sin el contorno de la esfera.

Figura A.13: Ejemplos de la esfera de Bloch para estados puros.

(a) 500 estados mezclas en la esfera de Bloch. Se ilustra la esfera.

(b) 500 estados mezclas en la esfera de Bloch. Sin el contorno de la esfera.

Figura A.14: Ejemplos de la esfera de Bloch para estados mezclas.

A.6.11. qb_ut_BiParticion

Calcula todas las particiones posibles de n qubits (sin repeticiones).

Uso:

`Lres = qb_ut_BiParticion(n)`

`n`: Cantidad de qubits.

`Lres`: Lista con todas las particiones posibles. Cada elemento es una lista con 2 vectores indicando los índices de la partición.

Ejemplo:

```
-->Lres = qb_ut_BiParticion(3)
Lres =
```

```
Lres(1)
```

```
Lres(1)(1)
```

```
1. 3.
```

```
Lres(1)(2)
```

```
2.
```

```
Lres(2)
```

```
Lres(2)(1)
```

```
1.
```

```
Lres(2)(2)
```

```
2. 3.
```

```
Lres(3)
```

```
Lres(3)(1)
```

```
1. 2.
```

```
Lres(3)(2)
```

```
3.
```

A.6.12. qb_ut_Spec

Calcula los autovalores y autovectores de una matriz A.

Uso:

```
[avec, aval] = qb_ut_Spec(A)
```

A: Matriz de entrada.

avec: Autovectores (matriz).

aval: Autovalores (vector, o matriz, ordenado por módulo de autovalores).

Ejemplo:

```
-->format(7); // Menos dígitos
-->rho = qb_st_rnd_denstates(1,2)
rho =

    0.3433          0.0388 - 0.0007i    0.0180 - 0.0719i    0.0577 + 0.0394i
    0.0388 + 0.0007i    0.2211          0.0116 + 0.0167i    0.0398 + 0.0631i
    0.0180 + 0.0719i    0.0116 - 0.0167i    0.2574          0.0251 + 0.0071i
    0.0577 - 0.0394i    0.0398 - 0.0631i    0.0251 - 0.0071i    0.1783

-->[avec, aval] = qb_ut_Spec(rho)
aval =

    0.4171    0    0    0
    0    0.2837    0    0
    0    0    0.1853    0
    0    0    0    0.1138
avec =

    0.8185          - 0.2277 - 0.0942i    - 0.0147 + 0.4716i    - 0.1864 - 0.1090i
    0.2533 + 0.1014i    - 0.1162 + 0.5922i    0.0080 - 0.5653i    - 0.2186 - 0.4404i
    0.1742 + 0.3381i    0.6356          0.6588          - 0.1078 + 0.0759i
    0.2949 - 0.1548i    0.3015 + 0.2833i    - 0.1531 + 0.0149i    0.8332

-->clean(avec'*avec)
ans =

    1.    0    0    0
    0    1.    0    0
    0    0    1.    0
    0    0    0    1.

-->format(10); // Valor por defecto
```

A.6.13. qb_ut_sqrtmSD

Calcula la matriz raíz cuadrada utilizando la descomposición espectral.

Uso:

```
res = qb_ut_sqrtmSD(rho)
```

rho: Matriz de entrada (normal).

res: Matriz raíz cuadrada de rho.

Ejemplo:

```
-->format(7); // Menos dígitos
```

```
-->rho = qb_st_rnd_denstates(1,2)
rho =

    0.3433          0.0388 - 0.0007i    0.0180 - 0.0719i    0.0577 + 0.0394i
    0.0388 + 0.0007i    0.2211          0.0116 + 0.0167i    0.0398 + 0.0631i
    0.0180 + 0.0719i    0.0116 - 0.0167i    0.2574          0.0251 + 0.0071i
    0.0577 - 0.0394i    0.0398 - 0.0631i    0.0251 - 0.0071i    0.1783

-->srqt_rho = qb_ut_sqrtmSD(rho)
srqt_rho =

    0.5766          0.0334 + 0.0025i    0.0148 - 0.0681i    0.0567 + 0.0394i
    0.0334 - 0.0025i    0.4612          0.0103 + 0.0178i    0.0432 + 0.0706i
    0.0148 + 0.0681i    0.0103 - 0.0178i    0.5013          0.0277 + 0.0030i
    0.0567 - 0.0394i    0.0432 - 0.0706i    0.0277 - 0.0030i    0.4072

-->clean(srqt_rho^2)
ans =

    0.3433          0.0388 - 0.0007i    0.0180 - 0.0719i    0.0577 + 0.0394i
    0.0388 + 0.0007i    0.2211          0.0116 + 0.0167i    0.0398 + 0.0631i
    0.0180 + 0.0719i    0.0116 - 0.0167i    0.2574          0.0251 + 0.0071i
    0.0577 - 0.0394i    0.0398 - 0.0631i    0.0251 - 0.0071i    0.1783

-->format(10); // Valor por defecto
```

A.6.14. qb_ut_logmSD

Calcula la matriz logarítmica utilizando la descomposición espectral.

Uso:

```
res = qb_ut_logmSD(rho)
```

rho: Matriz de entrada (normal).

res: Matriz logarítmica de rho.

Ejemplo:

```
-->format(7); // Menos dígitos
-->rho = qb_st_rnd_denstates(1,2)
rho =

    0.3433          0.0388 - 0.0007i    0.0180 - 0.0719i    0.0577 + 0.0394i
    0.0388 + 0.0007i    0.2211          0.0116 + 0.0167i    0.0398 + 0.0631i
    0.0180 + 0.0719i    0.0116 - 0.0167i    0.2574          0.0251 + 0.0071i
    0.0577 - 0.0394i    0.0398 - 0.0631i    0.0251 - 0.0071i    0.1783
```



```
-->log_rho = qb_ut_logmSD(rho)
log_rho =

- 1.139          0.1124 + 0.0232i    0.0483 - 0.2626i    0.2310 + 0.1624i
  0.1124 - 0.0232i - 1.588          0.0370 + 0.0739i    0.1932 + 0.3249i
  0.0483 + 0.2626i    0.0370 - 0.0739i - 1.4047          0.1246 - 0.0047i
  0.2310 - 0.1624i    0.1932 - 0.3249i    0.1246 + 0.0047i - 1.8611

-->clean(expm(log_rho))
ans =

  0.3433          0.0388 - 0.0007i    0.0180 - 0.0719i    0.0577 + 0.0394i
  0.0388 + 0.0007i    0.2211          0.0116 + 0.0167i    0.0398 + 0.0631i
  0.0180 + 0.0719i    0.0116 - 0.0167i    0.2574          0.0251 + 0.0071i
  0.0577 - 0.0394i    0.0398 - 0.0631i    0.0251 - 0.0071i    0.1783

-->format(10); // Valor por defecto
```

A.7. Funciones para cálculo de distancias y correlaciones cuánticas

Esta biblioteca consta de 6 rutinas para el cálculo de distancias entre estados cuánticos y correlaciones. Todas las rutinas empiezan con el prefijo `qb_corr_`.

■ <code>qb_corr_EntRho.</code>	■ <code>qb_corr_DistHS.</code>
■ <code>qb_corr_DistTrace.</code>	■ <code>qb_corr_DistMI.</code>
■ <code>qb_corr_DistFid.</code>	■ <code>qb_corr_CCM.</code>

A.7.1. `qb_corr_EntRho`

Calcula la entropía de un estado cuántico.

Uso:

```
resEnt = qb_corr_EntRho(rho)
```

`rho`: Estado cuántico.

`resEnt`: Entropía de `rho`.

Ejemplo:

```
-->st = qb_st_rnd_vecstates(1,3); // Estado puro
-->rhoP = qb_st_vec2dens(st);
```

```
-->res = qb_corr_EntRho(rhoP)
res =

    0.

-->rhoM = qb_st_rnd_denstates(1,3); // Estado mezcla
-->res = qb_corr_EntRho(rhoM)
res =

    2.2843494
```

A.7.2. qb_corr_DistTrace

Calcula la distancia entre dos estados cuánticos usando la traza.

Uso:

```
[res, tr] = qb_corr_DistTrace(rho1, rho2)

rho1, rho2: Estados cuánticos (matrices de densidad).

res: Distancia normalizada para particiones de GHZ.
tr: Distancia utilizando la traza.
```

Ejemplo:

```
-->st= qb_st_state([0 7], 3, [1 1]); // GHZ
-->rho =qb_st_vec2dens(st)
rho =

    0.5    0.    0.    0.    0.    0.    0.    0.5
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.5    0.    0.    0.    0.    0.    0.    0.5

-->rho1 = qb_st_partialtrace(rho, [1 2]);
-->rho2 = qb_st_partialtrace(rho, [3]);
-->rhoF = qb_ut_tensorial(rho1,rho2)
rhoF =

    0.25    0.    0.    0.    0.    0.    0.    0.
    0.    0.25    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.25    0.
    0.    0.    0.    0.    0.    0.    0.    0.25
```

```
-->[res, tr] = qb_corr_DistTrace(rho, rhoF)
tr =

    1.5
res =

    1.
```

A.7.3. qb_corr_DistFid

Calcula la distancia entre dos estados cuánticos usando la Fidelidad.

Uso:

```
[res, fid] = qb_corr_DistFid(rho1, rho2)

rho1, rho2: Estados cuánticos (matrices de densidad).

res: Distancia (acos(fid)) normalizada para particiones de GHZ.
fid: Fidelidad entre los estados.
```

Ejemplo:

```
-->st= qb_st_state([0 7], 3, [1 1]); // GHZ
-->rho =qb_st_vec2dens(st)
rho =

    0.5    0.    0.    0.    0.    0.    0.    0.5
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.5    0.    0.    0.    0.    0.    0.    0.5

-->rho1 = qb_st_partialtrace(rho, [1 2]);
-->rho2 = qb_st_partialtrace(rho, [3]);
-->rhoF = qb_ut_tensorial(rho1,rho2)
rhoF =

    0.25    0.    0.    0.    0.    0.    0.    0.
    0.    0.25    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.25    0.
    0.    0.    0.    0.    0.    0.    0.    0.25

-->[res, fid] = qb_corr_DistFid(rho, rhoF)
fid =

    0.5
```

```
res =
```

```
1.
```

A.7.4. qb_corr_DistHS

Calcula la distancia entre dos estados cuánticos usando la distancia de Hilbert-Smith.

Uso:

```
[res, hs] = qb_corr_DistHS(rho1, rho2)
```

rho1, rho2: Estados cuánticos (matrices de densidad).

res: res: Distancia normalizada para particiones de GHZ.

hs: Distancia de HS.

Ejemplo:

```
-->st= qb_st_state([0 7], 3, [1 1]); // GHZ
```

```
-->rho =qb_st_vec2dens(st)
```

```
rho =
```

```
0.5  0.  0.  0.  0.  0.  0.  0.5
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.5  0.  0.  0.  0.  0.  0.  0.5
```

```
-->rho1 = qb_st_partialtrace(rho, [1 2]);
```

```
-->rho2 = qb_st_partialtrace(rho, [3]);
```

```
-->rhoF = qb_ut_tensorial(rho1,rho2)
```

```
rhoF =
```

```
0.25  0.  0.  0.  0.  0.  0.  0.
0.   0.25  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.  0.
0.   0.  0.  0.  0.  0.  0.25  0.
0.   0.  0.  0.  0.  0.  0.  0.25
```

```
-->[res, hs] = qb_corr_DistHS(rho, rhoF)
```

```
hs =
```

```
0.75
```

```
res =
```

```
1.
```

A.7.5. qb_corr_DistMI

Calcula la información mutua entre las dos partes (bipartición) de un estado rho.

Uso:

```
[res, mi] = qb_corr_DistMI(rho, rhoA, rhoB)
```

rho: Estado cuántico original (matriz de densidad).

rhoA, rhoB: Bipartición de rho.

res: Distancia normalizada para particiones de GHZ.

mi: Información mutua entre rho y rhoA x rhoB

Ejemplo:

```
-->st= qb_st_state([0 7], 3, [1 1]); // GHZ
-->rho =qb_st_vec2dens(st)
rho =

    0.5    0.    0.    0.    0.    0.    0.    0.5
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.
    0.5    0.    0.    0.    0.    0.    0.    0.5

-->rho1 = qb_st_partialtrace(rho, [1 2])
rho1 =

    0.5    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.
    0.    0.    0.    0.5

-->rho2 = qb_st_partialtrace(rho, [3])
rho2 =

    0.5    0.
    0.    0.5

-->[res, mi] = qb_corr_DistMI(rho, rho1, rho2)
mi =

    2.
res =

    1.
```

A.7.6. qb_corr_CCM

Calcula la correlación acumulativa (CCM) del estado rho.

Uso:

```
[res, n_part, tot_res] = qb_corr_CCM(st_in [, ops])
```

st_in: Estado cuántico.

ops: Indica la métrica (distancia) que se utiliza.

1: Fidelidad.

2: Hilbert-Smith

3: Basada en la traza.

4: Informacion mutua (por defecto).

res: Índice de correlación. (resultado)

n_part: Lista con la partición utilizada (mínimo).

tot_res: Lista con los todos los resultados y las respectivas particiones.

En caso de que st_in tenga más de un estado, las salidas anteriores son listas con los respectivos resultados.

Ejemplo:

```
-->st= qb_st_state([0 3], 2, [1 1]); // GHZ, 2qbs
```

```
-->res = qb_corr_CCM(st)
```

```
res =
```

```
1.
```

```
-->st= qb_st_state([0 7], 3, [1 1]); // GHZ, 2qbs
```

```
-->res = qb_corr_CCM(st)
```

```
res =
```

```
2.5
```

A.8. Funciones para el cálculo de la isotropía cuántica

Esta biblioteca consta de 2 rutinas para el cálculo de índices de isotropía. Todas las rutinas empiezan con el prefijo qb_iso_.

- qb_iso_GeomIndex_vec1qb.

- qb_iso_PGraf.

- qb_iso_EstIndex.

A.8.1. qb_iso_GeomIndex_vec1qb

Calcula el índice de isotropía geométrico para estados de 1 qubit en formato vectorial.

Uso:

```
[res, vec_k_theta, vec_k_phi] = qb_iso_GeomIndex_vec1qb(sts_in, pars)
```

sts_in: Estados cuánticos de entrada (puros y vectoriales)

pars: Parámetros para el cómputo de los índices [inc_ang, delta_cono]

inc_ang: incremento angular (en phi) para los cálculos.
Por defecto igual a $\pi/36$.

delta_cono: define cuántos sectores adjacentes considerar en el cálculo de cada ángulo.
1 <= delta_cono <= 5. Por defecto igual a 3.

res: Índice isotrópico compuesto [res_phi res_theta]

vec_k_phi: vector con el índice de distribución de estados por sectores (deltas phi)

vec_k_theta: vector con el índice de separación de estados por sectores (deltas phi)

Ejemplo:

```
-->st = qb_st_state(0);
-->MError = qb_err_contIsoind(1, 1, [1; 0.2], 10000);
-->st_out = qb_cc_execCError(st, MError);
-->res = qb_iso_GeomIndex_vec1qb(st_out)
res =

    0.9219057    0.9410395

-->MError2 = qb_err_contXYZind(1, 1, [1 1 1; 0.2 0.2 0.2], 10000);
-->st_out2 = qb_cc_execCError(st, MError2);
-->res = qb_iso_GeomIndex_vec1qb(st_out2)
res =

    0.7846830    0.7072271
```

La figura A.15 ilustra los ensambles de estados generados para el ejemplo.

(a) 10000 de estados de error generados mediante *qb_err_contIsoind*.

(b) 10000 de estados de error generados mediante *qb_err_contXYZind*.

Figura A.15: Estados generados para el ejemplo A.8.1.

A.8.2. *qb_iso_EstIndex*

Calcula el índice de isotropía estadístico de un conjunto de estados respecto a un estado referencia.

Uso:

```
[res, rhoNiso] = qb_iso_EstIndex(sts_in, st_ref [, tbo1])
```

sts_in: Estados cuánticos de entrada (ruido). Se suponen equiprobables.

st_ref: Estado de referencia. Por defecto el estado $|0\rangle$ de la base canónica.

Debe ser único y en formato vectorial.

tbo1: Si es %F no calcula la desviación (*res*(2)) (por defecto %T).

res: Índice isotrópico estadístico (conjunto de índices).

res(1): Ancho isotrópico (probabilidad de la componente isotrópica).

res(2): Alineación de la componente no isotrópica [-1,1].

rhoNiso: Estado con la componente no isotrópica (si no está alineado).

Ejemplo:

```
-->st = qb_st_state(0);
-->MError = qb_err_contIsoind(1, 1, [1; 0.2], 5000);
-->st_out = qb_cc_execCError(st, MError);
-->rho = qb_st_st2ldens(st_out)
rho =

    0.9933702          - 0.0001404 + 0.0018735i
    - 0.0001404 - 0.0018735i    0.0066298

-->[res, rhoNiso] = qb_iso_EstIndex(rho, st)
rhoNiso =

    0.9999964          - 0.0001423 + 0.0018986i
    - 0.0001423 - 0.0018986i    0.0000036
```


res =

0.0132525 0.9980942

A.8.3. qb_iso_PGraf

Representa el índice de isotropía estadístico utilizando el *triangulo de isotropía*.

Uso:

qb_iso_PGraf(resIsoE)

resIsoE: Índices de isotropía estadístico.

En el caso de una matriz dibuja una sola curva

Ejemplo:

```
-->st = qb_st_state(0);

// Dos estados al azar

-->rhos = qb_st_rnd_denstates(2, 1, [%F 4]);

// Ejemplo para una curva

-->rhoL1=list();
-->for i=1:10;
-->    aux = qb_err_ISO_DepChannel(rhos(1), i*0.1);
-->    rhoL1($+1) = aux;
-->end;

-->res1=[];
-->for i=1:10;
-->    auxres = qb_iso_ProbIndex(rhoL1(i), st0);
-->    res1 = [res1;auxres];
-->end;

-->qb_iso_PGraf(res1);

// Ejemplo para múltiples curvas

-->rhoL2=list();

-->for i=1:10;
-->    L0p = qb_err_OSR_BitFlip(i*0.1);
-->    aux = qb_cc_execOSR1qb(rhos(2), 1, L0p);
-->    rhoL2($+1) = aux;
-->end;

-->res21 = res1;
-->res22 = [];
-->for i=1:10;
-->    auxres = qb_iso_ProbIndex(rhoL2(i), st0);
-->    res22 = [res22;auxres];
```

```
-->end;

-->res = list(res21, res22);
-->qb_iso_PGraf(res);
```

(a) Triángulo de isotropía. Una curva.

(b) Triángulo de isotropía. Múltiples curvas.

Figura A.16: Triángulos de isotropía para el ejemplo A.8.3.

Apéndice B

Cálculos y demostraciones

B.1. Propiedades de la CCM

En esta sección se demuestran las propiedades de la medida de correlación cumulativa anunciadas en la sección 3.2.3.

B.1.1. Propiedad 1

Demostración. Por la definición de distancia se verifica que $D(\rho, \sigma) \geq 0$ para todos los estados ρ y σ . Entonces, considerando la definición recursiva de la medida, y que todos los términos sumados son positivos (o nulos), se obtiene que $C(\rho) \geq 0$ para todo estado ρ . $\square$

B.1.2. Propiedad 2

Demostración. Esta propiedad es consecuencia directa de la propiedad 6 (la correlación es aditiva) y de que la correlación vale cero para estados de un qubit. $\square$

B.1.3. Propiedad 3

Demostración. Sea el estado $\rho' = U\rho U^\dagger$, definido mediante una transformación unitaria local $U = U^1 \otimes \dots \otimes U^N$ del estado ρ . En este caso la correlación del nuevo estado vale

$$C(\rho') = \min_{\{k\}} [2^{N-2} D(\rho', \rho'_{A_k} \otimes \rho'_{B_k}) + C(\rho'_{A_k}) + C(\rho'_{B_k})], \quad (\text{B.1})$$

siendo ρ'_{A_k} y ρ'_{B_k} la k -ésima bipartición del estado transformado. Utilizando esta bipartición (con los qubits convenientemente reordenados) y considerando $U = U_{A_k} \otimes U_{B_k}$ se tiene que

$$\rho'_{A_k} = \text{Tr}_{B_k}(\rho') = U_{A_k} \rho_{A_k} U_{A_k}^\dagger, \quad \rho'_{B_k} = \text{Tr}_{A_k}(\rho') = U_{B_k} \rho_{B_k} U_{B_k}^\dagger, \quad (\text{B.2})$$

lo que implica

$$\rho'_{A_k} \otimes \rho'_{B_k} = U_{A_k} \rho_{A_k} U_{A_k}^\dagger \otimes U_{B_k} \rho_{B_k} U_{B_k}^\dagger = U_k (\rho_{A_k} \otimes \rho_{B_k}) U_k^\dagger. \quad (\text{B.3})$$

Entonces, por la propiedad de invariancia de la distancia (observación 3.2 en 3.2.2)

$$D(\rho', \rho'_{A_k} \otimes \rho'_{B_k}) = D(\rho_k, \rho_{A_k} \otimes \rho_{B_k}). \quad (\text{B.4})$$

Por inducción finita es simple ver que $C(\rho'_{A_k}) = C(\rho_{A_k})$ y $C(\rho'_{B_k}) = C(\rho_{B_k})$. Luego CCM es invariante frente a operaciones unitarias locales. $\square$

B.1.4. Propiedad 4

Demostración. Esta propiedad es consecuencia inmediata de las propiedades 2 y 6. $\square$

B.1.5. Propiedad 5

Demostración. Sea el operador cuántico formado por operaciones locales $\varepsilon(\rho)$, $\varepsilon = \varepsilon^1 \otimes \dots \otimes \varepsilon^N$. La correlación de un estado $\varepsilon(\rho)$, resultado de la aplicación de la operación local, es

$$C(\varepsilon(\rho)) = \min_{\{k\}} [2^{N-2} D(\varepsilon(\rho), \sigma_{A_k} \otimes \sigma_{B_k}) + C(\sigma_{A_k}) + C(\sigma_{B_k})], \quad (\text{B.5})$$

siendo $\varepsilon = \varepsilon_{A_k} \otimes \varepsilon_{B_k}$, $\sigma_{A_k} = \text{Tr}_{B_k}(\varepsilon(\rho))$ y $\sigma_{B_k} = \text{Tr}_{A_k}(\varepsilon(\rho))$.

Mediante el tensor de Stokes [69] se tiene que $\sigma_{A_k} \otimes \sigma_{B_k} = \varepsilon(\rho_{A_k} \otimes \rho_{B_k})$, con $\rho_{A_k} = \text{Tr}_{B_k}(\rho)$ y $\rho_{B_k} = \text{Tr}_{A_k}(\rho)$. Utilizando la propiedad de contracción de la medida (observación 3.2 en 3.2.2) se obtiene

$$\begin{aligned} D(\varepsilon(\rho), \sigma_{A_k} \otimes \sigma_{B_k}) &= D(\varepsilon(\rho), \varepsilon(\rho_{A_k} \otimes \rho_{B_k})) \\ &\leq D(\rho, \rho_{A_k} \otimes \rho_{B_k}). \end{aligned} \quad (\text{B.6})$$

A partir de este resultado, se prueba por inducción finita que CCM es contractiva, o sea, que

$$C(\varepsilon(\rho)) \leq C(\rho). \quad (\text{B.7})$$

Para estados de dos qubits la propiedad es consecuencia directa de la distancia entre el estado y el producto de sus matrices reducidas.

Supongamos entonces que para todo estado en una dimensión menor o igual a $N - 1$, la medida es contractiva. Sea ρ un estado de N qubits. Si se aplica un operador local se tiene que la correlación vale

$$C(\varepsilon(\rho)) = \min_{\{k\}} [2^{N-2} D(\varepsilon(\rho), \sigma_{A_k} \otimes \sigma_{B_k}) + C(\sigma_{A_k}) + C(\sigma_{B_k})]. \quad (\text{B.8})$$

Para cada partición k , por (B.6) se tiene que

$$D(\varepsilon(\rho), \sigma_{A_k} \otimes \sigma_{B_k}) \leq D(\rho, \rho_{A_k} \otimes \rho_{B_k}). \quad (\text{B.9})$$

Como σ_{A_k} y σ_{B_k} tienen menos de N qubits, aplicando la hipótesis de inducción, se obtiene

$$\begin{aligned} C(\sigma_{A_k}) &\leq C(\rho_{A_k}) \quad \text{y} \\ C(\sigma_{B_k}) &\leq C(\rho_{B_k}). \end{aligned} \quad (\text{B.10})$$

Entonces para todo N se verifica

$$C(\varepsilon(\rho)) \leq C(\rho). \quad (\text{B.11})$$

□

B.1.6. Propiedad 6

Demostración. A continuación se demuestra que CCM es aditiva.

En el caso de un estado producto de 2 qubits (una única partición posible) la demostración es inmediata:

$$C(\rho) = C(\phi \otimes \varphi) = D(\rho, \phi \otimes \varphi) + C(\phi) + C(\varphi) = 0. \quad (\text{B.12})$$

Para completar la demostración por inducción se asume, como hipótesis, que la propiedad vale para todos los estados producto con menos de N qubits. Sea entonces un estado de N qubits $\rho = \phi \otimes \varphi$. Considerando todas las posibles particiones, existen tres casos posibles:

1. El estado producto ρ se particiona en los estados ϕ y φ . Para esta partición particular, k_0 , como $D(\rho, \phi \otimes \varphi) = 0$, se tiene que

$$C_{k_0}(\rho) = C(\phi) + C(\varphi). \quad (\text{B.13})$$

2. La partición k divide el estado ρ de forma que la primera traza parcial, ρ_{A_k} , incluye completamente el estado ϕ y una parte del estado φ (el mismo procedimiento se aplica si se alternan los estados parciales). En este caso

$$\begin{aligned}\rho_{A_k} &= \text{Tr}_{B_k}(\phi \otimes \varphi) = \phi \otimes \varphi_{A'_k} \quad \text{y} \\ \rho_{B_k} &= \text{Tr}_{A_k}(\phi \otimes \varphi) = \varphi_{B_k},\end{aligned}\tag{B.14}$$

siendo A'_k y B_k una partición en el subespacio $\mathcal{H}_\varphi$ (A_k y B_k son una partición en el espacio completo). Entonces, como por hipótesis la aditividad vale para los estados con menos de N qubits,

$$\begin{aligned}C_k(\rho) &= 2^{N-2} D\left(\rho, \phi \otimes \varphi_{A'_k} \otimes \varphi_{B_k}\right) + C(\phi) + \\ &\quad + C(\varphi_{A'_k}) + C(\varphi_{B_k}).\end{aligned}\tag{B.15}$$

Para el estado φ se tiene que

$$\begin{aligned}C(\varphi) &= \min_{\{k\}} [C_k(\varphi)], \quad M < N \\ &= \min_{\{k\}} \left[2^{M-2} D\left(\varphi, \varphi_{A'_k} \otimes \varphi_{B_k}\right) + C(\varphi_{A'_k}) + C(\varphi_{B_k}) \right].\end{aligned}\tag{B.16}$$

Como la distancia es monótona (observación 3.2 en 3.2.2),

$$D\left(\varphi, \varphi_{A'_k} \otimes \varphi_{B_k}\right) \leq D\left(\rho, \phi \otimes \varphi_{A'_k} \otimes \varphi_{B_k}\right).\tag{B.17}$$

Luego,

$$\Rightarrow C_{k_0}(\rho) = C(\varphi) + C(\phi) \leq C_k(\varphi) + C(\phi) \leq C_k(\rho), \quad \forall k.\tag{B.18}$$

3. La partición k divide el estado ρ de forma a que ambos subestados ϕ y φ son biparticionados, siendo A'_k y B'_k , y A''_k y B''_k las particiones de los subespacios $\mathcal{H}_\phi$ y $\mathcal{H}_\varphi$, respectivamente. En este caso se tiene que

$$\begin{aligned}\rho_{A_k} &= \text{Tr}_{B_k}(\phi \otimes \varphi) = \phi_{A'_k} \otimes \varphi_{A''_k} \\ \rho_{B_k} &= \text{Tr}_{A_k}(\phi \otimes \varphi) = \phi_{B'_k} \otimes \varphi_{B''_k}\end{aligned}\tag{B.19}$$

y C_k es igual a

$$\begin{aligned}C_k(\rho) &= 2^{N-2} D\left(\rho, \left(\phi_{A'_k} \otimes \varphi_{A''_k}\right) \otimes \left(\phi_{B'_k} \otimes \varphi_{B''_k}\right)\right) \\ &\quad + C(\phi_{A'_k}) + C(\varphi_{A''_k}) + C(\phi_{B'_k}) + C(\varphi_{B''_k}).\end{aligned}\tag{B.20}$$

Por un procedimiento análogo al utilizado anteriormente se tiene que

$$C_{k_0} \leq C_k, \quad \forall k.\tag{B.21}$$

Luego, para un estado de N qubits la propiedad es verdadera, y por inducción la medida es aditiva.

□

B.2. Cálculo eficiente del índice de isotropía estadístico

En la sección 5.2, definición 5.2, se ha propuesto un índice de isotropía para estados cuánticos definidos por operadores de densidad. A continuación se describe una forma eficiente de realizar su cálculo.

Sea un estado cuántico ρ y un estado de referencia puro cualquiera $|\varphi\rangle$. Para el cálculo del *ancho isotrópico* solamente es necesario calcular λ , el menor de los autovalores de ρ . En este caso, $p = 2^n \lambda$.

Para el cálculo de la *alineación isotrópica* es necesario calcular las fidelidades entre $\hat{\rho}$,

$$\hat{\rho} = \frac{\rho - p \frac{I}{2^n}}{1 - p}, \quad (\text{B.22})$$

y los estados:

- de referencia $|\varphi\rangle$, $Fid(\hat{\rho}, \rho_\varphi) = \sqrt{\langle \varphi | \hat{\rho} | \varphi \rangle}$,
- y el estado $\rho_{N\varphi}$, *mezcla isotrópica ortogonal*, $Fid(\hat{\rho}, \rho_{N\varphi})$, donde

$$\rho_{N\varphi} = \frac{I - |\varphi\rangle\langle\varphi|}{2^n - 1} = \frac{I - \rho_\varphi}{2^n - 1}. \quad (\text{B.23})$$

Esta última fidelidad se calcula como

$$Fid(\hat{\rho}, \rho_{N\varphi}) = \text{tr} \sqrt{\sqrt{\rho_{N\varphi}} \hat{\rho} \sqrt{\rho_{N\varphi}}}. \quad (\text{B.24})$$

Como $\rho_{N\varphi} = U^\dagger \rho_{N0} U$, siendo U la rotación que lleva el estado $|\varphi\rangle$ al $|0\rangle$ ($|0\rangle = U|\varphi\rangle$), y $\sqrt{\rho_{N0}} = \sqrt{2^n - 1} \rho_{N0}$, entonces

$$Fid(\hat{\rho}, \rho_{N\varphi}) = \text{tr} \sqrt{\sqrt{U^\dagger \rho_{N0} U} \hat{\rho} \sqrt{U^\dagger \rho_{N0} U}}, \quad (\text{B.25})$$

$$= \sqrt{2^n - 1} \text{tr} \sqrt{U^\dagger \rho_{N0} U \hat{\rho} U^\dagger \rho_{N0} U}, \quad (\text{B.26})$$

$$= \frac{1}{\sqrt{2^n - 1}} \text{tr} \sqrt{(I - \rho_\varphi) \hat{\rho} (I - \rho_\varphi)} = \frac{1}{\sqrt{2^n - 1}} \text{tr} \sqrt{G}. \quad (\text{B.27})$$

Luego,

$$Fid(\hat{\rho}, \rho_{N\varphi}) = \frac{\sum_i \sqrt{\mu_i}}{\sqrt{2^n - 1}} \quad (\text{B.28})$$

siendo μ_i los autovalores de $G = (I - \rho_\varphi) \hat{\rho} (I - \rho_\varphi)$.

Apéndice C

Algoritmo híbrido

Debido a la gran cantidad de aplicaciones en sistemas reales, y a la diversidad de problemas existentes, han aparecido diversas propuestas de algoritmos para la búsqueda de mínimos globales en las últimas décadas [68]. La utilización de meta-heurísticas ha contribuido en la solución, en forma aproximada, de muchos problemas reales de optimización global, siendo los algoritmos genéticos (GAs) una de las técnicas que ha sido empleada en gran variedad de casos. Sin embargo, una de las limitaciones en su utilización es la lenta convergencia que implica un alto costo computacional [54].

La utilización de algoritmos híbridos combinando meta-heurísticas, en particular GAs, con métodos de convergencia local rápida ya ha sido propuesta anteriormente [62][134]. Con esta técnica se han logrado métodos de búsqueda más eficientes y con convergencia más rápida. Otro aspecto importante a mejorar en la búsqueda, utilizando GAs, es la propuesta de la población inicial.

En este apéndice se describe la propuesta de un método híbrido para la búsqueda del estado puro completamente separable más cercano a un estado puro. Este estado es utilizado en el cálculo del índice de entrelazamiento geométrico descrito en la sección 3.1. Este algoritmo consta de tres partes: un algoritmo algebraico para la inicialización de la población y un algoritmo genético que es acelerado por el método del gradiente para regresiones locales.

C.1. Algoritmo algebraico

La primera parte del algoritmo para la determinación del estado totalmente separable más cercano se realiza separando en forma iterativa los sucesivos qubits del estado según la ecuación (C.1)

$$\begin{aligned} |\hat{\psi}\rangle_1 &= |\psi_1\rangle \otimes |\psi_{2\dots n}\rangle \\ |\hat{\psi}\rangle_2 &= |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_{3\dots n}\rangle \\ &\vdots \\ |\hat{\psi}\rangle_{n-1} &= |\psi_1\rangle \otimes |\psi_2\rangle \cdots \otimes |\psi_n\rangle \end{aligned} \tag{C.1}$$

En la primera línea de la iteración anterior se busca realizar una bipartición del estado cuántico de n qubits $|\psi\rangle$ en los estados $|\psi_1\rangle$, de 1 qubit, y $|\psi_{2\dots n}\rangle$, de $n - 1$ qubits, de forma a que el estado $|\hat{\psi}\rangle_1 = |\psi_1\rangle \otimes |\psi_{2\dots n}\rangle$ sea el más cercano según la métrica. En notación vectorial (matricial) significa resolver, en cada paso, el problema conocido como el *Producto de Kroncker más Cercano* (NKP, *Nearest Kronecker Product*) [75][76]:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2n} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{2(n-1)} \end{bmatrix}. \tag{C.2}$$

Para esta primera etapa se ha adaptado el resultado existente en [77] para resolver el problema de encontrar el mínimo de

$$\| |\psi\rangle - |\hat{\psi}\rangle \|_F^2 = \| |\psi\rangle - |\psi_1\rangle \otimes |\psi_{2\dots n}\rangle \|_F^2 \quad (\text{C.3})$$

en donde $\| \cdot \|_F^2$ es la norma de Frobenius, definida como

$$\begin{aligned} \| |\psi\rangle - |\varphi\rangle \|_F &= \sqrt{\langle \chi | \chi \rangle} \\ &= \sqrt{\chi^\dagger \chi}, \quad \text{siendo} \\ |\chi\rangle &= |\psi\rangle - |\varphi\rangle. \end{aligned} \quad (\text{C.4})$$

Con la adaptación se trabaja con vectores complejos y se obtienen como resultado vectores unitarios para las representaciones de $|\psi_1\rangle$ y $|\psi_{2\dots n}\rangle$. Es importante mencionar, que el problema global no está resuelto ya que al minimizar en las biparticiones, el resultado final no tiene porque coincidir con el mínimo global buscado. Además la minimización por el método NKP utiliza la métrica de Frobenius (C.4), mientras en la tesis estamos interesados en la métrica dada por la Fidelidad (ver ecuación (3.6)).

Considerando los resultados de todas las posibles permutaciones de los qubits del estado original $|\psi\rangle$, para la descomposición iterativa propuesta en (C.1), se tienen los estados iniciales (parte de la población inicial) para la búsqueda con la segunda etapa: el algoritmo genético.

C.2. Algoritmo genético

Los algoritmos genéticos son métodos adaptativos, basados en procesos biológicos, utilizados en problemas de búsqueda y optimización [18]. Considerando poblaciones de individuos, cada uno representando una posible solución al problema, se realiza una simulación de evolución natural, mediante operadores de selección, cruce y mutación. Habitualmente se utiliza una codificación de los puntos del espacio de búsqueda [129].

En este trabajo se propone utilizar como base el algoritmo genético propuesto en [16]. Es un algoritmo con codificación real y operador de cruce basado en morfología matemática, que ha demostrado buen desempeño en problemas complejos [15].

Como se menciona anteriormente se utilizan como parte de la población inicial los resultados de la solución algebraica iterativa. Con la finalidad de mejorar el espacio de búsqueda se agrega al azar una cantidad (programable) de elementos a esta población. Cada individuo en la población es un estado cuántico totalmente separable que es candidato a solución del problema.

En este punto empiezan los pasos del algoritmo evolutivo durante una cantidad determinada de generaciones. A cada procedimiento completo se le denomina una generación o época.

1. Para cada elemento de población se calcula su índice de adaptación (*fitness*) a la función objetivo (en este caso la fidelidad con el estado objeto de estudio).
2. Mediante un algoritmo de selección se elige una cantidad predefinida de individuos para el cruce. En el algoritmo propuesto se utiliza el sorteo al azar, utilizando el *fitness* como ponderador en la probabilidad de elección. Esto garantiza una mayor probabilidad de elección de los más adaptados, sin descartar la posibilidad de utilizar otros individuos.
3. Según el algoritmo propuesto se realiza el cruce considerando los genes de cada elemento seleccionado, lo que produce dos individuos nuevos. Estos individuos son agregados a la población aumentando en dos el número total.
4. Se vuelve a calcular el índice de adaptación de los individuos agregados y, considerando los ya calculados, se eliminan dos individuos, manteniendo así el número total constante.
5. Con la finalidad de incrementar la velocidad de convergencia en la búsqueda del más adaptado (algoritmo interpretado como optimizador), se realiza una optimización local utilizando como punto inicial al individuo con mejor índice de adaptación (ver C.3). Esto no se realiza en

cada generación, sino que se encuentra predefinida la cantidad de generaciones que se deben calcular solamente con el algoritmo genético. Por defecto la relación es de uno a diez (nueve pasos solamente con el genético y uno completo, o sea, con búsqueda local).

6. Se verifica la terminación de la cantidad de generaciones a calcular. Si faltan se vuelve al primer paso, sino se finaliza el algoritmo. Al finalizar, el individuo con mejor índice de adaptación es el elegido.

C.3. Optimización con gradiente

Con la finalidad de acelerar la convergencia local se ha incorporado al algoritmo un método de optimización local basado en derivadas [79]. Esta búsqueda se realiza utilizando como punto de partida el óptimo de la población (individuo con mejor adaptación) cada n generaciones (n programable, por defecto igual a 10).

En la implementación se ha utilizado uno de los algoritmos más clásicos en el área de las *Redes Neronales Artificiales* [52], que utiliza como dirección de búsqueda la contraria al gradiente en el punto, un paso de búsqueda adaptable y un término de momento (para evitar algunos de los mínimos locales y mejorar la convergencia). El algoritmo es el indicado en la ecuación (C.5).

$$\begin{aligned} X(k+1) &= X(k) + \eta(k)\vec{d} + \alpha\Delta(k) \\ \Delta(k) &= X(k) - X(k-1) \end{aligned} \tag{C.5}$$

siendo $\eta(k)$ el paso (adaptable) en la dirección indicada por $\vec{d}$ (contraria al gradiente), y α el término que pondera la influencia del cambio anterior.

El resultado sustituye el valor original (individuo original), lo que lleva a una rápida convergencia local.

Bibliografía

- [1] S. AARONSON AND A. AMBAINIS, *Quantum search of spatial regions*, in Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on, Oct 2003, pp. 200–209.
- [2] S. AARONSON AND D. GOTTESMAN, *Improved simulation of stabilizer circuits*, Phys. Rev. A, 70 (2004), p. 052328.
- [3] G. ABAL, R. DONANGELO, F. MARQUEZINO, A. OLIVEIRA, AND R. PORTUGAL, *Decoherence in search algorithms*, in Proceedings of the XXIX Brazilian Computer Society Congress (SEMISH), 2009, pp. 293–306.
- [4] D. AHARONOV AND M. BEN-OR, *Fault-tolerant quantum computation with constant error*, in Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, STOC '97, New York, NY, USA, 1997, ACM, pp. 176–188.
- [5] F. ALTINTAS AND R. ERYIGIT, *Correlation and nonlocality measures as indicators of quantum phase transitions in several critical systems*, Annals of Physics, 327 (2012), pp. 3084–3101.
- [6] A. AMARIUTEI AND S. CARAIMAN, *Parallel quantum computer simulation on the gpu*, in 15th International Conference on System Theory, Control, and Computing (ICSTCC), IEEE, October 2011, pp. 1–6.
- [7] A. AMBAINIS, A. BACKURS, N. NAHIMOV, AND A. RIVOSH, *Grover's algorithm with errors*, in Mathematical and Engineering Methods in Computer Science, A. Kucera, T. Henzinger, J. Nešetřil, T. Vojnar, and D. Antos, eds., vol. 7721 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 180–189.
- [8] A. AMBAINIS, J. KEMPE, AND A. RIVOSH, *Coins make quantum walks faster*, in Proceedings of SODA 2005, pp. 1099–1108, 2005.
- [9] L. AMICO, R. FAZIO, A. OSTERLOH, AND V. VEDRAL, *Entanglement in many-body systems*, Rev. Mod. Phys., 80 (2008), pp. 517–576.
- [10] A. ANFOSSI, P. GIORDA, A. MONTORSI, AND F. TRAVERSA, *Two-point versus multipartite entanglement in quantum phase transitions*, Phys. Rev. Lett., 95 (2005), p. 056402.
- [11] K. AUDENAERT AND J. EISERT, *Continuity bounds on the quantum relative entropy*, J. Math. Phys., 46 (2005), p. 102104.
- [12] A. AUERBACH, *Interacting electrons and quantum magnetism*, Springer Verlag, Berlin, 1998.
- [13] H. AZUMA, *Decoherence in grover's quantum algorithm: Perturbative approach*, Physical Review A, 65 (2002), p. 042311.
- [14] A. BARENCO, C. H. BENNETT, R. CLEVE, D. P. DIVINCENZO, N. MARGOLUS, P. SHOR, T. SLEATOR, J. A. SMOLIN, AND H. WEINFURTER, *Elementary gates for quantum computation*, Phys. Rev. A, 52 (1995), pp. 3457–3467.

- [15] D. BARRIOS, A. CARRASCAL, D. MANRIQUE, AND J. RÍOS, *Cooperative binary-real coded genetic algorithms for generating and adapting artificial neural networks*, Neural Computing and Applications, 12 (2003), pp. 49–60.
- [16] ———, *Optimisation with real-coded genetic algorithms based on mathematical morphology*, International Journal of Computer Mathematics, 80 (2003), pp. 275–293.
- [17] J. R. BAXTER, *Exactly solved models in statistical mechanics*, Academic Press, 1982.
- [18] D. BEASLEY, D. R. BULL, AND R. R. MARTIN, *An overview of genetic algorithms: Part 1, fundamentals.*, University Computing, 15 (1993), pp. 58–69.
- [19] P. BENIOFF, *Quantum mechanical hamiltonian models of turing machines*, Journal of Statistical Physics, 29 (1982), pp. 515–546.
- [20] P. BENIOFF, *Space searches with a quantum robot*, in Quantum Computation and Information, vol. 305 of AMS CONTEMPORARY MATHEMATICS, 2002.
- [21] C. BENNETT, A. GRUDKA, M. HORODECKI, P. HORODECKI, AND R. HORODECKI, *Postulates for measures of genuine multipartite correlations*, Phys. Rev. A, 83 (2011), p. 012312.
- [22] C. H. BENNETT, E. BERNSTEIN, G. BRASSARD, AND U. VAZIRANI, *Strengths and weaknesses of quantum computing*, SIAM J. Comput., 26 (1997), pp. 1510–1523.
- [23] P. S. BOURDON AND H. T. WILLIAMS, *Unital quantum operations on the bloch ball and bloch region*, Phys. Rev. A, 69 (2004), p. 022314.
- [24] G. BRASSARD, P. HØYER, M. MOSCA, AND A. TAPP, *Quantum amplitude amplification and estimation*, in Quantum Computation and Information, J. Samuel J. Lomonaco and H. E. Brandt, eds., vol. 305 of AMS CONTEMPORARY MATHEMATICS, 2002, ch. Quantum Amplitude Amplification and Estimation, pp. 53–74.
- [25] H.-P. BREUER AND F. PETRUCCIONE, *The Theory of Open Quantum Systems*, Oxford University Press, 2007.
- [26] A. BRODUTCH AND K. MODI, *Criteria for measures of quantum correlations*, Quantum Information and Computation, 12 (2012), p. 0721.
- [27] D. BRUSS AND C. MACCHIAVELLO, *Multipartite entanglement in quantum algorithms*, Phys. Rev. A, 83 (2011), p. 052313.
- [28] A. R. CALDERBANK AND P. W. SHOR, *Good quantum error-correcting codes exist*, Phys. Rev. A, 54 (1996), pp. 1098–1105.
- [29] Y. CAO AND A. M. WANG, *Revised geometric measure of entanglement*, Journal of Physics A: Mathematical and Theoretical, 40 (2007), p. 3507.
- [30] S. CARAIMAN AND V. MANTA, *Parallel simulation of quantum search*, Int. J. of Computers, Communications & Control, 5 (2010), pp. 634–641.
- [31] A. R. R. CARVALHO, F. MINTERT, AND A. BUCHLEITNER, *Decoherence and multipartite entanglement*, Phys. Rev. Lett., 93 (2004), p. 230501.
- [32] J. CHEN, D. KASZLIKOWSKI, L. KWEK, AND C. OH, *Searching a database under decoherence*, Physics Letters A, 306 (2003), pp. 296 – 305.
- [33] A. A. CLERK, M. H. DEVORET, S. M. GIRVIN, F. MARQUARDT, AND R. J. SCHOELKOPF, *Introduction to quantum noise, measurement, and amplification*, Rev. Mod. Phys., 82 (2010), pp. 1155–1208.

- [34] R. CLEVE, A. EKERT, L. HENDERSON, C. MACCHIAVELLO, AND M. MOSCA, *On quantum algorithms*, Complexity, 4 (1998), pp. 33–42.
- [35] J. CUI, J.-P. CAO, AND H. FAN, *Quantum information approach to the quantum phase transition in the kitaev honeycomb model*, Phys. Rev. A, 82 (2010), p. 022319.
- [36] J. CUI AND H. FAN, *Correlations in the grover search*, Journal of Physics A: Mathematical and Theoretical, 43 (2010), p. 045305.
- [37] ———, *Correlations in the grover search*, Journal of Physics A: Mathematical and Theoretical, 44 (2011), p. 279501.
- [38] H. K. CUMMINS, G. LLEWELLYN, , AND J. A. JONES, *Tackling systematic errors in quantum logic gates with composite rotations*, Phys. Rev. A, 67 (2003), p. 042308.
- [39] A. L. F. DE OLIVEIRA AND E. BUKSMAN, *Simulación de errores cuánticos en el ambiente scilab*, tech. report, Universidad ORT Uruguay, Julio 2007.
- [40] A. L. F. DE OLIVEIRA, E. BUKSMAN, AND J. GARCÍA-LÓPEZ, *Isotropic double index for quantum errors in one qubit*, Journal of Chemistry and Chemical Engineering, 5 (2011), pp. 1053–1058.
- [41] D. DEUTSCH, *Quantum theory, the church-turing principle and the universal quantum computer*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 400 (1985), pp. 97–117.
- [42] S. J. DEVITT, W. J. MUNRO, AND K. NEMOTO, *Quantum error correction for beginners*, Reports on Progress in Physics, 76 (2013), p. 076001.
- [43] R. DILLENSCHNEIDER, *Quantum discord and quantum phase transition in spin chains*, Phys. Rev. B, 78 (2008), p. 224413.
- [44] A. EKERT, P. HAYDEN, AND H. INAMORI, *Basic concepts in quantum computation*. arXiv:quant-ph/0011013v1, 2001.
- [45] R. P. FEYNMAN, *Simulating physics with computers*, International Journal of Theoretical Physics, 21 (1982), pp. 467–488.
- [46] ———, *Quantum mechanical computers*, Foundations of Physics, 16 (1986), pp. 507–531.
- [47] R. GALLEGO, L. E. WÜRFLINGER, A. ACÍN, AND M. NAVASCUÉS, *Quantum correlations require multipartite information principles*, Phys. Rev. Lett., 107 (2011), p. 210403.
- [48] J. GARCÍA-LÓPEZ AND F. G. MAZARÍO, *Modelos continuos de error en computación cuántica*, in Primer Congreso Internacional de Matemáticas en Ingeniería y Arquitectura, Madrid, España, 2007.
- [49] J. GARCÍA-LÓPEZ, F. GARCÍA-MAZARÓ, AND V. MARTÍN, *Modelos de error en computación cuántica. technical report n 20.*, tech. report, Universidad Politécnica de Madrid, 2003.
- [50] L. GATTI, *Implementación y simulación de circuitos cuánticos*. Proyecto de grado. Universidad ORT Uruguay, Marzo 2012.
- [51] P. GAWRON, J. KLAMKA, AND R. WINIARCZYK, *Noise effects in the quantum search algorithm from the viewpoint of computational complexity*, Applied Mathematics and Computer Science, 22 (2012), pp. 493–499.
- [52] J. GIBB, *Back propagation family album*, Tech. Report C/TR96-05, Macquarie University, 1996.

- [53] G. L. GIORGI, B. BELLOMO, F. GALVE, AND R. ZAMBRINI, *Genuine quantum and classical correlations in multipartite systems*, Phys. Rev. Lett., 107 (2011), p. 190501.
- [54] D. E. GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [55] D. GOTTESMAN, *Stabilizer codes and quantum error correction*, PhD thesis, California Institute of Technology, 1997.
- [56] ———, *Theory of fault-tolerant quantum computation*, Physical Review A, 57 (1998), pp. 127–137.
- [57] D. M. GREENBERGER, M. HORNE, AND A. ZEILINGER, *Bells theorem, quantum theory and conceptions of the universe*, vol. 37 of Fundamental Theories of Physics, Kluwer, 1989, ch. Going Beyond Bell’s Theorem, pp. 69–72.
- [58] L. K. GROVER, *A fast quantum mechanical algorithm for database search*, in Proceedings, 28th Annual ACM Symposium on the Theory of Computing p. 212, 1996.
- [59] ———, *Quantum mechanics helps in searching for a needle in haystack*, Phys. Rev. Lett., 79 (1997), pp. 325–328.
- [60] L. K. GROVER, *From Schrödinger’s equation to the quantum search algorithm*, Pramana, 56 (2001), pp. 333–348.
- [61] J.-L. GUO AND G.-L. LONG, *Quantum correlation dynamics of three-qubit system coupled to an xy chain*, Eur. Phys. J. D, 67 (2013), p. 53.
- [62] A.-R. HEDAR AND M. FUKUSHIMA, *Simplex coding genetic algorithm for the global optimization of nonlinear functions*, in Multi-Objective Programming and Goal Programming: Theory and applications., T. Tanino, T. Tanaka, and M. Inuiguchi, eds., Berlin-Heidelberg, 2003, Springer-Verlag, pp. 135–140.
- [63] L. HENDERSON AND V. VEDRAL, *Classical, quantum and total correlation*, J. Phys. A: Math. Gen., 34 (2001), p. 6899.
- [64] S. HILL AND W. K. WOOTTERS, *Entanglement of a pair of quantum bits*, Phys. Rev. Lett., 78 (1997), pp. 5022–5025.
- [65] D. HOCKER, R. KOSUT, AND H. RABITZ, *Peet: a matlab tool for estimating physical gate errors in quantum information processing systems*, Quantum Information Processing, 15 (2016), pp. 3489–3518.
- [66] M. HORODECKI AND P. HORODECKI, *Reduction criterion of separability and limits for a class of distillation protocols*, Phys. Rev. A, 59 (1999), pp. 4206–4216.
- [67] R. HORODECKI, P. HORODECKI, M. HORODECKI, AND K. HORODECKI, *Quantum entanglement*, Rev.Mod.Phys, 81 (2009), pp. 865–942.
- [68] R. HORST, P. M. PARDALOS, AND N. V. THOAI, *Introduction To Global Optimization*, Kluwer Academic Publishers, 2000.
- [69] G. JAEGER, *Quantum Information: An Overview*, Springer, 2007.
- [70] V. KENDON, *Decoherence in quantum walks – a review*, Mathematical Structures in Computer Science, 17 (2007), pp. 1169–1220.
- [71] P. L. KNIGHT, E. ROLDÁN, AND J. E. SIPE, *Quantum walk on the line as an interference phenomenon*, Phys. Rev. A, 68 (2003), p. 020301.

- [72] E. KNILL, R. LAFLAMME, AND L. VIOLA, *Theory of quantum error correction for general noise*, Phys. Rev. Lett., 84 (2000), pp. 2525–2528.
- [73] K. KRAUS, *Lecture notes in physics. States, effects, and operations: fundamental notions of quantum theory. Lectures in mathematical physics at the University of Texas at Austin.*, Springer-Verlag, 1983.
- [74] R. LAFLAMME, C. MIQUEL, J. P. PAZ, AND W. H. ZUREK, *Perfect quantum error correcting code*, Phys. Rev. Lett., 77 (1996), pp. 198–201.
- [75] A. LANGVILLE AND W. J. STEWART, *A kronecker product approximate preconditioner for sams*, Numerical Linear Algebra with Applications, 11 (2004), pp. 723–752.
- [76] ———, *Testing the nearest kronecker product preconditioner on markov chains and stochastic automata networks*, INFORMS Journal on Computing, 16 (2004), pp. 300–315.
- [77] C. V. LOAN AND N. PITSIANIS, *Approximation with kronecker products*, in Linear Algebra for Large Scale and Real Time Applications, Kluwer Publications, 1993, pp. 293–314.
- [78] O. MALOYER AND V. KENDON, *Decoherence versus entanglement in coined quantum walks*, New Journal of Physics, 9 (2007), p. 87.
- [79] E. MIZUTANI AND J.-S. R. JANG, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997, ch. Derivative-based Optimization, pp. 129–172.
- [80] K. MODI, A. BRODUTCH, H. CABLE, T. PATEREK, AND V. VEDRAL, *The classical-quantum boundary for correlations: discord and related measures*, Rev. Mod. Phys., 84 (2012), pp. 1655–1707.
- [81] G. E. MOORE, *Cramming more components onto integrated circuits*, PROCEEDINGS OF THE IEEE, (1998).
- [82] M. A. NIELSEN AND I. L. CHUANG, *Quantum computation and quantum information*, Cambridge University Press, 2000.
- [83] M. OKRASA AND Z. WALCZAK, *Quantum discord and multipartite correlations*, Europhysics Letters, 96 (2011), p. 60003.
- [84] H. OLLIVIER AND W. H. ZUREK, *Quantum discord: a measure of the quantumness of correlations*, Phys. Rev. Lett., 88 (2001), p. 017901.
- [85] T. J. OSBORNE AND M. A. NIELSEN, *Entanglement in a simple quantum phase transition*, Phys. Rev. A., 66 (2002), p. 032110.
- [86] A. OSTERLOH, L. AMICO, G. FALCI, AND R. FAZIO, *Scaling of entanglement close to a quantum phase transition*, Nature, 416 (2002), pp. 608–610.
- [87] A. PAL AND I. BOSE, *Transverse ising model: Markovian evolution of classical and quantum correlation under decoherence*, Eur. Phys. J. B, 85 (2012), p. 36.
- [88] R. PORTUGAL, *Quantum Walks and Search Algorithms*, Springer, 2013.
- [89] R. PORTUGAL, C. C. LAVOR, L. M. CARVALHO, AND N. MACULAN, *Uma introdução à computação quântica*, SBMAC, 2012.
- [90] J. PRESKILL, *Introduction to quantum computation and information*, World Scientific, 1998, ch. Fault-Tolerant Quantum Computation. arXiv:quant-ph/9712048v1.
- [91] H. D. RAEDT AND K. MICHIELSEN, *Computational methods for simulating quantum computers*, in Handbook of Theoretical and Computational Nanotechnology, Rieth and W. Schommers, eds., vol. 3: Quantum and molecular computing, quantum simulations, American Scientific Publisher, Los Angeles, USA, 2006, ch. 1, p. 248.

- [92] R. RAUSSENDORF, D. E. BROWNE, AND H. J. BRIEGEL, *Measurement-based quantum computation on cluster states*, Phys. Rev. A, 68 (2003), p. 022312.
- [93] O. REGEV AND L. SCHIFF, *Impossibility of a quantum speed-up with a faulty oracle*, in Automata, Languages and Programming, L. Aceto, I. Damgård, L. Goldberg, M. Halldórsson, A. Ingólfssdóttir, and I. Walukiewicz, eds., vol. 5125 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 773–781.
- [94] A. RIVAS AND S. F. HUELGA, *Open quantum systems. An introduction*, Springer, 2012.
- [95] A. ROMANELLI, R. SIRI, G. ABAL, A. AUYUANET, AND R. DONANGELO, *Decoherence in the quantum walk on the line*, Physica A: Statistical Mechanics and its Applications, 347 (2005), pp. 137 – 152.
- [96] C. C. RULLI AND M. S. SARANDY, *Entanglement and local extremes at an infinite-order quantum phase transition*, Phys. Rev. A, 81 (2010), p. 032334.
- [97] ———, *Global quantum discord in multipartite systems*, Phys. Rev. A, 84 (2011), p. 042109.
- [98] S. SACHDEV, *Quantum phase transitions*, John Wiley & Sons Ltd, 2007.
- [99] P. J. SALAS, *Noise effect on grover algorithm*, The European Physical Journal D, 46 (2008), pp. 365–373.
- [100] P. J. SALAS AND A. L. SANZ, *Numerical simulation of information recovery in quantum computers*, Phys. Rev. A, 66 (2002), p. 022302.
- [101] M. S. SARANDY, *Classical correlation and quantum discord in critical systems*, Phys. Rev. A, 80 (2009), p. 022108.
- [102] M. SCHLOSSHAUER, *Decoherence, the measurement problem, and interpretations of quantum mechanics*, Rev. Mod. Phys., 76 (2005), pp. 1267–1305.
- [103] E. SCHRÖDINGER, *Probability relations between separated systems*, Mathematical Biosciences-Proceedings of the Cambridge Philosophical Society, 32 (1936), pp. 446–452.
- [104] D. SHAPIRA, S. MOZES, AND O. BIHAM, *Effect of unitary noise on grover’s quantum search algorithm*, Physical Review A, 67 (2003), p. 042301.
- [105] P. W. SHOR, *Algorithms for quantum computation: Discrete logarithms and factoring*, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science, SFCS ’94, Washington, DC, USA, 1994, IEEE Computer Society, pp. 124–134.
- [106] ———, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A, 52 (1995), p. 2493.
- [107] ———, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, 26 (1997), pp. 1484–1509.
- [108] G. SONG AND A. KLAPPENECKER, *The simplified toffoli gate implementation by margolus is optimal*, (2003). eprint quant-ph/0312225v1.
- [109] R. SRIKANTH, S. BANERJEE, AND C. M. CHANDRASHEKAR, *Quantumness in a decoherent quantum walk using measurement-induced disturbance*, Phys. Rev. A, 81 (2010), p. 062123.
- [110] K. M. SVORE, A. V. AHO, A. W. CROSS, I. CHUANG, AND I. L. MARKOV, *A layered software architecture for quantum computing design tools*, IEEE Computer, 39 (2006), pp. 74–83.
- [111] K. TEMME, *Runtime of unstructured search with a faulty hamiltonian oracle*, Physical Review A, 90 (2014), p. 022310.

- [112] A. TULSI, *Faster quantum-walk algorithm for the two-dimensional spatial search*, Phys. Rev. A, 78 (2008), p. 012310.
- [113] T. TULSI, L. GROVER, AND A. PATEL, *A new algorithm for fixed point quantum search*, Quantum Information and Computation, 26 (2006), pp. 483–494.
- [114] M. VAN DEN NEST, *Universal quantum computation with little entanglement*, Phys. Rev. Lett., 110 (2013), p. 060504.
- [115] V. VEDRAL, *The role of relative entropy in quantum information theory*, Rev. Mod. Phys., 74 (2002), p. 197.
- [116] S. E. VENEGAS-ANDRACA, *Quantum walks: a comprehensive review*, Quantum Information Processing, 11 (2012), pp. 1015–1106.
- [117] F. VERSTRAETE, M. POPP, AND J. I. CIRAC, *Entanglement versus correlations in spin systems*, Phys. Rev. Lett., 92 (2004), p. 027901.
- [118] G. F. VIAMONTES, *Efficient quantum circuit simulation*, PhD thesis, The University of Michigan, 2007.
- [119] G. F. VIAMONTES, I. L. MARKOV, AND J. P. HAYES, *Improving gate-level simulation of quantum circuits*, Quantum Information Processing, 2 (2003), pp. 347–380.
- [120] ———, *Graph-based simulation of quantum computation in the density matrix representation*, Quantum Information and Computation, 5 (2005), pp. 113–130.
- [121] G. F. VIAMONTES, I. L. MARKOV, AND J. P. HAYES, *Is quantum search practical?*, Computing in Science Engineering, 7 (2005), pp. 62–70. arXiv:quant-ph/0405001.
- [122] G. F. VIAMONTES, I. L. MARKOV, AND J. P. HAYES, *Quantum Circuit Simulation*, Springer, 2009.
- [123] G. F. VIAMONTES, M. RAJAGOPALAN, I. L. MARKOV, AND J. P. HAYES, *Gate-level simulation of quantum circuits*, in Proceedings of the ASP-DAC 2003. Asia and South Pacific., 2003. arXiv:quant-ph/0208003.
- [124] G. VIDAL, *Efficient classical simulation of slightly entangled quantum computations*, Phys. Rev. Lett., 91 (2003), p. 147902.
- [125] P. VRANA, D. REEB, D. REITZNER, AND M. M. WOLF, *Fault-ignorant quantum search*, New Journal of Physics, 16 (2014), p. 073033.
- [126] T.-C. WEI AND P. M. GOLDBART, *Geometric measure of entanglement and applications to bipartite and multipartite quantum states*, Phys. Rev. A, 68 (2003), p. 042307.
- [127] T. WERLANG, G. A. P. RIBEIRO, AND G. RIGOLIN, *Interplay between quantum phase transition and behavior of quantum correlations at finite temperatures*, International Journal of Modern Physics B, 27 (2013), p. 1345032.
- [128] T. WERLANG, C. TRIPPE, G. A. P. RIBEIRO, AND G. RIGOLIN, *Quantum correlations in spin chains at finite temperatures and quantum phase transitions*, Phys. Rev. Lett., 105 (2010), p. 095702.
- [129] D. WHITLEY, *A genetic algorithm tutorial*, Statistics and Computing, 4 (1994), pp. 65–85.
- [130] J. WILMS, J. VIDAL, F. VERSTRAETE, AND S. DUSUEL, *Finite-temperature mutual information in a simple phase transition*, J. Stat. Mech., (2012), p. P01023.
- [131] W. K. WOOTTERS, *Entanglement of formation of an arbitrary state of two qubits*, Phys. Rev. Lett., 80 (1998), pp. 2245–2248.

- [132] W. K. WOOTTERS, *Entanglement of formation and concurrence*, Quantum Information and Computation, Vol. 1, No. 1 (2001), pp. 27–44.
- [133] Y. YAO, H.-W. LI, C.-M. ZHANG, Z.-Q. YIN, W. CHEN, G.-C. GUO, AND Z.-F. HAN, *Performance of various correlation measures in quantum phase transitions using the quantum renormalization-group method*, Phys. Rev. A, 86 (2012), p. 042102.
- [134] J. YEN, J. C. LIAO, B. LEE, AND D. RANDOLPH, *A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method*, IEEE Transactions on Systems, and Cybernetics B, 28 (1998), pp. 197–201.
- [135] C. ZALKA, *Grover’s quantum searching algorithm is optimal*, Physical Review A, 60 (1999), pp. 2746–2751.
- [136] J.-S. ZHANG AND A.-X. CHEN, *Review of quantum discord in bipartite and multipartite systems*, Quant. Phys. Lett., Vol. 1 No. 2 (2012), pp. 69–77.
- [137] D. L. ZHOU, B. ZENG, Z. XU, AND L. YOU, *Multiparty correlation measure based on the cumulant*, Phys. Rev. A, 74 (2006), p. 052110.
- [138] W. H. ZUREK, *Decoherence, einselection, and the quantum origins of the classical*, Rev. Mod. Phys., 75 (2003), pp. 715–775.