



**UNIVERSIDAD
AUTÓNOMA
METROPOLITANA
Unidad Iztapalapa**

División de Ciencias Básicas e Ingeniería

Licenciatura en Física

Proyecto Terminal I: Investigación teórica

**ESTADOS DE FASE CÚBICA EN EL
CÓMPUTO CUÁNTICO DE VARIABLE
CONTINUA**

REALIZADO POR: **Aland Rodríguez Martín del Campo**

MATRÍCULA: **2173010763.**

PARA OBTENER EL GRADO DE: **Licenciatura en Física**

ASESOR: **DR. MIGUEL ANGEL BASTARRACHEA MAGNANI**

CO-ASESOR: **DR. ERNESTO BENÍTEZ RODRÍGUEZ**

COORDINADOR: **DR. MARCO ANTONIO MACEDA SANTAMARÍA**

6 DE MAYO DE 2025

IZTAPALAPA, CIUDAD DE MÉXICO

Índice

1. Introducción	2
1.1. Estados de Fock	3
1.2. Estados y operaciones Gaussianas	4
1.3. Estados Gaussianos	4
1.3.1. Estados Coherentes	5
1.3.2. Estados comprimidos	5
1.4. Función de Wigner	7
1.5. Cómputo Cuántico	9
1.5.1. Cómputo universal	9
2. Teoría de recursos y sus elementos	11
2.1. Teoría de recursos cuánticos	11
2.2. Operaciones libres	11
2.3. Estados libres	12
2.4. Recursos	12
2.4.1. Medidas monótonas	13
2.4.2. Testigos	15
3. Cómputo cuántico de variable continua	
3.1. Teorema de Gottesman-Knill	16
3.2. Estados de fase cúbica $ \gamma, r\rangle$	18
4. Negatividad de la función de Wigner	
4.1. NLW como medida del recurso	23
5. Resultados	24
5.1. Limitaciones numéricas de la expansión de $\hat{\Gamma}(\gamma)$	26
5.2. NLW para algunos estados	31
6. Conclusiones	38
A. Códigos	39
A.1. Comparación entre <i>QuTiP</i> y <i>expansión en serie</i>	39
A.2. NLW e integrales asociadas para (γ, r) específicas.	41
A.3. Funciones $W(q, p)$ bajo el efecto de variar γ ó r	42
A.4. Evolución de la NLW en función de N	44
A.5. Mapa de calor de $\text{NLW}(\gamma, r)$	46
A.6. Evolución de $\text{NLW}(\gamma)$ y $\text{NLW}(r)$	49

1. Introducción

Uno de los pilares más importantes de la física moderna es la mecánica cuántica, la cual, desde sus inicios ha transformado nuestra comprensión de la naturaleza hasta escalas fundamentales. Sus principios, como la superposición y el entrelazamiento de estados, han trascendido el interés puramente teórico hacia la práctica impulsando el desarrollo de nuevas tecnologías en diversos campos de nuestra vida cotidiana, por ejemplo los teléfonos inteligentes que usamos día a día o incluso aquellas que son aplicables en la medicina. Un campo destacable que busca aprovechar estos fenómenos cuánticos es el de la información cuántica, en donde se busca procesar y transmitir información de formas que son inaccesibles para la computación clásica. Mientras esta última codifica la información en unidades básicas (*bits*) que representan un estado bien definido (0 ó 1), la computación cuántica, relevante dentro del marco de la información cuántica, busca codificar la información en unidades básicas que pueden existir en una superposición de ambos estados simultáneamente.

Dentro de la computación cuántica existen dos paradigmas con gran relevancia: El cómputo cuántico de variable discreta (CCVD), basado en qubits, es quizás el más conocido, pero también existe el cómputo cuántico de variable continua (CCVC), el cual es una variante de la computación cuántica que utiliza variables físicas continuas para la codificación de la información, por ejemplo: la posición y el momento de una partícula, o los modos de un campo cuántico.

El objetivo de este trabajo es caracterizar un conjunto específico de estados cuánticos denominados estados de fase cúbica, que pertenecen a la clase de estados no Gaussianos. Estos estados resultan de aplicar un operador no lineal, conocido como el operador de fase cúbica, sobre el estado de vacío comprimido. En términos generales, los estados no Gaussianos son fundamentales para extender las capacidades del procesamiento de información cuántica más allá de lo que permiten las operaciones Gaussianas, como desplazamientos y compresiones. En particular, el operador de fase cúbica introduce una no linealidad cúbica al depender del operador de posición elevado al cubo. Esta característica permite explorar nuevas posibilidades en la manipulación de estados cuánticos y habilita, en teoría, una ventaja computacional frente a los métodos clásicos. El interés en los estados de fase cúbica radica en su posible implementación dentro del marco del CCVC. En este contexto, los estados no Gaussianos como los de fase cúbica pueden complementar operaciones Gaussianas para lograr universalidad en el cómputo cuántico.

Para formalizar el valor de estos estados como recursos computacionales, se adopta una teoría de recursos basada en la función de Wigner, una función de quasi-probabilidad que describe el estado cuántico de un sistema en el espacio fase. Una propiedad distintiva de los estados no Gaussianos es que su función de Wigner puede presentar regiones con valores negativos, lo cual es una manifestación directa de su no clasicidad. Esta negatividad de la función de Wigner se interpreta como un recurso cuántico útil en el CCVC.

Concretamente, los estados de fase cúbica exhiben dicha negatividad, lo que los convierte en candidatos prometedores para el cómputo cuántico con ventajas reales. Para cuantificar este recurso, se utiliza la negatividad logarítmica de la función de Wigner. El trabajo se

enfoca en determinar la combinación óptima de parámetros que maximice esta medida de recurso, mediante simulaciones numéricas. Para ello, se emplean dos métodos de aproximación del operador de fase cúbica: **(a)** la expansión en serie de Taylor del operador de fase cúbica, y **(b)** el algoritmo de Clenshaw, implementado en la biblioteca QuTiP de Python, la cual permite una simulación eficiente de ciertos sistemas cuánticos.

1.1. Estados de Fock

Debido a que en el CCVC puede trabajarse con modos de campos electromagnéticos, es prudente mencionar que una forma común de describirlos es a partir de los estados de Fock. Un estado de Fock $|n\rangle$ (también conocido como *estado de número*), es un estado físico con una cantidad bien definida (n) de fotones (o, en general, bosones), en el que las amplitudes de los campos electromagnéticos no están bien definidas debido la no commutación entre los operadores de creación (\hat{a}^\dagger) y aniquilación (\hat{a}) con el operador de número ($\hat{N} = \hat{a}^\dagger \hat{a}$) [1, 2]. Esto es así ya que, si bien existe un número n de fotones en el sistema, la aplicación de alguno de estos dos operadores cambiará esta cantidad sucesivamente con cada aplicación.

Los estados de Fock forman una base completa del espacio de Hilbert asociado a un oscilador armónico cuántico. En el CCVC, el espacio de Fock proporciona una base ortogonal fundamental ($\langle m|n\rangle = \delta_{mn}$), ya que en un sentido físico, la descripción cuántica del campo electromagnético puede representarse mediante la expansión del operador de densidad en la base de estados de Fock, la cual incluye términos diagonales y no diagonales. Debido a la importancia que tienen estos estados en el CCVC, es prudente hacer mención de su definición y la acción que tienen los operadores de creación y aniquilación sobre estos estados [1, 2]:

$$\hat{a}^\dagger = \frac{1}{\sqrt{2\hbar}}(\hat{q} - i\hat{p}), \quad \hat{a} = \frac{1}{\sqrt{2\hbar}}(\hat{q} + i\hat{p}). \quad (1.1)$$

En donde \hbar es la constante de Planck reducida, la cual, aunque se mostrará explícitamente en las ecuaciones siguientes, para la sección de resultados y los códigos que se utilicen, se hará $\hbar = 1$. Ahora, teniendo en cuenta una función de onda ψ en el espacio de posición, los operadores de posición y momento (\hat{q} y \hat{p} , respectivamente) se definen como:

$$\hat{q}\psi(q) = q\psi(q), \quad \hat{p}\psi(q) = -i\hbar \frac{d}{dq}\psi(q). \quad (1.2)$$

Por otra parte, la acción de los operadores \hat{a} y \hat{a}^\dagger sobre $|n\rangle$ es tal que:

$$\hat{a}^\dagger|0\rangle = \sqrt{n+1}|n+1\rangle, \quad \hat{a}|n\rangle = \sqrt{n}|n-1\rangle. \quad (1.3)$$

Por supuesto, los operadores de posición (\hat{q}) y momento (\hat{p}) pueden escribirse en términos de los de creación y aniquilación, a saber:

$$\hat{q} = \sqrt{\frac{\hbar}{2}} (\hat{a}^\dagger + \hat{a}), \quad \hat{p} = i\sqrt{\frac{\hbar}{2}} (\hat{a}^\dagger - \hat{a}). \quad (1.4)$$

El estado de vacío, $|0\rangle$, resulta de suma importancia, pues los estados de Fock pueden generarse a partir de este estado base [1]:

$$|n\rangle = \frac{(\hat{a}^\dagger)^n}{\sqrt{n!}} |0\rangle. \quad (1.5)$$

Esto es relevante en el contexto de CCVC, porque muchos estados (como los estados Gaussianos) son superposiciones infinitas de estados de Fock.

1.2. Estados y operaciones Gaussianas

Los estados Gaussianos representan un tipo particular de estado en los sistemas cuánticos, a menudo considerados como los más “simples” o “clásicos” dentro de su descripción. Las operaciones Gaussianas son transformaciones cuánticas que provienen de interacciones físicas descritas por Hamiltonianos que son, como máximo, cuadráticos en los operadores canónicos (como posición \hat{q} y momento \hat{p} , o los operadores de aniquilación \hat{a} y creación \hat{a}^\dagger) [3]. Una característica importante de estas operaciones es que, cuando se aplican a estados Gaussianos, el resultado sigue siendo un estado Gaussiano [4].

La relevancia clave de esta combinación de estados y operaciones simples es que se ha demostrado que si un proceso físico cuántico utiliza únicamente estados Gaussianos y operaciones Gaussianas que no añaden una complejidad cuántica más profunda, este proceso puede ser reproducido y simulado eficientemente en una computadora clásica convencional [3, 4, 5, 6].

Por lo tanto, para poder llevar a cabo tareas que van más allá de lo que una computadora clásica puede simular de manera eficiente, se vuelve necesario el uso de estados o transformaciones que no sean de este tipo Gaussiano, es decir, que manifiesten propiedades cuánticas que rompen con esta “simplicidad” o “clasicidad” particular.

1.3. Estados Gaussianos

Los estados Gaussianos son un tipo de estados cuánticos con una gran relevancia experimental en el marco de la óptica cuántica ya que las operaciones sobre ellos (operaciones Gaussianas) son matemáticamente manejables, es decir, pueden manipularse con cierta facilidad y experimentalmente accesibles [7]. La característica que define a un estado Gaussiano es que está completamente especificado por sus dos primeros momentos estadísticos: medias (primer momento) y la covarianza (segundo momento) [5, 6]. Se les llama “Gaussianos” porque sus funciones de cuasiprobabilidad en el espacio fase, como

la función de Wigner ($W(q, p)$), la cual se verá posteriormente, tiene una forma Gaussiana. En general, la función de onda de los estados Gaussianos puros (que son los que se tratarán en este trabajo) es [8]:

$$\psi(x) = e^{-\frac{1}{2}(ax^2 + 2bx + c)}, \quad (1.6)$$

en donde $a, b, c \in \mathbb{C}$, además $\text{Re}(a) > 0$ y c se debe elegir de forma que se asegure la normalización [8].

1.3.1. Estados Coherentes

Estos estados pueden describirse como un estado en el que el paquete de ondas del estado fundamental se ha desplazado en el espacio de fases de posiciones y momentos (q, p) . Así, pueden generarse haciendo uso del *operador de desplazamiento* $D(\alpha) = e^{\alpha\hat{a}^\dagger - \alpha^*\hat{a}}$, aplicándolo sobre el estado de vacío $|0\rangle$ dando como resultado [1, 2]:

$$|\alpha\rangle = D(\alpha)|0\rangle = e^{-\frac{1}{2}|\alpha|^2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle, \quad (1.7)$$

en donde $\alpha \in \mathbb{C}$. Una característica muy importante de los estados coherentes es que minimizan la relación de incertidumbre de Heisenberg [1]. Además, poseen la propiedad de mantener una fase bien definida durante un periodo de tiempo determinado, debido a que están constituidos por una superposición cuántica de estados con número indefinido de fotones. Esto contrasta con los estados de Fock (o estados de número), los cuales presentan una fase completamente aleatoria [1].

1.3.2. Estados comprimidos

Los estados comprimidos representan una extensión de los estados coherentes, y son fundamentales en la descripción de sistemas cuánticos donde las fluctuaciones en una cuadratura, como la posición o el momento (Ec. 1.4), necesitan minimizarse por debajo de las que se asocian con los estados coherentes. A diferencia de estos últimos que, como se ha visto, mantienen una incertidumbre que es igual en ambas cuadraturas y minimizan la relación de incertidumbre de Heisenberg, los estados comprimidos las redistribuyen entre las cuadraturas de manera que alguna de ellas puede reducirse por debajo del límite de Heisenberg a costa de aumentar las fluctuaciones en la cuadratura complementaria para respetar la relación de incertidumbre [1, 2].

Este tipo de estados, de manera similar a cómo son creados los estados coherentes, también se crean mediante la aplicación de un operador dependiente de algún operador canónico, particularmente \hat{a} y \hat{a}^\dagger , solo que en este caso se aplica a un estado coherente ($|\alpha\rangle$). Este operador es conocido como *operador de compresión* y es definido como [1, 2]:

$$\hat{S}(\xi) = e^{\frac{1}{2}(\xi^*\hat{a}^2 - \xi\hat{a}^\dagger)^2}. \quad (1.8)$$

En donde $\xi = re^{i\theta}$ determina el grado de compresión que sufrirá el estado al que se aplique a través de la magnitud $r \in [0, \infty)$ y en dirección de $\theta \in [0, 2\pi]$, es decir, no es forzoso que la compresión se dé a lo largo de q o de p , sino que puede darse en cualquier dirección del espacio fase [1, 2].

A diferencia de $\hat{D}(\alpha)$, que actúa creando o destruyendo fotones de manera individual, $\hat{S}(\xi)$ lo hace en pares de fotones correlacionados [1], lo cual puede observarse en el argumento de la función exponencial, que, a diferencia del operador de desplazamiento, contiene \hat{a}^2 y $\hat{a}^{\dagger 2}$.

Para poder analizar cómo cambian las cuadraturas y propiedades de los fotones en este tipo de estados, es necesario observar qué sucede cuando $\hat{S}(\xi)$ (y $\hat{S}^\dagger(\xi) = \hat{S}^{-1}(\xi) = \hat{S}(-\xi)$) se aplica sobre los operadores de creación y aniquilación:

$$\begin{aligned}\hat{S}^\dagger(\xi)\hat{a}\hat{S}(\xi) &= \hat{a}\cosh r - \hat{a}^\dagger e^{i\theta} \sinh r, \\ \hat{S}^\dagger(\xi)\hat{a}^\dagger\hat{S}(\xi) &= \hat{a}^\dagger\cosh r - \hat{a}e^{-i\theta} \sinh r.\end{aligned}\tag{1.9}$$

El operador de compresión puede aplicarse a un estado general $|\psi\rangle$ el cual se transforma en un nuevo estado $|\psi_s\rangle = \hat{S}(\xi)|\psi\rangle$ que mantiene propiedades de $|\psi\rangle$ pero con diferencias en las cuadraturas. Sin embargo, también puede aplicarse directamente sobre el estado fundamental introduciendo pares de fotones correlacionados distribuidos de manera que las fluctuaciones difieran en las cuadraturas [1].

De esta manera, se puede obtener un estado comprimido más general al aplicar $\hat{D}(\alpha)$ sobre las ecuaciones 1.9 y el estado de vacío, resultando en [1]:

$$|\alpha, \xi\rangle = \hat{D}(\alpha)\hat{S}(\xi)|0\rangle.\tag{1.10}$$

Así, si $\xi = 0$, se obtiene un estado coherente, o por el contrario, hacer $\alpha = 0$ y se obtiene el *vacío comprimido*.

El operador de compresión también puede expandirse en términos de estados de Fock [1], sin embargo, esta expansión únicamente involucra términos pares debido a que $\hat{S}(\xi)$ los introduce (o aniquila) en pares. Así, el estado de vacío comprimido se expresa como [1]:

$$|\xi\rangle = \frac{1}{\sqrt{\cosh r}} \sum_{m=0}^{\infty} (-1)^m \frac{\sqrt{(2m)!}}{2^m m!} e^{im\theta} (\tanh r)^m |2m\rangle = A_m(r, \theta) |2m\rangle.\tag{1.11}$$

Una forma simple de analizar este tipo de estados surge de notar que el operador $\hat{S}(\xi)$ reescalas la función de onda de acuerdo con la siguiente ecuación [9]:

$$\psi_\xi(q) = e^{\xi/2}\psi_0(e^\xi q). \quad (1.12)$$

Estos estados Gaussianos pueden visualizarse en el espacio fase de posiciones y momentos mediante una distribución de probabilidad, que particularmente en este estudio se utilizará la función de Wigner.

1.4. Función de Wigner

Se trata de una distribución de *cuasi-probabilidad* en el espacio fase, esto quiere decir que, a diferencia de una distribución de probabilidad común, esta puede tomar valores negativos para estados de naturaleza no clásica y resulta de mucha utilidad para representar estados cuánticos de una manera similar a como se hace en la mecánica clásica [1]. Cabe mencionar que, aunque la función de Wigner puede definirse mediante convenciones ligeramente diferentes, todas ellas son equivalentes. Una de ellas es:

$$W(q, p) = \frac{1}{2\pi\hbar} \int \psi^* \left(q + \frac{1}{2}y \right) \psi \left(q - \frac{1}{2}y \right) e^{-ipy/\hbar} dy. \quad (1.13)$$

En esta ecuación, q y p son las variables de posición y momento en el espacio fase y corresponden a los eigenvalores de los operadores \hat{q} y \hat{p} , respectivamente. Por supuesto, $\psi(x)$ corresponde a la función de onda del estado en cuestión.

Mediante la obtención de la función de Wigner asociada a algún estado, puede visualizarse su densidad de probabilidad en el espacio fase. Particularmente se muestran las asociadas a los estados Gaussianos mencionados en la sección previa. En la Fig. 1 se muestra la función de Wigner asociada a un estado coherente desplazado hacia la posición $(1, 1)$ en el espacio fase, mientras que en la Fig. 2 se visualiza la función de Wigner correspondiente a un estado comprimido centrado en el origen.

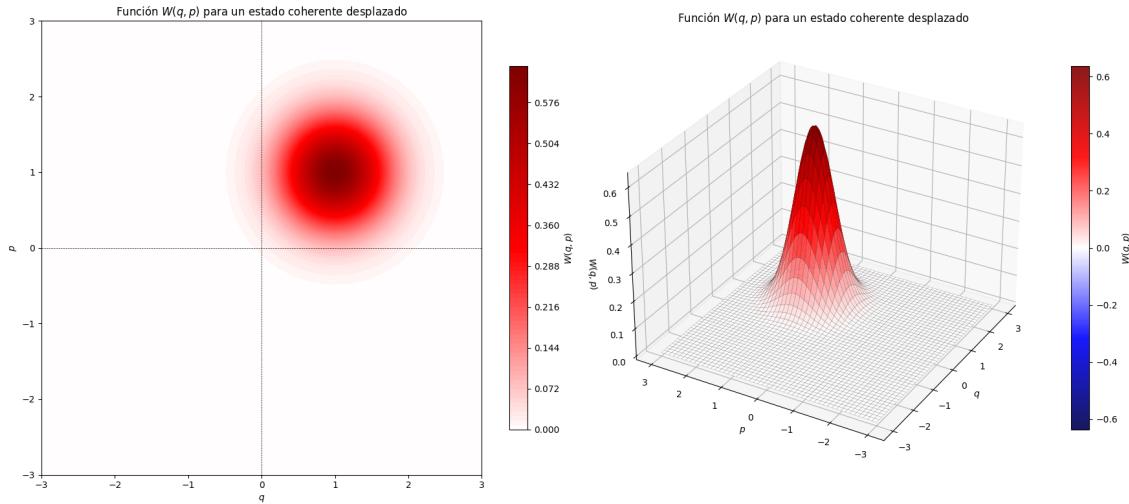


Figura 1: Se muestra la función de Wigner asociada a un estado coherente desplazado en las cuadraturas de posición y momento hacia ($q = 1, p = 1$). Puede notarse en la gráfica en 2 dimensiones que la incertidumbre para la posición es igual que la del momento, mientras que en 3 dimensiones se observa que esta función tiene una forma Gaussiana.

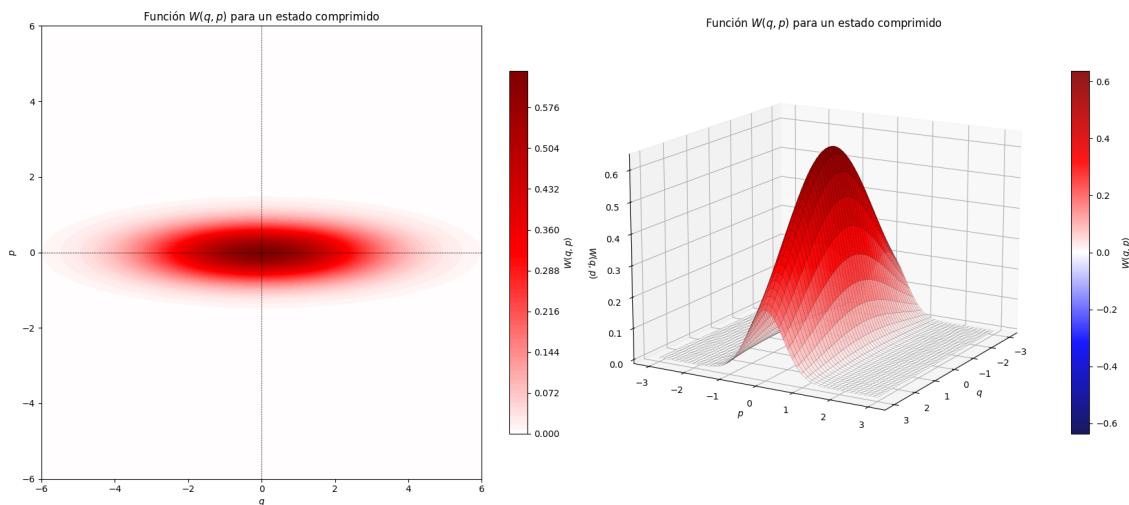


Figura 2: Se muestra la función de Wigner asociada a un estado comprimido centrado en el origen. Puede notarse en la gráfica en 2 dimensiones que la incertidumbre para la posición es mayor que la del momento, mientras que en 3 dimensiones se observa que esta función tiene una forma Gaussiana “alargada” hacia un lado, efecto debido a una mayor incertidumbre en q que en p .

1.5. Cómputo Cuántico

La computación clásica y la computación cuántica representan enfoques distintos para el procesamiento de información. Mientras que la computación clásica opera con bits binarios (0 ó 1), lo que constituye la base de la computación digital, también pueden utilizarse variables continuas mediante sistemas analógicos [10].

La computación cuántica, por su parte, se distingue por aprovechar fenómenos descritos por la mecánica cuántica. Uno de los fenómenos fundamentales para este tipo de cómputo es la superposición, esto es: la capacidad de que un sistema cuántico puede existir en una combinación de estados simultáneamente, a diferencia del bit clásico que solo puede estar en un estado discreto (0 ó 1).

Esta información en estado de superposición puede codificarse de manera discreta, utilizando sistemas con un número finito de estados, como los *qubits*, los cuales pueden estar en una superposición lineal de los estados base $|0\rangle$ y $|1\rangle$ [6]. Este tipo de computación, conocido como cómputo cuántico de variable discreta (CCVD), es el más conocido y estudiado [6], sin embargo, se han encontrado algunas limitantes en su implementación y escalamiento para una gran cantidad de qubits, por ejemplo, lograr mantener la coherencia entre ellos por un tiempo prolongado, es decir, mantener los estados de superposición y el enredamiento entre ellos [1, 11].

Por otra parte, la información en estado de superposición, también puede codificarse de manera continua, de manera semejante a la computación clásica. La computación cuántica de variable continua (CCVC) opera sobre sistemas cuánticos descritos por variables continuas, como la posición y el momento. En este marco, los estados también pueden existir en superposición de estados base continuos [6]. Los elementos básicos utilizados en CCVC, que a menudo son modos de un campo cuántico (por ejemplo, modos del campo electromagnético [6, 7, 12]), suelen ser referidos como *qumodes*, en analogía con los qubits utilizados en la computación cuántica de vacíos discretos (CCVD) [13].

La capacidad de codificar la información en estados dotados de superposición hace que, en el cómputo cuántico, se puedan representar múltiples valores al mismo tiempo, lo que permite procesar información de manera paralela, es decir, realizar una operación o evaluar una función con varias entradas simultáneamente en un solo paso, con lo que potencialmente, se podrían resolver ciertos problemas mucho más rápido que una computadora clásica [6].

1.5.1. Cómputo universal

El cómputo universal es la capacidad de un sistema computacional para simular cualquier algoritmo mediante un conjunto finito de operaciones básicas [6]. Para el cómputo universal clásico las operaciones básicas que permiten simular cualquier algoritmo son las operaciones lógicas AND, OR y NOT. Su versión cuántica amplía este concepto al dominio de las transformaciones unitarias propias de la mecánica cuántica. Es decir, implica la capacidad de aproximar cualquier transformación unitaria mediante un conjunto finito de operaciones cuánticas elementales [6].

En el mundo de las variables continuas, hablar de computación cuántica universal se vuelve un poco más delicado que en el caso discreto, esto se debe a que una transformación unitaria completamente arbitraria sobre una sola variable continua implicaría usar una cantidad infinita de parámetros para poder transformar por completo todo el espacio, lo cual resulta poco práctico. Es por ello que la idea de universalidad se restringe a ciertas subclases de transformaciones más manejables. En particular, aquellas que provienen de Hamiltonianos polinómicos: es decir, Hamiltonianos que se pueden escribir como funciones polinómicas de los operadores canónicos continuos (por ejemplo, posición (\hat{q}) y momento (\hat{p})) [6].

A pesar de que las operaciones Gaussianas permiten manipular los estados cuánticos de diversas formas, estas por sí solas no son suficientes para la computación cuántica universal [3, 14, 15]. Esto se debe a que los procesos cuánticos que involucran este tipo de operaciones y estados Gaussianos pueden ser simulados eficientemente de manera clásica [3], lo que significa que no proporcionan una ventaja computacional cuántica significativa sobre las computadoras clásicas.

Por otra parte, se ha mostrado que mediante la combinación de operaciones lineales (como desplazamientos o transformaciones gaussianas) con al menos una operación por lo menos cúbica (que contenga, por ejemplo, \hat{q}^3 o \hat{p}^3) podría lograrse la computación cuántica universal en el marco de variables continuas [14, 15]. Es aquí donde las teorías de recursos, particularmente las de no-Gaussianidad y negatividad de la función de Wigner, adquieren relevancia, pues proporcionan un marco formal para identificar y cuantificar las propiedades cuánticas que son necesarias para lograr la computación cuántica universal y poder alcanzar una ventaja cuántica computacional sobre los métodos clásicos en algunas tareas específicas [15].

2. Teoría de recursos y sus elementos

Para entender de que trata una teoría de recursos es útil comprender, en primer lugar, qué es un recurso. Un recurso es algo que adquiere valor debido a que existen restricciones que impiden obtenerlo libremente dentro del sistema que se esté considerando, así por ejemplo, en la economía, algunos recursos pueden ser el agua, el oro o el petróleo ya que no pueden producirse de manera arbitraria, sino que el sistema, para este ejemplo, el planeta, tiene una cantidad limitada de estos. En un contexto computacional, el poder de cómputo (CPU, RAM, GPU, etc.) es un recurso limitado ya que algún software que se ejecute dentro de una PC no puede utilizar más memoria o procesamiento que el que es permitido por el hardware actualmente implementado en la misma, de manera que, si se busca aumentar el poder de cómputo, será forzoso añadir hardware adicional en la placa madre de la computadora para lograrlo, es decir, agregarlo al sistema [16].

Dentro de una teoría de recursos, existen operaciones y estados que son permisibles, de manera que a estos se les denomina como *libres* [17]. Por otro lado, los estados que no son libres, se denominan como *estados de recurso* y sus operaciones como prohibidas [17].

2.1. Teoría de recursos cuánticos

Por supuesto, las teorías de recursos no son exclusivas del mundo clásico, sino que también existen en el contexto de la mecánica cuántica, así, análogamente a los ejemplos económico y computacional mencionados, en el mundo cuántico existen propiedades que no pueden ser creadas libremente bajo ciertas restricciones físicas o matemáticas, como pueden ser el enredamiento, la coherencia o la negatividad de la función de Wigner [18] (de la cual se hablará con más detalle posteriormente).

De manera análoga al caso clásico, surge la pregunta de qué es posible hacer con los recursos que brinda la física cuántica. Es así que las teorías de recursos cuánticos (TRC) sirven para cuantificar un determinado efecto cuántico a partir de definir medidas, desarrollar protocolos para su detección o identificar procesos que permitan optimizar su aprovechamiento para una aplicación dada [18]. Debido a esto, las TRC han propiciado ver estos efectos con miras a la utilidad y no solo como fenómenos físicos.

2.2. Operaciones libres

Las operaciones libres son aquellas transformaciones permitidas dentro de una teoría de recursos que no generan el recurso en cuestión. Es decir, son operaciones que conservan la cantidad de recurso y, por lo tanto, no permiten obtener un estado con mayor cantidad de este a partir de uno con menor cantidad. Estas operaciones son consideradas “gratuitas”, en el sentido de que no requieren gasto del recurso [18].

Las restricciones que definen qué operaciones son libres dependen de la teoría de recursos específica que se esté considerando, por lo que pueden variar drásticamente entre distintos

marcos teóricos. Sin embargo, todas estas comparten una característica fundamental: No es posible generar recursos a partir de operaciones libres [18].

En el contexto de la mecánica cuántica, las operaciones libres no incluyen todas las transformaciones permitidas por la teoría cuántica, sino que representan un subconjunto restringido. Como consecuencia, no todos los estados cuánticos pueden ser generados a partir de un estado inicial fijo utilizando solo operaciones libres.

La teoría de recursos de interés para este trabajo es la relacionada con la no gaussianidad cuántica y negatividad de Wigner, en la cual las operaciones libres se denominan Protocolos Gaussianos (PG), de los cuales los Protocolos Gaussianos Deterministas (PGD) son un subconjunto que preserva la traza [5], es decir, si se aplica una transformación unitaria U a un estado ρ , se cumple que $\text{Tr}[U\rho U^\dagger] = \text{Tr}[\rho]$. Esto permite que no se pierda información al aplicar U .

2.3. Estados libres

Son los estados accesibles o que se pueden generar exclusivamente mediante operaciones libres, es decir, sin necesidad de consumir el recurso que se esté considerando, de manera coloquial, puede decirse que son estados “gratuitos” [18]. Por ejemplo, en el marco de la termodinámica clásica, un sistema en equilibrio térmico a temperatura T es un estado libre, ya que puede obtenerse sin necesidad de gastar energía adicional.

Un principio fundamental en cualquier teoría de recursos es que las operaciones libres preservan los estados libres, así, si se aplica una operación de este tipo a un estado libre, el resultado seguirá siendo un estado libre. Esto garantiza que los estados de recurso no puedan obtenerse a partir de estados libres mediante operaciones permitidas dentro de la teoría. Así, cualquier estado que no sea libre, es denominado como un recurso [17, 19].

En el marco de la TRC que se estudiará se pueden definir los estados libres como aquellos cuya función de Wigner es positiva (Ec. 1.13). Es decir, para un conjunto de operadores de densidad en el espacio de Hilbert \mathcal{H} de un número arbitrario, pero finito se cumple [5]:

$$\mathcal{W}_+ = \{\rho \in F | W_\rho \geq 0\}.$$

En el marco de la teoría de recursos de la *negatividad de la función de Wigner*, estos son los estados libres [5].

2.4. Recursos

La existencia de recursos plantea una cuestión natural acerca de cuánto de ellos hay disponible dentro del sistema bajo estudio, por lo que, para responderla, este debe ser cuantificado y con ello surge la necesidad de definir una medida para hacerlo. Existen dos tipos de medidas relevantes relacionadas a un recurso en una TRC: las medidas *monótonas*, que cuantifican la cantidad existente y los *testigos*, que identifican la presencia del recurso mediante observables [5, 14].

2.4.1. Medidas monótonas

En una TRC, las *medidas monótonas* son funciones que cuantifican la cantidad de recurso presente en un estado o sistema cuántico. Para que una función M sea una medida monótona válida, debe cumplir con algunos principios característicos como:

1. **Orden entre recursos:** Si se tienen dos estados diferentes, a y b , cada uno con cierta cantidad de recurso, de los cuales a posee al menos la misma cantidad de recurso que b , entonces esto significa que es posible transformar a en b mediante operaciones permitidas (*operaciones libres*). En términos de la medida, esto significa que si a puede transformarse en b , entonces la medida asignada al recurso a debe ser mayor o igual a la medida asignada al b [20]:

$$M(a) \geq M(b).$$

En términos más intuitivos, esto quiere decir que si un recurso puede ser convertido en otro, la cantidad de recurso existente en el estado inicial debe ser, al menos, tanta como la del resultante.

2. **Cero para estados libres:** Si se denota a F como el conjunto que contiene a los estados libres, entonces puede decirse que las medidas monótonas asignan un valor nulo a los *estados libres* $\rho \in F$, ya que estos no poseen recurso:

$$M(\rho) = 0, \quad \forall \rho \in F,$$

lo cual indica que si un estado es considerado “gratuito”, hablando en términos del recurso, entonces no debe contribuir al valor de la medida [5].

3. **No creación de recursos mediante operaciones libres:** Si se tiene una *operación libre* C que actúa sobre un estado de recurso, $\sigma \notin F$, esta no debe incrementar el recurso en dicho estado. Es decir, la medida antes de aplicar la operación debe ser mayor o igual que la medida final [17]:

$$M(\sigma) \geq M(C(\sigma)).$$

Este principio implica que no se genera recurso adicional al aplicar una operación libre sobre algún estado de recurso.

Entropía relativa.

Existe una medida dentro de las teorías de recursos conocida como *entropía relativa*, con la que se busca cuantificar la cantidad de recurso en un sistema a partir de la desviación de un estado ρ con respecto a un estado libre $\sigma \in S$ [17, 19].

La entropía relativa de ρ respecto a σ se define formalmente como [19]:

$$S(\rho|\sigma) = \text{Tr}[\rho \log \rho] - \text{Tr}[\rho \log \sigma]. \quad (2.1)$$

Esta expresión mide la discrepancia entre los estados ρ y σ en términos de la información cuántica y resulta de utilidad para compararlos entre sí.

De aquí se desprende que:

- Si $\rho = \sigma$, entonces $S(\rho|\sigma) = 0$, lo que indica que ambos estados son idénticos.
- Si ρ y σ son muy diferentes, $S(\rho|\sigma)$ toma valores grandes, reflejando una gran discrepancia entre ellos.

No obstante, esta medida podría no ser suficiente, de modo que es de utilidad calcular la desviación entre un estado ρ y un conjunto de estados libres S , lo cual se logra a través de la *distancia de entropía relativa*, definida como:

$$E_r(\rho) = \inf_{\sigma \in S} S(\rho|\sigma). \quad (2.2)$$

Aquí, \inf quiere decir “ínfimo”, es decir, el valor mínimo posible de $S(\rho|\sigma)$ cuando σ varía dentro del conjunto de estados S . Esta medida cuantifica que tan diferente es ρ del conjunto de estados libres S a través de encontrar el estado $\sigma \in S$ más cercano a ρ en términos de la entropía relativa. De esta manera, si $\rho \in S$, entonces $E_r(\rho) = 0$, pero si resulta que $\rho \notin S$, entonces $E_r(\rho) > 0$. Cabe mencionar que la entropía relativa de un recurso es una medida monótona [18].

Para conocer la posibilidad de transformar n copias de un estado ρ en m copias de otro, por ejemplo σ , lo cual es una tarea fundamental en las Teorías de Recursos Cuánticos (TRC) y cuya tasa de conversión óptima se cuantifica en el límite asintótico de muchas copias, se introduce y estudia el comportamiento de la entropía relativa regularizada. Esta medida se define precisamente considerando un gran número de copias del estado

Para conocer la posibilidad de transformar n copias de un estado ρ en m copias de otro, por ejemplo σ , se estudia el comportamiento de la distancia de entropía relativa para una cantidad muy grande de copias de dicho estado. El motivo de utilizar una gran cantidad de copias es porque, de esta manera, las tasas de conversión óptimas se vuelven claras y estables en el límite asintótico ($n \rightarrow \infty$), algo que no siempre ocurre al considerar pocas copias. La medida que cuantifica este recurso en dicho límite asintótico, y que rige las tasas de conversión es la *entropía relativa regularizada*, definida como la entropía relativa por copia en el límite de un número infinito de copias [18, 19]:

$$E_r^\infty(\rho) = \lim_{n \rightarrow \infty} \frac{E_r(\rho^{\otimes n})}{n}. \quad (2.3)$$

Notar que aquí se ha definido $\rho^{\otimes n}$ como las n copias del estado ρ , así que no debe confundirse con un exponente.

La finalidad de buscar transformar un estado en otro es para observar si resulta ser más fácil de manipular. La *tasa de conversión asintótica* entre ρ y σ , dada como:

$$R(\rho \rightarrow \sigma) = \frac{E_r^\infty(\rho)}{E_r^\infty(\sigma)}, \quad (2.4)$$

define el número de copias de σ que pueden producirse por cada copia de ρ en el caso en que se tienen una gran cantidad de copias [17].

En la teoría cuántica, distinguir que tan diferentes son dos estados es clave para comprender la información que transportan. Cuando el objetivo es medir qué tanto un estado cuántico se aparta del conjunto de estados “libres” o clásicos, la entropía relativa se vuelve una herramienta muy útil. A través de su forma regularizada, esta medida revela cuánta que tan útil es un sistema en comparación con otro y qué tan eficiente puede ser su conversión en otros recursos. En el límite de muchas copias, estas herramientas nos dicen qué transformaciones cuánticas son posibles, qué tan costosas son y qué tanto se gana.

2.4.2. Testigos

Los *testigos* son funciones que están diseñadas para detectar la presencia de un recurso específico en un estado cuántico debido a que un testigo es un observable [17]. Es por esto que los testigos son útiles en el sentido experimental ya que si se preparan múltiples copias de un sistema cuántico en el mismo estado desconocido, un recurso puede detectarse siempre que un testigo produzca un valor esperado negativo [17]. Las características de los testigos son:

1. **No negatividad en estados libres:** Estos testigos están diseñados de forma que su valor esperado sea positivo para todos los estados pertenecientes al conjunto que se denominó como S , es decir, los estados libres [17]. Si se denota como W a esta observable, esto puede expresarse como:

$$\forall \rho \in F \Rightarrow \text{Tr}[W\rho] \geq 0. \quad (2.5)$$

2. **Sensibilidad a los recursos:** Como se ha mencionado, estas observables son sensibles a los recursos, es decir, su medición resulta negativa para un estado que contiene el recurso bajo estudio, [17], esto es:

$$\exists \sigma \notin F : \text{Tr}[W\sigma] < 0. \quad (2.6)$$

3. Cómputo cuántico de variable continua

Como ya se hizo una breve mención acerca del CCVC, este se basa en sistemas de variables continuas, como los modos del campo electromagnético o la posición y momento de una partícula cuántica.

En comparación con el CCVD, donde se manipulan qubits mediante aplicarles puertas lógicas cuánticas específicas, en el CCVC, las manipulaciones a menudo involucran transformaciones de las variables continuas, por ejemplo, comprimir el rango de posibles valores de la intensidad de la luz o desplazar la posición promedio de una partícula [6]. Una ventaja de este tipo de computación cuántica radica en que es potencialmente escalable y resistente al ruido [12] [21], no obstante, esta aún no ha alcanzado la universalidad. Es aquí en donde entra en juego la necesidad de una teoría de recursos que permita hacer cómputo universal mediante CCVC.

3.1. Teorema de Gottesman-Knill

El teorema de Gottesman-Knill (GK) es un resultado clave en la teoría de la información cuántica para sistemas de variables discretas. Establece que cualquier algoritmo cuántico que comience en la base computacional (es decir, estados que representan las configuraciones básicas de los qubits, como $|0\rangle$ y $|1\rangle$) y utilice solo una clase restringida de compuertas (como Hadamard, fase, CNOT y las compuertas de Pauli), junto con mediciones proyectivas en la misma base computacional, puede ser simulado eficientemente en una computadora clásica. Este teorema muestra que una gran clase de algoritmos cuánticos no proporciona una ventaja de velocidad sobre los procedimientos clásicos [4, 14].

También existe una extensión del teorema GK para variables continuas [4, 6]. Este establece condiciones para que un proceso de información cuántica de variable continua pueda ser simulado eficientemente utilizando una computadora clásica. Estas condiciones generalmente implican lo siguiente, [4, 14]:

- Comenzar con estados Gaussianos no entrelazados: El proceso debe iniciarse con estados Gaussianos, que pueden ser productos de estados de vacío desplazados y comprimidos.
- Realizar solo transformaciones generadas por Hamiltonianos que son cuadráticos en los operadores canónicos (posición o momento), como:
 - Desplazamientos en el espacio fase: Generados por polinomios lineales en q o p . Son análogos a las compuertas de Pauli, usadas en el CCVD. Estas son [4, 14]:

$$\hat{X}(q) = e^{-(i/\hbar)q\hat{p}}, \quad \hat{Z}(p) = e^{(i/\hbar)p\hat{q}} \quad \text{con } q, p \in \mathbb{R}. \quad (3.1)$$

3 CÓMPUTO CUÁNTICO DE VARIABLE CONTINUA

3.1 Teorema de Gottesman-Knill

Estos obedecen la relación $\hat{X}(q)\hat{Z}(p) = e^{-(i/\hbar)qp}\hat{Z}(p)\hat{X}(q)$. Y su acción sobre la base de posiciones es:

$$\hat{X}(q')|q\rangle = |q + q'\rangle, \quad \hat{Z}(p)|q\rangle = e^{(i/\hbar)qp}|q\rangle. \quad (3.2)$$

- Compresión en un oscilador: Este tipo de operaciones se encuentran definidas por el operador de compresión, dado por la Ec. [1.8], a saber este operador es: $\hat{S}(\xi) = e^{\frac{1}{2}(\xi^* \hat{a}^2 - \xi \hat{a}^\dagger)^2}$.
- Compuerta SUM: Es el análogo, en variable continua, a la compuerta CNOT de variable discreta, pues sirve como la compuerta de interacción básica para dos sistemas de osciladores o sistemas de variables continuas [4, 14]. Son componentes que permiten la simulación clásica eficiente. Formalmente, se define para dos sistemas i y j como [4, 14]:

$$\text{SUM}_{ij} = e^{-(i/\hbar)\hat{q}_i \otimes \hat{p}_j}. \quad (3.3)$$

La acción de la compuerta SUM_{ij} sobre las compuertas de Pauli es:

$$\hat{X}_i(q) \otimes \hat{I}_j \rightarrow \hat{X}_i(q) \otimes \hat{X}_j(q) \quad (3.4)$$

$$\hat{Z}_i(p) \otimes \hat{I}_j \rightarrow \hat{Z}_i(p) \otimes \hat{I}_j \quad (3.5)$$

$$\hat{I}_i \otimes \hat{X}_j(q) \rightarrow \hat{I}_i \otimes \hat{X}_j(q) \quad (3.6)$$

$$\hat{I}_i \otimes \hat{Z}_j(p) \rightarrow \hat{Z}_i(p)^{-1} \otimes \hat{Z}_i(p) \quad (3.7)$$

- Mediciones en la base de eigenestados de posición o momento, como la detección homodina, la cual mide una de las dos cuatraturas (posición o momento) de un modo bosónico [3, 14].
- Operaciones del grupo de Clifford, las cuales preservan la Gaussianidad de los estados, por ejemplo:
 - Compuerta de fase: Es una operación generada por un Hamiltoniano cuadrático (proporcional a \hat{q}^2 o \hat{p}^2). Es considerada como una operación de compresión para variable continua, la cual introduce una fase dependiente del cuadrado de la posición o el momento [4, 14].

$$P(\eta) = \exp\left(\frac{i}{2\hbar}\eta\hat{q}^2\right). \quad (3.8)$$

Su acción sobre los operadores de Pauli es:

$$\hat{X}(q) \rightarrow e^{\frac{i}{2\hbar}\eta\hat{q}^2} \hat{X}(q)\hat{Z}(\eta q), \quad (3.9)$$

$$\hat{Z}(p) \rightarrow \hat{Z}(p). \quad (3.10)$$

3.2 Estados de fase cúbica $|\gamma, r\rangle$

- Transformada de Fourier: Análoga a la compuerta de Hadamard, en el caso de los qubits, y es definida como [4, 14]:

$$\hat{\mathcal{F}} = \exp\left(\frac{i\pi}{4\hbar}(\hat{q}^2 + \hat{p}^2)\right). \quad (3.11)$$

Su acción sobre los operadores de Pauli es:

$$\hat{X}(q) \rightarrow \hat{Z}(q), \quad (3.12)$$

$$\hat{Z}(p) \rightarrow \hat{X}(p)^{-1}. \quad (3.13)$$

Procesos como la teleportación cuántica de variable continua, la criptografía cuántica y la corrección de errores, basadas en variable continua, pueden satisfacer estas condiciones y, por lo tanto, pueden ser simulados eficientemente de forma clásica. Aunque estos procesos son fundamentalmente cuánticos e involucran enredamiento, no proporcionan una aceleración sobre una simulación clásica bajo las restricciones del teorema extendido [14].

En resumen, el teorema GK y su extensión a variables continuas son herramientas cruciales para comprender la frontera entre lo que puede ser simulado eficientemente por una computadora clásica y lo que requiere recursos cuánticos para una simulación eficiente.

Para lograr la universalidad en la computación cuántica con variables continuas, es necesario combinar operaciones Gaussianas con algún elemento no-Gaussiano, el cual puede ser una operación unitaria no-Gaussiano (es decir, operaciones generadas por Hamiltonianos con polinomios de orden cúbico o superior en los operadores canónicos, como los estados de fase cúbica), que no preservan el grupo de Pauli [3, 4, 15, 22], también podría ser un estado no-Gaussiano [15, 23] o una medición del mismo tipo [3, 15] . y están fuera del grupo de Clifford. Como mención breve, el grupo de Pauli (o también conocido como el grupo de *Heisenberg-Weyl*) para variables continuas en n sistemas de osciladores acoplados está conformado por los operadores de desplazamiento en el espacio fase (Ec. 3.1) para los n osciladores [4, 6].

3.2. Estados de fase cúbica $|\gamma, r\rangle$

Como es bien sabido, un estado físico válido dentro de la teoría cuántica debe ser normalizable, ya que esta propiedad garantiza que las predicciones teóricas puedan ser comparadas consistentemente con los resultados experimentales [14]. Esta exigencia cobra relevancia al considerar ciertas construcciones teóricas que, si bien son útiles conceptualmente, no siempre corresponden a estados físicamente realizables.

Un ejemplo representativo es el llamado estado de fase cúbica ideal, el cual surge al aplicar un operador de fase cúbica, como $e^{i\gamma\hat{q}^3}$ (una operación no-Gaussiano), al estado de vacío $|0\rangle$. Este estado se define como $|\gamma\rangle = e^{i\gamma\hat{q}^3}|0\rangle_p$ donde \hat{q} es el operador de posición y $|0\rangle_p$ es el estado de momento igual a cero [3, 15, 23]. Sin embargo, este último no es normalizable [3, 15, 23], ya que se trata de un eigenestado impropio [23] y se encuentra

3.2 Estados de fase cúbica $|\gamma, r\rangle$

fueras del espacio de Hilbert [3]. Por esta razón, $|\gamma\rangle$ no puede considerarse un estado físico válido.

El estado de fase cúbica que resulta útil como recurso en computación cuántica debe ser físicamente generable. Para ello, se recurre a una versión “imperfecta” del estado ideal, la cual se obtiene al aplicar la compuerta de fase cúbica, $e^{i\gamma\hat{q}^3}$, sobre un estado de vacío comprimido [15, 23]:

$$|\gamma, r\rangle = \hat{\Gamma}(\gamma) \hat{S}(\xi) |0\rangle, \quad (3.14)$$

tal que:

$$\hat{\Gamma}(\gamma) = e^{i\gamma\hat{q}^3}, \quad \hat{S}(\xi) = e^{\frac{1}{2}(\xi^* \hat{a}^2 - \xi \hat{a}^{\dagger 2})}, \quad (3.15)$$

en donde \hat{q} es el operador de posición.

Invocando la expresión del operador de compresión actuando sobre el estado de vacío (estado de vacío comprimido) en términos de estados de Fock (Ec. 1.11):

$$|\xi\rangle = \frac{1}{\sqrt{\cosh r}} \sum_{m=0}^{\infty} (-1)^m \frac{\sqrt{(2m)!}}{2^m m!} e^{im\theta} (\tanh r)^m |2m\rangle = A_m(r, \theta) |2m\rangle. \quad (3.16)$$

En la que, en la última igualdad, se ha hecho:

$$A_m(r, \theta) = \frac{1}{\sqrt{\cosh r}} \sum_{m=0}^{\infty} (-1)^m \frac{\sqrt{(2m)!}}{2^m m!} e^{im\theta} (\tanh r)^m.$$

Se busca realizar una comparación con los resultados mostrados en [23], por lo que se hará la elección de tomar $\theta = \pi$, y como $\xi = re^{i\theta}$, entonces se tendrá que $\xi = -r$, y con ello $A_m(r, \theta) \rightarrow A_m(r, \pi)$.

El paso siguiente para calcular el estado $|\gamma, r\rangle$ es aplicar el operador de fase cúbica $\hat{\Gamma}(\gamma)$ al estado de vacío comprimido (Ec. 1.11), sin embargo, la naturaleza de este operador involucra no linealidad, de manera que realizar una expansión en serie del mismo sería prudente para poder continuar con mayor facilidad:

$$\begin{aligned} \hat{\Gamma}(\gamma) &= e^{i\gamma\hat{x}^3} \approx \sum_{j=0}^{\infty} \frac{(i\gamma\hat{x}^3)^j}{j!} = \sum_{j=0}^{\infty} \frac{(i\gamma)^j}{j!} \hat{x}^{3j} \\ &= 1 + i\gamma\hat{x}^3 - \frac{\gamma^2}{2}\hat{x}^6 - i\frac{\gamma^3}{6}\hat{x}^9 + \frac{\gamma^4}{24}\hat{x}^{12} + \dots \end{aligned} \quad (3.17)$$

3.2 Estados de fase cúbica $|\gamma, r\rangle$

Al calcular la acción consecutiva del operador \hat{x} de la expansión en serie del operador de fase cúbica $\hat{\Gamma}(\gamma)$ se obtiene que:

$$\hat{x}A_m(r, \theta)|2m\rangle = cA_m(r, \theta) \left(\sqrt{2m}|2m-1\rangle + \sqrt{2m+1}|2m+1\rangle \right),$$

en donde:

$$c = \sqrt{\frac{\hbar}{2}},$$

así, al volver a aplicar \hat{x} :

$$\begin{aligned} \hat{x}^2 A_m |2m\rangle &= c^2 A_m \left(\sqrt{2m}\sqrt{2m-1}|2m-2\rangle + \sqrt{2m+1}\sqrt{2m+1}|2m\rangle \right. \\ &\quad \left. + \sqrt{2m}\sqrt{2m}|2m\rangle + \sqrt{2m+1}\sqrt{2m+2}|2m+2\rangle \right) \\ &= c^2 A_m \left(\sqrt{(2m)(2m-1)}|2m-2\rangle + (2m+1+2m)|2m\rangle \right. \\ &\quad \left. + \sqrt{(2m+1)(2m+2)}|2m+2\rangle \right) \\ &= c^2 A_m \left(\sqrt{(2m)(2m-1)}|2m-2\rangle + (4m+1)|2m\rangle \right. \\ &\quad \left. + \sqrt{(2m+1)(2m+2)}|2m+2\rangle \right). \quad (3.18) \end{aligned}$$

Por lo que, aplicando \hat{x} una vez más, se obtiene:

$$\begin{aligned} \hat{x}^3 A_m |2m\rangle &= c^3 A_m \left(\sqrt{(2m)(2m-1)}\sqrt{2m-2}|2m-3\rangle \right. \\ &\quad \left. + (4m+1)\sqrt{2m}|2m-1\rangle + \sqrt{(2m+1)(2m+2)}\sqrt{2m+2}|2m+1\rangle \right. \\ &\quad \left. + \sqrt{(2m)(2m-1)}\sqrt{2m-1}|2m-1\rangle \right. \\ &\quad \left. + (4m+1)\sqrt{2m+1}|2m+1\rangle + \sqrt{(2m+1)(2m+2)}\sqrt{2m+3}|2m+3\rangle \right) \end{aligned}$$

3.2 Estados de fase cúbica $|\gamma, r\rangle$

$$\begin{aligned}
&= c^3 A_m \left[\sqrt{(2m)(2m-1)(2m-2)} |2m-3\rangle + \left((4m+1)\sqrt{2m} \right. \right. \\
&\quad \left. \left. + (2m-1)\sqrt{2m} \right) |2m-1\rangle + \left((2m+2)\sqrt{2m+1} \right. \right. \\
&\quad \left. \left. + (4m+1)\sqrt{2m+1} \right) |2m+1\rangle + \sqrt{(2m+1)(2m+2)(2m+3)} |2m+3\rangle \right] \\
&= c^3 A_m \left(\sqrt{(2m)(2m-1)(2m-2)} |2m-3\rangle + 6m\sqrt{2m} |2m-1\rangle \right. \\
&\quad \left. + (6m+3)\sqrt{2m+1} |2m+1\rangle + \sqrt{(2m+1)(2m+2)(2m+3)} |2m+3\rangle \right). \quad (3.19)
\end{aligned}$$

Como puede notarse de la expansión de $\hat{\Gamma}(\gamma)$ (Ec. 3.17), los operadores que actuarán sobre el estado $|-\rangle$ son las potencias $3j$ -ésimas de \hat{x} . Por lo que, realizar la acción de cada uno de ellos conllevaría un gran trabajo algebraico para valores grandes de j . Para realizar esta tarea sería más práctico hacerlo mediante un código computacional.

Una vez que se ha obtenido la expansión del operador $\hat{\Gamma}(\gamma)$, se aplica sobre el estado $|\xi\rangle$, de manera que se logra llegar a una aproximación del estado $|\gamma, r\rangle$. Los resultados obtenidos se mostrarán posteriormente.

4. Negatividad de la función de Wigner

La teoría de recursos basada en la negatividad de la función de Wigner tiene como objetivo identificar y cuantificar las propiedades cuánticas que permiten ventajas en tareas de procesamiento de información cuántica [5].

En esta teoría, los *estados libres* son aquellos cuya función de Wigner es completamente positiva en todo el espacio de fase (ver sección 2.3). Este conjunto de estados, denotado como \mathcal{W}_+ , incluye tanto a los estados Gaussianos puros (aquellos estados $|\psi\rangle\langle\psi|$ que pueden describirse mediante un operador de densidad de la forma $\hat{\rho} = |\psi\rangle\langle\psi|$) como a los mixtos (cuyo operador de densidad se escribe como una suma ponderada de estados $|\psi_i\rangle\langle\psi_i|$, es decir: $\hat{\rho} = \sum_i p_i |\psi_i\rangle\langle\psi_i|$) [1]. De hecho, Hudson [8] muestra que los estados puros cuya función de Wigner es positiva son los estados Gaussianos [5]. Sin embargo, también existen ciertos estados *mixtos* con función de Wigner positiva que no pueden escribirse como combinaciones de estados Gaussianos [5]. No obstante, este trabajo se centra en los estados puros.

Las *operaciones libres* en esta teoría están compuestas por transformaciones cuadráticas (como las transformaciones generadas por Hamiltonianos de segundo orden) y por mediciones Gaussianas imprecisas, conocidas como mediciones *coarse-grained* [5]. Estas últimas son operaciones consideradas dentro de los protocolos operacionales Gaussianos, relevantes también en la teoría de recursos de la no-Gaussianidad, y se caracterizan por lo siguiente:

- Primero, se realiza una medición Gaussianas (como una medición de la posición o el momento).
- Luego, se aplica una operación condicional que depende del resultado de esa medición.
- En la práctica, los dispositivos de medición tienen resolución finita, por lo que los resultados no se obtienen como valores exactos, sino agrupados en intervalos. Por ejemplo, en vez de obtener el valor exacto 1.2345 ..., se dice que el valor cae en un rango como [1.2, 1.3]. Esta agrupación implica una imprecisión que puede llevar a la obtención de un estado *mixto* como resultado de la medición.

Por otra parte, en esta teoría, los estados de recurso son aquellos estados $\sigma \notin \mathcal{W}_+$, es decir, aquellos cuya función de Wigner no es completamente positiva. Además de esto, la negatividad logarítmica de la función de Wigner (NLW) es la medida monótona dentro de esta TRC [5].

Sobre la existencia de los testigos, en la página 34 de la Ref.[5], sección *E. Witness-based measures*, se menciona que: "Para la teoría de recursos de la no Gaussianidad con los estados Gaussianos o los estados con función de Wigner no negativa como estados libres, entonces el teorema del hiperplano separador asegura que cada estado de recurso posee al menos un testigo".

 4.1 NLW como medida del recurso

4.1. NLW como medida del recurso

La negatividad de la función de Wigner es considerada un recurso cuántico debido a su profunda conexión con la no-clasicidad de los estados cuánticos. Los estados con función de Wigner negativa son, esenciales para obtener una ventaja computacional sobre los algoritmos clásicos en ciertas tareas [5]. Esto se debe a que las tareas que involucran únicamente estados con función de Wigner completamente positiva y operaciones que no pueden crear negatividad, pueden ser simulados eficientemente de manera clásica [15].

Para cuantificar la cantidad de negatividad, y por ende de recurso, presente en un estado, se recurre a la negatividad logarítmica de la función de Wigner (NLW) ya que cumple con las propiedades de una medida monótona [5], y para un estado ρ se define como:

$$\text{NLW}(\rho) = \log \left(\int |W_\rho(q, p)| dq dp \right), \quad (4.1)$$

en donde la integral se realiza sobre todo el espacio, mientras que, aunque la base del logaritmo es irrelevante para la definición y no afecta en sus propiedades, se ha utilizado el logaritmo en base 2, lo cual coincide con [5].

La NLW se considera computable en el sentido de que su valor puede evaluarse mediante integración numérica, no obstante, obtener expresiones analíticas puede ser difícil debido a la necesidad de integrar el valor absoluto de la función de Wigner, $W(q, p)$, lo que requiere encontrar sus ceros [5].

Cabe mencionar que, debido a la normalización que debe usarse para que un estado cuántico ρ sea válido y tenga un sentido probabilístico, la integral de $W(q, p)$ debe ser igual a la unidad, es decir:

$$\int W(q, p) dq dp = Tr[\rho] = 1$$

Para el caso en el que ρ sea un estado clásico, por ejemplo algún estado coherente, comprimido o térmico, su función de Wigner será Gaussiana y completamente positiva, cumpliendo con que la integral en la Ec. (4.1) será igual a 1, y con ello haciendo que $\text{NLW} = 0$.

Por otra parte, si el estado en cuestión resulta ser no clásico, como los estados de Fock, los del tipo de gato de Schrödinger o, como los considerados en este trabajo, estados de fase cúbica, su función de Wigner podría tomar valores negativos, lo cual reflejaría interferencia cuántica y, en general, su naturaleza no clásica [24]. En este caso, se obtendría que $\int |W(q, p)| dq dp > 1$, y consecuentemente, $\text{NLW} \neq 0$.

5. Resultados

Con base en la teoría explicada en las secciones previas, se han realizado códigos, escritos en Python, para obtener, de manera numérica, los valores de la negatividad logarítmica de la función de Wigner (NLW) para algunos estados de fase cúbica, $|\gamma, r\rangle$. La elección de los valores de γ y r no ha sido de manera arbitraria, sino que se busca replicar los resultados mostrados en [23], además de dar una aproximación del estado que maximiza la NLW.

Como primer paso, se realiza una comparación gráfica de la función de Wigner en el espacio fase del mismo estado a través de dos códigos: El primero (Sec. A.1) sirve para poder visualizar la acción aproximada del operador de fase cúbica, $\hat{\Gamma}(\gamma)$, sobre el estado de vacío comprimido (Ec. 1.11) como una expansión en serie, tal como se menciona en la sección 3.2, utilizando los valores ($\gamma = 0.1, r = 0.6$) para los parámetros del estado. Nota: Este código realiza la Fig. 4, particularmente, la correspondiente al inciso (b) se calcula mediante la "segunda mitad del código", la cual se indica con el comentario:

```
# Grafica 2: Aproximacion por expansion en serie dentro del mismo.
```

Debido a que en los términos de la serie, $\frac{(i\gamma\hat{x})^{3j}}{j!}$, para valores “pequeños” de j , el numerador domina sobre el denominador (para este caso en particular, $j < 15$), entonces se requieren valores de j suficientemente grandes para que el denominador sea quien domine. Este comportamiento puede visualizarse en la Fig. 3, en la que se observa que, a partir de $j = 15$, es el denominador quien domina. De aquí se desprende que una elección de $j = 150$ será más que suficiente. La función de Wigner asociada al estado de fase cúbica aproximado por este método puede observarse en la Fig. 4(b).

Por otra parte, el segundo código se ha implementado usando la función `wigner()`, incluida en la librería *QuTiP*, la cual puede calcular numéricamente la función de Wigner a través de 4 métodos distintos, basados en los estados de Fock, los cuales pueden indicarse mediante el valor del parámetro `method` de esta función [25], estos son:

- `wigner(method='iterative')`: Un método iterativo que calcula $W(q, p)$ para cada entrada de la matriz de densidad que se le ingrese como parámetro.
- `wigner(method='laguerre')`: Hace una aproximación por medio de los polinomios de Laguerre.
- `wigner(method='clenshaw')`: La aproximación se hace a través del método de *Clenshaw*. Este método se utiliza por defecto si no se especifica el parámetro `method` [25].
- `wigner(method='fft')`: Se utiliza la transformada rápida de Fourier.

El método que se utilizará en este trabajo es el *método de Clenshaw*, implementado por defecto por *QuTiP*, el cual evalúa series de funciones que satisfacen relaciones de recurrencia (como los polinomios de Laguerre o Hermite). Una relación de recurrencia permite calcular el siguiente término de una secuencia a partir de términos anteriores [26].

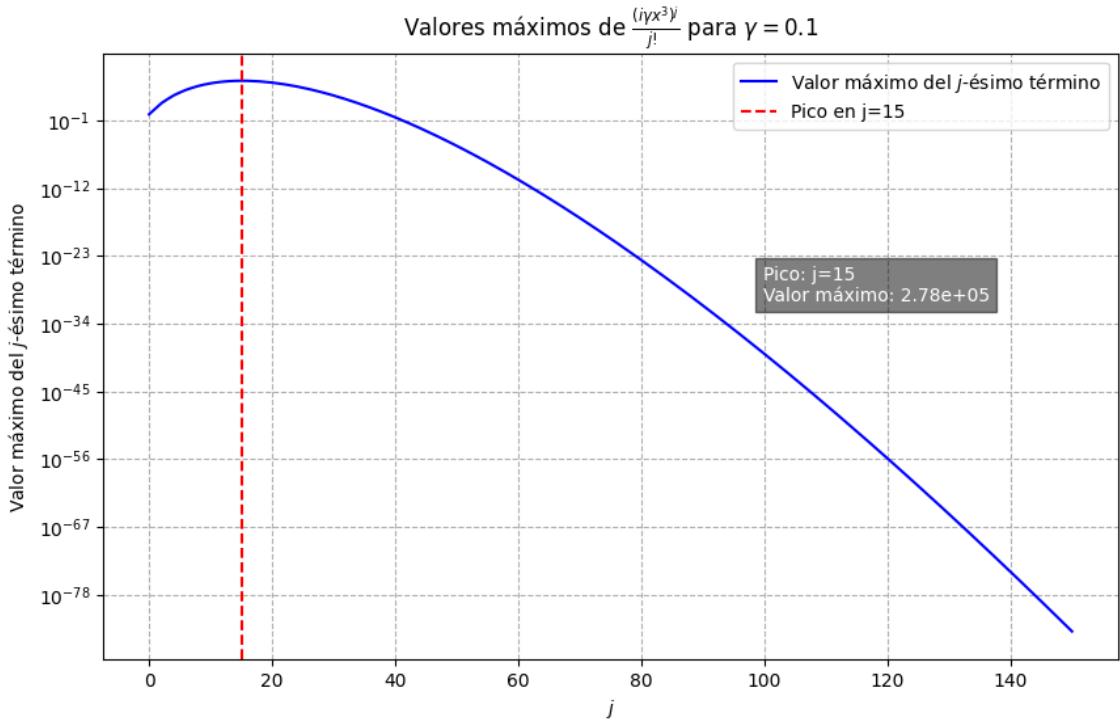


Figura 3: Se muestra el comportamiento del j -ésimo término de la expansión en serie del operador $\hat{\Gamma}(\gamma)$. A partir del término $j = 15$, el denominador comienza a dominar sobre el numerador.

Particularmente, este método es implementado por *QuTiP* para calcular la función de Wigner como [25]:

$$W(q, p) \propto \sum_L c_L \frac{(2x)^L}{\sqrt{L!}}, \quad (5.1)$$

donde los coeficientes c_L dependen de la matriz de densidad y polinomios de Laguerre generalizados. La implementación en *QuTiP* evalúa esta suma mediante una recurrencia iterativa hacia atrás mediante un ciclo `while`. Las ventajas de este método incluyen: estabilidad numérica ya que el método de Clenshaw es generalmente estable para la evaluación de la suma (en este caso, la suma sobre L), es decir, es bastante insensible a los errores de redondeo [27], lo cual minimiza la acumulación de estos errores. También tiene la virtud de ser eficiente computacionalmente al hacer uso de la recursividad [26], además de una alta precisión que se obtiene con relativamente pocos valores [25, 26]. Mediante el uso de este método implementado en *QuTiP*, para los mismos valores de los parámetros (γ, r) se tiene un resultado similar al obtenido a través del primer código. Este puede visualizarse en la Fig. 4(a).

Para estos valores específicos de (γ, r) , las funciones de Wigner obtenidas son bastante similares, sin embargo, esto no dicta con certeza que la aproximación de $\hat{\Gamma}(\gamma)$ por medio de una expansión en serie sea adecuada computacionalmente.

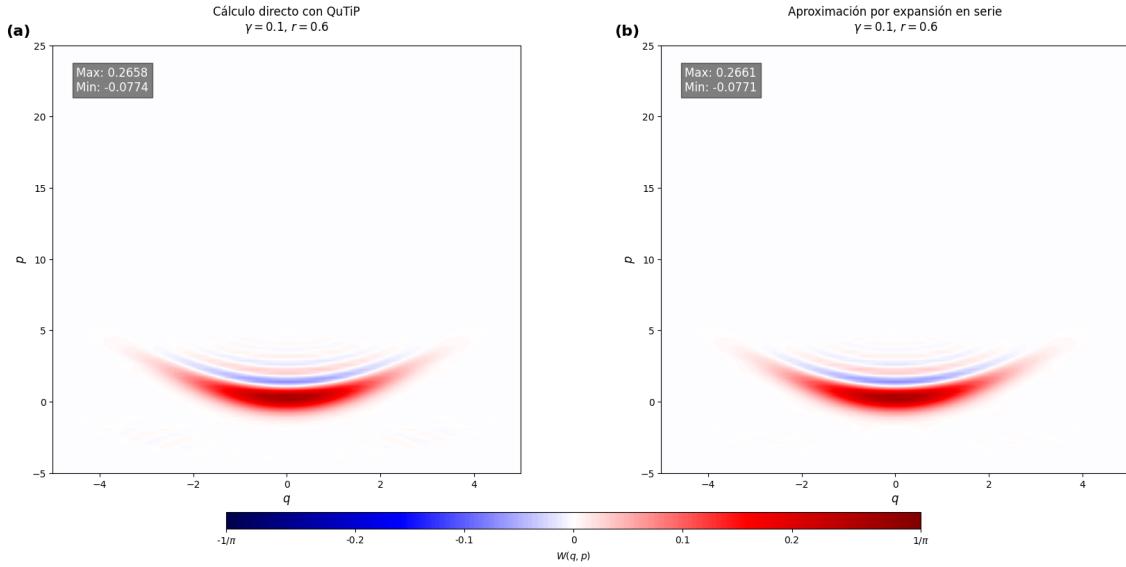


Figura 4: Comparación de la función de Wigner para el estado $|\gamma, r\rangle$ con $(\gamma = 0.1, r = 0.6)$ a través de: (a) usar la librería *QuTiP* y (b) la aproximación en serie del operador $\hat{\Gamma}(\gamma)$. El tamaño del espacio de Fock usado es $N = 35$.

5.1. Limitaciones numéricas de la expansión de $\hat{\Gamma}(\gamma)$

Para este punto es necesario mencionar las limitaciones observadas. Si se desea continuar con el método de la aproximación del operador $\hat{\Gamma}(\gamma)$ por medio de una serie de Taylor, se encuentra una limitante computacional conforme el valor de γ incrementa. En la Fig. 4 se ha utilizado un espacio de Hilbert de $N = 35$ dimensiones, y los parámetros $(\gamma = 0.1, r = 0.6)$, mientras que el orden de la expansión se ha tomado con base en el resultado que se muestra en la Fig. 3, lo cual genera elementos del orden de \hat{x}^{450} , es decir $j = 150$. Esto con el fin de obtener una buena aproximación además de que, considerando la ecuación 1.11, se ha hecho $m = 5$ (este valor se ha elegido debido a que en realidad no hay una gran diferencia entre tomar $m = 5$ o cualquier otro valor mayor, por ejemplo $m = 15$, sin embargo, este último requiere un espacio de Fock mayor, ya que, en este caso, se requieren $2m + 1 = N$ dimensiones debido a que se requiere por lo menos una dimensión en el espacio de Hilbert para poder ejecutar el código, por lo tanto si $m = 0 \rightarrow N = 1$). Cabe decir que, a menos que se especifiquen otros valores para los parámetros mencionados, por ahora se tomarán estos como fijos.

Ahora, si se incrementa el valor de γ , el resultado es cada vez menos semejante a lo que se esperaría. Un ejemplo de esto se ve claramente al intentar aproximar la función de Wigner para otros valores de parámetros, por ejemplo: $(\gamma = 0.4, r = 0.8)$ (el resto de parámetros se han mantenido fijos). Con estos valores, se esperaría un resultado similar al de la Fig. 7(a), el cual coincide, visualmente, con lo reportado en [23] pero lo que se obtiene se puede observar en la Fig. 5(b).

Este resultado ha incitado a preguntarse: *¿acaso debe considerarse una mayor cantidad de dimensiones del espacio de Hilbert para lograr una mejor aproximación?* Esta pregunta ha sugerido revisar los resultados que se obtendrían para distintas cantidades de

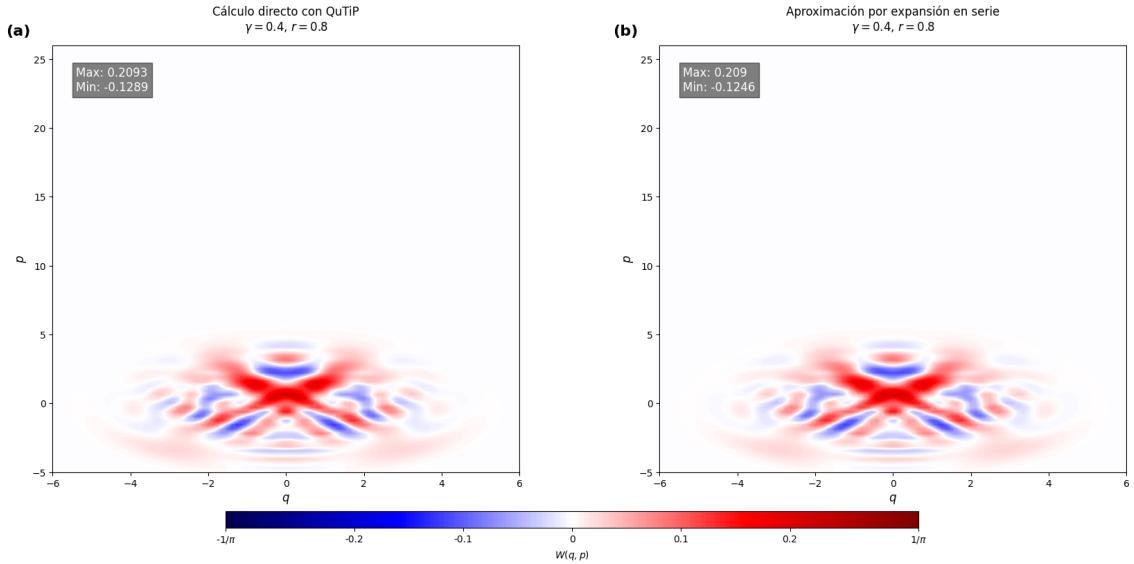


Figura 5: Comparación de la función de Wigner para el estado $|\gamma, r\rangle$ con $(\gamma = 0.4, r = 0.8)$ a través de: (a) usar la librería *QuTiP* y (b) la aproximación en serie del operador $\hat{\Gamma}(\gamma)$. El espacio de Fock usado es $N = 35$.

dimensiones, N . Se ha encontrado que para $N \geq 17$, los valores divergen (Fig. 6). En este conjunto de gráficas puede verse que, si se toma $N = 35$ (como se ha hecho para las gráficas de la Fig. 4), este método da una muy mala aproximación, de modo que para las gráficas de la Fig. 5 se ha tenido que usar $N = 15$, y aún con esto, el resultado no es para nada cercano a lo esperado con ninguno de los dos métodos (expansión en serie de $\hat{\Gamma}(\gamma)$ o utilizando *QuTiP*).

Esta mala aproximación se debe a un desbordamiento en el cálculo ya que el término \hat{x}^{3j} actúa en un espacio de Fock de dimensión finita (N), y su norma $\|\hat{x}^{3j}\|$ escala como $\mathcal{O}(N^{3j/2})$. Para valores de $\gamma \geq 0.2$ y $N = 15$, escala más rápidamente que para $\gamma = 0.1$, de modo que, en algún momento, se produce el desbordamiento para cierto valor de j , (el cual sucede de manera más rápida conforme N sea mayor) antes de que el denominador comience a dominar sobre el numerador y se logre la convergencia. Es decir, para valores cada vez mayores de γ , se requieren valores cada vez más grandes de j para lograr la convergencia, sin embargo estos valores grandes de j causan el desbordamiento.

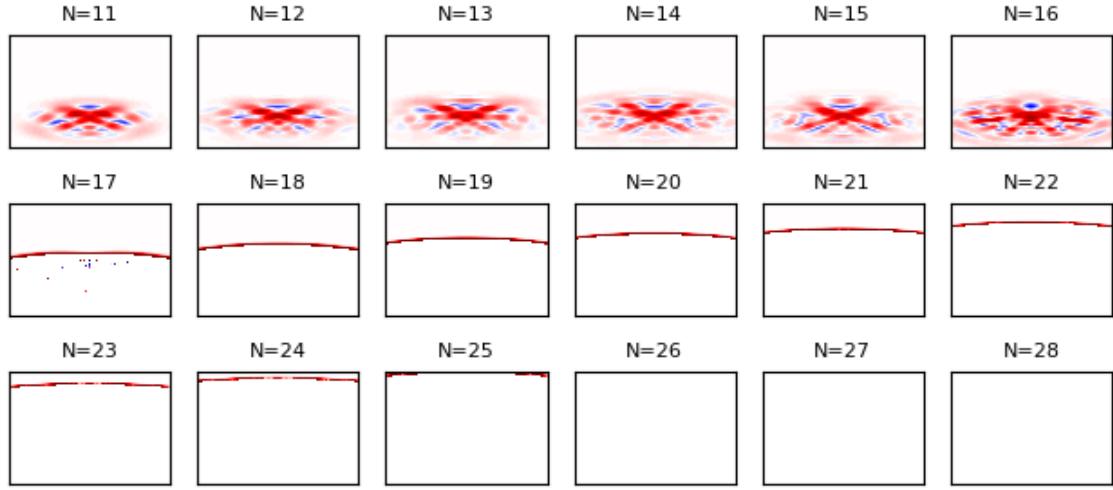


Figura 6: Función de Wigner para la aproximación del estado $|\gamma, r\rangle$ con $(\gamma = 0.4, r = 0.8)$ por medio de una serie. Se usan dimensiones diferentes del espacio de Fock (N).

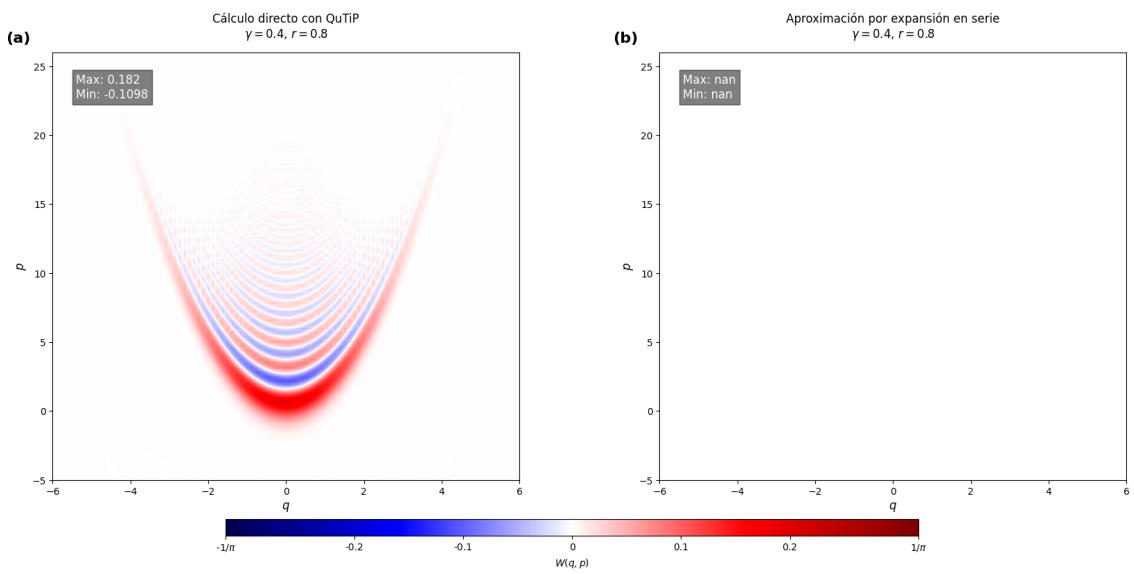


Figura 7: Comparación de la función de Wigner para el estado $|\gamma, r\rangle$ con $(\gamma = 0.4, r = 0.8)$ a través de: (a) usar la librería *QuTiP* y (b) la aproximación en serie del operador $\hat{\Gamma}(\gamma)$ (se muestra en blanco debido a que, para este espacio de Fock, $N = 400$, ya se ha llegado al desbordamiento numérico).

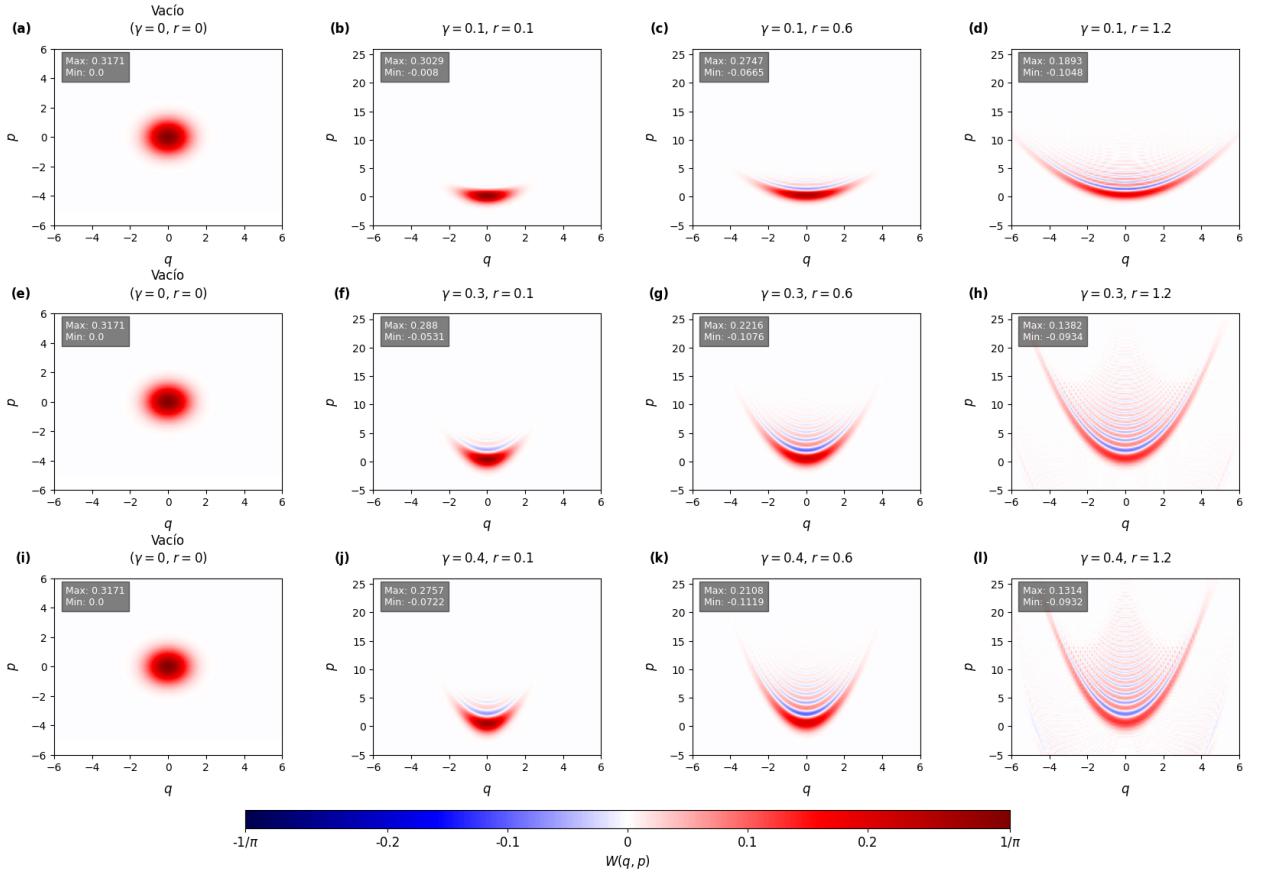


Figura 8: Funciones de Wigner bajo el efecto de incrementar r , para un mismo valor de γ en el estado $|\gamma, r\rangle$. Cada renglón inicia con el estado de vacío y, a la derecha, se sigue mantenerse fijo un mismo valor de γ mientras, para cada uno, se incrementa r .

En vista de estas limitaciones, el uso de las funciones predefinidas que están incluidas en la librería *QuTiP*, simplifica bastante el código necesario para la obtención de las funciones de Wigner para $|\gamma, r\rangle$ con distintos parámetros, de manera que se trabajará con esta librería, y aprovechando su estabilidad, ahora los estados se aproximarán con un espacio mucho más grande, de $N = 400$ dimensiones, para mejorar la precisión. Como ya se mencionó antes, se continuará con el *método de Clenshaw* (implementado en *QuTiP*) para calcular las funciones de Wigner ya que para excitaciones grandes (> 50) éste método es rápido y numéricamente estable [25]. Las gráficas de la Fig. 7 se han hecho considerando $N = 400$. Particularmente la Fig. 7(b) muestra lo que podría extrapolarse a partir de la Fig. 6. Para un espacio de Hilbert de este tamaño, el método de la aproximación en serie de $\hat{\Gamma}(\gamma)$ ha provocado un desbordamiento numérico.

En las figuras 8 y 9 se muestran un conjunto de funciones de Wigner para distintos valores de (γ, r) de tal modo que sirvan para obtener una idea del efecto que tiene variar alguno de estos parámetros mientras el otro se mantiene fijo.

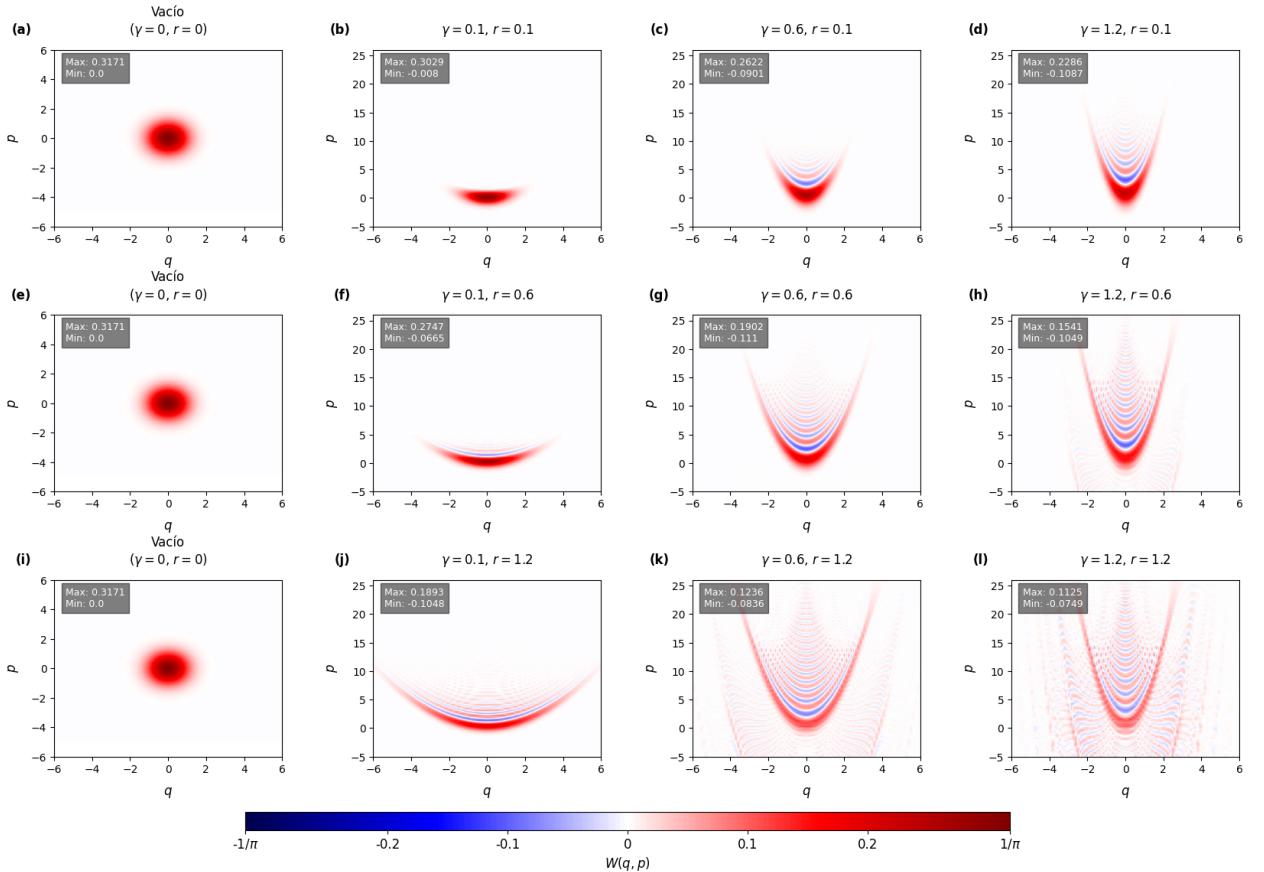


Figura 9: Funciones de Wigner bajo el efecto de incrementar γ , para un mismo valor de r en el estado $|\gamma, r\rangle$. Cada renglón inicia con el estado de vacío y, a la derecha, se sigue mantener fijo un mismo valor de r mientras, para cada uno, se incrementa γ .

De manera cualitativa, puede notarse en la Fig. 8 que, para algún valor fijo de γ , incrementar r “expande” horizontalmente la cuasi-distribución, lo cual es esperado debido a que es una compresión. Mientras que, de la Fig. 9, puede verse que el efecto de incrementar γ , manteniendo una r fija, es que aparecen patrones de “media luna” para valores tanto positivos (zonas rojas) como negativos (zonas azules). Estos patrones incrementan conforme la cubicidad (γ) crece. De hecho, se observa que si la cubicidad es no nula, comienzan a aparecer los valores negativos para la función de Wigner. Además, es notable que, una vez que existe cubicidad, la negatividad de la función aumenta si se incrementa cualquiera de los dos parámetros, aunque lo hace en mayor medida si la cubicidad es el parámetro que se incrementa.

Es prudente mencionar que el “ruido” en la función de Wigner obtenida para los valores $\gamma = r = 1.2$ (Fig. 9(I)) puede reducirse incrementando la dimensionalidad del espacio de Hilbert utilizado. No obstante, aún utilizando un espacio $N = 2000$ el ruido persiste y para valores todavía mayores, por ejemplo $N = 4000$, ya se tiene un desbordamiento numérico para el método que se está usando en este trabajo, además de requerir un lapso mucho mayor para la ejecución del código. En vista de esto, se continuará trabajando con el espacio de $N = 400$.

5.2. NLW para algunos estados

Ya se ha observado, de manera cualitativa, que incrementar alguno de los dos parámetros de los estados de fase cúbica incrementa las regiones negativas de la función $W(q, p)$, entonces, una vez obtenida esta función, se procede a calcular su negatividad logarítmica (NLW), la cual, como se mencionó en la sección 4.1, es adecuado calcularla de manera numérica. Así, también se ha escrito un código en Python que realiza la tarea (Sec. A.2), además de calcular $|W(q, p)|$, el cual, en caso de existir negatividad en algún estado, deberá ser superior a la unidad, indicando la presencia del recurso en dicho estado. También se calcula la integral de la función de Wigner, la cual deberá ser $W(q, p) \approx 1$, para todos los casos, debido a la naturaleza de la normalización.

En la Tabla I se muestran estos resultados para algunos estados elegidos de manera arbitraria pero que contienen a los mencionados en la Ref. [23]. Si se observa, es notorio lo que se manifestó algunos párrafos antes acerca de que γ tiene mayor influencia sobre la NLW que r , particularmente, $\text{NLW}_{0.1,0.6} \approx 0.3779 < \text{NLW}_{0.6,0.1} \approx 0.4947$.

γ	r	$NLW_{\gamma,r}$	$\int W_{\gamma,r}(q,p) dq dp$	$\int W_{\gamma,r}(q,p) dq dp$
0.1	0.6	0.3779	1.2994	0.9999
0.4	0.8	1.3423	2.5356	0.9940
0.1	0.1	0.0211	1.0148	0.9999
0.1	0.8	0.6424	1.5609	0.9999
0.05	0.1	0.0005	1.0004	0.9999
0.05	0.6	0.1431	1.1043	0.9999
0.6	0.1	0.4947	1.4090	0.9999

Tabla 1: Valores de la Negatividad Logarítmica de la función de Wigner (NLW) y las integrales asociadas para diferentes parámetros γ y r .

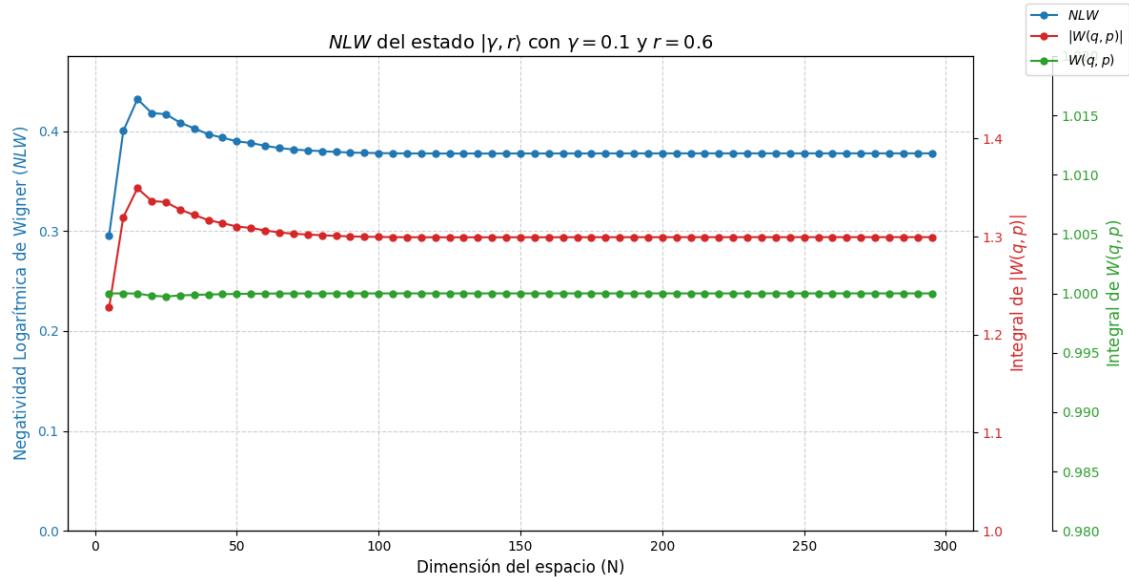


Figura 10: Evolución, en función de la dimensión del espacio de Hilbert (N), de la NLW , y las integrales $W(q,p)$ y $|W(q,p)|$ para el estado con $\gamma = 0.1$ y $r = 0.6$

Estos resultados incitan, naturalmente, a la pregunta: *¿cuál es el estado de fase cúbica con mayor cantidad de recurso?*, es decir, *¿cuál combinación de (γ, r) es la que maximiza la NLW ?* Como ya se ha mencionado, la dimensión del espacio de Hilbert que se utilice para las aproximaciones es un elemento clave que influye en la precisión de los resultados, lo cual conlleva a una pregunta más: *¿Qué valor de N dará una buena aproximación para estos estados con el fin de asegurar que los resultados obtenidos para la NLW sean muy cercanos a los que se obtendrían si se utilizase un espacio de Hilbert infinito?* Particularmente, *¿haber hecho $N = 400$ es una buena elección?*

Claramente, para dar respuesta a la primer pregunta, es necesario responder esta última. Esta puede obtenerse a partir de estimar una N suficientemente grande como para hacer que $\int W(q, p)dqdp \approx 1$. En este caso, resulta de bastante utilidad observar la evolución de la NLW, $\int |W(q, p)|dqdp$ y $\int W(q, p)dqdp$ en función de N . De manera particular, para el estado de fase cúbica con $(\gamma = 0.1, r = 0.6)$, esta evolución se muestra en la Fig. 10, para el cual se encontró que:

	$Prom \pm \sigma$
$NLW_{0.1,0.6}$	0.379660 ± 0.017603
$\int W_{0.1,0.6}(q, p) dqdp$	1.301129 ± 0.015714
$\int W_{0.1,0.6}(q, p)dqdp$	0.999985 ± 0.000046

Tabla 2: Estadísticas para $|\gamma, r\rangle$ con $(\gamma = 0.1, r = 0.6)$.

Por otra parte, también se ha hecho lo mismo para el estado con los parámetros más grandes que se considerarán, a saber, $\gamma = 0.5$ y $r = 1.7$. Así, para este estado, la evolución con respecto a N se observa en la Fig. 11, mientras que los promedios hallados para este estado se muestran en la Tabla 3. Estos valores surgen a partir del hecho de que, en las referencias utilizadas, se ha encontrado que el límite experimentalmente alcanzado para la amplitud de compresión es $r \approx 1.7$ [12], mientras que teóricamente se han explorado valores de $\gamma \approx 0.5$ (aunque esto último, para un valor de $r \approx 0.58$) [12].

	$Prom \pm \sigma$
$NLW_{0.5,1.7}$	2.080062 ± 0.263242
$\int W_{0.5,1.7}(q, p) dqdp$	4.286781 ± 0.609970
$\int W_{0.5,1.7}(q, p)dqdp$	0.758956 ± 0.058115

Tabla 3: Estadísticas para el estado de fase cúbica con $\gamma = 0.5$ y $r = 1.7$.

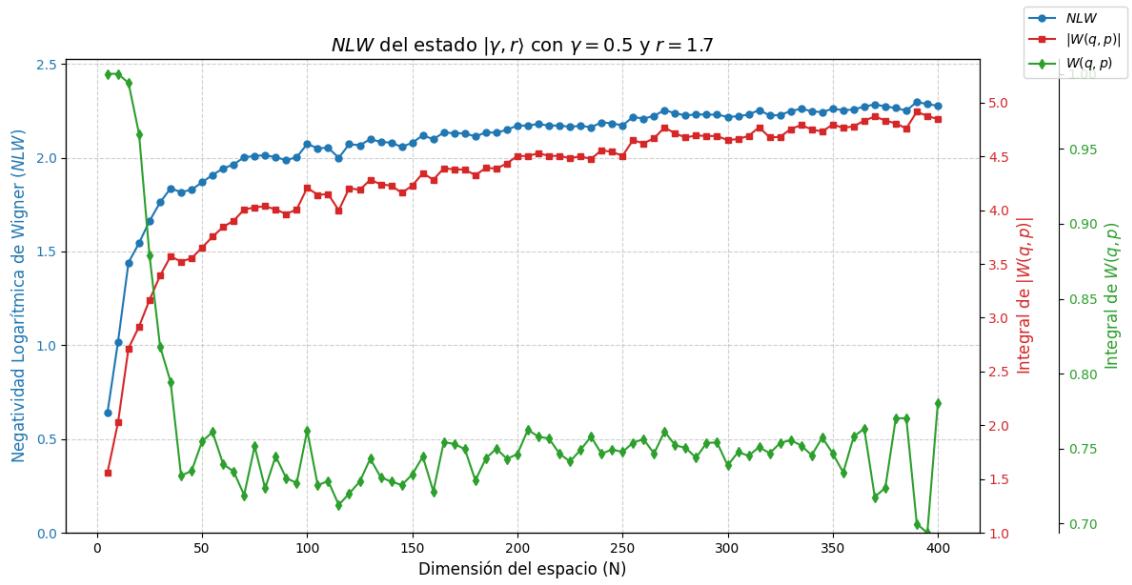


Figura 11: Evolución, en función de la dimensión del espacio de Hilbert (N), de la NLW, y las integrales de $W(q, p)$ y $|W(q, p)|$ para el estado con $\gamma = 0.5$ y $r = 1.7$.

Debido a que el valor promedio de $\int W_{0.5,1.7}(q, p)dqdp < 1$, el estado $|\gamma, r\rangle$ con valores ($\gamma = 0.5, r = 1.7$) no puede ser considerado como un estado válido para nuestros propósitos. Esto es así ya que, al no poder incrementar los límites de integración, el valor de esta integral no tiende a 1 (incrementarlos provoca un desbordamiento numérico, por lo que actualmente se han tomado los límites tales que $q \in [-6, 6]$ y $p \in [-5, 26]$). Siendo así que sería de esperarse que valores mayores de (γ, r) tampoco tiendan a la unidad, de modo que este par de valores pueden ser buenos candidatos para ser valores máximos a considerar para estos parámetros.

Entonces, para responder la pregunta sobre los valores que maximizan la NLW se ha recurrido a un código que realice un barrido de todas las combinaciones posibles para los pares (γ, r) tales que $\gamma \in [0.01, 0.05]$ y $r \in [0.1, 1.7]$ con pasos discretos de 0.01 entre cada valor de γ y de 0.1 entre cada valor de r . Esto se ha graficado en un mapa de calor (Fig. 12), para el cual, además de buscar aquel estado que maximiza la NLW , se ha condicionado para que cumpla con $\int W(q, p)dqdp \approx 1 \pm 0.02$, es decir, se ha dado (arbitriariamente) una tolerancia del 2 %. Así se han encontrado que los parámetros óptimos son $\gamma = 0.29$ y $r = 1.3$, dando los valores mostrados en la Tabla 4.

NLW_{opt}	≈ 1.9019
$\int W_{opt}(q, p) dqdp$	≈ 3.7371
$\int W_{opt}(q, p)dqdp$	≈ 0.9827

Tabla 4: Estadísticas para el estado de fase cúbica con parámetros óptimos encontrados ($\gamma = 0.29, r = 1.3$).

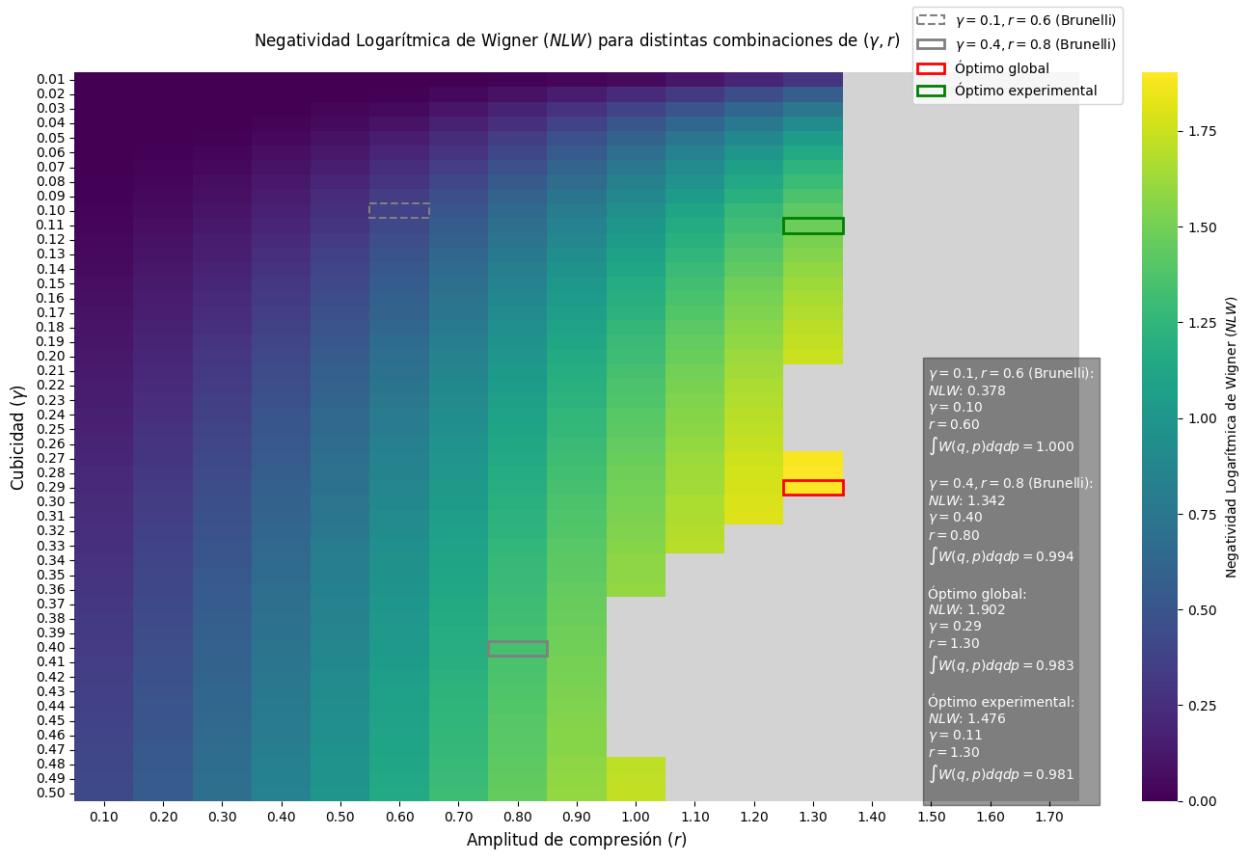


Figura 12: Mapa de calor para diferentes valores de (γ, r) . En color gris se muestran los estados que no cumplen con la normalización, es decir: $\int W(q, p)dqdp \not\approx 1.00 \pm 0.02$.

Este mapa de calor (Fig. 12) también muestra los dos estados mencionados en la Ref. [23], es decir, aquellos con parámetros $(\gamma = 0.1, r = 0.6)$ y $(\gamma = 0.4, r = 0.8)$ (en recuadros color gris con línea punteada y sólida, respectivamente), además de remarcar el estado que maximiza la NLW (color rojo) y el estado que combina los límites experimentalmente alcanzados (de manera independiente) para la amplitud de compresión $r = 1.7$ [12], y la cubicidad $\gamma = 0.11$ [28, 29] (color verde). Debido a que se ha respetado la condición de que el valor de la normalización esté dentro de la tolerancia del 2% otorgada, este último estado se ha recorrido al correspondiente a los valores $(\gamma = 0.11, r = 1.3)$, pues, como puede verse en el mapa de calor, el estado con parámetros $(\gamma = 0.11, r = 1.7)$ cae en la región grisácea, de modo que no cumple con el criterio de normalización impuesto, nuevamente, debido a que los límites de integración no son lo suficientemente “grandes” como para capturar toda la información del estado.

La información obtenida para este estado se muestra en la Tabla 5:

NLW_{exp_1}	≈ 1.9537
$\int W_{exp_1}(q, p) dqdp$	≈ 3.8736
$\int W_{exp_1}(q, p)dqdp$	≈ 0.8869

Tabla 5: Información para el estado de fase cúbica con parámetros máximos alcanzados experimentalmente ($\gamma = 0.11, r = 1.7$). Este estado no cae dentro de la tolerancia otorgada a la normalización.

Por otra parte, para el estado que si ha cumplido con la normalización, es decir, el que tiene los valores ($\gamma = 0.11, r = 1.3$), la información se muestra en la Tabla 6:

NLW_{exp_2}	≈ 1.4755
$\int W_{exp_2}(q, p) dqdp$	≈ 2.7808
$\int W_{exp_2}(q, p)dqdp$	≈ 0.9808

Tabla 6: Información para el estado de fase cúbica con parámetros cercanos a los máximos alcanzados experimentalmente ($\gamma = 0.11, r = 1.3$). Este estado si cae dentro de la tolerancia otorgada a la normalización.

Otro resultado que puede ser interesante, es visualizar la evolución de la NLW en función de los parámetros (γ, r), variando únicamente uno de los dos y manteniendo fijo el otro. Esto se ha hecho únicamente para tres valores de γ y r , particularmente, los correspondientes a los mencionados en la Ref. [23] y los que maximizan la NLW, según lo encontrado en el mapa de calor de la Fig. 12. Gráficamente esto aparece en las Figs. 13 y 14.

Las últimas dos figuras (13 y 14) muestran la evolución de la NLW en función de r y γ , respectivamente. En ellas puede observarse que, para valores fijos de cubicidad, conforme esta es mayor, la NLW también lo será; a partir de esto, conforme el valor de r se incrementa, nuevamente la NLW lo hará de manera casi lineal. Sin embargo, este incremento contrasta con el que se tiene si, en lugar de γ , es r el parámetro fijo, pues el incremento de la negatividad logarítmica conforme la cubicidad aumenta sucede de manera más rápida para valores pequeños de γ siguiendo una forma parecida al crecimiento que muestra la función raíz cuadrada.

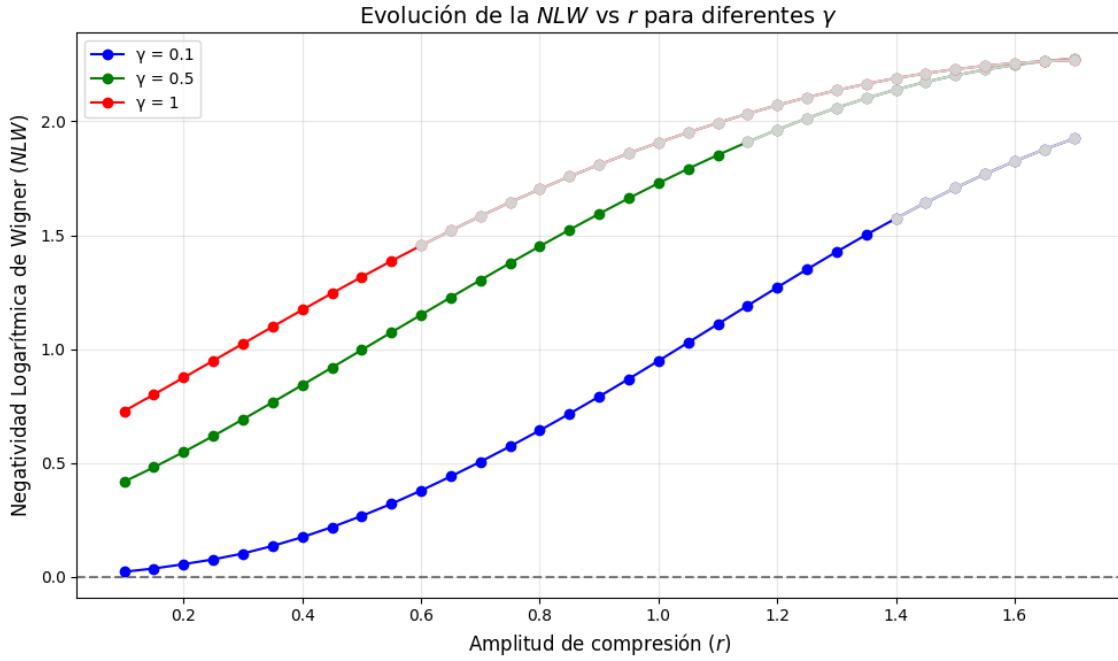


Figura 13: Evolución de $NLW(r)$ para los valores fijos de $\gamma = 0.1$, $\gamma = 0.29$ y $\gamma = 0.5$. Las zonas grises son asociadas a los estados que no cumplen con la normalización buscada.

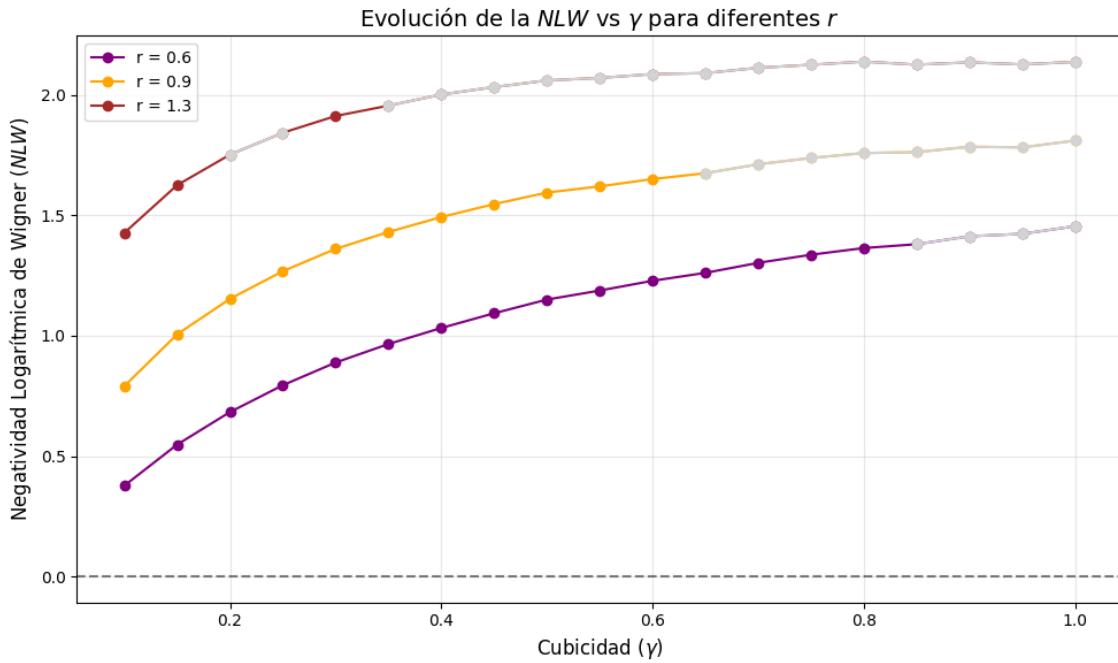


Figura 14: Evolución de $NLW(\gamma)$ para los valores fijos de $r = 0.6$, $r = 0.9$ y $r = 1.3$. Las zonas grises son asociadas a los estados que no cumplen con la normalización buscada.

6. Conclusiones

El objetivo principal, además de caracterizar los estados de fase cúbica, ha sido determinar la combinación de valores (γ, r) que maximicen el recurso presente en este tipo de estados mediante la negatividad logarítmica de la función de Wigner, lo cual ha podido lograrse de manera local en un rango específico de valores para estos parámetros, siendo limitados por errores numéricos.

Como ya se ha comentado, existen diferentes factores que juegan un papel importante en la precisión de los resultados que pueden obtenerse sobre la NLW calculada mediante los métodos empleados en este trabajo. Uno de ellos tiene que ver con el tamaño del espacio de Hilbert, el cual debe ser lo suficientemente grande como para poder generar una buena aproximación de la función de Wigner de los estados de fase cúbica. Sin embargo, esto no aplica para la aproximación de $\hat{F}(\gamma)$ mediante una serie de Taylor, ya que la convergencia de la serie no se alcanza antes del desbordamiento numérico para valores “grandes” de N . No obstante, al hacer uso de *QuTiP*, parece ser que la elección de $N = 400$ ha entregado resultados razonables que no distan demasiado de los que se obtienen mediante espacios mucho mayores y que, por ende, son más demandantes de poder de cómputo.

Por otra parte, este no es el único factor, pues el espacio sobre el que se integra para obtener $W(q, p)$ también es relevante, ya que, para distintos valores de γ y r , se requiere integrar sobre espacios más grandes. No obstante, esto provoca un desbordamiento computacional, por lo que no es posible incrementar los límites de integración arbitrariamente. Esto se ha reflejado en las zonas grises en el mapa de calor mostrado (Fig. 12) ya que, si se pudieran incrementar estos límites, es probable que pudieran cumplir con la tolerancia otorgada a la normalización del estado, lo cual, podría resultar en que el estado que maximice la NLW sea distinto al que se encontró actualmente.

Dentro de los límites con los que se ha podido trabajar en este estudio, se ha observado que el hecho de incrementar alguno de los dos parámetros en el estado de fase cúbica (γ, r) incrementa la negatividad de la función de Wigner, siempre y cuando $\gamma \neq 0$, aunque el valor de este último es el que tiene más peso dentro de dicha negatividad.

Para mejorar los resultados obtenidos en el mapa de calor de la Fig. 12 podrían intentarse algunas cosas como escribir el código en otro lenguaje o, en lugar de utilizar métodos numéricos para aproximar el valor de cada función de Wigner, utilizar una expresión analítica. Otro camino puede ser investigar los estados de fase cúbica para otras distribuciones de cuasi-probabilidad, como las funciones de Husimi y Glauber-Sudarshan.

A. Códigos

Para poder ejecutar los códigos, escritos en Python, es necesario instalar la librería *QuTiP* mediante:

```
pip install qutip
```

A.1. Comparación entre *QuTiP* y expansión en serie

Para realizar las figuras 4, 5 y 7 se ha utilizado el siguiente código, aunque con los valores de $N = 35, 15$ y 400 , respectivamente:

```
import numpy as np
import matplotlib.pyplot as plt
from qutip import *
from scipy.special import factorial

N = 35
gamma = 0.1
r = 0.6
theta = np.pi
c = 1/np.sqrt(2)
xvec = np.linspace(-6, 6, 100)
yvec = np.linspace(-5, 26, 100)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 8))
plt.subplots_adjust(wspace=0.3)

# Gráfica 1: Cálculo directo con QuTiP
a = destroy(N)
adag = a.dag()
x = c * (a + adag)

Gamma = (1j * gamma * x**3).expm()
xi = r * np.exp(1j * theta)
S = squeeze(N, xi)
vacuum = basis(N, 0)
state = Gamma * S * vacuum

W1 = wigner(state, xvec, yvec, method='iterative')
W1_max = np.max(W1).round(4)
W1_min = np.min(W1).round(4)

cont1 = ax1.contourf(xvec, yvec, W1, 100, cmap="seismic",
                      levels=np.linspace(-1/np.pi, 1/np.pi, 400))
ax1.set_title(f'Cálculo directo con QuTiP\n\ngamma = {gamma}$, $r = {r}$', pad=15)
ax1.set_xlabel('$q$', fontsize=12)
ax1.set_ylabel('$p$', fontsize=12)

ax1.text(0.05, 0.95,
         f"Max: {W1_max}\nMin: {W1_min}",
         transform=ax1.transAxes,
         color='white',
         fontsize=12,
         verticalalignment='top',
         bbox=dict(facecolor='black', alpha=0.5))

ax1.text(-0.1, 1.05, '(a)', transform=ax1.transAxes,
         fontsize=16, fontweight='bold', va='top')

# Gráfica 2: Aproximación por expansión en serie
m_max = 5
j_max = 150
N_approx = 35
```

```

def squeezed_vacuum_state(r, theta, m_max, N):
    states = []
    coefficients = []
    for m in range(m_max + 1):
        num = (-1)**m * np.sqrt(factorial(2*m))
        den = 2**m * factorial(m)
        coeff = num / den * np.exp(1j * m * theta) * (np.tanh(r))**m
        coeff /= np.sqrt(np.cosh(r))
        coefficients.append(coeff)
        state = basis(N, 2*m)
        states.append(state)
    return sum([coeff * state for coeff, state in zip(coefficients, states)])

def gamma_operator_expansion(gamma, j_max, N):
    a = destroy(N)
    adag = a.dag()
    x = c * (a + adag)
    gamma_op = 0
    for j in range(j_max + 1):
        term = (1j * gamma * x**3)**j / factorial(j)
        gamma_op += term
    return gamma_op

xi_state = squeezed_vacuum_state(r, -np.pi, m_max, N_approx)
gamma_op = gamma_operator_expansion(gamma, j_max, N_approx)
gamma_r_state = gamma_op * xi_state

W2 = wigner(gamma_r_state, xvec, yvec, method='iterative')
W2_max = np.max(W2).round(4)
W2_min = np.min(W2).round(4)

cont2 = ax2.contourf(xvec, yvec, W2, 100, cmap="seismic",
                      levels=np.linspace(-1/np.pi, 1/np.pi, 400))
ax2.set_title(f'Aproximación por expansión en serie\n$\gamma = \{gamma\}$, $r = \{r\}$', pad=15)
ax2.set_xlabel('$q$', fontsize=12)
ax2.set_ylabel('$p$', fontsize=12)

ax2.text(0.05, 0.95,
         f"Max: {W2_max}\nMin: {W2_min}",
         transform=ax2.transAxes,
         color='white',
         fontsize=12,
         verticalalignment='top',
         bbox=dict(facecolor='black', alpha=0.5))

ax2.text(-0.1, 1.05, '(b)', transform=ax2.transAxes,
         fontsize=16, fontweight='bold', va='top')

cbar_ax = fig.add_axes([0.25, 0.01, 0.5, 0.03])
cbar = fig.colorbar(cont1, cax=cbar_ax, orientation='horizontal',
                    label="$W(q,p)$", extend="both")
cbar.set_ticks([-1/np.pi, -0.2, -0.1, 0, 0.1, 0.2, 1/np.pi])
cbar.set_ticklabels(['-$1/\pi$', '-0.2', '-0.1', '0', '0.1', '0.2', '$1/\pi$'])

plt.show()

```

A.2. NLW e integrales asociadas para (γ, r) específicas.

```

import numpy as np
import qutip as qt

def wigner_log_negativity_sum(state, xvec, yvec):
    W = qt.wigner(state, xvec, yvec)

    dx = xvec[1] - xvec[0]
    dy = yvec[1] - yvec[0]

    I_abs = np.sum(np.abs(W)) * dx * dy
    I_total = np.sum(W) * dx * dy

    WLN = np.log2(I_abs)

    return WLN, I_abs, I_total

N = 400
gamma = 0.1
r = 1.2
theta = np.pi
c = 1 / np.sqrt(2)

xvec = np.linspace(-5, 5, 100)
yvec = np.linspace(-5, 25, 100)

a = qt.destroy(N)
adag = a.dag()
x = c * (a + adag)

Gamma = (1j * gamma * x**3).expm()
xi = r * np.exp(1j * theta)
S = qt.squeeze(N, xi)
vacuum = qt.basis(N, 0)
state = Gamma * S * vacuum

WLN, I_abs, I_total = wigner_log_negativity_sum(state, xvec, yvec)

print(f"Negatividad Logarítmica de Wigner (WLN): {WLN}")
print(f"Integral |W(q, p)|: {I_abs} (Debe ser > 1 si hay negatividad)")
print(f"Integral W(q, p): {I_total} (Debe ser 1)")

```

A.3. Funciones $W(q, p)$ bajo el efecto de variar γ ó r

Las figuras 8 y 9 se han realizado mediante el siguiente código, con ligeras variaciones en las listas que contienen los valores de γ y r , según el parámetro que se mantiene fijo y el que varía:

```

import numpy as np
import matplotlib.pyplot as plt
from qutip import *

N = 400
theta = np.pi
c = 1/np.sqrt(2)
xvec = np.linspace(-6, 6, 100)
yvec = np.linspace(-5, 26, 100)

# Parámetros a variar
gamma_groups = [0.1, 0.3, 0.4]
r_values = [0.0, 0.1, 0.6, 1.2]

labels = [f'({chr(97 + i)})' for i in range(12)]

fig, axes = plt.subplots(3, 4, figsize=(20, 12))
plt.subplots_adjust(wspace=0.4, hspace=0.5)

a = destroy(N)
adag = a.dag()
x = c * (a + adag)

for row, gamma in enumerate(gamma_groups):
    for col, r in enumerate(r_values):
        ax = axes[row, col]
        label_idx = row * 4 + col

        vacuum = basis(N, 0)

        if r == 0.0:
            state = vacuum
            title = f'Vacío\n($\gamma = 0$, $r = 0$)'
        else:
            Gamma = (1j * gamma * x**3).expm()
            xi = r * np.exp(1j * theta)
            S = squeeze(N, xi)
            state = Gamma * S * vacuum
            title = f'$\gamma = {gamma}$, $r = {r}$'

        W = wigner(state, xvec, yvec, method='clenshaw')
        W_max = np.max(W).round(4)
        W_min = np.min(W).round(4)

        cont = ax.contourf(xvec, yvec, W, 100, cmap="seismic",
                            levels=np.linspace(-1/np.pi, 1/np.pi, 400))
        ax.set_title(title, pad=15)
        ax.set_xlabel('$q$', labelpad=10, fontsize=12)
        ax.set_ylabel('$p$', labelpad=10, fontsize=12)

        if r == 0.0:
            ax.set_xlim(-6, 6)
            ax.set_ylim(-6, 6)
        else:
            ax.set_xlim(-6, 6)
            ax.set_ylim(-5, 26)

        ax.text(0.05, 0.85,
                f"Max: {W_max}\nMin: {W_min}",
                transform=ax.transAxes,
                color='white',

```

```
    fontsize=9,
    bbox=dic(facecolor='black', alpha=0.5))

ax.text(-0.1, 1.15, labels[label_idx], transform=ax.transAxes,
        fontsize=12, fontweight='bold', va='top', ha='right')

cbar_ax = fig.add_axes([0.25, 0.03, 0.5, 0.02])
cbar = fig.colorbar(cont, cax=cbar_ax, orientation='horizontal', label="$W(q,p)$")
cbar.ax.set_xlabel("$W(q,p)$", fontsize=12)
cbar.set_ticks([-1/np.pi, -0.2, -0.1, 0, 0.1, 0.2, 1/np.pi])
cbar.set_ticklabels(['-$1/\pi$', '-0.2', '-0.1', '0', '0.1', '0.2', '$1/\pi'], fontsize=12)

plt.show()
```

A.4. Evolución de la NLW en función de N

Las figuras 10 y 11 se han hecho a partir del siguiente código, variando únicamente los valores de los parámetros (γ, r):

```

import numpy as np
import qutip as qt
import matplotlib.pyplot as plt

def wigner_log_negativity_sum(state, xvec, yvec):
    W = qt.wigner(state, xvec, yvec)
    dx = xvec[1] - xvec[0]
    dy = yvec[1] - yvec[0]
    I_abs = np.sum(np.abs(W)) * dx * dy
    I_total = np.sum(W) * dx * dy
    WLN = np.log2(I_abs)
    return WLN, I_abs, I_total

gamma = 0.5
r = 1.7
theta = np.pi
c = 1 / np.sqrt(2)
xvec = np.linspace(-6, 6, 100)
yvec = np.linspace(-5, 26, 100)

N_values = np.arange(5, 405, 5)
WLN_list = []
I_abs_list = []
I_total_list = []

for N in N_values:
    try:
        a = qt.destroy(N)
        adag = a.dag()
        x = c * (a + adag)

        Gamma = (1j * gamma * x**3).expm()
        xi = r * np.exp(1j * theta)
        S = qt.squeeze(N, xi)
        vacuum = qt.basis(N, 0)
        state = Gamma * S * vacuum

        WLN, I_abs, I_total = wigner_log_negativity_sum(state, xvec, yvec)
        WLN_list.append(WLN)
        I_abs_list.append(I_abs)
        I_total_list.append(I_total)

        print(f"N = {N}: WLN = {WLN:.6f}, I_abs = {I_abs:.6f}, I_total = {I_total:.6f}")
    except Exception as e:
        print(f"Error en N = {N}: {str(e)}")
        WLN_list.append(np.nan)
        I_abs_list.append(np.nan)
        I_total_list.append(np.nan)

fig, ax1 = plt.subplots(figsize=(12, 6))

color_WLN = 'tab:blue'
ax1.set_xlabel('Dimensión del espacio (N)', fontsize=12)
ax1.set_ylabel('Negatividad Logarítmica de Wigner ($NLW$)', color=color_WLN, fontsize=12)
ax1.plot(N_values, WLN_list, 'o-', color=color_WLN, label='$NLW$', markersize=5)
ax1.tick_params(axis='y', labelcolor=color_WLN)
ax1.grid(True, linestyle='--', alpha=0.6)

ax2 = ax1.twinx()
color_I_abs = 'tab:red'
ax2.set_ylabel('Integral de $|W(q, p)|$', color=color_I_abs, fontsize=12)
ax2.plot(N_values, I_abs_list, 's-', color=color_I_abs, label='$|W(q, p)|$', markersize=5)
ax2.tick_params(axis='y', labelcolor=color_I_abs)

```

```
ax3 = ax1.twinx()
color_I_total = 'tab:green'
ax3.spines['right'].set_position(('outward', 60))
ax3.set_ylabel('Integral de  $W(q, p)$ ', color=color_I_total, fontsize=12)
ax3.plot(N_values, I_total_list, 'd-', color=color_I_total, label=' $W(q, p)$ ', markersize=5)
ax3.tick_params(axis='y', labelcolor=color_I_total)

ax1.set_ylim(0, max(WLN_list) * 1.1)
ax2.set_ylim(1, max(I_abs_list) * 1.1)
ax3.set_ylim(min(I_total_list), 1.01)

plt.title(f'$NLW$ del estado $|\gamma$, $r \rangle$ con $\gamma = \{gamma\}$ y $r = \{r\}$',
          fontsize=14)
fig.legend(loc=(0.6, 0.9), bbox_to_anchor=(0.9, 0.9))
fig.tight_layout()
plt.show()
```

A.5. Mapa de calor de $NLW(\gamma, r)$

El mapa de calor de la fig. I2 se ha realizado con:

```

import numpy as np
import qutip as qt
import matplotlib.pyplot as plt
import seaborn as sns
from tqdm import tqdm
import pandas as pd

N = 400
theta = np.pi
c = 1/np.sqrt(2)
xvec = np.linspace(-6, 6, 100)
yvec = np.linspace(-5, 26, 100)

gamma_values = np.linspace(0.01, 0.5, 50)
r_values = np.linspace(0.1, 1.7, 17)

NLW_matrix = np.zeros((len(gamma_values), len(r_values)))
I_total_matrix = np.zeros_like(NLW_matrix)
I_abs_matrix = np.zeros_like(NLW_matrix)
Wigner_values = {}

def calcs(N, gamma, r, theta, xvec, yvec):
    try:
        a = qt.destroy(N)
        x = c*(a + a.dag())
        state = (1j*gamma*(x**3)).expm() * qt.squeeze(N, r*np.exp(1j*theta)) * qt.basis(N,0)
        W = qt.wigner(state, xvec, yvec, method='clenshaw')
        dxdy = (xvec[1]-xvec[0])*(yvec[1]-yvec[0])
        return np.log2(np.sum(np.abs(W)*dxdy), np.sum(W)*dxdy, np.sum(np.abs(W))*dxdy, W)
    except:
        return np.nan, np.nan, np.nan, np.nan

for i, gamma in enumerate(tqdm(gamma_values, desc="Calculando métricas")):
    for j, r in enumerate(r_values):
        NLW_matrix[i,j], I_total_matrix[i,j], I_abs_matrix[i,j], W = calcs(N, gamma,
                                                                           r, theta,
                                                                           xvec, yvec)
        Wigner_values[f"gamma_{gamma:.4f}_r_{r:.4f}"] = W

data = []
for i, gamma in enumerate(gamma_values):
    for j, r in enumerate(r_values):
        data.append({
            'gamma': gamma,
            'r': r,
            'NLW': NLW_matrix[i,j],
            'Integral_W': I_total_matrix[i,j],
            'Integral_abs_W': I_abs_matrix[i,j],
            'Wigner_values_key': f"gamma_{gamma:.4f}_r_{r:.4f}"
        })

df = pd.DataFrame(data)
df.to_csv('ValoresNLW.csv', index=False)

np.savez('Wigner_values.npz', **Wigner_values)

tolerance = 0.02
valid_I_total = np.abs(I_total_matrix - 1) < tolerance

lab1 = '$\gamma=0.1, r=0.6$ (Brunelli)'
lab2 = '$\gamma=0.4, r=0.8$ (Brunelli)'
highlight_states = [
    {'gamma': 0.1, 'r': 0.6, 'color': 'gray', 'linestyle': '--', 'label': lab1},
    {'gamma': 0.4, 'r': 0.8, 'color': 'gray', 'linestyle': '-', 'label': lab2}
]

```

```

if np.any(valid_I_total):
    valid_NLW = np.where(valid_I_total, NLW_matrix, -np.inf)
    max_NLW = np.nanmax(valid_NLW)
    max_idx = np.where((NLW_matrix == max_NLW) & valid_I_total)
    opt_gamma, opt_r = gamma_values[max_idx[0][0]], r_values[max_idx[1][0]]
    opt_I_total = I_total_matrix[max_idx[0][0], max_idx[1][0]]
    highlight_states.append({
        'gamma': opt_gamma,
        'r': opt_r,
        'color': 'red',
        'linestyle': '-',
        'label': 'Óptimo global',
        'NLW': max_NLW,
        'I_total': opt_I_total
    })
else:
    max_NLW, opt_gamma, opt_r, opt_I_total = np.nan, np.nan, np.nan, np.nan

gamma_mask = (gamma_values >= 0.01) & (gamma_values <= 0.11)
r_mask = (r_values >= 0.1) & (r_values <= 1.7)
experimental_mask = np.outer(gamma_mask, r_mask)

if np.any(valid_I_total & experimental_mask):
    valid_exp_NLW = np.where(valid_I_total & experimental_mask, NLW_matrix, -np.inf)
    max_exp_NLW = np.nanmax(valid_exp_NLW)
    max_exp_idx = np.where((NLW_matrix == max_exp_NLW) & valid_I_total & experimental_mask)
    opt_exp_gamma, opt_exp_r = gamma_values[max_exp_idx[0][0]], r_values[max_exp_idx[1][0]]
    opt_exp_I_total = I_total_matrix[max_exp_idx[0][0], max_exp_idx[1][0]]
    highlight_states.append({
        'gamma': opt_exp_gamma,
        'r': opt_exp_r,
        'color': 'green',
        'linestyle': '-',
        'label': 'Óptimo experimental',
        'NLW': max_exp_NLW,
        'I_total': opt_exp_I_total
    })

plt.figure(figsize=(14, 9))
cmap = sns.color_palette("viridis", as_cmap=True)
cmap.set_bad(color='lightgrey')

plot_matrix = np.where(valid_I_total, NLW_matrix, np.nan)

ax = sns.heatmap(
    plot_matrix,
    xticklabels=[f"{r:.2f}" for r in r_values],
    yticklabels=[f"{g:.2f}" for g in gamma_values],
    cmap=cmap,
    annot=False,
    cbar_kws={'label': 'Negatividad Logarítmica de Wigner ($NLW$)'}
)

txt1 = "Negatividad Logarítmica de Wigner ($NLW$) "
txt2 = "para distintas combinaciones de ($\gamma, r$)"
plt.title(txt1, pad=20, fontsize=12)
plt.xlabel("Amplitud de compresión ($r$)", fontsize=12)
plt.ylabel("Cubicidad ($\gamma$)", fontsize=12)

for state in highlight_states:
    if 'gamma' in state and 'r' in state:
        i = np.where(gamma_values >= state['gamma'])[0][0]
        j = np.where(r_values >= state['r'])[0][0]

        edgecolor = state['color']
        linestyle = state.get('linestyle', '-')
        linewidth = 2 if linestyle == '-' else 1.5

        ax.add_patch(plt.Rectangle(

```

```
(j, i), 1, 1,
fill=False,
edgecolor=edgecolor,
linestyle=linestyle,
linewidth=linewidth,
label=state.get('label', ''))

info_text = ""
for state in highlight_states[-2:]:
    if 'NLW' in state:
        info_text += (f"{state['label']}:\n"
                      f"${NLW}: ${state['NLW']:.3f}\n"
                      f"${\gamma} = {state['gamma']:.2f}\n"
                      f"${r} = {state['r']:.2f}\n"
                      f"${int W(q,p)dqdp} = {state['I_total']:.3f}\n\n")

plt.text(1.05, 0.15,
         info_text,
         transform=ax.transAxes,
         color='white',
         ha='right', va='center',
         bbox=dict(facecolor='black', alpha=0.4))

plt.legend(bbox_to_anchor=(1.05, 1.1), loc='upper right')
plt.tight_layout()
plt.show()
```

A.6. Evolución de NLW(γ) y NLW(r)

Las figuras [13] y [14] se han hecho a partir del siguiente código:

```

import numpy as np
import qutip as qt
import matplotlib.pyplot as plt
from tqdm import tqdm

N = 400
theta = np.pi
c = 1/np.sqrt(2)
xvec = np.linspace(-6, 6, 100)
yvec = np.linspace(-5, 26, 100)

def calculate_WLN(N, gamma, r, theta, xvec, yvec):
    try:
        a = qt.destroy(N)
        x = c*(a + a.dag())
        state = (1j*gamma*(x**3)).expm() * qt.squeeze(N, r*np.exp(1j*theta)) * qt.basis(N, 0)
        W = qt.wigner(state, xvec, yvec)
        dxdy = (xvec[1]-xvec[0])*(yvec[1]-yvec[0])
        return np.log2(np.sum(np.abs(W)) *dxdy)
    except:
        return np.nan

# 1. Gráfico con gamma
gamma_values = [0.1, 0.5, 1]
r_range = np.linspace(0.1, 1.7, 33)

series_config = {
    0: { "# = 0.1",
         'color': 'b',
         'marker': 'o',
         'special_sections': [
             {'start': 1.4, 'end': 1.7, 'color': 'lightgray', 'marker': 'o'}
         ]
     },
    1: { "# = 0.5",
         'color': 'g',
         'marker': 'o',
         'special_sections': [
             {'start': 1.1, 'end': 1.7, 'color': 'lightgray', 'marker': 'o'}
         ]
     },
    2: { "# = 1.0",
         'color': 'r',
         'marker': 'o',
         'special_sections': [
             {'start': 0.6, 'end': 1.7, 'color': 'lightgray', 'marker': 'o'}
         ]
     }
}

plt.figure(figsize=(10, 6))

print("\nCalculando WLN vs r para diferentes ...")
for idx, gamma in enumerate(gamma_values):
    WLN_vs_r = []
    for r in tqdm(r_range, desc=f' = {gamma}'):
        wln = calculate_WLN(N, gamma, r, theta, xvec, yvec)
        WLN_vs_r.append(wln)

    config = series_config[idx]

    plt.plot(r_range, WLN_vs_r, f'{config["marker"]}-',
              color=config['color'], label=f' = {gamma}')

    if config['special_sections'] is not None:

```

```

for section in config['special_sections']:
    mask = (r_range >= section['start']) & (r_range <= section['end'])
    if any(mask):
        plt.plot(r_range[mask], np.array(WLN_vs_r)[mask],
                  f"{section['marker']}-", color=section['color'])

plt.axhline(0, color='k', linestyle='--', alpha=0.5)
plt.xlabel("Amplitud de compresión ($r$)", fontsize=12)
plt.ylabel("Negatividad Logarítmica de Wigner ($NLW$)", fontsize=12)
plt.title("Evolución de la $NLW$ vs $r$ para diferentes $\gamma$", fontsize=14)
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

# 2. Gráfico con r fijo
r_values = [0.6, 0.9, 1.3]
gamma_range = np.linspace(0.1, 1.0, 19)

series_config_2 = {
    0: { # r = 0.6
        'color': 'purple',
        'marker': 'o',
        'special_sections': [
            {'start': 0.82, 'end': 1.0, 'color': 'lightgray', 'marker': 'o'}
        ]
    },
    1: { # r = 0.9
        'color': 'orange',
        'marker': 'o',
        'special_sections': [
            {'start': 0.63, 'end': 1.0, 'color': 'lightgray', 'marker': 'o'}
        ]
    },
    2: { # r = 1.3
        'color': 'brown',
        'marker': 'o',
        'special_sections': [
            {'start': 0.18, 'end': 0.27, 'color': 'lightgray', 'marker': 'o'},
            {'start': 0.31, 'end': 1.0, 'color': 'lightgray', 'marker': 'o'}
        ]
    }
}

plt.figure(figsize=(10, 6))

print("\nCalculando WLN vs para diferentes r...")
for idx, r in enumerate(r_values):
    WLN_vs_gamma = []
    for gamma in tqdm(gamma_range, desc=f'r = {r}'):
        wln = calculate_WLN(N, gamma, r, theta, xvec, yvec)
        WLN_vs_gamma.append(wln)

    config = series_config_2[idx]

    plt.plot(gamma_range, WLN_vs_gamma, f'{config["marker"]}-',
              color=config['color'], label=f'r = {r}')

    if config['special_sections'] is not None:
        for section in config['special_sections']:
            mask = (gamma_range >= section['start']) & (gamma_range <= section['end'])
            if any(mask):
                plt.plot(gamma_range[mask], np.array(WLN_vs_gamma)[mask],
                          f'{section["marker"]}-',
                          color=section['color'])

plt.axhline(0, color='k', linestyle='--', alpha=0.5)
plt.xlabel("Cubicidad ($\gamma$)", fontsize=12)
plt.ylabel("Negatividad Logarítmica de Wigner ($NLW$)", fontsize=12)

```

```
plt.title("Evolución de la $NLW$ vs $\gamma$ para diferentes $r$", fontsize=14)
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```

Referencias

- [1] C. C. Gerry and P. L. Knight. *Introductory Quantum Optics*. Cambridge University Press, 2005.
- [2] D. F. Walls and G. J. Milburn. *Quantum Optics, 2nd edition*. Springer, 2008.
- [3] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd. Gaussian quantum information. *Rev. Mod. Phys.*, 84:621–669, 2012.
- [4] S. D. Bartlett, B. C. Sanders, S. L. Braunstein, and K. Nemoto. Efficient classical simulation of continuous variable quantum information processes. *Phys. Rev. Lett.*, 88:097904, 2002.
- [5] F. Albarelli, M. G. Genoni, M. G. A. Paris, and A. Ferraro. Resource theory of quantum non-gaussianity and wigner negativity. *Phys. Rev. A*, 98:052350, 2018.
- [6] S. L. Braunstein and A. K. Pati. *Quantum Information with Continuous Variables*. Kluwer Academic Publishers, 2003.
- [7] Gerardo Adesso, Sammy Ragy, and Antony R. Lee. Continuous variable quantum information: Gaussian states and beyond. *Open Systems & Information Dynamics*, 21(01n02):1440001, 2014.
- [8] R.L. Hudson. When is the wigner quasi-probability density non-negative? *Reports on Mathematical Physics*, 6(2):249–252, 1974.
- [9] U. Leonhardt. *Measuring the Quantum State of Light*. Cambridge University Press, 1997.
- [10] Daniel Gottesman, Alexei Kitaev, and John Preskill. Encoding a qubit in an oscillator. *Physical Review A*, 64(1):012310, 2001.
- [11] Pengfei Wang, Chun-Yang Luan, Mu Qiao, Mark Um, Junhua Zhang, Ye Wang, Xiao Yuan, Mile Gu, Jingning Zhang, and Kihwan Kim. Single ion qubit with estimated coherence time exceeding one hour. *Nat. Commun.*, 12:233, 2021.
- [12] Y. Zheng, O. Hahn, P. Stadler, P. Holmvall, F. Quijandría, A. Ferraro, and G. Ferrini. Gaussian conversion protocols for cubic phase state generation. *PRX Quantum*, 2:010327, 2021.
- [13] Sophie Choe. Quantum computing overview: discrete vs. continuous variable models. arXiv:2206.07246, 2022.
- [14] S. L. Braunstein and P. van Loock. Quantum information with continuous variables. *Rev. Mod. Phys.*, 77:513–577, 2005.
- [15] R. Takagi and Q. Zhuang. Convex resource theory of non-gaussianity. *Phys. Rev. A*, 97:062337, 2018.

- [16] Dusan Zivkov, Istvan Papp, Daniel Kurtjak, Marko Grumic, and Marko Ivanovic. Application engine for resource-constrained embedded platforms. In *2015 IEEE 1st International Workshop on Consumer Electronics (CE WS)*, pages 48–51, 2015.
- [17] E. Chitambar and G. Gour. Quantum resource theories. *Rev. Mod. Phys.*, 91, 2019.
- [18] F. G. S. L. Brandão and G. Gour. The general structure of quantum resource theories. *Phys. Rev. Lett.*, 115:070503, 2015.
- [19] M. Horodecki and J. Oppenheim. (quantumness in the context of) resource theories. *Int. J. Mod. Phys. B*, 27(1-3), 2013.
- [20] B. Coecke, T. Fritz, and R. W. Spekkens. A mathematical theory of resources. *Inf. Comput.*, 250:59–86, 2016.
- [21] J. Yoshikawa, S. Yokoyama, T. Kaji, C. Sornphiphatphong, Y. Shiozawa, K. Makino, and A. Furusawa. Invited article: Generation of one-million-mode continuous-variable cluster state by unlimited time-domain multiplexing. *APL Photonics*, 1:060801, 2016.
- [22] B. Q. Baragiola, G. Pantaleoni, R. N. Alexander, A. Karanjai, and N. C. Menicucci. All-gaussian universality and fault tolerance with the gottesman-kitaev-preskill code. *Phys. Rev. Lett.*, 123:200502, 2019.
- [23] M. Brunelli and O. Houhou. Linear and quadratic reservoir engineering of non-gaussian states. *Phys. Rev. A*, 100:013831, 2019.
- [24] Anatole Kenfack and Karol Życzkowski. Negativity of the wigner function as an indicator of non-classicality. *J. Opt. B: Quantum Semiclass. Opt.*, 6(10):396, 2004.
- [25] N. Lambert, E. Giguère, P. Menczel, B. Li, P. Hopf, G. Suárez, M. Gali, J. Lishman, R. Gadhvi, R. Agarwal, A. Galicia, N. Shammah, P. D. Nation, J. R. Johansson, S. Ahmed, S. Cross, A. Pitchford, and F. Nori. QuTiP 5: The quantum toolbox in Python, 2024.
- [26] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007. Sec. 5.4: Recurrence Relations and Clenshaw’s Recurrence Formula.
- [27] W. Morven Gentleman. Implementing clenshaw-curtis quadrature, i methodology and experience. *Commun. ACM*, 15(5):337–342, 1972.
- [28] A. V. Baeva, A. S. Losev, and I. V. Sokolov. Optical schrödinger cat states generation using cubic phase resource state and beamsplitter, 2025.
- [29] Axel M. Eriksson, Théo Sépulcre, Mikael Kervinen, Timo Hillmann, Marina Kudra, Simon Dupouy, Yong Lu, Maryam Khanahmadi, Jiaying Yang, Claudia Castillo-Moreno, Per Delsing, and Simone Gasparinetti. Universal control of a bosonic mode via drive-activated native cubic interactions. *Nat. Commun.*, 15(1):2512, 2024.